

Universidad Miguel Hernández de Elche

**MASTER UNIVERSITARIO EN
ROBÓTICA**



**"PUESTA EN MARCHA DE UN
ROBOT UR5 Y PRUEBAS DE PICK &
PLACE Y LIJADO"**

TRABAJO FIN DE MÁSTER

Enero - 2023

AUTOR: Rubén Agulló Soto

DIRECTOR: Carlos Pérez Vidal

AGRADECIMIENTOS

Ha sido un bonito trayecto hasta llegar a este momento en cual culmino el Máster de Robótica. Con la misma ilusión que en la carrera de seguir aprendiendo cosas nuevas y conocer gente con las mismas inquietudes por eso quiero agradecer a todos mis compañeros del Máster ya que sin ellos este proceso no habría sido lo mismo, a mis amigos y mi familia que siempre me han estado ayudando y dando ánimos. Quiero agradecer a la empresa CFZ Cobots por permitirme aprender de los robots colaborativos y a sus empleados que me acogieron como si fuera uno más y siempre que necesitaba ayuda estaban allí.

Por último quiero dar las gracias a Carlos Pérez por estar ahí siempre presente y ayudándome en todo lo que hiciera falta.

RESUMEN

En este trabajo vamos a utilizar un robot colaborativo como es el UR5, estos tipos de robots se pueden implantar en ambientes de trabajo junto con personas sin necesidad de ningún tipo de valla de seguridad como ocurre con los robots industriales.

En este proyecto nos centraremos en conocer los elementos básicos del robot UR5, los diferentes módulos con los que se puede realizar los programas de robot y realizaremos dos tareas con estos conceptos aprendidos que van a ser un pick & place que es una de las tareas más utilizadas en la industria y realizaremos una prueba de lijado y para ello utilizaremos un módulo de fuerza que está disponible dentro del software de UR.

El objetivo principal de este trabajo es hacer una introducción al uso de robots colaborativos como UR5 y mostrar programar tareas reales.

ABSTRACT

In this work we are going to use a collaborative robot such as the UR5, these types of robots can be implemented in work environments together with people without the need for any type of security fence as occurs with industrial robots.

In this project we will focus on knowing the basic elements of the UR5 robot, the different modules with which the robot programs can be carried out and we will carry out two tasks with these learned concepts that will be a pick & place, which is one of the most used in the industry and we will carry out a sanding test and for this we will use a force module that is available within the UR software.

The main objective of this work is to introduce the use of collaborative robots like UR5 and how to program real tasks.

ÍNDICE

1	INTRODUCCIÓN	13
1.1	Motivación	13
1.2	Objetivos	13
1.3	Organización	13
2	ESTADO DEL ARTE	14
2.1	Robots colaborativos	14
2.2	Tareas de acabado de superficie	17
3	MATERIALES	19
3.1	Robot UR5	19
3.2	Caja de control	21
3.3	Pinza SMC	22
3.4	Controlador de motor paso a paso	22
3.5	Fuente de alimentación	23
4	CONCEPTOS PREVIOS	24
4.1	PolyScope	24
4.2	Aspectos básicos del funcionamiento	27
4.2.1	Puesta en Marcha del Robot UR5	27
4.2.2	Punto Central de Herramienta	29
4.2.3	Carga útil y centro de gravedad	32
4.2.4	Configuración de entradas y salidas	33
4.2.5	Variables	34
4.2.6	Freedrive	35
4.3	Funciones básicas de programación	36
4.3.1	Movimiento	36
4.3.2	Punto de Paso	38
4.3.3	Esperar	40
4.3.4	Ajustar	40
4.3.5	Dirección	41

4.4	Funciones avanzadas de programación.....	42
4.4.1	Bucle.....	42
4.4.2	Subprograma.....	42
4.4.3	Asignación.....	43
4.4.4	If.....	43
4.4.5	Switch.....	44
5	TAREAS DEL PROYECTO.....	45
5.1	Características del pick & place.....	45
5.2	Montaje pick & place.....	46
5.3	Tarea pick & place en PolyScope.....	47
5.3.1	Inicialización de la Pinza SMC.....	47
5.3.2	Programación Pick & Place.....	49
5.4	Tarea de lijado.....	52
5.4.1	Características de la Tarea de Lijado.....	52
5.4.2	Modo Fuerza.....	53
5.4.3	Montaje Lijado.....	54
5.4.4	Programación Prueba de Lijado.....	54
6	Conclusión.....	58
6.1	Trabajos Futuros.....	59
7	BIBLIOGRAFÍA.....	60

ÍNDICE DE FIGURAS

Figura 1: El primer robot colaborativo de la historia	14
Figura 2: Robot Tech Man Robot	15
Figura 3: Cobot Saywer.....	15
Figura 4: Cobot Ned	16
Figura 5: Cobots Gama UR	16
Figura 6: Descomposición en articulaciones del UR5	19
Figura 7: Espacio de trabajo UR5	19
Figura 8: Ficha de especificaciones técnicas	20
Figura 9: Caja de Control	21
Figura 10: Pinza SMC de la serie LEHZ.....	22
Figura 11: Controlador JXC917-BC-E	23
Figura 12: Controlador JXC917-BC-E	23
Figura 13: Fuente de alimentación	23
Figura 14: Menú PolyScope	24
Figura 15: Paso 1 de la Puesta en Marcha UR5	27
Figura 16: Paso 2 de la Puesta en Marcha UR5	28
Figura 17: Menú final Puesta en Marcha UR5.....	28
Figura 18: Parada de protección	29
Figura 19: Punto Central de Herramienta (PCH).....	30
Figura 20: Medición del PCH.....	30
Figura 21: Configuración PCH.....	31
Figura 22: Cuatro Orientaciones para obtener el PCH.....	31
Figura 23: Obtención del PCH mediante cuatro orientaciones.....	32
Figura 24: Cálculo de la Carga Útil y el Centro de Gravedad.....	32
Figura 25: Menú General Configuración E/S.....	33
Figura 26: Menú E/S Variables.....	34
Figura 27: Botón de Fredrive.....	35
Figura 28: Árbol de programa.....	36
Figura 29: MoveJ.....	37
Figura 30: MoveL	37
Figura 31: MoveP	38
Figura 32: Menú Punto de Paso Fijo	38
Figura 33: Menú Punto de Paso Relativo	39
Figura 34: Menú Punto de Paso Variable.....	39
Figura 35: Menú Esperar.....	40
Figura 36: Menú Ajustar	40
Figura 37: Menú Dirección	41
Figura 38: Menú Bucle	42
Figura 39: Menú Subprograma.....	43
Figura 40: Menú Asignación.....	43
Figura 41: Menú de la función IF.....	44
Figura 42: Menú Switch.....	44
Figura 43: Montaje Pick & Place	46
Figura 44: Conexiones del Controlador JXC917-BC-E	47
Figura 45: Comando LEH init	48
Figura 46: Comando LEH Move	48
Figura 47: Pick & Place Posición Inicial	49
Figura 48: Comando Fuerza modo Simple.....	53

Figura 50: Herramienta de Lijado	54
Figura 51: PCH Herramienta de Lijado	55
Figura 52: Carga y Centro de gravedad	55

1 INTRODUCCIÓN

1.1 Motivación

La motivación de este trabajo es el poder aprender sobre los robots colaborativos como el UR5 y su programación que permite realizar cualquier tipo de tarea que se nos pueda imaginar.

1.2 Objetivos

Los robots colaborativos cada vez más se encuentran entre nosotros ya sea como robots de asistencia, robots camareros... .

El objetivo de este proyecto es el de mostrar cómo se comienza a trabajar con los robots colaborativos desde 0 y más concretamente con el robot UR5 que presenta un gran número de opciones a la hora de programar tareas como las que vamos a realizar pick & place y una prueba de lijado.

1.3 Organización

Este trabajo se dividirá en 5 capítulos:

- Capítulo 1: Introducción donde se incluye la motivación para realizar este proyecto y los objetivos que se pretenden alcanzar con la realización de este trabajo.
- Capítulo 2: Estado del arte en este apartado se hace un repaso de cómo han evolucionado los robots colaborativos hasta la actualidad y se hace una revisión sobre trabajos que incluyen robots colaborativos para tareas de lijado.
- Capítulo 3: Materiales en este apartado mostramos los distintos materiales que vamos a utilizar durante el proyecto haciendo una breve explicación del uso de cada uno de ellos.
- Capítulo 4: Conceptos previos en este capítulo se explicarán los conceptos básicos necesarios para comenzar con la programación de las tareas.
- Capítulo 5: Tareas del proyecto en este capítulo se comentarán las tareas que vamos a realizar así como se han realizado su programación.
- Capítulo 6: Conclusión en este capítulo se hará una conclusión del proyecto realizado y se expondrán los proyectos futuros.

2 ESTADO DEL ARTE

2.1 Robots colaborativos

El primer robot colaborativo fue construido por Peshkin y Colgate en el año 1996 y su tarea principal era la de asistir en la elevación de cajas. Durante la década de los 90 muchas empresas ya se habían automatizado algunas tareas en la línea de producción sin embargo había ciertas tareas como la de levantar cajas que no estaba automatizada y suponía una gran carga de trabajo y numerosas lesiones para los operarios [1].

Como las empresas no sabían cómo solucionar este problema se pusieron en contacto con la Universidad de Northwestern y fueron Michael Peshkin y J.Edward Colgate quienes pusieron fin a todo estos problemas construyendo el primer robot colaborativo de la historia (*Figura 1*).



Figura 1: El primer robot colaborativo de la historia

Se construyó un robot capaz de trabajar junto con los operarios sin ningún tipo de peligro para ellos ya que estaban configurados de forma que solo pudieran actuar mediante el impulso de los trabajadores. Estos robots se movían mediante sensores táctiles y hápticos por lo que les permitía guiarlos de forma directa por el operador.

Peshkin y Colgate crearon un sistema de cabrestante inteligente el cual al detectar la intención de movimiento levantaba la carga por lo que los trabajadores solo necesitaban hacer un pequeño esfuerzo para levantar una carga que antes le podría ocasionar problemas físicos al moverla repetidamente.

Los primeros cobots (robots colaborativos) disponían de sistemas de apagado automático cuando se detectaba el contacto humano. Otra cosa que hicieron fue ajustar la velocidad y fuerza del robot según la tarea que iban a realizar para evitar lesiones. También idearon una serie de vallas virtuales para limitar el

espacio de la tarea al entorno del robot.

Gracias a todas estas mejoras hizo que se implantarán de forma exitosa los cobots en todo el mundo y que hoy en día se sigan usando en todas las industrias.

En esta última década los robots colaborativos han mejorado mucho y se han fabricado modelos de última generación como los siguientes:

En la *Figura 2* se muestra el cobot de Tech Man Robot el cual está equipado con una cámara la cual es capaz de reconocer las formas y los bordes de los objetos y otros tipos de patrones.



Figura 2: Robot Tech Man Robot

El cobot Saywer de RethinkRobotics que se muestra en la *Figura 3* es característico por este diseño similar al humano y presenta una interfaz de fácil uso lo que nos permitirá realizar la programación de forma intuitiva. Está equipado de un sensor de fuerza de alta precisión para realizar movimientos con gran precisión.

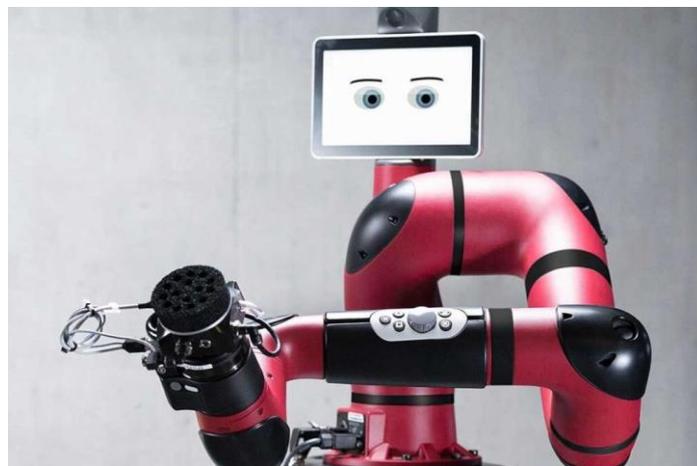


Figura 3: Cobot Saywer

El cobot Ned de la compañía Nyrío se puede controlar mediante comandos de control de movimiento y voz. También reconocimiento de imagen para procesar objetos de forma más exacta (*Figura 4*).



Figura 4: Cobot Ned

Por último comentar los robots colaborativos de la gama UR los cuales tienen la capacidad de manejar distintas cargas desde 3, 5, 10, 16 y 20 kg de izquierda a derecha en la *Figura 5*.

Estos cobots tienen seis grados de libertad, gran flexibilidad a la hora de realizar tareas y son de fácil integración en entornos de trabajo gracias a la facilidad de su puesta en marcha y su programación [2].



Figura 5: Cobots Gama UR

2.2 Tareas de acabado de superficie

En esta parte vamos a hacer una revisión de los trabajos más importantes en los cuales utilizan robots para tareas de lijado, pulido y esmerilado. Esta revisión la vamos a dividir en dos, por una parte los trabajos en los que la operación se realizará de forma automática por el robot sin interacción con el operador y por otro lado los trabajos en los que hay una interacción entre el robot y el operador.

En primer lugar vamos a tratar los trabajos en los que el robot realiza la tarea de forma automática:

- En el trabajo de Villagrosi et al. (2017) en el efector final del robot se instala un sensor laser el cual va a permitir escanear la pieza con la que se va a trabajar para generar trayectorias que serán seguidas por el robot para realizar una operación de desbarbado [3].
- En relación con el trabajo anterior tenemos el de Diao et al. (2018) que incorpora un sistema de visión montado sobre el robot para posicionar la herramienta sobre la pieza y poder realizar la operación de lijado [4].
- En Mohsin et al. (2017,2019) realizan la operación de pulido mediante un algoritmo que crea un recorrido para la herramienta en el cual se va a controlar la fuerza con la cual un brazo robótico va a realizar la tarea de pulido [5].
- En el caso de Oba and Kakinuma (2017) mediante un robot se va a realizar la tarea de pulido a partir de una serie de datos que provienen de un sensor de fuerza que permite adaptar la herramienta a la superficie y realizar la presión correcta.[6]
- En el trabajo de Sagreto and Teti (2019) se utiliza un algoritmo de machine learning basado en una serie de datos que se obtienen por unos sensores para determinar las posiciones que debe tomar el robot para realizar el proceso de pulido [7].
- En Kharidege et al. (2017) se crea un camino mediante una aplicación de path planning que está creada con datos CAD de la pieza con la que se va a trabajar. Sin embargo los puntos de referencia se toman de forma offline (antes de realizar la tarea) y no se realiza ningún control de la fuerza empleada mientras se está puliendo la pieza [8].

Ahora vamos a comentar los trabajos en los que el operador y el robot colaboran para realizar las operaciones de acabado de superficies:

- En primer lugar comentar el trabajo de Ochoa and Cortesao (2019), en este trabajo el operador guiará a mano al robot por los puntos que quiera realizar el pulido y se registrarán tanto las posiciones como las fuerzas empleadas. Después el robot será capaz de seguir el trayecto configurado por el operario anteriormente pero al estar programada la trayectoria no es capaz de adaptarse a cambios de posición, forma, tamaño, etc [9].
- En el caso de Gaz et al. (2018) en este caso el robot sujeta la pieza y es el operador el que con una herramienta de lijado, si quiere cambiar la orientación de la pieza lo hará presionando sobre el robot que mediante un sensor de fuerza que va conectado a la muñeca de este. Sin embargo al ser el operario el que tiene que ajustar la herramienta a la pieza podrían surgir problemas de mantener la ortogonalidad entre la superficie de lijado y la herramienta [10].
- En el trabajo Gracia et al. 2019 se tiene en cuenta tanto la fuerza de contacto como la ortogonalidad entre la superficie de trabajo y la herramienta del robot. Para ello instalaron un sensor fuerza/torque en el efector final del robot y además se instaló otro sensor en la herramienta del robot para que el operador pueda guiar la herramienta de forma óptima por la superficie de trabajo [11].

