



Universidad de Valladolid



ESCUELA DE INGENIERÍAS  
INDUSTRIALES

Máster en Electrónica Industrial y Automática

# MÁSTER EN ELECTRÓNICA INDUSTRIAL Y AUTOMÁTICA

ESCUELA DE INGENIERÍAS INDUSTRIALES  
UNIVERSIDAD DE VALLADOLID

TRABAJO FIN DE MÁSTER

**Diseño, simulación y fabricación de una estación  
robotizada universitaria**

Autor: D. Víctor Lobo Granado

Tutor: D. Alberto Herreros López

Valladolid, Julio, 2020.





Universidad de Valladolid



ESCUELA DE INGENIERÍAS  
INDUSTRIALES

Máster en Electrónica Industrial y Automática

# MÁSTER EN ELECTRÓNICA INDUSTRIAL Y AUTOMÁTICA

ESCUELA DE INGENIERÍAS INDUSTRIALES  
UNIVERSIDAD DE VALLADOLID

TRABAJO FIN DE MÁSTER

Diseño, simulación y fabricación de una estación  
robotizada universitaria

Autor: D. Víctor Lobo Granado  
Tutor: D. Alberto Herreros López

Valladolid, Julio, 2020.



## Resumen

En este trabajo final de master se plantea el proceso de diseño, simulación y fabricación de entornos de trabajo para una estación robotizada orientada a la docencia. Los entornos de trabajo disponen de varios objetos de trabajo, una semiesfera y un cilindro, que serán colocados en diferentes puntos de la instalación para que el robot trabaje sobre ellos. El robot real está comunicado mediante TCP/IP con un ordenador que utiliza Matlab como software principal. Ordenador y robot se comunican mediante un entorno grafico que permite simular la estación con Simulink.

Asimismo, los entornos de trabajo y la estación han sido fabricados empleando impresión 3D.

## Palabras clave

Matlab, Simulink, RobotStudio, simulación, estación, diseño.

## Abstract

In this final master project the process of design and simulation of work environments for a robotic station oriented to teaching is presented. The work environments have work objects such as a sphere and a cylinder that can be placed at different points of the installation so that the robot works on them. The robot is communicated by TCP / IP with a computer in which Matlab is used as the main software. This communicates with the robot through a graphic environment that also allows simulating the station using Simulink.

In addition, the work environments and the station have been manufactured using 3D printing.

## Keywords

Matlab, Simulink, RobotStudio, simulation, station, design.



## Agradecimientos

Quiero comenzar este trabajo agradeciendo a mis padres y mi hermana la paciencia y el apoyo recibido desde el primer día. Gracias por estar dónde y cuándo se os ha necesitado y por hacerme mejorar y esforzarme cada día. No sería la persona que soy hoy sin vuestra gran aportación.

También tengo que agradecer la aportación de Alberto Herreros López. Su idea de hacer que un robot trabaje en orientaciones inusuales y en nuevos entornos de trabajo me ha permitido completar el proyecto que he desarrollado.





# ÍNDICE

<b>CAPITULO 1. INTRODUCCION, OBJETIVOS Y ESTADO DEL ARTE</b> .....	1
1.1. Introducción .....	1
1.2. Objetivos .....	1
1.3. Estado del arte .....	1
<b>CAPITULO 2. DISEÑO PREVIO</b> .....	3
2.1. Análisis de Soluciones .....	3
2.1.1. Robot .....	3
2.1.2. IDE de desarrollo .....	4
2.1.3. Interfaz de usuario (GUI) .....	5
2.1.4. Programas CAD .....	5
2.2. Diseño 3D .....	6
2.3. Estructura software .....	6
2.4. Comunicaciones .....	7
<b>CAPITULO 3. DESARROLLO DEL PROYECTO</b> .....	9
3.1. Diseño 3D con Catia .....	9
3.1.1. Semiesfera .....	9
3.1.2. Cilindro .....	10
3.1.3. Soporte de herramientas y herramientas .....	11
3.1.4. Mesa de trabajo .....	12
3.1.5. Estructura lateral .....	13
3.1.6. Estructura superior .....	15
3.1.7. Estación completa .....	17
3.2. Programación con Matlab .....	18
3.2.1. Código de Matlab con trayectorias .....	18
3.2.2. Simulink .....	25
3.2.3. App Designer .....	32
3.3. Gestión del robot con RobotStudio .....	34
3.3.1. Declaración de datos .....	36
3.3.2. Funciones .....	36
3.3.3. Procedimientos .....	36
3.3.4. Simulación .....	37
<b>CAPITULO 4. MANUAL DE USUARIO</b> .....	39
<b>CAPITULO 5. RESULTADOS</b> .....	41
5.1. Interfaz gráfico junto a simulación en Simulink .....	41

5.2.	Interfaz gráfico junto a simulación en RobotStudio .....	53
5.3.	Objetos fabricados .....	64
5.3.1.	Semiesfera.....	64
5.3.2.	Cilindro .....	66
5.3.3.	Herramientas y soporte herramientas .....	67
5.3.4.	Anclaje a la mesa y soporte de objetos .....	69
<b>CAPITULO 6.</b>	<b>CONCLUSIONES</b> .....	<b>71</b>
<b>CAPITULO 7.</b>	<b>BIBLIOGRAFÍA</b> .....	<b>73</b>









# CAPITULO 1. INTRODUCCION, OBJETIVOS Y ESTADO DEL ARTE

## 1.1. Introducción

La robótica es una de las áreas de mayor crecimiento en la industria debido a que los procesos cada vez tienden más hacia la automatización, esto ha permitido fabricar en grandes cantidades, en tiempos muy cortos, a precios competitivos y aumentando la seguridad del personal y la calidad final del producto. Un robot industrial siempre está conectado a otros dispositivos, como los autómatas (PLC's), con los que intercambia información para que el sistema robótico general funcione.

Centrándonos en el ámbito educativo, este proyecto abre la puerta a una amplia gama de oportunidades. Los alumnos podrían aprender sobre Matlab, Simulink, RobotStudio e incluso, introducir avances, mejoras o alternativas.

Este proyecto pretende desarrollar una estación robotizada plenamente operativa que pueda ser empleada en docencia universitaria, comenzando con el diseño y simulación de la estación hasta la fabricación de los elementos que formen parte de ella. En este campo permitirá, por ejemplo, el desarrollo de nuevas prácticas de asignaturas relacionadas con robótica que permitan ver el alcance de operación de un robot, el diseño y fabricación de modelos 3D y la distribución e integración del software Matlab en la robótica.

La estación robotizada ofrece una amplia gama de posibilidades, dando lugar a un sistema muy abierto que permitirá incorporar y probar nuevas rutinas y tareas novedosas.

## 1.2. Objetivos

El objetivo principal de este TFM es el desarrollo de entornos de trabajo para la estación robotizada de la Escuela de Ingenierías Industriales. Para conseguir este objetivo general se han marcado diferentes objetivos intermedios:

- Diseño CAD y fabricación de los entornos de trabajo con impresión 3D.
- Programación de trayectorias del robot en los entornos de trabajo.
- Simulación de los entornos de trabajo
- Diseño del interfaz gráfico de control de la estación
- Programación de la comunicación con el robot.

## 1.3. Estado del arte

La Escuela de Ingenierías Industriales de la Universidad de Valladolid dispone en uno de sus laboratorios de una instalación formada por un robot ABB-IRB-120, anclado firmemente sobre una mesa junto con una serie de elementos y entornos para realizar diversas prácticas en asignaturas impartidas en la escuela.

Estos elementos y entornos han sido incorporados a la instalación robotizada a través de proyectos educativos dentro del departamento de Sistemas y Automática de la Escuela de Ingenierías Industriales y por trabajos de final de grado y master de diferentes alumnos.

El profesor, y tutor de este proyecto, Alberto Herreros López dotó al robot de la capacidad de escribir sobre un papel el texto leído de un fichero. Se podían emplear distintos tipos de letra y distintos planos de escritura.

En 2014, Miguel Ángel Mato San José [1] realizó su proyecto final de carrera que consistía en el desarrollo de una interfaz gráfica desde Matlab que, haciendo uso de la librería “Robotic ToolBox”, movía el robot ABB IRB 120 simulado simultáneamente con el robot real mediante comunicación OPC.

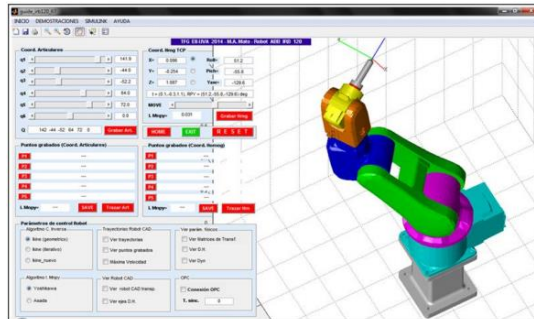


Figura 1 Interfaz gráfico que mueve el robot vía OPC

Entre 2015 y 2016, Juan Antonio Ávila Herrero [2] y Álvaro Galindo de los Santos [3], realizaron sus proyectos de manera simultánea dando lugar a la estación robótica actual de la escuela. Juan Antonio Ávila Herrero, modeló la célula robótica utilizando el software propio de ABB RobotStudio y Álvaro Galindo de los Santos, centró su proyecto en el control remoto del robot mediante tecnología WIFI. Ambos trabajaron con fines docentes diseñando diferentes escenarios de educación.

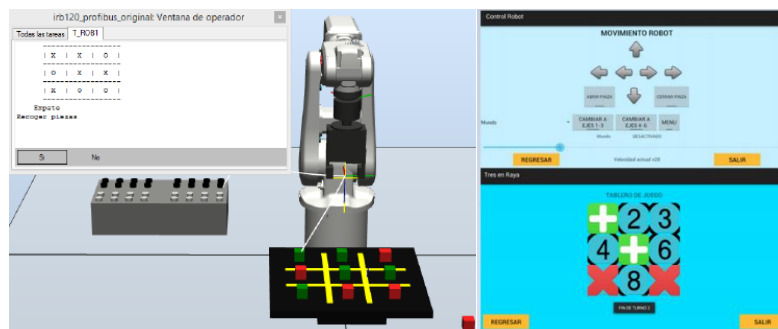


Figura 2 Célula robotizada en RoboStudio y aplicación Android comunicada vía WIFI

En 2019, Carlos Jiménez Jiménez [4] realizó su trabajo final de master que consistía en el desarrollo de un sistema educativo capaz de jugar al ajedrez con un robot industrial vía TCP/IP a través de Matlab. A través de un entorno gráfico el usuario podía definir su jugada y ver el tablero.



Figura 3 Robot ABB-IRB-120 jugando al ajedrez



## CAPITULO 2. DISEÑO PREVIO

En este capítulo se describe el análisis de soluciones y elementos disponibles para la ejecución de este proyecto, el diseño de la solución más válida para el proyecto y su estructura software y la comunicación entre los diferentes elementos del sistema.

### 2.1. Análisis de Soluciones

#### 2.1.1. Robot

La Escuela de Ingenierías Industriales de la Universidad de Valladolid dispone de dos modelos diferentes de robots industriales, un UR3, Figura 4, y un ABB-IRB-120, Figura 5.

El robot UR3 [5] es un robot colaborativo ultraligero y compacto, ideal para la aplicación sobre mesas de trabajo. Su tamaño reducido lo convierte en el más adecuado para implementarse directamente dentro de maquinaria o en otros espacios de trabajo pequeños. Este robot actualmente se emplea en tareas de investigación en cirugía endonasal en un laboratorio de la escuela.



*Figura 4 Robot de Universal Robots UR3*

El robot ABB-IRB-120 [6] es un robot industrial multiusos más pequeño de ABB únicamente 25 kg y capaz de soportar una carga de 3 kg, con un alcance de 580mm. Es el robot empleado en la docencia en la asignatura de 4º Curso, Sistemas Robotizados en el Grado en Electrónica Industrial y Automática, por lo que dispone de un entorno preparado para la realización de las prácticas de la asignatura. El robot se encuentra anclado a una mesa donde se encuentran ubicados diferentes objetos sobre los que operar, un soporte de rotuladores, una pizarra inclinada sobre la que dibujar y un tablero para jugar al 3 en raya y al ajedrez. A su vez, esta mesa se encuentra dentro de un vallado lateral que actúa como protección durante el funcionamiento del robot.



Figura 5 Robot ABB-IRB-120

El robot escogido para este proyecto es el robot ABB-IRB-120 ya que es el robot disponible orientado a la docencia y cuenta con un entorno preparado para ello. Al tratarse de un robot de la empresa ABB, el software necesario para su programación es RobotStudio, programado en lenguaje propio como es RAPID (Robotics Application Programming Interactive Dialogue). Este software nos permite programar el robot y también realizar simulaciones con los objetos y trayectorias que formarán parte de la estación.

#### 2.1.2. IDE de desarrollo

Para poder realizar la gestión de las trayectorias y la comunicación con el robot se requiere de un entorno de desarrollo que sea capaz de agrupar todo el código de la gestión de la aplicación.

Dentro de las múltiples opciones que existen actualmente, la primera alternativa que se plantea es la herramienta Matlab. Matlab es un sistema de cómputo numérico que ofrece un IDE, entorno de desarrollo integrado, disponible para las plataformas Unix, Windows, macOS y GNU/Linux. Posee un lenguaje de programación propio y entre sus prestaciones básicas se hallan la manipulación de matrices, presentes en el campo de la robótica, 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.

El IDE de desarrollo Visual Studio es un entorno integrado para las mismas plataformas que Matlab, donde se puede programar en diferentes lenguajes como C++, C#, Python o Visual Basic. El entorno de desarrollo Qt es similar a Visual Studio, con la diferencia de que la aplicación creada puede ser compilada en cualquier sistema operativo, mientras que en Visual Studio emplea .NET Framework.



Figura 6 Entornos de desarrollo Matlab, Visual Studio y Qt

Analizando las alternativas disponibles, Figura 6, el entorno en el que se desarrollará la aplicación será Matlab, por su capacidad de trabajo con matrices y la posibilidad de realizar, además, simulaciones con su herramienta Simulink.

### 2.1.3. Interfaz de usuario (GUI)

El interfaz de usuario es un conjunto de objetos gráficos para representar la información y acciones disponibles dentro de la aplicación diseñada.

Matlab dispone de varias herramientas para diseñar interfaces para las aplicaciones. Dispone de la herramienta GUIDE y de App Designer. El entorno App Designer es el más adecuado, puesto que la herramienta GUIDE será eliminado en versiones futuras de Matlab. Éste nos permite desarrollar tanto la organización de los componentes visuales como botones, barras deslizantes o entradas numéricas como la programación orientada a objetos del comportamiento de la aplicación.

Tanto Visual Studio como Qt son entornos de desarrollo que están orientados al diseño gráfico de aplicaciones. Disponen de librerías muy completas y de una gran comunidad en el entorno de estos interfaces.

Analizando las alternativas disponibles, Figura 6, el interfaz de usuario se realizará empleando la herramienta de Matlab App Designer, para tener todo integrado en la misma aplicación.

### 2.1.4. Programas CAD

A la hora de modelar objetos en 3D existen multitud de programas, pero en este caso se buscarán softwares de diseño industrial como son Catia y Autodesk Fusion e Inventor.

Catia, realizado por Dassault Systèmes, es un programa informático de diseño, fabricación e ingeniería asistida por computadora. El programa está desarrollado para proporcionar apoyo desde la concepción del diseño hasta la producción y el análisis de productos.

Autodesk dispone de varios programas CAD como son Fusion 360 o Inventor. Autodesk Fusion 360 combina el diseño industrial y mecánico en una sola herramienta mientras que Autodesk Inventor combina diseño mecánico avanzado en 3D, con modelado paramétrico, directo y libre, tiene una capacidad base para realizar diseño de piezas, sus dibujos y ensamblajes de partes, como el resto de softwares mencionados.

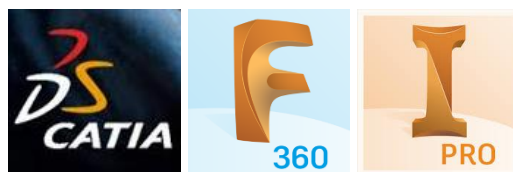


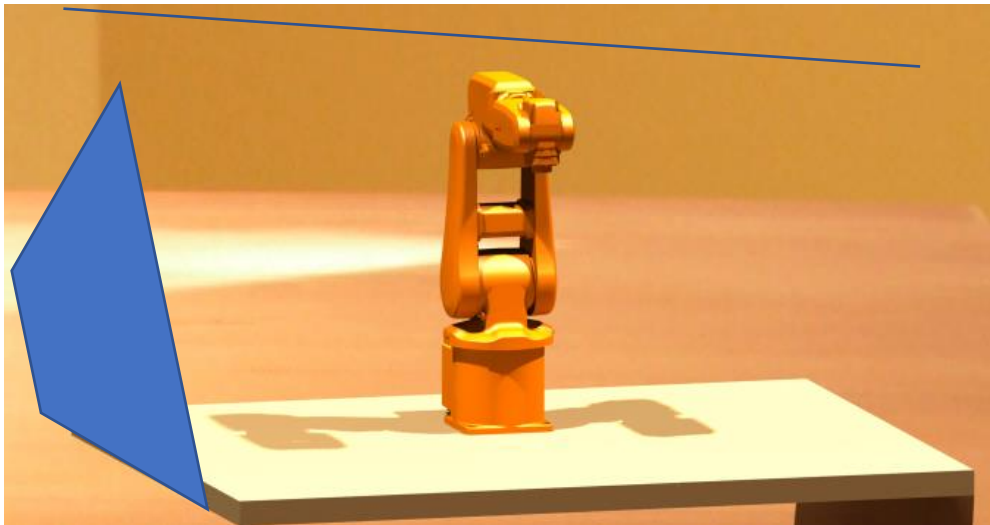
Figura 7 Softwares de diseño CAD Catia, Fusion 360 e Inventor

Analizando las alternativas disponibles, Figura 7, el software de diseño CAD a emplear es Catia, por la formación adquirida en el Master, se trata de una herramienta conocida, potente, robusta y ampliamente empleada en el mundo laboral.

## 2.2. Diseño 3D

Un aspecto importante en la realización de entornos de trabajo para el robot es el diseño de los mismos. Empleando el software CAD se han diseñado varios objetos con los que el robot debe interactuar y la estructura de soporte de los mismos.

Los entornos de trabajo disponibles hasta el momento hacen que el robot actúe con una orientación perpendicular a la mesa o con una ligera inclinación respecto a ésta. Esto impide que se pueda apreciar completamente el volumen de trabajo o las diferentes configuraciones que pueden adquirir el robot. Por eso se pretende realizar entornos de trabajo en los que el robot tenga que trabajar tanto sobre la mesa como en una “pared lateral” y en el “techo” de la estación, Figura 8.



*Figura 8 Lugares de trabajo previstos para el robot*

Los objetos sobre los que va a interactuar el robot se caracterizan por su flexibilidad tanto en montaje como en colocación. Así, pueden estar en distintos lugares y posiciones, y al mismo tiempo, ser modulares ante posibles ampliaciones y reparaciones. Además, se completa al robot una serie de herramientas nuevas para emplear en este entorno.

## 2.3. Estructura software

Con los elementos que van a emplearse en el proyecto ya seleccionados, es el momento de organizar la separación de los distintos niveles que le forman: la programación en Matlab, incluyendo las trayectorias del robot, la simulación y el interfaz gráfico; y todo lo relativo al robot ABB-IRB-120 y su software RobotStudio.

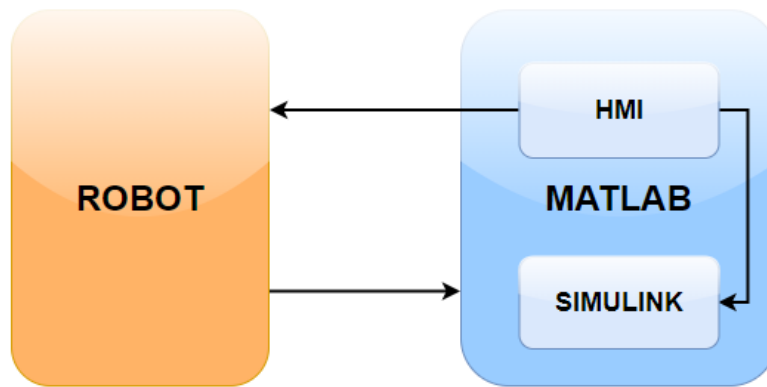


Figura 9 Módulos de la estructura software

Como se puede apreciar en la Figura 9, el software Matlab ejerce de nodo principal y es el que alberga tanto el HMI (Human-Machine Interface) como el entorno de simulación Simulink. Matlab se comunicará con el robot real y su software RobotStudio para realizar trayectorias y movimientos.

Se podrá utilizar el módulo de Matlab si queremos realizar únicamente simulaciones de movimiento y trayectorias dentro de la estación diseñada o se podrá utilizar Matlab, HMI y simulación con Simulink, junto al robot real y la estación real.

#### 2.4. Comunicaciones

La comunicación entre el nodo de Matlab y el nodo del robot emplea el modelo TCP/IP que es un grupo de protocolos que conforman la base de internet y otras redes.

El modelo TCP/IP, [7], cuenta con una gran ventaja: funciona con independencia del hardware y el software subyacente. No importa qué sistema operativo o dispositivo se use para la comunicación a través de la red, porque los protocolos están estandarizados de tal forma que funcionan en cualquier contexto.

La capa de transporte, TCP, [8], proporciona un transporte de datos fiable, su función es hacer llegar los paquetes de datos sin errores y en secuencia, y la capa de internet, IP, determina la ruta y el destinatario del paquete de datos, fragmenta el mensaje y lo reensambla para que se pueda transmitir correctamente.

Dentro del modelo TCP/IP existen más capas aparte de las dos ya mencionadas, estas capas, Figura 10, están poseen una jerarquía y cada capa provee y demanda de servicios a las capas superiores e inferiores respectivamente.



Figura 10 Estructura del modelo TCP/IP

En un protocolo TCP/IP a conexión, no hay relaciones habituales como maestro/esclavo, sino que utilizan un modelo cliente/servidor en las comunicaciones. El servidor, como su propio nombre indica, ofrece un servicio a usuarios mientras que el cliente es el que solicita este servicio. Un servidor puede interactuar con múltiples clientes al mismo tiempo.

En el caso de este proyecto, en función de los servicios que solicita o presta cada bloque, se puede entender que el robot actúa como servidor y Matlab actúa como cliente. Matlab solicita al robot realizar el movimiento o trayectoria que manda. Tras realizar la acción mandada responde a Matlab como ha ido el proceso, Figura 9.

## CAPITULO 3. DESARROLLO DEL PROYECTO

En este capítulo se describe detalladamente el contenido de las partes que forman el proyecto. Estas partes son el diseño 3D y fabricación de los objetos, las herramientas y la estructura de la estación, la programación en Matlab del interfaz de usuario, las rutinas y las trayectorias del robot y la comunicación con el robot y la programación en lenguaje propio del robot de esta comunicación.

### 3.1. Diseño 3D con Catia

Con la herramienta de diseño CAD Catia se van a diseñar tanto los componentes individuales que formaran parte de la estación como el resultado final de la estación, garantizando que todos los componentes están bien diseñados. Además, estos componentes se pueden exportar tanto para poder ser fabricados empleando impresión 3D como para formar parte de las estaciones simuladas en Simulink y RobotStudio.

#### 3.1.1. Semiesfera

El primer elemento diseñado es una semiesfera, Figura 11. Posee 17 orificios, ubicados a 0, 30 y 60 grados respecto de la vertical. La intención de estos orificios es la de permitir a robot, con diferentes orientaciones, incidir dentro de la semiesfera.



Figura 11 Semiesfera

Como las dimensiones de la semiesfera exceden el volumen de impresión de la impresora 3D que se va a emplear, se divide, Figura 12, en 4 secciones iguales y se unirán posteriormente con imanes, [9]. El resultado de la fabricación de este objeto se puede comprobar en el apartado Objetos fabricados y las dimensiones se pueden encontrar en los anexos del proyecto.



Figura 12 Semiesfera cortada

### 3.1.2. Cilindro

El segundo elemento diseñado es un cilindro, Figura 13. Posee 60 orificios laterales, ubicados a diferentes alturas, y 26 orificios entre las bases, ubicados formando un hexágono. La intención de estos orificios es la de permitir a robot, con diferentes orientaciones incidir dentro del cilindro.

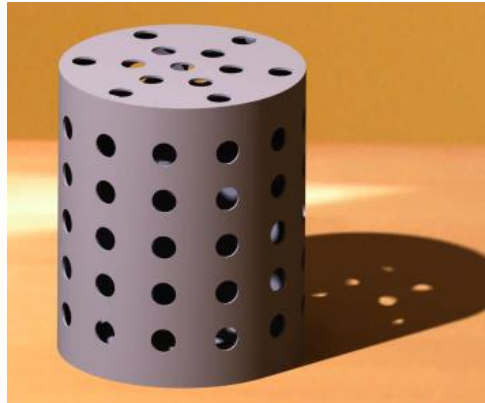


Figura 13 Cilindro

Como las dimensiones del cilindro exceden el volumen de impresión de la impresora 3D que va a emplear, se divide, Figura 14, en 8 secciones iguales y se unirán posteriormente con imanes, [9]. El resultado de la fabricación de este objeto se puede comprobar en el apartado Objetos fabricados y las dimensiones se pueden encontrar en los anexos del proyecto.

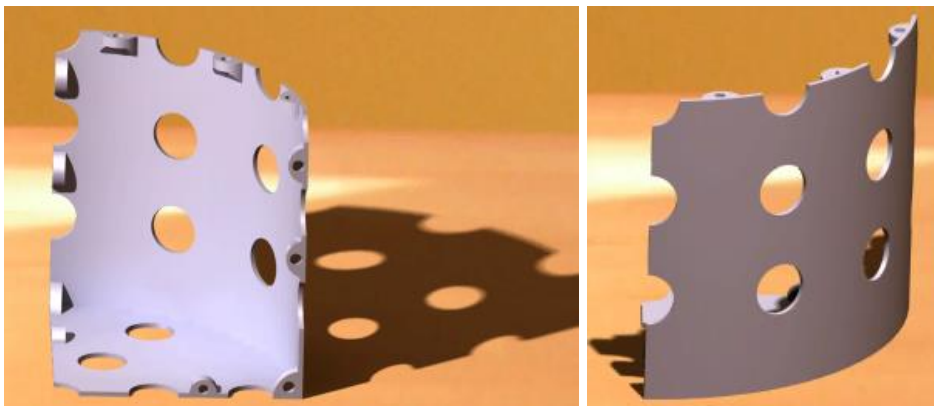


Figura 14 Cilindro cortado



### 3.1.3. Soporte de herramientas y herramientas

Para que el robot interactúe con estos objetos diseñados, se han dimensionado unas herramientas, Figura 15, de diferentes longitudes para que el robot tenga distintos alcances. El resultado de la fabricación de este objeto se puede comprobar en el apartado Objetos fabricados.

