



UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE CUENCA
CARRERA DE MECATRÓNICA

**CONTROL DEL MOVIMIENTO REMOTO DEL BRAZO ROBÓTICO
MITSUBISHI USANDO EL SISTEMA OPERATIVO ROS A TRAVÉS DE
LA WEB**

Trabajo de titulación previo a la obtención
del título de Ingeniero en Mecatrónica

AUTORES: JUAN SEBASTIÁN SOLANO CRIOLLO

TUTOR: ING. MÓNICA ALEXANDRA ROMERO SACOTO

Cuenca – Ecuador

2023

CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN

Yo, Juan Sebastián Solano Criollo con documento de identificación N° 0106439920 manifiesto que:

Soy el autor y responsable del presente trabajo; y, autorizo a que sin fines de lucro la Universidad Politécnica Salesiana pueda usar, difundir, reproducir o publicar de manera total o parcial el presente trabajo de titulación.

Cuenca, 8 de marzo del 2023

Atentamente,



Juan Sebastián Solano Criollo
0106439920

CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE TITULACIÓN A LA UNIVERSIDAD POLITÉCNICA SALESIANA

Yo, Juan Sebastián Solano Criollo con documento de identificación N° 0106439920, expreso mi voluntad y por medio del presente documento cedo a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que soy autor del Proyecto Técnico: "Control del movimiento remoto del brazo robótico Mitsubishi usando el sistema operativo ROS a través de la web", el cual ha sido desarrollado para optar por el título de: Ingeniero en Mecatrónica, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En concordancia con lo manifestado, suscribo este documento en el momento que hago la entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.

Cuenca, 8 de marzo del 2023

Atentamente,



Juan Sebastián Solano Criollo

0106439920

CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN

Yo, Mónica Alexandra Romero Sacoto con documento de identificación N° 0302209911, docente de la Universidad Politécnica Salesiana, declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: CONTROL DEL MOVIMIENTO REMOTO DEL BRAZO ROBÓTICO MITSUBISHI USANDO EL SISTEMA OPERATIVO ROS A TRAVES DE LA WEB, realizado por Juan Sebastián Solano Criollo con documento de identificación N° 0106439920, obteniendo como resultado final el trabajo de titulación bajo la opción Proyecto Técnico que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Cuenca, 8 de marzo del 2023

Atentamente,



Ing. Mónica Alexandra Romero Sacoto, MSc.
0302209911

Dedicatoria

Juan Sebastián

Quiero dedicar el presente trabajo de titulación a mis padres René y Patricia, quienes han sido pilar fundamental para alcanzar este objetivo tan importante en mi vida gracias a su amor fundamental y apoyo en los momentos más difíciles del trayecto.

Finalmente, dedico a las personas que formaron parte del proceso y me brindaron siempre su apoyo de una u otra forma para alcanzar este objetivo.

Agradecimientos

Juan Sebastián

Agradezco principalmente a Dios por permitirme llegar a cumplir esta meta tan importante en mi vida, por brindarme salud, fuerza y sabiduría a lo largo del proceso y no dejarme caer en los momentos más difíciles.

Agradezco a mis padres René y Patricia por el constante apoyo, comprensión y cariño que me brindaron durante toda mi vida personal como institucional. Gracias a mi madre por darme siempre su apoyo y su guía en cada situación de mi vida universitaria y personal, por escucharme y siempre darme consejos para mi vida; a mi padre por enseñarme el trabajo y el esfuerzo, a valorar los resultados conseguidos por los mismos y nunca dejarme vencer por las adversidades. De igual manera quiero agradecer a mis abuelitos Román, Florencia, Gerardo, Elsa y a mi tía Sandra, por su amor, apoyo incondicional y por darme grandes enseñanzas durante toda mi infancia.

Un agradecimiento especial también a mi Tutora de Proyecto de Titulación Ing. Mónica Romero, por brindarme inspiración y sus conocimientos profesionales para la culminación de este trabajo de titulación.

Este documento fue realizado enteramente en L^AT_EX

Índice

Certificado de responsabilidad y autoría del trabajo de titulación	I
Certificado de cesión de derechos de autor del trabajo de titulación a la Universidad Politécnica Salesiana	II
Certificado de dirección del trabajo de titulación	III
Dedicatoria	IV
Agradecimientos	V
Resumen	XIII
Abstract	XIV
1. Introducción	1
2. Problema	2
2.1. Antecedentes	2
2.2. Descripción del problema	2
2.3. Importancia y alcances	3
2.4. Delimitación	3
2.4.1. Espacial o geográfica	3
2.4.2. Temporal	4
2.4.3. Sectorial o institucional	4
2.5. Problema General	4
2.6. Problemas Específicos	5
3. Objetivos	5
3.1. Objetivo General	5
3.2. Objetivos Específicos	5
4. Hipótesis	6
4.1. Hipótesis General	6
4.2. Hipótesis Específicas	6

5. Marco Teórico	6
5.1. Robótica	6
5.2. Robot Industrial	7
5.2.1. Robot manipulador	7
5.2.2. Robot cartesiano	8
5.2.3. Robots industriales Scara	8
5.2.4. 6 grados de libertad	8
5.2.5. Tipos de robots redundantes	9
5.2.6. Máquinas de doble brazo	9
5.3. Brazo Antropomórfico	9
5.4. Brazo robótico Mitsubishi RV-2AJ	10
5.5. Cinemática de robots antropomórficos	12
5.6. Ubuntu	12
5.6.1. Sistema Operativo Robótico (ROS)	12
5.6.2. Gazebo	13
5.6.3. MoveIt	15
5.7. Acceso remoto	17
5.8. Entorno virtual	17
5.8.1. VMware	18
6. Marco metodológico	19
6.1. Fundamento matemático acerca de la cinemática del robot antropomórfico Mitsubishi RV-2AJ	19
6.1.1. Análisis de grados de libertad	19
6.1.2. Cinemática directa del robot Mitsubishi RV-2AJ	20
6.1.3. Cinemática inversa del robot Mitsubishi RV-2AJ	25
6.1.4. Área de trabajo del robot Mitsubishi RV-2AJ	27
6.2. Modelado del robot Mitsubishi RV-2AJ en SolidWorks	28
6.2.1. Modelo del robot antropomórfico	28
6.2.2. Ensamblaje de modelo robot Mitsubishi RV-2AJ en SolidWorks	34
6.2.3. Conversión del ensamble del robot Mitsubishi RV-2AJ desde SolidWorks a formato URDF	35
6.3. Interfaz gráfica para visualización de trayectorias del robot	37
6.3.1. Creación de una máquina virtual	37
6.3.2. Robot Operating System(ROS)	38

6.3.3. Interfaz MoveIt	40
6.3.4. Workspace para configuración del modelo de ensamble	41
6.3.5. Modelo representado en el software Gazebo	42
6.3.6. Creación de posiciones en el software Gazebo	43
6.4. Creación de repositorio GitHub	46
6.5. Creación del entorno web en la plataforma TheConstructSim	48
7. Resultados	51
7.1. Modelo del robot Mitsubishi RV-2AJ planteado en el software SolidWorks . .	51
7.2. Creación de posiciones específicas para el robot Mitsubishi RV-2AJ en el software MoveIt	51
7.3. Planificación de trayectorias mediante el software RVIZ	57
7.4. Simulación Web en plataforma TheConstructSim	58
8. Cronograma y actividades	62
8.1. Presupuesto	64
9. Conclusiones	64
10. Recomendaciones	65
Referencias	67
ANEXOS	68
Anexo A: Matriz de Consistencia Lógica	69
Anexo B: Dibujo del modelo realizado en SolidWorks	70

Lista de Tablas

- 1. Características del brazo robótico Mitsubishi RV-2AJ 11
- 2. Parámetros del robot Mitsubishi para determinar los grados de libertad . . . 20
- 3. Parámetros del robot Mitsubishi para determinar la cinemática directa mediante el método Denavit-Hartenberg 21
- 4. Posiciones especificadas para el robot Mitsubishi RV-2AJ. 56
- 5. Cronograma de actividades 63
- 6. Recurso de talento humano 64
- 7. Matriz de consistencia 69

Lista de Figuras

1.	Universidad Politécnica Salesiana	4
2.	Robot Manipulador	8
3.	Brazo Antropomórfico	9
4.	Brazo robótico Mitsubishi RV-2AJ	10
5.	Características del Sistema Operativo Robótico(ROS).	13
6.	Entorno realizado en Gazebo.	14
7.	Entorno virtual de movimiento con MoveIt.	16
8.	Acceso remoto.	17
9.	Entorno virtual.	18
10.	Oracle VMware.	19
11.	Dimensiones de eslabones del robot Mitsubishi RV-2AJ.	21
12.	Espacio de trabajo del robot Mitsubishi RV-2AJ.	28
13.	Base del robot Mitsubishi RV-2AJ.	29
14.	Primer eslabón del robot Mitsubishi RV-2AJ.	30
15.	Segundo eslabón del robot Mitsubishi RV-2AJ.	31
16.	Tercer eslabón del robot Mitsubishi RV-2AJ.	32
17.	Cuarto eslabón del robot Mitsubishi RV-2AJ.	33
18.	Quinto eslabón del robot Mitsubishi RV-2AJ.	33
19.	Gripper FESTO especial para el robot Mitsubishi RV-2AJ.	34
20.	Herramienta para exportar el modelo a formato URDF.	35
21.	Selección de eslabones con relación padre-hijo.	36
22.	Configuración de propiedades de las juntas del ensamble.	37
23.	Entorno en VMware para la creación de una máquina virtual.	38
24.	Verificación de funcionamiento de sistema operativo ROS.	39
25.	Interfaz de configuración de MoveIt.	41
26.	Entorno de visualización Gazebo.	43
27.	Ubicación del modelo en formato URDF.	44
28.	Modelo dentro del software MoveIt.	45
29.	Optimización de comprobación de colisiones.	46
30.	Creación de repositorio GitHub para modelo del robot.	47
31.	Creación de repositorio GitHub para la simulación del modelo.	47
32.	Código para la exportación de archivos hacia GitHub.	48
33.	Plataforma TheConstructSim.	49

34.	Copia de repositorio del modelo de robot antropomórfico.	50
35.	Copia de repositorio de la simulación del robot antropomórfico para creación de trayectorias.	50
36.	Modelo del robot antropomórfico en SolidWorks.	51
37.	Grupos de planificación brazo.	52
38.	Eslabones del grupo de planificación brazo.	53
39.	Pinzas del grupo de planificación hand.	54
40.	Creación de posiciones específicas del robot.	55
41.	Entorno RViz para planificación de trayectorias del robot.	58
42.	Clonación del repositorio del modelo de robot antropomórfico.	59
43.	Clonación del repositorio de la simulación del robot antropomórfico para creación de trayectorias.	59
44.	Software MoveIt dentro de la plataforma virtual TheConstructSim.	60
45.	Modelo del robot antropomórfico dentro de la plataforma virtual TheConstructSim.	61
46.	Software RViz dentro de la plataforma virtual TheConstructSim.	62

Resumen

El presente trabajo de titulación plantea la realización de un entorno virtual para el brazo robótico Mitsubishi RV-2AJ generado en la plataforma Linux con la ayuda del Sistema Operativo Robótico (ROS) en el cual se permite la visualización y configuración de un sistema previamente ensamblado en el software SolidWorks. De esta forma, se logra la determinación de puntos específicos establecidos por el usuario mediante el entorno MoveIt que hacen referencia a los lugares físicos de acción en donde se necesite trabajar con el robot antropomórfico generando diferentes trayectorias de movimiento planteadas en el sistema RViz para su posterior visualización a través de la web.

Como resultado se ha desarrollado una interfaz gráfica generada en RViz en donde se puede simular y generar trayectorias en función del espacio real de trabajo del robot Mitsubishi RV-2AJ y que permiten verificar los fundamentos matemáticos presentes en la cinemática del robot antropomórfico, por otro lado se logró realizar distintas posiciones de accionamiento del robot dentro de Moveit, mismas que serán preestablecidas por el usuario y que da paso a una amplia posibilidad de simulación del modelo dentro del espacio virtualizado.

Palabras clave: Robot antropomórfico, ROS, Entorno virtual, Interfaz gráfica.

Abstract

This thesis describes the development of a virtual environment for the Mitsubishi RV-2AJ robotic arm, by using Linux and the robot's own Robotic Operating System, which lets us visualize and configure, the previous system, which was ensembled using SolidWorks; by making use of the MoveIt virtual environment and the RViz system we can define movement, that the user of said environment will perform, as specific points in space that will in turn be interpreted as trajectories, that the anthropomorphic robot will traverse and those given orders will finally be transmitted, to be visualized, via the internet.

As a result a graphic interface was developed using RViz software, this system allows us to generate and simulate trajectories of the real workspace in which the Mitsubishi RV-2AJ robotic arm is located at; the software enables us to verify the mathematical theory involved in controlling a robotic arm; within MoveIt we achieved the control and execution of movement by the anthropomorphic robot, said movement can be personalized according to the users needs; this opens up the possibilities of further development of different programs by making use of the virtual environment.

Keywords: Development, Personalized, Anthropomorphic.

1. Introducción

Los ambientes industriales dan a conocer que cada día habrá retos los cuales pueden ser solventados gracias al estudio, desarrollando conocimiento y capacidades de resolución de problemas, por lo tanto, es necesario la creación de entornos de simulación que propongan procesos que se llevan a cabo en las áreas industriales.

Para el desarrollo profesional es necesario conocer el funcionamiento de procesos industriales en entornos robóticos, es por tanto que la Universidad Politécnica Salesiana cuenta con un laboratorio de sistemas de producción modular (MPS), el mismo que cuenta con varios módulos que simulan el trabajo de distintos procesos industriales; entre los cuales se tiene el robot antropomórfico Mitsubishi RV-2AJ, el cual para su correcto manejo nos hace referencia a conocer el funcionamiento y procesos que se puedan desarrollar con el robot, para lo cual se debe estudiar parámetros fundamentales de robótica industrial que permitan el control y simulación de movimientos del robot antropomórfico.

La creación de un entorno virtual para el robot Mitsubishi RV-2AJ será dada gracias al Sistema Operativo Robótico (ROS), el cual permite la creación de un sistema de control de movimientos y trayectorias gracias a sus softwares MoveIt y RViz, mismos que dan la facilidad para el usuario de crear distintos puntos de posicionamiento del robot. Esto a su vez, se puede manejar a través de la web con la ayuda de la plataforma TheConstructSim, la cual da la posibilidad de crear un entorno de ROS para posterior clonación de repositorios cargados en GitHub que contengan los archivos creados anteriormente del modelo del robot y de su respectiva simulación para creación de trayectorias en el sistema de Linux.

2. Problema

2.1. Antecedentes

La Universidad Politécnica Salesiana cuenta con un laboratorio de automatización de procesos de producción, el cual permite el aprendizaje con entornos de tipo industrial, este espacio es necesario para que los estudiantes tengan un enfoque industrializado en las áreas de mecánica, electrónica, robótica, automatización y programación. En el ámbito de automatización industrial, el laboratorio de sistema modular de producción (MPS), resulta ideal para la simulación de procesos referidos a líneas de producción. Entre los módulos con los que cuenta el laboratorio se destaca el robot Mitsubishi RV-2AJ, el cual es un brazo antropomórfico, cuya programación se desarrolla empleando el software CIROS, debido a que la Universidad Politécnica Salesiana cuenta con una sola licencia a su disposición, el uso del robot es limitado.

El robot antropomórfico no cuenta con un sistema de control remoto, lo que dificulta su versatilidad de trabajo fuera del laboratorio, por lo que, el robot se puede manejar únicamente si se está conectado al sistema propio del mecanismo y observando los procesos que se realizan de manera personal, además de no contar con una interfaz gráfica que permita el monitoreo y control de las acciones que realiza el robot cuando el operario o estudiante se encuentre en cualquier otro punto de acceso fuera de la Universidad Politécnica Salesiana.

En un trabajo de titulación previo realizado en la Universidad Politécnica Salesiana "PROGRAMACIÓN DE PROCESOS INDUSTRIALES UTILIZANDO EL ROBOT MITSUBISHI RV-2AJ Y EL TORNO CNC EMCO CONCEPT TURN 55" se realiza una conexión y manejo del robot empleando el software Python, lo que se considera un punto de partida pues garantiza que el robot puede conectarse con plataformas de software libre (Chocho y Cuenca, 2022).

2.2. Descripción del problema

El brazo robótico Mitsubishi RV-2AJ que forma parte del sistema de producción modular MPS, es una herramienta versátil que puede emplearse en varios procesos, sin embargo debido al contar con una sola licencia del software CIROS empleado para la programación del mismo su uso se vuelve limitado a un solo ordenador y operario.

Actualmente no puede realizar simulación del movimiento del robot planteando trayectorias definidas.

2.3. Importancia y alcances

La propuesta de realizar la conexión de los procesos en el laboratorio MPS de la Universidad Politécnica Salesiana en el ámbito de automatización industrial, pretende desarrollar una herramienta de simulación que recoja todas las características del robot Mitsubishi RV-2AJ con el fin de poder generar trayectorias de movimiento y monitorizarlas a través de la web.

El presente trabajo tiene como objetivo generar un entorno de simulación para el robot Mitsubishi RV-2AJ usando el sistema operativo ROS, permitiendo generar diferentes tipos de trayectorias y manejar dichos movimientos a través de la web, pudiendo realizarse la conexión con el robot de manera física.

2.4. Delimitación

2.4.1. Espacial o geográfica

El proyecto se llevará a cabo en la Universidad Politécnica Salesiana, ubicada en la ciudad de Cuenca provincia del Azuay, de manera específica en los laboratorios del área mecánica. Cabe recalcar que la información necesaria será recabada de otros trabajos de titulación, industrias y documentos relacionados al tema.

Figura 1

Universidad Politécnica Salesiana.



Nota: Universidad Politécnica Salesiana sede Cuenca.(SIHE (2016))

2.4.2. Temporal

Se considera un tiempo de 400 horas para la realización del trabajo de titulación, las cuales se contemplan en la tabla 4 referentes a la asignatura de TT1 y su posterior continuidad en la asignatura TT2.

2.4.3. Sectorial o institucional

El proyecto será realizado en la Universidad Politécnica Salesiana, en el ámbito de simulación de procesos industriales para la enseñanza.

2.5. Problema General

¿Es probable controlar de manera remota el movimiento del brazo robótico Mitsubishi RV-2AJ con el sistema operativo Ros a través de la web?

2.6. Problemas Específicos

- ¿Es posible establecer parámetros iniciales para el movimiento remoto del brazo robótico Mitsubishi RV-2AJ con el sistema operativo ROS a través de la web?
- ¿Será factible proponer el movimiento remoto del brazo robótico Mitsubishi RV-2AJ con el sistema operativo ROS a través de la web?
- ¿Se podrá plantear una plataforma de entorno virtual para el control del movimiento remoto del brazo robótico Mitsubishi RV-2AJ con el sistema operativo ROS a través de la web?
- ¿Es viable implementar la plataforma de entorno virtual para el control del movimiento remoto del brazo robótico Mitsubishi RV-2AJ con el sistema operativo ROS a través de la web?