3 MATERIALES

3.1 Robot UR5

El robot colaborativo con el que vamos a trabajar es el UR5 de Universal Robots cuya carga útil es de 5 kg. Tiene seis grados de libertad y además está formado por 6 articulaciones: base, hombro, codo y muñeca 1, 2 y 3 (*Figura 6*).

La base es el punto en el que se monta el robot y en la brida de herramienta es donde se acoplan las herramientas.

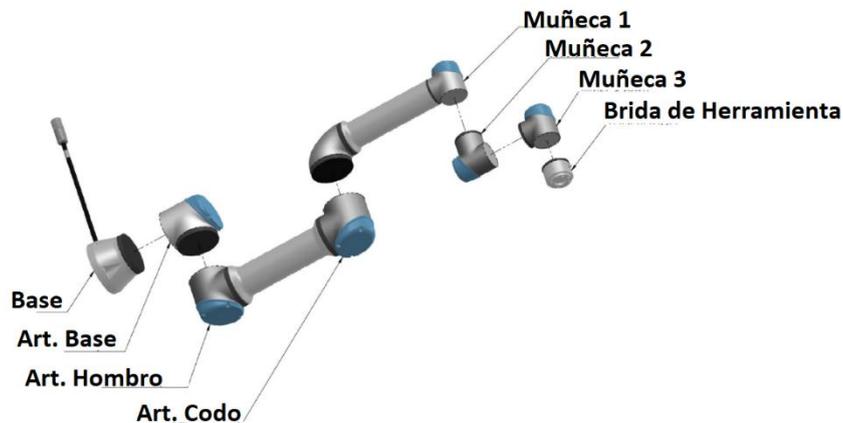


Figura 6: Descomposición en articulaciones del UR5

El espacio de trabajo del robot UR5 es una esfera de radio 850 mm desde la junta de base aunque hay un volumen cilíndrico arriba y debajo de la base que deberemos evitar ya que el robot trabaja de forma ineficiente (*Figura 7*).

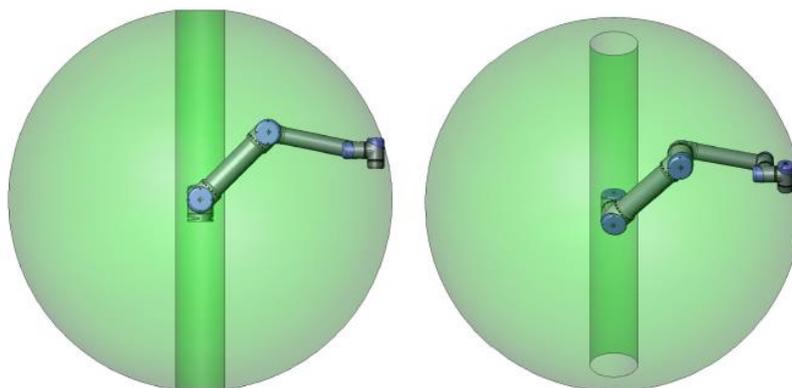


Figura 7: Espacio de trabajo UR5

En la siguiente *Figura 8* mostraremos las especificaciones técnicas del robot UR5:

UR5 Especificaciones técnicas

No. Artículo 110105

Brazo robótico de 6 ejes con un radio de funcionamiento de 850 mm

Peso:	18,4 kg
Carga:	5 kg
Alcance:	850 mm
Rango giro articulaciones:	+/- 360°
Velocidad:	Articulación: Máx. 180°/s. Herramienta: Típico 1 m/s
Repetibilidad:	+/- 0,1 mm
Espacio necesario:	Ø149 mm
Grados de libertad:	6 articulaciones giratorias
Tamaño de la caja de control: (ancho x alto x largo)	475 mm x 423 mm x 268 mm
Puertos de E/S:	
	Caja de control Conector herramienta
	Entradas digitales 16 2
	Salidas digitales 16 2
	Entradas analógicas 2 2
	Salidas analógicas 2 -
Fuente de alimentación de E/S:	24 V 2A en caja de control y 12 V/24 V 600mA en herramienta
Comunicación:	TCP/IP 100 Mbit: IEEE 802.3u, 100BASE-TX Ethernet socket & Modbus TCP
Programación:	Interfaz gráfica de usuario PolyScope en pantalla táctil de 12" con soporte
Ruido:	Relativamente silencioso
Clasificación IP:	IP54
Consumo de energía:	Aprox. 200W utilizando un programa típico
Funcionamiento colaborativo:	15 Funciones de seguridad avanzadas Operaciones colaborativas de acuerdo con: EN ISO 13849:2008 PL d EN ISO 10218-1:2011, Cláusula 5.4.3
Materiales:	Aluminio, plástico PP
Temperatura:	El robot puede funcionar en un intervalo de temperaturas de 0-50°C
Fuente de alimentación:	100-240 VAC, 50-60 Hz
Cables:	Cable entre el robot y la caja de control (6 m) Cable entre la pantalla táctil y la caja de control (4,5 m)



ALCANZAR

850 mm / 33,5 pulgadas



CARGA ÚTIL

5 kg / 11 libras



HUELLA

Ø 149 mm



PESO

20,6 kg / 45,4 libras

Figura 8: Ficha de especificaciones técnicas

3.2 Caja de control

En la caja de control o también llamada controladora vamos a encontrar las entradas y salidas tanto digitales como analógicas y también diferentes conexiones eléctricas para la seguridad y el sistema de encendido del robot.

Los módulos de la interfaz eléctrica los podemos dividir (*Figura 9*):

- A. Es la parte de seguridad donde encontramos la parada de emergencia y la parada de protección.
- B. Encendido del robot.
- C. Conexión para el seguimiento de un encoder.
- D. Conexiones para configuración de señales de entrada y salida.
- E. Conexión para señales digitales (8 de entrada y 8 de salida).
- F. Conexión para señales digitales (8 de entrada y 8 de salida).

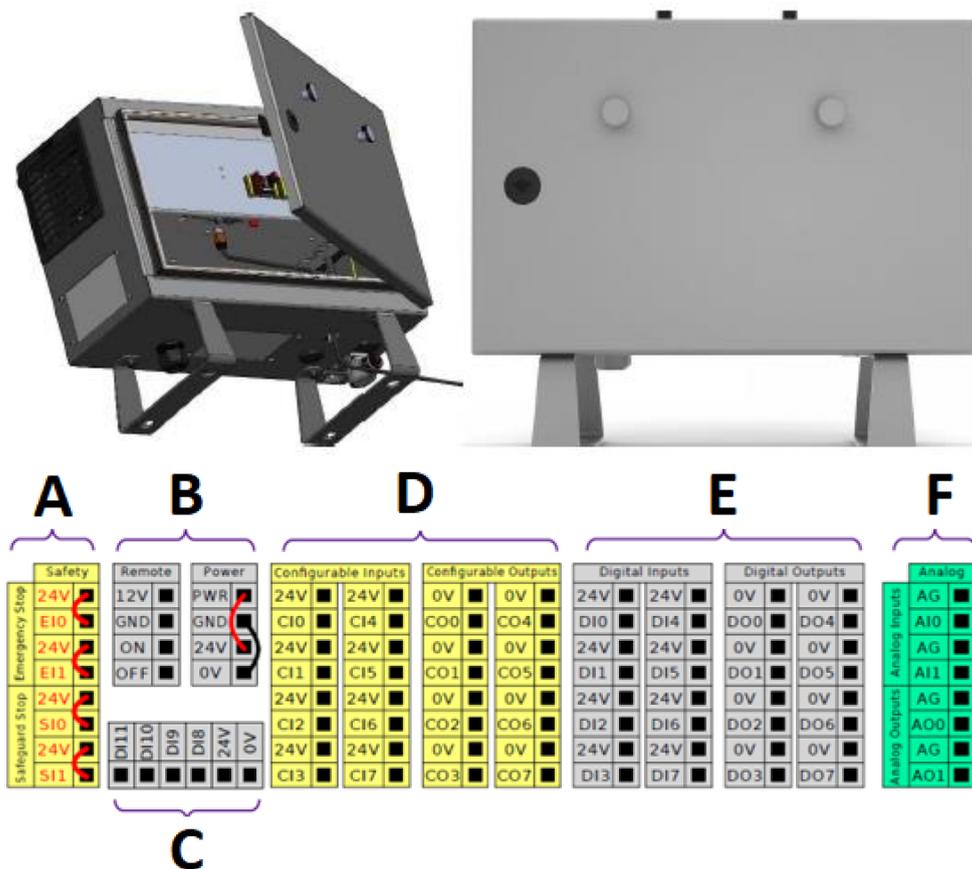


Figura 9: Caja de Control

3.3 Pinza SMC

El modelo de pinza SMC que vamos a emplear en este proyecto será la serie LEHZ10 ,el número 10 nos indica el tamaño. En la *Figura 10* podemos ver que para un tamaño de 10 la carrera sería de 4 mm y la fuerza de amarre que será de 6 a 14 N.

Serie LEHZ



Tamaño	Carrera / ambos lados [mm]	Fuerza de amarre [N]	
		Básico	Compacto
10	4	6 a 14	2 a 6
16	6		3 a 8
20	10	16 a 40	11 a 28
25	14		
32	22	52 a 130	—
40	30	84 a 210	—

Figura 10: Pinza SMC de la serie LEHZ

3.4 Controlador de motor paso a paso

Para conectar la pinza a nuestro UR5 deberemos utilizar un controlador de motor paso a paso que deberemos alimentar con una fuente de alimentación de 24V. El controlador de motor paso a paso será el modelo JXC917-BC-E del fabricante SMC (Figura 11) el cual se comunica con el robot por Ethernet/IP y se fija mediante tornillos [12].

Ahora vamos a detallar las diferentes partes del controlador que se observan en la *Figura 12*:

1. Led que indica el estado del controlador.
2. Se trata de conectores para realizar la comunicación Ethernet/IP.
3. Se trata de conmutadores para establecer la dirección IP.
4. Conector para la teaching box (LEC-T1).
5. Conector para el actuador (Conector encoder 16 polos).
6. Conector para el actuador (Conector para motor 6 polos).
7. Conector para la alimentación de 24VDC.
8. Etiqueta donde indica el número de actuador que puede ir conectado al controlador.
9. Etiqueta que indica el número de controlador (*Figura 11*).
10. Donde se colocan los tornillos para montar el controlador.



Figura 11: Controlador JXC917-BC-E

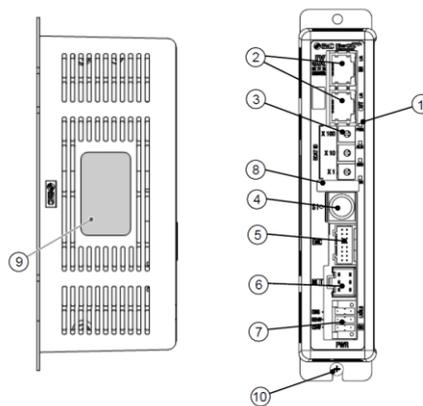


Figura 12: Controlador JXC917-BC-E

3.5 Fuente de alimentación

Para alimentar el controlador que hemos visto en el apartado anterior necesitaremos una fuente de alimentación de 24 V en continua. Para ello se hizo uso de la fuente de alimentación de Tektronix (Figura 13).

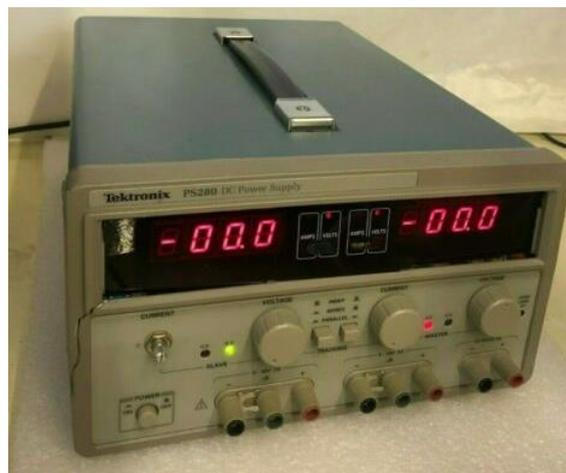


Figura 13: Fuente de alimentación

4 CONCEPTOS PREVIOS

4.1 PolyScope

En primer lugar hay que mostrar la interfaz de usuario de robot conocida con el nombre de PolyScope. Se trata de la interfaz gráfica de usuario que se muestra en una pantalla táctil en la cual el usuario podrá manejar el brazo robótico, la controladora y se podrá crear y ejecutar programas del robot.

PolyScope se compone de 3 apartados (*Figura 14*):

- A. **Encabezado:** Donde se encuentran las pestañas/iconos para realizar diferentes acciones con respecto a la manipulación del robot como a la creación de programas.
- B. **Pie de página:** En el cual hay una serie de botones para controlar los programas cargados.
- C. **Pantalla:** En esta aparecen una serie de campos para controlar y supervisar las acciones de robot.

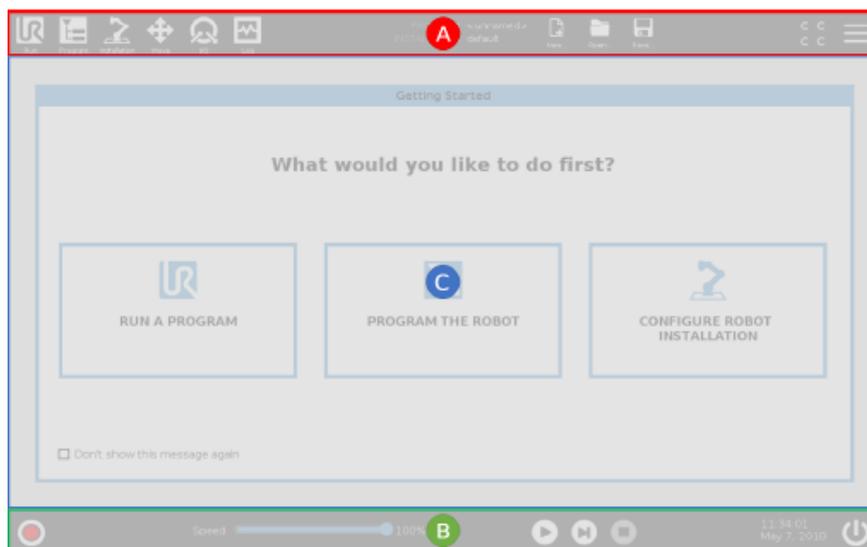


Figura 14: Menú PolyScope

En el **Encabezado** se encuentran los siguientes iconos:



Ejecutar: mediante este icono podremos controlar el robot mediante los programas ya preparados.



Programa: mediante este icono podremos crear o modificar programas de robot.



Instalación: mediante este icono se podrá configurar parámetros de ajustes del brazo robótico y el equipo externo (montaje, seguridad ...).



Mover: Podremos controlar o regular el movimiento del robot.



E/S: mediante este icono se podrá supervisar y ajustar tanto las señales de Entrada como las de Salida que van hacia y desde la caja de control del robot (la controladora).



Registro: mediante este icono se podrá conocer el estado del robot así como cualquier mensaje de advertencia o error.

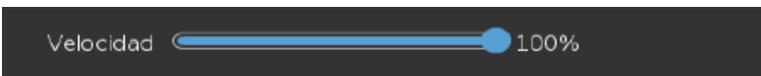


Opciones de Programa: El primer icono nos permitirá crear un programa nuevo, el segundo abrir un programa que ya tengamos creado y el último nos permitirá guardar el programa.