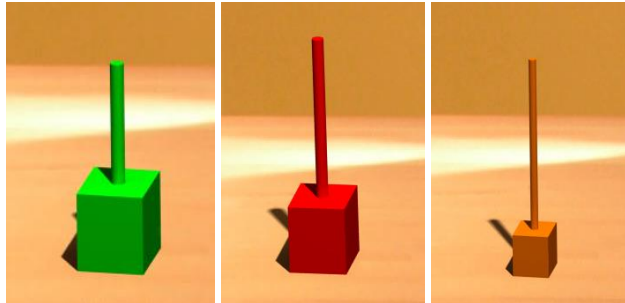


Figura 15 Herramientas diseñadas para el robot de 15,20 y 30cm

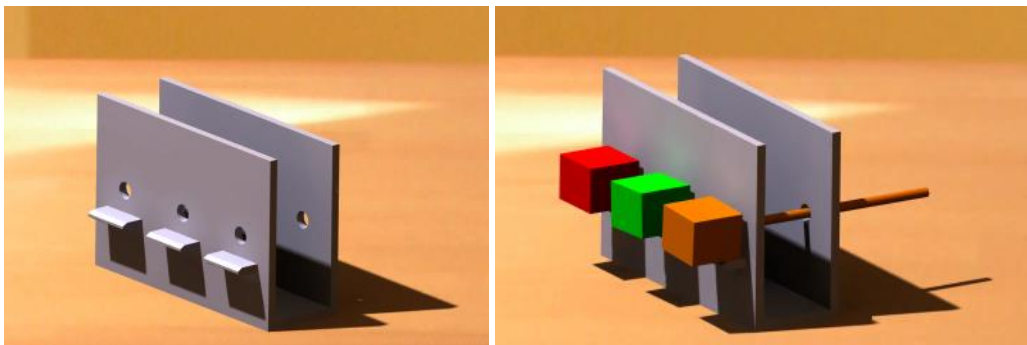


Figura 16 Soporte de herramientas

Como las dimensiones del soporte exceden el volumen de impresión de la impresora 3D que se va a emplear, se divide, Figura 17, en 2 partes iguales y se pegarán juntas. El resultado de la fabricación de este objeto se puede comprobar en el apartado Objetos fabricados y las dimensiones se pueden encontrar en los anexos del proyecto.

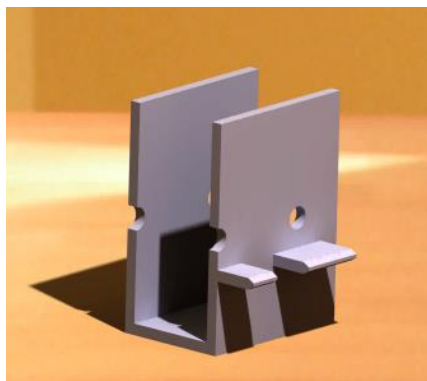


Figura 17 Soporte de herramientas cortado

### 3.1.4. Mesa de trabajo

Uno de los lugares de trabajo del robot será la mesa sobre la que se encuentra el robot. La semiesfera y el cilindro se encuentran sobre la superficie de la mesa. El centro de la semiesfera se encuentra a 350mm sobre el eje Y del sistema de coordenadas de la base del robot y el centro del cilindro se encuentra a 300mm sobre el eje Y, apreciable en la Figura 18, y a 200mm sobre el eje X, apreciable en la Figura 19, del sistema de coordenadas de la base del robot.

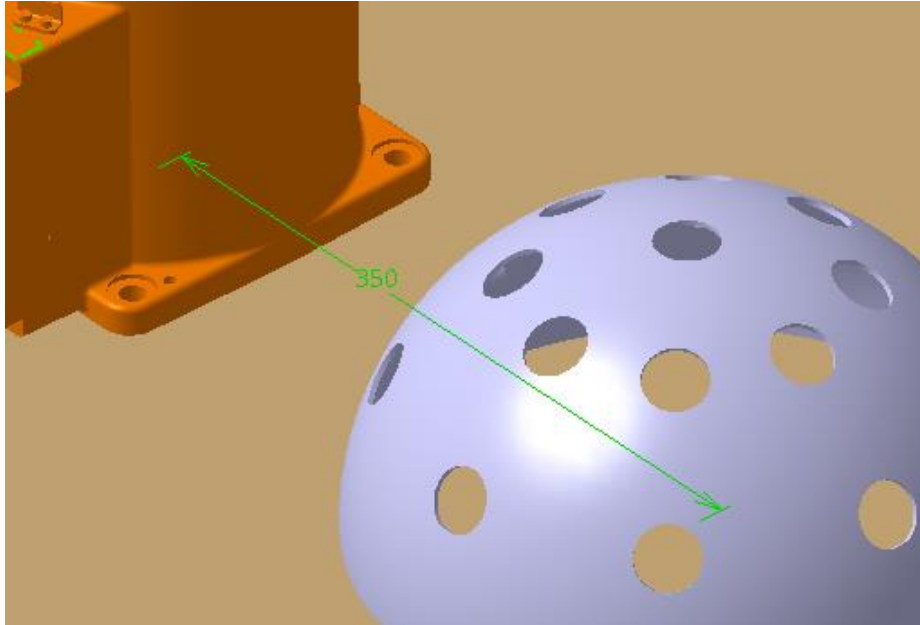


Figura 18 Posición de la semiesfera sobre la mesa

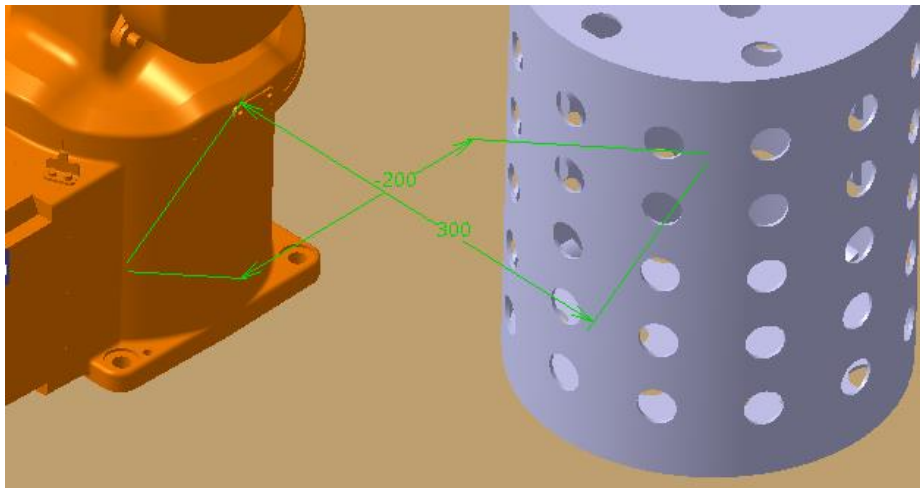


Figura 19 Posición del cilindro sobre la mesa

### 3.1.5. Estructura lateral

Otro lugar de trabajo del robot será el lateral izquierdo, si vemos el robot de frente. La estructura consta de unos anclajes a la mesa, unos listones de madera, un adaptador para sostener tanto la esfera como el cilindro y los objetos de trabajo, la semiesfera y el cilindro.

Los objetos deberán colocarse en los orificios del adaptador para que el robot pueda realizar las trayectorias correctamente. El punto central de cada objeto se encuentra a 350mm sobre el nivel de la mesa y están alejados 750mm en Y del sistema de coordenadas “mundo”, ubicado en la base del robot, apreciable en las Figura 20 y Figura 21.

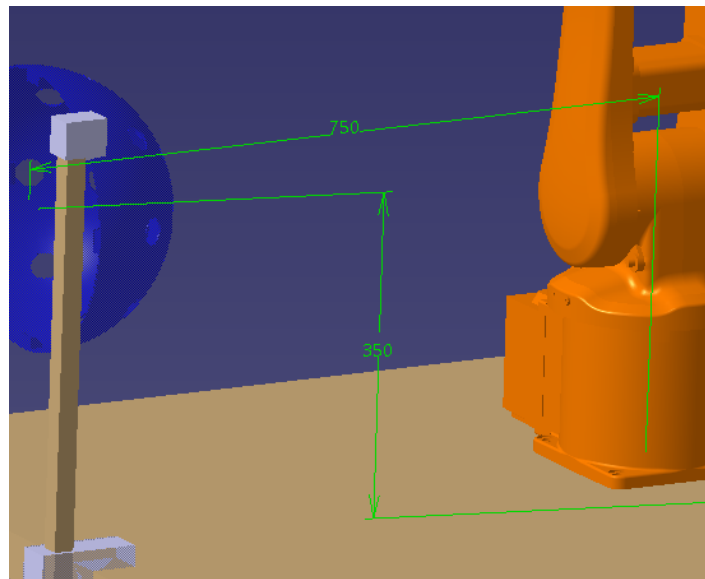


Figura 20 Posición de la semiesfera en la estructura lateral

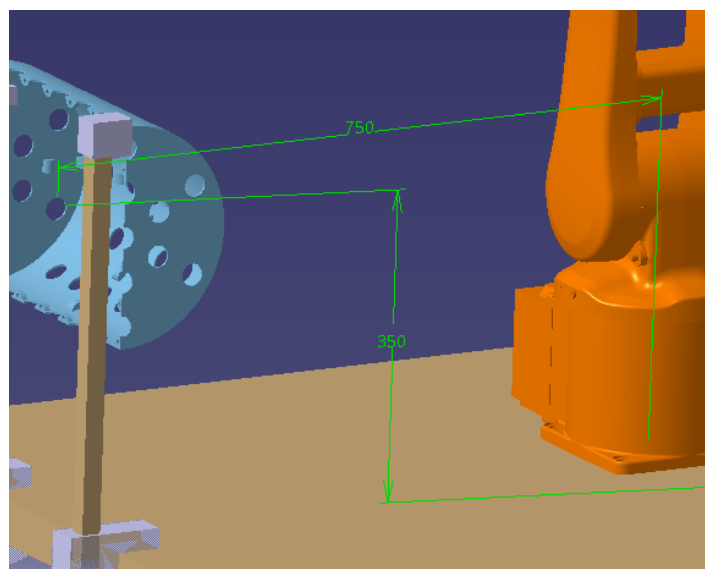
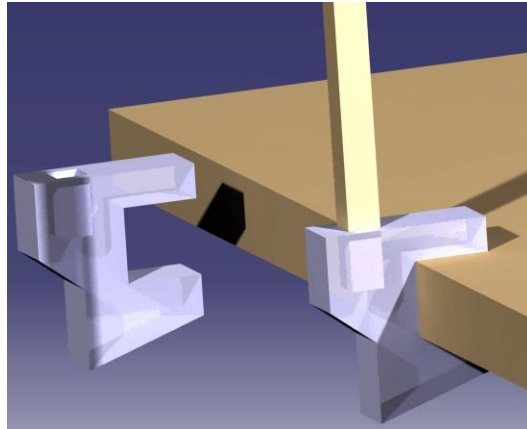


Figura 21 Posición del cilindro sobre la estructura lateral

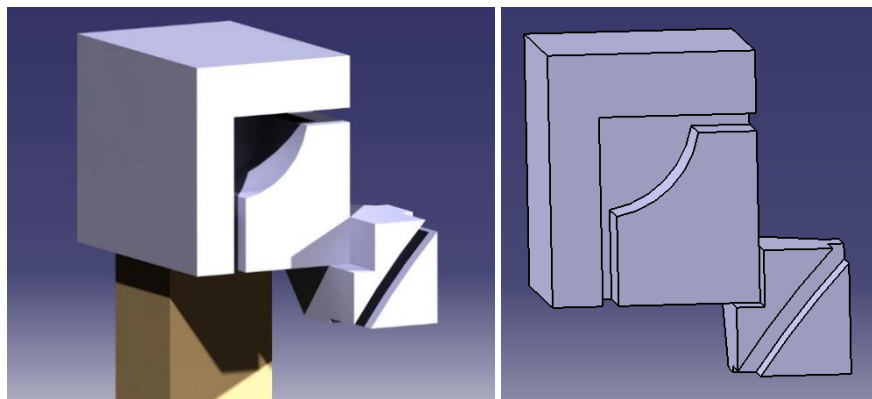
Los anclajes a la mesa, Figura 22, tienen el espesor de la mesa para poder encajar y dejar firme la estructura. Además, se podrá retirar cuando no se necesite o colocar en otra posición cuando así que se desee, haciendo más cómodo el montaje de la estructura. El resultado de la fabricación de este objeto se puede comprobar en el apartado Objetos fabricados.



*Figura 22 Anclajes a la mesa de la estructura lateral*

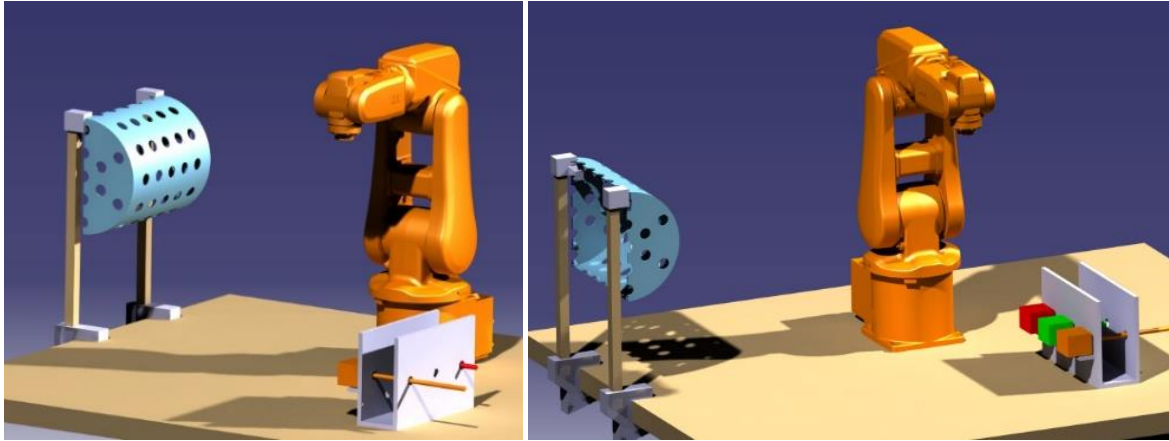
Los listones de madera van encajados en los anclajes. Sus dimensiones de 22x22x500mm, suficientes para soportar el peso de la estructura, [10].

En la parte superior de cada listón, encaja el soporte de los objetos de trabajo, Figura 23. Una franja está destinada a la colocación de la mitad del cilindro, y otra está destinada a la colocación de la semiesfera. El resultado de la fabricación de este objeto se puede comprobar en el apartado Objetos fabricados.

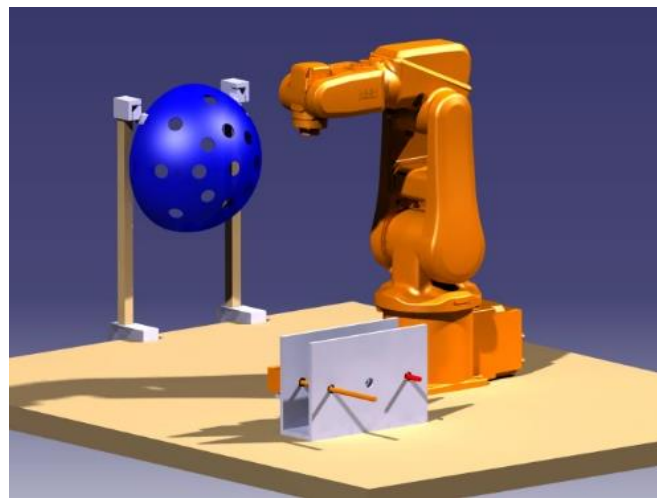


*Figura 23 Ranuras de los soportes de los objetos*

La estructura lateral completa, con los anclajes de la mesa, los listones y los soportes de los objetos, se muestra en la Figura 24 y Figura 25.



*Figura 24 Cilindro en estructura lateral*

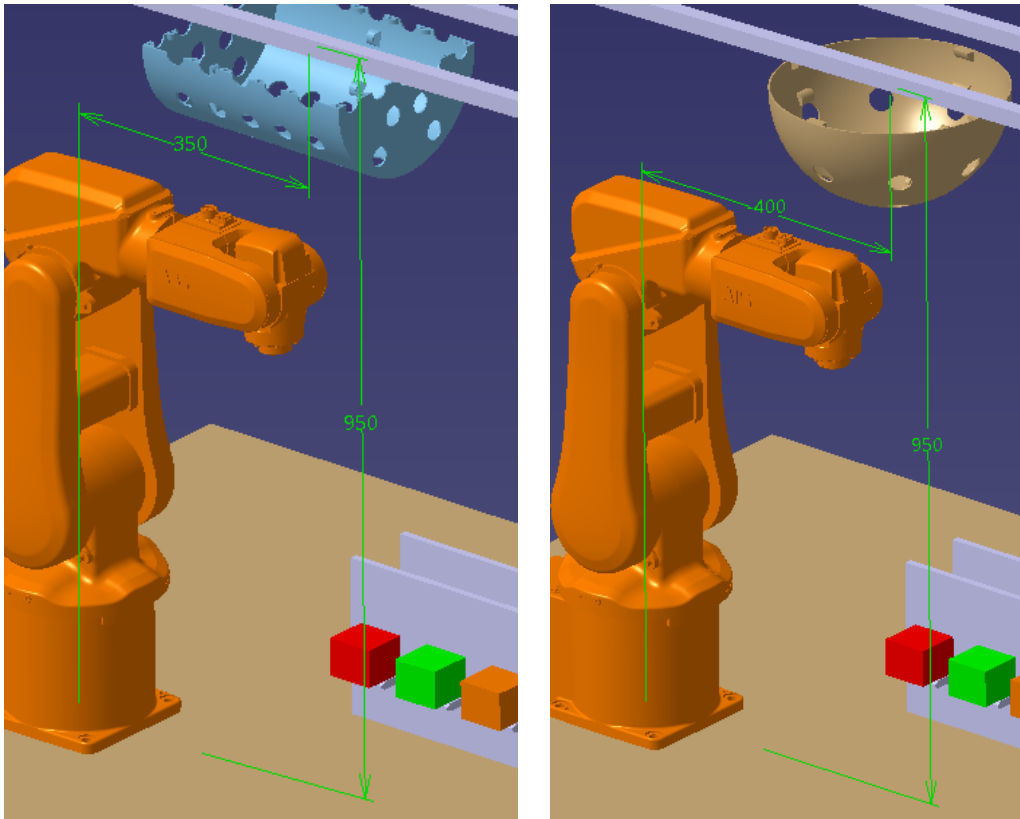


*Figura 25 Semisfera en estructura lateral*

### 3.1.6. Estructura superior

El último lugar de trabajo del robot será el techo del robot. La estructura consta de unos listones de madera, los objetos de trabajo, la semiesfera y el cilindro, y un gancho para sujetarlos.

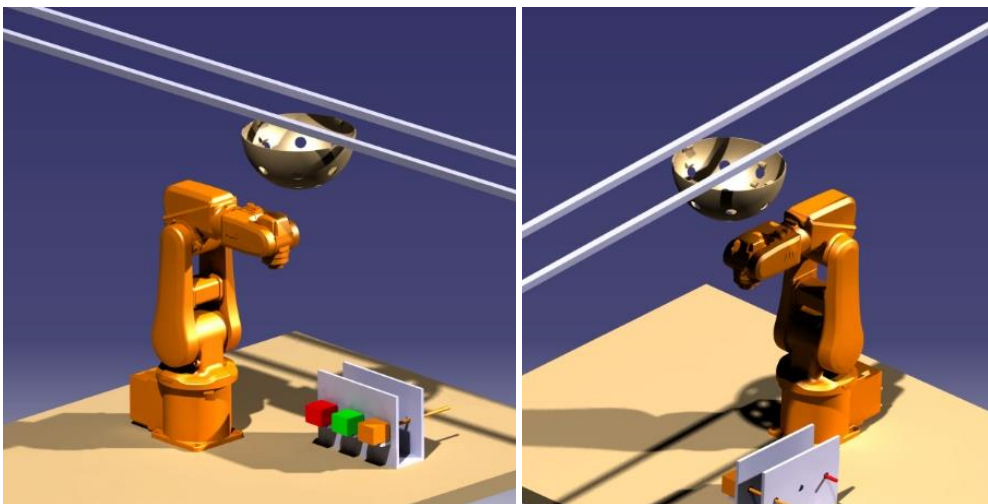
Los objetos deberán suspenderse por los orificios con los ganchos que se encuentran al final de los hilos colgantes de los listones para que el robot pueda realizar las trayectorias correctamente. El punto central de ambos objetos deberá estar a 950mm sobre el nivel de la mesa, y alejados 350mm y 400mm en X del sistema de coordenadas “mundo” en la base del robot, en el caso del cilindro y la semiesfera respectivamente, Figura 26.



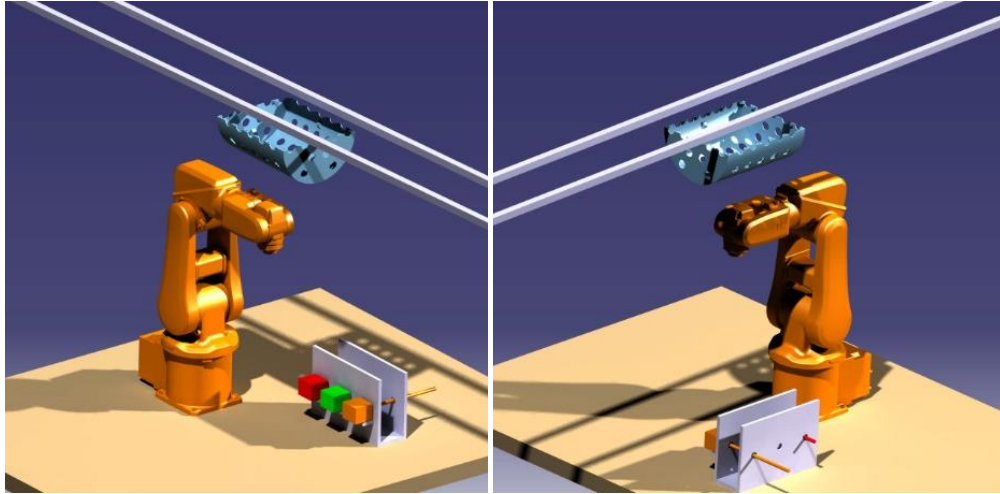
*Figura 26 Posición de la semiesfera y el cilindro en la estructura superior*

Dos listones se apoyarán sobre el vallado de seguridad que posee la instalación del robot y de ahí se suspenderán los objetos de trabajo.

La estructura superior completa con los listones, los objetos y ganchos, se puede apreciar en la Figura 27 y Figura 28.



*Figura 27 Semiesfera suspendida en la estructura superior*

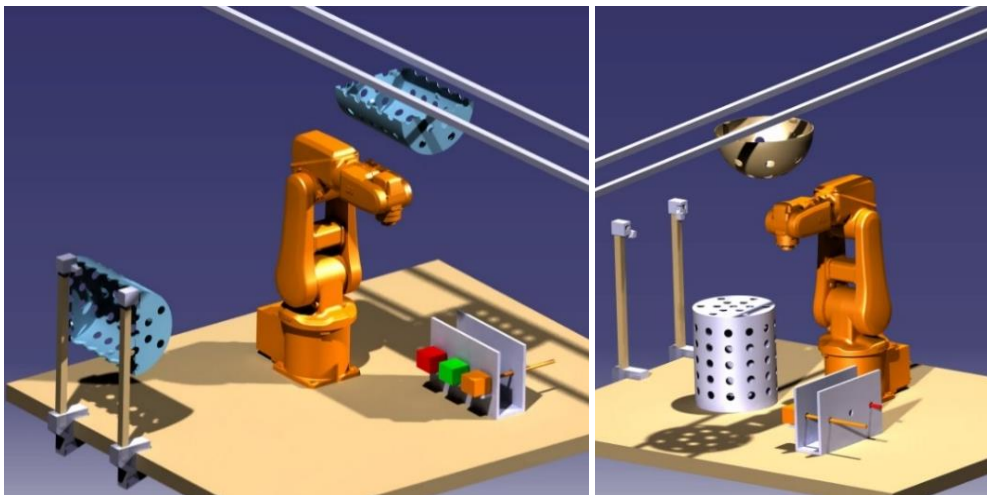


*Figura 28 Cilindro suspendido en la estructura superior*

### 3.1.7. Estación completa

La estación completa dota a la instalación ya existente del robot de unos recursos adicionales que permitirán al docente realizar operaciones en diferentes lugares de trabajo, menos convencionales que la mesa de trabajo y comprobar la capacidad de realizar movimientos de un robot industrial.

Se pueden plantear realizaciones de prácticas en las que se deba trabajar en varios ámbitos a la vez, Figura 29, debido a la flexibilidad del diseño de los objetos. Se podrán hacer prácticas operando sobre las mitades del cilindro en la estructura lateral y superior simultáneamente, o trabajar sobre la esfera en un lugar y el cilindro en otro.



*Figura 29 Posibles prácticas a realizar en los entornos y objetos de trabajo*

## 3.2. Programación con Matlab

La programación en Matlab se divide en varias partes:

- Código de Matlab donde se programarán las trayectorias a realizar por el robot.
- Desarrollo, diseño y programación del interfaz de usuario,
- Diseño del robot ABB en Simulink empleando la librería Multibody para poder simularlo.

### 3.2.1. Código de Matlab con trayectorias

El código programado de Matlab permite mover tanto el robot real como el robot simulado en Simulink a través de las clases Robot y Robot\_Simulacion, Herramienta, tcpABB y Kin, véase anexos del proyecto.