3. Objetivos

3.1. Objetivo General

Controlar el movimiento del brazo robótico Mitsubishi RV-2AJ de manera remota con el sistema operativo ROS a través de la web.

3.2. Objetivos Específicos

- Establecer parámetros iniciales para el movimiento remoto del brazo robótico Mitsubishi RV-2AJ con el sistema operativo ROS a través de la web.
- Proponer el movimiento remoto del brazo robótico Mitsubishi RV-2AJ con el sistema operativo ROS a través de la web.
- Plantear una plataforma de entorno virtual para el control de movimiento remoto del brazo robótico Mitsubishi RV-2AJ con el sistema operativo ROS a través de la web.
- Implementar la plataforma de entorno virtual para el control del movimiento remoto del brazo robótico Mitsubishi RV-2AJ con el sistema operativo ROS a través de la web.

4. Hipótesis

4.1. Hipótesis General

- Se controlará de manera remota el movimiento del brazo robótico Mitsubishi RV-2AJ con el sistema operativo ROS a través de la web.

4.2. Hipótesis Específicas

- Se establecerá parámetros iniciales para el movimiento remoto del brazo robótico Mitsubishi RV-2AJ con el sistema operativo ROS a través de la web.
- Se propondrá el movimiento remoto del brazo robótico Mitsubishi RV-2AJ con el sistema operativo ROS a través de la web.
- Se planteará una plataforma de entorno virtual para el control del movimiento remoto del brazo robótico Mitsubishi RV-2AJ con el sistema operativo ROS a través de la web.
- Se implementará la plataforma de entorno virtual para el control del movimiento remoto del brazo robótico Mitsubishi RV-2AJ a través de la web.

5. Marco Teórico

5.1. Robótica

La robótica es un área interdisciplinaria formada por la ingeniería mecánica, eléctrica, electrónica y sistemas computacionales. La mecánica comprende tres aspectos: diseño mecánico de la máquina, análisis estático y análisis dinámico. La microelectrónica le permite al robot transmitir la información que se le entrega, coordinando impulsos eléctricos que hacen que el robot realice los movimientos requeridos por la tarea. La informática provee de los programas necesarios para lograr la coordinación mecánica requerida en los movimientos del robot, dar un cierto grado de inteligencia a la máquina, es decir adaptabilidad, autonomía y capacidad interpretativa y correctiva. (Ana María González, s.f.)

La robótica está experimentando en la actualidad un crecimiento explosivo propulsado por los avances en computación, sensores, electrónica y software. Saturadas en cierta medida las aplicaciones del robot en talleres y líneas de fabricación, los robots están tomando posiciones en aplicaciones fuera del taller, caracterizadas fundamentalmente por la falta de estandarización

del producto, del proceso y la no estructuración del entorno. Buena parte de la clasificación de robots está destinado a la fabricación de series media y es conocido como Robot Industrial. (Barrientos A, 1999)

5.2. Robot Industrial

Según la (RIA, 2023) define a un robot industrial como un robot manipulador multifuncional reprogramable, capaz de mover materias, piezas, herramientas, o dispositivos especiales, según trayectorias variables, programadas para realizar tareas diversas.

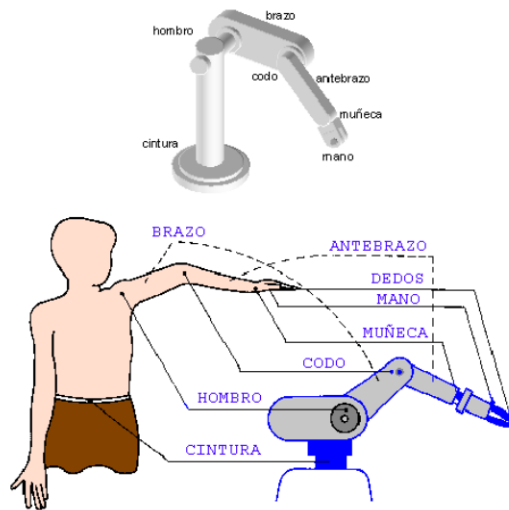
La maquinaria para la automatización rígida dio paso al robot con el desarrollo de controladores rápidos, basados en el microprocesador, así como un empleo de servos en bucle cerrado, que permiten establecer con exactitud la posición real de los elementos del robot y establecer el error con la posición deseada. (V. R. González, 2002) Se tiene la siguiente clasificación:

5.2.1. Robot manipulador

Son sistemas mecánicos multifuncionales, con un sencillo sistema de control, que permite gobernar el movimiento de sus elementos, de los siguientes modos: manual, de secuencia fija, de secuencia variable. Existen muchas operaciones básicas que pueden ser realizadas óptimamente mediante manipuladores, por lo que se debe considerar seriamente el empleo de estos dispositivos, cuando las funciones de trabajo sean sencillas y repetitivas. En la figura 2 se muestra la similitud entre un robot manipulador y el brazo humano. (V. R. González, 2002)

Figura 2

Robot Manipulador.



Nota: Se puede observar la semejanza de un robot manipulador con la de un brazo humano. (V. González, 2002)

5.2.2. Robot cartesiano

Los robots industriales cartesianos se caracterizan por posicionarse mediante 3 articulaciones lineales. Además, hacen movimientos perpendiculares a partir de los 3 ejes cartesianos X, Y y Z. (Esneca, 2020)

5.2.3. Robots industriales Scara

Este tipo de robot industrial también se mueve en iguales planos cartesianos que el anterior. Su diferencia radica en la incorporación de un eje final del plano Z. Con este es posible hacer girar la herramienta en la parte final del brazo robótico. (Esneca, 2020)

5.2.4. 6 grados de libertad

Estos tipos de robots industriales son llamados así por sus articulaciones que pueden colocar su pinza en una posición con 3 movimientos. Esto permite tener más flexibilidad para las diferentes tareas o aplicaciones industriales. Además, tienen la capacidad de transformarse en robots colaborativos. (Esneca, 2020)

5.2.5. Tipos de robots redundantes

Los redundantes o cíclicos pueden poner su herramienta en una posición determinada y bajo diferentes posturas. A diferencia de los robots de 6 ejes que sólo pueden tener una postura para una cierta posición. Los redundantes son como un brazo humano que pueden sujetar algo y al mismo tiempo mover el hombro y el codo.(Esneca, 2020)

5.2.6. Máquinas de doble brazo

Por último, uno de los tipos de robots industriales más complejos es el de doble brazo. Esto es así porque trabajan simétricamente sobre un solo objeto.(Esneca, 2020)

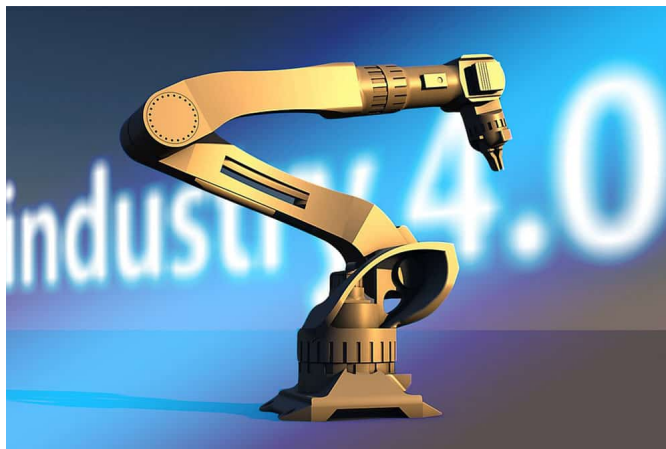
Además, tienen funciones de visión y detección de fuerza, y pueden hacer tareas autónomas. Dentro de ellas pueden tomar decisiones y hacer acciones complejas como la identificación de formas o el ajuste de cantidades y fuerzas.(Esneca, 2020)

5.3. Brazo Antropomórfico

Un robot antropomórfico también llamado manipulador de codo, es un tipo de robot que cuenta con 3 articulaciones las cuales son configurables para regular su posicionamiento, además de 3 articulaciones que se encargan de su orientación, al final de la estructura del robot cuenta generalmente con una pinza o un gripper.

Figura 3

Brazo Antropomórfico.



Nota: Se puede observar un ejemplo de un brazo antropomórfico de acuerdo a Esneca (2019)

Según (ABC-Pack, 2016), el robot antropomorfo puede programarse para manipular cualquier tipo de elemento, para ello se diseña y fabrica garras que permiten el manipulado de cajas, garrafas, botellas, sacos o cualquier elemento. Además de la programación del propio robot, en función de el tipo de elemento a manipular se integran en la celda toda una serie de maquina auxiliares, como pueden ser movimentaciones, formadoras de botellas, embudos, etc.

5.4. Brazo robótico Mitsubishi RV-2AJ

El robot Mitsubishi Melfa RV2AJ es un robot manipulador didáctico de 5 grados de libertad y un alcance máximo de 410 mm desde su base a su gripper, con la capacidad de levantar hasta 2 kg, a una velocidad total máxima de 2100 mm/s y una repetitividad de posición de ± 0.02 mm (R., 2014).

Figura 4

Brazo robótico Mitsubishi RV-2AJ.



Nota: Se puede observar un brazo robótico Mitsubishi RV-2AJ de acuerdo a Márquez (2015)

A continuación, se mostrará las principales características del brazo robótico Mitsubishi RV-2AJ:

Tabla 1*Características del brazo robótico Mitsubishi RV-2AJ*

Grados de libertad	5	
Motores	Servomotores AC (Ejes J1, J3 y J5 con freno)	
Dirección de posición	Codificadores absolutos	
Máxima Carga (kg)	2	
Longitud del brazo (mm)	250	
Alcance radial máximo (mm)	410	
Limite (grados)	J1	±150
	J2	180 (-60 a +120)
	J3	230 (-110 a +120)
	J4	-
	J5	±90
	J6	±200
Velocidad máxima (grados/s)	J1	180
	J2	90
	J3	135
	J4	-
	J5	180
	J6	210
Velocidad máxima (mm/s)	2100	
Repetitividad (mm)	±0,02	
Peso (kg)	17	
Instalación	Suelo, techo o base FESTO	
Controlador del robot	CR1-571	

Nota: Se puede observar una tabla con las principales características del brazo robótico Mitsubishi RV-2AJ de acuerdo a (Fonseca, 2011)

5.5. Cinemática de robots antropomórficos

Los robots clásicos presentan una arquitectura antropomórfica serial, semejante al brazo humano. Consisten de una serie de barras rígidas unidas entre sí a través de articulaciones de un grado de libertad del tipo rotacional o prismática. En general cada articulación logra su movimiento a través de un accionamiento de potencia e incluye otros dispositivos como reductores de velocidad, frenos y sensores de posición o velocidad.

5.6. Ubuntu

Ubuntu es una distribución GNU/Linux que ofrece un interesante sistema operativo para equipos de escritorio y servidores en el ámbito educativo. Es una distribución basada en Debian cuyas principales características son:

- Facilidad de manejo.
- Actualizaciones frecuentes.
- Facilidad de instalación del sistema.
- Búsqueda e instalación de programas robusta y fácil al basarse en paquetes.
- Libertad de uso y distribución.

5.6.1. Sistema Operativo Robótico (ROS)

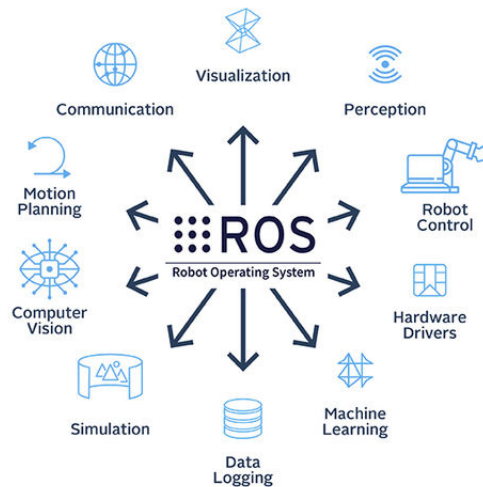
Según (Delgado, 2017) el Robot Operating System (ROS) es un middleware robótico, es decir, una colección de frameworks para el desarrollo de software de robots. El sistema ROS a pesar de no ser un sistema operativo, facilita los servicios que provee un sistema operativo cualquiera como abstracción del hardware, control de dispositivos de bajo nivel, implementación de funcionalidad de uso común, paso de mensajes entre procesos y mantenimiento de paquetes. Se debe entender además que un sistema creado con ROS consta de una serie de procesos, potencialmente en una serie de hosts diferentes, conectados en tiempo de ejecución en una topología punto a punto. Aunque los marcos están basados en un servidor central, también puede obtener los beneficios del diseño multiproceso y multihost, ya que en un servidor de datos central puede llegar a ser problemático si las computadoras están conectados en una red heterogénea. (M. Montemerlo y Thrun, 2003)

ROS permite programar cinemáticas complejas para una gran variedad de aplicaciones y sincronizar robots en buses industriales como EtherCAT o Profinet, haciéndolos sincrónicos con el resto de dispositivos de la red.

ROS es un componente fundamental de todas las soluciones robóticas y accionamientos de AutomationWare. Todos nuestros robots colaborativos se suministran con la instalación de un paquete ROS básico que incluye todos los controladores y un modelo 3D del robot, pero también herramientas de localización y uso de mapas y simuladores para la planificación de movimientos. (Industry, 2023)

Figura 5

Características del Sistema Operativo Robótico(ROS).



Nota: Características de ROS según Industry (2023)

5.6.2. Gazebo

Gazebo es una colección de bibliotecas de software de código abierto diseñadas para simplificar el desarrollo de aplicaciones de alto rendimiento. La audiencia principal de Gazebo son los desarrolladores, diseñadores y educadores de robots. Sin embargo, Gazebo se ha estructurado para adaptarse a muchos casos de uso diferentes. Cada biblioteca dentro de Gazebo tiene dependencias mínimas, lo que les permite usarse en tareas que van desde resolver transformaciones matemáticas hasta codificación de video y hasta simulación y gestión de

procesos. Simplemente se elige las bibliotecas que necesite para su aplicación sin comprometerse con un ecosistema completo.(Gazebo, 2022)

Figura 6

Entorno realizado en Gazebo.



Nota: Entorno virtual realizado en la aplicación Gazebo (2022)

El desarrollo de Gazebo sigue estos principios básicos:

- Generalidad: Gazebo no se especializa en un tipo de robot. Las características están destinadas a aplicarse a tantos casos de uso en robótica como sea posible.
- Estabilidad: Periódicamente se proporcionan versiones estables a la comunidad. Las mejoras posteriores a las versiones estables se esfuerzan por mantener la compatibilidad con versiones anteriores de API, ABI y comportamiento. Los cambios importantes apuntan a una próxima versión.
- Actualizaciones: Siempre que sea posible, se agregan nuevas funciones compatibles con versiones anteriores a las versiones estables. De esta forma, los usuarios pueden aprovechar las nuevas funciones sin necesidad de actualizar a una versión posterior y pueden posponer los esfuerzos de actualización hasta que se necesiten cambios importantes.
- Facilidad de uso: los comportamientos predeterminados deben ser aptos para principiantes. Por ejemplo, de manera predeterminada, Gazebo comienza a ejecutar una

interfaz gráfica. Los usuarios avanzados pueden ejecutar sin cabeza, pero se necesita una configuración adicional para lograr este comportamiento.

- Modularidad: los paquetes de Gazebo son lo más independientes posible para que puedan usarse en una variedad de proyectos además de Gazebo. Por ejemplo, un usuario de Gazebo Rendering no necesita instalar, compilar o incluso conocer Gazebo Physics.
- Extensibilidad: los usuarios no necesitan bifurcar las bibliotecas principales o compilar desde la fuente para agregar funcionalidad. Por ejemplo, se puede cargar un nuevo motor de renderizado como complemento.
- Flexibilidad: los usuarios tienen opciones para activar o desactivar funciones siempre que sea razonable. Por ejemplo, la física está integrada en Gazebo como un complemento, por lo que los usuarios pueden cargar una simulación sin física o con física alternativa, si lo desean.
- Mantenibilidad: la mantenibilidad a largo plazo del código se tiene en cuenta durante el desarrollo. Con ese fin, el proyecto sigue estrictos estándares de prueba, documentación, linting y estilo.
- Portabilidad: Gazebo es compatible con todos los principales sistemas operativos: Linux, macOS y Windows. Esto afecta la forma en que se eligen las dependencias y se implementan las características.

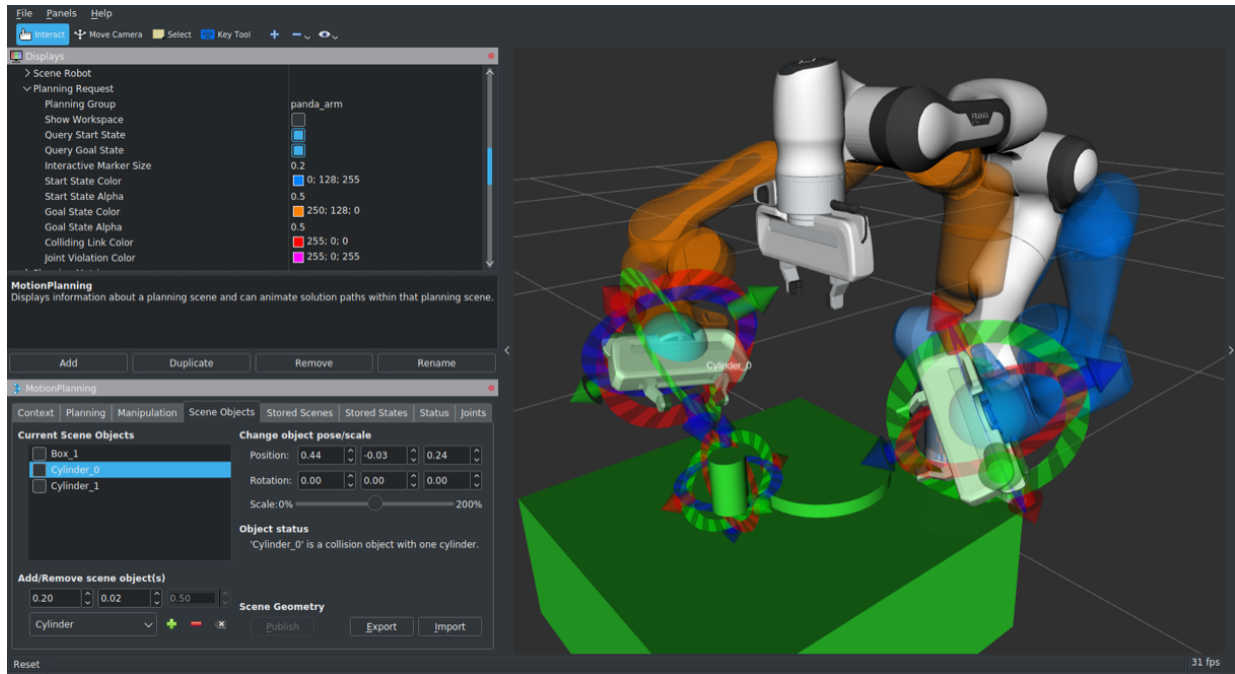
5.6.3. MoveIt

MoveIt es el software de manipulación más utilizado y se ha utilizado en más de 150 robots. Se publica bajo los términos de la licencia BSD y, por lo tanto, es gratuito para uso industrial, comercial y de investigación. (figura 7)

Al incorporar los últimos avances en planificación de movimiento, manipulación, percepción 3D, cinemática, control y navegación, MoveIt es un software de última generación para la manipulación móvil. (MoveIt, 2022)

Figura 7

Entorno virtual de movimiento con MoveIt.