En el **pie de página** de la página encontramos los siguientes iconos:



Inicializar: gestiona el estado del robot. Según el estado del robot el icono cambiará de color: verde (normal), amarillo(inactivo) y rojo (parado).



Control deslizante de velocidad: Nos permite saber en tiempo real la velocidad relativa a la que se mueve el brazo robótico.



Reproducir: Para iniciar el programa de robot.



Paso: Permite ejecutar el programa con una única etapa.



Parar: Detiene la ejecución del programa de robot que se está ejecutando.



Apagado: Permite apagar o reiniciar el robot.

En la parte de **pantalla** hay tres opciones:



Ejecutar un programa: en esta opción ejecutaremos un programa ya creado.



Programar el robot: para realizar programas nuevos.



Configuración de la instalación del robot: para configurar parámetros del robot durante la etapa de instalación.

4.2 Aspectos básicos del funcionamiento

4.2.1 Puesta en Marcha del Robot UR5

En primer lugar habrá que encender la consola de programación mediante el botón de encendido situado a la derecha de la seta de emergencia roja. Después para encender el robot habrá que pulsar el botón rojo de la esquina inferior izquierda.

Después habrá que pulsar el botón de ENCENDER para transmitir energía a las articulaciones (*Figura 15*).

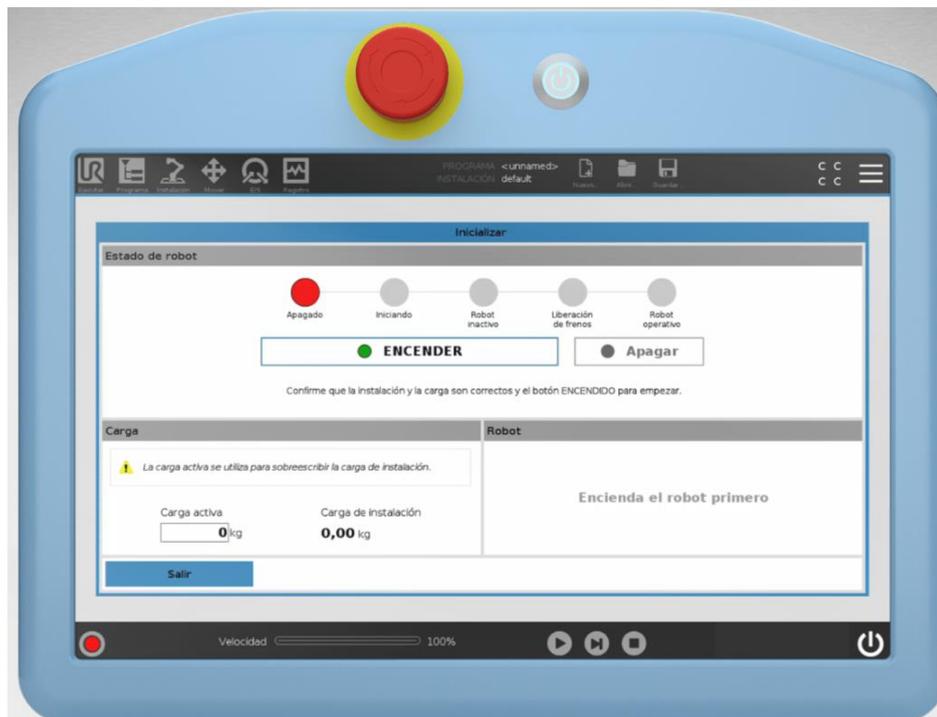


Figura 15: Paso 1 de la Puesta en Marcha UR5

Después pulsaremos el botón de INICIAR para liberar los frenos neumáticos (Figura 16).

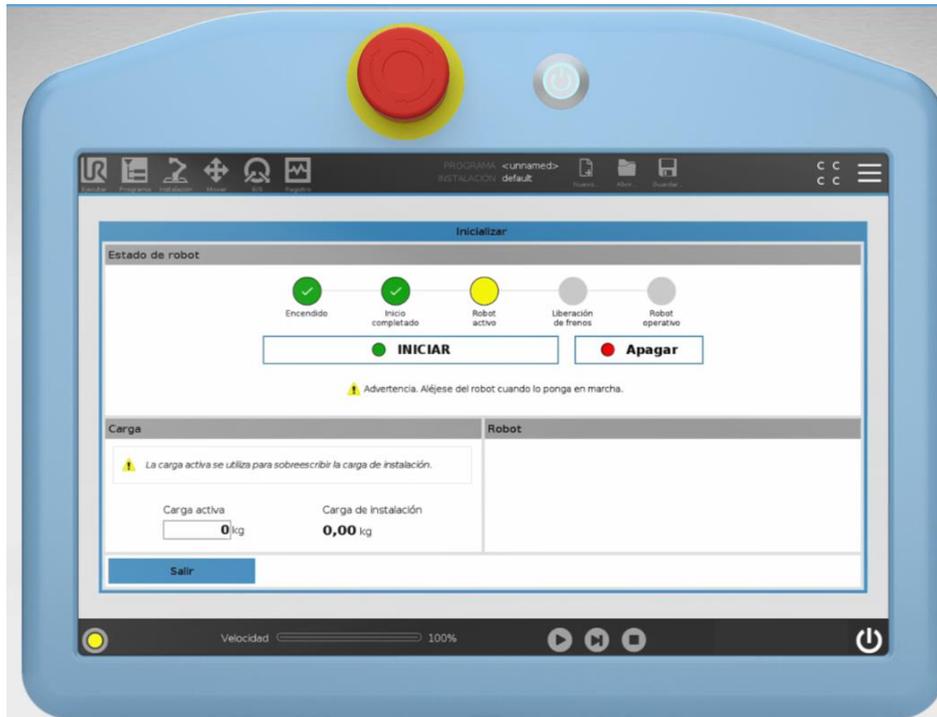


Figura 16: Paso 2 de la Puesta en Marcha UR5

Una vez se liberen los frenos neumáticos el robot entra en modo normal en el que ya podremos ejecutar y crear programas Figura 17.

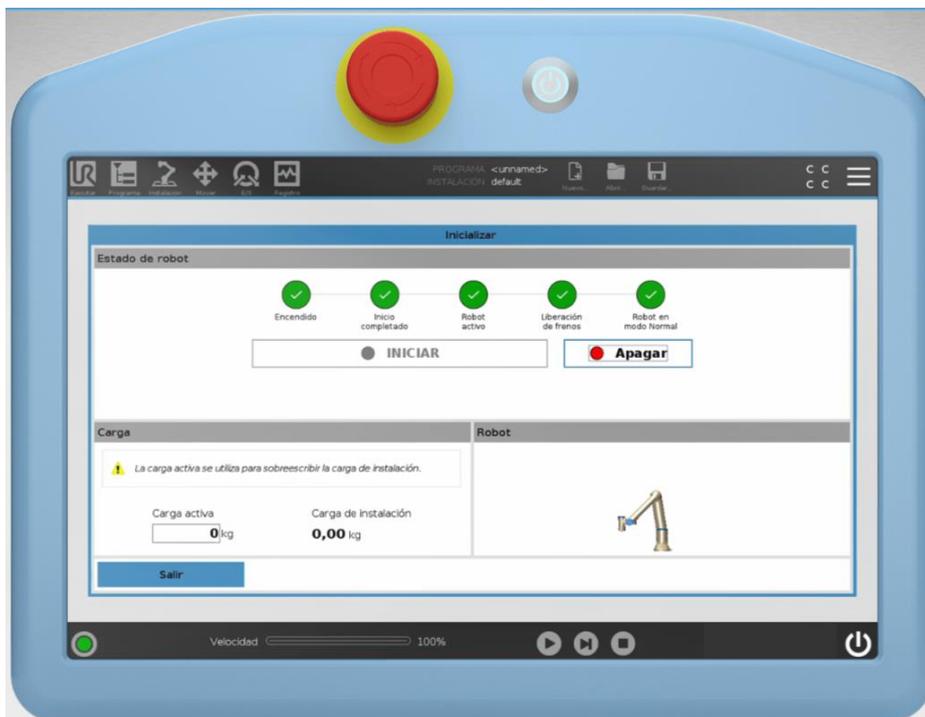


Figura 17: Menú final Puesta en Marcha UR5

Para terminar el apartado de puesta en marcha quería comentar una de las funciones más importantes con las que cuenta el robot y es la función de seguridad que se realiza de forma automática cuando el robot se alcanza un límite articular o se encuentra con algún obstáculo. Para volver a poner el robot en funcionamiento habrá que asegurarse que no hay ningún obstáculo o personas que se interpongan en el movimiento, una vez hecho esto habrá que pulsar el botón habilitar el robot (*Figura 18*).

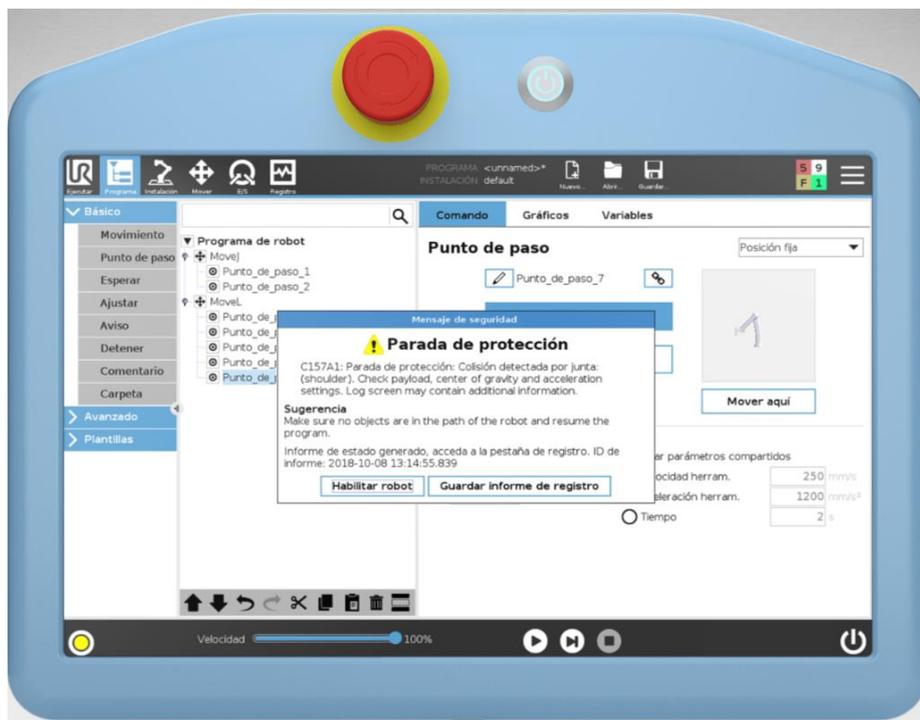


Figura 18: Parada de protección

4.2.2 Punto Central de Herramienta

En primer lugar habrá que definir que es el Punto Central de Herramienta a partir de ahora PCH, también conocido como TCP (Tool Center Point).

El PCH es el punto utilizado para el posicionamiento del robot en el espacio cartesiano. Es importante obtener de manera correcta el PCH sino el robot no se moverá como nosotros queremos.

El PCH se encuentra entre los puntos de contacto de la herramienta como se observa en la siguiente *Figura 19*:

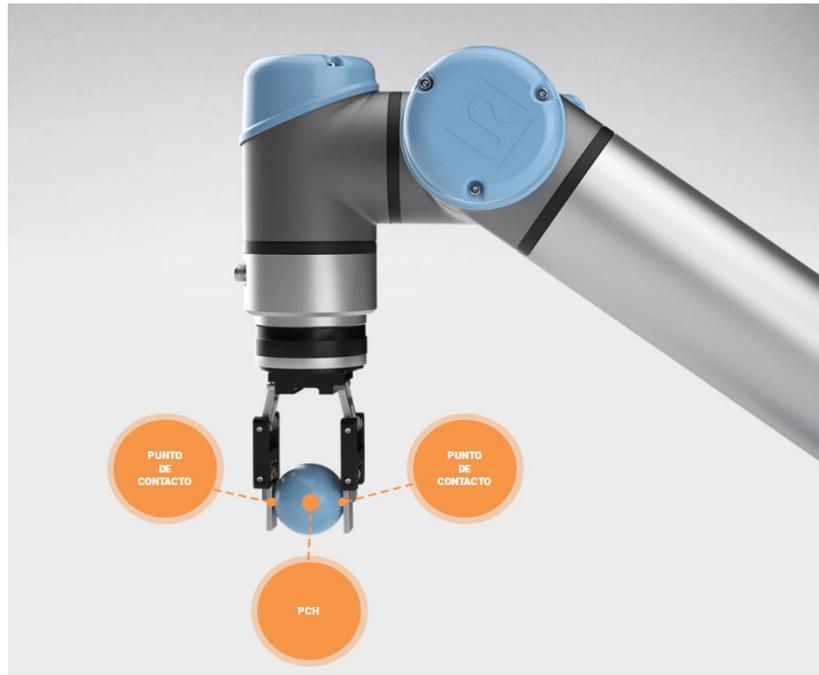


Figura 19: Punto Central de Herramienta (PCH)

Ahora que ya conocemos donde se encuentra el PCH deberemos configurarlo en PolyScope.

En este caso como el PCH está ubicado directamente sobre el eje Z y no se encuentra desplazado de los ejes X e Y podremos medir directamente la distancia desde el PCH hasta la brida de herramienta como se muestra en la siguiente *Figura 20*:

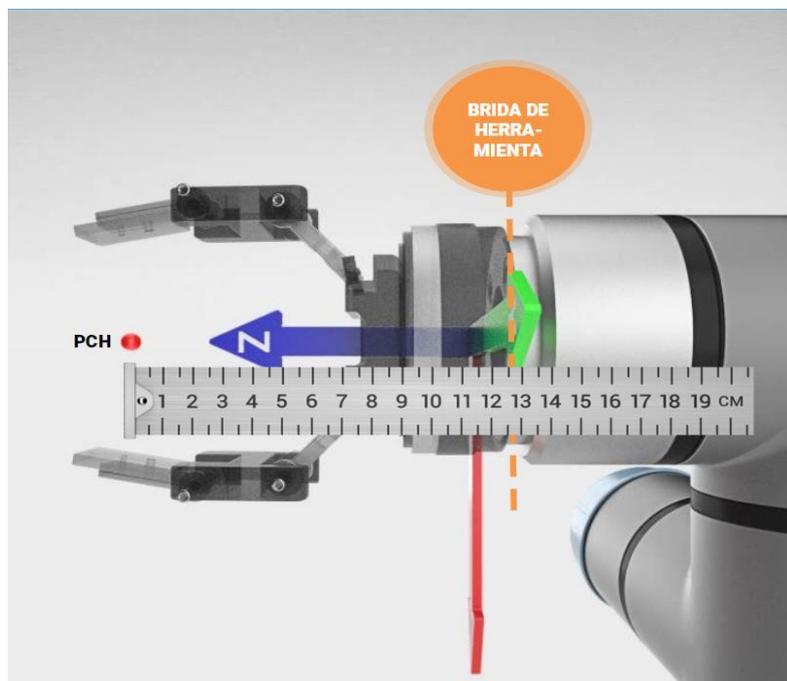


Figura 20: Medición del PCH

Ahora iremos al menú de Instalación y donde pone PCH introduciremos la distancia que en este caso concreto es de 125mm en la coordenada Z.