#### 3.2.1.1. Clases programadas

##### 3.2.1.1.1. Robot

La clase Robot permite controlar el robot real ABB-IRB-120 y el de simulación en RobotStudio a través de Matlab. Esta clase se comunica empleando la clase tcpABB con el código RAPID del robot real.

Posee métodos que le permiten seleccionar la base de trabajo actual, la herramienta con la que trabajar y realizar movimientos lineales, articulares y absolutos del robot. También posee las rutinas con las trayectorias programadas para los objetos y lugares de trabajo. La clase Robot trabaja en milímetros.

##### 3.2.1.1.2. Clase Robot\_Simulacion

La clase Robot permite controlar el robot simulado ABB-IRB-120 en Simulink. Al pertenecer al entorno Matlab, no es necesario establecer ninguna conexión, únicamente cargar el modelo e introducir sus variables.

Posee métodos que le permiten mostrar el objeto de trabajo y la herramienta actual, realizar movimientos lineales, articulares y absolutos del robot. También posee las rutinas con las trayectorias programadas para los diferentes objetos y lugares de trabajo. La clase Robot\_Simulacion trabaja en metros.

##### 3.2.1.1.3. Clase Herramienta

La clase Herramienta modela las barras de diferentes longitudes que puede emplear el robot. Está definida por la longitud de entrada a la esfera y al cilindro, los puntos en los que se encuentra la herramienta almacenada y las unidades de medida.

##### 3.2.1.1.4. Clase tcpABB

La clase tcpABB permite la comunicación entre el robot real y la clase Robot. Recibe la información que le pasa la clase Robot, la traduce a lenguaje RAPID y la envía a través de TCP/IP al robot real.

##### 3.2.1.1.5. Clase Kin

La clase Kin permite la comunicación entre el robot simulado en Simulink y la clase Robot\_Simulacion. Recibe la información que le pasa la clase Robot\_Simulacion, realiza la cinemática directa y la envía al robot simulado.



### 3.2.1.2. Trayectorias programadas

Las trayectorias programadas tienen como intención hacer que el robot adquiera diferentes orientaciones para incidir con la herramienta en los objetos de trabajo. Para ello se han buscado trayectorias repetitivas que pueden simular la colocación de tornillería o la soldadura por puntos en entornos reales de trabajo.

Para cada objeto de trabajo se han programado dos trayectorias en cada lugar de trabajo. No todas las herramientas pueden realizar todas las trayectorias debido a las limitaciones físicas del robot, Tabla 1.

Tabla 1 Lugares de trabajo y herramientas validas

	Mesa	Lateral derecho	Techo
Semiesfera	Verde, Roja, Naranja	Verde, Roja	Verde, Roja
Cilindro	Verde, Roja	Verde, Roja	Verde, Roja

A continuación, se explican cada una de las trayectorias programadas sobre el objeto y el lugar de trabajo seleccionados.

#### Trayectoria Semiesfera Mesa

La semiesfera sobre la mesa está ubicada a -350 mm sobre el eje Y del sistema de coordenadas mundo que se encuentra en la base del robot.

Las trayectorias programadas en este lugar de trabajo hacen incidir al robot en los orificios que posee la esfera de dos formas.

Una de las trayectorias, denominada “Sencilla”, hace que el robot incida en los orificios de la esfera a lo largo de su eje X, flecha roja en Figura 30, orientando la herramienta para que penetre perpendicularmente al orificio. El orden seguido, si viéramos el objeto desde un punto de vista cenital, se puede observar en la Figura 30.

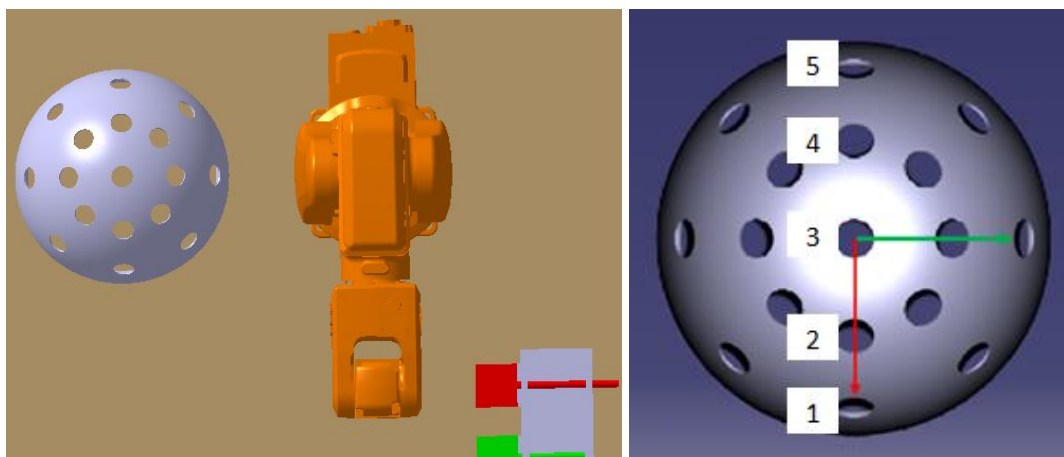


Figura 30 Semiesfera en mesa: vista y orden de los orificios en trayectoria “Sencilla”

La otra trayectoria, denominada “Rotando”, hace que el robot incida en los orificios de la esfera a lo largo de su eje X, flecha roja en Figura 31, y los ubicados a  $\pm 45^\circ$  sobre el eje Z, orientando la herramienta para que penetre perpendicularmente al orificio. El orden seguido, si viéramos el objeto desde un punto de vista cenital, se puede observar en la Figura 30.

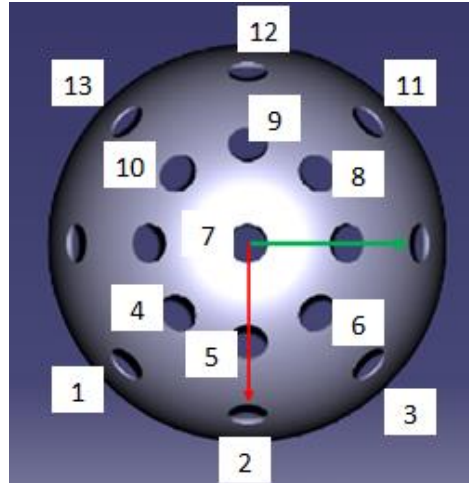


Figura 31 Semiesfera en mesa: orden de los orificios en trayectoria “Rotando”

#### Trayectoria Cilindro Mesa

El cilindro sobre la mesa está ubicado a 200mm sobre el eje X y a -300mm sobre el eje Y desde la base del robot.

Las trayectorias programadas en este lugar de trabajo hacen incidir al robot en los orificios que posee el cilindro de dos formas.

Una de las trayectorias, denominada “Sencilla”, hace que el robot incida en los orificios de la base superior del cilindro, orientando la herramienta para que penetre perpendicularmente a cada orificio. El orden seguido, si viéramos el objeto desde un punto de vista cenital, se puede observar en la Figura 32.

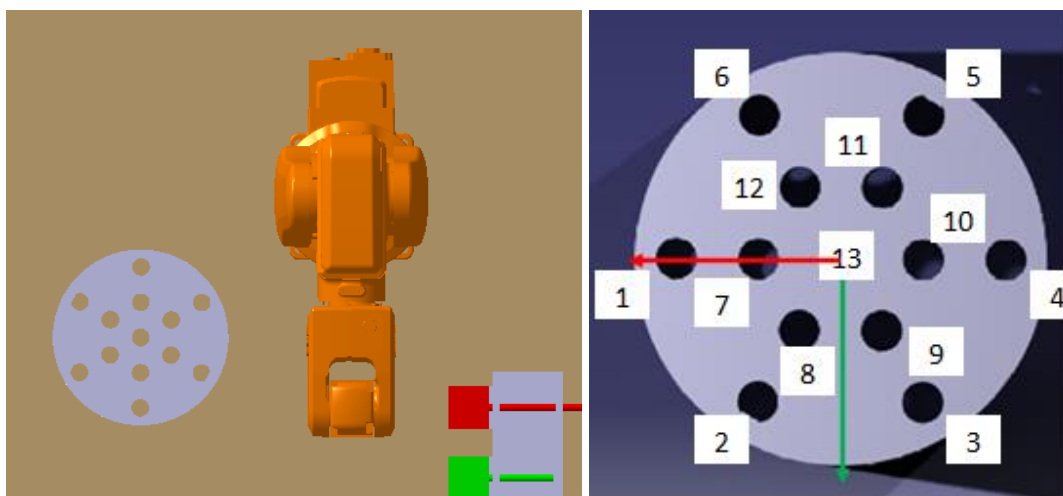


Figura 32 Cilindro en mesa: vista y orden de los orificios en la trayectoria “Sencilla”

La otra trayectoria, denominada “Rotando” hace que el robot incida en los orificios laterales a lo largo de su eje Z, flecha azul en Figura 33, orientando la herramienta para que penetre perpendicularmente al orificio. El orden seguido, si viéramos el objeto desde un punto de vista trasero, Figura 33.

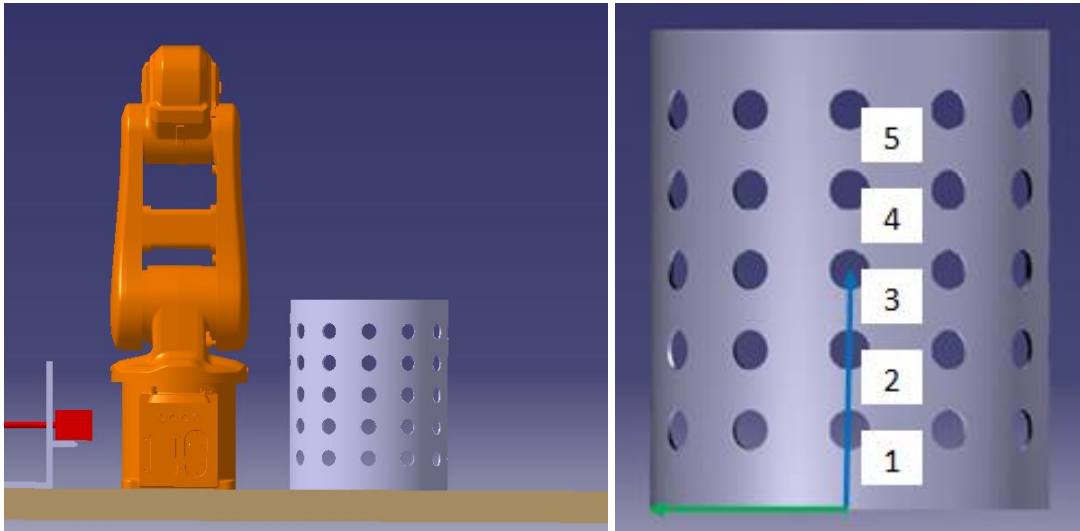


Figura 33 Cilindro en mesa: vista y orden de los orificios en la trayectoria “Rotando”

#### Trayectoria Semiesfera Lateral Derecho

La semiesfera se encuentra a 350mm sobre el eje Z y a -750mm sobre el eje Y del origen de coordenadas de la base del robot.

Las trayectorias programadas en este lugar de trabajo hacen incidir al robot en los orificios que posee la esfera de dos formas.

Una de las trayectorias, denominada “Sencilla”, hace que el robot incida en los orificios a lo largo del eje X, flecha roja en Figura 34, y posteriormente del eje Y, flecha verde en Figura 34, orientando la herramienta para que penetre perpendicularmente a cada orificio. El orden seguido, si viéramos el objeto de frente, se puede observar en la Figura 34.

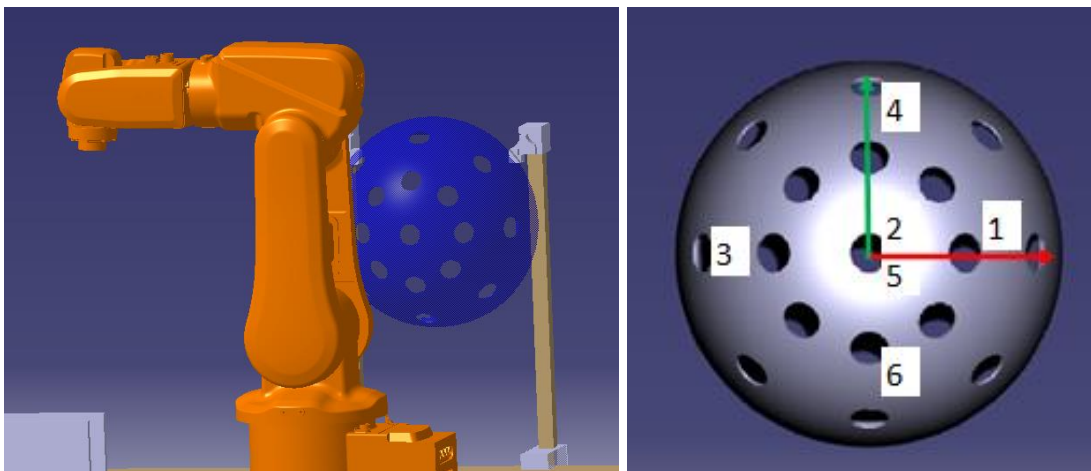


Figura 34 Semiesfera en lateral derecho: vista,orden de orificios en trayectoria Sencilla

La trayectoria, denominada “Rotando” hace que el robot incida en los orificios de la esfera a lo largo de su eje X, flecha roja en Figura 35, y los ubicados a  $\pm 45^\circ$  sobre el eje Z, orientando la herramienta para que penetre perpendicularmente al orificio. El orden seguido, si viéramos el objeto de frente, se puede observar en la Figura 34.

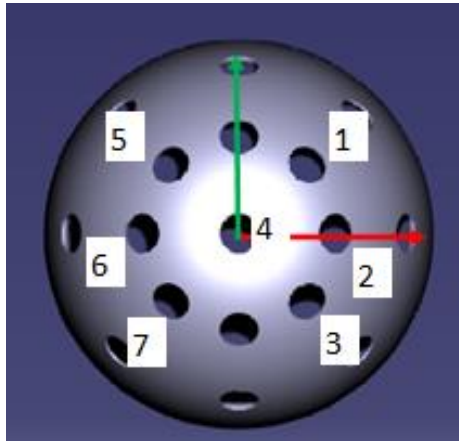


Figura 35 Semiesfera en lateral derecho: orden de orificios en trayectoria “Rotando”

#### Trayectoria Cilindro Lateral Derecho

El cilindro se encuentra a 350mm sobre el eje Z y a -750mm sobre el eje Y del origen de coordenadas de la base del robot.

Las trayectorias programadas en este lugar de trabajo hacen incidir al robot en los orificios que posee la esfera de dos formas.

Una de las trayectorias, denominada “Sencilla”, hace que el robot incida en los orificios laterales a lo largo de su eje X, flecha roja en Figura 36, orientando la herramienta para que penetre perpendicularmente a el orificio. El orden seguido, si viéramos el objeto de frente, se puede observar en la Figura 36.

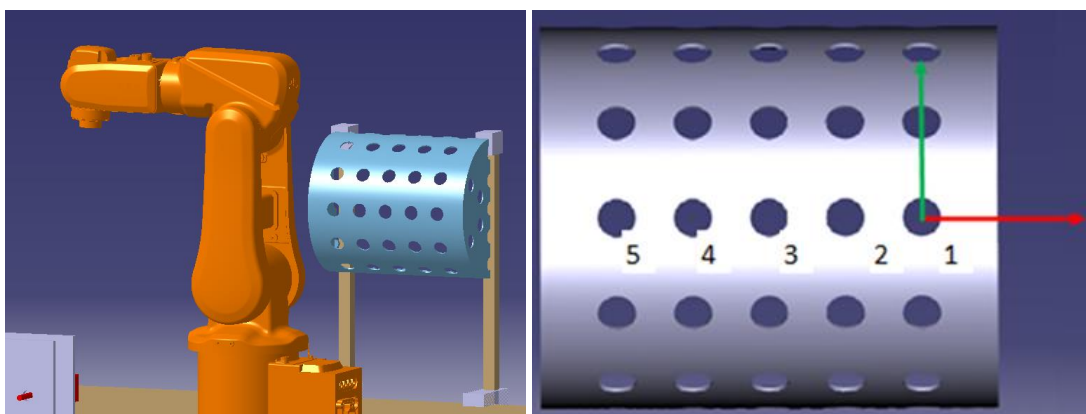


Figura 36 Cilindro en lateral derecho: vista y orden de orificios en trayectoria “Sencilla”

La otra trayectoria, denominada “Rotando”, hace que el robot incida en los orificios laterales a lo largo de su eje Z, flecha azul en Figura 37, orientando la herramienta para que penetre perpendicularmente al orificio. El orden seguido, si viéramos el objeto de frente, se puede observar en la Figura 36.

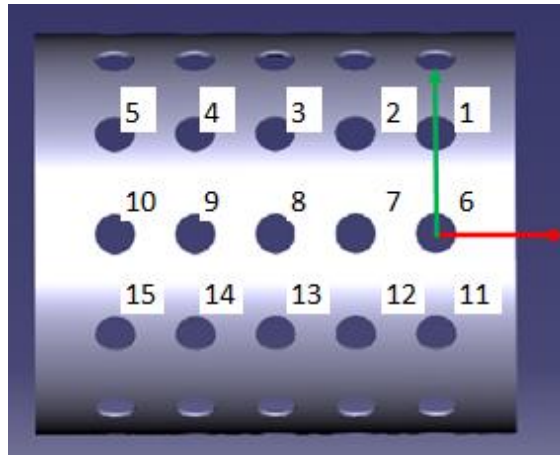


Figura 37 Cilindro en lateral derecho: orden de los orificios en la trayectoria "Rotando"

### Trayectoria Semiesfera Techo

La semiesfera se encuentra a 950mm sobre el eje Z y a 400mm sobre el eje X del origen de coordenadas de la base del robot.

Las trayectorias programadas en este lugar de trabajo hacen incidir al robot en los orificios que posee la esfera de dos formas.

Una de las trayectorias, denominada "Sencilla", hace que el robot incida en los orificios a lo largo del eje X, flecha roja en Figura 38, y posteriormente del eje Y, flecha verde en Figura 38, orientando la herramienta para que penetre perpendicularmente a cada orificio. El orden seguido, si viéramos el objeto desde la parte inferior, se puede observar en la Figura 38.

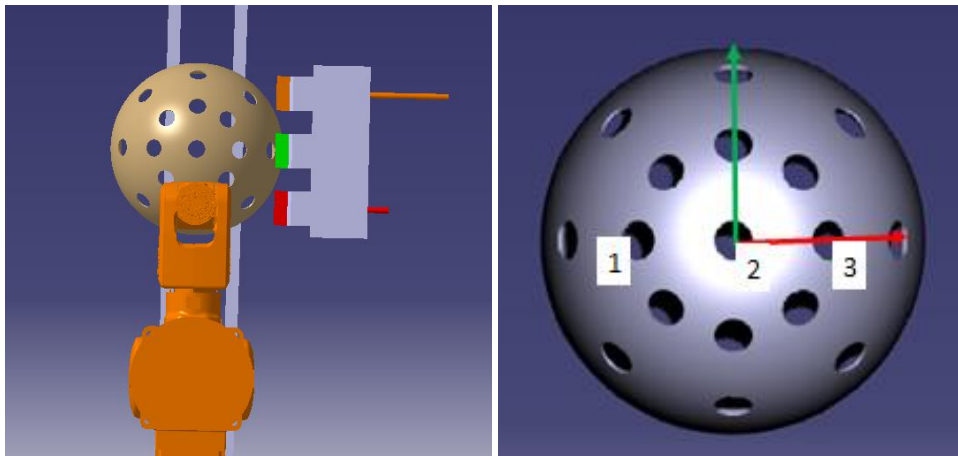


Figura 38 Semiesfera en techo: vista y orden de orificios en trayectoria "Sencilla"

La otra trayectoria, denominada "Rotando", hace que el robot incida en los orificios de la esfera a lo largo de su eje X, flecha roja en Figura 39, y los ubicados a  $\pm 45^\circ$  sobre el eje Z, orientando la herramienta para que penetre perpendicularmente al orificio. El orden seguido, si viéramos el objeto desde la parte inferior, se puede observar en la Figura 38.

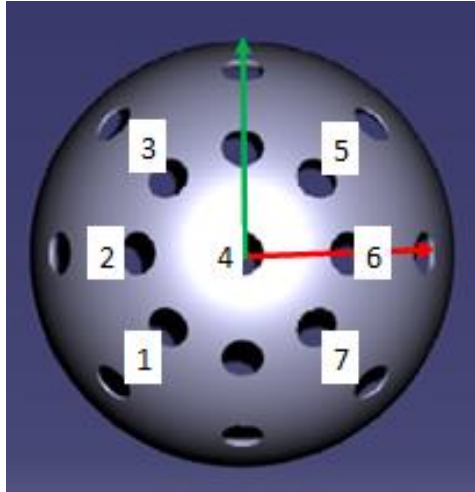


Figura 39 Semiesfera en techo: orden de los orificios en la trayectoria "Rotando"

#### Trayectoria Cilindro Techo

El cilindro se encuentra a 950mm sobre el eje Z y a 200mm sobre el eje X del origen de coordenadas de la base del robot.

Las trayectorias programadas en este lugar de trabajo hacen incidir al robot en los orificios que posee la esfera de dos formas.

Una de las trayectorias, denominada "Sencilla", hace que el robot incida en los orificios laterales a lo largo de su eje Y, flecha verde en Figura 40, orientando la herramienta para que penetre perpendicularmente a el orificio. El orden seguido, si viéramos el objeto desde la parte inferior, se puede observar en la Figura 40.

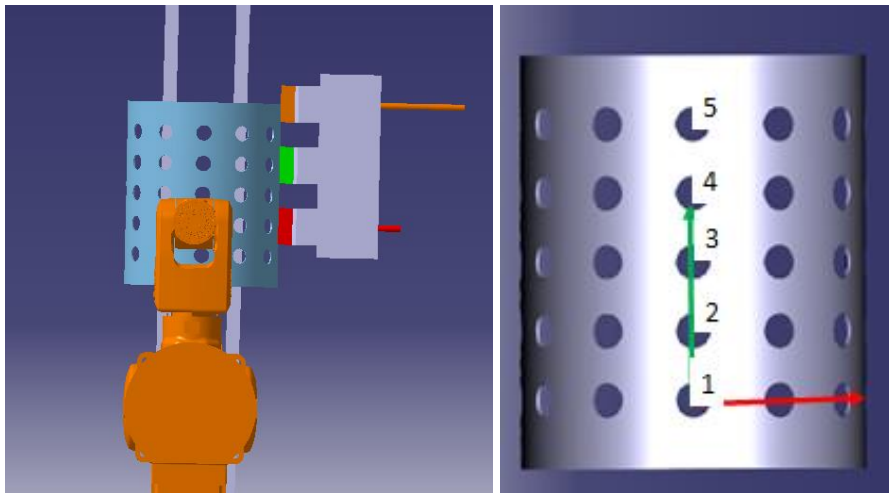


Figura 40 Cilindro en techo: vista y orden de orificios en trayectoria "Sencilla"

La otra trayectoria, denominada "Rotando", hace que el robot incida en los orificios laterales a lo largo de su eje Y, flecha verde en Figura 41, orientando la herramienta para que penetre perpendicularmente al orificio. El orden seguido, si viéramos el objeto desde la parte inferior, se puede observar en la Figura 40.

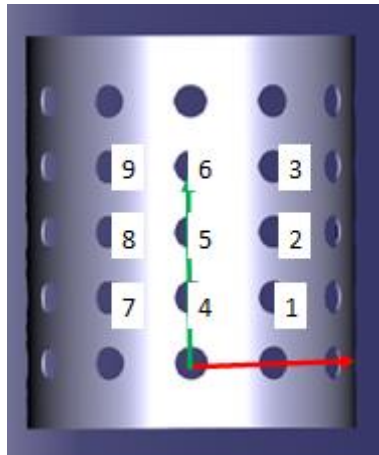


Figura 41 Cilindro en techo: orden de los orificios en la trayectoria "Rotando"

### 3.2.2. Simulink

La aplicación Simulink de Matlab junto con las librerías Multibody y Robotics System Toolbox permite modelar y realizar simulaciones de robots industriales.

Para modelar el robot el robot ABB-IRB-120, como tiene 6 grados de libertad, se necesitan 6 articulaciones de rotación, Figura 42, que deben colocarse en el esquema de Simulink.

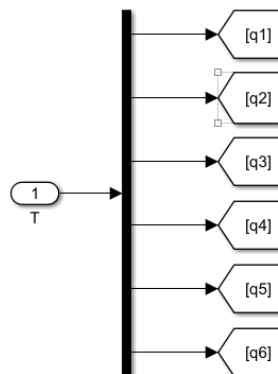


Figura 42 Parámetros de posición articular de entrada al modelo de Simulink

Comenzamos el robot por la base, donde definimos un sistema de referencia mundo, Figura 43, para que todo el robot este referenciado respecto a ese sistema, un bloque de configuración de mecanismo, que establece parámetros mecánicos y de simulación aplicados a la máquina de destino conectada con el bloque. En la sección Propiedades se puede especificar la gravedad,  $g$ , para todo el mecanismo y también establecer el delta de linealización y un bloque para resolver los ajustes de la simulación.

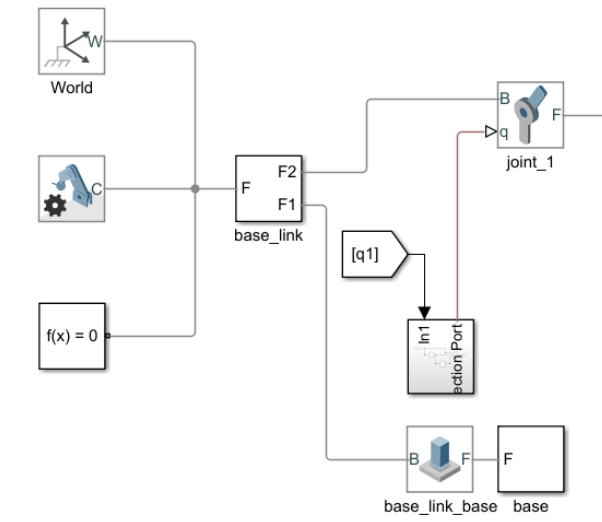


Figura 43 Modelado de la base del robot ABB-IRB-120

Estos bloques sirven de entrada a un subsistema “base\_link” en Figura 43 en el que se realiza el posicionamiento de la base del robot, la referenciación respecto al sistema de coordenadas mundo y un fichero .stl para la visualización en la simulación.