Nota: Entorno virtual de movimiento utilizando el software MoveIt (2022)

Entre las principales características que este software ofrece están:

- Visualizador interactivo 3D: Las demostraciones visuales listas para usar en Rviz permiten a los usuarios experimentar con varios algoritmos de planificación alrededor de obstáculos.
- Simulación en Gazebo: Acelera los ciclos de desarrollo y prueba aprovechando un simulador completo basado en la física con MoveIt. Combina Gazebo, ROS Control y MoveIt para una poderosa plataforma de desarrollo de robótica.
- Asistente de configuración: Configura cualquier robot para que funcione con MoveIt con el asistente de configuración paso a paso, o usando configuraciones preconfiguradas populares. También incluye configuración de Gazebo y ROS Control.
- Constructor de tareas: Una forma flexible y transparente de definir y planificar acciones que consisten en múltiples subtareas interdependientes.

5.7. Acceso remoto

Según Splashtop (2022) el acceso remoto (o escritorio remoto) es la capacidad de acceder a un ordenador o dispositivo desde otro dispositivo, en cualquier momento y desde cualquier lugar. Con un software de acceso remoto instalado en su ordenador, tiene la libertad de acceder a ese ordenador, y a todos sus archivos y aplicaciones, desde otro dispositivo y controlarlo como si estuviera sentado frente a él.

Figura 8

Acceso remoto.



Nota: Ilustración de acceso remoto. Splashtop (2022)

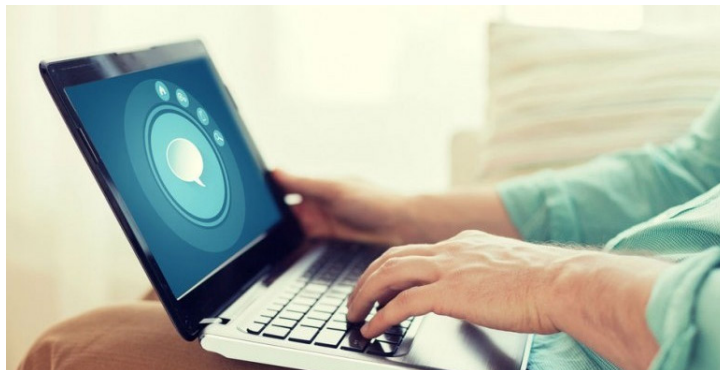
5.8. Entorno virtual

Un entorno virtual se refiere a cualquier software, programa o sistema que implementa, administra y controla múltiples instancias virtuales.

El software se instala dentro de la infraestructura de TI (tecnología de la información) existente en una organización y se controla desde dentro de la propia organización. ARATECNIA (2022)

Figura 9

Entorno virtual.



Nota: Representación de un entorno virtual. ARATECNIA (2022)

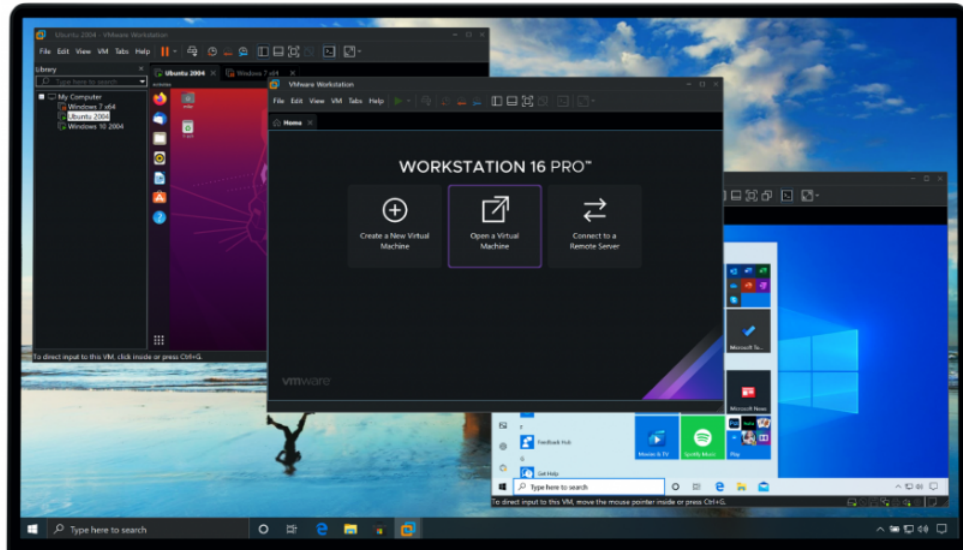
5.8.1. VMware

VMware Workstation es una línea de productos de Hipervisor de escritorio que permiten a los usuarios ejecutar máquinas virtuales, contenedores y clústeres de Kubernetes. VMware (2022)

Por medio de esta aplicación es posible instalar sistemas operativos adicionales, conocidos como «sistemas invitados», dentro de otro sistema operativo «anfitrión», cada uno con su propio ambiente virtual. Entre los sistemas operativos soportados (en modo anfitrión) se encuentran GNU/Linux, Mac OS X, OS/2 Warp, Genode, Windows y Solaris/OpenSolaris. VMware (2022) Se muestra a continuación en la (figura 10) el entorno de VMware.

Figura 10

VMware.



Nota: Entorno característico de VMware. VMware (2022)

6. Marco metodológico

6.1. Fundamento matemático acerca de la cinemática del robot antropomórfico Mitsubishi RV-2AJ

6.1.1. Análisis de grados de libertad

El análisis de grados de libertad permite determinar la posición del efector final de un robot, la mayoría de ocasiones el número de elementos del robot coincide con el número de grados de libertad que posee el mismo; para determinarlos se sustenta en el método de Kutzbach-Grübler, el cual manifiesta que se deben analizar el número de eslabones, el número de juntas de 1 grado de libertad y el número de juntas de 2 grados de libertad y aplicar la siguiente ecuación:

$$m = 3(n - 1) - 2j_1 - j_2 \quad (1)$$

Donde:

m = Grados de libertad

n = Numero de eslabones

j1 = Número de juntas con un grado de libertad

j2 = Número de juntas con dos grados de libertad

Una vez especificado este método matemático al robot Mitsubishi RV-2AJ, se definen los datos a emplearse en el cálculo.

Tabla 2

Parámetros del robot Mitsubishi para determinar los grados de libertad.

Parámetro	Nomenclatura	Valor
Número de eslabones	n	5
Juntas con 1 grado de libertad	j1	3
Juntas con 2 grados de libertad	j2	1

Nota: La tabla 2 presenta el número de eslabones y de juntas que permitirán calcular la cantidad de grados de libertad del robot.(Chocho y Cuenca, 2022)

Se aplica la fórmula presentada en la ecuación 1, obteniendo:

$$m = 3(5 - 1) - 2(3) - 1$$

$$m = 5 \quad \text{GDL}$$

Una vez resuelta la ecuación 1 con los datos del robot, se tiene como resultado 5 grados de libertad.

6.1.2. Cinemática directa del robot Mitsubishi RV-2AJ

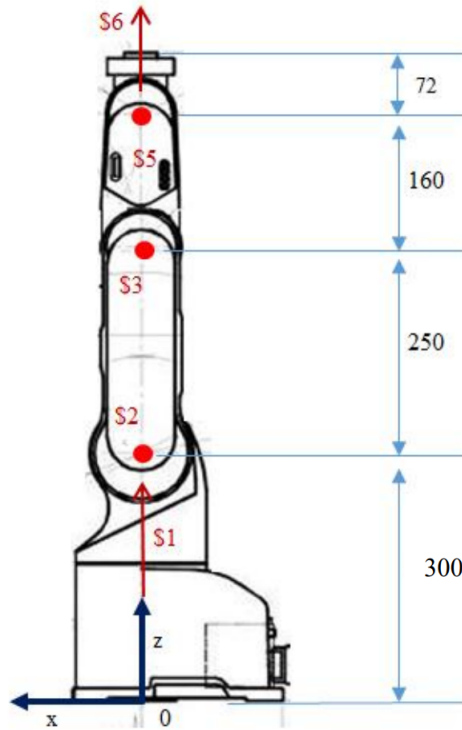
Gracias al análisis de la cinemática directa del robot Mitsubishi RV-2AJ y a los datos proporcionados como ángulos de movimiento de las articulaciones y longitud de eslabones del robot, se logra conocer la orientación del efector final.

Para la resolución del modelo cinemático directo se aplica el método Denavit-Hartenberg, teniendo en cuenta las longitudes exactas de los eslabones que conforman el robot antropomórfico mismas que determinan los datos necesarios para el planteamiento del modelo matemático.

En la figura 11 se puede observar un boceto del robot mitsubishi RV-2AJ con sus dimensiones.

Figura 11

Dimensiones de eslabones del robot Mitsubishi RV-2AJ.



Nota: Se visualiza las medidas en mm de los eslabones que conforman el robot Mitsubishi RV-2AJ. Villalobos (2015)

Tabla 3

Parámetros del robot Mitsubishi para determinar la cinemática directa mediante el método Denavit-Hartenberg.

	θ	d	a	α
1	θ_1	300	0	90°
2	$\theta_2 + 90^\circ$	0	250	0
3	θ_3	0	160	0
4	$\theta_4 + 90^\circ$	0	0	-90°
5	θ_5	72	0	0

Nota: Tabla Denavit Hartenberg para el robot Mitsubishi RV-2AJ.

En la tabla 3 se muestran los parámetros usados para la obtención de las ecuaciones de

cinemática directa del robot antropomórfico.

Chocho y Cuenca (2022) plantea las matrices de transformación. Dichas matrices de transformación se plantean a partir de la matriz modelo:

$$A_{n-1}^n = \begin{bmatrix} \cos(\theta_n) & -\sin(\theta_n) \cdot \cos(\alpha_n) & \sin(\theta_n) \cdot \sin(\alpha_n) & a_n \cdot \cos(\theta_n) \\ \sin(\theta_n) & \cos(\theta_n) \cdot \cos(\alpha_n) & -\cos(\theta_n) \cdot \sin(\alpha_n) & a_n \cdot \sin(\theta_n) \\ 0 & \sin(\alpha_n) & \cos(\alpha_n) & d_n \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

Una vez planteado el modelo, se procede a calcular la matriz de transformación para cada junta del robot Mitsubishi RV-2AJ.

Junta 1:

$$A_0^1 = \begin{bmatrix} \cos(\theta_1) & 0 & \sin(\theta_1) & 0 \\ \sin(\theta_1) & 0 & -\cos(\theta_1) & 0 \\ 0 & 1 & 0 & 300 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Junta 2:

$$A_1^2 = \begin{bmatrix} \cos(\theta_2) & -\sin(\theta_2) & 0 & 250 \cdot \cos(\theta_2) \\ \sin(\theta_2) & \cos(\theta_2) & 0 & 250 \cdot \sin(\theta_2) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Junta 3:

$$A_2^3 = \begin{bmatrix} \cos(\theta_3) & -\sin(\theta_3) & 0 & 160 \cdot \cos(\theta_3) \\ \sin(\theta_3) & \cos(\theta_3) & 0 & 160 \cdot \sin(\theta_3) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Junta 4:

$$A_3^4 = \begin{bmatrix} \cos(\theta_4) & 0 & -\sin(\theta_4) & 0 \\ \sin(\theta_4) & 0 & \cos(\theta_4) & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Junta 5:

$$A_4^5 = \begin{bmatrix} \cos(\theta_5) & -\sin(\theta_5) & 0 & 0 \\ \sin(\theta_5) & \cos(\theta_5) & 0 & 0 \\ 0 & 0 & 1 & 72 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

La matriz de transformación total se obtiene de la multiplicación de las matrices de las juntas.

$$A_0^5 = A_0^1 \times A_1^2 \times A_2^3 \times A_3^4 \times A_4^5 \quad (3)$$

Al resolver la ecuación 3, se obtiene la matriz total:

$$A_0^5 = \begin{bmatrix} a_1 & b_1 & c_1 & d_1 \\ a_2 & b_2 & c_2 & d_2 \\ a_3 & b_3 & c_3 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4)$$

Donde:

$$a_1 = \cos(\theta_4) + \cos(\theta_5) - \cos(\theta_1) \cdot \sin(\theta_2) \cdot \sin(\theta_3) + \cos(\theta_1) \cdot \cos(\theta_2) \cdot \cos(\theta_3)$$

$$b_1 = -\sin(\theta_5) - \cos(\theta_1) \cdot \cos(\theta_2) \cdot \sin(\theta_3) - \cos(\theta_1) \cdot \cos(\theta_3) \cdot \sin(\theta_2)$$

$$c_1 = -\sin(\theta_1) - \sin(\theta_4)$$

$$d_1 = 250 \cos(\theta_1) \cdot \cos(\theta_2) - 160 \cos(\theta_1) \cdot \sin(\theta_2) \cdot \sin(\theta_3) + 160 \cos(\theta_1) \cdot \cos(\theta_2) \cdot \cos(\theta_3)$$

$$a_2 = \sin(\theta_4) + \sin(\theta_5) + \cos(\theta_2) \cdot \cos(\theta_3) \cdot \sin(\theta_1) - \sin(\theta_1) \cdot \sin(\theta_2) \cdot \sin(\theta_3)$$

$$b_2 = \cos(\theta_5) - \cos(\theta_2) \cdot \sin(\theta_1) \cdot \sin(\theta_3) - \cos(\theta_3) \cdot \sin(\theta_1) \cdot \sin(\theta_2)$$

$$c_2 = \cos(\theta_4) - \cos(\theta_1)$$

$$d_2 = 250 \cos(\theta_2) \cdot \sin(\theta_1) + 160 \cos(\theta_2) \cdot \cos(\theta_3) \cdot \sin(\theta_1) - 160 \sin(\theta_1) \cdot \sin(\theta_2) \cdot \sin(\theta_3)$$

$$a_3 = \cos(\theta_2) \cdot \sin(\theta_3) + \cos(\theta_3) \cdot \sin(\theta_2)$$

$$b_3 = \cos(\theta_2) \cdot \cos(\theta_3) - \sin(\theta_2) \cdot \sin(\theta_3) - 1$$

$$c_3 = 1$$

$$d_3 = 250 \sin(\theta_2) + 160 \cos(\theta_2) \cdot \sin(\theta_3) + 160 \cos(\theta_3) \cdot \sin(\theta_2) + 372$$

Al aplicar el producto cruz de la matriz de la ecuación 4 y la matriz de posición P0 se obtiene como resultado las ecuaciones que describen la cinemática del robot Mitsubishi RV-2A.

$$P = A_0^5 \times P_0 \quad (5)$$

Donde P_0 :

$$P_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

P de la ecuación 5 se muestra a continuación:

$$P = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} a_1 & b_1 & c_1 & d_1 \\ a_2 & b_2 & c_2 & d_2 \\ a_3 & b_3 & c_3 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \quad (6)$$

Al resolver las matrices de la ecuación 6, da como resultado las ecuaciones de x, y, y z.
Así:

$$x = 250 \cos(\theta_1) \cdot \cos(\theta_2) - 160 \cos(\theta_1) \cdot \sin(\theta_2) \cdot \sin(\theta_3) + 160 \cos(\theta_1) \cdot \cos(\theta_2) \cdot \cos(\theta_3) \quad (7)$$

$$y = 250 \cos(\theta_2) \cdot \sin(\theta_1) + 160 \cos(\theta_2) \cdot \cos(\theta_3) \cdot \sin(\theta_1) - 160 \sin(\theta_1) \cdot \sin(\theta_2) \cdot \sin(\theta_3) \quad (8)$$

$$z = 250 \sin(\theta_2) + 160 \sin(\theta_3) \cos(\theta_2) + 160 \cos(\theta_3) \cdot \sin(\theta_2) + 372 \quad (9)$$

6.1.3. Cinemática inversa del robot Mitsubishi RV-2AJ

La cinemática inversa expresa los ángulos de movimiento de las juntas, teniendo como base valores de orientación y posición del efector final. Se tiene varios métodos como análisis geométrico y matemático para determinar la cinemática inversa del robot y de esta forma obtener el valor de los ángulos solicitados.

Tomando en cuenta el análisis de los grados de libertad antes realizado por (Chocho y Cuenca, 2022), se define 3 ángulos de movimiento de juntas: θ_1 , θ_2 y θ_3 ; para la muñeca se tiene los grados de libertad: θ_4 y θ_5 .