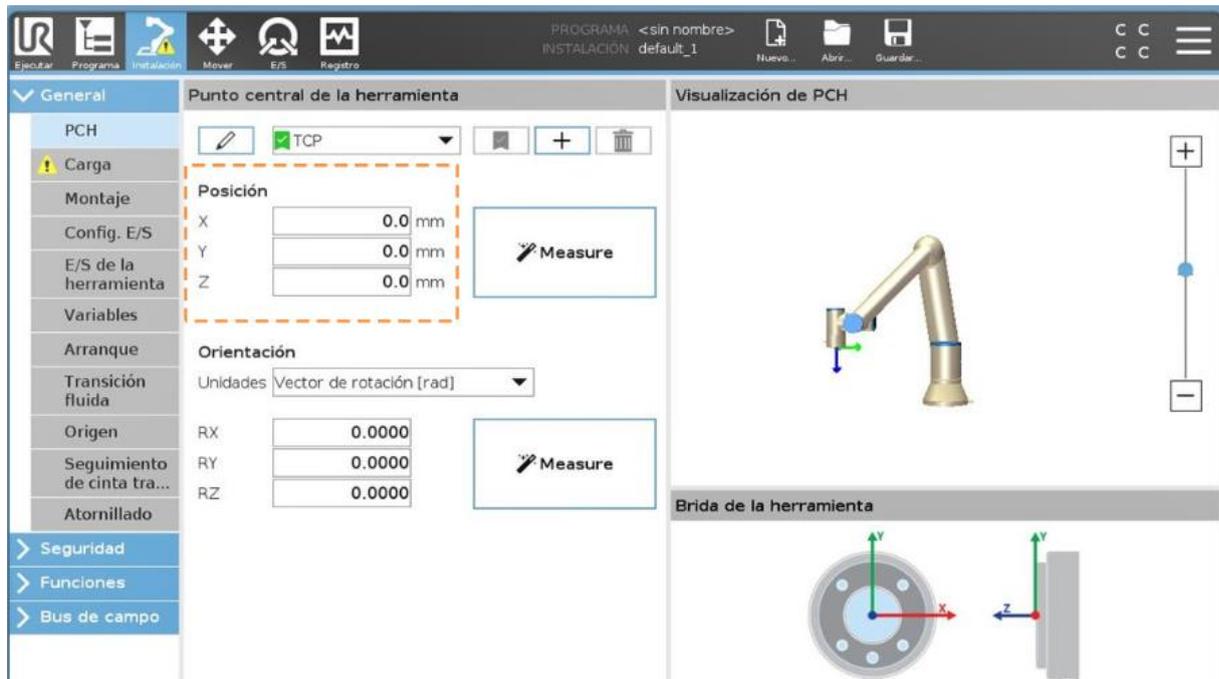


Figura 21: Configuración PCH

En los casos en los que el PCH este desplazado de la brida de herramienta ya no será tan fácil como ponerse a medir por lo que utilizaremos una función que está disponible para obtener el PCH a través de establecer cuatro puntos con diferentes orientaciones (Figura 22).

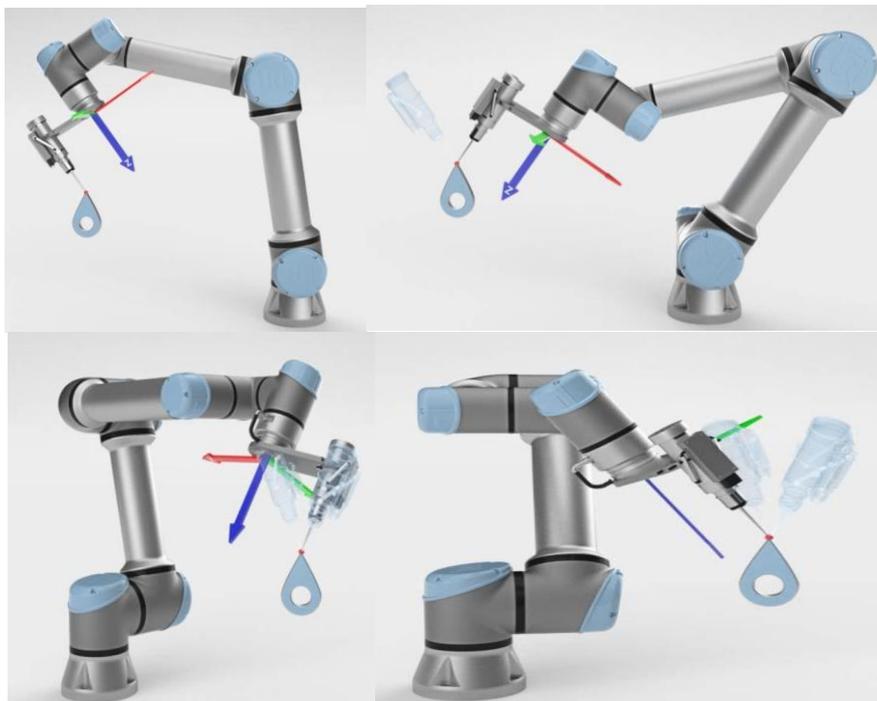


Figura 22: Cuatro Orientaciones para obtener el PCH

Y una vez tenemos configuradas las cuatro orientaciones deberemos pulsar el botón establecer (*Figura 23*).

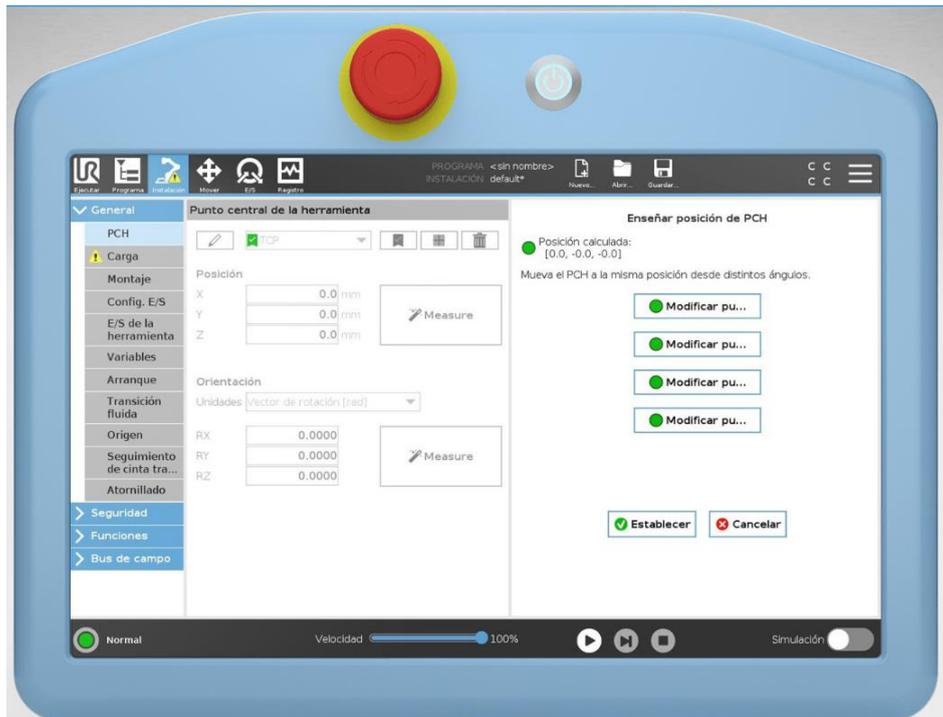


Figura 23: Obtención del PCH mediante cuatro orientaciones

4.2.3 Carga útil y centro de gravedad

La carga útil y centro de gravedad o también conocida como payload en inglés es el peso que va a manejar el robot en el efector final. Cuando no conoces el peso de esa carga la podremos estimar mediante el robot siguiendo un procedimiento similar al procedimiento seguido para calcular el PCH. La gran diferencia es que cuando vamos a situar el robot en las cuatro orientaciones no debemos tocar la herramienta o la brida de herramienta porque eso podría alterar a los cálculos para ello moveremos el robot por la muñeca (*Figura 24*).

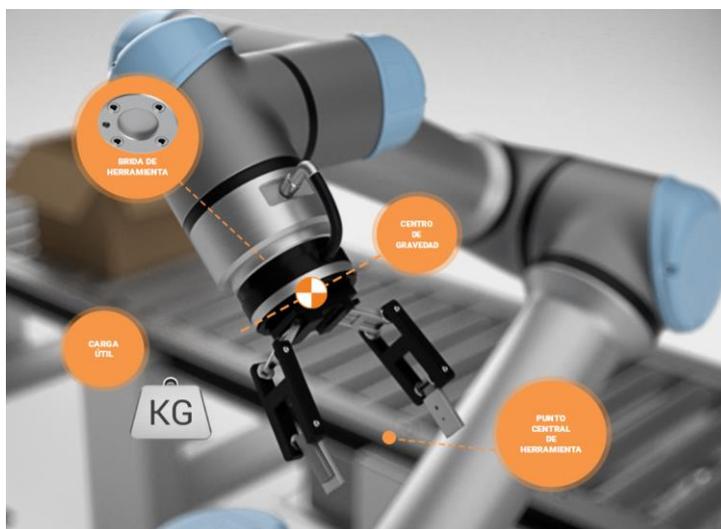


Figura 24: Cálculo de la Carga Útil y el Centro de Gravedad

4.2.4 Configuración de entradas y salidas

Un programa de robot básico consiste en el manejo de movimiento y señales las cuales deberemos configurar para poder usarlas a la hora de realizar la programación de nuestra tarea.

Para nombrar las señales y que sean fácilmente reconocibles por el usuario deberemos ir al menú que se muestra en la *Figura 25* y ahí renombrar tanto las entradas digitales como las salidas digitales.

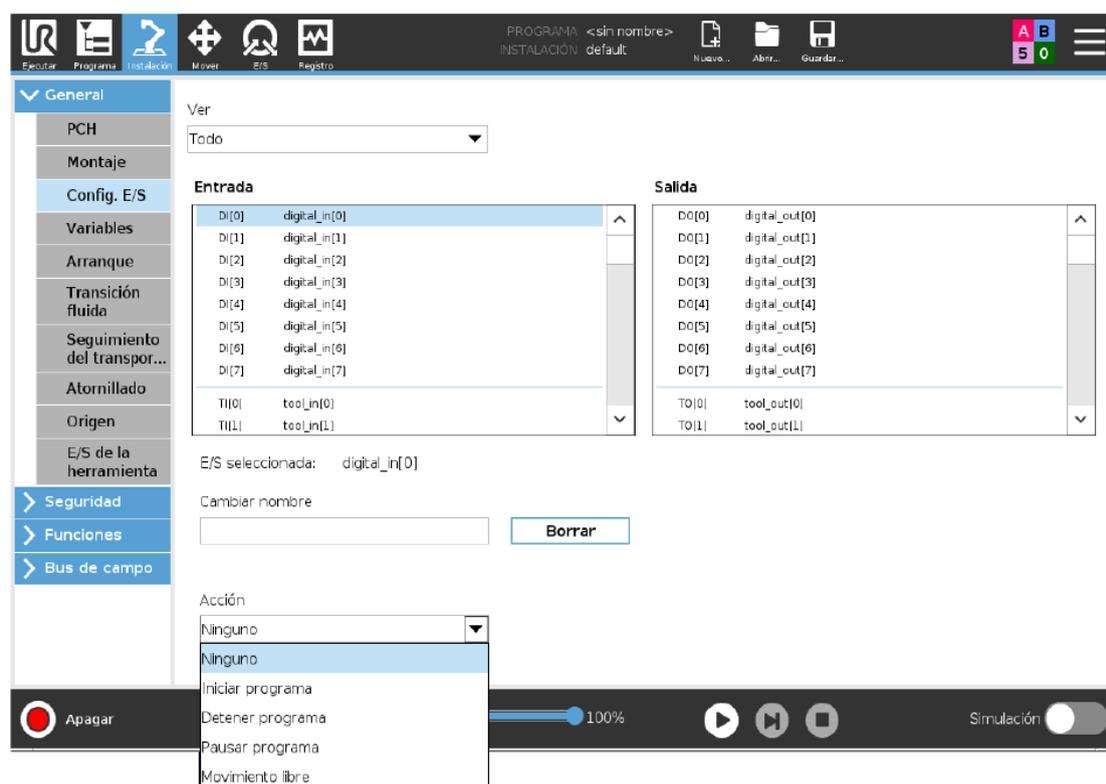


Figura 25: Menú General Configuración E/S

Si queremos ver como se activan las entradas o salidas digitales podremos ir al Menú de E/S en la barra superior de PolyScope (*Figura 26*) en el cual podremos activar las entradas y salidas de forma manual y observar los valores de tensión y corriente a los que se activan.

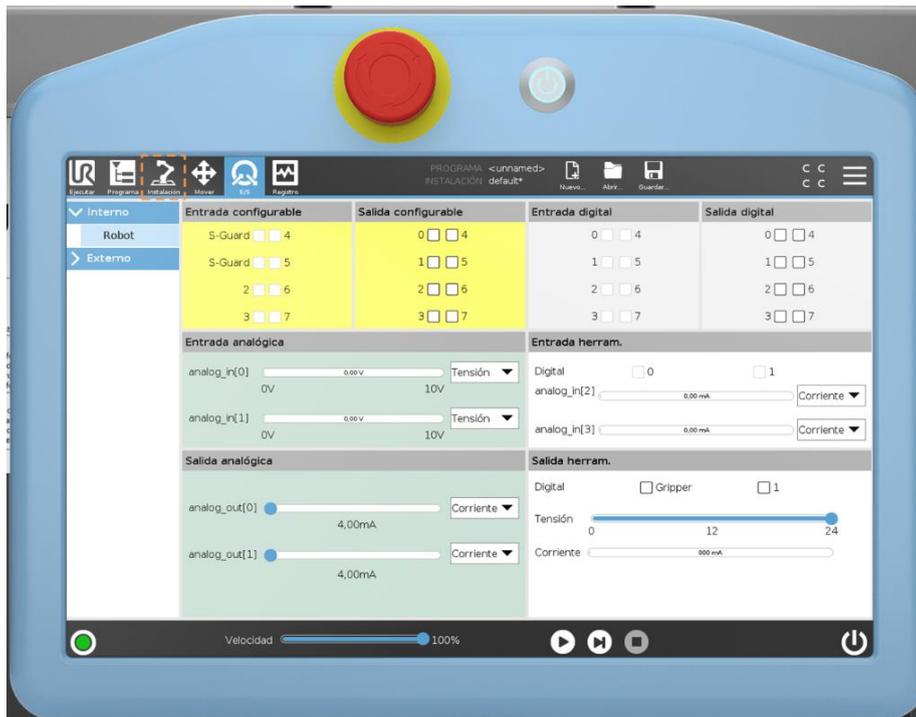


Figura 26: Menú E/S Variables

4.2.5 Variables

Las variables se utilizan para guardar y actualizar valores mientras se ejecuta el programa de robot. Podemos distinguir dos tipos de variables:

- Variables de instalación: son aquellas que se pueden usar en cualquier programa de robot y sus nombres y valores se mantienen aunque se reinicie el robot.
- Variables normales del programa: estas variables son las que solo estarán disponibles en el programa de robot que son creadas y sus valores se pierden cuando se para el programa.

Los tipos de estas variables pueden ser:

- Booleana: Es una variable cuyo valor es Verdadero o Falso.
- Entera: Un número de 32 bits como máximo.
- Flotante: Un número en punto flotante (decimal y 32 bits).
- Cadena: Se trata de una secuencia de caracteres.
- Pose: Se trata de un vector de seis valores en los cuales los tres primeros definen la ubicación (x, y, z) y los tres últimos valores definen la orientación (rx, ry, rz) en el espacio cartesiano. El resultado final es un vector llamado $p[x, y, z, rx, ry, rz]$.
- Lista: Es una secuencia de variables.