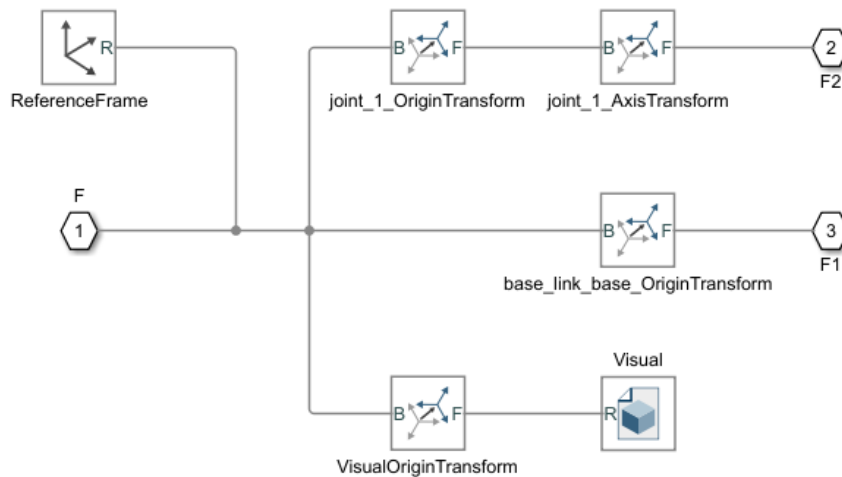


Figura 44 Contenido del subsistema de la base del robot ABB-IRB-120

Los parámetros introducidos en las transformaciones de sistemas de coordenadas se han dejado con los valores por defecto, es decir sin transformación, puesto que el origen de la base coincide con el origen del mundo en esta instalación robótica. También se asigna la ruta del fichero .stl de la base del robot, Figura 45. Este subsistema tiene dos salidas, Figura 43, una para el eje 1 del robot y otra para el entorno de la instalación, como pueden ser el soporte de las herramientas y los objetos de trabajo.



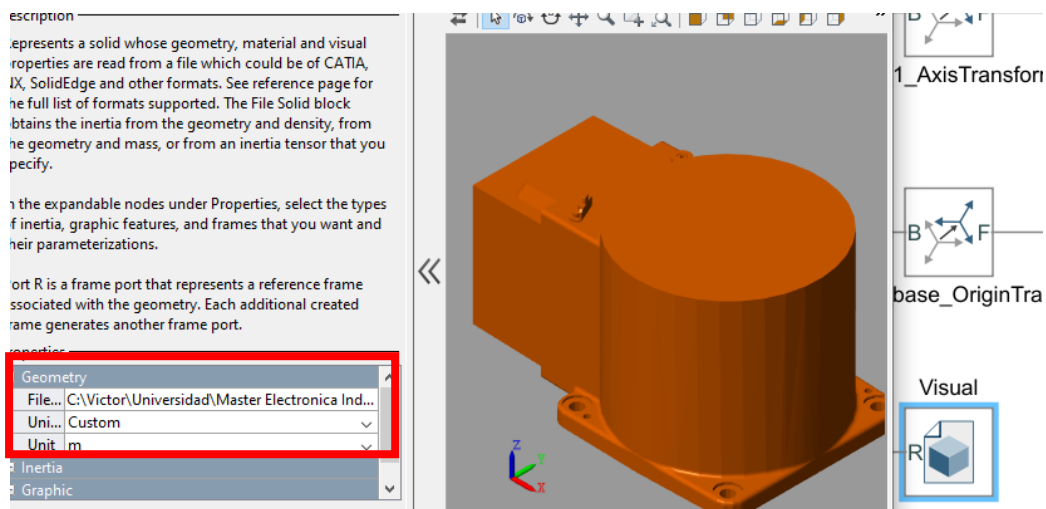


Figura 45 Asignación de fichero .stl a la base del robot ABB-IRB-120

Como se puede ver en la Figura 43, una salida va al bloque de la instalación, que ha sido modelado con un bloque de articulación sin grados de libertad, para únicamente servir de lazo al subsistema donde se introducirán los objetos constituyentes de la estación, Figura 46. Cada componente tiene asignado un fichero .stl, ver Diseño 3D con Catia, y una posición respecto al origen de la base, a excepción de primer elemento llamado “objeto”, que será una variable modificada a través de la aplicación según sea la semiesfera o el cilindro y el lugar de trabajo, mesa, lateral derecho o techo.

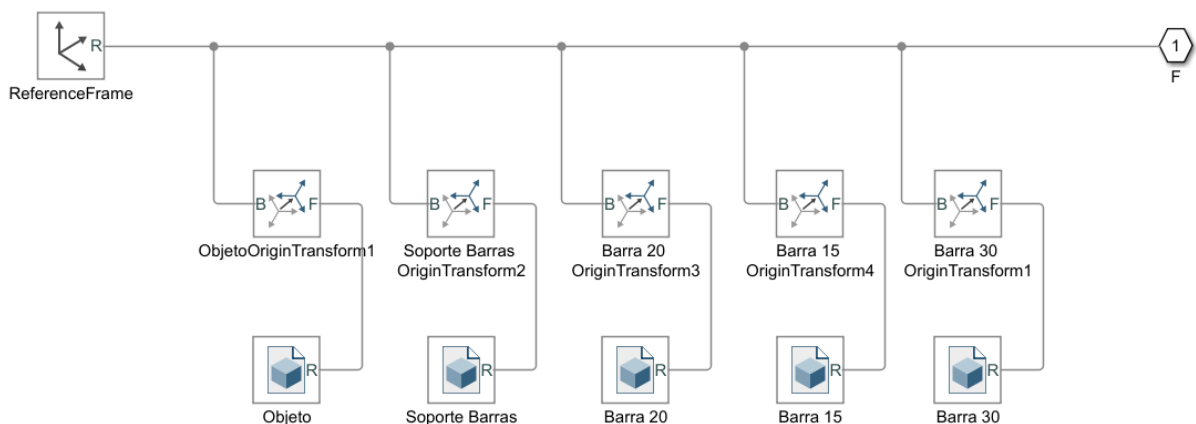


Figura 46 Contenido del subsistema de la estación

Como se puede ver en la Figura 43, una salida va a un bloque de la primera articulación del robot. En este bloque hay que modificar una de sus propiedades para que realice movimientos acordes a la entrada con el valor de la posición articular, Figura 47.

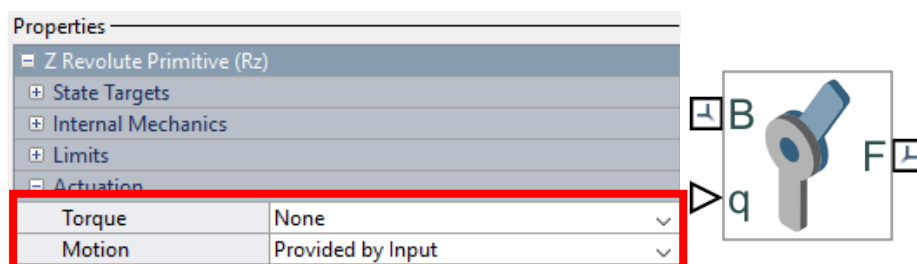


Figura 47 Propiedad a modificar del bloque Revolute Joint

Cada bloque tiene dos entradas y una salida. La entrada “q” es el valor de la articulación debe de alcanzar. La entrada “B” y la salida “F” son los sistemas de ejes de coordenadas de la sobre el que se posiciona la articulación y el sistema de ejes de coordenadas de la siguiente articulación respectivamente, siendo este último modificado en el subsistema creado entre articulaciones, Figura 48.

En ese subsistema debemos emplear los parámetros de Denavit Hartenberg para que el modelado del robot sea adecuado. Los parámetros de este modelo de robot, ABB-IRB-120, se adjuntan en la Tabla 2.

Tabla 2 Parámetros Denavit-Hartenberg del robot ABB-IRB-120

Eslabón	$\theta_i$	$d_i$ (mm)	$a_i$ (mm)	$\alpha_i$	Rango ( $\theta_i$ )
1	$\theta_1$	290	0	$-90^\circ$	$+165 -165^\circ$
2	$\theta_2-90^\circ$	0	270	$0^\circ$	$+110 -110^\circ$
3	$\theta_3$	0	70	$-90^\circ$	$+70 -110^\circ$
4	$\theta_4$	302	0	$90^\circ$	$+160 -160^\circ$
5	$\theta_5$	0	0	$-90^\circ$	$+120 -120^\circ$
6	$\theta_6+180^\circ$	72	0	$0^\circ$	$+400 -400^\circ$

Para ello, creamos un subsistema en el que vamos a introducir esos parámetros y además signaremos una masa y momentos de inercia a la articulación, [11], y un archivo. stl para la visualización en la simulación, Figura 49.

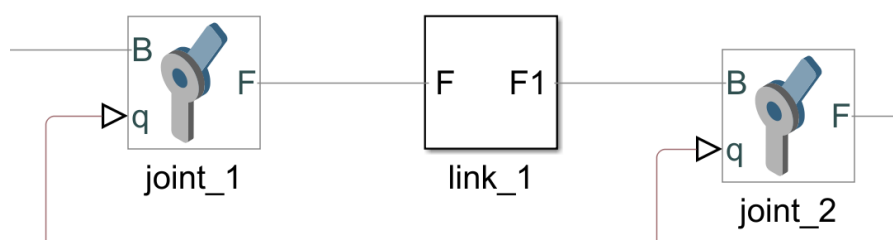


Figura 48 Unión entre articulaciones del robot

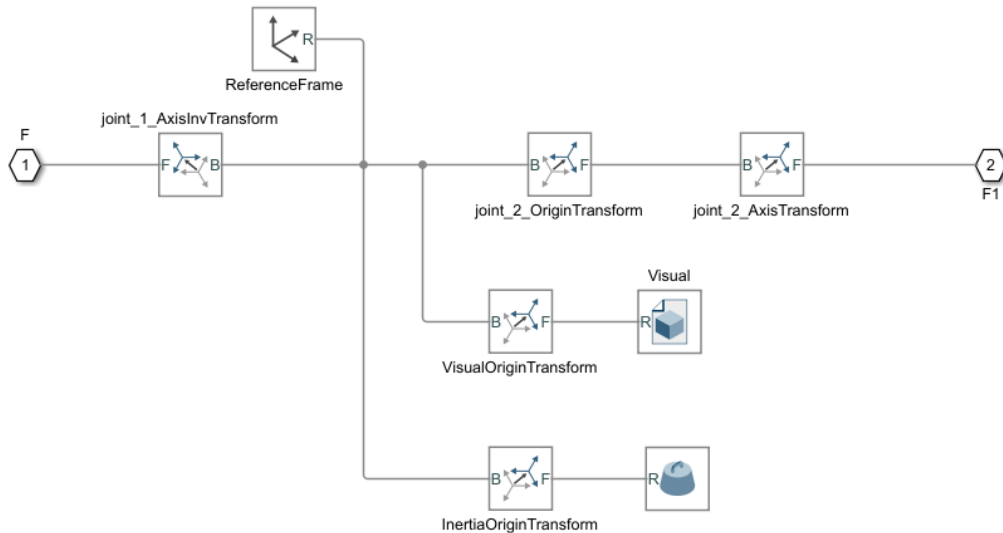


Figura 49 Contenido del subsistema entre articulaciones del robot

En el bloque de inercia introducimos los valores obtenidos de [11] en su campo correspondiente. Son datos empleados en modelado de robots en ROS, aunque no sean datos oficiales publicados por la propia empresa ABB.

Type	Custom	
Mass	3.06736850696	kg
Center of Ma...	[0, 0, 0]	m
Moments of ...	[0.0142194659396, 0.0104561541196, 0.0144054074894]	kg*m^2
Products of I...	[1.93410231314e-05, -1.08613978372e-05, 2.3107817...	kg*m^2

Figura 50 Valores de masa a inercias del robot ABB-IRB-120

En los dos bloques colocados antes de la salida del subsistema F1 de Figura 49, se introducen los parámetros de Denavit-Hartenberg “d” y “a” y “α” de Tabla 2, del eje 1, Figura 51.

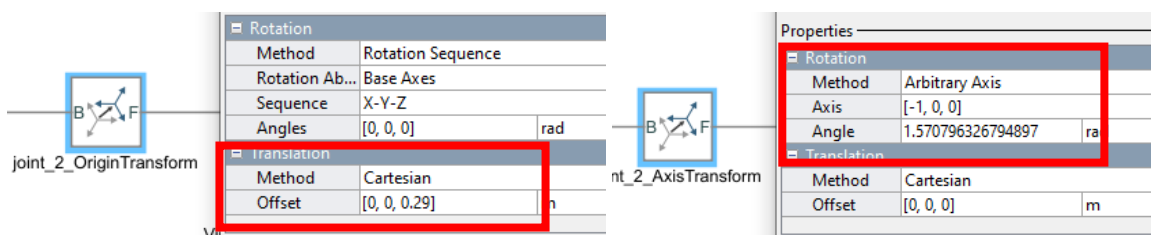


Figura 51 Parámetros Denavit Hartenberg del eje 1 del robot ABB-IRB-120

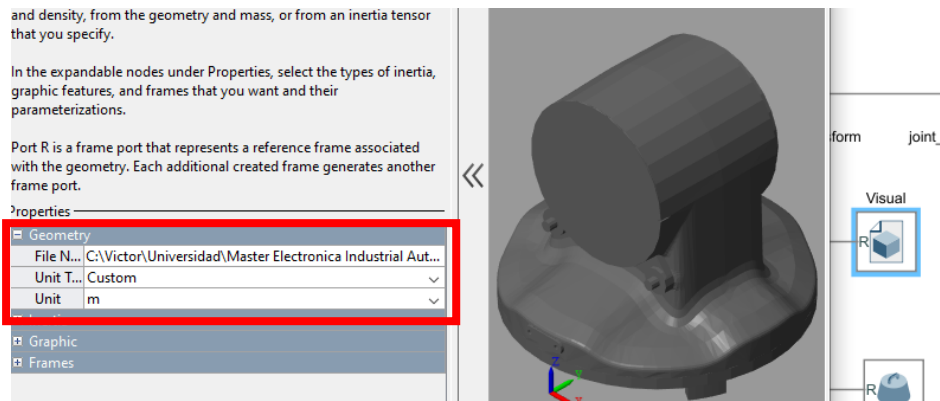


Figura 52 Asignación del fichero .stl para visualización en simulación

Este mismo proceso se debe repetir con todas las articulaciones del robot, desde el eje 1 hasta el eje 6.

La entrada de la posición articular de Figura 47 proviene de un subsistema, Figura 53, cuya entrada es la posición que recibe el modelo de Simulink. Esta posición se deriva para obtener tanto la velocidad como la aceleración articular y se introducen en un convertor de señal física, ya que estas señales no tienen unidades. Después se llevan al bloque de la articulación.

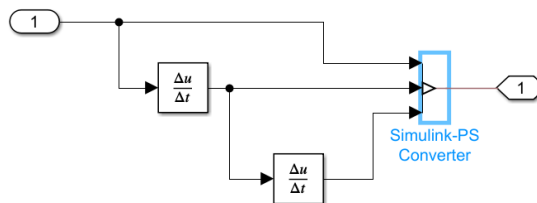


Figura 53 Contenido del subsistema de tratamiento de la entrada del modelo

Finalmente, las herramientas con las que trabaja el robot, pinza y barras de diferentes longitudes se representan como articulaciones sin grados de libertad, Figura 54.

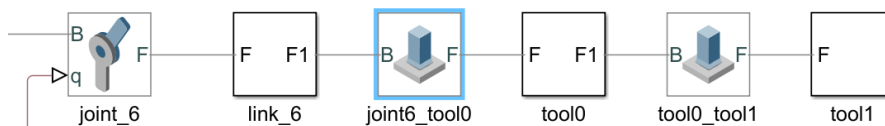


Figura 54 Unión entre el eje 6 y las herramientas: pinza (tool0) y barras (tool1)

Al igual que lo hemos hecho entre las articulaciones del robot, creamos un subsistema entre las herramientas del robot donde especificaremos la posición relativa de la herramienta y asignaremos un fichero .stl para visualizarla en la simulación, Figura 55.

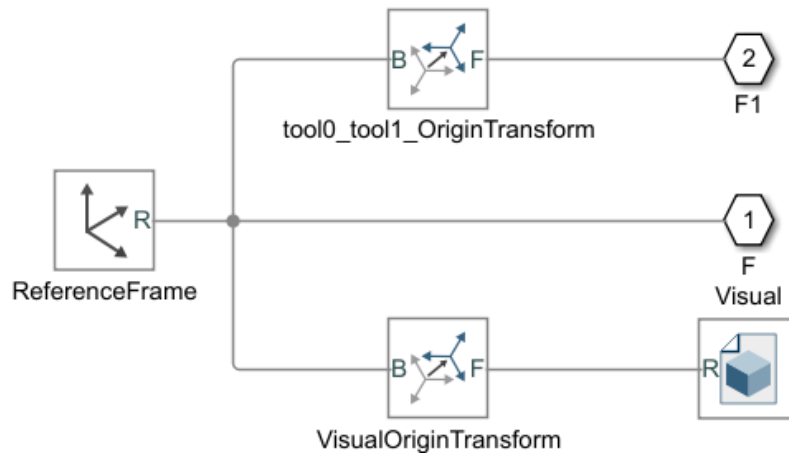


Figura 55 Contenido del subsistema de las herramientas

Simulando del modelo aparece una ventana con los elementos visuales que hemos asignado a cada articulación, herramienta y elementos de la estación, pero no realiza ningún movimiento porque no tiene entradas de posiciones articulares véase la Figura 56.

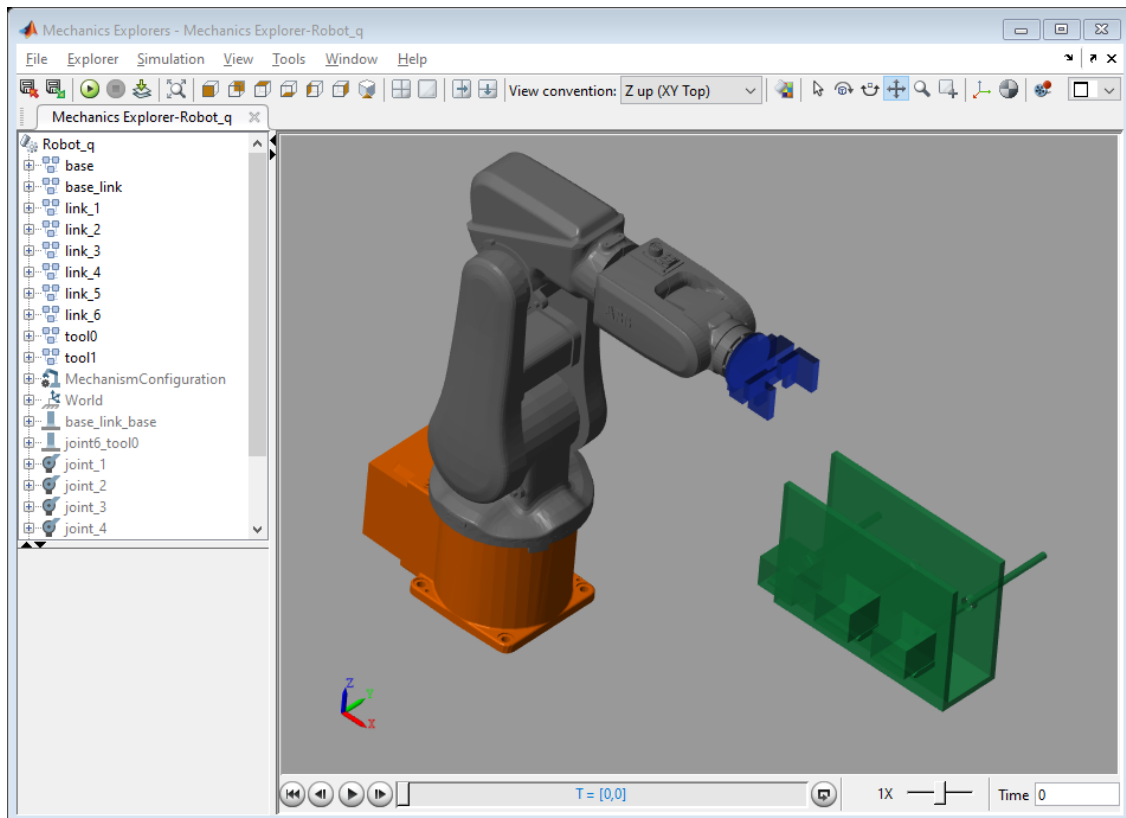


Figura 56 Ventana de simulación del modelo de Simulink

### 3.2.3. App Designer

La aplicación App Designer de Matlab permite desarrollar interfaces para emplear en tareas que se desarrollan con Matlab. Permite programar la aplicación tanto visualmente, con botones, barras deslizantes o menús, como el código y eventos que surjan dentro de la aplicación.

La finalidad del HMI es dotar a toda la estación diseñada de un programa con el que controlarla. Permite al usuario elegir entre realizar una simulación con Simulink del robot dentro de la estación o poder conectarse al robot real. Se pueden realizar trayectorias probadas previamente en simulación, seleccionar una herramienta o guardar los puntos que desee el usuario.

El interfaz también debe de incluir las trayectorias programadas para el entorno de trabajo en el que se encuentre, por lo que habrá un menú de selección del objeto de trabajo y el lugar donde se encuentra este objeto. También existirá una diferenciación entre un modo automático, con estas trayectorias programadas, y un movimiento manual, donde se pueda mover libremente al robot.

El funcionamiento del interfaz gráfico queda recogido en el **MANUAL DE USUARIO**, por lo que a continuación se comentarán los elementos incluidos y su función dentro del interfaz.

- **Switch Simulación**: interruptor de dos estados que permite realizar la simulación de la estación del robot con el modelo de Simulink. Cuando pasa a estado On se carga el modelo y aparece una pantalla nueva con el robot y la estación.
- **Switch Modo**: interruptor de dos estados que permite cambiar el modo de trabajo, tanto en la simulación como en el robot real. El modo manual permite mover el robot a través del grupo de barras deslizantes y del grupo de coordenadas y rotación en el espacio. El modo automático permite realizar las trayectorias programadas sobre los objetos de trabajo a través de los Radio botón de la parte izquierda del HMI.
- **Botón Conectar**: botón de dos posiciones que permite conectarse vía TCP/IP al robot real y realizar las trayectorias programadas y movimientos manuales y desconectarse. El botón cambia su apariencia y texto dependiendo del estado del botón.
- **Botón Movimiento**: botón que manda el movimiento al robot, tanto a la simulación como al robot real.
- **Botón Stop**: botón que manda al robot real parar su movimiento, dado que en la simulación no se rompe nada. Su función es la de prevenir posibles choques o movimientos no deseados en el robot.
- **Radio Button Objeto**: conjunto de botones que permite seleccionar el objeto sobre el que se va a trabajar, esfera o cilindro.
- **Radio Button Lugar de Trabajo**: conjunto de botones que permite seleccionar el lugar del trabajo, la base o mundo del robot, la mesa (referida a las trayectorias de la esfera y el cilindro en la mesa), la pared o lateral derecho y el techo.

- **Radio Button Trayectoria:** conjunto de botones que permite seleccionar la trayectoria programada, la “Sencilla” como la “Rotando”.
- **Grupo de botones Barras:** conjunto de botones que permite seleccionar la barra que se desea tomar de su soporte.
- **Botón Tomar Barra:** botón que manda al robot realizar el movimiento de tomar la barra seleccionada en el grupo de botones de Barras.
- **Botón Dejar Barra:** botón que manda al robot realizar el movimiento de dejar la barra en posesión en el grupo de botones de Barras.
- **Botón Abrir Pinza:** botón que abre la pinza del robot.
- **Botón Cerrar Pinza:** botón que cierra la pinza del robot.
- **Grupo de Barras deslizantes Ejes del robot:** barras que permiten mover al robot articularmente a través de cada eje del robot.
- **Grupo de Campo numérico X, Y, Z, RotX, RotY, RotZ:** campos numéricos que mueven al robot al punto seleccionado a través de su la posición X, Y, Z sobre la base seleccionada en el Radio Button de lugar de trabajo con la rotación de la herramienta fijada con las RotX, RotY, RotZ.
- **Grupo de botones Guardar, Repetir y Borrar Articulaciones:** el botón “Guardar articulación” guarda la posición articular que marquen las barras deslizantes, el botón “repetir articulación” realiza una trayectoria a través de todas las posiciones articulares que se han guardado, y el botón “borrar articulación” elimina las posiciones articulares almacenadas.
- **Grupo de botones Guardar Punto, Repetir y Borrar Secuencia:** el botón “Guardar punto” almacena el punto en el espacio que tengan los campos X, Y, Z con la rotación de los campos RotX, RotY, RotZ, el botón “Repetir Secuencia” realiza una trayectoria a través de todos los puntos guardados sobre la base que este seleccionada en el Radio Button de lugar de trabajo, y el botón “Borrar Secuencia” elimina los puntos almacenados.

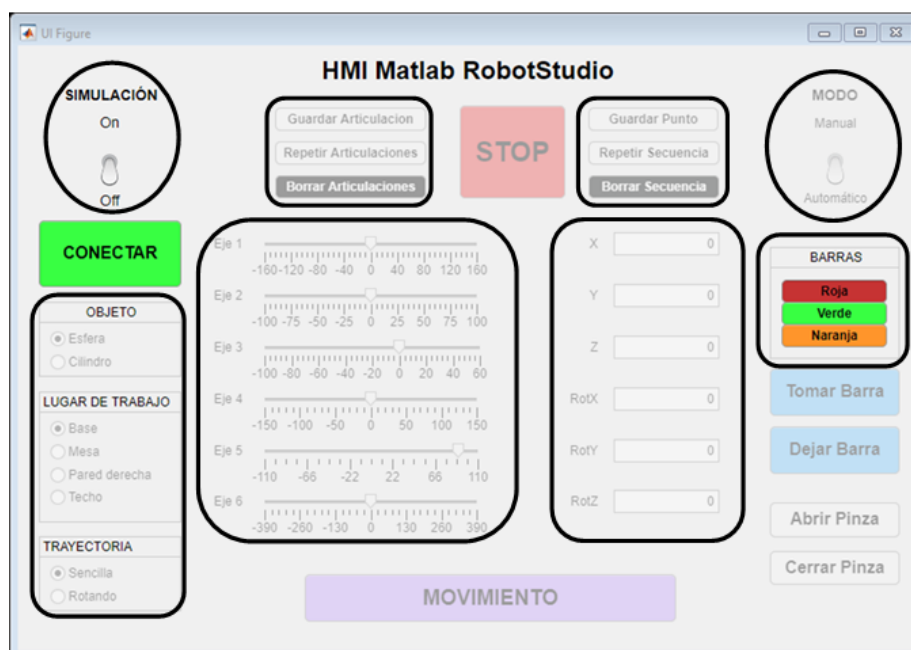


Figura 57 HMI desarrollado en App Designer de Matlab

### 3.3. Gestión del robot con RobotStudio

El software RobotStudio, Figura 58, es una herramienta de simulación y programación offline propio de la marca ABB. Proporciona las herramientas para la programación, optimización y simulación sin necesitar el robot real, [12].