Para el análisis se toma en cuenta las 3 primeras juntas:

$$A_0^3 = A_0^1 \times A_1^2 \times A_2^3 = T \quad (10)$$

Obtenemos la siguiente ecuación:

$$(A_0^1)^{-1} \cdot T = A_1^2 \times A_2^3$$

Teniendo los valores reemplazados, se obtiene la matriz total con la cual se resolverá los valores de los diferentes ángulos.

$$A_1^3 = \begin{bmatrix} \cos(\theta_1) & \sin(\theta_1) & 0 & 0 \\ 0 & 0 & 1 & -300 \\ \sin(\theta_1) & -\cos(\theta_1) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} nx & ox & ax & X \\ ny & oy & ay & Y \\ nz & oz & az & Z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (11)$$

$$= \begin{bmatrix} \cos(\theta_2 + \theta_3) & -\sin(\theta_2 + \theta_3) & 0 & 250 \cos(\theta_2) + 160 \cos(\theta_2 + \theta_3) \\ \sin(\theta_2 + \theta_3) & \cos(\theta_2 + \theta_3) & 0 & 250 \sin(\theta_2) + 160 \sin(\theta_2 + \theta_3) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Para el planteamiento de las ecuaciones que supondrán el valor de los ángulos, se opera la fila 3 de la primera matriz de la ecuación 11 por la columna 4 de la segunda matriz:

$$X \sin(\theta_1) - Y \cos(\theta_1) = 0$$

$$\frac{\sin(\theta_1)}{\cos(\theta_1)} = \frac{Y}{X}$$

$$\tan(\theta_1) = \frac{Y}{X} \quad (12)$$

Se plantea por lo tanto un sistema de ecuaciones en donde se opera la segunda fila y cuarta columna de la ecuación 11, de igual manera la primera fila y la cuarta columna de la misma matriz, dándonos como resultado un sistema de dos ecuaciones con dos incógnitas.

$$Z - 300 = 250 \sin \theta_2 + 160 \sin(\theta_2 + \theta_3) \quad (13)$$

$$X \cos(\theta_1) + Y \sin(\theta_1) = 250 \cos(\theta_2) + 160 \cos(\theta_2 + \theta_3) \quad (14)$$

El robot Mitsubishi RV-2AJ posee dos movimientos que describen la rotación: pitch como θ_4 y roll como θ_5 presentes en la muñeca. Se determina la rotación en z presentando la siguiente matriz:

$$R = T(1 : 3, 1 : 3) \times Rot(z, 90^\circ)$$

$$R = A_0^3 \cdot \begin{bmatrix} \cos 90^\circ & -\sin 90^\circ & 0 \\ \sin 90^\circ & \cos 90^\circ & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

En la ecuación 4 podemos determinar las ecuaciones de los ángulos pitch y roll gracias a la matriz de rotación R: Para θ_4 :

$$\sin \theta_4 = -R_{3,1}$$

$$\sin \theta_4 = -\cos \theta_2 \cos \theta_3 + \sin \theta_2 \sin \theta_3 + 1$$

$$\theta_4 = 90^\circ + \arcsin(1 - \cos(\theta_2 + \theta_3)) \quad (15)$$

Para θ_5 :

$$\tan \theta_5 = \frac{R_{3,2}}{\cos \theta_4}$$

$$\theta_5 = \arctan\left(\frac{-\sin(\theta_2 + \theta_3)}{\cos \theta_4}\right) \quad (16)$$

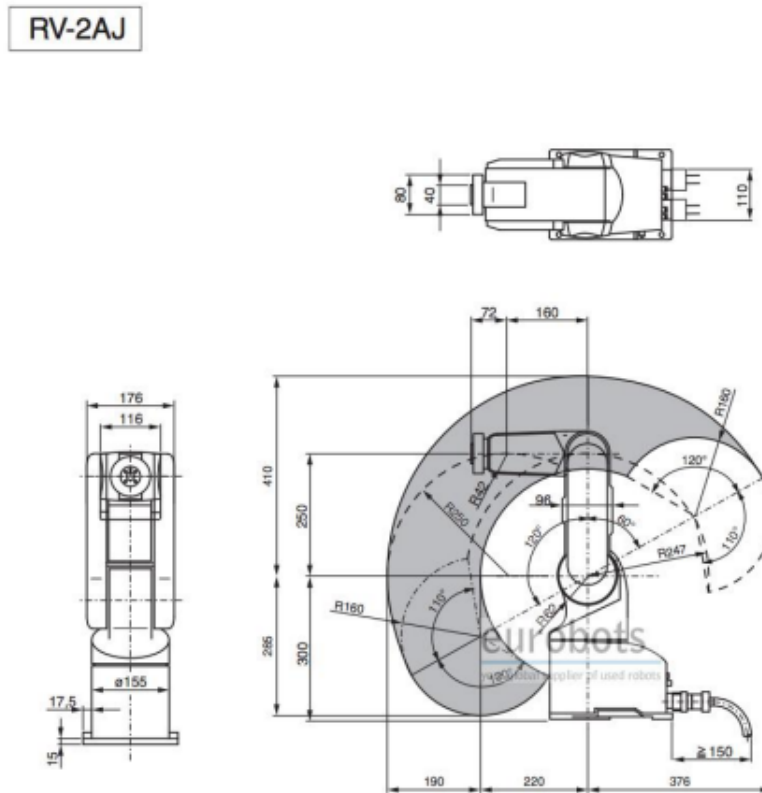
Gracias a este análisis se puede determinar los valores de los ángulos de movimiento de las juntas del robot para que se logre ubicar en la posición que el usuario disponga.

6.1.4. Área de trabajo del robot Mitsubishi RV-2AJ

El espacio que dispone el robot para realizar los trabajos dispuestos por el usuario está dado por los grados de libertad, movimientos y limitaciones articulares mostrados en la (figura 12).

Figura 12

Espacio de trabajo del robot Mitsubishi RV-2AJ.



Nota: Se muestra el espacio total de trabajo del robot antropomórfico Mitsubishi RV-2AJ.(Villalobos, 2015)

Un factor determinante en el espacio de trabajo total que tiene el robot son las dimensiones del mismo, ya que establecen un radio de operación definido en 410 mm, su altura máxima y su rango de alcance definido en 120° , estos se obtienen gracias a las longitudes de los eslabones que conforman el robot antropomórfico.

6.2. Modelado del robot Mitsubishi RV-2AJ en SolidWorks

6.2.1. Modelo del robot antropomórfico

Para modelar un brazo robótico en SolidWorks, primero debe crear los componentes individuales del brazo robótico. Estos componentes pueden incluir un eje, un motor, un

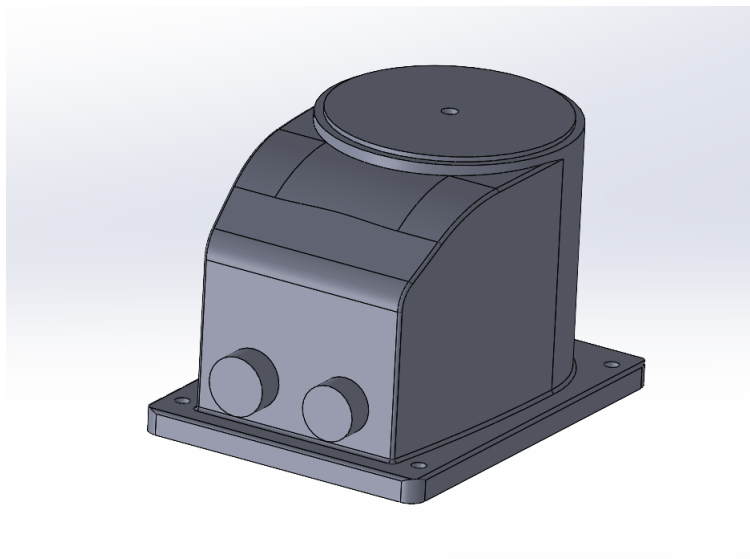
servomotor, una placa de base, una placa superior, una articulación, varillas, tornillos, pernos y una caja de control. Una vez que haya creado todos los componentes, debe ensamblarlos en SolidWorks para crear el brazo robótico. Esto significa colocar los componentes según las instrucciones de montaje para asegurar que todos encajen correctamente. Una vez que haya ensamblado el brazo, puede agregar los cables y tuberías necesarios para conectar los componentes.

Los componentes individuales según (Abedloo, 2014) se pueden observar desde la figura 13 hasta la figura 19:

1. Base del robot:

Figura 13

Base del robot Mitsubishi RV-2AJ.

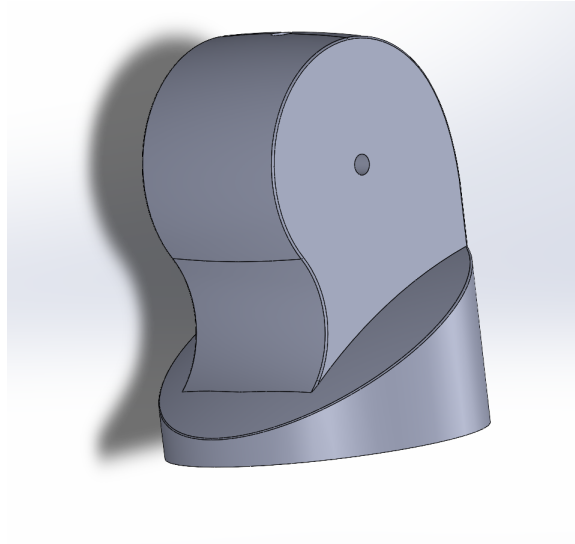


Nota: En la imagen se observa la base del robot antropomórfico la cual se encuentra fija y es la parte principal para el ensamble.(Abedloo, 2014)

2. Eslabón 1:

Figura 14

Primer eslabón del robot Mitsubishi RV-2AJ.

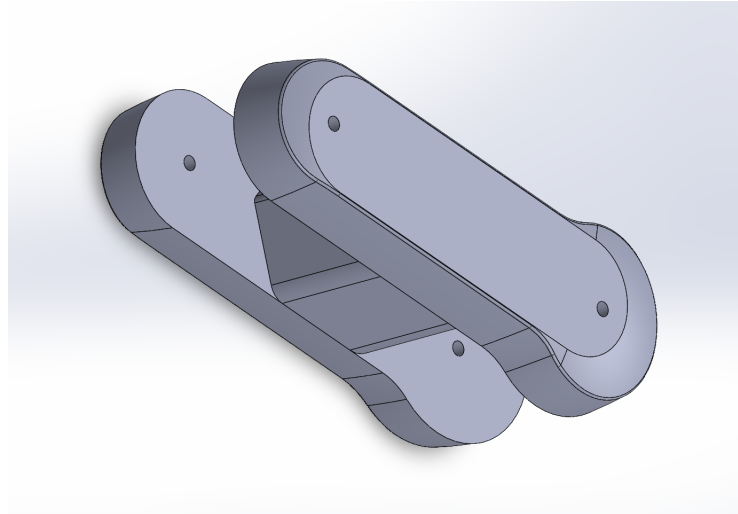


Nota: Se presenta el primer eslabón siguiente a la base del robot antropomórfico el cual se presenta un grado de libertad rotacional. (Abedloo, 2014)

3. Eslabón 2:

Figura 15

Segundo eslabón del robot Mitsubishi RV-2AJ.

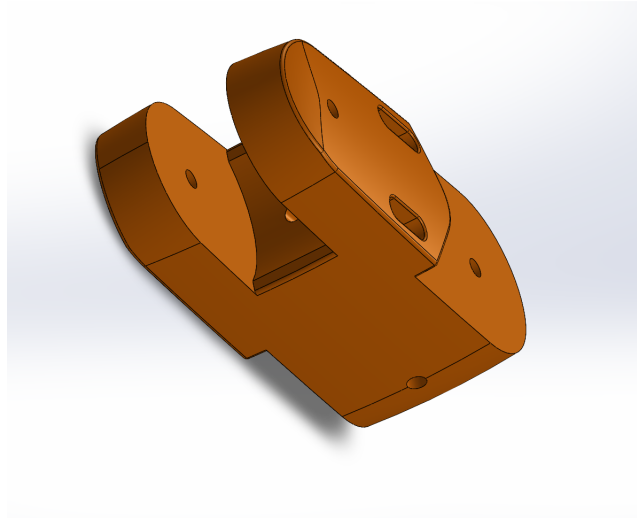


Nota: Como se puede ver en la imagen 15 el segundo eslabón que va conectado al eslabón 1 mencionado anteriormente, este eslabón cuenta con un grado de libertad rotacional. (Abedloo, 2014)

4. Eslabón 3:

Figura 16

Tercer eslabón del robot Mitsubishi RV-2AJ.

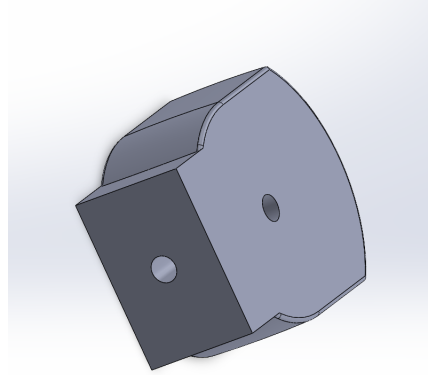


Nota: Se visualiza el tercer eslabón que se conecta al eslabón 2 antes mencionado, este eslabón cuenta con un grado de libertad rotacional. (Abedloo, 2014)

5. Eslabón 4:

Figura 17

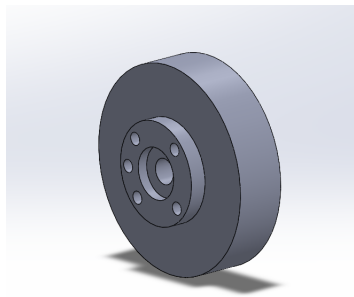
Cuarto eslabón del robot Mitsubishi RV-2AJ.



Nota: Se presenta el cuarto eslabón que va conecta al eslabón 3 antes mencionado, este eslabón cuenta con un grado de libertad rotacional. (Abedloo, 2014)

Figura 18

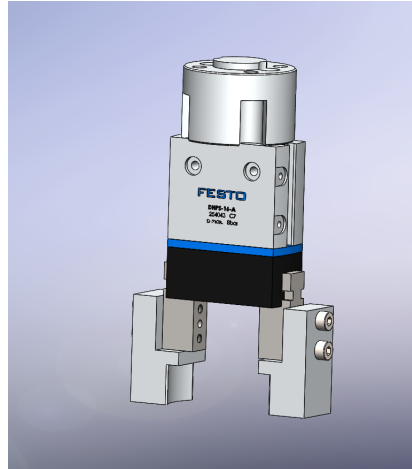
Quinto eslabón del robot Mitsubishi RV-2AJ.



Nota: Como se puede apreciar en la figura 18 el quinto eslabón que se conecta al eslabón 4 antes mencionado, este eslabón cuenta con un grado de libertad rotacional de 360°. (Abedloo, 2014)

Figura 19

Gripper FESTO especial para el robot Mitsubishi RV-2AJ.



Nota: Se presenta en la (figura 19) el gripper FESTO el cual va conectado al eslabón 5 mencionado anteriormente, este gripper cuenta con dos pinzas las cuales tienen un grado de libertad prismático a través de la estructura del mismo.(FESTO, 2022)

6.2.2. Ensamblaje de modelo robot Mitsubishi RV-2AJ en SolidWorks

El proceso de ensamblaje de un modelo en SolidWorks es el siguiente:

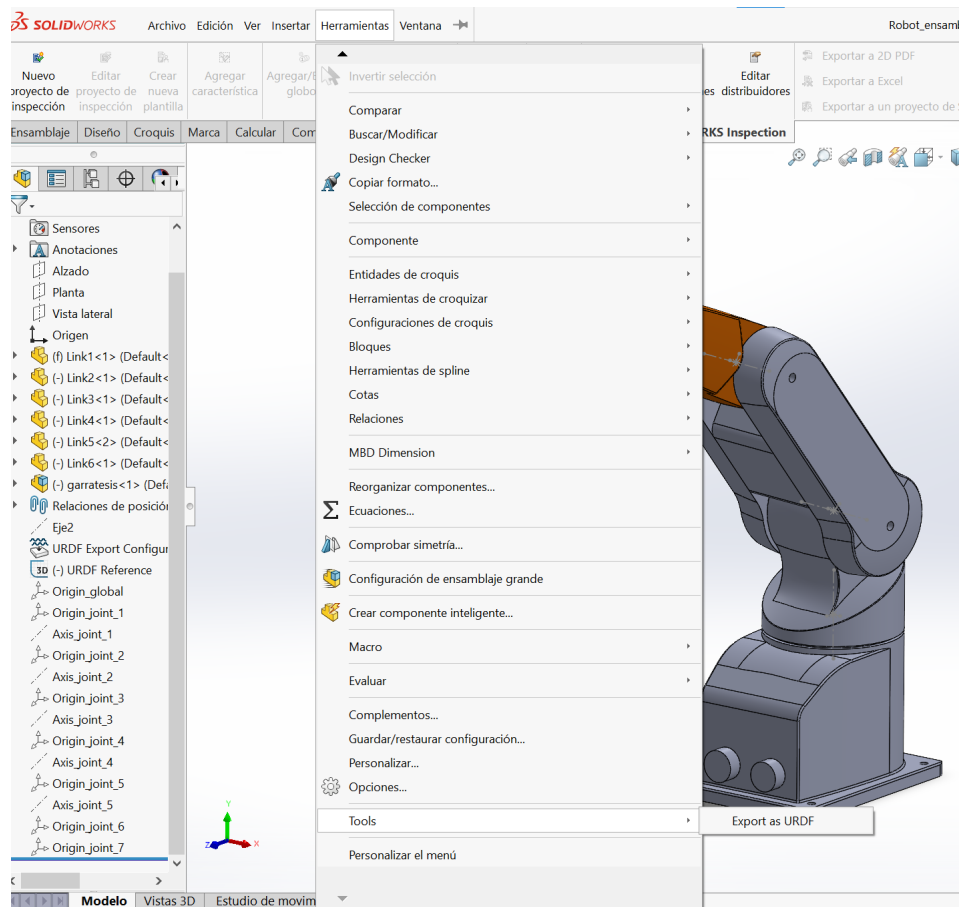
1. Abrir el archivo de SolidWorks.
2. Hacer clic en el botón “Ensamble” en la barra de herramientas.
3. Seleccionar los componentes del modelo que se desean ensamblar.
4. Conectar los componentes usando el comando “Insertar”.
5. Utilizar el comando "Movimiento" para mover los componentes en la posición deseada.
6. Añadir todos los elementos necesarios para completar el ensamblaje, tales como pernos, tuercas, etc.
7. Usar el comando “Fuera de centro” para ajustar la posición y alineación de los componentes.
8. Usar el comando “Sellado” para asegurar que los componentes estén bien ajustados.
9. Usar el comando “Simulación” para verificar que el ensamblaje se ha realizado correctamente.
10. Guardar el ensamblaje.

6.2.3. Conversión del ensamblaje del robot Mitsubishi RV-2AJ desde SolidWorks a formato URDF

Para poder configurar el modelo realizado en SolidWorks se tiene que convertir dicho modelo a formato URDF, este formato es aceptado por el software de configuración MoveIt en ROS. Por lo tanto, se debe instalar la herramienta de conversión desde el navegador y se procede a seleccionarla para empezar la conversión del modelo (figura 20).

Figura 20

Herramienta para exportar el modelo a formato URDF.



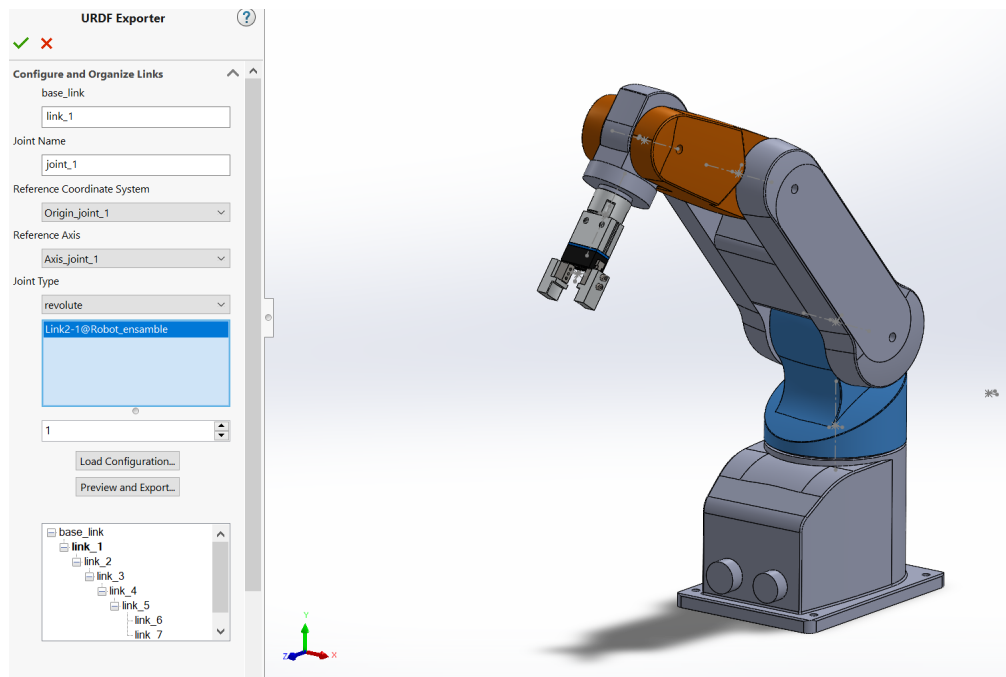
Nota: Selección de la herramienta export to URDF, misma que nos permite iniciar la configuración previa del modelo antes de su exportación. (Abedloo, 2014)

Una vez seleccionada la herramienta, se procede a seleccionar cada uno de los eslabones que

conforman el modelo, teniendo en cuenta que cada eslabón debe tener la relación padre e hijo, siendo el padre el eslabón seleccionado en el instante y el hijo el eslabón que va consiguiente. Luego de crear todas las relaciones especificando el tipo de grado de libertad que se tendrá, la herramienta genera los distintos orígenes y ejes necesarios para que el modelo se simule con los movimientos correctamente especificados (figura 21).