4.2.6 Freedrive

El botón del Freedrive situado en la parte posterior de la pantalla táctil conocida como Teach Pendant en inglés. Si mantenemos pulsado este botón podremos mover el robot sin necesidad de utilizar ningún comando o los menús de Polyscope simplemente con la mano podremos mover el robot a la posición que queramos siempre que no se sobrepase los límites articulares o se encuentra con una singularidad en tal caso saltará un aviso en la tablet (*Figura 27*).



Figura 27: Botón de Freedrive

4.3 Funciones básicas de programación

Ahora vamos a desarrollar la pestaña de programa con la cual podremos desarrollar programas con funciones básicas y avanzadas.

El árbol de programa que se observa en la siguiente *Figura 28* es donde se añaden los comandos Básicos como los Avanzados o incluso de puede utilizar plantillas (las plantillas consisten en un árbol de programa ya listo para configurar).

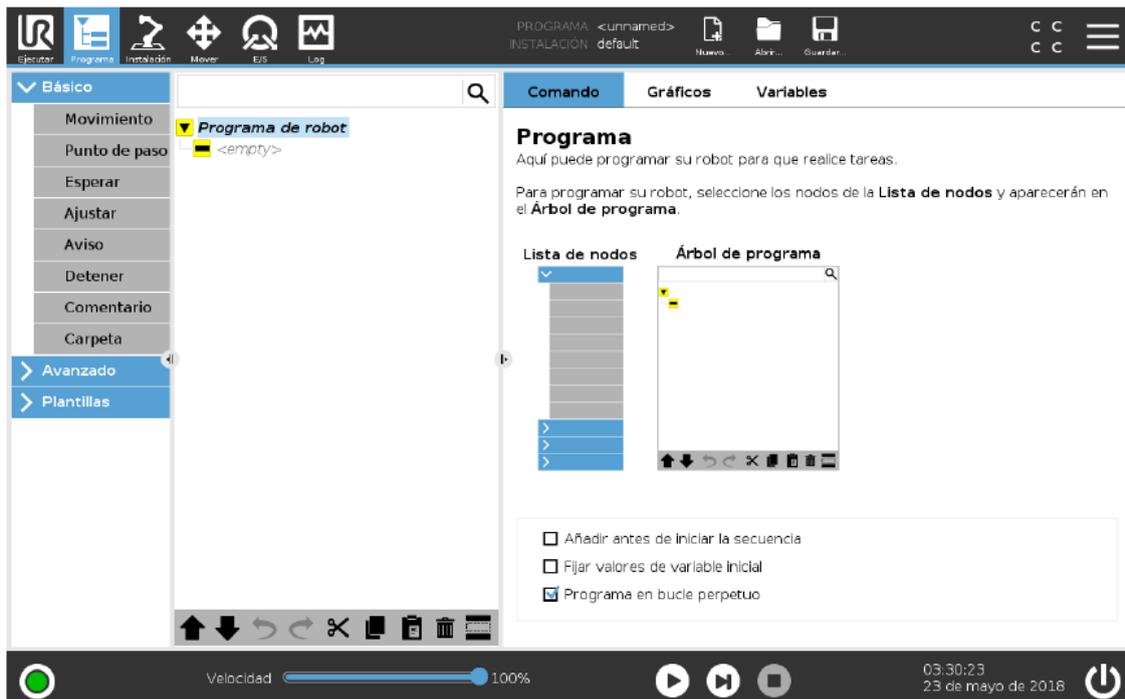


Figura 28: Árbol de programa

4.3.1 Movimiento

Uno de los comandos de programación básicos más importante:

Move: Mediante este comando se puede controlar el tipo de movimiento que el robot va a realizar para mover de un punto de paso a otro. También se puede definir la velocidad y aceleración con la que se moverá el robot entre cada punto de paso.

Existen tres tipos de movimientos:

- **MoveJ:** en este movimiento movemos el PCH (Punto Central de Herramienta) del robot de forma no lineal dando lugar a trayectorias curvas. Utilizamos este tipo de movimiento cuando no importa la ruta a tomar por el PCH y se mueva en un espacio libre (*Figura 29*).

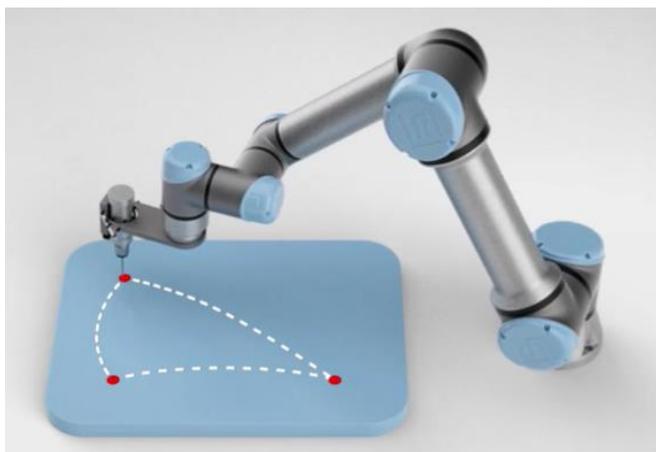


Figura 29: MoveJ

- **MoveL:** este tipo de movimiento mueve el PCH de forma lineal entre los puntos de paso. Este tipo de movimiento lo utilizamos cuando queremos coger o dejar algún objeto desplazándonos linealmente entre los puntos de paso y la ruta de paso es importante. También es utilizado cuando el espacio de la tarea es reducido ya que es un movimiento más preciso que el MoveJ y se realiza a una velocidad más baja (*Figura 30*).

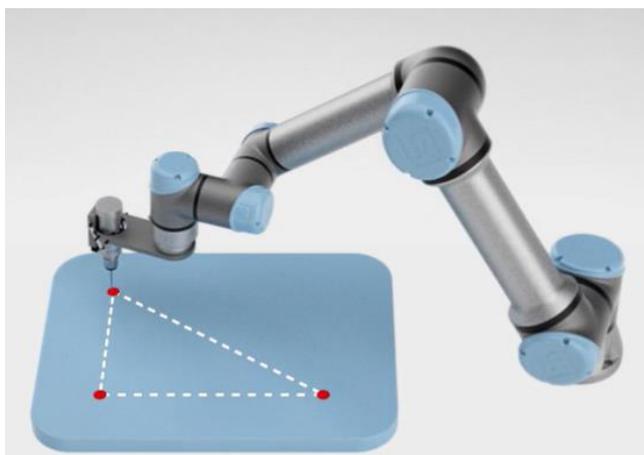


Figura 30: MoveL

- **MoveP:** el movimiento es muy similar al MoveL ya que el PCH se mueve linealmente pero la diferencia es que en este caso la velocidad de movimiento es constante. Al mantener la velocidad constante el PCH no puede realizar cambios bruscos de dirección en ángulos cerrados por lo que necesita realizar un movimiento circular llamado radio de transición. Este tipo de movimiento está pensado para realizar tareas como encolado o la dispensación (*Figura 31*).

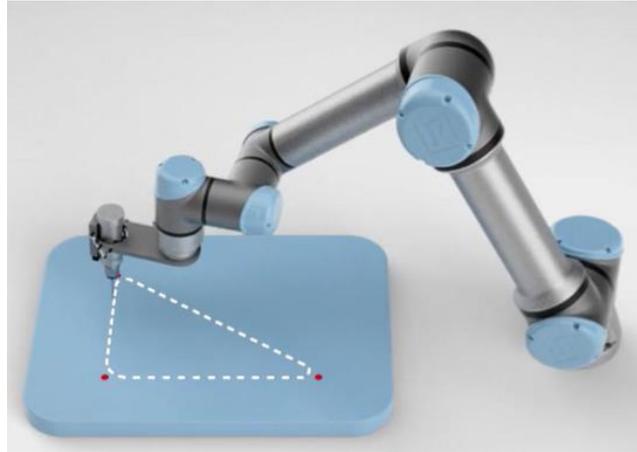


Figura 31: MoveP

4.3.2 Punto de Paso

Un punto de paso indica la localización y orientación en coordenadas cartesianas basadas en el Punto Central de Herramienta actual.

El robot se moverá entre los puntos de paso que marquen su trayectoria y se podrán ajustar de forma manual (pulsando el botón del Freedrive visto anteriormente en la *Figura 27*) o mediante el menú mover para hacerlo de forma más precisa.

Existen distintos tipos de puntos de paso:

- Punto de paso fijo: Estos puntos se tratan de puntos en la trayectoria del robot que no cambiarán una vez modificados para configurar un punto de paso fijo moveremos el robot a una posición determinada y guardaremos ese punto que permanecerá siempre en el mismo lugar (*Figura 32*).



Figura 32: Menú Punto de Paso Fijo

- Punto de paso relativo: en este caso la posición del punto está relacionada con el punto anterior del brazo robótico. Esa relación es la diferencia entre las dos posiciones dadas (de izquierda a derecha) como se puede observar en la *Figura 33*.



Figura 33: Menú Punto de Paso Relativo

- Punto de paso variable: en este caso la posición viene dada por una variable en la cual especificaremos el lugar donde queremos que se dirija el robot (*Figura 34*). Podremos configurar la variable en el menú avanzado dentro del apartado de asignación.



Figura 34: Menú Punto de Paso Variable

4.3.3 Esperar

Mediante esta función podremos pausar el programa durante un tiempo definido o esperar a que ocurra una acción (se active una entrada o que una variable tenga cierto valor) todo esto se podrá configurar en el menú de programación básica dentro del apartado ajustar en la *Figura 35*.

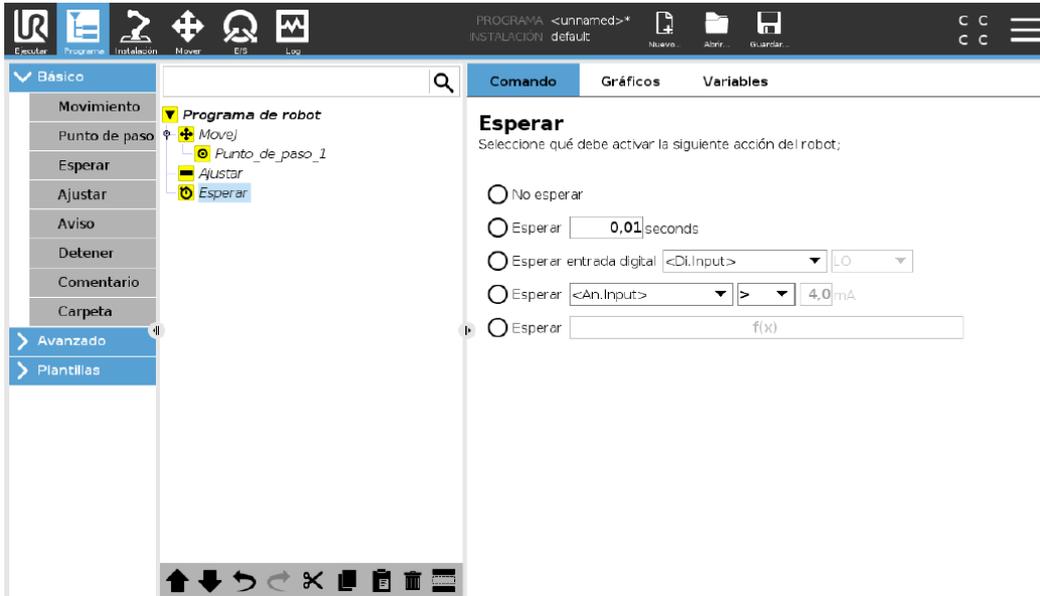


Figura 35: Menú Esperar

4.3.4 Ajustar

Sirve para dar valor a las salidas digitales o analógicas. Además podremos cambiar el valor de las variables descritas anteriormente y podremos modificar la carga útil o cambiar el PCH activo (*Figura 36*).

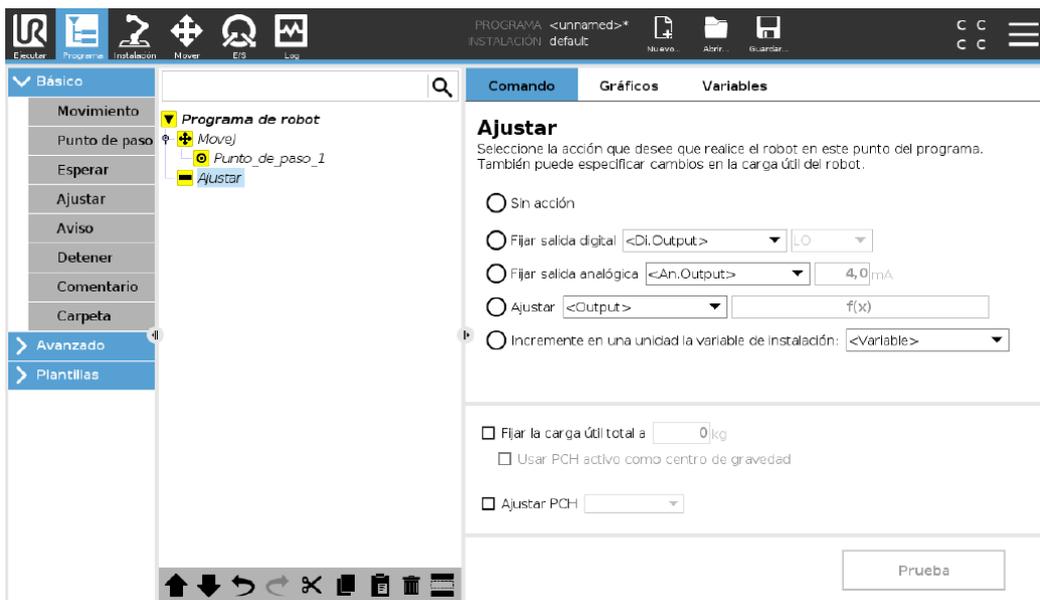


Figura 36: Menú Ajustar

4.3.5 Dirección

Este nodo se encarga de mover el robot según una dirección que viene dada por el PCH hasta que se especifica otra acción (*Figura 37*).

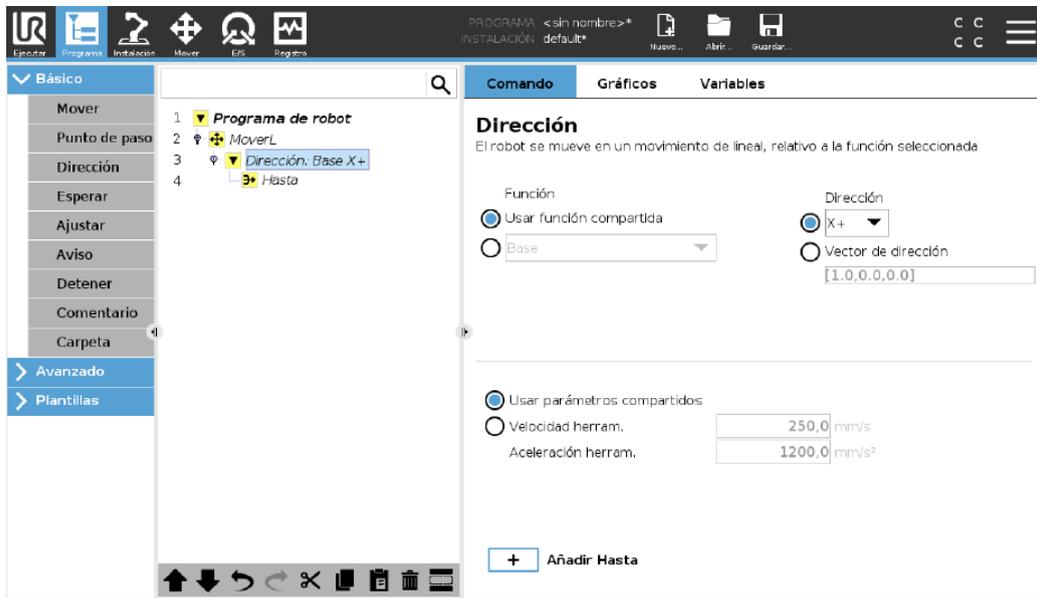


Figura 37: Menú Dirección

4.4 Funciones avanzadas de programación

4.4.1 Bucle

Repita los comandos que se encuentren dentro de este como ocurre en cualquier lenguaje de programación. Hay diferentes opciones como se puede ver en la *Figura 38*:

- Bucle infinito: se repite de forma ilimitada.
- Bucle X veces: se repite las veces que configuremos.
- Bucle dependiendo de la expresión: en este caso se repetirá mientras que la expresión que pongamos sea verdadera.