Figura 58 Icono del software RobotStudio de ABB

El lenguaje de programación empleado en este programa y los robots ABB se denomina RAPID (Robotics Application Programming Interactive Dialogue). RAPID es un lenguaje de programación textual de alto nivel y modular, Figura 59. Una aplicación en RAPID está formado por un programa y módulos de sistema, [13].

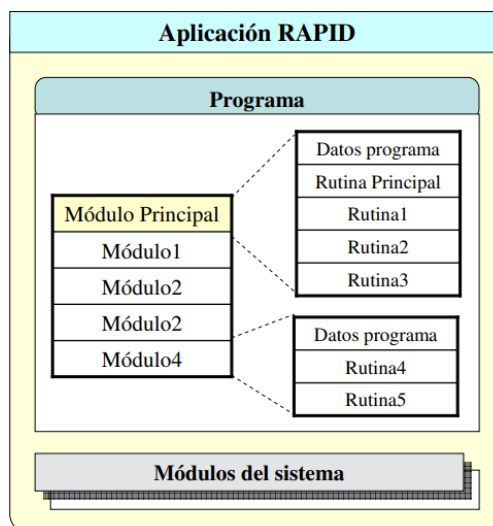


Figura 59 Estructura de una aplicación RAPID

Un programa en RAPID es una secuencia de instrucciones que permiten controlar al robot y normalmente está formado por tres partes, Figura 60, claramente diferenciadas:

- **Rutina principal (main)**: rutina con la que se inicia la ejecución del programa.
- **Subrutinas**: código dividido en partes más pequeñas para tener un programa modular.
- **Datos de programa**: se tratan de constantes, posiciones, sistemas de coordenadas.



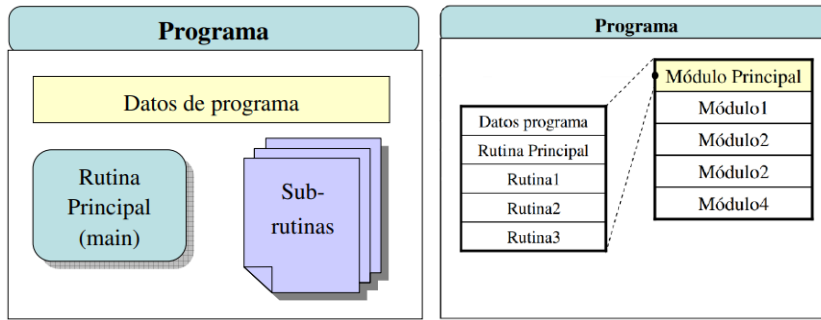


Figura 60 Estructura de un programa en RAPID

Las rutinas son un conjunto de declaraciones de datos junto al conjunto de instrucciones que se van a ejecutar en una tarea. Pueden dividirse en tres grupos:

- **Procedimientos:** conjunto de instrucciones que no devuelve valor.
- **Función:** conjunto de instrucciones que devuelve un valor.
- **Trap:** conjunto de instrucciones ejecutado cuando se produce una interrupción.

En este proyecto se han empleado dos módulos, el módulo principal, M\_Main, con la rutina main, que únicamente realiza una llamada al módulo M\_tcpABB, que es el que gestiona toda la comunicación entre el robot y Matlab, Figura 61.

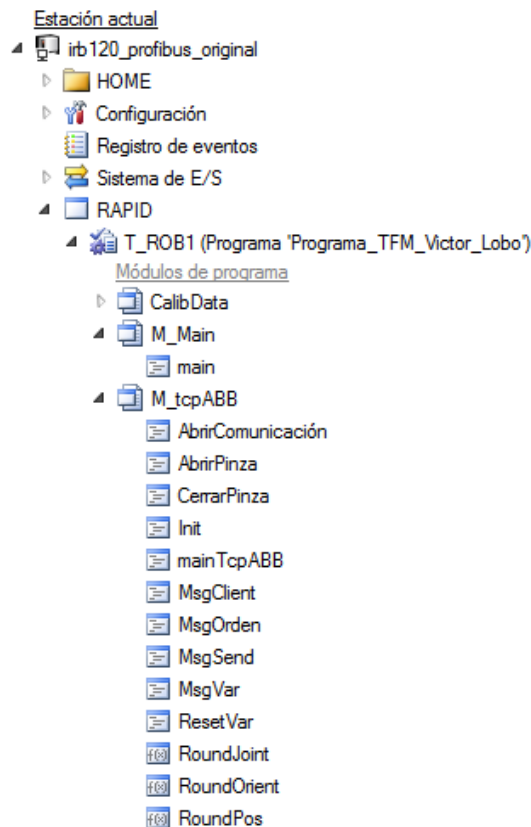


Figura 61 Módulos de la programación en RAPID

El módulo M\_tcpABB está formado por declaraciones de datos y varios procedimientos y funciones, Figura 61, explicados bajo estas líneas.

### 3.3.1. Declaración de datos

En la declaración de datos se han definido el conjunto de instrucciones que se pueden recibir, los tipos de variables básicos con los que operar, el socket de la comunicación con Matlab, variables genéricas, ya que el programa Matlab las modificada, como la herramienta y la base de trabajo, y un vector para almacenar los datos guardados a través del interfaz de usuario. Estas declaraciones se pueden ver en los anexos del proyecto.

### 3.3.2. Funciones

Las funciones son un conjunto de instrucciones que devuelve un valor. En este caso las funciones se han empleado para redondear el valor recibido de Matlab para que entren en una cadena de caracteres apropiada. Las funciones del módulo redondean los valores de la posición recibida, RoundPos, la orientación recibida, RoundOrient, y los valores articulares, RoundJoint. El contenido de estas funciones se puede ver en los anexos del proyecto.

### 3.3.3. Procedimientos

Los procedimientos son un conjunto de instrucciones que no devuelven un valor. Como los procedimientos programados son más numerosos que las funciones, se van a enumerar y explicar a continuación:

- mainTcpABB: rutina de inicio para a abrir la comunicación TCP/IP.
- AbrirComunicacion: define el socket del cliente y espera a que se conecten.
- AbrirPinza: abre la pinza del robot.
- CerrarPinza: cierra la pinza del robot.
- Init: bucle que está esperando mensajes del cliente.
- MsgClient: rutina que recibe el mensaje del cliente y diferencia si es un mensaje de orden o de variables
- MsgOrden: rutina que recibe una orden y la ejecuta.
- MsgVar: rutina que recibe una cadena de caracteres y modifica variables del robot.
- MsgSend: rutina que envía al cliente el resultado de la instrucción que ha realizado, siendo error o instrucción correctamente realizada.
- ResetVar: rutina que resetea todas las variables modificadas a sus valores iniciales.

### 3.3.4. Simulación

El software RobotStudio también permite la simulación de estaciones robotizadas. En este caso tenemos que importar el robot y los archivos que componen la estación para que sea una simulación lo más real posible.

Los objetos y las piezas modeladas en el diseño, se importan a fichero .stl y se pueden cargar en RobotStudio, únicamente tenemos que posicionarlos acorde a las posiciones definidas en apartados anteriores.

Dentro de este software se pueden guardar “estados”, con los que podremos simular los diferentes escenarios por separado

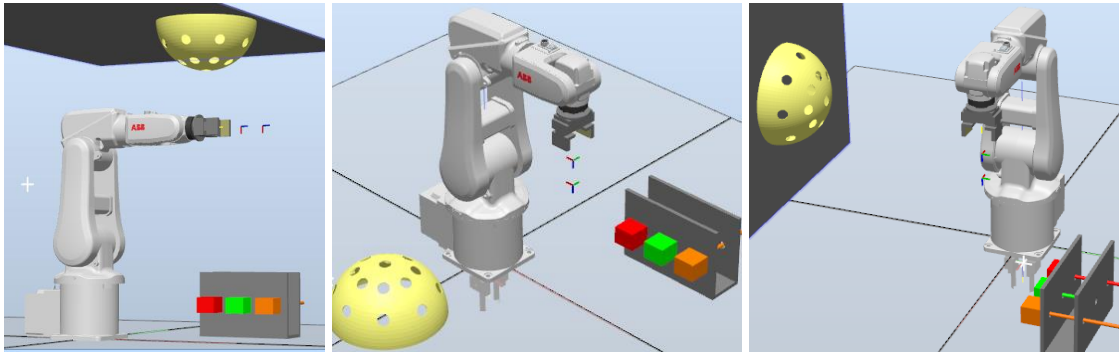


Figura 62 Estados de simulación de la semiesfera

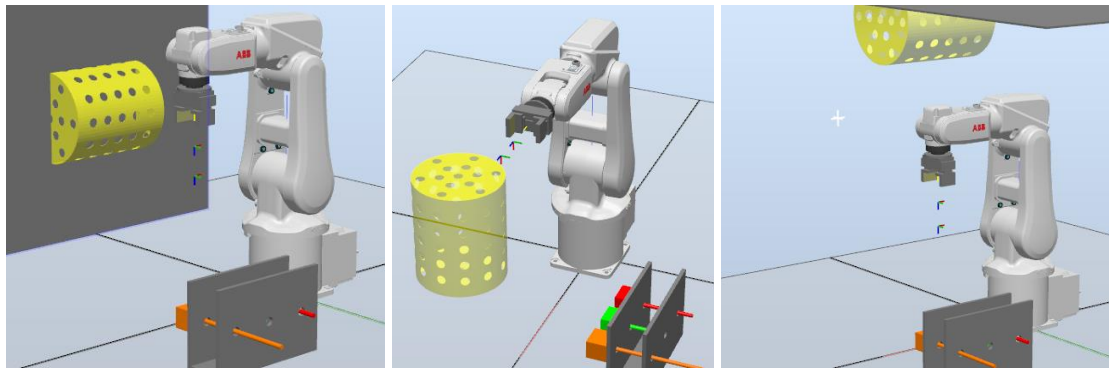


Figura 63 Estados de simulación del cilindro



## CAPITULO 4. MANUAL DE USUARIO

En este apartado se recoge el funcionamiento del interfaz de usuario para un correcto uso de la aplicación, Figura 64.

Al ejecutar la aplicación únicamente están disponibles el switch de la simulación o el botón de conectar. Para comenzar algún tipo de acción hay que o bien realizar una simulación de la estación o conectarse al robot real. En caso de conectarse al robot real y no haber ejecutado ninguna simulación, se carga la simulación para ver el movimiento que se va a realizar en el robot real previamente en el simulador.

Una vez seleccionada una de estas opciones, es momento de seleccionar una herramienta con la que se va a trabajar, por lo que se desbloquea el botón de tomar barra. Con una herramienta del soporte de herramientas, se puede realizar lo que desee el usuario:

- Si desea realizar las trayectorias programadas se debe seleccionar el modo automático y realizar la combinación de radio buttons de la trayectoria que desee, por ejemplo: Esfera, Pared derecha y Sencilla. Para que el robot se mueva se debe pulsar el botón “movimiento”.
- Si se desea realizar un movimiento manual, basta con seleccionar el modo manual y mover el robot a través de las opciones disponibles, o por ejes, el robot se mueve en cada cambio del valor de las barras deslizantes o por puntos en el espacio referidos a la base y objeto que estén seleccionados en los diferentes radio buttons, para que se mueva hay que pulsar el botón “movimiento”.
- Si se desea guardar alguna configuración articular se debe pulsar el botón “guardar articulación”, si se desean repetir todas las configuraciones de ejes guardadas se debe pulsar “repetir articulaciones” y se desean borrar todas las configuraciones guardadas se debe pulsar el botón “borrar articulaciones”.
- Si se desea guardar algún punto se debe pulsar el botón “guardar punto”, si se desean repetir todas las configuraciones de ejes guardadas se debe pulsar “repetir trayectorias” y se desean borrar todas las configuraciones guardadas se debe pulsar el botón “borrar trayectorias”.

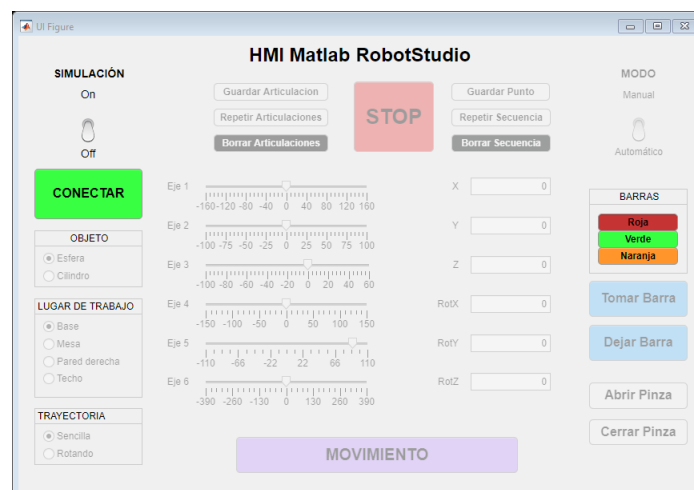


Figura 64 HMI desarrollado en App Designer de Matlab



## CAPITULO 5. RESULTADOS

En este capítulo se recogen los resultados obtenidos de todo el proyecto, realizando pruebas del interfaz de usuario junto con los entornos de simulación en Simulink y RobotStudio y objetos y entornos fabricados.

En primer lugar, se describirán las pruebas realizadas en el interfaz gráfico junto con la simulación en Simulink, posteriormente se realizarán esas mismas pruebas en RobotStudio que serían extrapolables al robot real y finalmente se mostraran los objetos fabricados que forman parte de los entornos de trabajo diseñados.

### 5.1. Interfaz gráfico junto a simulación en Simulink

Lanzamos la aplicación con el interfaz gráfico poniendo el switch de simulación a “On” y cargando la ventana con el modelo de Simulink, Figura 65.

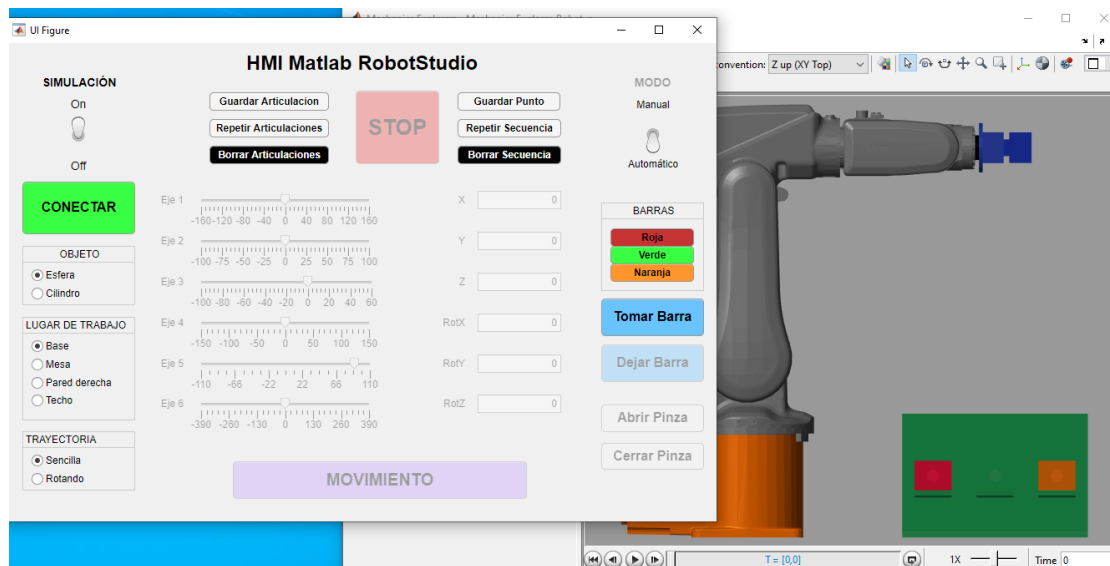


Figura 65 HMI con la simulación de Simulink lanzada

Seguidamente seleccionamos una herramienta del soporte de barras, la barra roja de 20cm. El robot, realiza en primer lugar un movimiento a la posición “home”, después la trayectoria completa de tomar la herramienta y finalmente el último movimiento hasta la “home”.

La simulación carga movimientos individuales reproduciendo, en primer lugar, el movimiento a su posición de reposo. Luego toda la trayectoria de coger la barra seleccionada y finalmente volver a la posición de reposo. El valor de la velocidad de simulación se puede modificar para poder ver todos los movimientos correctamente. Nótese que al comienzo del movimiento no tiene barra, Figura 66, y al final la tiene tomada, Figura 68.

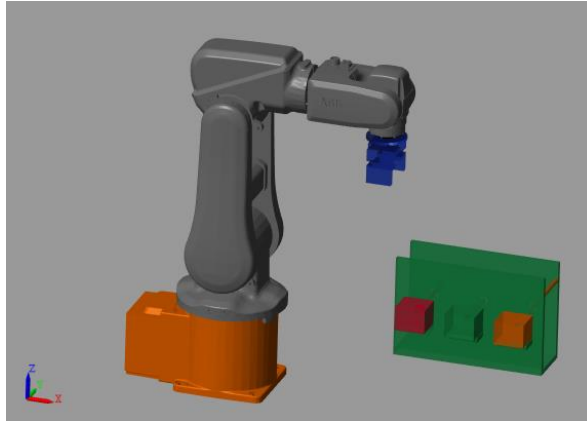


Figura 66 Robot de simulación en posición de reposo

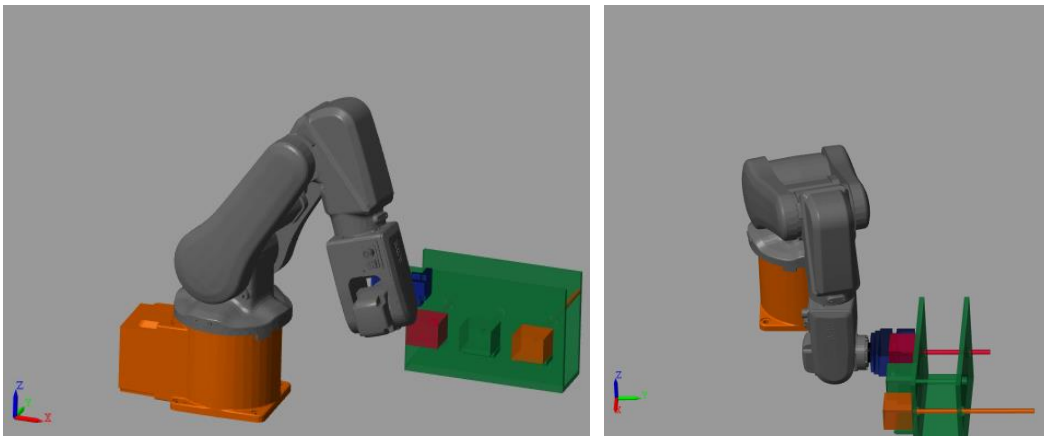


Figura 67 Robot ejecutando trayectoria de tomar herramienta

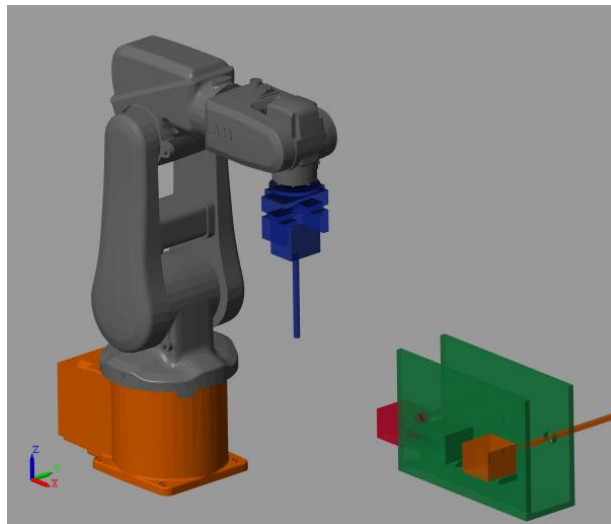


Figura 68 Robot en posición de reposo con la herramienta tomada

Tras tomar una de las barras, se permite al usuario realizar la acción que quiera con el robot, puede realizar una trayectoria de las programadas sobre uno de los objetos de trabajo sobre el lugar de trabajo que desee. Esta elección se realiza con el modo automático y a través de la combinación de los radio buttons laterales. Para mostrarlo en este trabajo vamos a ejecutar todas las trayectorias programadas, sobre la esfera en la mesa, el lateral derecho y techo y sobre el cilindro en la mesa, lateral derecho y el techo.



Seleccionamos la opción que deseamos marcando los radio buttons y pulsaremos el botón “Movimiento”, Figura 69. Apréciase que en la carga de la estación no aparece ningún objeto y cuando comencemos la trayectoria se cargara para realizar una simulación correcta.



Figura 69 Selección de la trayectoria con los radio buttons

En primer lugar, el robot se posiciona para ejecutar la trayectoria, Figura 70, y la ejecuta siguiendo el orden descrito en el apartado Trayectoria Semiesfera Mesa, ver Figura 71 y Figura 72.

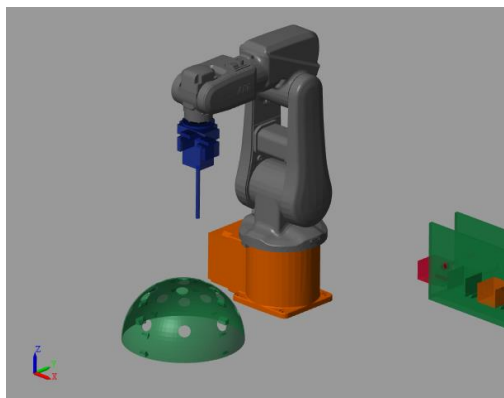


Figura 70 Posicionamiento del robot para realizar la trayectoria

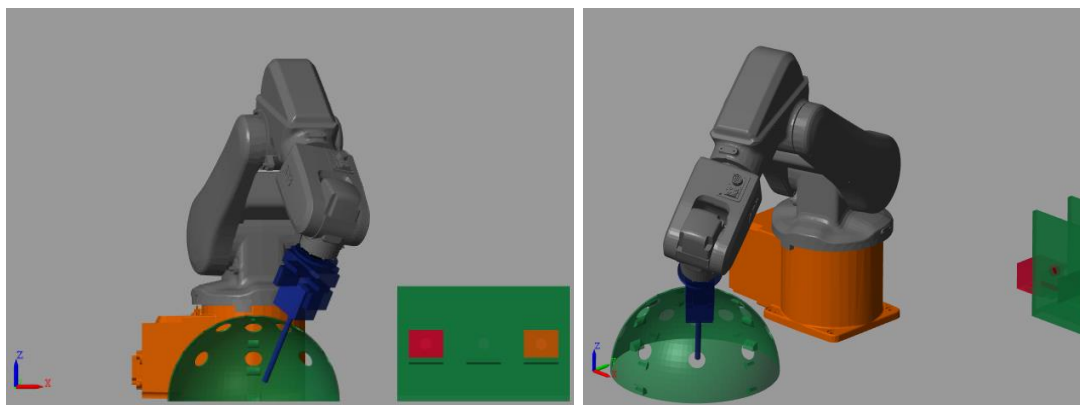


Figura 71 Robot orientado realizando la trayectoria

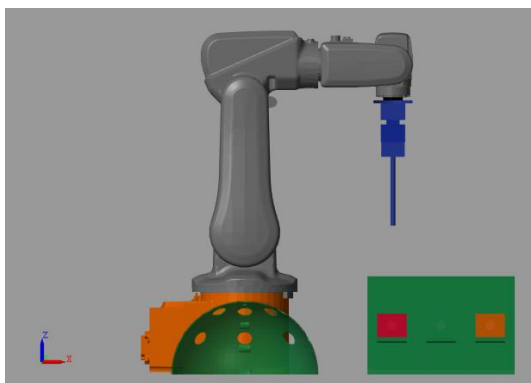


Figura 72 Robot en la posición de reposo

A continuación, realizamos la Trayectoria Semiesfera Lateral Derecho. Seleccionamos la opción que deseamos marcando los radio buttons y pulsaremos el botón “Movimiento”, Figura 73. En primer lugar, el robot se posiciona para ejecutar la trayectoria, Figura 74, Figura 75 y Figura 76.



Figura 73 Selección de la trayectoria con los radio buttons

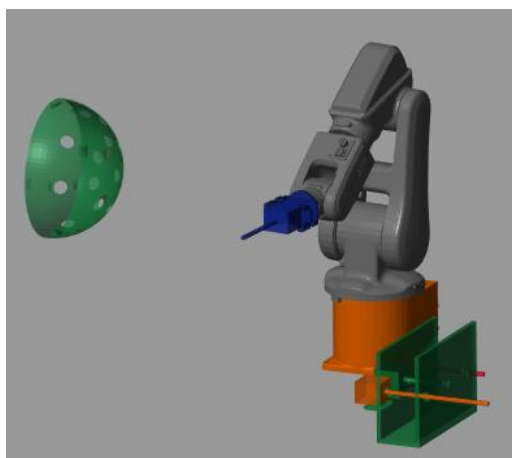


Figura 74 Posicionamiento del robot para realizar la trayectoria

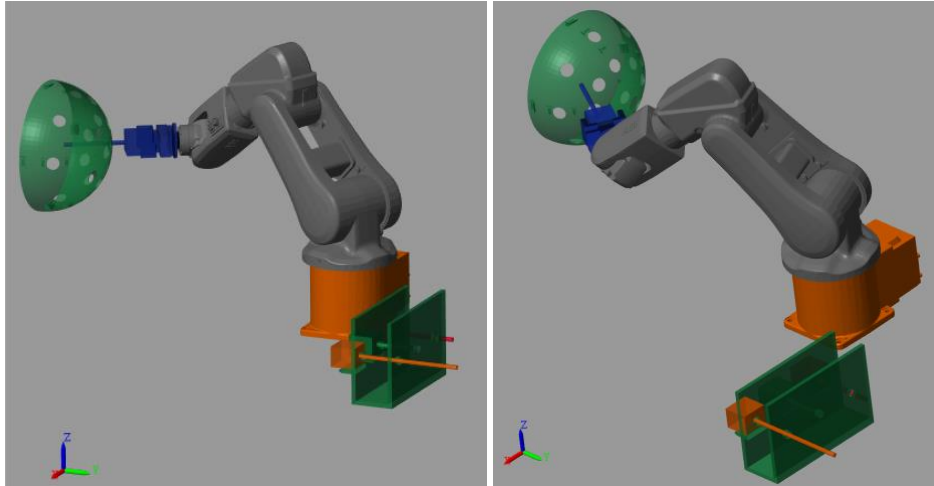


Figura 75 Robot orientado realizando la trayectoria

Moviéndonos en la ventana de simulación podemos apreciar la rotación que adquiere el robot para entrar perpendicularmente en los orificios del objeto, ver Figura 76. Finalmente, el robot vuelve a la posición de reposo, Figura 77.