Figura 21

Selección de eslabones con relación padre-hijo.

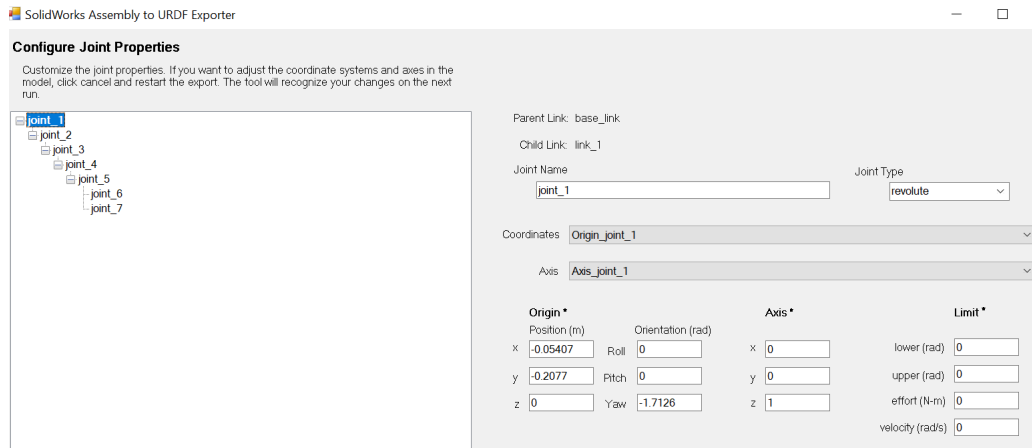


Nota: Se visualiza las relaciones creadas para el modelo en donde se especifica cada una de las juntas que tendrá el ensamble, además de los diferentes orígenes y ejes de dicho ensamble. (Abedloo, 2014)

Luego, se procede a exportar el modelo, para lo cual se observará la ventana de configuración en la que se debe tener en cuenta que el tipo de junta este correctamente seleccionado (figura 22).

Figura 22

Configuración de propiedades de las juntas del ensamblaje.



Nota: Se presenta la ventana de configuración en la que se debe configurar cada una de las juntas y verificar el tipo correcto para que el movimiento sea el adecuado. (Abedloo, 2014)

Como paso final se procede a exportar el modelo en formato URDF para su posterior configuración en el entorno de ROS.

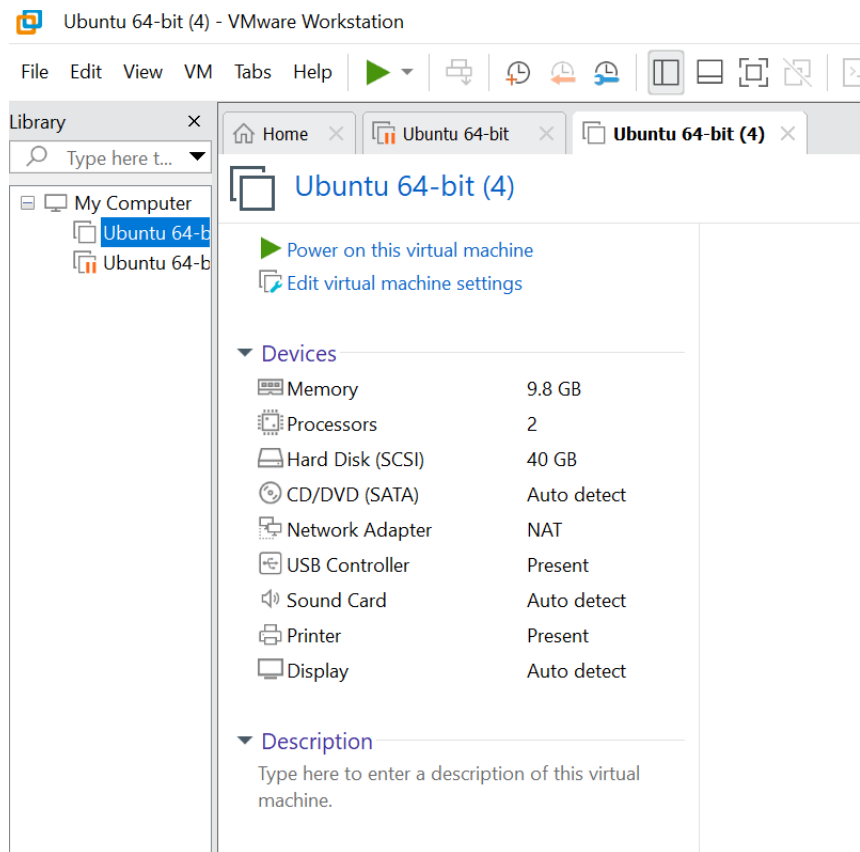
6.3. Interfaz gráfica para visualización de trayectorias del robot

6.3.1. Creación de una máquina virtual

Para la creación del entorno virtual en los distintos softwares que serán utilizados, primero se tiene la creación de una máquina virtual en la cual se instalará el sistema operativo Linux, dicho sistema es esencial para trabajar con ROS, únicamente se puede manejar por medio del sistema operativo antes mencionado. (figura 23)

Figura 23

Entorno en VMware para la creación de una máquina virtual.



Nota: Se muestra la ventana principal de VMware en donde permite la creación de máquinas virtuales y propiamente la máquina virtual a utilizarse para el proyecto

6.3.2. Robot Operating System(ROS)

Robot Operating System (ROS) es un marco de software para el desarrollo de simulación de robots. Está diseñado para facilitar el desarrollo de robots con una interfaz de programación orientada a objetos y un amplio conjunto de herramientas para el diseño, la simulación y el análisis de robots. Está basado en protocolos de comunicación abiertos, por lo que se puede usar para controlar una variedad de robots, desde pequeños robots móviles hasta robots de servicio complejos. ROS también proporciona una gran cantidad de paquetes de software para el procesamiento de imágenes, el procesamiento de lenguaje natural, el aprendizaje automático y otros.

Para instalar ROS, necesitará una versión reciente de Ubuntu (18.04 o superior). Una vez que se tenga una instalación de Ubuntu, se procede a seguir estos pasos para instalar ROS:

1. Se abre una terminal y se actualiza la lista de paquetes:

- sudo apt-get update

2. Se instala los paquetes de ROS:

- sudo apt-get install ros-noetic-desktop-full

3. Se inicializa los entornos de ROS para que se puedan utilizar en la terminal:

- source /opt/ros/noetic/setup.bash

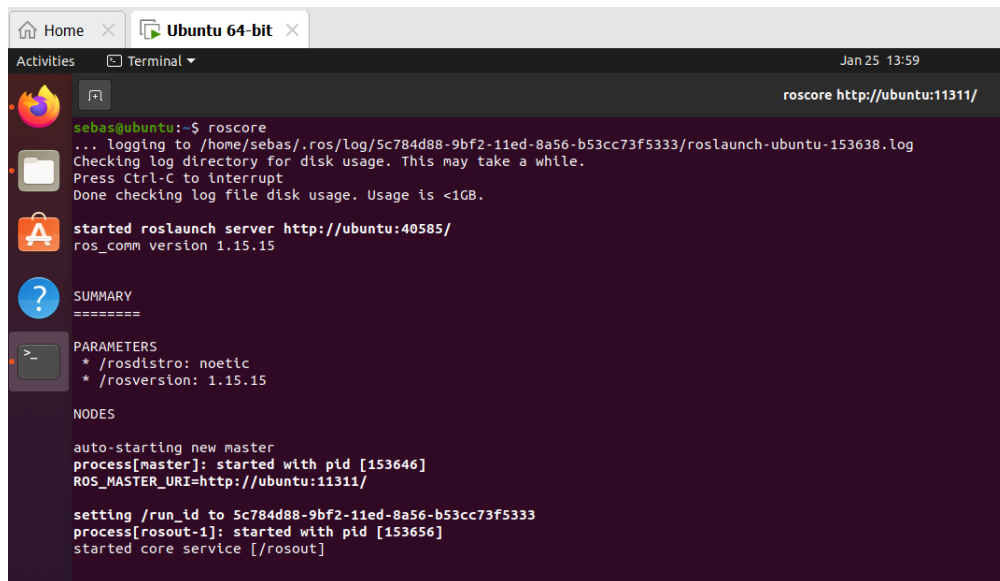
4. Instalar paquetes adicionales útiles:

- sudo apt-get install python-rosinstall python-rosinstall-generator python-wstool build-essential

Con estos pasos, se debe tener ROS instalado y listo para usar, se puede verificar la correcta instalación utilizando el comando “roscore” en la terminal y observando lo que se muestra en la (figura 24):

Figura 24

Verificación de funcionamiento de sistema operativo ROS.



```
sebas@ubuntu:~$ roscore
... logging to /home/sebas/.ros/log/5c784d88-9bf2-11ed-8a56-b53cc73f5333/roslaunch-ubuntu-153638.log
Checking log directory for disk usage. This may take a while.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://ubuntu:40585/
ros_comm version 1.15.15

SUMMARY
=====
PARAMETERS
* /roscore: noetic
* /rosversion: 1.15.15

NODES

auto-starting new master
process[master]: started with pid [153646]
ROS_MASTER_URI=http://ubuntu:11311/

setting /run_id to 5c784d88-9bf2-11ed-8a56-b53cc73f5333
process[rosout-1]: started with pid [153656]
started core service [/rosout]
```

Nota: En la imagen se observa la terminal del sistema Linux en la cual se ejecuta el comando “roscore” permitiendo verificar que el sistema ROS se encuentra correctamente instalado.

6.3.3. Interfaz MoveIt

MoveIt es un paquete ROS de código abierto para la planificación cinemática y de movimiento de robots. Proporciona una plataforma fácil de usar para desarrollar aplicaciones robóticas avanzadas. MoveIt también proporciona una interfaz fácil de usar para interactuar con los sensores, actuadores y controladores de un robot. MoveIt se puede utilizar para programar y controlar una amplia variedad de robots, incluidos robots móviles, brazos y otras plataformas robóticas especializadas. MoveIt se ha utilizado para desarrollar aplicaciones robóticas en una amplia variedad de industrias, incluidas la fabricación, la atención médica y el entretenimiento.

Para la instalación de MoveIt se debe tener en consideración que el sistema operativo ROS se encuentre correctamente instalado, una vez verificado esto se procede a digitar el siguiente comando en una nueva terminal:

- `sudo apt install ros-noetic-moveit`

Para verificar el funcionamiento y cargar el entorno ROS, primero se debe ejecutar el comando:

- `source devel/setup.bash`

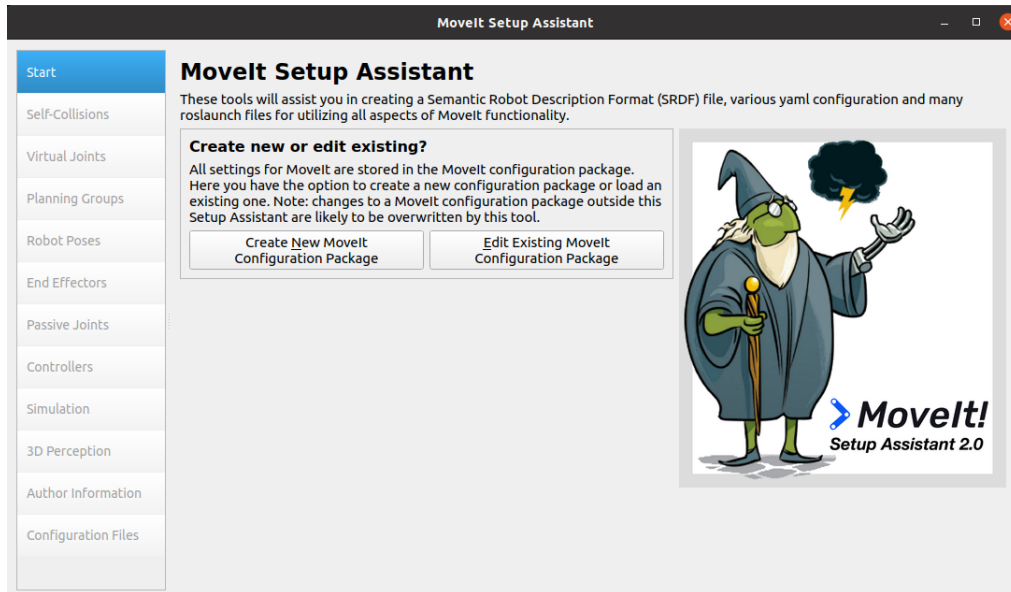
Posteriormente para iniciar el asistente de configuración de MoveIt se debe ejecutar el siguiente comando:

- `roslaunch moveit_setup_assistant setup_assistant.launch`

Una vez realizada la instalación y ejecución de los comandos se procede a visualizar la ventana de configuración de MoveIt como se muestra en la (figura 25).

Figura 25

Interfaz de configuración de MoveIt.



Nota: Se visualiza la ventana de inicio de MoveIt en la cual actúa el asistente de configuración para los proyectos.

6.3.4. Workspace para configuración del modelo de ensamble

Para la correcta aplicación y configuración del modelo presentado previamente en el programa Solidworks como punto de partida se debe crear el espacio de trabajo para almacenar todos los archivos relacionados al proyecto.

Se abre una nueva terminal y se ejecuta el comando:

- mkdir moveit_ws

Luego de esto, se procede a crear una subcarpeta dentro de la misma, esta carpeta llamada “src” contendrá la carpeta generada anteriormente del modelo transformado a formato urdf.

- mkdir moveit_ws/src

Para la posterior creación de paquetes se utiliza la herramienta Catkin, la cual proporciona un entorno de compilación común para los paquetes de software para robots, así como un marco de trabajo para la composición de aplicaciones de robot. Por lo tanto, se digita:

- cd ..

- catkin_make

Esto crea dentro de la carpeta `moveit_ws` todas las herramientas de ejecución necesarias para la posterior configuración del modelo.

Se procede a continuación a copiar la carpeta “`robot_arm_urdf`” dentro de la carpeta “`src`”.

6.3.5. Modelo representado en el software Gazebo

Gazebo ROS es una interfaz desarrollada para permitir a los usuarios la integración de robots virtuales y entornos tridimensionales dentro de la plataforma Robot Operating System. Esta interfaz es usada para simular robots, entornos y escenas físicas en un ambiente virtual que es controlado por ROS. La interfaz se ha desarrollado para proporcionar un ambiente de prueba para la creación de robots antes de pasar a la producción real. Esta herramienta es útil para los desarrolladores de robots que quieran probar y mejorar el rendimiento de su robot sin el riesgo de dañar el robot real. Además, esta herramienta se usa para desarrollar y diseñar nuevos robots en un entorno seguro y controlado. Para la inicialización de carga del modelo dentro del entorno Gazebo se procede a realizar los siguientes pasos:

1. Abrir una terminal nueva y proceder a escribir los comandos que nos direccionarán a la carpeta en la cual se tiene cargado el modelo:

- `cd /moveit_ws`

2. Cargar el entorno ROS:

- `source devel/setup.bash`

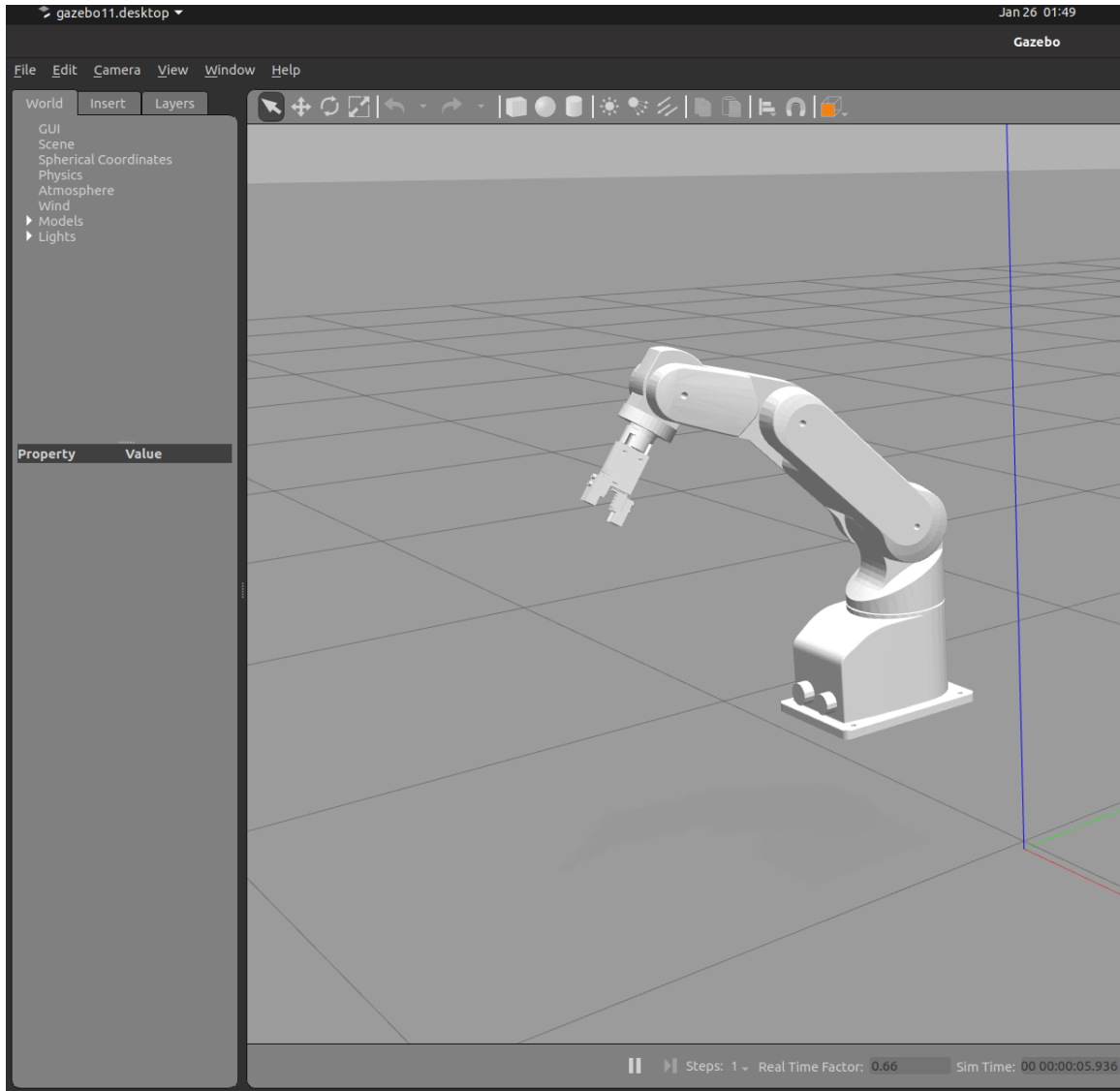
3. Se inicializa el entorno de Gazebo cargando el modelo realizado:

- `roslaunch robot_arm_urdf arm_urdf.launch`

De esta forma se presenta el modelo correctamente cargado en el entorno gráfico de visualización Gazebo (figura 26).