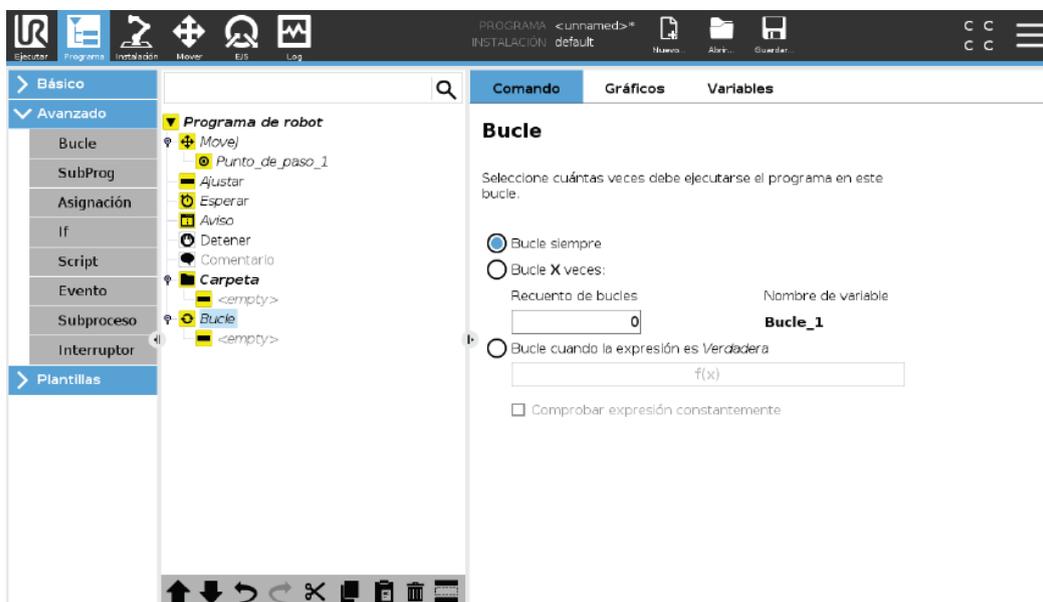


Figura 38: Menú Bucle

4.4.2 Subprograma

Esta función nos sirve para guardar partes del programa para usar en posteriores programas de robot sin necesidad de volver a programar otra vez lo mismo, solamente habría que incluir el subprograma que queramos usar (*Figura 39*).

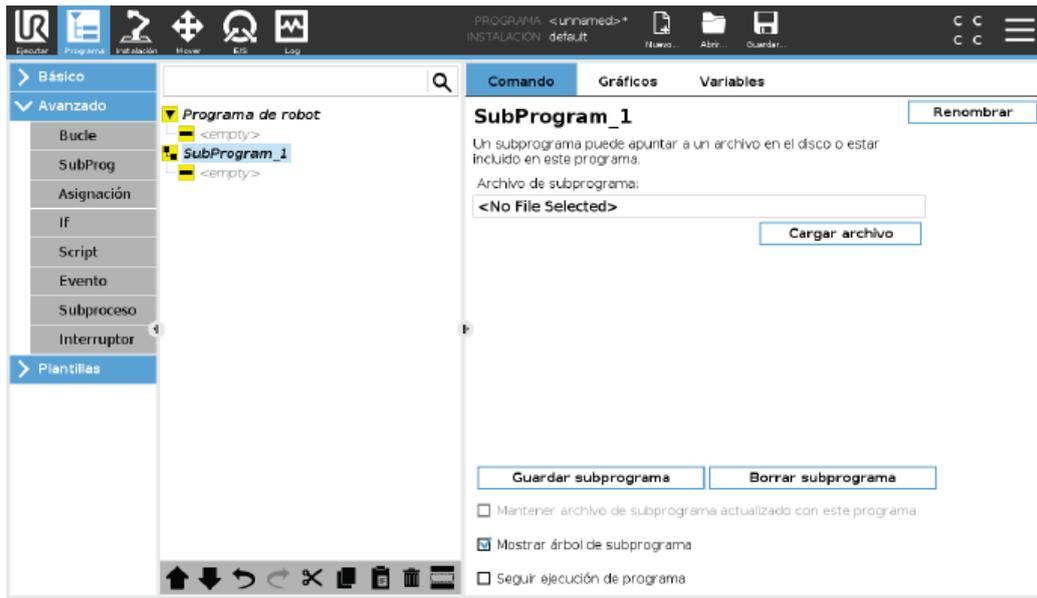


Figura 39: Menú Subprograma

4.4.3 Asignación

Esta función nos permitirá asignar valores a las variables que hemos creado (Figura 40).



Figura 40: Menú Asignación

4.4.4 If

Es uno de los comandos más básicos en la programación y se utiliza para evaluar una condición y si es correcta se ejecuta las líneas de código que estén dentro de ese if sino se ejecutarán las que este dentro del Elseif o Else (Figura 41).

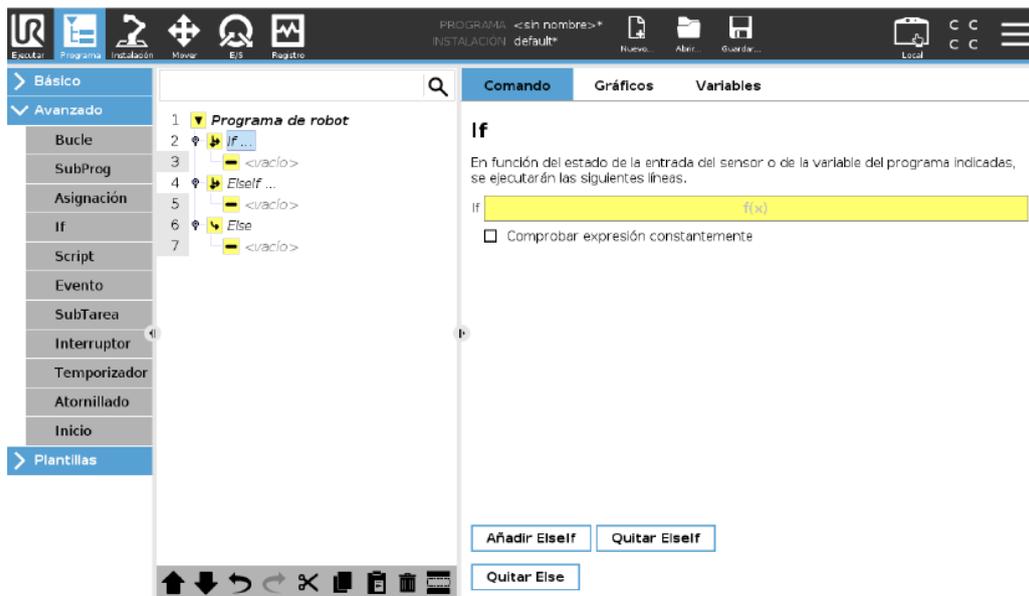


Figura 41: Menú de la función IF

4.4.5 Switch

Mediante este comando podremos evaluar diferentes casos de forma rápida y su estructura es la que aparece en la siguiente Figura 42.

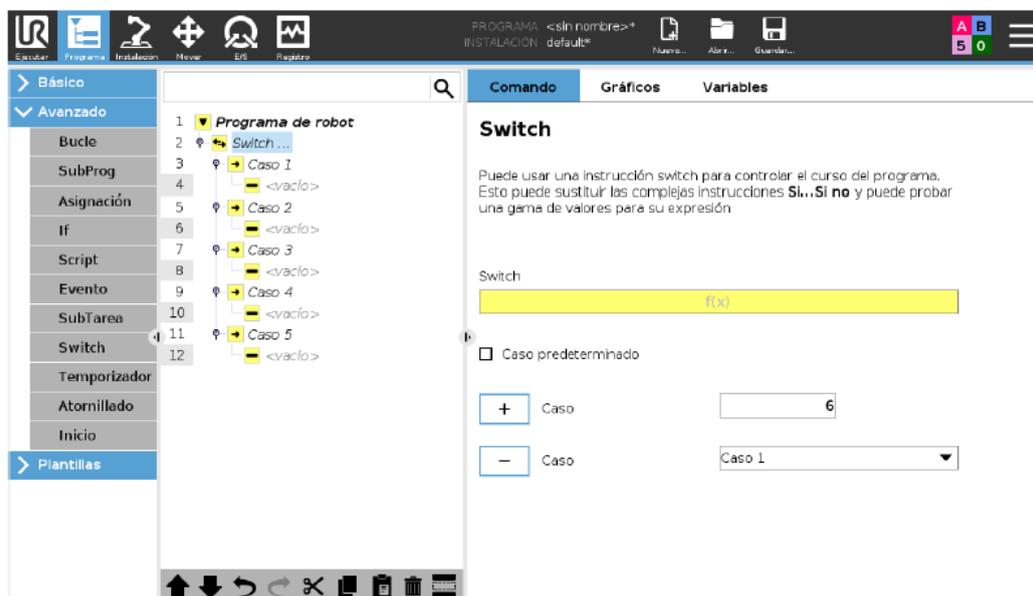


Figura 42: Menú Switch

5 TAREAS DEL PROYECTO

5.1 Características del pick & place

En esta parte vamos a hablar de una de las tareas más utilizadas en la industria que es el pick & place que se puede definir literalmente como recoger y colocar. Este tipo de tarea es una de las que menor ergonomía presenta en toda la cadena de producción por lo que representa una gran carga mental y física para los trabajadores [13].

Debido a esto se plantea el uso de robots colaborativos para realizar la tarea de pick&place (en nuestro caso el robot será el UR5).

Loss robots colaborativos se pueden usar de diferentes formas para realizar la tarea de pick & place, las más destacadas son las siguientes:

- **Ensamblaje:** en un proceso de ensamblaje el robot agarra piezas de una cinta transportadora y la ensambla en otra pieza.
- **Empaque:** se trata de un proceso similar al ensamblaje en el que se coge una pieza de una cinta transportadora pero en este caso no se ensambla a otra pieza sino que se coloca en un paquete a gran velocidad.
- **Bin Picking:** en este caso los robots disponen de sistemas de visión artificial avanzados pueden coger piezas que se encuentran mezcladas y colocarlas de forma correcta para trabajar con ellas.
- **Inspección:** en este caso también se disponen de sistemas de visión para monitorización y la detección de productos que pasan por una cinta transportadora y que pueden tener defectos, una vez lo detecta el cobot podrá retirar el producto defectuoso antes de que se comercialice.

Antes de exponer como hemos realizado la tarea con el UR5 vamos a exponer los beneficios de usar un cobot para esta tarea de pick and place:

- **Rendimiento:** un cobot puede operar durante las 24 horas del día y puede trabajar de forma autónoma o en cooperación con operarios.
- **Seguridad:** este tipo de robots cuentan con sistemas de seguridad avanzados y son menos potentes que los robots industriales tradicionales por lo que pueden trabajar junto con personas sin necesidad de ninguna valla de seguridad.
- **Precisión:** disponen de una precisión milimétrica que nos permitirá realizar tareas que normalmente no podríamos.

- Programación: se pueden reprogramar de forma sencilla para realizar otra tarea.
- Coste: se reducen los costes de mano de obra.

5.2 Montaje pick & place

Para realizar la tarea de Pick & Place una vez inicializado el robot UR5 deberemos realizar las siguientes conexiones (*Figura 43*).

- Conectaremos la Pinza SMC al controlador de motor paso a paso en los puertos 5 y 6 (*Figura 44*).
- Alimentaremos al controlador con 24V en DC a través de la conexión del puerto 7 con fuente de alimentación.
- Conectaremos el cable de Ethernet a la caja de control para tener estar comunicados con la Pinza SMC.

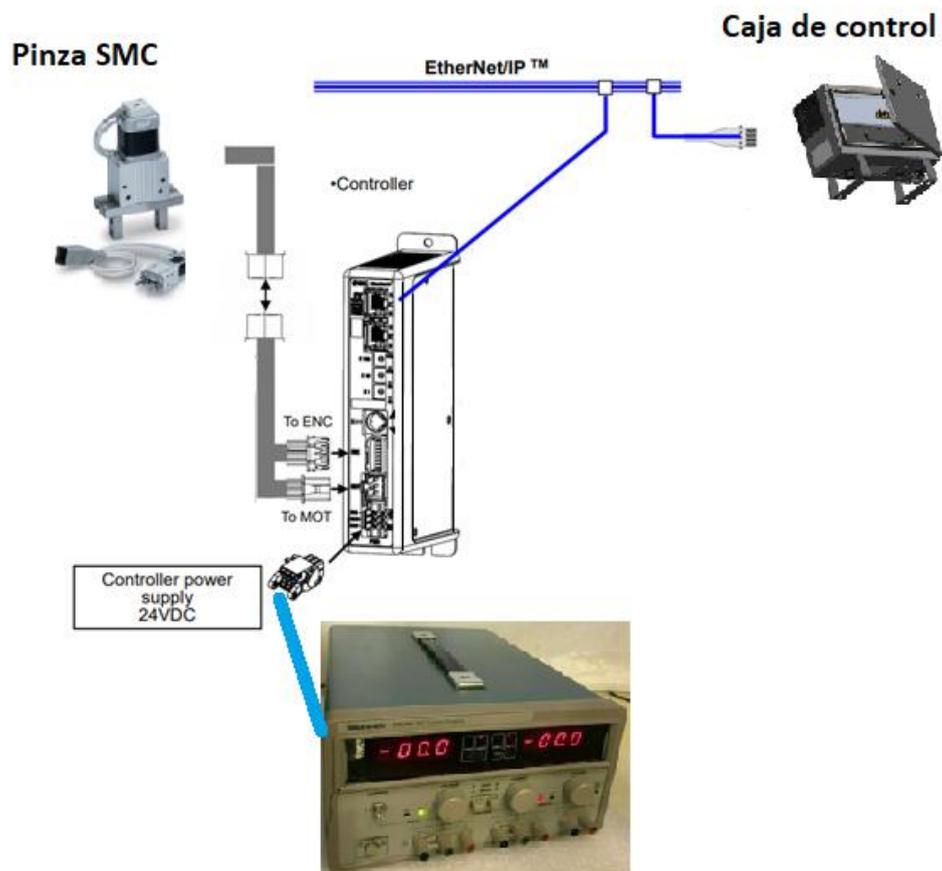


Figura 43: Montaje Pick & Place

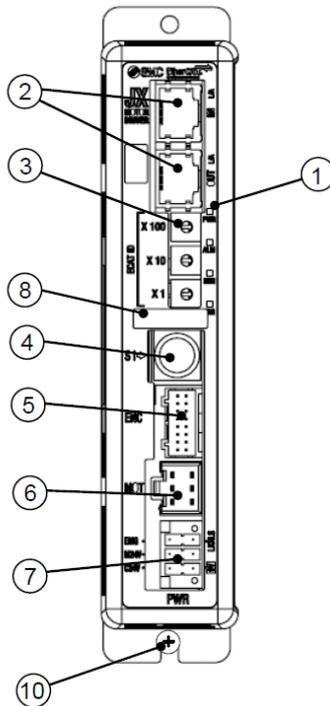


Figura 44: Conexiones del Controlador JXC917-BC-E

5.3 Tarea pick & place en PolyScope

5.3.1 Inicialización de la Pinza SMC

Antes de entrar en la parte puramente de programación hay que entender cómo funciona la pinza SMC dentro de PolyScope.

Una vez conectada la pinza según hemos visto en el apartado de montaje pick & place deberemos inicializarla dentro de PolyScope. Para ello utilizaremos el comando LEH init y se tendrán que cumplir las siguientes condiciones para que se pueda comenzar a trabajar con la pinza (*Figura 45*):

- 1) Se comprueba si está conectada la pinza.
- 2) Se comprueba si está el servomotor en ON.
- 3) Se trata de la última comprobación en la que si SVRE (el servomotor esta activo) podremos comenzar a utilizar la pinza.

Todas estas comprobaciones se realizan de forma automática y cada vez que se comprueba se marca en azul un rectángulo en azul sobre el estado como se ve en la *Figura 45* al realizar todas las comprobaciones.

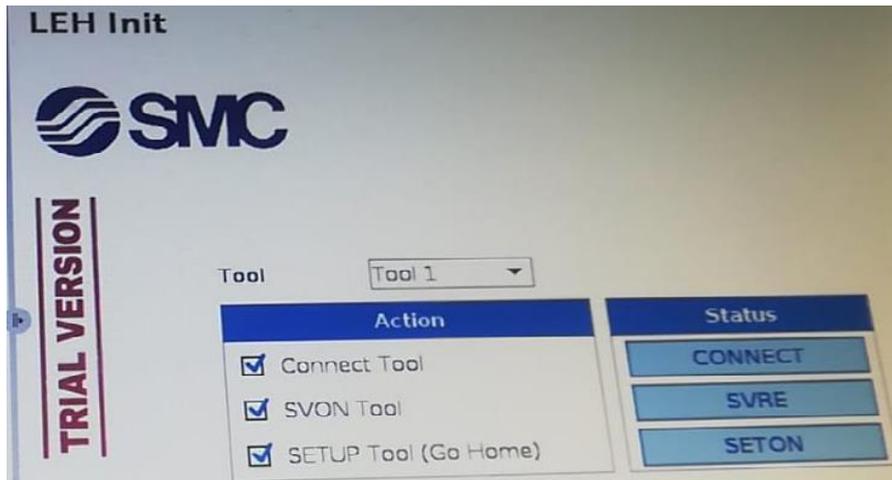


Figura 45: Comando LEH init

Ahora que ya tenemos inicializada la herramienta deberemos ver cómo funciona para poder hacer la tarea de pick & place. Como vemos en el menú del comando LEH Move de la *Figura 46* hay diferentes parámetros que podemos modificar como por ejemplo el de posición que nos permitirá el cierre y la apertura de la pinza. También podremos modificar la velocidad y aceleración de estos movimientos.

A la derecha aparecen de la *Figura 46* con esos dos botones de apertura y cierre de la pinza podremos probar manualmente a abrir y cerrarla sin necesidad de un programa de robot (solo que la pinza este inicializada). Esto nos será útil para comprobar si agarra la pieza de forma correcta o si necesita de algún ajuste antes de cargar el programa de robot.

En PolyScope cada vez que queramos que abrir o cerrar la pinza deberemos añadir este comando indicando en el apartado de la posición lo que queramos que se abra o cierre la pinza.

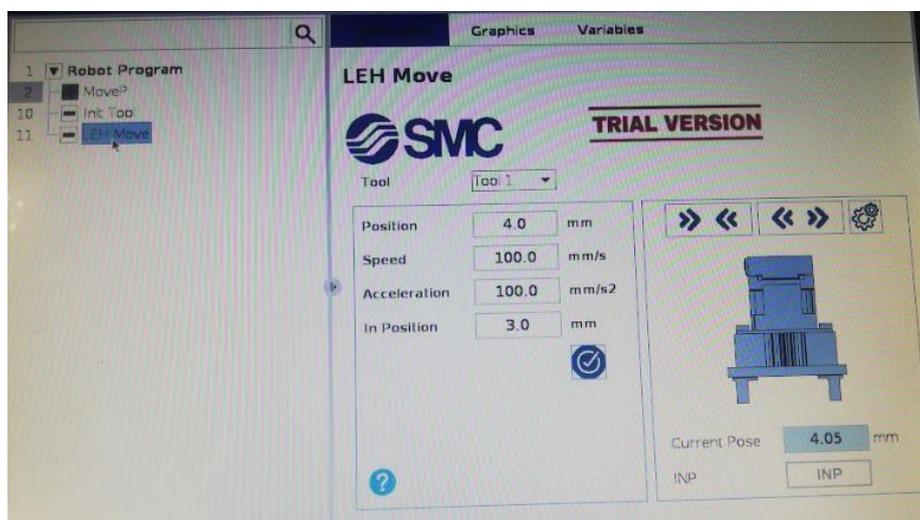


Figura 46: Comando LEH Move

5.3.2 Programación Pick & Place

Una vez sabemos cómo inicializar la Pinza SMC y como configurar para que agarre y suelte las piezas pasaremos a hablar sobre cómo se ha programado la tarea de Pick & Place.

Hay muchas formas de programar esta tarea por ejemplo se podría configurar directamente las posiciones de las piezas de recogida y dejada como puntos de paso, esto se suele hacer cuando las piezas vienen de una en una por una cinta transportadora ya que solo tendremos que configurar una sola posición de recogida y dejada. En el caso que se va a realizar en este proyecto las piezas se encuentran apiladas unas sobre otras (*Figura 47*) por lo que podríamos programar tantas posiciones como piezas tanto de dejada como de salida pero esta es una solución poco óptima porque si movemos el robot deberíamos reprogramar todas las nuevas posiciones de nuevo y si añadimos más piezas deberíamos añadir nuevas posiciones.



Figura 47: Pick & Place Posición Inicial

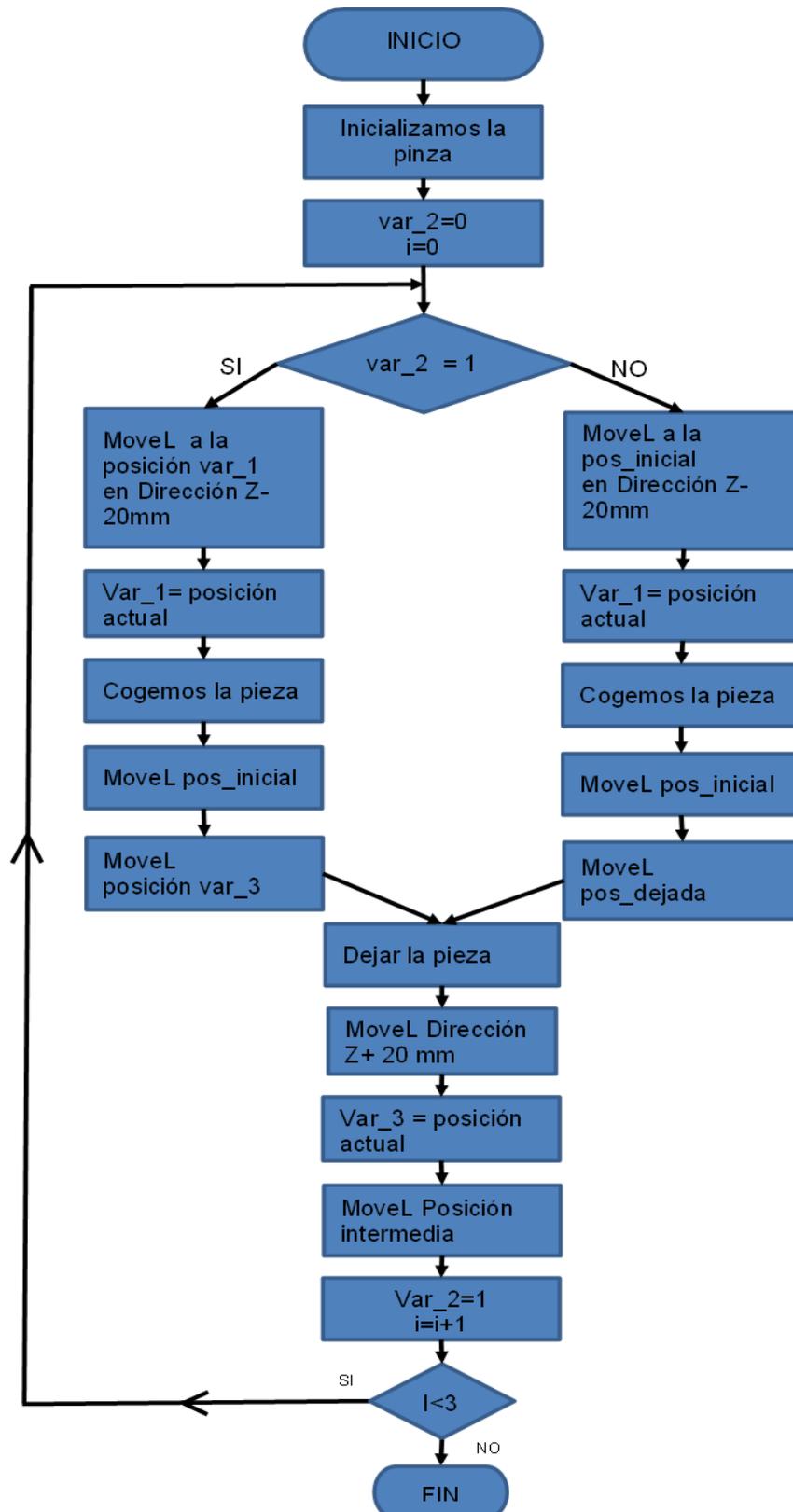
Se va a realizar de la siguiente forma:

- 1) Dentro de la parte del programa del robot que se ejecuta antes del resto y solo una vez conocida como (BeforeStart) inicializamos la pinza SMC mediante el comando Init Tool y creamos una variable que inicializamos a 0 (var_2=0).
- 2) El programa de robot tendrá un bucle que en cuanto realice 3 iteraciones el brazo robótico dejará de moverse y finalizará el programa. Todo los

pasos que vamos a comentar irán dentro del bucle para que cada pieza realice el mismo trayecto. En primer lugar haremos un MoveL a una posición que vendrá determinada por un if cuya condición ($var_2 = 1$) sino se cumple se moverá a var_1 (será una variable en la que configuraremos las posiciones de recogida de piezas cada vez que quitamos). En el caso que no se cumpla se moverá a $pos_inicial$ que será la posición que habremos configurado anteriormente que estará situada perpendicularmente sobre las piezas como se puede ver en la *Figura 47*. En la primera iteración del bucle irá a $pos_inicial$.

- 3) Ahora el movimiento configurado en el paso anterior se realizará con una dirección que será la de Z- es decir hacía el suelo según hemos configurado el PCH en la Pinza y este movimiento será de 20 mm en esa dirección. Una vez ahí registramos la posición actual y la asignamos a la variable var_1 y esa será la posición de recogida de la pieza en la siguiente iteración del bucle.
- 4) Cogemos la pieza en ese punto mediante el comando LEH Move configurando el agarre en 2 mm (el valor de posición dentro del comando LEH Move).
- 5) Realizamos un MoveL para volver a la posición inicial.
- 6) Una vez estamos en la posición inicial nos desplazamos con un MoveL a la posición de dejada de la pieza que en la primera iteración del bucle será una posición que hemos prefijado (pos_dejada). En las siguientes iteraciones la posición de dejada cambiará porque apilamos los bloques uno encima de otro por lo que estas posiciones las obtenemos mediante una variable que registrará la posición una vez dejada la pieza y 20 mm hacia arriba, esta variable será var_3 .
- 7) Dejamos la pieza mediante el comando LEH Move pero en este caso abrimos la pinza por lo que pondremos 4 mm en la campo de posición.
- 8) Una vez dejada la pieza el robot realizará un MoveL en dirección Z+ (hacia arriba) de 20 mm y registraremos la posición actual en la var_3 que luego nos servirá para dejar la pieza siguiente de forma que queden apiladas.
- 9) Realizamos un MoveL a una posición intermedia para evitar posibles obstáculos y asignamos a la variable $var_2=1$.
- 10) Y esta secuencia se realiza 3 veces que es el número de piezas tenemos en nuestra tarea de pick & place.

En el siguiente diagrama de flujo se expone lo comentado anteriormente para que quede de forma más clara.



5.4 Tarea de lijado

5.4.1 Características de la Tarea de Lijado

El lijado es una de las tareas más importantes que se realizan en las fases de acabados de una línea de producción. Esta tarea es muy repetitiva y en muchas empresas las realizan operarios por lo que se podría utilizar la robótica colaborativa para liberar de gran carga de trabajo a los operarios y así aumentar la productividad y la eficacia.

Hay que tener en cuenta que no todos los robots sirven para realizar este tipo de tareas por lo que se necesitan usar robots colaborativos como los que hemos comentado en el Estado del Arte dentro del apartado de robots colaborativos.

En este proyecto usaremos un robot UR5 que nos permite tener un control de fuerza aplicada sobre la superficie de trabajo y la capacidad de adaptarse al entorno de trabajo.

Algunas de las características de usar el robot UR5 para este tarea son:

- Uniformidad y una gran calidad en el resultado.
- Aplicación de fuerzas y pares concretos a lo largo de los ejes predefinidos.
- Liberar de tareas repetitivas y pesadas a los operarios.
- Seguridad de los empleados.

Como hemos visto anteriormente en el apartado del estado del arte en la revisión de los trabajos hay diferentes métodos para realizar el lijado o pulido. En alguno de los trabajos se incorporan sensores de fuerza. Para ver si es necesario incorporar un sensor de fuerza adicional al que viene integrado en el robot UR vamos a basarnos en el estudio de Stefan Gadringer et al. (2022) [14].

En este estudio se hizo una comparación entre el sensor de fuerza/torque del robot UR5 y de una herramienta externa con control de fuerza como FerRobotics ACF-K 109/04 [15]. Los resultados de este estudio afirman que el sensor del UR funcionan de forma adecuada para movimiento en línea recta con una velocidad baja mientras que el de ACF funciona bien para cualquier velocidad y cambio de orientaciones de forma rápida.

En nuestro caso utilizaremos el sensor que dispone el UR5 ya que no necesitamos tanto nivel de precisión a la hora de adaptar la herramienta a la pieza de trabajo y la velocidad que vamos a emplear será baja.

5.4.2 Modo Fuerza

En este apartado comentaremos el modo de Fuerza que es el método empleado para realizar una prueba de la tarea de lijado.

Para realizar esta tarea hemos empleado el comando fuerza que se encuentra en el apartado de plantillas del menú de programa de PolyScope.

El modo fuerza se suele utilizar en aplicaciones en las que se necesite aplicar una fuerza deseada a lo largo de un eje como sería en nuestro caso la tarea de lijado (*Figura 48*).

Hay diferentes formas de usar el modo fuerza según el tipo que elijamos:

- Simple: Solo se aplica una fuerza determinada en un eje que siempre será el eje z de la función seleccionada (según configuremos el PCH).
- Marco: Esta opción es mucho más avanzada que la anterior ya que nos permite configurar la fuerza en los seis grados de libertad.
- Punto: En este caso se va a tener que definir una función punto para que el marco de la tarea tenga el eje Y apuntando desde el PCH hacia el punto que hemos seleccionado previamente.
- Movimiento: En este caso el marco de tarea cambiará según la dirección del movimiento del PCH.



Figura 48: Comando Fuerza modo Simple

En nuestro caso utilizaremos el modo de fuerza tipo marco ya que nos va a permitir ajustar la herramienta utilizada para que permanezca perpendicular a la superficie de lijado ya que con esta opción podremos configurar los valores de fuerza aplicados en cada eje (X, Y, Z) y el par en cada uno de ellos (Rx, Ry, Rz).

5.4.3 Montaje Lijado

En esta parte vamos a mostrar el montaje para realizar la tarea de lijado.

Una de las primeras pruebas que se hizo fue con la herramienta de lijado con goma espuma que se muestra a la izquierda de la *Figura 50* pero la goma espuma hacía que no se adaptará bien a la superficie de trabajo y que al desplazarse el robot mantenía la ortogonalidad con la superficie por lo que descartamos este material.

Después probamos con un fieltro adhesivo como se puede ver a la derecha de la *Figura 50* y en este caso sí que nos permitía adaptarnos a la superficie de trabajo de forma rápida y nos permitía mantener la ortogonalidad con la superficie de trabajo por lo que decidimos escoger este material. El montaje es como se puede ver en la *Figura 51* se conecta la herramienta al brazo robótico mediante cuatro tornillos Allen.



Figura 49: Herramienta de Lijado

5.4.4 Programación Prueba de Lijado

Una vez conectada la herramienta de lijado al robot UR5 como hemos indicado en el apartado anterior Montaje Lijado los pasos que tendremos que realizar serán los siguientes:

- 1) Configurar el PCH de la herramienta de lijado, se hará que será en el centro de naranja como se puede ver en la siguiente *Figura 51*.
En este caso como el PCH está ubicado directamente sobre el eje Z y no se encuentra desplazado de los ejes X e Y podremos medir directamente la distancia desde el PCH hasta la brida de herramienta (es donde se conecta la herramienta al robot). Ahora iremos al menú de Instalación y donde pone PCH introduciremos la distancia que en este caso concreto es de 123mm en la coordenada Z.

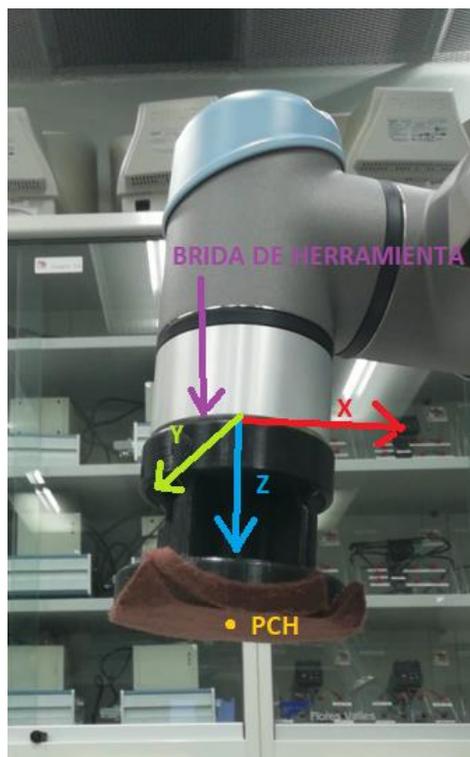


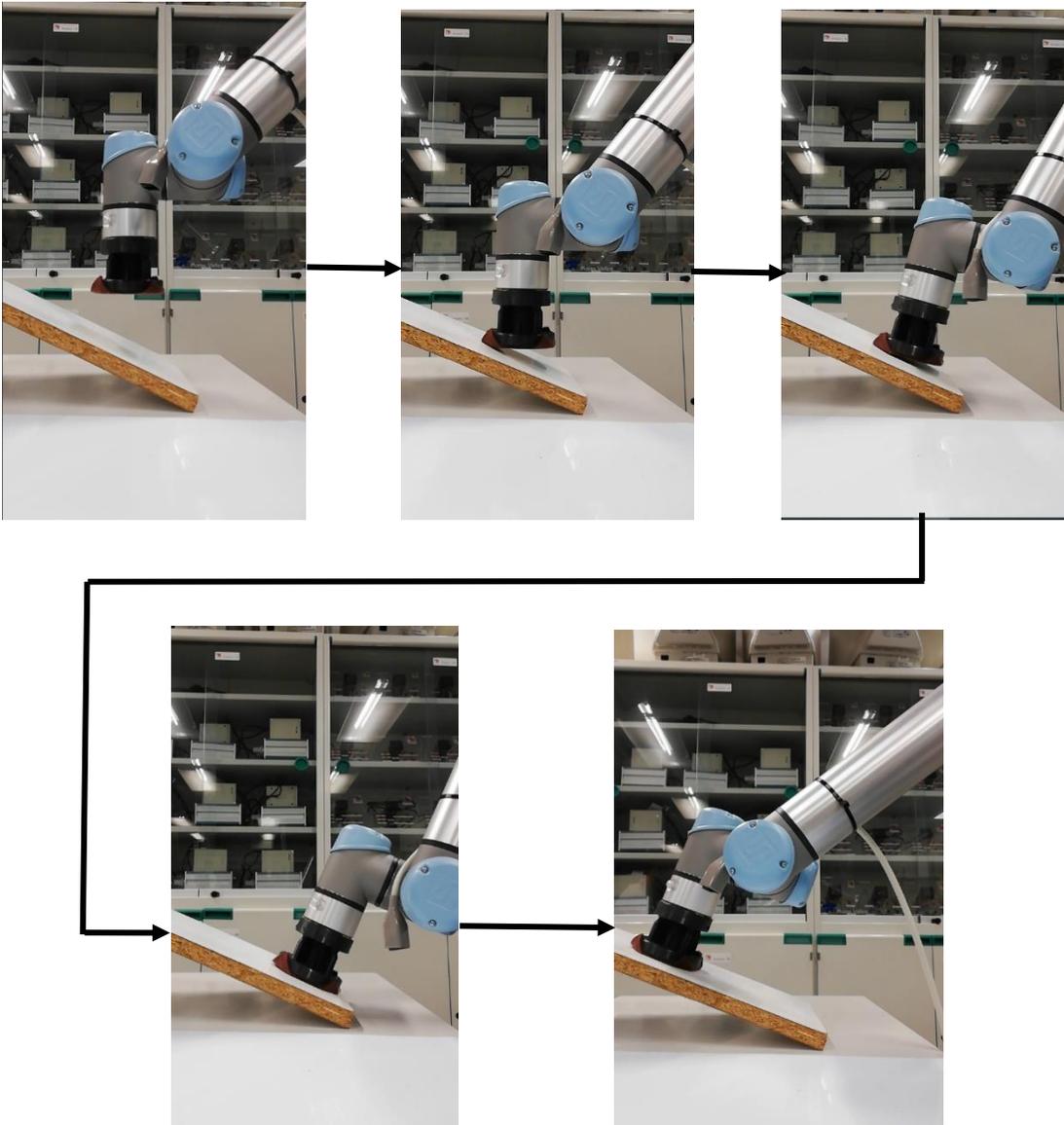
Figura 50: PCH Herramienta de Lijado

- 2) Una vez configurado el PCH pasaremos a configurar el centro de gravedad y la carga útil (como hemos visto en el punto 4.2.3). El resultado será el que se muestra en la *Figura 52*.

Carga	
	<input type="text" value="0.75"/> kg
Centro de gravedad	
CX	<input type="text" value="-79.8"/> mm
CY	<input type="text" value="8.2"/> mm
CZ	<input type="text" value="22.3"/> mm

Figura 51: Carga y Centro de gravedad

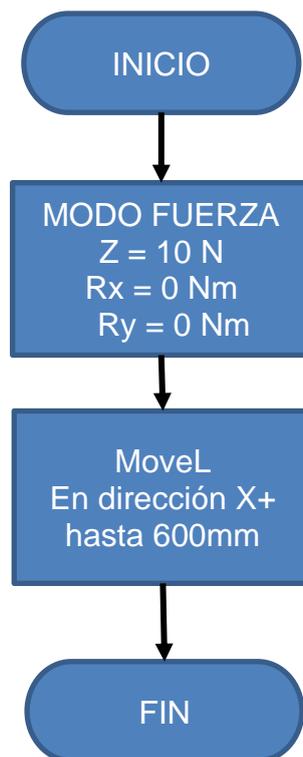
- 3) Una vez configurado tanto el PCH como la carga y centro de gravedad de la herramienta pasaremos a las pruebas de lijado. La primera prueba que se hace es para comprobar cómo se adapta a los cambios de plano utilizando el modo de fuerza de tipo marco comentado en el punto 5.4.2. Con esta prueba lo que esperamos es encontrar es los parámetros idóneos para mantener la herramienta perpendicular a la superficie aunque cambiemos de plano. Para ello elegiremos una Fuerza de 10N en el Eje Z con una velocidad límite de 75 mm/s y pondremos que el par en X e Y sea 0 con una velocidad límite de 45 °/s para mantener la ortogonalidad con la superficie.
- 4) Una vez definidos los parámetros que vamos a usar veremos una prueba en la que hay movimiento y en la que se tiene que adaptar la herramienta al plano.



Ahora vamos a comentar como se hizo el programa:

En el primer lugar situamos el brazo robótico sobre la pieza que queremos lijar y para demostrar cómo se adapta a la superficie la herramienta de lijado lo dejamos de forma que al bajar tenga que adaptarse al plano inclinado como se pueden observar en las siguientes imágenes como se adapta. Una vez se encuentra perpendicular gracias a que lo hemos configurado que los pares en Rx y Ry permanecerá perpendicular a la superficie aunque variemos el ángulo de inclinación del plano o incluso cambiemos el plano. Ahora realizaremos un movimiento en dirección X+ hasta una distancia que hemos prefijado de 600 mm. También podríamos prefijar puntos donde los que ir con el robot para que realice un recorrido completo de la pieza mientras se mantiene perpendicular a esta.

A continuación se muestra en un diagrama de flujo los pasos comentados anteriormente.



- 5) Como última prueba se realizó una prueba de lijado sobre una superficie plana en la que configuramos cuatro puntos de paso que serán los vértices del cuadrado y activado el modo de fuerza simple ya que en este caso no haría falta usar el modo marco porque la superficie de lijado es plana.

6 CONCLUSIÓN

En este proyecto se ha tratado de plasmar como sería comenzar de cero la puesta en marcha de un robot UR5 y los primeros pasos a realizar a la hora de realizar un programa de robot. También se ha mostrado como utilizar herramientas en el robot como son la pinza SMC que la hemos utilizado para realizar el pick & place o la herramienta de lijado que la hemos utilizado para la prueba de lijado.

Me ha sorprendido la gran cantidad de opciones de programación, no echamos en falta ninguna función clásica de los lenguajes de programación y todo ello incluido dentro del programa PolyScope. Además de contar con una gran variedad de funciones podemos crear funciones propios y trabajar con ellas como ha ocurrido al usar la URCap de la pinza SMC que es un programa que no está incluido en el software del robot pero lo podemos ejecutar de todas formas.

Hay que decir que la pinza SMC ha dado algunos problemas ya que su carrera era demasiado corta es de unos 4 mm (en realidad era algo menos incluso) lo que nos permitía coger las piezas sin apenas margen de error. A pesar de ello con la precisión de movimiento del UR5 nos ha facilitado este problema que si el robot hubiera sido menos preciso tendríamos que haber elegido otro tipo de piezas.

En cuanto a la prueba de lijado se probó con diferentes materiales para simular en primer lugar se probó recubrir la superficie de lijado con goma eva pero enseguida nos dimos cuenta de que no fue una buena idea ya que al ejercer la fuerza y querer desplazarnos la herramienta ofrecía una gran resistencia al movimiento por lo que desechamos ese material. También se probó con goma espuma que no se podía mantener la perpendicularidad con la superficie de trabajo. Después probamos un fieltro adhesivo una material que nos permitiría poder realizar la fuerza para mantener la herramienta perpendicular a la superficie de lijado y poder realizar el movimiento de la herramienta por la superficie de lijado.

En conclusión estoy contento de poder haber trabajado con un robot real y tan práctico como el UR5 y haber aprendido de profesionales como llevar a cabo diferentes tipos de tareas con el robot UR5.

6.1 Trabajos Futuros

Como trabajos futuros se podría incorporar un sistema de visión para realizar una tarea de pick & place que sea capaz de segmentar por ejemplo por el color o la forma. Así se demostraría que se puede implantar sistemas más complejos y que no solo es capaz de hacer tareas sencillas sino que se puede escalar a problemas más complejos.

Probar diferentes herramientas de lijado para ver cual se adapta mejor a la superficie de lijado y así tener mejor resultado.

Diseñar una sistema que permita al operario manipular un al robot mientras se está realizando el lijado por si se ha dejado algo por lijar o si se quiere pasar más por una zona en específico.

7 BIBLIOGRAFÍA

- [1] What is the history of collaborative robots? September 14, 2021 de Jean-marc Buchert. Disponible en <https://manplasmachines.com/history-collaborative-robots/>
- [2] Robots Colaborativos de Universal Robots. Disponible en <https://www.universal-robots.com/es/productos/>
- [3] Villagrossi, E., Cerati, C., Pedrocchi, N., Beschen, M., & Molinari Tosatti, L. (2017). Flexible robot-based cast iron deburring cell for small batch production using singlepoint laser sensor. *The International Journal of Advanced Manufacturing Technology*, 92, 1425–1438.
- [4] Diao, S., Chen, X., & Luo, J. (2018). Development and experimental evaluation of a 3Dvision system for grinding robot. *Sensors*, 18, 1–20.
- [5] Mohsin, I., He, K., Cai, J., Chen, H., & Du, R. (2017). Robotic polishing with force controlled end effector and multi-step path planning. *IEEE Int. Conference on Information and Automation (ICIA)*, 2017, 344–348.
- [6] Oba, Y., & Kakinuma, Y. (2017). Simultaneous tool posture and polishing force control of unknown curved surface using serial-parallel mechanism polishing machine. *Precision Engineering*, 49, 24–32.
- [7] Segreto, T., & Teti, R. (2019). Machine learning for in-process end-point detection in robot-assisted polishing using multiple sensor monitoring. *Int. J. Adv. Manuf. Technol.*, 103, 4173–4187.
- [8] Kharidege, A., Ting, D., & Yajun, Z. (2017). A practical approach for automated polishing system of free-form surface path generation based on industrial arm robot. *International Journal of Advanced Manufacturing Technology*, 93, 3921–3934.
- [9] H. Ochoa, R. Cortesao, Control architecture for robotic-assisted polishing tasks based on human skills, in: *IECON 2019–45th Annual Conference of the IEEE Industrial Electronics Society*, volume 1, 2019, pp. 630–637.
- [10] Gaz, C., Magrini, E., & De Luca, A. (2018). A model-based residual approach for human-robot collaboration during manual polishing operations. *Mechatronics*, 55, 234–247.
- [11] Gracia, L., Solanes, J., Muñoz-Benavent, P., Valls Miro, J., Perez-Vidal, C., & Tornero, J. (2019). Human-robot collaboration for surface treatment tasks. *Interaction Studies*, 20, 148–184.
- [12] Controladora de motor paso a paso de la marca SMC. Disponible en <https://www.smc.eu/es-es/productos/controlador-de-motor-paso-a-paso-jxceh-9h-ph~170810~cfg>

[13] Pick and place. Qué es y aplicación de la robótica colaborativa. Disponible en <https://cadecobots.com/pick-and-place-que-es-y-aplicacion-de-la-robotica-colaborativa/>

[14] Gadringer, S., Gattringer, H., & Mueller, A. (2022). Assessment of force control for surface finishing—an experimental comparison between Universal Robots UR10e and FerRobotics active contact flange. *Mechanical Sciences*, 13(1), 361-370.

[15] FerRobotics Compliant Robot Technology GmbH: Active Contact Flange Kit. Disponible en <https://www.ferrobotics.com/en/services/products/active-contact-flange-kit/>