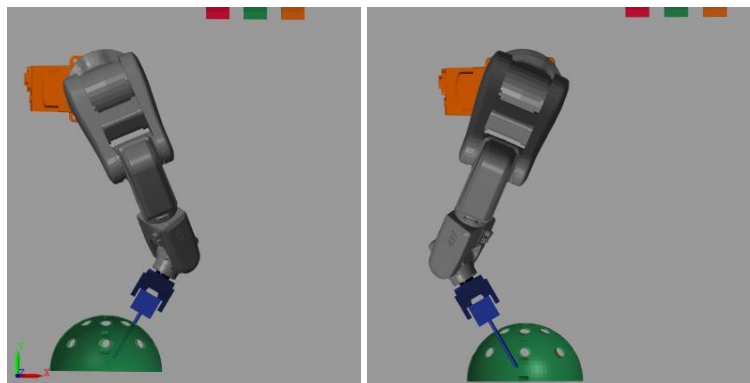


Figura 76 Orientación del robot realizando la trayectoria

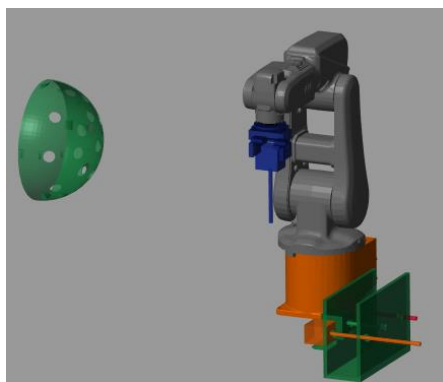


Figura 77 Robot en la posición de reposo

A continuación, realizamos la trayectoria de Trayectoria Semiesfera Techo. Seleccionamos la opción que deseamos marcando los radio buttons y pulsaremos el botón “Movimiento”, Figura 78. En primer lugar, el robot se posiciona para ejecutar la trayectoria, Figura 79, y la ejecuta siguiendo el orden descrito en el apartado Trayectoria Semiesfera Techo, Figura 80. Finalmente, el robot vuelve a su posición de reposo, Figura 81.



Figura 78 Selección de la trayectoria con los radio buttons

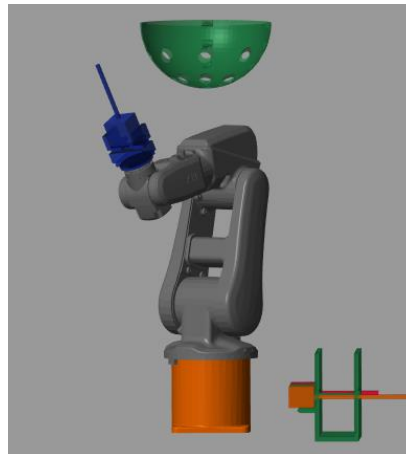


Figura 79 Robot posicionándose para realizar la trayectoria

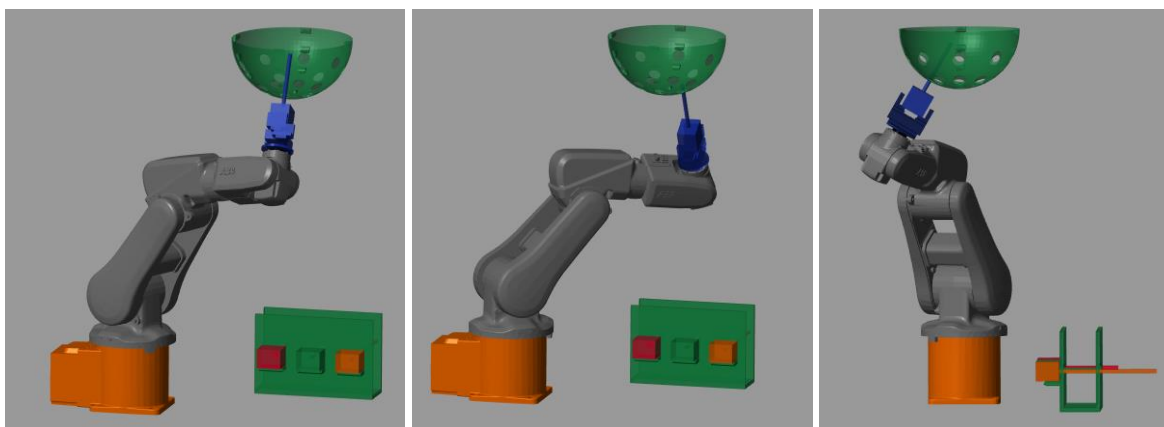


Figura 80 Robot orientándose realizando la trayectoria

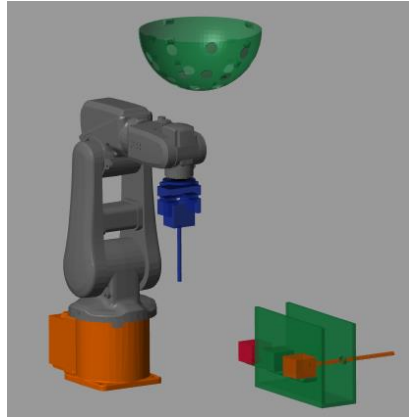


Figura 81 Robot en posición de reposo

A continuación, realizaremos la trayectoria sobre el cilindro en la mesa. Seleccionamos, empleando los radio buttons, el cilindro, la mesa y pulsamos movimiento, Figura 82. En primer lugar, el robot se posiciona para ejecutar la trayectoria, Figura 83 y la ejecuta siguiendo el orden descrito en el apartado Trayectoria Cilindro Mesa, Figura 84. Finalmente vuelve a la posición de reposo, Figura 85.



Figura 82 Selección de la trayectoria con los radio buttons

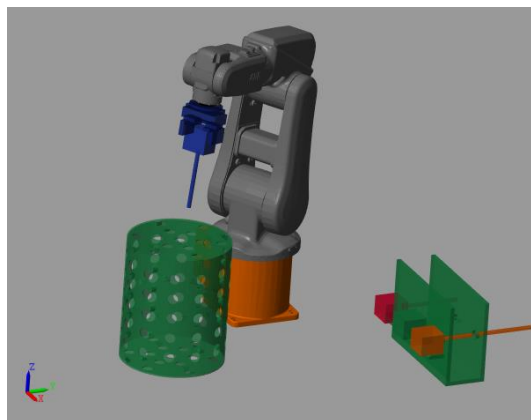


Figura 83 Robot posicionándose para realizar la trayectoria

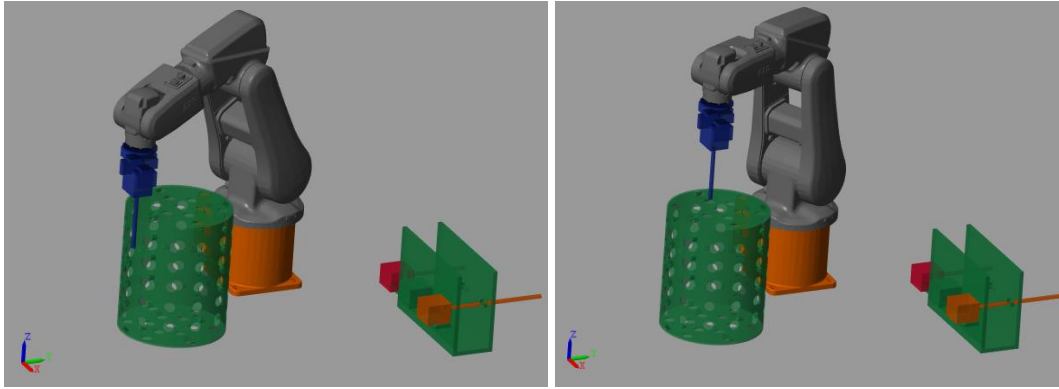


Figura 84 Robot orientándose realizando la trayectoria

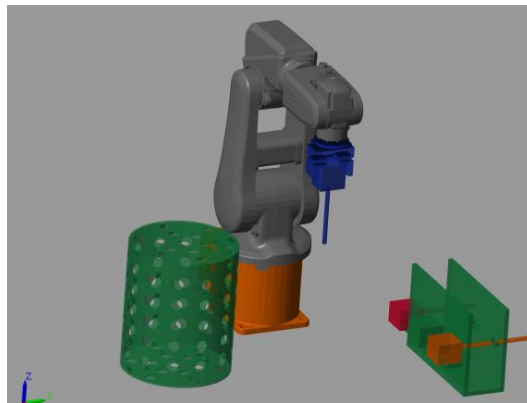


Figura 85 Robot en la posición de reposo

A continuación, realizaremos la Trayectoria Cilindro Lateral Derecho. Seleccionamos, empleando los radio buttons, el cilindro, el lateral derecho y pulsamos movimiento, Figura 86. En primer lugar, el robot se posiciona para ejecutar la trayectoria, ver Figura 87, y la ejecuta siguiendo el orden descrito en el apartado Trayectoria Cilindro Techo, Figura 88. Finalmente vuelve a la posición de reposo, Figura 89.



Figura 86 Selección de la trayectoria con los radio buttons

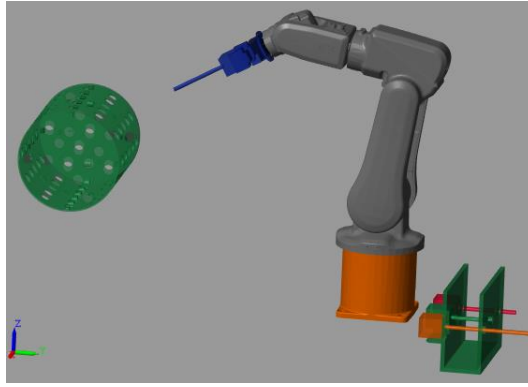


Figura 87 Robot posicionándose para realizar la trayectoria

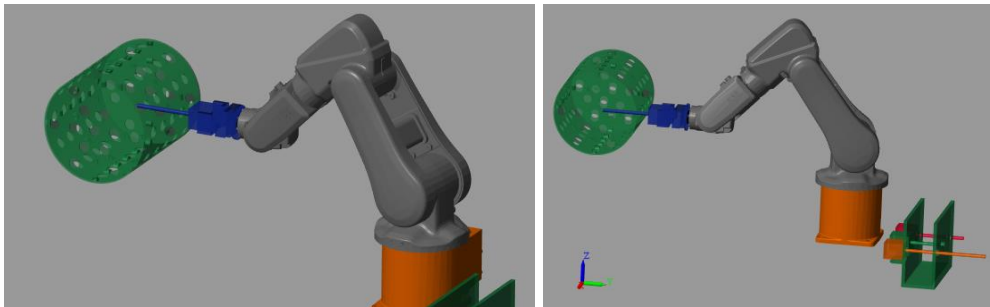


Figura 88 Robot orientándose realizando la trayectoria

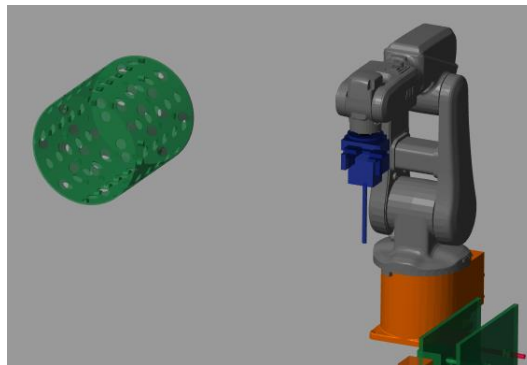


Figura 89 Robot en posición de reposo

A continuación, realizaremos la Trayectoria Cilindro Techo. Seleccionamos, empleando los radio buttons, el cilindro, el techo y pulsamos movimiento, Figura 90. En primer lugar, el robot se posiciona para ejecutar la trayectoria, Figura 91 y la ejecuta siguiendo el orden descrito en el apartado Trayectoria Cilindro Techo y finalmente vuelve a la posición de reposo, Figura 92.



Figura 90 Selección de la trayectoria con los radio buttons

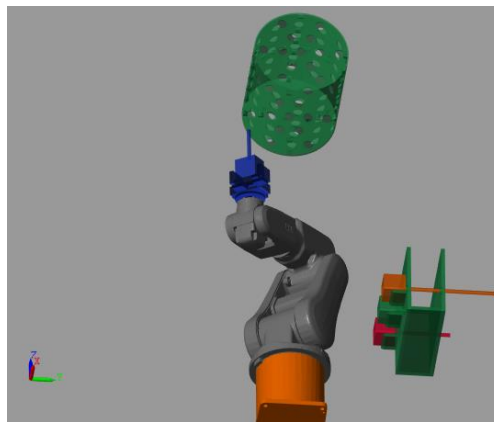


Figura 91 Robot posicionándose para realizar la trayectoria

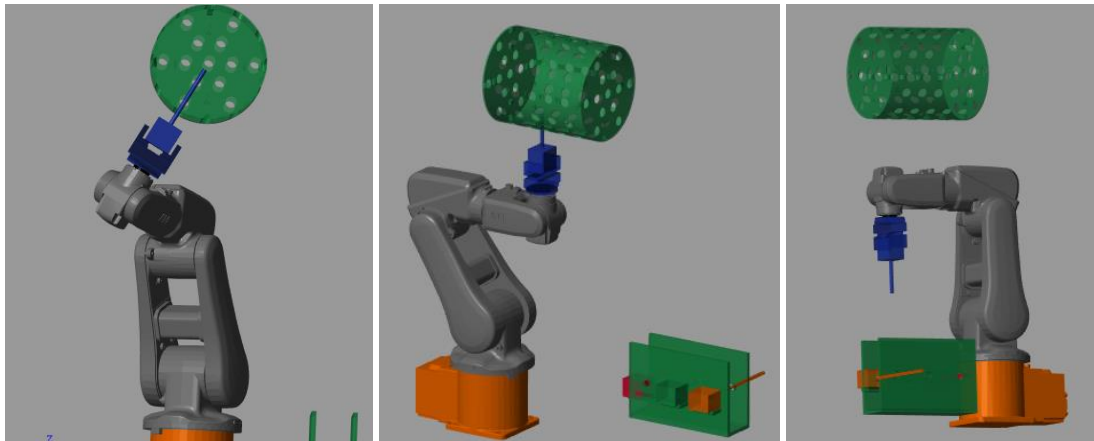


Figura 92 Robot orientándose realizando la trayectoria y en posición de reposo



También es posible controlar el robot de simulación en modo manual. Este modo manual se selecciona empleando el switch de Modo. Empleando el modo manual podemos controlar el robot por ejes, con las barras deslizantes, Figura 93 y Figura 94. Además, es posible guardar cada una de las articulaciones, repetir todas las articulaciones almacenadas y borrarlas.

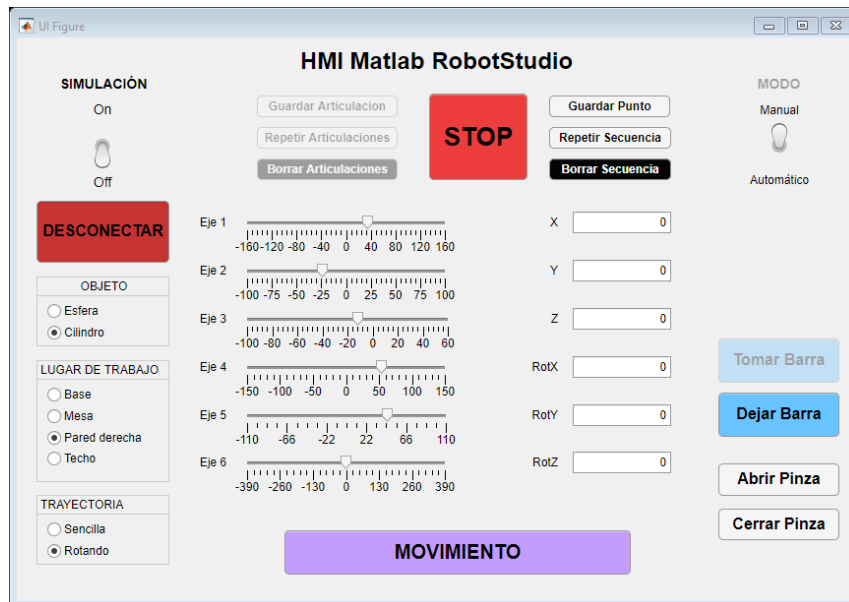


Figura 93 Barras deslizantes en modo manual

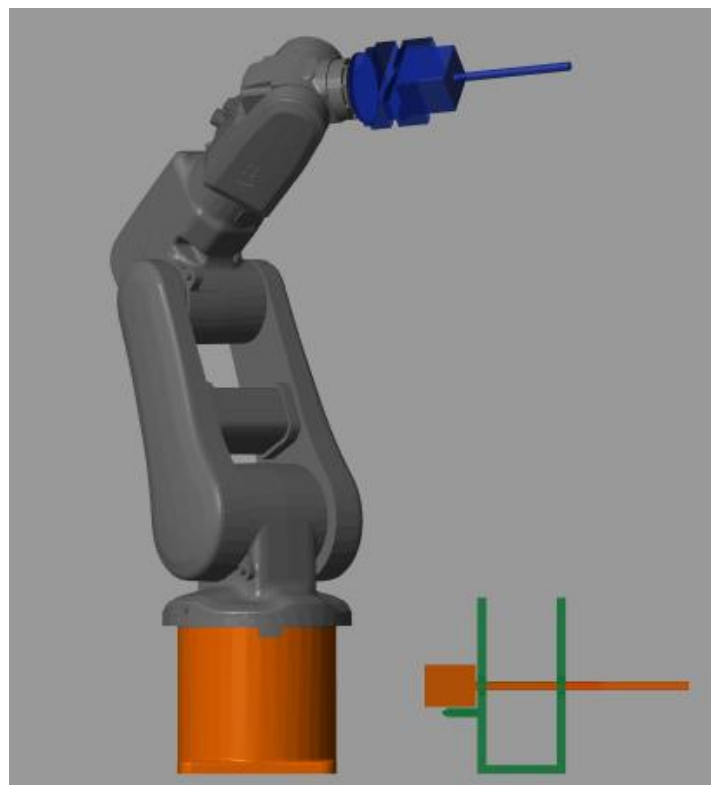


Figura 94 Robot adquiriendo los valores de las barras deslizantes

Del mismo modo se puede mover al robot por coordenadas, introduciendo el punto en los campos numéricos y seleccionando la base de trabajo seleccionada y el robot ira a esa posición, Figura 95 y Figura 96. Además, es posible guardar cada uno de los puntos, repetir todos los puntos almacenados y borrarlos.



Figura 95 Campos numéricos y selección de la base de movimiento manual

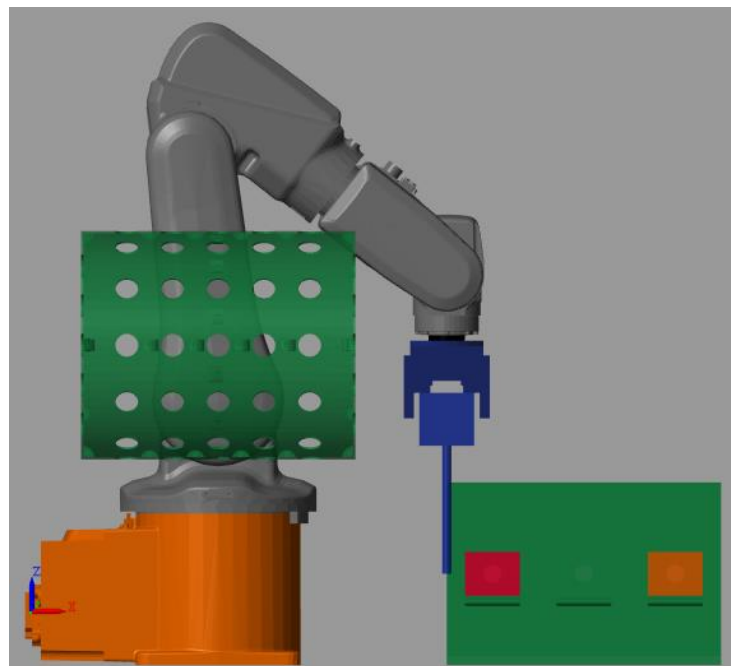


Figura 96 Robot en la posición mandada desde el interfaz en modo manual

## 5.2. Interfaz gráfica junto a simulación en RobotStudio

Lanzamos la aplicación con el interfaz gráfico y, como se ha explicado en el capítulo **MANUAL DE USUARIO**, pulsamos el botón de conectar una vez el robot este esperando conexión, Figura 97.

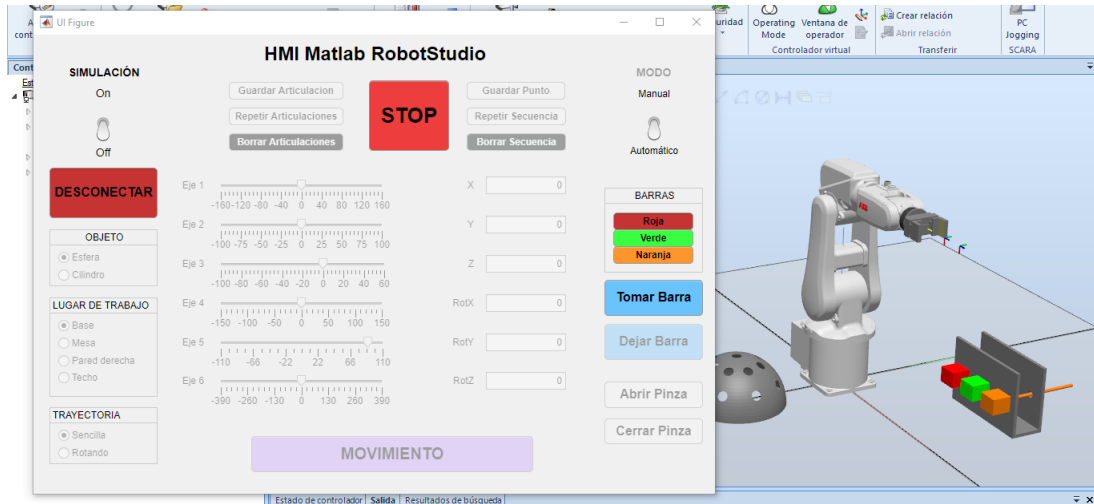


Figura 97 Conexión con RobotStudio del interfaz de usuario

Como no hemos realizado la simulación, se lanza también la ventana de simulación de Simulink. Como vamos a recrear los mismos movimientos que en el apartado anterior no se van a mostrar los movimientos que realice el modelo de Simulink. Como tampoco podemos conectarnos al robot real debido a la situación vivida en el momento de ejecución de este proyecto por el Covid-19, realizaremos una simulación con RobotStudio. Cargamos el estado de la esfera en la pared derecha.

El siguiente paso es seleccionar una herramienta del soporte de barras, así que seleccionaremos la barra roja, de 20cm. El robot, realiza en primer lugar un movimiento a la posición “home”, después la trayectoria completa de tomar la herramienta y finalmente el último movimiento hasta la “home”.

La simulación realiza los movimientos de la trayectoria de tomar barra, comenzando con su posición de reposo, Figura 98, después la trayectoria de coger la barra seleccionada, Figura 99 y finalmente su vuelta a la posición de reposo, Figura 100.

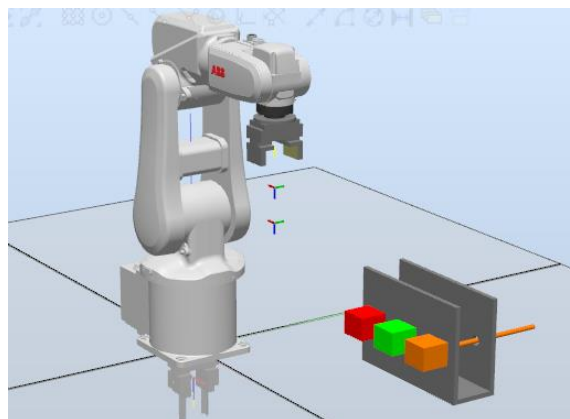


Figura 98 Posición de reposo del robot en RobotStudio

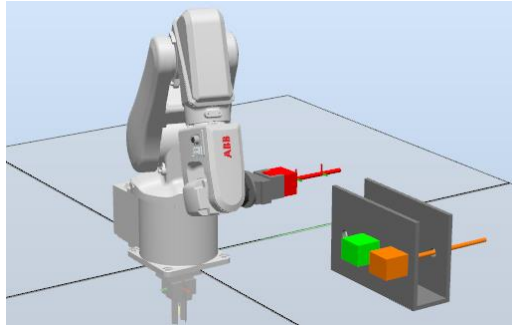


Figura 99 Robot realizando la trayectoria de tomar herramienta

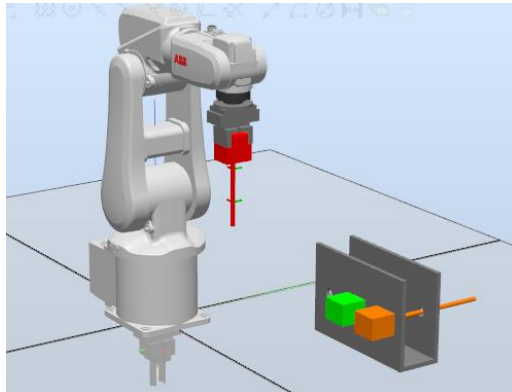


Figura 100 Posición de reposo del robot en RobotStudio

Tras tomar una de las barras, Figura 100, se permite al usuario realizar la acción que quiera con el robot, puede realizar una trayectoria de las programadas sobre uno de los objetos de trabajo sobre el lugar de trabajo que desee. Esta elección se realiza con el modo automático y a través de la combinación de los radio buttons laterales, Figura 101. Para mostrarlo en este trabajo vamos a ejecutar todas las trayectorias programadas, sobre la semiesfera en la mesa, el lateral derecho y techo y el cilindro en la mesa, lateral derecho y el techo.