Figura 26

Entorno de visualización Gazebo.



Nota: El entorno de visualización permite la verificación del modelo cargado en el sistema operativo ROS.

6.3.6. Creación de posiciones en el software Gazebo

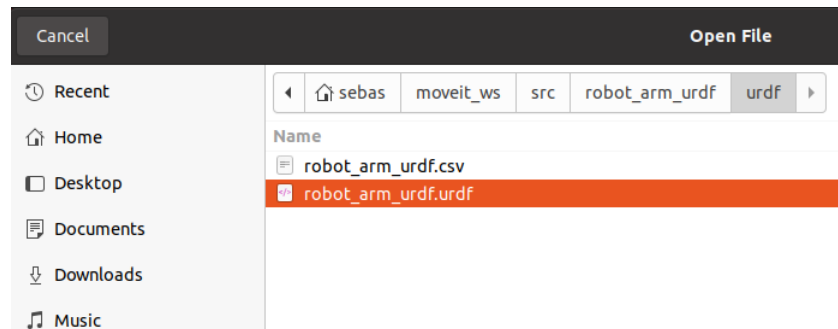
Como primer paso para poder empezar la creación de las posiciones específicas que tendrá el robot dentro del entorno virtual se procede a inicializar el software MoveIt de la siguiente

manera:

1. Se abre una nueva terminal y se inicializa el software MoveIt
2. Se procede a abrir el modelo con formato urdf ubicado en la carpeta importada desde SolidWorks (figura 27).

Figura 27

Ubicación del modelo en formato URDF.

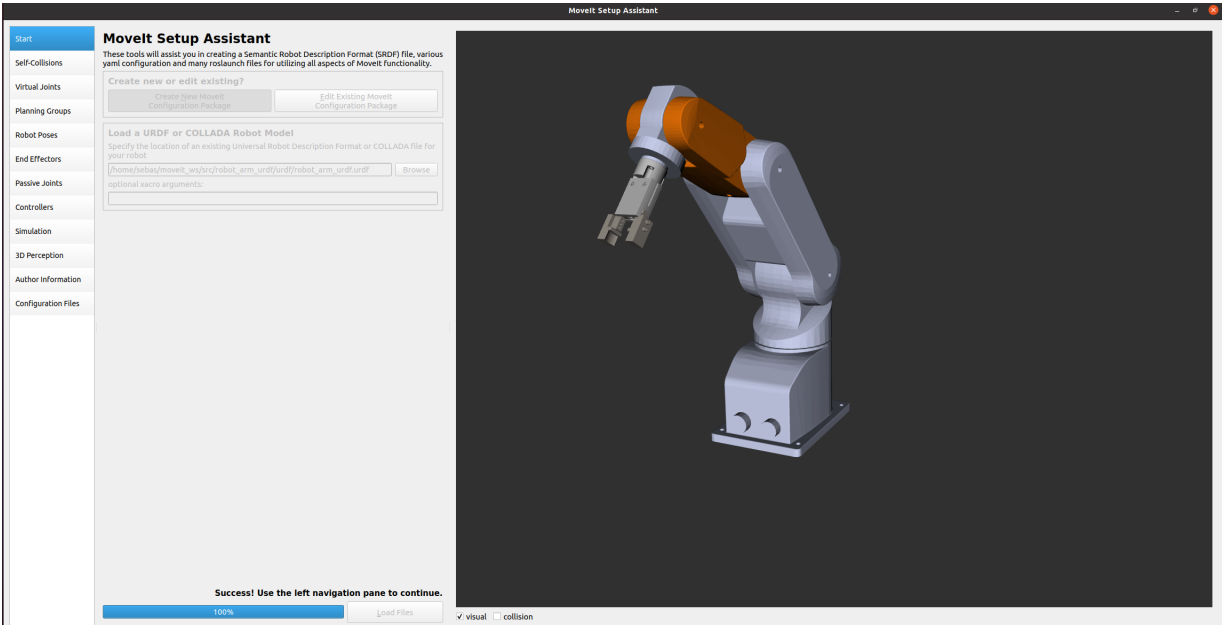


Nota: Selección del modelo en formato URDF para su configuración dentro del software MoveIt.

Una vez seleccionado el modelo en la dirección especificada se observará el mismo cargado dentro del entorno del software MoveIt (figura 28).

Figura 28

Modelo dentro del software MoveIt.

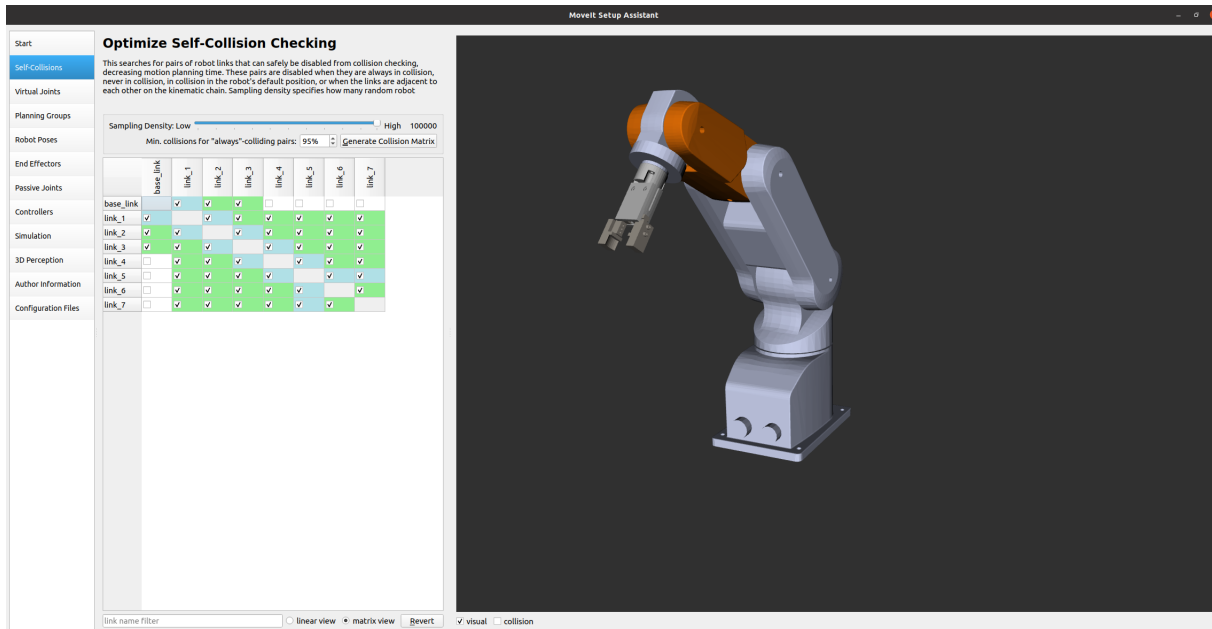


Nota: Modelo dentro del software MoveIt.

Como parámetro a considerarse, se debe generar la optimización de comprobación de colisiones mostrados en la (figura 29).

Figura 29

Optimización de comprobación de colisiones.



Nota: Optimización de colisiones del modelo dentro del software MoveIt.

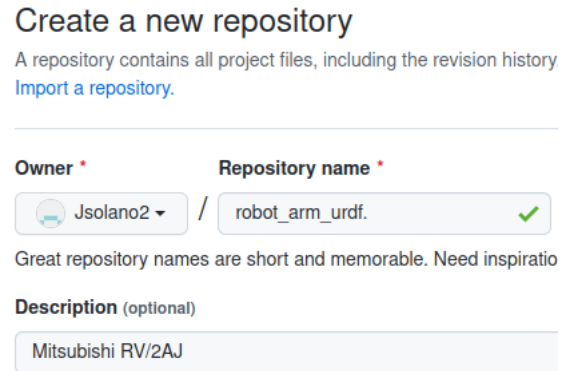
6.4. Creación de repositorio GitHub

GitHub es una plataforma de desarrollo colaborativo para alojar proyectos utilizando el sistema de control de versiones Git. Ofrece una interfaz web donde los desarrolladores pueden alojar y revisar el código, administrar proyectos y construir software. Esta plataforma también facilita el trabajo en equipo, proporcionando herramientas de colaboración para compartir y discutir código, realizar pruebas y desplegar aplicaciones.

Como primer paso se debe crear una cuenta en la página web oficial de GitHub, de esta manera se procede a crear el repositorio con el mismo nombre de la carpeta de archivos en donde se encuentra el modelo del robot antropomórfico y la carpeta en la cual se creó la simulación del mismo, se procede con la creación como se muestra en la (figura 30 y 31).

Figura 30

Creación de repositorio GitHub para modelo del robot.



Create a new repository
A repository contains all project files, including the revision history
[Import a repository.](#)

Owner * **Repository name ***

/

Great repository names are short and memorable. Need inspiratio

Description (optional)

Nota: Página de GitHub en donde se permite la creación de repositorio del modelo del robot antropomórfico.

Figura 31

Creación de repositorio GitHub para la simulación del modelo.



Create a new repository
A repository contains all project files, including the revision history
[Import a repository.](#)

Owner * **Repository name ***

/

Great repository names are short and memorable. Need inspiratic

Description (optional)

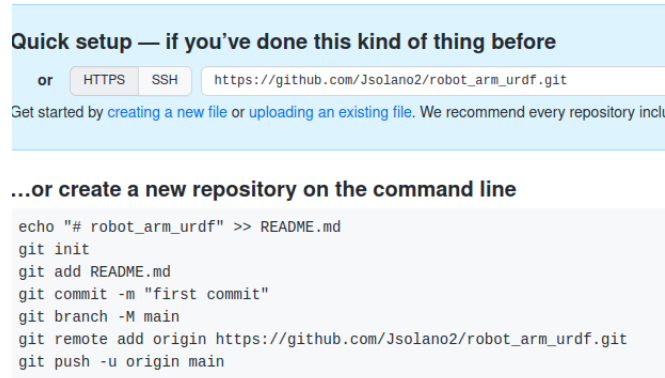
Nota: Página de GitHub en donde se permite la creación de repositorio acerca de la carpeta de simulación del robot.

Una vez creado el repositorio, la página web nos otorgará un código único para cada uno de los repositorios, con el cual se podrá realizar la exportación de archivos directamente desde la carpeta en donde se encuentra el modelo del robot antropomórfico hacia el repositorio

online como se muestra en la (figura 32).

Figura 32

Código para la exportación de archivos hacia GitHub.



The image shows a screenshot of the GitHub website's 'Quick setup' section. It features a light blue header with the text 'Quick setup — if you've done this kind of thing before'. Below this, there are two buttons: 'HTTPS' (selected) and 'SSH'. To the right of the buttons is the repository URL: 'https://github.com/Jsolano2/robot_arm_urdf.git'. A line of text below the buttons reads: 'Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a README file.' Below this is a section titled '...or create a new repository on the command line' with a list of terminal commands:

```
echo "# robot_arm_urdf" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/Jsolano2/robot_arm_urdf.git
git push -u origin main
```

Nota: Código necesario para la exportación de archivos hacia el repositorio creado en GitHub.

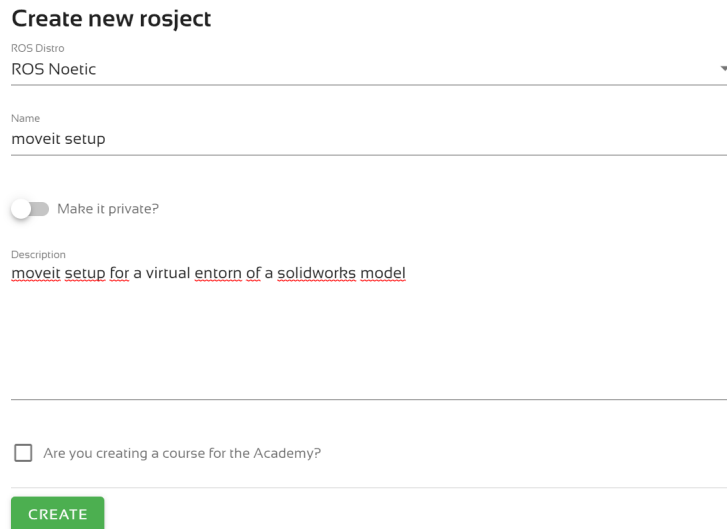
6.5. Creación del entorno web en la plataforma TheConstructSim

TheConstructSim es una plataforma de simulación de robots para la industria de la fabricación que permite a los usuarios diseñar, programar y simular robots industriales antes de implementar los sistemas de control de robots en el entorno real. Esta plataforma se utiliza para mejorar la productividad y la eficiencia al reducir los costos de fabricación y los tiempos de puesta en marcha. Esta plataforma es compatible con robots de varios fabricantes, como KUKA, ABB, FANUC, Universal Robots, etc.

Principalmente se empezará con la creación del entorno de ROS en la plataforma TheConstructSim como se muestra en la (figura 33).

Figura 33

Plataforma *TheConstructSim*.



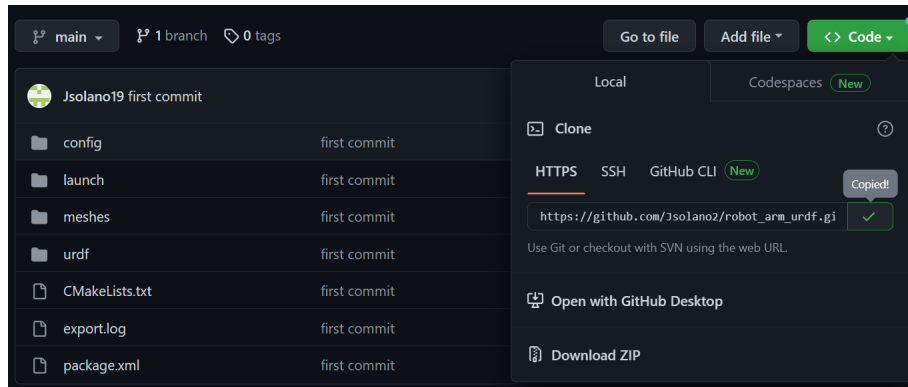
The screenshot shows a web form titled "Create new rosject". At the top, there is a dropdown menu for "ROS Distro" with "ROS Noetic" selected. Below this is a text input field for "Name" containing "moveit setup". A toggle switch for "Make it private?" is currently turned off. The "Description" field contains the text "moveit setup for a virtual entorn of a solidworths model". At the bottom, there is a checkbox for "Are you creating a course for the Academy?" which is unchecked. A green "CREATE" button is positioned at the bottom of the form.

Nota: Creación del entorno virtual de ROS en la plataforma TheConstruct.

Luego de realizar la creación de la plataforma, se procederá a copiar los links de los repositorios cargados previamente en Github como se muestra en las (figuras 34 y 35).

Figura 34

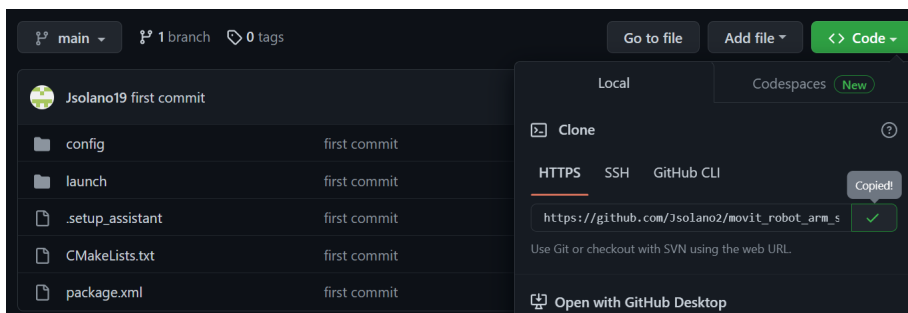
Copia de repositorio del modelo de robot antropomórfico.



Nota: Copiado de link del repositorio creado en GitHub para el modelo del robot antropomórfico.

Figura 35

Copia de repositorio de la simulación del robot antropomórfico para creación de trayectorias.



Nota: Copia de link del repositorio creado en GitHub para la simulación del robot antropomórfico para creación de trayectorias en RVIZ.

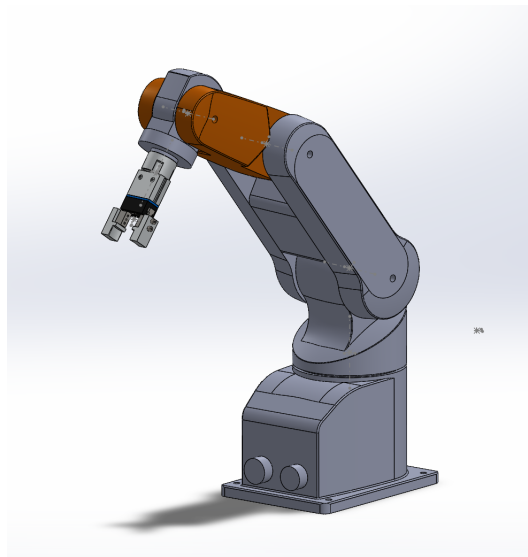
7. Resultados

7.1. Modelo del robot Mitsubishi RV-2AJ planteado en el software SolidWorks

El modelo realizado en el software SolidWorks sirve como base para la posterior realización de puntos específicos y trayectorias con los softwares MoveIt y RViz, para lo cual se plantea el modelo como se muestra en la (figura 36).

Figura 36

Modelo del robot antropomórfico en SolidWorks.



Nota: En el software SolidWorks se puede realizar el ensamble de cada uno de los elementos del robot Mitsubishi RV-2AJ necesarios para su simulación. (Abedloo, 2014)

7.2. Creación de posiciones específicas para el robot Mitsubishi RV-2AJ en el software MoveIt

Se define los grupos de acción que tendrá el robot para su posterior configuración en distintas posiciones específicas, se plantea por lo tanto dos grupos mismos que tendrán sus respectivos eslabones de acción (figura 37):

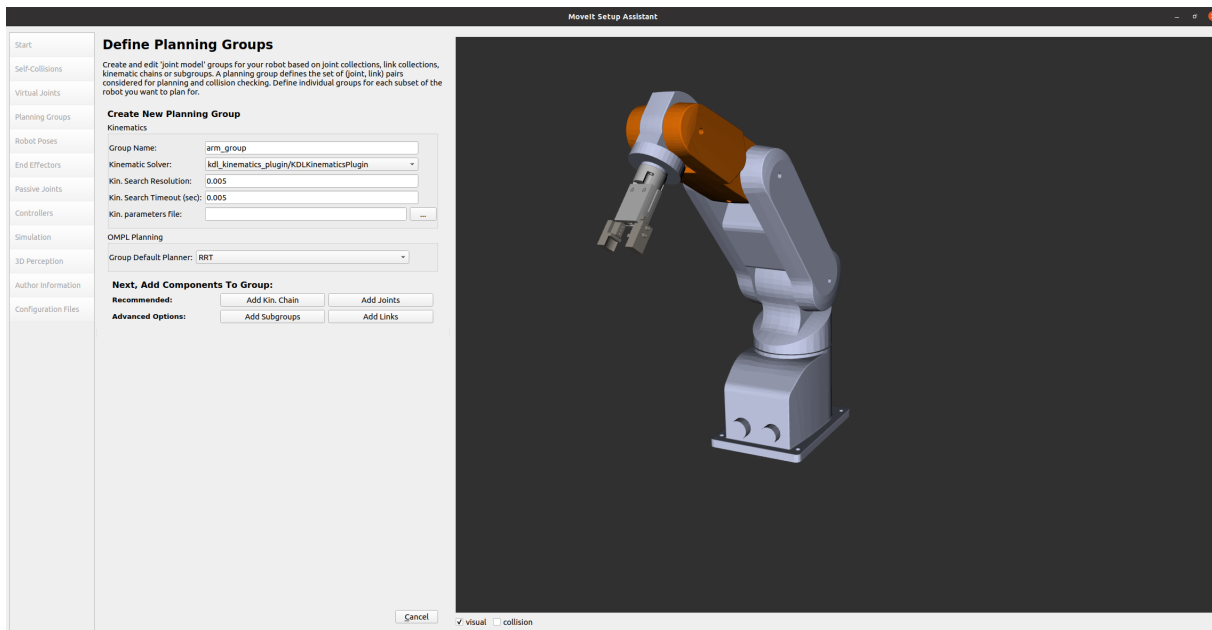
- arm_gruop

- hand

En el caso del grupo arm se debe tener en cuenta la selección correcta del kinematic solver:

Figura 37

Grupos de planificación brazo.