Seleccionamos la opción que deseamos marcando los radio buttons y pulsaremos el botón “Movimiento”, Figura 101. Empezamos por la semiesfera en la mesa.



Figura 101 Selección de la trayectoria con los radio buttons

El robot en primer lugar se posiciona para ejecutar la trayectoria, Figura 102, y la ejecuta siguiendo el orden descrito en el apartado Trayectoria Semiesfera Mesa, Figura 103. Finalmente vuelve a su posición de reposo, Figura 104.

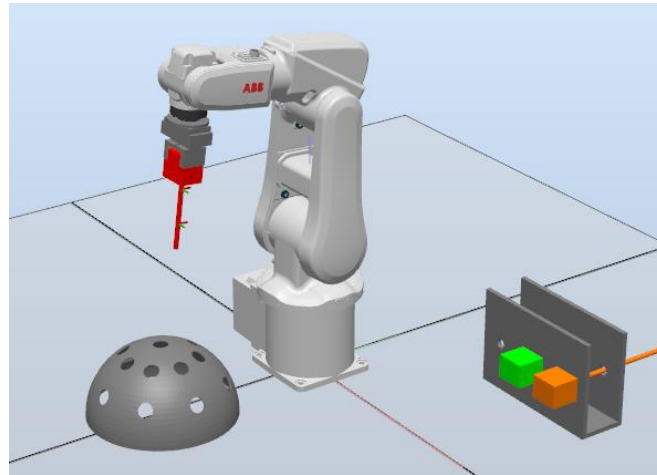


Figura 102 Posicionamiento del robot para ejecutar la trayectoria

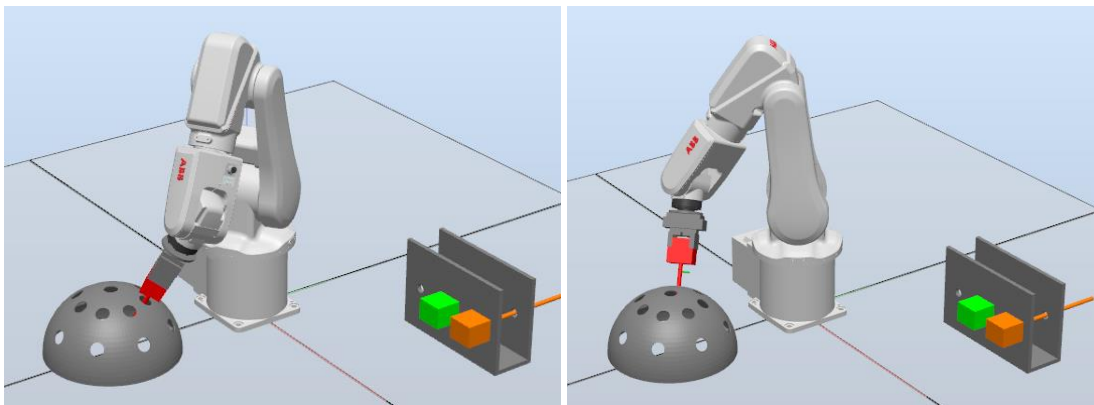


Figura 103 Robot orientándose para ejecutar la trayectoria

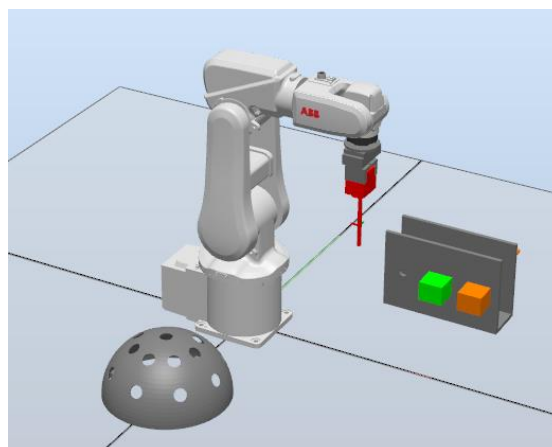


Figura 104 Robot en posición de reposo

A continuación, realizamos la trayectoria sobre el lateral derecho. Seleccionamos la opción que deseamos marcando los radio buttons y pulsaremos el botón “Movimiento”, Figura 105.



Figura 105 Selección de la trayectoria con los radio buttons

El robot en primer lugar se posiciona para ejecutar la trayectoria, Figura 106, y la ejecuta siguiendo el orden descrito en el apartado Trayectoria Semiesfera Lateral Derecho, ver Figura 107. Finalmente vuelve a su posición de reposo, Figura 108.

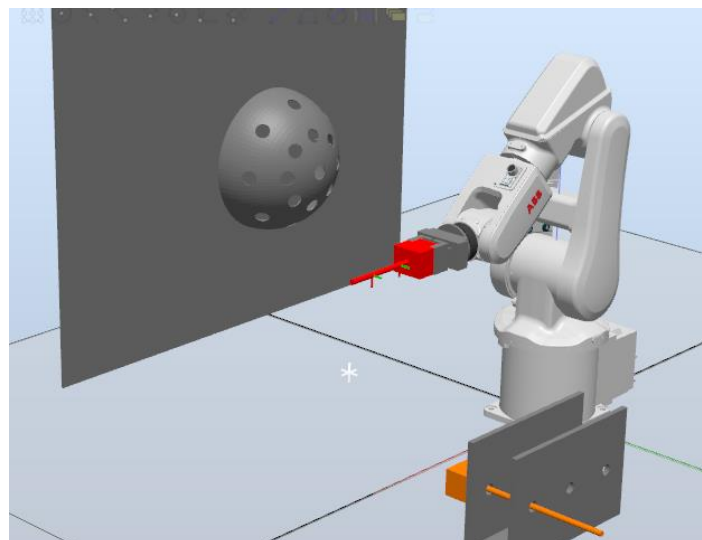


Figura 106 Posicionamiento del robot para ejecutar la trayectoria

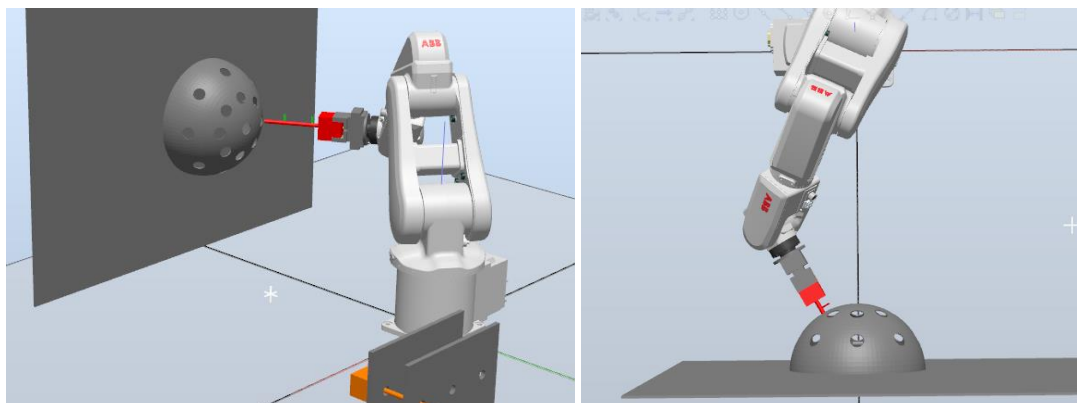


Figura 107 Robot orientándose para ejecutar la trayectoria

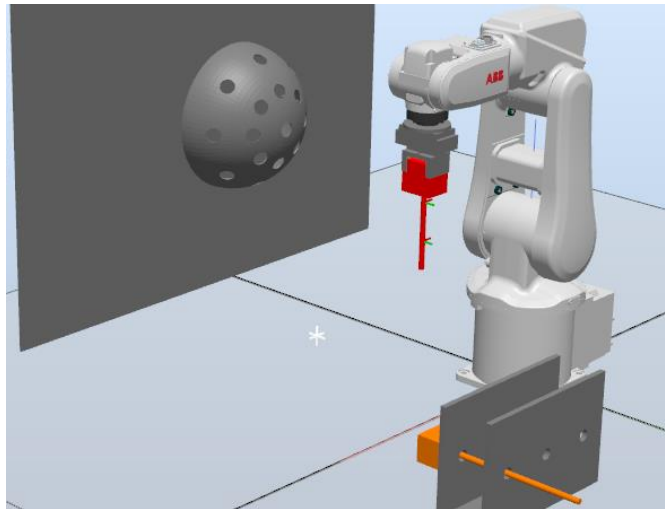


Figura 108 Robot en posición de reposo

A continuación, realizamos la trayectoria sobre el techo. Seleccionamos la opción que deseamos marcando los radio buttons y pulsaremos el botón “Movimiento”, Figura 109.

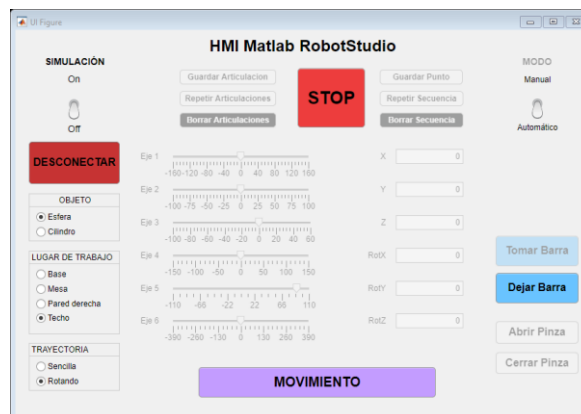


Figura 109 Selección de la trayectoria con los radio buttons

El robot en primer lugar se posiciona para ejecutar la trayectoria, Figura 110, y la ejecuta siguiendo el orden descrito en el apartado Trayectoria Semiesfera Techo y finalmente vuelve a su posición de reposo, Figura 111.

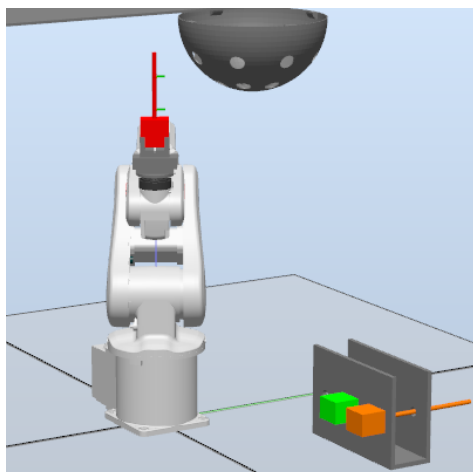


Figura 110 Robot posicionándose para ejecutar la trayectoria

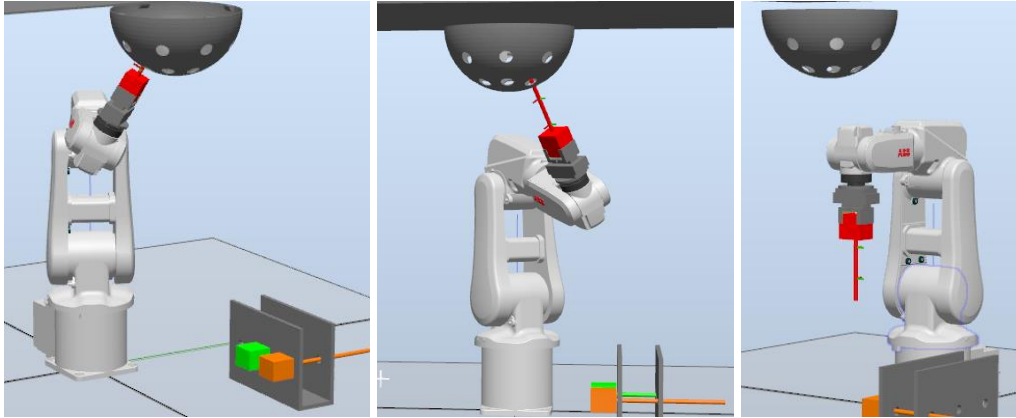


Figura 111 Robot orientándose para ejecutar la trayectoria y en posición de reposo

A continuación, realizaremos la trayectoria sobre el cilindro en la mesa, seleccionamos empleando los radio buttons el cilindro, la mesa y pulsamos movimiento, Figura 112.



Figura 112 Selección de la trayectoria con los radio buttons

El robot en primer lugar se posiciona para ejecutar la trayectoria, Figura 113, y la ejecuta siguiendo el orden descrito en el apartado Trayectoria Cilindro Mesa, ver Figura 114. Finalmente, vuelve a su posición de reposo, Figura 115.

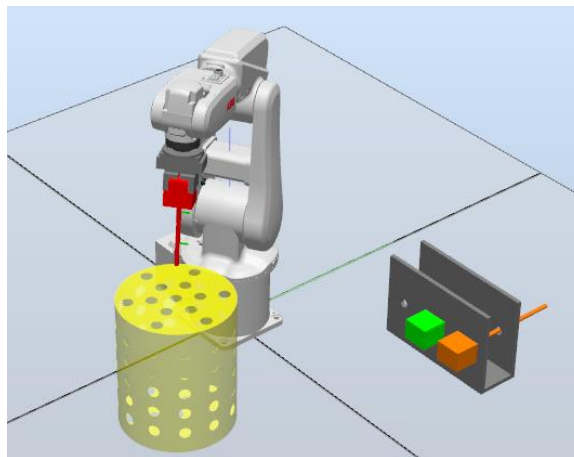


Figura 113 Robot posicionándose para ejecutar la trayectoria



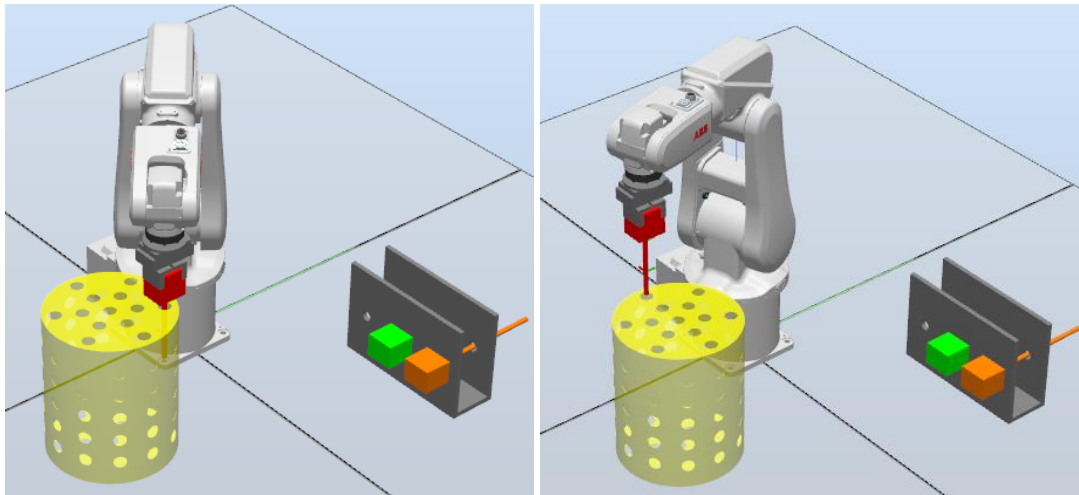


Figura 114 Robot orientándose para ejecutar la trayectoria

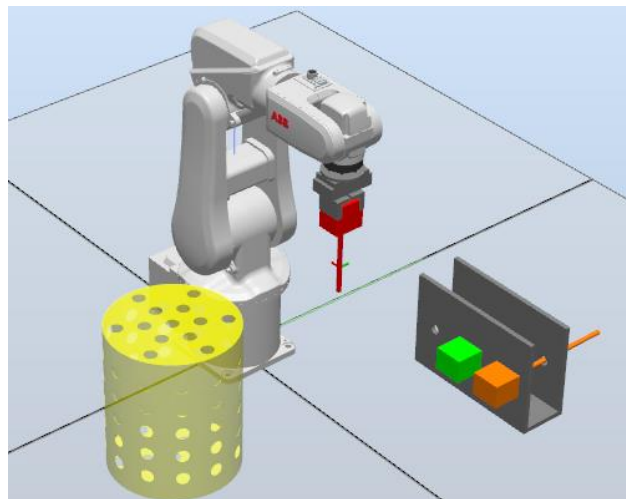


Figura 115 Robot en posición de reposo

A continuación, realizaremos la trayectoria sobre el cilindro en el lateral derecho, seleccionamos empleando los radio buttons el cilindro, la pared derecha y pulsamos movimiento, Figura 116.



Figura 116 Selección de la trayectoria con los radio buttons

El robot en primer lugar se posiciona para ejecutar la trayectoria, Figura 117, y la ejecuta siguiendo el orden descrito en el apartado Trayectoria Cilindro Lateral Derecho, Figura 118 y Figura 119. Finalmente vuelve a su posición de reposo Figura 120.

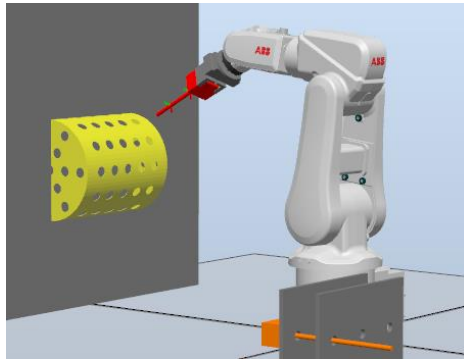


Figura 117 Robot posicionándose para ejecutar la trayectoria

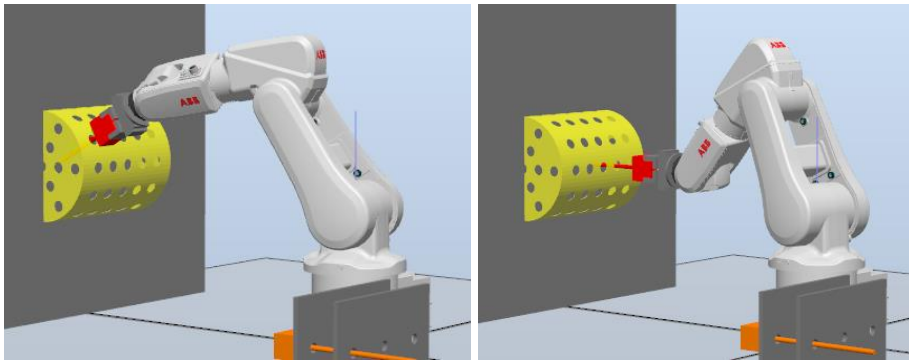


Figura 118 Robot orientándose para ejecutar la trayectoria

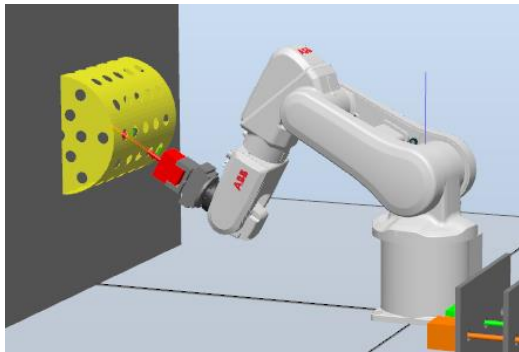


Figura 119 Robot orientándose para ejecutar la trayectoria

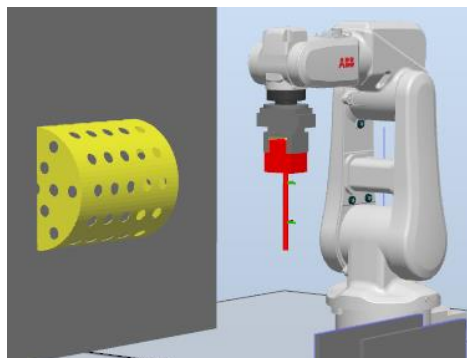


Figura 120 Robot en posición de reposo

A continuación, realizaremos la trayectoria sobre el cilindro en el techo, seleccionamos empleando los radio buttons el cilindro, el techo y pulsamos movimiento, Figura 121.



Figura 121 Selección de la trayectoria con los radio buttons

El robot en primer lugar se posiciona para ejecutar la trayectoria, Figura 122, y la ejecuta siguiendo el orden descrito en el apartado Trayectoria Cilindro Techo y finalmente vuelve a su posición de reposo, Figura 123.

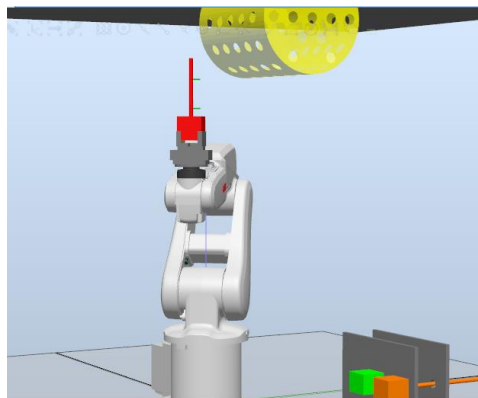


Figura 122 Robot posicionándose para ejecutar la trayectoria

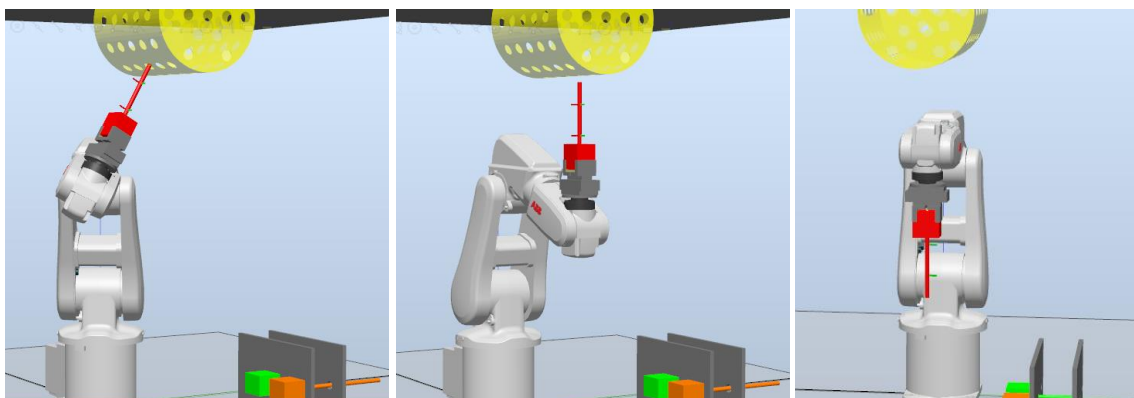


Figura 123 Robot orientándose para ejecutar la trayectoria

También es posible controlar el robot de RobotStudio en modo manual. Este modo manual se selecciona empleando el switch de Modo. Empleando el modo manual podemos controlar el robot por ejes, con las barras deslizantes, Figura 124 y Figura 125. Además, es posible guardar cada una de las articulaciones, repetir todas las articulaciones almacenadas y borrarlas.



Figura 124 Barras deslizantes en modo manual

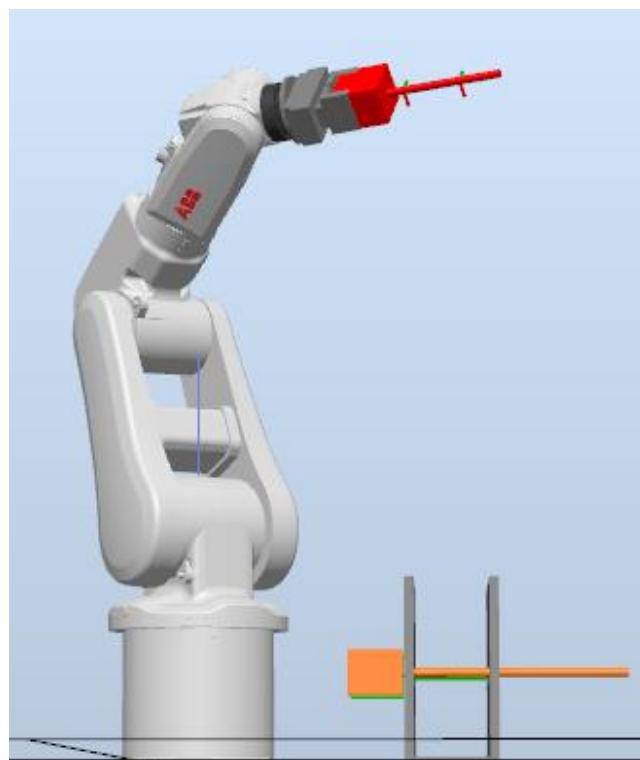


Figura 125 Robot adquiriendo los valores de las barras deslizantes

Del mismo modo se puede mover al robot por coordenadas, introduciendo el punto en los campos numéricos y seleccionando la base de trabajo seleccionada y el robot ira a esa posición, Figura 95 y Figura 96. Además, es posible guardar cada uno de los puntos, repetir todos los puntos almacenados y borrarlos.



Figura 126 Campos numericos y selección de la base de movimiento manual

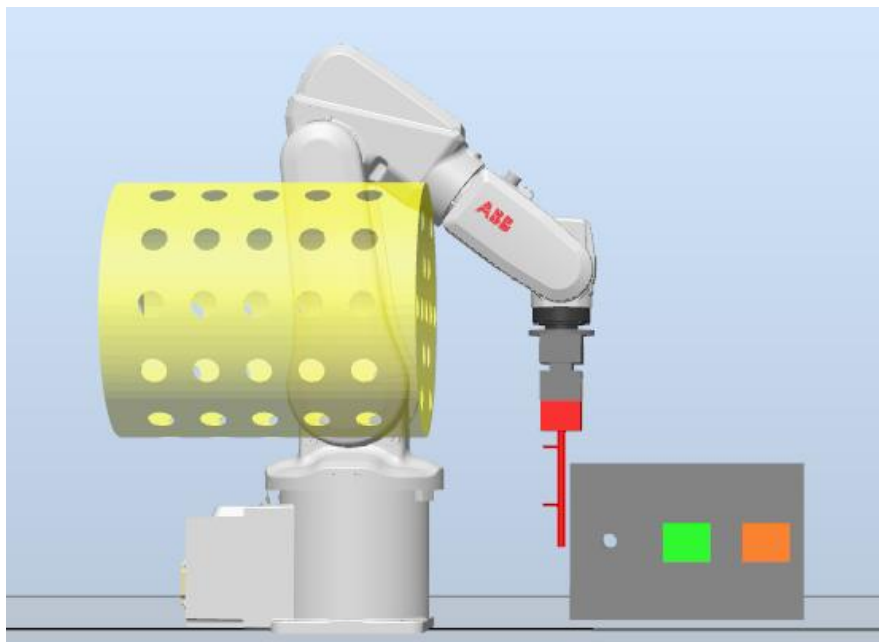


Figura 127 Robot en la posición mandada desde el interfaz en modo manual

### 5.3. Objetos fabricados

Empleando impresión 3D se han fabricado los modelos diseñados en el apartado Diseño 3D con Catia. Desde el programa de diseño CAD se exportan los ficheros a un archivo para que el programa de laminado de impresión 3D pueda segmentar la pieza y crear un archivo .gcode para que la impresora realice la pieza.

La impresora 3D que se va a emplear es una Creality Ender 3, Figura 128, se trata de una impresora FDM, Modelado de Deposición Fundida, con un volumen de impresión de 220 \* 220 \* 250mm. Todas las piezas se van a imprimir en plástico PLA, ácido poliláctico, que es un polímero termoplástico formado por moléculas de ácido láctico. Se obtiene a partir de almidón de productos vegetales (almidón de maíz, yuca o caña de azúcar) siendo biodegradable y respetando el medio ambiente.



Figura 128 Impresora 3D Creality Ender 3

#### 5.3.1. Semiesfera

Las diferentes piezas en las que se ha dividido la semiesfera se exportan a fichero .stl y se introducen en un programa de laminación como es Cura, Figura 129. La impresora realizará las piezas empleando un nozzle de 0.8mm de diámetro y altura de capa 0.4mm a 35mm/s. El tiempo de impresión de cada una de las piezas es de 8 horas aproximadamente.

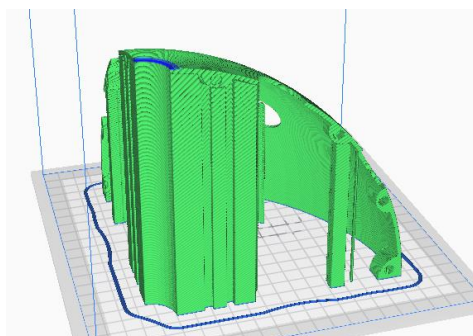
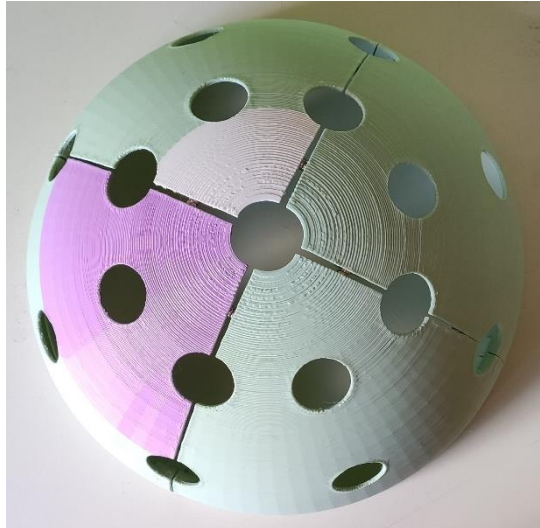


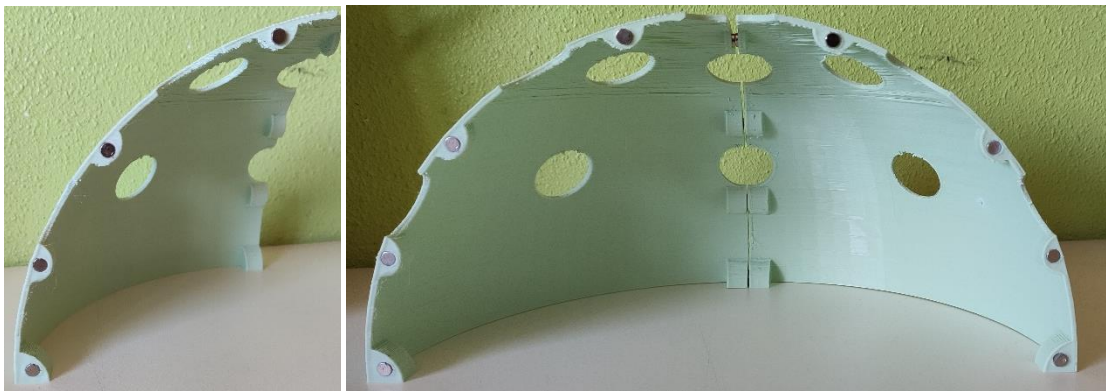
Figura 129 Laminado de una parte de la semiesfera

Con las piezas ya impresas únicamente hay que colocar los imanes, [9], en los soportes habilitados para ello, prestando atención a la polaridad con la que se colocan para que posteriormente se puedan unir todas las partes, Figura 130.



*Figura 130 Semiesfera fabricada con impresión 3D con los imanes integrados*

Los imanes son lo suficientemente fuertes y se integran 4 imanes por cada lateral de la pieza, Figura 131, para soportar el peso de la estructura, tanto si se coloca suspendido en el techo como en un lateral.



*Figura 131 Parte de la semiesfera con los imanes*

Al tratarse de un diseño modular y fácilmente montable y desmontable, es fácil de transportar, colocar, almacenar y reparar, y posee diferentes alternativas de empleo. Estas características hacen que su diseño sea adecuado para la docencia.

### 5.3.2. Cilindro

Las diferentes piezas en las que se ha dividido el cilindro se exportan a fichero .stl y se introducen en un programa de laminación como es Cura, Figura 132. La impresora realizará las piezas empleando un nozzle de 0.8mm de diámetro y altura de capa 0.4mm a 35mm/s. El tiempo de impresión de cada una de las piezas es de 4 horas aproximadamente. Es necesario imprimir 4 piezas del modelo original y otras 4 del mismo modelo, pero simétrico. Esta acción se puede realizar en el mismo programa de laminado con la herramienta espejo.

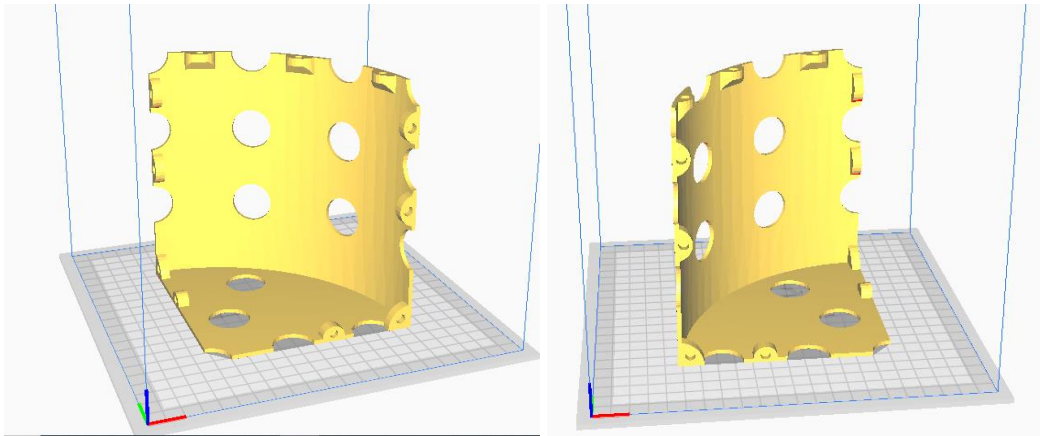


Figura 132 Laminado las partes del cilindro, original y simétrica

Con las piezas ya impresas únicamente hay que colocar los imanes en los soportes habilitados para ello, prestando atención a la polaridad con la que se colocan para que posteriormente se puedan unir todas las partes, Figura 133.

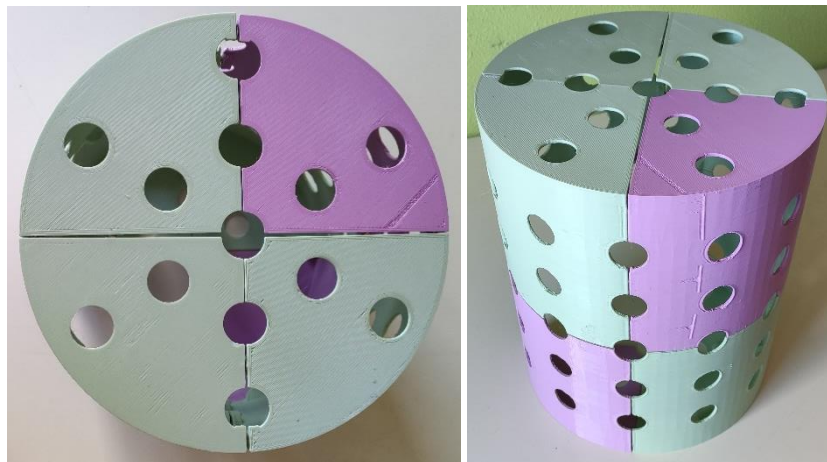


Figura 133 Cilindro fabricado con impresión 3D con los imanes integrados

Los imanes son lo suficientemente fuertes y se han integrado 4 por cada lateral y 3 en la parte superior, Figura 134, para soportar el peso de la estructura, tanto si se coloca suspendido en el techo como en un lateral.



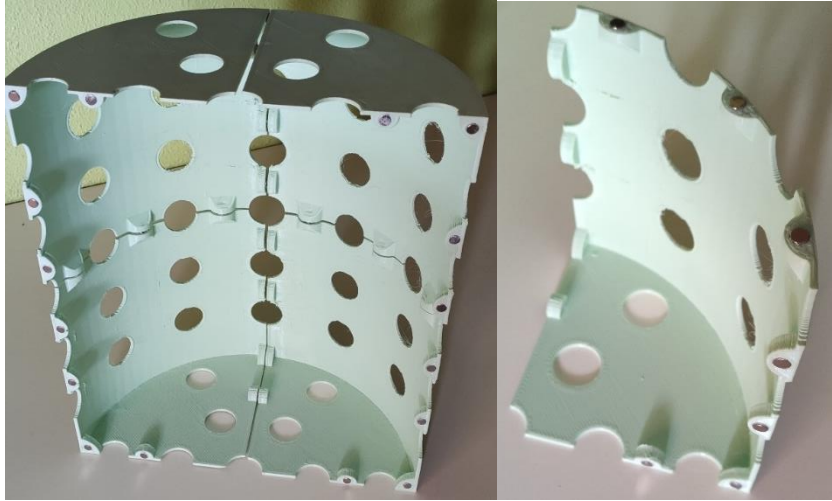


Figura 134 Parte del cilindro con los imanes

Al tratarse de un diseño modular y fácilmente montable y desmontable, es fácil de transportar, colocar, almacenar y reparar, y posee diferentes alternativas de empleo. Estas características hacen que su diseño sea adecuado para la docencia.

### 5.3.3. Herramientas y soporte herramientas

Las piezas se introducen en un programa de laminación como es Cura, Figura 135. La impresora 3D Creality Ender 3 es la impresora que realizará las piezas, empleando un nozzle de 0.8mm de diámetro y altura de capa 0.4mm a 40mm/s. El tiempo de impresión de cada una de las piezas es de 3 horas aproximadamente, la herramienta de 30cm se divide en dos partes porque no entra en el volumen de impresión y posteriormente se pegarán, Figura 136.

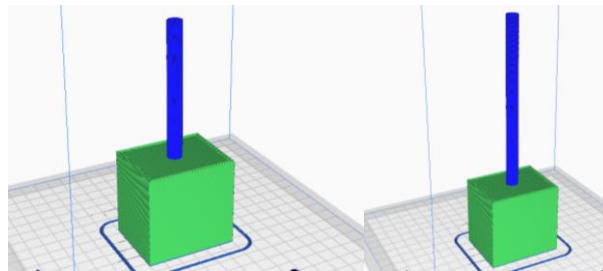


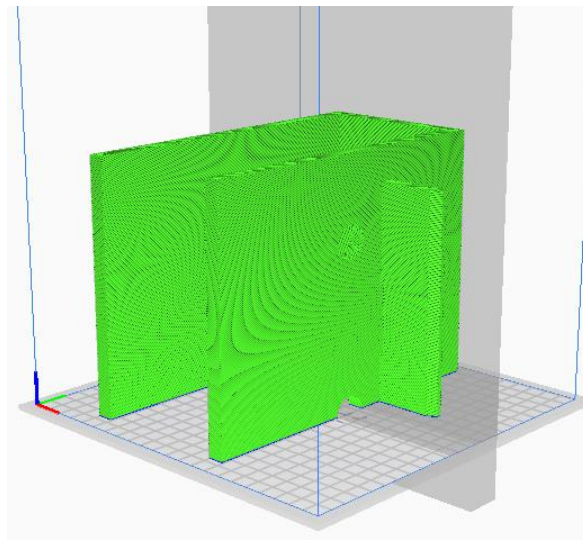
Figura 135 Laminado de las herramientas del robot



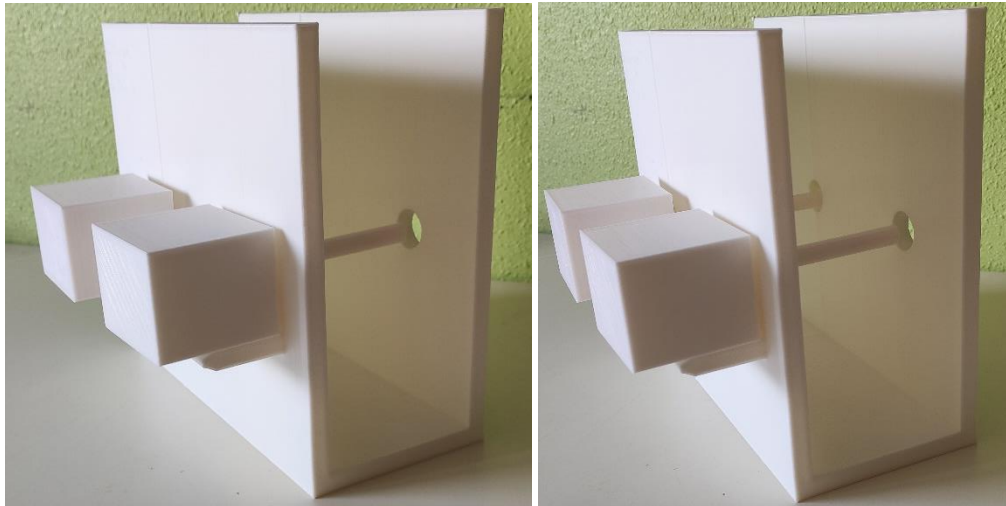
Figura 136 Herramientas de 15, 20 cm

Estas herramientas se encuentran alojadas en un soporte horizontal, Figura 16, de donde el robot las tomara para poder realizar las rutinas. Como las dimensiones del soporte exceden el volumen de impresión de la impresora 3D que se va a emplear, se divide, Figura 17, en 2 partes iguales y se pegaran juntas. Se pueden encontrar en los anexos del proyecto.

La pieza se introduce en un programa de laminación como es Cura. La impresora 3D Creality Ender 3 es la impresora que realizará las piezas, empleando un nozzle de 0.8mm de diámetro y altura de capa 0.4mm a 50mm/s. El tiempo de impresión de cada una de las piezas es de 6 horas aproximadamente.



*Figura 137 Laminado del soporte de herramientas del robot*



*Figura 138 Soporte de herramientas fabricado*

#### 5.3.4. Anclaje a la mesa y soporte de objetos

Las piezas se introducen en un programa de laminación como es Cura, Figura 139. La impresora 3D Creality Ender 3 es la impresora que realizará las piezas, empleando un nozzle de 0.8mm de diámetro y altura de capa 0.4mm a 40mm/s. El tiempo de impresión de cada una de las piezas es de 4 horas aproximadamente para el anclaje a la mesa, Figura 140.

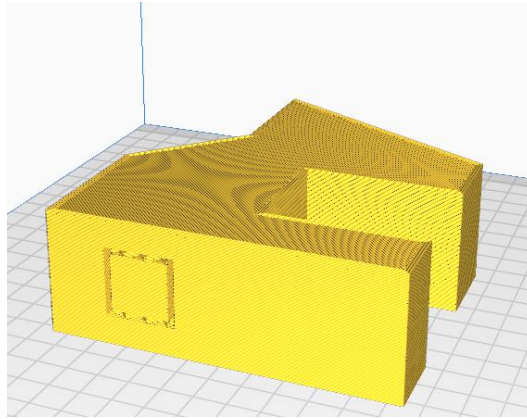


Figura 139 Laminado del anclaje a la mesa

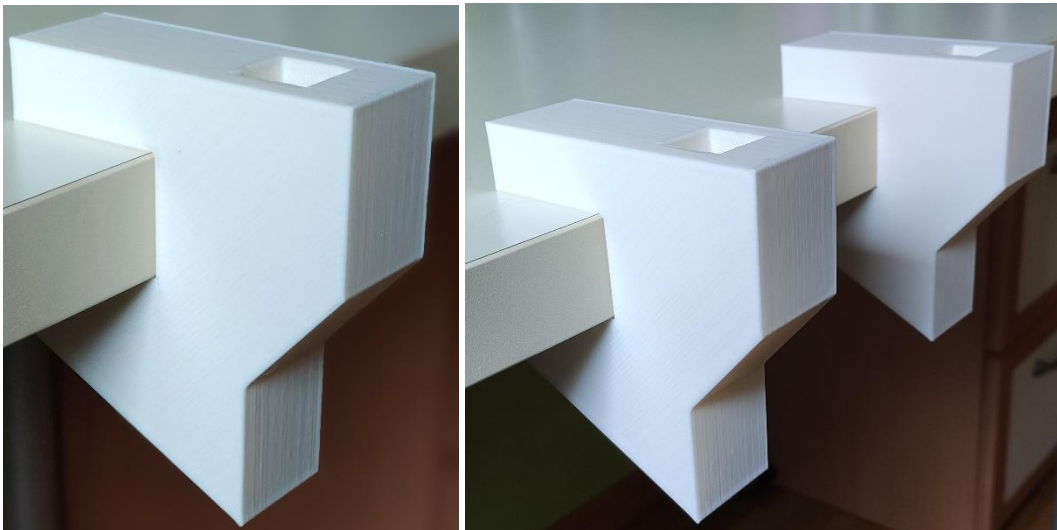
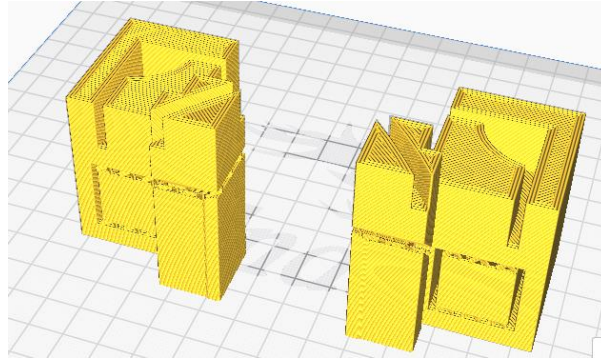


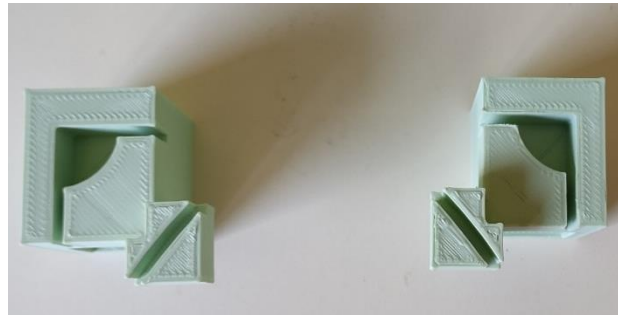
Figura 140 Anclajes a la mesa fabricados

Para el soporte de los objetos se debemos imprimir el modelo original y su simétrico, así que en el programa de laminación debemos realizar una operación de simetría para poder imprimir las piezas correctamente.

La pieza se introduce en un programa de laminación como es Cura, Figura 141. La impresora 3D Creality Ender 3 es la impresora que realizará las piezas, empleando un nozzle de 0.8mm de diámetro y altura de capa 0.4mm a 50mm/s. El tiempo de impresión de cada una de las piezas es de 3 horas aproximadamente, Figura 142.



*Figura 141 Soportes de los objetos en el programa de laminacion*



*Figura 142 Soportes de los objetos fabricados*

## CAPITULO 6. CONCLUSIONES

Con el presente TFM se ha conseguido, como se propuso al principio, el diseño de una estación robotizada para su uso en la enseñanza universitaria. La estación diseñada es controlada a través de un interfaz gráfico, que permite tanto realizar una simulación empleando Matlab como conectarse al robot real y realizar trayectorias y movimientos.

Se ha puesto especial interés en el diseño de la estación y en los objetos para que fuese flexible, permitiendo que la colocación, reparación y fabricación resultase lo más sencilla posible. Los objetos están divididos en partes iguales que se unen empleando imanes, haciendo que su montaje y almacenamiento no genere molestias al tutor o a los alumnos. Los entornos de trabajo son fáciles de montar y desmontar, ofreciendo un gran abanico de prácticas para aplicar a la docencia.

Para poder hacer pruebas con este diseño, se ha realizado un interfaz gráfico que permite tanto la simulación de la estación completa en Simulink y RobotStudio como la ejecución en el robot real.

Se ha modelado el robot real en Simulink, permitiendo realizar la simulación de la estación completa únicamente empleando Matlab. Haciendo uso de los parámetros de Denavit-Hartenberg, conociendo los valores de masa e inercia del robot y disponiendo de modelos 3D para la representación del robot en la simulación, se ha conseguido realizar un entorno de simulación completo y fiable.

Se ha modelado la estación completa en RobotStudio para su aplicación directa en el robot real. Empleando los modelos 3D diseñados y a través de comunicación TCP/IP, la simulación en RobotStudio se realiza a través de Matlab.

Con el interfaz de usuario diseñado y programado, comunicándose con Simulink y RobotStudio, se realizan una serie de trayectorias sobre los diferentes objetos en los diferentes lugares de trabajo diseñados.

La motivación que me ha llevado a realizar este TFM se basa en mi interés por la robótica, ya que es un campo de la ingeniería de suma importancia por la automatización de los procesos y su capacidad para realizar. Además, poder aplicar una tecnología como la impresión 3D en este proyecto ha aumentado mi interés por este tipo de fabricación, diseño de piezas y las aplicaciones que tiene en la actualidad y tendrá en el futuro.

La idea de este proyecto era probarlo en el robot real pero no ha podido realizarse por la situación vivida en España por el Covid-19.



## CAPITULO 7. BIBLIOGRAFÍA

- [1] M. A. M. San Joé, "*Simulación, control cinemático y dinámico de robots comerciales usando la herramienta Matlab, "Robotic Toolbox" "*", Valladolid: Escuela de Ingenierías Industriales, Universidad de Valladolid, 2014.
- [2] J. A. A. Herrero, "*Modelado de una célula robótica con fines educativos usando el programa RobotStudio*", Valladolid: Escuela de Ingenierías Industriales, Universidad de Valladolid, 2015.
- [3] A. G. d. I. Santos, "*Control remoto de una célula robotizada con fines educativos mediante tecnología WIFI*", Valladolid: Escuela de Ingenierías Industriales, Universidad de Valladolid, 2016.
- [4] C. J. Jiménez, "*Diseño de un sistema robótico educativo para jugar al ajedrez con robots industriales*", Valladolid: Escuela de Ingenierías Industriales, Universidad de Valladolid, 2019.
- [5] U. Robots, <https://www.universal-robots.com/es/productos/robot-ur3/>, Mayo, 2020.
- [6] ABB, <https://new.abb.com/products/robotics/es/robots-industriales/irb-120>, Mayo, 2020.
- [7] D. guide, <https://www.ionos.es/digitalguide/servidores/know-how/tcpip/>, Mayo, 2020.
- [8] A. d. Reder, <https://aprendederedes.com/redes/introduccion/modelo-tcp-ip/>, Mayo, 2020.
- [9] A. Wukong, [https://www.amazon.es/gp/product/B07D33PG7Y/ref=ppx\\_yo\\_dt\\_b\\_asin\\_title\\_o02\\_s00?ie=UTF8&psc=1](https://www.amazon.es/gp/product/B07D33PG7Y/ref=ppx_yo_dt_b_asin_title_o02_s00?ie=UTF8&psc=1), Mayo, 2020.
- [10] L. Merlin, <https://www.leroymerlin.es/fp/13988744/pack-6-listones-cuadrados-cepillado-2-2x200x2-2-cm>, Mayo, 2020.
- [11] J. Meyer, [https://github.com/ros-industrial/abb\\_experimental/blob/e7e6c119c23ab257cb545450b927ed6df3920dcc/abb\\_irb120\\_support/urdf/irb120\\_3\\_58\\_macro.xacro](https://github.com/ros-industrial/abb_experimental/blob/e7e6c119c23ab257cb545450b927ed6df3920dcc/abb_irb120_support/urdf/irb120_3_58_macro.xacro), Mayo, 2020.
- [12] ABB, <https://new.abb.com/products/robotics/es/robotstudio>, Mayo, 2020.
- [13] D. A. Khamis, <http://personal.biada.org/~jhorrrillo/INTRODUCCIO%20RAPID.pdf>, Mayo, 2020.







Universidad de Valladolid



ESCUELA DE INGENIERÍAS  
INDUSTRIALES

Máster en Electrónica Industrial y Automática

# MÁSTER EN ELECTRÓNICA INDUSTRIAL Y AUTOMÁTICA

ESCUELA DE INGENIERÍAS INDUSTRIALES  
UNIVERSIDAD DE VALLADOLID

TRABAJO FIN DE MÁSTER

**Diseño, simulación y fabricación de una estación  
robotizada universitaria**

Autor: D. Víctor Lobo Granado

Tutor: D. Alberto Herreros López

Valladolid, Julio, 2020