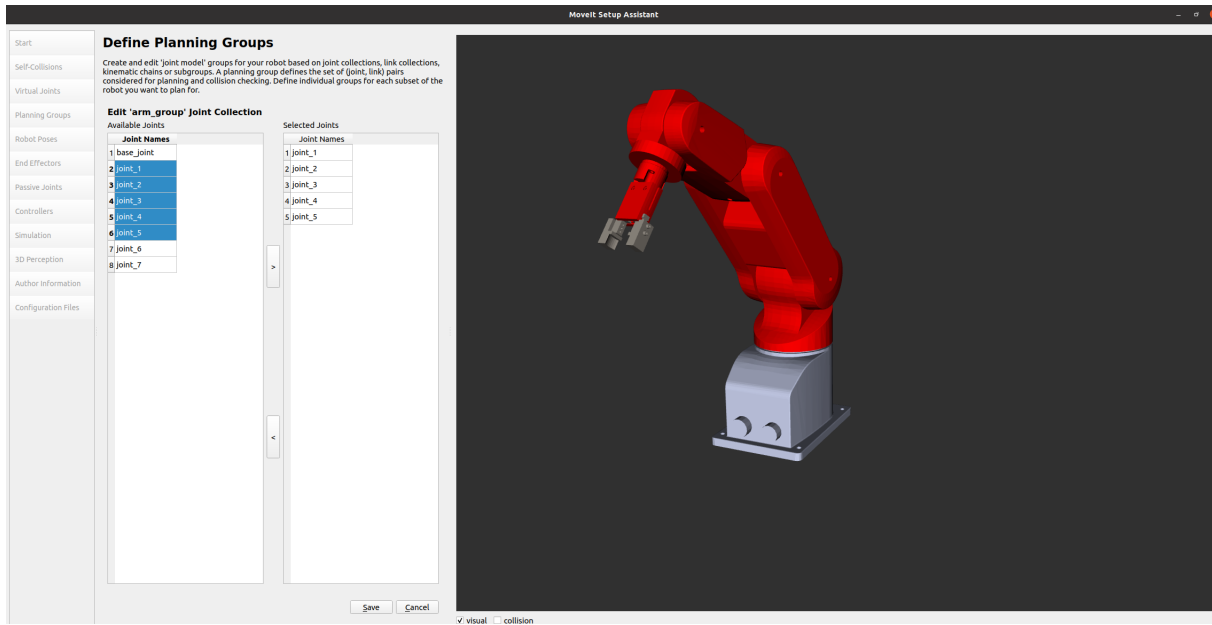


Nota: Creación del grupo arm con su respectivo solucionador cinemático.

Posteriormente, se selecciona los eslabones que tendrá cada uno de los grupos de acción (figura 38 y 39):

Figura 38

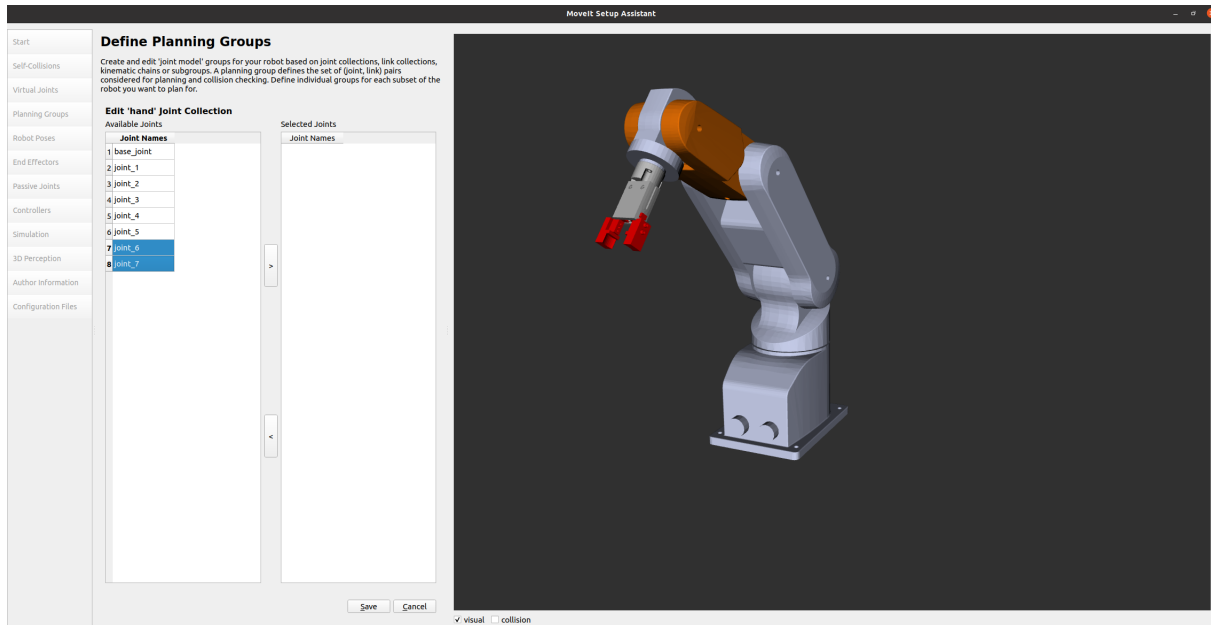
Eslabones del grupo de planificación brazo.



Nota: Selección de eslabones de acción para el grupo arm.

Figura 39

Pinzas del grupo de planificación hand.

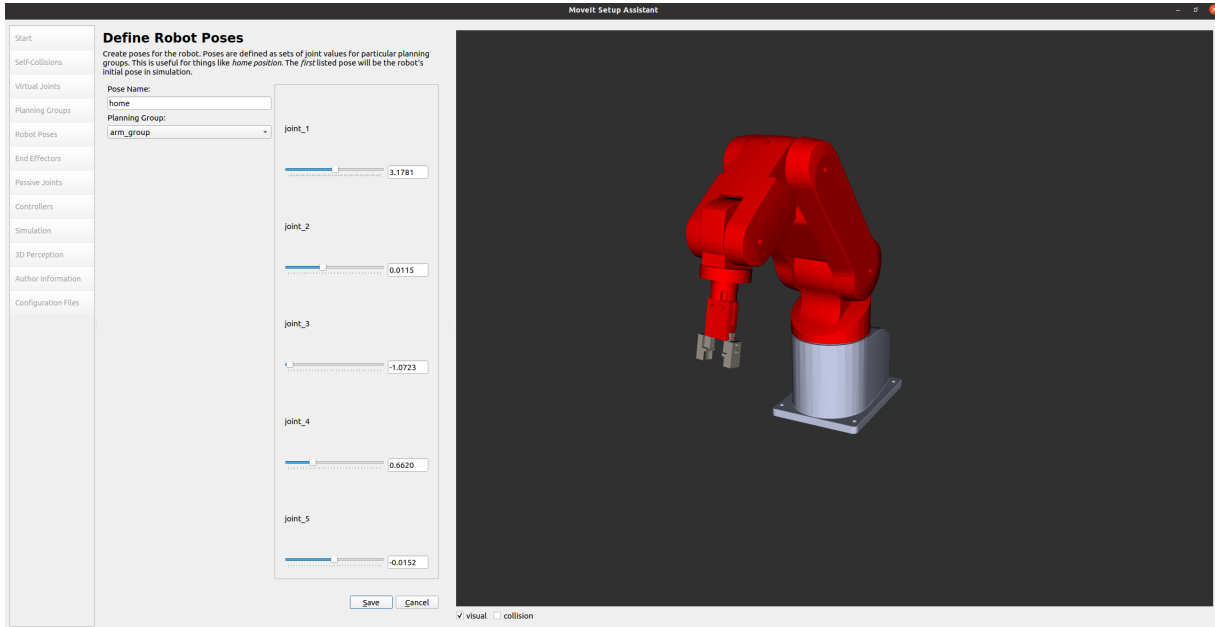


Nota: Selección de pinzas del gripper para el grupo hand.

A continuación, en la (figura 40) se especifica que, una vez creados los grupos de accionamiento, se procede a crear las distintas posiciones que tomará el robot, mismas que serán definidas por el usuario para su posterior planificación de trayectorias.

Figura 40

Creación de posiciones específicas del robot.



Nota: Entorno de creación de posiciones del robot para posterior planificación de trayectorias.

Se presenta en la (tabla 4) cada una de las posiciones que se especificó para este modelo con su respectivo grupo de accionamiento:

Tabla 4

Posiciones especificadas para el robot Mitsubishi RV-2AJ.

Posición	Grupo de acción
home	arm_group
obj1	arm_group
dejar_obj1	arm_group
obj2	arm_group
levantar_obj2	arm_group
obj3	arm_group
levantar_obj3	arm_group
dejar_obj2	arm_group
dejar_obj3	arm_group
gripper_open	hand
gripper_close	hand

Nota: La tabla presenta el número de eslabones y de juntas que permitirán calcular la cantidad de grados de libertad del robot. Chocho y Cuenca (2022)

7.3. Planificación de trayectorias mediante el software RVIZ

RViz ROS es una herramienta de visualización 3D destinada a la robótica, basada en ROS (Robot Operating System). Esta herramienta se utiliza para visualizar datos de sensores, navegación, planificación y otros datos de robótica, y proporciona una interfaz gráfica para la depuración de algoritmos de robótica. RViz ROS también puede visualizar los datos para la supervisión y el control remotos. Esta herramienta está escrita en C++ y se ejecuta en Linux y Windows.

Se procede a cargar el modelo configurado con las posiciones específicas establecidas, las cuales serán consideradas para la planificación de trayectorias en la simulación del robot Mitsubishi RV-2AJ como se puede observar en la (figura 41). Para realizar esto se debe seguir los siguientes pasos:

1. Abrir una nueva terminal y proceder a escribir los comandos que nos direccionarán a la carpeta en la cual se tiene cargado el modelo:

- `cd /moveit_ws`

2. Cargar el entorno ROS:

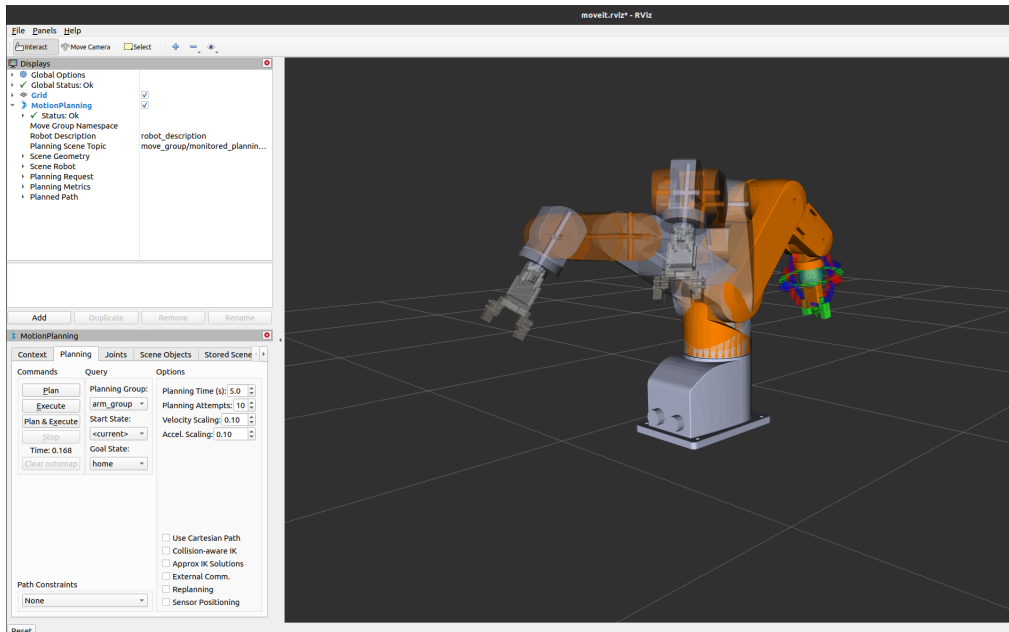
- `source devel/setup.bash`

3. Se inicializa el entorno de RViz cargando el modelo previamente configurado:

- `roslaunch movit_robot_arm_sim full_robot_arm_sim.launch`

Figura 41

Entorno RViz para planificación de trayectorias del robot.



Nota: Entorno de planificación de trayectorias del robot con las posiciones previamente configuradas en MoveIt.


Con este software se procede a planificar cada una de las trayectorias entre las posiciones que se preconfiguró en el software MoveIt, permitiendo la visualización del movimiento simulado que tendrá el brazo robótico Mitsubishi RV-2AJ en el espacio de trabajo.

7.4. Simulación Web en plataforma TheConstructSim

En la plataforma TheConstructSim se procede a iniciar una terminal en la cual se escribe el código para la clonación de las carpetas ubicadas en los repositorios de GitHub dentro de la plataforma virtual como se muestra en las (figuras 42 y 43).

Figura 42

Clonación del repositorio del modelo de robot antropomórfico.

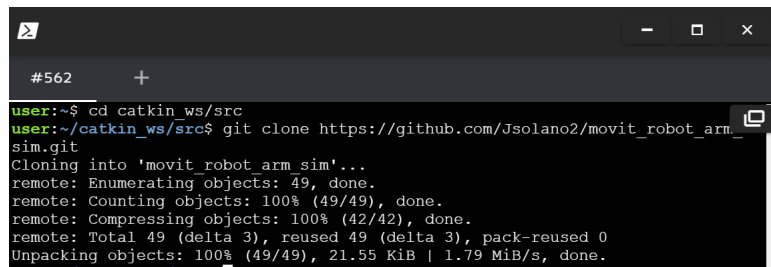


```
#559 +
user:~$ cd catkin_ws/src
user:~/catkin_ws/src$ git clone https://github.com/Jsolano2/robot_arm_urdf.git
```

Nota: Clonación dentro de la terminal de la plataforma del repositorio creado en GitHub para el modelo del robot antropomórfico.

Figura 43

Clonación del repositorio de la simulación del robot antropomórfico para creación de trayectorias.



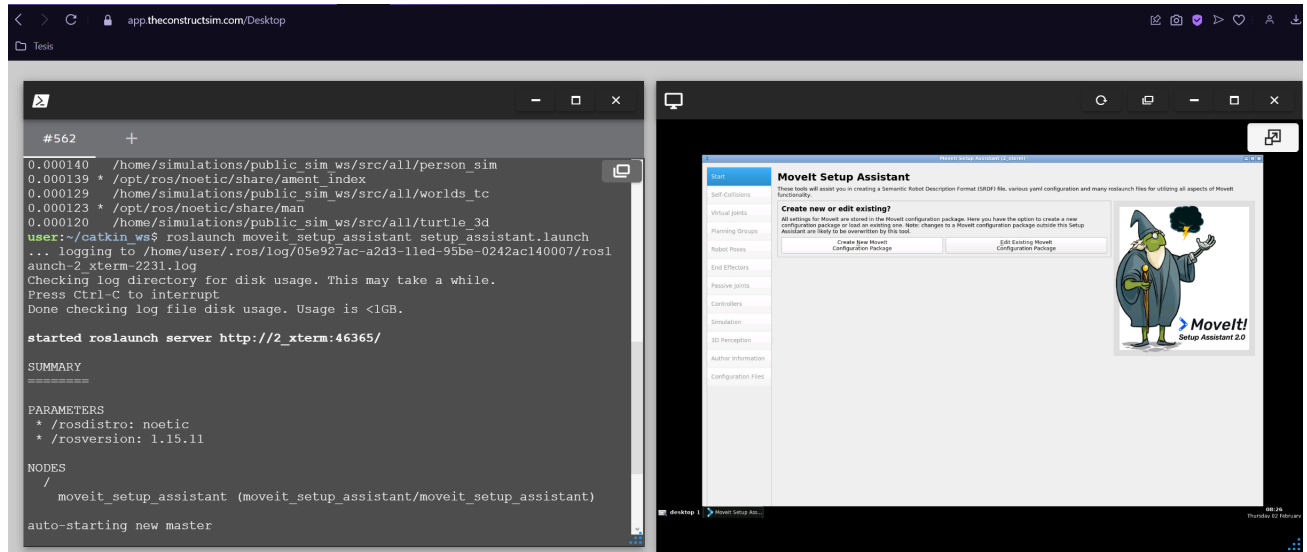
```
#562 +
user:~$ cd catkin_ws/src
user:~/catkin_ws/src$ git clone https://github.com/Jsolano2/movit_robot_arm_sim.git
Cloning into 'movit_robot_arm_sim'...
remote: Enumerating objects: 49, done.
remote: Counting objects: 100% (49/49), done.
remote: Compressing objects: 100% (42/42), done.
remote: Total 49 (delta 3), reused 49 (delta 3), pack-reused 0
Unpacking objects: 100% (49/49), 21.55 KiB | 1.79 MiB/s, done.
```

Nota: Clonación dentro de la terminal de la plataforma del repositorio creado en GitHub para la simulación del robot antropomórfico para creación de trayectorias en RVIZ.

Consecuente a lo realizado, se escribe el código para la inicialización del software MoveIt dentro de la plataforma virtual. (figura

Figura 44

Software MoveIt dentro de la plataforma virtual TheConstructSim.

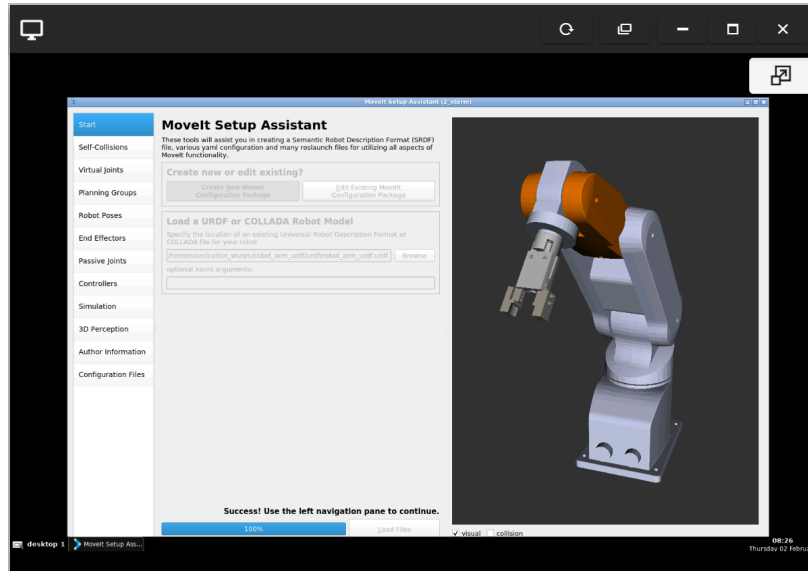


Nota: Inicialización del software MoveIt con ayuda de la terminal de la plataforma virtual TheConstructSim.

Se realiza la apertura del modelo del robot antropomórfico en la plataforma virtual gracias al software MoveIt.(figura

Figura 45

Modelo del robot antropomórfico dentro de la plataforma virtual TheConstructSim.

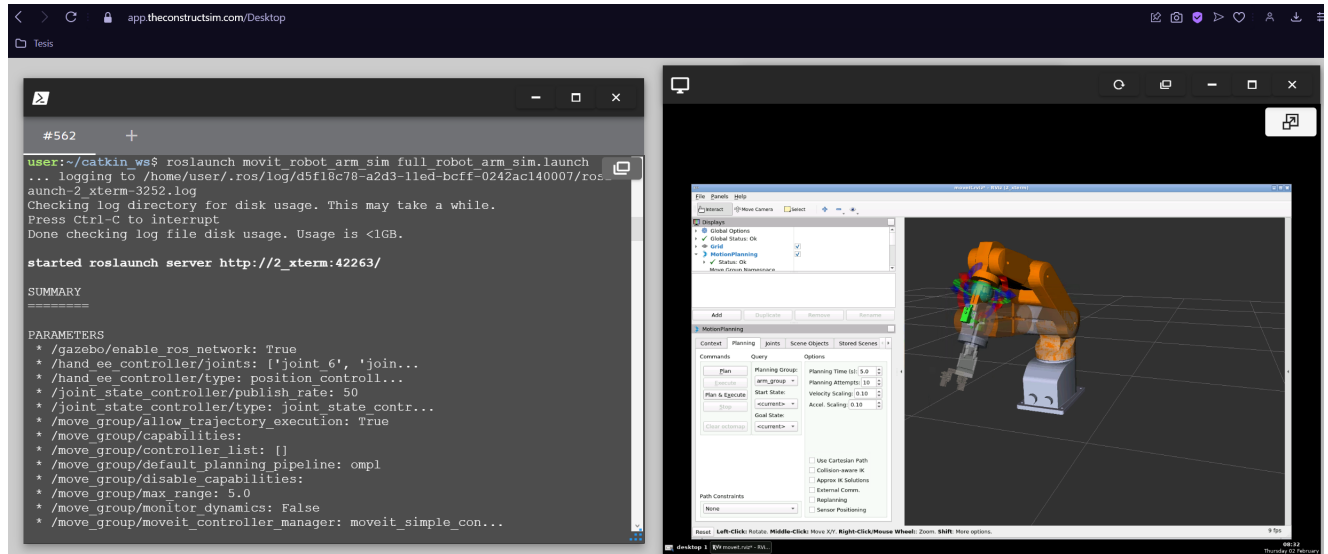


Nota: Modelo del robot antropomórfico Mitsubishi RV-2AJ dentro del software MoveIt en la plataforma virtual TheConstructSim.

De la misma manera se puede inicializar el software RViz el cual permitirá la simulación para la creación de trayectorias dentro de la plataforma virtual como se muestra en la (figura

Figura 46

Software RViz dentro de la plataforma virtual TheConstructSim.



Nota: Inicialización del software RViz con ayuda de la terminal de la plataforma virtual TheConstructSim para creación de trayectorias del robot antropomórfico.

Gracias a la plataforma virtual se puede realizar las trayectorias del robot antropomórfico a través de la web como fue descrito anteriormente en el sistema operativo Ubuntu.

8. Cronograma y actividades

Tabla 5

Cronograma de actividades.

CRONOGRAMA DE ACTIVIDADES																											
OBJETIVOS ESPECÍFICOS	ACTIVIDADES	MES 1				MES 2				MES 3				MES 4				MES 5				MES 6				HORAS	
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4		
Establecer parámetros iniciales para el movimiento remoto del brazo robótico Mitsubishi RV-2AJ con el sistema operativo ROS a través de la web.	Revisión del estado del arte de los procesos de control remoto del robot usando el sistema operativo ROS.	X																								80	
	Establecer condiciones iniciales del robot relacionadas con el funcionamiento.		X																							20	
	Manejo de características del robot y su funcionamiento con el sistema CIROS.			X	X																					20	
Proponer el movimiento remoto del brazo robótico Mitsubishi RV 2AJ con el sistema operativo ROS a través de la web.	Comunicación del brazo antropomórfico a través del sistema ROS.					X	X	X																		20	
	Generar movimientos y trayectorias del robot utilizando el sistema operativo ROS.							X	X	X																10	
	Establecer la conexión remota del sistema operativo ROS con el robot antropomórfico.										X	X	X													40	
Plantear una plataforma de entorno virtual para el control del movimiento remoto del brazo robótico Mitsubishi RV 2AJ a través de la web.	Diseñar una plataforma de entorno virtual para la comunicación remota entre el sistema operativo ROS y el robot antropomórfico.													X	X	X	X	X								40	
	Probar la comunicación y el control del robot antropomórfico a través de la conexión virtual.															X	X	X								20	
	Desarrollar una plataforma virtual para el control del robot antropomórfico a través de la web.																	X	X							40	
Implementar la plataforma de entorno virtual para el control del movimiento remoto del brazo robótico Mitsubishi RV 2AJ a través de la web.	Implementación de la plataforma de entorno virtual para el control del robot antropomórfico.																		X							20	
	Realización de pruebas de funcionamiento de la plataforma virtual.																			X						20	
	Generar documentación que sirva de guía para la utilización de la plataforma virtual.																				X	X				40	
	Redacción de documento																				X	X	X	X		80	
																							Total de horas empleadas				

Nota: La tabla presenta las actividades para cumplir con los objetivos planteados.

8.1. Presupuesto

La tabla 6 presenta los recursos materiales que serán utilizados para el desarrollo del proyecto.

Tabla 6

Recurso de talento humano.

Denominación	Cantidad	Costo Unitario	Costo total
Software SolidWorks	1	4725	4725 \$
Software Inventor	1	2300	2300 \$
Laboratorios	1	200	200 \$
Computador	1	1500	1500 \$

Nota: En la tabla se observa el costo a considerar de diferentes elementos para el desarrollo del trabajo de titulación. El costo es aproximado.

9. Conclusiones

- La solución de los problemas de cinemática directa e inversa, permite verificar el funcionamiento del robot considerando la geometría del mismo, de esta manera se logra la creación de coordenadas específicas constatando las posibilidades permitidas en el espacio de trabajo, lo cual permite establecer las características específicas que debe cumplir el diseño del robot planteado en SolidWorks.
- La interfaz gráfica generada en el Sistema Operativo Robótico (ROS) permite la interacción del modelo robótico en los softwares MoveIt y RViz, mismos que dan paso a la creación de coordenadas específicas planteadas para su funcionamiento y posterior planeación de trayectorias.
- El entorno virtual del Sistema Operativo Robótico (ROS) generado en la plataforma web TheConstructSim permite visualizar y desarrollar movimientos y trayectorias en el modelo del robot Mitsubishi RV-2AJ creando un sistema de acceso remoto.

10. Recomendaciones

Para continuación de estudio, se recomienda la conexión del entorno con el robot Mitsubishi RV-2AJ para la verificación de las trayectorias planificadas en el sistema de simulación.

Referencias

ABC-Pack. (2016). Robot antropomorfo. <https://www.abc-pack.com/productos/robot-antropomorfo/>.

Abedloo, E. (2014). Rv2aj mitsubishi robot - assembly model.

Ana María González, M. E. (s.f.). Robótica y aplicaciones. <https://www.monografias.com/trabajos10/robap/robaprobo>.

ARATECNIA. (2022). Las ventajas de los entornos virtuales. <https://aratecna.es/las-ventajas-los-entornos-virtuales/>.

Barrientos A, B. C. A. R., Peñin L. (1999). Fundamentos de robótica. <https://eltrasteroloco.files.wordpress.com/2017/03/267380685-fundamentos-de-robotica.pdf>.

Chocho, y Cuenca. (2022). Programación de procesos industriales utilizando el robot mitsubishi rv-2aj y el torno cnc emco concept turn 55. <https://dspace.ups.edu.ec/handle/123456789/23481>.

Delgado, D. O. (2017). Qué es ros (robot operating system). <https://openwebinars.net/blog/que-es-ros/>.

Esneca. (2019). Qué es el brazo robótico y en qué industrias se emplea. <https://www.esneca.com/blog/brazo-robotico-industrias/>.

Esneca. (2020). Tipos de robots industriales: clasificación y características. <https://www.esneca.lat/blog/tipos-robots-industriales-caracteristicas/>.

FESTO. (2022). Parallel grippers dhps.

Fonseca, E. (2011). Sistemateleoperadode unbrazo robótico mitsubishi rv-2aj. <https://1library.co/document/qulw9w0y-sistema-teleoperado-robotico-mitsubishi-interfaz-haptica-retroalimentacion-fuerza.html>.

Gazebo. (2022). Acerca de gazebo. <https://gazeboim.org/about>.

González, V. (2002). Estructura de un robot industrial. <http://platea.pntic.mec.es/vgonzale/cyr0204/cyr01/robotica/sistema/morfologia.htm>.

González, V. R. (2002). Robots industriales. http://platea.pntic.mec.es/vgonzale/cyr0204/ctrl_r0b/robotica/in

Industry, D. (2023). Software de programación ros. <https://www.directindustry.es/prod/automationware/product-192516-2394360.html>.

M. Montemerlo, N. R., y Thrun, S. (2003). Perspectives on standardization in mobile robot programming: The carnegie mellon navigation toolkit,.

MoveIt. (2022). Moveit. <https://moveit.ros.org>.

Márquez, M. M. (2015). Brazo robotico mitsubishi melfa rv-2aj.

<https://mariuszmerinox.wordpress.com/2015/04/13/brazo-robotico-mitsubishi-melfa-rv-2aj/>.

R., G. (2014). Sistema de seguimiento inteligente para un robot manipulador. *http://repositorio.ute.edu.ec/bitstream/123456789/5791/1/58769_1.pdf*.

RIA. (2023). “definición de robot industrial. *<https://rivasrobotics.com/que-es-un-robot-industrial/>*.

SIHE. (2016). Universidad politécnica salesiana. *<https://ius-sdb.com/red-mundial-ius/america/universidad-politecnica-salesiana-ecuador/?lang=es>*.

Splashtop. (2022). ¿qué es el acceso remoto? *<https://www.splashtop.com/es/what-is-remote-access>*.

Villalobos, L. (2015). Análisis cinemático directo del robot rv-2ajde mitsubishi utilizando teoría de screws.

VMware. (2022). VMware. *<https://www.vmware.com/latam.html>*.

ANEXOS

Anexo A: Matriz de consistencia lógica

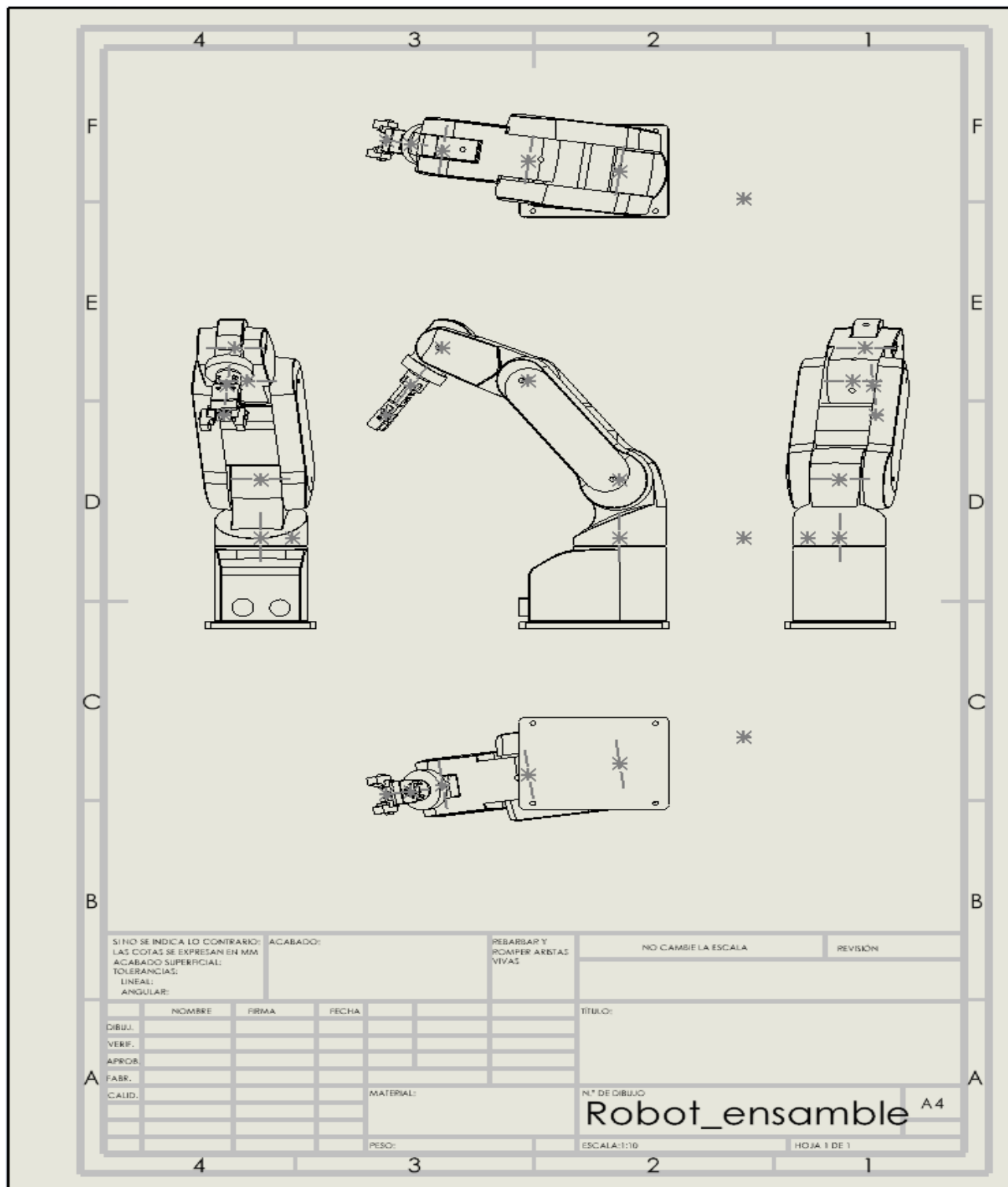
Tabla 7

Matriz de consistencia.

MATRIZ DE CONSISTENCIA				
PROBLEMA GENERAL	OBJETIVO GENERAL	HIPÓTESIS GENERAL	VARIABLES	MARCO TEÓRICO
¿Es probable controlar de manera remota el movimiento del brazo robótico Mitsubishi RV-2AJ con el sistema operativo ROS a través de la web?	Controlar de manera remota el movimiento del brazo robótico Mitsubishi RV-2AJ con el sistema operativo ROS a través de la web.	Se controlará de manera remota el movimiento del brazo robótico Mitsubishi RV-2AJ con el sistema operativo ROS a través de la web.	-VI: Sistema operativo. -VD: movimiento remoto	Robótica
PROBLEMAS ESPECÍFICOS	OBJETIVOS ESPECÍFICOS	HIPÓTESIS ESPECÍFICAS	VARIABLES	MARCO TEÓRICO
¿Es posible establecer parámetros iniciales para el movimiento remoto del brazo robótico Mitsubishi RV-2AJ con el sistema operativo ROS a través de la web?	Establecer parámetros iniciales para el movimiento remoto del brazo robótico Mitsubishi RV-2AJ con el sistema operativo ROS a través de la web.	Se establecerá parámetros iniciales para el movimiento remoto del brazo robótico Mitsubishi RV-2AJ con el sistema operativo ROS a través de la web.	-VI: Parámetros iniciales -VD: Movimiento remoto	Brazo robótico Mitsubishi RV-2AJ
¿Será factible proponer el movimiento remoto del brazo robótico Mitsubishi RV-2AJ con el sistema operativo ROS a través de la web?	Proponer el movimiento remoto del brazo robótico Mitsubishi RV-2AJ con el sistema operativo ROS a través de la web.	Se propondrá el movimiento remoto del brazo robótico Mitsubishi RV-2AJ con el sistema operativo ROS a través de la web.	-VI: Sistema operativo. -VD: Movimiento remoto	Programación de equipos con el sistema operativo ROS Acceso remoto
¿Se podrá plantear una plataforma de entorno virtual para el control del movimiento remoto del brazo robótico Mitsubishi RV-2AJ con el sistema operativo ROS a través de la web?	Plantear una plataforma de entorno virtual para el control del movimiento remoto del brazo robótico Mitsubishi RV-2AJ a través de la web.	Se planteará una plataforma de entorno virtual para el control del movimiento remoto del brazo robótico Mitsubishi RV-2AJ con el sistema operativo ROS a través de la web.	-VI: Plataforma de entorno virtual -VD: Movimiento remoto	Plataforma virtual basado en el sistema operativo ROS
¿Es viable implementar la plataforma de entorno virtual para el control del control del movimiento remoto del brazo robótico Mitsubishi RV-2AJ con el sistema operativo ROS a través de la web?	Implementar la plataforma de entorno virtual para el control del movimiento remoto del brazo robótico Mitsubishi RV-2AJ a través de la web.	Se implementará la plataforma de entorno virtual para el control del movimiento remoto del brazo robótico Mitsubishi RV-2AJ a través de la web.	-VI: Plataforma de entorno virtual -VD: Movimiento remoto	

Nota: La matriz de consistencia presentada facilita identificar la relación que existe entre las variables y los objetivos además de como se relaciona con el marco teórico referencial.

Anexo B: Dibujo del modelo realizado en SolidWorks



SI NO SE INDICA LO CONTRARIO: ACABADO: REBARBAR Y ROMPER ARISTAS VIVAS NO CAMBIE LA ESCALA REVISION
 LAS COTAS SE EXPRESAN EN MM ACABADO SUPERFICIAL: TOLERANCIAS: LINEAL: ANGULAR:

DIBUJ.	NOMBRE	FIRMA	FECHA	TITULO:
VERIF.				
APROS.				
FABR.				
CALID.				

MATERIAL: N.º DE DIBUJO Robot_ensamble A4
 ESCALA:1:10 HOJA 1 DE 1
 PESO: