



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



Escuela Técnica Superior de Ingeniería del Diseño

Escuela Técnica Superior de Ingeniería del Diseño

Grado en Ingeniería Aeroespacial

Trabajo Final de Grado

---

# Diseño de Sistema de Control Robótico para un Puesto de Soldadura con Medición 3D Asistida por CAD.

---

MEMORIA

**Autor:**

Guillermo García-España Brines

**Director:**

Sergio García-Nieto Rodríguez

**Codirectores:**

Alfredo Rosado Muñoz

Antonio Valls Obrer

JULIO 2017





# Índice general

<b>1. Objetivos y alcance del proyecto</b>	<b>1</b>
1.1. Objetivos	1
1.2. Alcance del proyecto	2
<b>2. Introducción y antecedentes</b>	<b>3</b>
2.1. Introducción a los robots	3
2.1.1. Definición de robot	3
2.1.2. Historia	4
2.1.3. El uso del robot	6
2.1.4. Tipos de robot	7
2.1.4.1. Robots antropomórficos	7
2.1.4.2. Robots SCARA	8
2.1.4.3. Robots delta	9
2.2. Introducción a la cinemática del robot	9
2.2.1. Traslación	10
2.2.2. Rotación	10
2.2.2.1. Matrices de rotación, R	10
2.2.2.2. Cuaterniones	11
2.2.3. Cinemática directa del brazo del robot manipulador	11
2.2.3.1. Representación de Denavit-Hartenberg	12
<b>3. Descripción del proceso</b>	<b>15</b>
3.1. Puesta a punto del laser tracker	15
3.2. Medición de la mesa y la pieza	16
<b>4. Soluciones alternativas</b>	<b>19</b>
4.1. Hardware	19
4.1.1. PLC	19
4.1.1.1. Schneider y Omron	20
4.1.2. Robot	22
4.2. Software	24
4.2.1. Polyworks	26
<b>5. Solución adoptada</b>	<b>29</b>
5.1. Hardware	29
5.1.1. CPU	29
5.1.2. Módulos de entradas digitales	30
5.1.3. Módulo de salida digital	31
5.1.4. Controlador Robótico	32

5.1.5. FlexPendant . . . . .	34
5.1.6. Brazo robótico . . . . .	35
5.1.7. Leica AT402 laser tracker . . . . .	37
5.1.8. Reflector Bola . . . . .	38
5.2. Software . . . . .	39
5.2.1. GRAFCET . . . . .	39
5.2.2. TIA Portal . . . . .	43
5.2.3. RobotStudio . . . . .	45
5.2.4. OpenIndy . . . . .	46
<b>6. Tests y resultados</b>	<b>49</b>
6.1. Test del primer tramo . . . . .	49
6.2. Test del tramo 2 . . . . .	51
6.3. Test del tramo 3 . . . . .	53
6.4. Test del tramo 4 . . . . .	55
6.5. Test del tramo final . . . . .	57
<b>7. Conclusiones</b>	<b>61</b>
<b>Bibliografía</b>	<b>63</b>

# Índice de figuras

2.1. Reloj en la catedral de Wells, 1175 . . . . .	4
2.2. Manipulador tipo maestro-esclavo . . . . .	5
2.3. Primer robot <i>Unimate</i> . . . . .	5
2.4. Semejanza del robot industrial con el cuerpo humano . . . . .	7
2.5. Robot antropomórfico . . . . .	7
2.6. Robot SCARA . . . . .	8
2.7. Robot delta . . . . .	9
2.8. Sistemas de referencia . . . . .	10
2.9. Interpretación geométrica de la matriz de transformación homogénea . . . . .	12
2.10. Sistemas de coordenadas para la convención de D-H . . . . .	13
3.1. Componentes del laser tracker Leica AT402 . . . . .	16
3.2. Medición de un punto de la mesa . . . . .	16
4.1. Schneider Modicon M221 . . . . .	21
4.2. Omron CP1L . . . . .	22
4.3. Brazo robótico KUKA KR 30-3 C . . . . .	23
4.4. Rango de movimientos robot KUKA KR 30-3 C . . . . .	24
4.5. Lenguaje Ladder para la función $M = A(B'+C)D'$ . . . . .	25
4.6. Bucle FOR en Matlab . . . . .	25
4.7. Diagrama de flujo . . . . .	26
4.8. Error generado con Polyworks . . . . .	27
5.1. Vista frontal Siemens SIMATIC S7-1200 . . . . .	30
5.2. Vista trasera Siemens SIMATIC S7-1200 . . . . .	30
5.3. Vista lateral Siemens SIMATIC S7-1200 . . . . .	30
5.4. Vista frontal Siemens SM 1221 DC . . . . .	31
5.5. Vista trasera Siemens SM 1221 DC . . . . .	31
5.6. Vista lateral Siemens SM 1221 DC . . . . .	31
5.7. Vista frontal SM 1222 DC . . . . .	32
5.8. Vista trasera SM 1222 DC . . . . .	32
5.9. Vista lateral SM 1222 DC . . . . .	32
5.10. Controlador ABB IRC5 . . . . .	33
5.11. Módulos del controlador IRC5 . . . . .	33
5.12. FlexPendant . . . . .	34
5.13. Brazo robot ABB IRB6600 . . . . .	35
5.14. Ejes robot ABB IRB6600 . . . . .	36
5.15. Rango de movimientos robot ABB IRB6600 . . . . .	37
5.16. Leica AT402 laser tracker . . . . .	38

5.17. Reflector bola . . . . .	39
5.18. Etapas inicial y primera . . . . .	40
5.19. Acciones asociadas a las etapas . . . . .	40
5.20. Transiciones entre etapas . . . . .	40
5.21. GRAFCET del proceso . . . . .	41
5.22. Dispositivos implementados en TIA Portal . . . . .	43
5.23. Estado inicial del GRAFCET . . . . .	44
5.24. Temporizador de establecimiento del GRAFCET . . . . .	44
5.25. Salida de soldadura del GRAFCET . . . . .	45
5.26. Estación con RobotStudio . . . . .	46
5.27. Distribución de puntos medidos con OpenIndy . . . . .	46
5.28. Error generado con OpenIndy . . . . .	47
6.1. Posición en el tramo 1 . . . . .	49
6.2. Vista cenital en Ref0 . . . . .	50
6.3. Velocidad en el tramo 1 . . . . .	50
6.4. Tiempo total en realizar el tramo 1 . . . . .	51
6.5. Posición en el tramo 2 . . . . .	51
6.6. Vista cenital en Ref1 . . . . .	52
6.7. Velocidad en el tramo 2 . . . . .	52
6.8. Tiempo total en realizar el tramo 2 . . . . .	53
6.9. Posición en el tramo 3 . . . . .	53
6.10. Vista cenital en Ref3 . . . . .	54
6.11. Velocidad en el tramo 3 . . . . .	54
6.12. Tiempo total en realizar el tramo 3 . . . . .	55
6.13. Posición en el tramo 4 . . . . .	55
6.14. Vista cenital en Ref4 . . . . .	56
6.15. Velocidad en el tramo 4 . . . . .	56
6.16. Tiempo total en realizar el tramo 4 . . . . .	57
6.17. Posición en el tramo 5(I) . . . . .	57
6.18. Velocidad en el tramo 5(I) . . . . .	58
6.19. Tiempo total en realizar el tramo 5(I) . . . . .	58
6.20. Posición en el tramo 5(II) . . . . .	59
6.21. Velocidad en el tramo 5(II) . . . . .	59
6.22. Tiempo total en realizar el tramo 5(II) . . . . .	60

# Índice de tablas

2.1. Numeraciones de cuaterniones, donde $\epsilon$ representa Craig y $q$ es la notación en este trabajo . . . . .	11
2.2. Definición de cuaternión . . . . .	11
4.1. Características técnicas del Modicon M221 . . . . .	21
4.2. Características técnicas del CP1L . . . . .	22
4.3. Características técnicas del KUKA KR 30-3 C . . . . .	23
4.4. Valores del robot KUKA KR 30-3 C . . . . .	24
5.1. Características técnicas de la CPU . . . . .	29
5.2. Características técnicas del módulo de entrada . . . . .	30
5.3. Características técnicas del módulo de salida . . . . .	32
5.4. Panel de control . . . . .	34
5.5. FlexPendant . . . . .	34
5.6. Características generales del IRB6600 . . . . .	36
5.7. Capacidad de movimientos del IRB6600 . . . . .	37
5.8. Acciones del GRAFCET . . . . .	41
5.9. Transiciones del GRAFCET . . . . .	41

---

# Capítulo 1

## Objetivos y alcance del proyecto

El presente proyecto abordará la implementación de un sistema de control robótico para un puesto de soldadura con medición 3D asistida por CAD que funcione de manera automática. Esto es, que una vez introducidos los datos de la medición, los programas y el PLC empleados sean capaces de dirigir al robot al destino deseado, siempre respetando las normas de seguridad y con la máxima precisión posible.

Se desea optimizar la fase de recolección e introducción de los datos necesarios para reducir todo lo posible el tiempo de espera de actuación del robot. Por tanto, habrá que implementar un sistema en el cual el ordenador a bordo sea capaz de transmitir la información necesaria para realizar la tarea de forma correcta y eficiente.

Primero se analizará el proceso y se planteará el sistema de control que se debe realizar para el manejo de todas las variables. Posteriormente, se implementará físicamente este sistema a un robot industrial, el cual es el encargado de realizar la tarea designada. Cabe mencionar que este proyecto se centra en la soldadura, pero el proceso sería exactamente análogo para cualquier herramienta utilizada en el robot, teniendo en cuenta el tamaño de ésta.

Finalmente, para comprobar el correcto funcionamiento del sistema de control y del robot, se simulará una prueba por ordenador con éste último para verificar su utilidad.

### 1.1. Objetivos

Este proyecto aborda el diseño de un sistema de control para gobernar un robot industrial en un proceso de soldadura.

Para ello, en primer lugar habrá que obtener los datos necesarios de las mediciones de las piezas, que posteriormente serán transmitidas al robot. Después será necesario analizar las variables que controlan nuestro sistema, es decir, definir que va a hacer el robot y como se pueden alcanzar las especificaciones teniendo en cuenta las limitaciones del propio robot. El siguiente paso es la implementación, traducir dichas especificaciones a un conjunto de códigos y algoritmos que sean capaces de gestionar el robot.

Una vez realizado esto, se debe testear en simulación, a fin de detectar fallos que pueden ser fatales, bien sea por motivos de seguridad o deterioro del equipo, además de reducir el tiempo de trabajo. La simulación se realiza tanto con el software RobotStudio como con el robot real. El software mencionado permite realizar simulaciones reales con cualquier tipo

de robot, y permite introducir entradas y salidas digitales que permiten al operario utilizar el programa como si fuese el PLC dando las instrucciones en cada momento, por lo que su fiabilidad es muy alta.

Por tanto, los objetivos a cumplir en este proyecto son:

- La correcta implementación del laser tracker para la medición de la mesa con la máxima precisión posible.
- Obtención de los puntos a seguir por el brazo robot.
- Adaptación del sistema para albergar posibles modificaciones o mejoras.
- Implementación del PLC que gobernará todo el proceso.
- Diseño del mecanismo a través del cual el robot y el ordenador podrán transferirse las variables necesarias.
- Realización de pruebas con el software de simulación.
- Realización de pruebas con el robot.
- Validación y comprobación de resultados experimentales.

## 1.2. Alcance del proyecto

Este proyecto ha sido realizado en colaboración con la empresa InPro electric con sede en España. Por esto motivo, no es un proyecto cerrado, ya que siempre se va a buscar la mejoría para sacar el máximo provecho al proceso en cuestión. Por ello, se partirá de unos algoritmos creados para la medición y la calibración de las mesas de trabajo, sobre los cuales se han introducido cambios para mejorar el comportamiento del sistema.

Como alcance del proyecto, lo que se pretende es automatizar un proceso de soldadura de la industria automovilística, es decir, que la obtención de datos y su lectura por parte del robot sea implementada por un controlador que gobierne todo el proceso.

---

## Capítulo 2

# Introducción y antecedentes

A lo largo de la historia del ser humano, el hombre ha venido creando herramientas y artefactos que lo ayuden a realizar tareas complicadas y/o peligrosas. De esta manera se desarrollaron máquinas que tienen varias ventajas entre ellas: maximizar costos, reducir riesgos para obreros, disminuir tiempos de operación, producción en serie, etc.

Como se ve, los robots juegan un papel fundamental en el desarrollo de empresas y en la creación de nuevas tecnológicas. El robot industrial es la unión de una parte mecánica con una parte electrónica que a su vez funciona como dispositivo de control principal.

Antes de entrar de lleno en el proyecto, se va a realizar una introducción a los robots articulados antropomórficos. Primero se definirá que es un robot y se dará una pequeña clasificación en función de sus características y aplicaciones, para luego describir la cinemática que los gobierna.

## 2.1. Introducción a los robots

### 2.1.1. Definición de robot

Establecer una definición precisa y formal a la hora de describir un robot puede ser complicado. Es por ello, que según la referencia bibliográfica, la definición puede ser muy dispar comparada con otra.

En el *Robot Institute of America, 1979* se define un robot como:

*“Es un dispositivo reprogramable y multifuncional diseñado para mover materiales, piezas, herramientas o dispositivos especializados a través de movimientos programados.”*

La Real Academia Española lo define de la siguiente forma:

*“Máquina o ingenio electrónico programable, capaz de manipular objetos y realizar operaciones antes reservadas solo a las personas.”*

Esta disparidad a la hora de definir un robot es debida al amplio abanico de funciones que ofrecen los robots, que tiene como resultado dificultar una única definición que unifique el término.

### 2.1.2. Historia

El concepto de máquinas automatizadas se remonta a la antigüedad, donde las primeras muestras eran los brazos mecánicos que unían los antiguos egipcios a las estatuas de sus dioses, o los sistemas hidráulicos que construyeron los griegos para fascinar a los asoradores de los templos. También aparecían autómatas, o máquinas semejantes a personas, en los relojes de las iglesias mediavales, época en la cual los relojeros eran famosos por sus ingeniosas criaturas mecánicas.



Figura 2.1: Reloj en la catedral de Wells, 1175

No obstante, el control por realimentación, la división del trabajo en tareas más pequeñas que pudieran ser realizadas por máquinas y el desarrollo de herramientas especializadas fueron los ingredientes esenciales en la automatización de las fábricas en el siglo XVIII. A medida que avanzaba la tecnología, acciones como poner tapones a las botellas o verter caucho líquido en moldes para neumáticos ya eran realizadas por máquinas especializadas. Sin embargo, estas máquinas aún no poseían la versatilidad del brazo humano, y no podían alcanzar objetos alejados y colocarlos en una posición determinada.

En la década de 1890, Nikola Tesla, visionario que escribió sobre mecanismos inteligentes que fuesen tan capaces como los humanos, ya construía vehículos controlados a distancia por radio.

Las máquinas más próximas a lo que hoy entendemos por robots fueron los teleoperadores, utilizados en la industria nuclear para la manipulación de sustancias radiactivas. Básicamente eran servomecanismos que, mediante sistemas de control mecánicos, repetían las operaciones que simultáneamente estaba realizando un operador.

Como muchos otros inventos a lo largo de la historia, los primeros trabajos que llevan a los robots industriales fueron iniciados durante o justo después de una guerra, en este caso inmediatamente después de la Segunda Guerra Mundial. A finales de los 40 se inician pro-

gramas de investigación en los laboratorios de *Oak Ridge* y *Argonne National Laboratories* para desarrollar manipuladores mecánicos con el fin de operar con elementos radiactivos. Eran manipuladores del tipo “maestro-esclavo”, diseñados para que produjeran fielmente los movimientos de brazos y manos realizados por un operario.

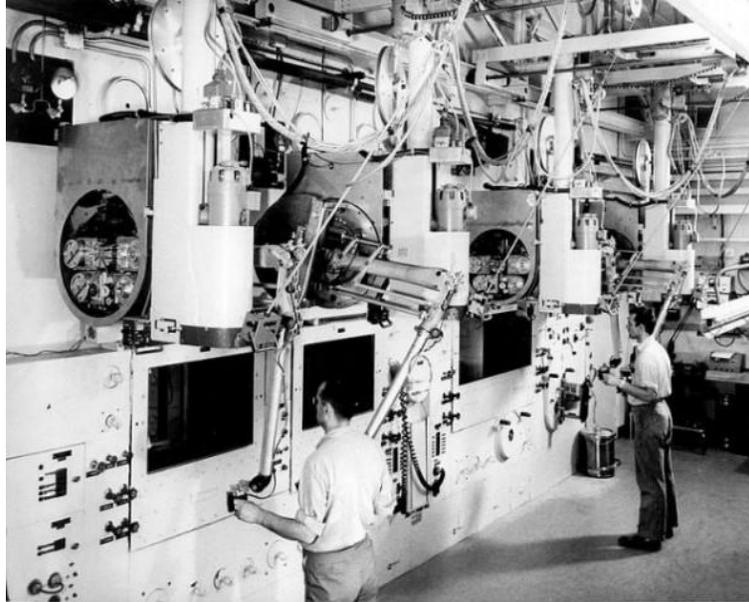


Figura 2.2: Manipulador tipo maestro-esclavo

En 1954 George C. Devol desarrolló un dispositivo de transferencia programada articulada (así lo bautizó él), y en 1958, junto con Joseph F. Engelberger, contruyeron un robot al que llamaron *Unimate*. Era un dispositivo que utilizaba un computador junto con un manipulador que conformaban una máquina que podía ser “enseñada” para la realización de tareas variadas de forma automática. Fué en 1962 cuando el primer *Unimate* se instaló a modo de prueba en una planta de *General Motors* para funciones de manipulación de piezas y ensamblaje, con lo que pasó a convertirse en *el primer robot industrial*.

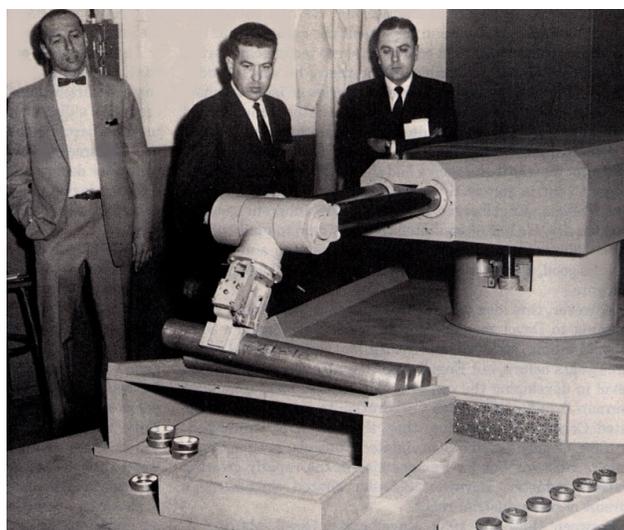


Figura 2.3: Primer robot *Unimate*

Este punto se puede considerar como el inicio de la era robótica tal como la conocemos hoy, mediante la utilización de los robots programados, una nueva y potente herramienta de fabricación.

En vistas a una mayor flexibilidad, en la época de los 60 nace la realimentación sensorial, con la que las manos robóticas eran capaces de detectar un objeto mediante un sensor de presión que proporcionaba una señal de realimentación al motor.

Es en la época de los 70 cuando aparecen las primeras aplicaciones industriales en Europa, y cuando la investigación en la robótica se centra en gran parte en el uso de sensores externos para su utilización en tareas de manipulación. Es en estos años cuando se consolida definitivamente la presencia de robots en las cadenas de montaje y plantas industriales.

### 2.1.3. El uso del robot

Los robots son, por lo general, sistemas electromecánicos que normalmente van conducidos por programas de una computadora o por circuitos eléctricos. Hoy en día los robots se utilizan para la automatización de producción de gran volumen, como por ejemplo, en la industria automovilística. Para obtener suficiente precisión en el proceso de montaje, los robots tienen que calibrarse a mano. Esto es un trabajo muy laborioso y no es rentable para producción en series de corto periodo, aunque esto está cambiando en la última década con los nuevos modelos de robot.

Como es bien sabido, los robots son capaces de repetir el mismo movimiento durante largos periodos de tiempo. De hecho, hoy en día los robots tienen una precisión de ca.  $\pm 1 \text{ mm}$ , la cual suele ser demasiado baja para utilizarla sin usar un manual de calibración para cada operación. Esto, combinado con la gran velocidad de ejecución y el hecho de que no requieren tiempos de descanso, hacen que cada vez más los robots esten relegando a los humanos a un segundo plano.

Por ello, en la empresa Inpro electric España, y debido a su gran volumen de producción, se utilizan este tipo de técnicas. No obstante, es necesario introducir los datos del posicionamiento del robot a mano una vez este está calibrado. Esto ralentiza el proceso ya que hay que hacerlo para cada uno de los robots que intervienen en el proceso.

Este proyecto se centra en automatizar el proceso de medición y calibración del robot junto con el movimiento del mismo, para reducir en la medida de lo posible el tiempo de espera mencionado anteriormente.

Su constitución física se asemeja mucho a la anatomía de las extremidades superiores del cuerpo humano en cuanto a movilidad de articulaciones se refiere. Esta imitación al cuerpo humano es por la gran variedad de movimientos que podemos realizar nosotros con cada articulación.

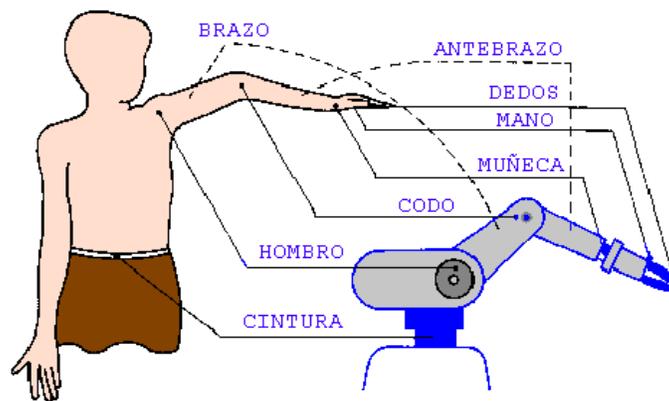


Figura 2.4: Semejanza del robot industrial con el cuerpo humano

Era muy común no hace mucho tiempo ver fábricas donde la mayoría de los robots eran de 5 grados de libertad (gdl). En la actualidad prácticamente todos son de 6 gdl, incluso muchas empresas han dejado de construir los de 5 gdl. Es más, se puede añadir un séptimo gdl de forma sencilla, y consiste en que el robot se pueda mover de su posición en un instante de tiempo, por ejemplo, mediante una cinta transportadora.

#### 2.1.4. Tipos de robot

Existen cientos de tipos de robots dependiendo del uso que se le de y las habilidades que posea. No obstante, a continuación se describen los tipos de robots articulados que hay en la actualidad, dado que el fin de este proyecto es la mejora de un proceso industrial en el cual se utilizan este tipo de robots.

##### 2.1.4.1. Robots antropomórficos

Son robots manipuladores con forma de brazo, por lo general con 6 o más grados de libertad (gdl) y que suelen incorporar una muñeca articulada para controlar la orientación de la herramienta (normalmente con 3 gdl) y el resto de articulaciones para posicionar la herramienta en el espacio Cartesiano.



Figura 2.5: Robot antropomórfico

**VENTAJAS:**

- Se asemeja al brazo humano (tareas repetitivas y pesadas).
- Variación de la posición y orientación (muchos grados de libertad).
- Alta maniobrabilidad.

**INCONVENIENTES:**

- Conllevan una alta inversión inicial.
- Modelo matemático complejo (posee muchas variables articulares, 5 o 6).
- Desplazamiento de personal.

**2.1.4.2. Robots SCARA**

De sus siglas en inglés, **S**elective **C**ompliant **A**ssembly **R**obot **A**rm, es un robot con 4 gdl con gran repetitividad, rapidez y gran carga de trabajo. Los dos primeros gdl permiten posicionar el TCP (Tool Center Point) del robot en el plano Cartesiano de trabajo (coordenadas Cartesianas X e Y), el tercer gdl permite orientar el TCP, mientras que el cuarto gdl permite controlar la altura del TCP (coordenada Cartesiana Z).



Figura 2.6: Robot SCARA

**VENTAJAS:**

- Gran precisión y alta velocidad
- Su montaje en pedestal solo requiere poco espacio y proporciona una forma fácil, sin obstáculos de montaje.
- Es generalmente más rápido y más limpio que los sistemas cartesianos.

**INCONVENIENTES:**

- Puede resultar más caro que los sistemas cartesianos.
- El software de control requiere cinemática inversa para movimientos lineales interpolados.

### 2.1.4.3. Robots delta

Es un robot con tres brazos articulados (paralelos) que se unen en el TCP con juntas universales que mantienen la orientación de su extremo.



Figura 2.7: Robot delta

#### VENTAJAS:

- Presentan una alta rigidez, lo cual se traduce en mayores precisiones respecto a un robot tipo serie.
- Su arquitectura les permite alcanzar altas velocidades y aceleraciones, lo cual les permite realizar tareas industriales de manera más eficiente.
- La relación carga/potencia es alta ya que los accionamientos de potencia conectan directamente la base del robot al efector final, permitiéndoles manipular cargas superiores a su propio peso.

#### INCONVENIENTES:

- Cinemática compleja.
- Espacio de trabajo pequeño y difícil de definir.
- Alta exigencia en los algoritmos de control dada la alta dinámica que proporcionan estas estructuras.

## 2.2. Introducción a la cinemática del robot

En esta sección se van a tratar los procesos matemáticos necesarios para entender el trabajo realizado. La ubicación de un objeto en el espacio tiene 6 grados de libertad, 6gdl. Los tres primeros corresponden a las posición en el sistema Cartesiano de coordenadas  $(x, y, z)^T$ , mientras que los tres restantes representan la rotación alrededor de los ejes de coordenadas  $(\alpha, \beta, \gamma)^T$ . La ubicación de un objeto se suele visualizar como un sistema de coordenadas adjunto al objeto. Esto hace posible tratar la ubicación del objeto como una transformación de un sistema de coordenadas a otro. Representaremos la transformación del sistema de

coordenadas A a B como  $T_B^A$ . Esta transformación requiere que se cumpla la siguiente relación:  $A = T_B^A B$ .

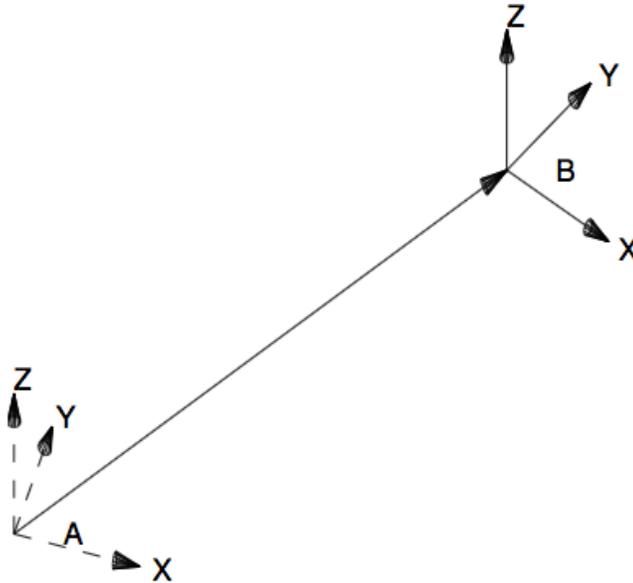


Figura 2.8: Sistemas de referencia

### 2.2.1. Traslación

Todas las traslaciones se dan en un sistema de coordenadas dado. Una traslación es un vector con tres componentes

$$v = \begin{pmatrix} x \\ y \\ z \end{pmatrix} \quad (2.1)$$

El vector  $v$  también puede ser parte de una transformación homogénea. La traslación del vector  $A$  con el vector  $v$  es una suma de vectores de la forma  $A + v$ .

### 2.2.2. Rotación

Existen muchas formas de representar y calcular una rotación, teniendo cada una de ellas sus propios pros y contras. Ninguna de estas representaciones es mejor que las demás en todas las situaciones. Se explicará en este trabajo una selección de las mencionadas representaciones.

#### 2.2.2.1. Matrices de rotación, R

Las matrices de rotación son fáciles de calcular y poseen propiedades simples, las cuales causan afición entre los distintos métodos de rotación. La matriz se puede parametrizar con 3 números  $(\alpha, \beta, \gamma)$ . Entre sus propiedades, cabe destacar que mantienen su longitud al multiplicarse, y todas ellas son ortonormales, por lo que tienen la siguiente propiedad

$$R^T = R^{-1} \quad (2.2)$$

la cual es muy útil a la hora de guardar recursos computacionales. Es importante recordar que la multiplicación de las matrices no es conmutativa, es decir  $AB \neq BA$ .  $ABv$  es un vector  $v$  rotado primero por  $B$  y luego por  $A$ . Haciendo referencia a los elementos individuales de la matriz se utilizarán los siguientes índices

$$R = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix} \quad (2.3)$$

### 2.2.2.2. Cuaterniones

Un cuaternión es un vector cuadrimensional. Si la longitud del cuaternión es igual a la unidad se puede utilizar para describir rotaciones según Craig. Notar que el elemento 4 en Craig se llama elemento 0 en este trabajo según la Tabla 2.1. Esto es porque ABB utiliza la notación previa.

$$\begin{aligned} q_0 &= \epsilon_1 \\ q_1 &= \epsilon_2 \\ q_2 &= \epsilon_3 \\ q_3 &= \epsilon_4 \end{aligned}$$

Tabla 2.1: Numeraciones de cuaterniones, donde  $\epsilon$  representa Craig y  $q$  es la notación en este trabajo

$$\begin{aligned} q_0 &= \cos\left(\frac{\theta}{2}\right) \\ q_1 &= x \sin\left(\frac{\theta}{2}\right) \\ q_2 &= y \sin\left(\frac{\theta}{2}\right) \\ q_3 &= z \sin\left(\frac{\theta}{2}\right) \end{aligned}$$

Tabla 2.2: Definición de cuaternión

### 2.2.3. Cinemática directa del brazo del robot manipulador

El estudio de la cinemática de un robot se puede realizar de dos formas. La primera es el método directo, en el que se intenta determinar la posición y la orientación del extremo final de la cadena cinemática del robot conocidos los valores de las coordenadas articulares y las características geométricas del robot. El segundo método consiste en hallar las variables articuladas conocida la posición y la orientación del extremo de la cadena cinemática. Se procede a estudiar el método directo.

Básicamente el problema cinemático directo se plantea en términos de encontrar una matriz de transformación que relaciona el sistema de coordenadas ligado al cuerpo en movimiento respecto a un sistema de transformación homogénea 4x4, la cual incluye las operaciones de traslación y la orientación como se ha mencionado anteriormente.

La matriz de transformación homogénea es una matriz de 4x4 que transforma un vector expresado en coordenadas homogéneas desde un sistema de coordenadas hasta otro sistema de coordenadas. Tiene la siguiente estructura:

$$T = \begin{pmatrix} \text{matriz de rotación} & \text{vector de posición} \\ f_{1x3} & \text{escalado} \end{pmatrix} = \begin{pmatrix} n_x & s_x & a_x & p_x \\ n_y & s_y & a_y & p_y \\ n_z & s_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} n & s & a & p \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.4)$$

donde los vectores  $n, s, a$ , son vectores ortogonales unitarios y  $p$  es un vector que describe la posición  $x, y, z$  del origen del sistema actual respecto del sistema de referencia. Se adjunta un gráfico con un ejemplo de la matriz de transformación homogénea.

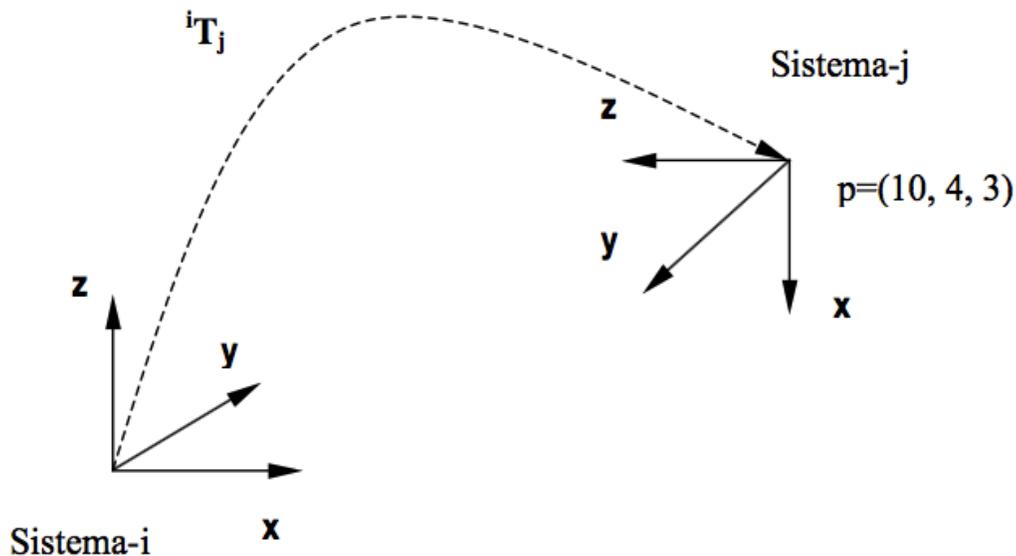


Figura 2.9: Interpretación geométrica de la matriz de transformación homogénea

$${}^i T_j = \begin{pmatrix} 0 & 0 & -1 & 10 \\ 0 & -1 & 0 & 4 \\ -1 & 0 & 0 & 3 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.5)$$

Si se analizan las columnas de la submatriz de rotación de la matriz de transformación homogénea  ${}^i T_j$ , un observador localizado en el origen del *sistema*  $- i$  puede ver como están orientados los ejes  $x, y, z$  del *sistema*  $- j$ , además de observar como se ha desplazado en coordenadas cartesianas el origen del *sistema*  $- j$  respecto del origen del sistema de referencia con la información del vector posición.

### 2.2.3.1. Representación de Denavit-Hartenberg

La representación de D-H se aplica a robots de cadena cinemática abierta y consiste en una serie de reglas para colocar los sistemas de referencia de cada eslabón del robot. Antes de aplicar el método de D-H es importante tener en cuenta los siguientes aspectos:

- Se parte de una configuración cualquiera del robot, preferiblemente en una posición sencilla de analizar.

- Se numeran los eslabones, asignando el 0 para la base y n-1 para el último eslabón, siendo n el número de grados de libertad del robot.
- El sistema de coordenadas ortonormal dextrógiro de la base  $(x_0, y_0, z_0)$  se establece con el eje  $z_0$  localizado a lo largo del eje de movimiento de la articulación 1 y apuntando hacia fuera del hombro del brazo del robot.
- El sistema de referencia de cada eslabón se coloca al final del mismo, en el extremo de la articulación a la cual está conectado el eslabón siguiente.
- El ángulo ó desplazamiento de cada eslabón siempre se mide tomando como base el sistema de referencia del eslabón anterior.
  - El eje  $z_i$  del sistema de referencia debe quedar alineado a lo largo de la articulación.
  - El eje  $x_i$  debe colocarse con orientación normal al plano formado por los ejes  $z_{i-1}$  y  $z_i$ .

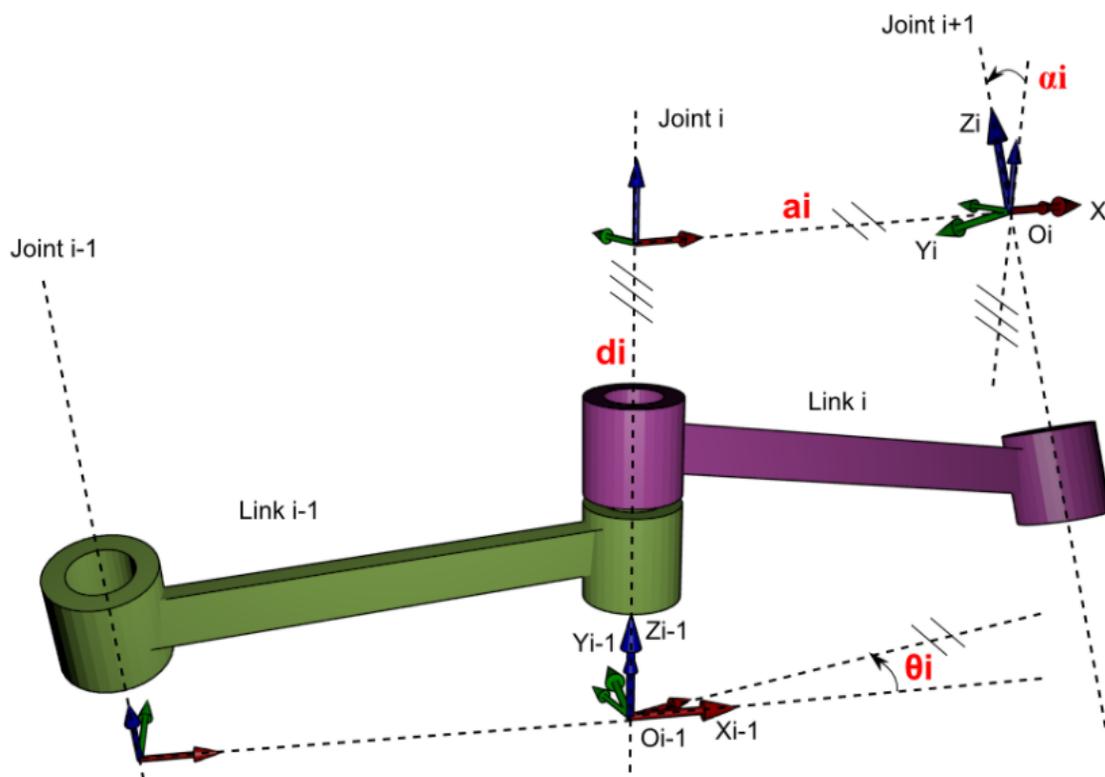


Figura 2.10: Sistemas de coordenadas para la convención de D-H

Cada sistema de coordenadas se establece sobre las siguientes reglas:

- $\theta_i$ : Es el ángulo de la articulación desde el eje  $x_{i-1}$  hasta el eje  $x_i$  medido respecto del eje  $z_{i-1}$ , usando la regla de la mano derecha.
- $d_i$ : Es la distancia medida desde el origen del sistema i-1, a lo largo del eje  $z_{i-1}$  hasta la intersección del eje  $z_{i-1}$  con el eje  $x_i$
- $a_i$ : Es la distancia de separación entre los orígenes de los sistemas de referencia i-1 e i, medida a lo largo del eje  $x_i$  hasta la intersección con el eje  $z_{i-1}$  (o la distancia más corta entre los ejes  $z_{i-1}$  y  $z_i$  cuando no se interceptan).
- $\alpha_i$ : Es el ángulo que separa los ejes  $z_i$  y  $z_{i-1}$ , medido respecto del eje  $x_i$ .



---

# Capítulo 3

## Descripción del proceso

En este capítulo se pretende describir el proceso completo del presente proyecto, analizando cada parte en profundidad.

Primero, se utiliza un laser tracker que puede medir tanto la posición como la orientación con gran precisión para monitorizar la posición del robot. Se analiza este proceso en detalle en la sección 3.1. Una vez realizadas las mediciones, y junto con los softwares que más adelante se estudiarán, ya tenemos el robot ubicado en el espacio 3D.

Para la parte teórica, hay que programar el PLC y el robot para que ejecuten la acción de soldadura, junto con las normas de seguridad correspondientes. Cuando esto está listo, se conecta el robot al PLC mediante cables de entradas y salidas, los cuales transfieren la información directamente de uno al otro.

Finalmente se ejecutan ensayos y simulaciones para comprobar el correcto funcionamiento de todo el proceso.

### 3.1. Puesta a punto del laser tracker

Antes de iniciar cualquier acción es necesario montar el laser tracker para poder medir los puntos deseados sobre la mesa de trabajo. El modelo utilizado es el Leica AT402, del cual se realiza un estudio más profundo en la sección 5.1.7.

Se coloca el mandril sobre el trípode y se ancla bien fuerte. Después se pone directamente el laser tracker sobre el mandril, y se cierra la sujeción de seguridad que une a ambos. Una vez que todo está bien sujeto se procede a conectar todos los cables.

El primero es el puerto remoto de temperatura ambiente. Se conecta una punta al laser y se deja la otra al aire para que pueda calcular la temperatura y la humedad de la sala para optimizar el funcionamiento. El segundo cable es el de ethernet (este no es necesario si se utiliza Wifi). Se conecta una punta y la otra se introduce directamente en el ordenador para que reciba la información. Por último se conecta el cable de la alimentación, el cual provee potencia al laser tracker. Se conecta la otra punta del cable de alimentación a la fuente de alimentación. En este punto ya está todo listo para encender el laser.

Se pulsa el botón de encendido, situado en la parte trasera. Trás esto, se abre el software que se vaya a utilizar para el seguimiento mediante CAD del laser y se conecta al tracker mediante la dirección IP. Una vez el programa termine la puesta en marcha estará listo para su utilización.

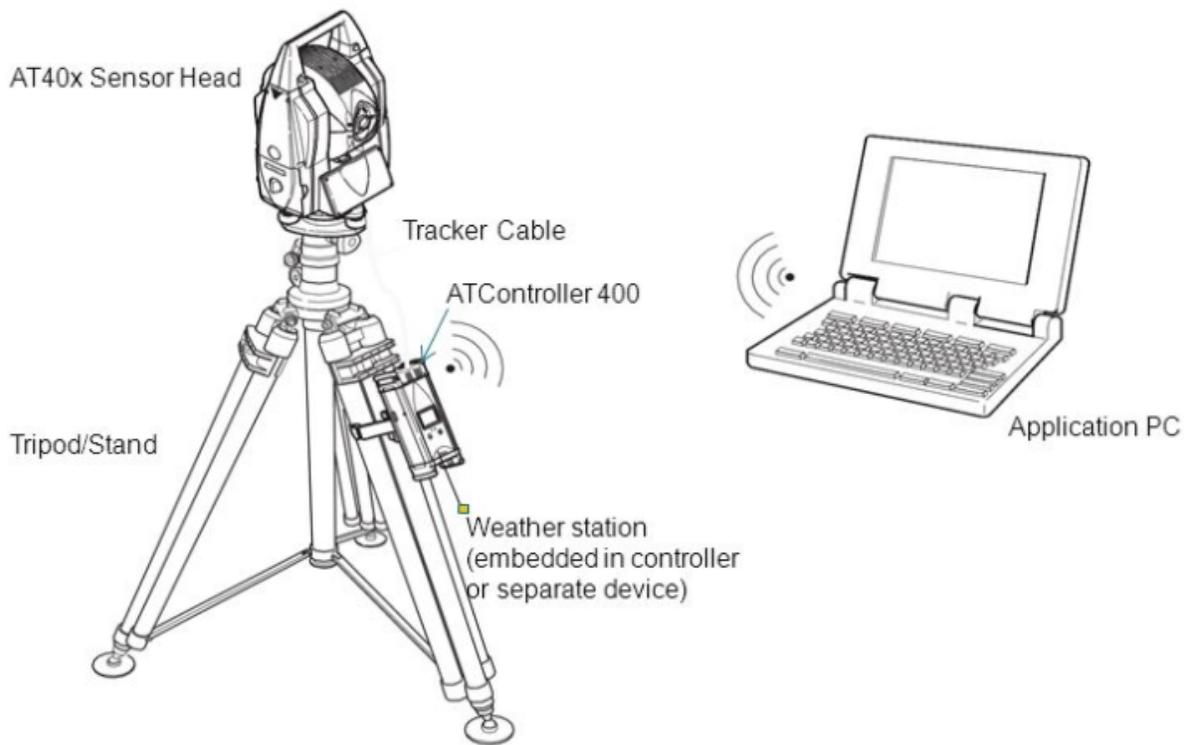


Figura 3.1: Componentes del laser tracker Leica AT402

### 3.2. Medición de la mesa y la pieza

Se procede a describir como sería el proceso de mediciones reales en fábrica para referenciar el robot y las piezas.

El primer paso es medir la estación (mesa) y referenciarla al  $0_{coche}$ . Para ello, se miden los puntos nominales. Una vez medidos, se procede a medir los mismos puntos con el laser. Es necesario, para cada punto, hacer 4 mediciones a la misma altura (eje  $z$ ) para formar un plano alrededor del punto a medir. Después se mide el punto central y finalmente un punto de compensación, el cual se mide desplazando el reflector en el eje  $z$ . Como su puede ver, para cada punto distinto son necesarios 6 mediciones con el laser. Una vez obtenidos se comparan con los valores nominales, y la desviación entre ambos debería ser como máximo de  $0,1mm$ . Con esto ya tenemos la mesa referenciada con el  $0_{coche}$ .

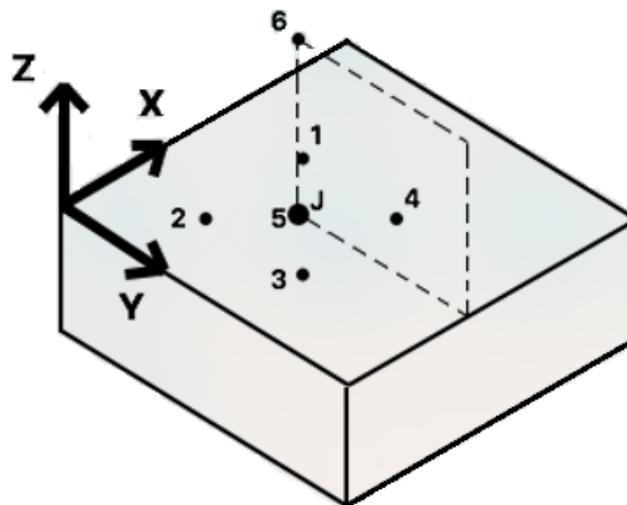


Figura 3.2: Medición de un punto de la mesa

El paso que sigue consiste en medir el TCP (tool center point) del robot. Este es el

punto focal de la pinza o herramienta acoplada en el brazo robot, es decir, el punto del cual se relaciona todo posicionamiento del robot.

Después hay que medir el robot y referenciarlo también al  $0_{coche}$ . Se crea un programa con 8 movimientos distintos del robot, en el cual se busca la máxima movilidad de los 6 ejes. Para cada movimiento, hay que medir la posición con el laser. Por seguridad, el programa te obliga a hacer dos mediciones por posición, para comparar ambas y asegurar que la medición es correcta. Añadimos ahora las coordenadas del robot al software, y alineamos los puntos medidos del robot con los del laser.

Ahora que se tienen todas las mediciones requeridas, hay que exportar la nueva base, trasladándola  $(x, y, z)$  y rotándola  $(\alpha, \beta, \gamma)$ .

Finalmente se prueba la nueva base en el robot. Se añaden las coordenadas de la nueva base al robot. Se mueve el robot a 3 posiciones distintas y se miden con el laser estas posiciones. Para cada posición, se comparan las coordenadas del laser y las del robot, las cuales deberían ser iguales.



---

# Capítulo 4

## Soluciones alternativas

En este apartado se van a analizar posibles alternativas tanto en hardware como en software para la realización del trabajo. En él, se expondran los diferentes apartados que conforman el hardware y las alternativas para ellos, realizando un análisis de las posibles ventajas e inconvenientes de cada uno. Se procederá de la misma forma para el software.

### 4.1. Hardware

La palabra hardware hace referencia en el ámbito de la informática a las partes físicas tangibles de un sistema informático; sus componentes eléctricos, electrónicos, electromagnéticos y mecánicos. Entre ellos también se incluyen todo tipo de cables, gabinetes o cajas y cualquier elemento físico involucrado. Esta definición se extrapola también a la vida diaria y la tecnología, ya que entre los componentes del hardware podemos encontrar desde los aparatos electrónicos, maquinas y herramientas utilizadas para la realización de una tarea, hasta robots en el ámbito industrial como es el caso de este proyecto.

#### 4.1.1. PLC

Un controlador lógico programable, más conocido como PLC (Programmable Logic Controller), es una computadora utilizada en muchos ámbitos de la ingeniería, la cual sirve para automatizar procesos electromecánicos, tales como el control de la maquinaria en las cadenas de producción.

Su diferencia con las computadoras de propósito general es que el PLC está diseñado para múltiples señales de entrada y salida, rangos variables de temperatura, inmunidad al ruido electrónico y resistencia al impacto y a la vibración. Son estos factores los que hacen que los PLC sean elegidos ante las computadoras de sobremesa para realizar las tareas en las plantas de producción, ya que los ordenadores convencionales no están diseñados para soportar estas condiciones. Se pueden considerar como sistemas de tiempo real, ya que los resultados de salida son producidos en respuesta a las diferentes condiciones de entrada en ese instante.

Entre su amplio abanico de funciones cabe destacar el control de movimiento, control de procesos, control del relé secuencial y la comunicación por red. Sus capacidades básicas, como pueden ser de almacenamiento, procesamiento de información y comunicación son equivalentes a las de las computadores de escritorio para los PLC modernos.

En cuanto a su estructura interna, está formado básicamente por la unidad central de proceso (CPU), y las interfaces de entrada y salida. Si nos adentramos más en la estructura, la CPU, que podemos decir que es el cerebro de nuestro autómatas programable, está compuesta por el procesador y la memoria. El primero se encarga de ejecutar el programa, que se encuentra almacenado en la memoria.

Las ventajas de utilizar este tipo de computadoras es la posibilidad de ahorrar tiempo en la elaboración de los proyectos, incluyendo el hecho de poder realizar modificaciones sin añadir ningún coste. Además, son baratos de mantener y de tamaño reducido, por lo que permiten ahorrar dinero en mano de obra. Otra ventaja fundamental que ofrecen es la posibilidad de controlar más de una máquina con el mismo equipo, algo que hace que los procesos de producción sean cada vez más eficientes. En contra, tan solo el hecho de necesitar gente adiestrada y técnicos calificados para su correcto funcionamiento.

#### **4.1.1.1. Schneider y Omron**

Las principales alternativas en cuanto a la marca del PLC barajadas por la empresa han sido Schneider y Omron. Sin duda ambas ofrecen un amplio abanico de posibilidades, pero el factor determinante ha sido la trayectoria. La empresa siempre ha trabajado con Siemens y así ha seguido en el proyecto actual. No obstante, se analizarán ambas opciones a continuación.

En cuanto a este proyecto, la disponibilidad ha sido lo que ha marcado la elección. Por suerte, tanto la Universidad Politécnica de Valencia como la Universidad de Valencia disponen de PLCs de la marca Siemens, coincidiendo así con los utilizados por la empresa, por lo que no ha sido necesario utilizar ningún otro.

#### **Schneider**

El modelo barajado de la marca Schneider es el Modicon M221. Está especialmente diseñado para máquinas automatizadas con arquitectura de control cableadas, tal y como se realiza en el presente proyecto. Además es rápido para aplicaciones con requisitos de rendimiento de nivel alto, como es el proceso de soldadura requerido. A continuación se presenta el modelo en imagen y sus características.



Figura 4.1: Schneider Modicon M221

<b>Serie del Fabricante</b>	Modicon M221
<b>Número de E/S</b>	16
<b>Tipo de Salida</b>	Digital
<b>Capacidad del Programa</b>	10.000 pasos
<b>Interfaz de Programación</b>	Mini USB
<b>Entradas/Salidas Máximas</b>	16
<b>Tipo de Montaje</b>	Montaje en panel
<b>Idioma de Programación Utilizado</b>	CoDeSys

Tabla 4.1: Características técnicas del Modicon M221

## Omron

En lo que se refiere a Omron, se ha estudiado la opción del CP1L. Posee características similares al Modicon M221, como pueden ser los 10.000 pasos de capacidad de programa, pero ofrece una mayor cantidad de entradas y salidas sin necesidad de módulos adicionales. En cambio, la interfaz de programación es solo por ordenador, lo cual puede ser perjudicial si se desea reprogramar rápidamente en planta. Se muestra a continuación, como con el Modicon M221, el modelo en imagen y sus características.



Figura 4.2: Omron CP1L

<b>Serie del Fabricante</b>	CP1L
<b>Número de Entradas</b>	18 (dc)
<b>Categoría de Tensión</b>	85 – 264 V ac
<b>Número de Salidas</b>	12 (relé)
<b>Tipo de Red</b>	USB
<b>Capacidad del Programa</b>	10.000 pasos
<b>Interfaz de Programación</b>	Ordenador
<b>Idioma de Programación Utilizado</b>	Ladder Logic
<b>Corriente de Salida</b>	2 A
<b>Dimensiones</b>	130 x 110 x 85 mm

Tabla 4.2: Características técnicas del CP1L

#### 4.1.2. Robot

En lo que se refiere al robot, la decisión también estaba ya fija dada la disponibilidad que ofrecía tanto la empresa como la universidad. No obstante, a pesar de haber trabajado en muchos proyectos con robots de la marca KUKA, en el presente proyecto la empresa ha utilizado robots ABB. También coinciden estos con el disponible en la universidad. A pesar de esto, se procede a analizar el posible sustituto de robot utilizado en el proyecto actual.

El modelo es el KR 30-3 C de KUKA. Se caracteriza por ofrecer una calidad muy alta con los tiempos de ciclo más cortos y con una precisión de repetición máxima, ideal para el proceso de soldadura. Además, está diseñado para garantizar su durabilidad, aspecto importante si consideramos la cantidad de veces que será utilizado a lo largo de todos los proyectos distintos que se presenten.



Figura 4.3: Brazo robótico KUKA KR 30-3 C

<b>Modelo</b>	KR 30-3 C
<b>Categoría</b>	Carga mediana (30-60 kg)
<b>Carga</b>	30 kg
<b>Carga total nominal</b>	65 kg
<b>Máx. alcance</b>	2033 mm
<b>Número de ejes controlables</b>	6
<b>Repetibilidad de posición</b>	$\pm 0,06$ mm
<b>Temperatura ambiente</b>	10 °C a + 55 °C
<b>Unidad de control</b>	KR C4
<b>Tipo de protección</b>	IP 64
<b>Tipo de protección muñeca central</b>	IP 65

Tabla 4.3: Características técnicas del KUKA KR 30-3 C

Se muestra a continuación el rango y la capacidad de movimientos del robot KUKA KR 30-3 C.

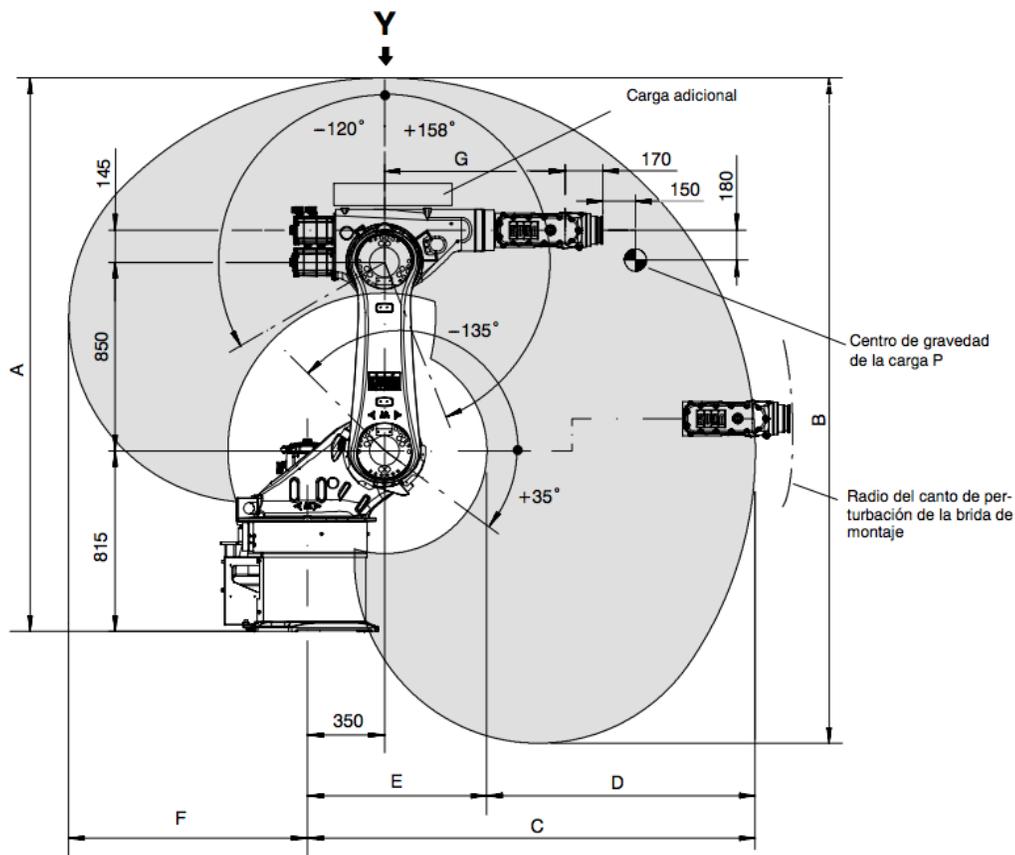


Figura 4.4: Rango de movimientos robot KUKA KR 30-3 C

A	B	C	D	E	F	G
2498	3003	2033	1218	815	1084	820

Tabla 4.4: Valores del robot KUKA KR 30-3 C

## 4.2. Software

El software se refiere al equipo lógico o soporte lógico de un sistema informático, que comprende el conjunto de los componentes lógicos necesarios que hacen posible la realización de tareas específicas, en contraposición al hardware, que como hemos visto antes, se refiere a los componentes físicos.

El software no solo incluye los programas de computación en sus distintos estados: código fuente, binario o ejecutable; también su documentación, los datos a procesar e incluso la información de usuario. Básicamente, abarca todo lo intangible relacionado a un proceso específico.

Para el desarrollo de éste, primero habrá que hacer uso de la ayuda de algún tipo de herramienta gráfica para describir el proceso, para posteriormente poder implementarlo con los algoritmos adecuados según el lenguaje de programación.

Podemos definir el lenguaje de programación como un lenguaje formal diseñado para realizar procesos que pueden ser llevados a cabo por máquinas como las computadoras. En otras palabras, no es más que el conjunto de reglas que definen un programa y el significado de sus elementos.

Existen numerosos lenguajes de programación, de los cuales las ventajas o inconvenientes dependerán de la utilidad que le de el usuario y, sobre todo, del conocimiento del programador.

Entre los lenguajes de programación que existen, podemos destacar los gráficos, literales y mixtos.

- Lenguajes gráficos.** Su herramienta más utilizada son los bloques lógicos, los cuales tienen cierta similitud a los lenguajes literales de alto nivel, pero con la ventaja de ser mucho más visuales, factor que hace que su comprensión sea más fácil. El más común dentro de los autómatas programables es el lenguaje Ladder (diagrama de contactos o escalera), ya que está basado en los esquemas eléctricos de control clásicos. Se presenta un ejemplo para un función concreta.

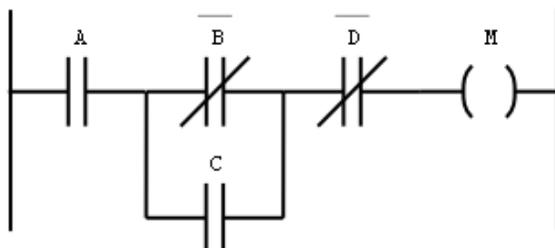


Figura 4.5: Lenguaje Ladder para la función  $M = A(B'+C)D'$

- Lenguajes literales.** Estos están formados por instrucciones elementales, que normalmente se representan a través de una secuencia de texto. Dependen mucho del lenguaje utilizado y de su complejidad. Entre sus elementos más comunes se pueden destacar los más sencillos como pueden ser las puertas lógicas, como las AND, OR o NOR, o los más complejos, como serían los bucles WHILE y FOR. Este tipo de lenguaje se encuentra en softwares como C++, Matlab, etc, y es el más común para programación de máquinas que no utilicen PLC, como pueda ser una máquina expendedora o una agenda electrónica.

```

for time=1:10 % In minutes
    index=1;
    for delta=-100:10:100
        d= Avoidance(Heading1,Heading2,TAS1,TAS2,time*60,delta);
        current(index)=d; %Filling current vector for plotting
        index=index+1;
    end
    t(time,1:21)=current;
end

```

Figura 4.6: Bucle FOR en Matlab

- Lenguajes mixtos.** Estos componen la mezcla de los dos anteriores. Su intención es combinar la claridad de los lenguajes gráficos con la sencillez de los literales.

Una vez analizado el tipo de lenguaje con el que se va a programar, se debe hacer una descripción de aquello que debe hacer el programa. Aquí se definirán las acciones y el orden en el que deben ser ejecutadas. Para realizarlo, a continuación se analizan las diversas estrategias que se pueden utilizar.

- Diagrama de flujo.** Es una representación gráfica del algoritmo que se va a implementar. Suele tener un único punto de inicio y de fin, pero esto puede variar siempre que el diagrama siga una lógica y no quede abierto. En la imagen que se muestra a continuación podemos ver un diagrama de flujo en el que al iniciar el diagrama, se ejecuta la acción A. Trás esto, si la condición B es falsa, se acabará el programa. En caso de ser cierta, se dará paso a la acción D seguida de la C. Del suceso C el flujo volverá a la condición B, y esto se repetirá hasta que ésta sea falsa y se acabe la secuencia.

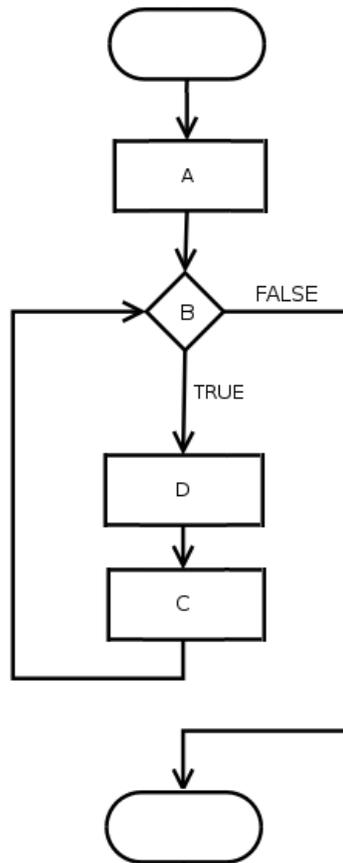


Figura 4.7: Diagrama de flujo

- Descripción gráfica.** El concepto es muy similar al diagrama de flujo, donde se muestra de forma gráfica el comportamiento del programa. Sin embargo, la diferencia reside en que en éste se pueden ejecutar distintos flujos de forma simultánea. Como ejemplo se pone el GRAFCET que se detallará más adelante, en la sección 5.2.1.
- Descripción literal.** Se basa en la definición del programa mediante palabras. Puede parecer sencillo, pero a menos que la descripción del programa sea detallada, ciertos aspectos del funcionamiento pueden resultar ambiguos. Por ello, a menos que el proceso a ejecutar sea simple, no es conveniente la realización de este tipo de descripciones.

#### 4.2.1. Polyworks

En un principio, la empresa trabajaba con Polyworks para realizar la medición de la mesa. Con él se puede visualizar el entorno en el propio ordenador mediante una simulación del lugar del trabajo. Esto es muy eficiente para resolver problemas de mediciones en el cual desconoces el valor nominal de lo que pretendes medir, ya que puedes medirlo directamente con el propio software. Una vez medidos los puntos en el espacio 3D con el reflector y el laser tracker, el programa los recibe instantaneamente y pueden ser visualizados en la pantalla del ordenador. Cuando los puntos nominales y los medidos con el laser coinciden, el software

realiza los cálculos oportunos para trasladar el sistema de referencia del  $0_{coche}$  a la base del robot.

El problema con este programa es que es necesario medir el TCP con el robot. Esto hace que a medida que el brazo se estira, aumenta el error en la medición, llegando a un error inaceptable en zonas alejadas de la mesa. El principal factor por el que esto ocurre es el momento generado alrededor del brazo robótico cuando éste está extendido, ya que la fuerza de la gravedad lo desplaza más cuanto más estirado está.

Como veremos más adelante, este error puede corregirse con el software empleado en el proyecto.

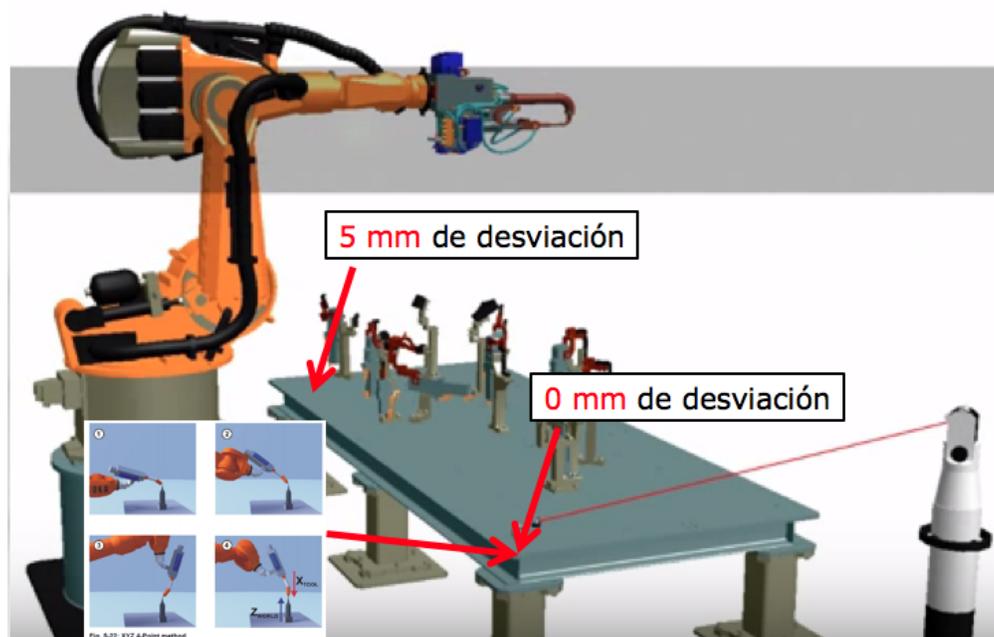


Figura 4.8: Error generado con Polyworks

En la imagen podemos ver dos puntos medidos en la mesa de trabajo. El primero muestra una desviación de 0 mm mientras que el segundo de 5 mm. Esto es por dos motivos: el primero es debido al momento generado mencionado anteriormente; el segundo es por la posición del láser tracker con respecto a los puntos medidos. Vemos que a medida que se aleja el punto de referencia, mayor es el error generado al medir el punto con el TCP del robot. Esto se traduce en unos valores inaceptables en muchas mesas de trabajo debido a la distribución de los elementos, y consecuentemente a la búsqueda de una alternativa de medición o un software nuevo que reduzca este error hasta los límites aceptables.



---

# Capítulo 5

## Solución adoptada

En este capítulo se pretende estudiar en profundidad la solución adoptada en el presente proyecto. Como en el capítulo anterior, se analiza tanto el hardware como el software empleados, analizando cada uno de ellos en detalle y el motivo por el cual se ha utilizado.

### 5.1. Hardware

#### 5.1.1. CPU

Para la realización del proyecto se ha utilizado una SIMATIC S7-1200, más concretamente el modelo 1214 AC/DC/RLY 6E57 214-1BG40-0XB0 de Siemens. Dispone de 14 entradas digitales, 10 salidas digitales y 2 entradas analógicas. La tensión de alimentación oscila entre 20,4V y 28,8V. Aparte de ofrecer estas características, también posee un puerto PROFINET para su comunicación y programación, el cual hace que su acceso sea muy fácil y cómodo. Otro punto importante que posee este modelo son los 100kB de memoria para guardar datos y programas. Este último punto hace que su utilidad en la industria sea óptima en múltiples tareas.

<b>Tipo</b>	SIMATIC S7-1200 CPU
<b>Clase</b>	AC/DC/RLY
<b>Entradas digitales</b>	14 (6 contadores de gran velocidad)
<b>Entradas analógicas</b>	2 (0-10V)
<b>Salidas digitales</b>	10 (4;100 kHz tren de impulsos)
<b>Guardado de datos</b>	100 kB
<b>Voltaje de entrada</b>	24 V DC
<b>Rango admisible</b>	20,4...28,8 V DC
<b>Intensidad de entrada</b>	1 mA
<b>Salida de corriente permanente</b>	0,5 A
<b>Montaje</b>	Carril DIN 35 mm
<b>Dimensiones</b>	110 x 100 x 75 mm
<b>Rango Temp. ambiente</b>	-20 – 60 °C
<b>Reloj de tiempo real</b>	Si
<b>Comunicación</b>	PROFINET

Tabla 5.1: Características técnicas de la CPU



Figura 5.1: Vista frontal Siemens SIMATIC S7-1200



Figura 5.2: Vista trasera Siemens SIMATIC S7-1200



Figura 5.3: Vista lateral Siemens SIMATIC S7-1200

### 5.1.2. Módulos de entradas digitales

En este proyecto se ha trabajado con 2 módulos de entrada Siemens SM 1221 DC - 6ES7221-1BH32-0XB0. Con esto se añaden 32 entradas digitales del SIMATIC S7-1200 (16 por tarjeta). Además, cada una de ellas procesa señales de entrada de 24V.

<b>Tipo</b>	Módulo de entrada digital
<b>Clase</b>	SM 1221
<b>Entradas digitales</b>	16
<b>Voltaje de entrada</b>	24 V DC
<b>Rango admisible</b>	20,4...28,8 V DC
<b>Intensidad de entrada</b>	4 mA
<b>Conexión</b>	9-pin Sub-D conector
<b>Montaje</b>	Carril DIN 35 mm
<b>Dimensiones</b>	45 x 100 x 75 mm
<b>Rango Temp. ambiente</b>	-20 – 60 °C
<b>Protección</b>	IP20

Tabla 5.2: Características técnicas del módulo de entrada



Figura 5.4: Vista frontal Siemens SM 1221 DC



Figura 5.5: Vista trasera Siemens SM 1221 DC



Figura 5.6: Vista lateral Siemens SM 1221 DC

### 5.1.3. Módulo de salida digital

La función de los módulos de salida es inversa a las de entrada, es decir, añadir posibles salidas al CPU. En este caso, y a diferencia de los módulos de entrada utilizadas, el Siemens SM 1222 DC solo añade 8 salidas, no 16. Esto es más que suficiente debido a las características de este proyecto, donde las únicas salidas del PLC serán las entradas del robot. Este modelo tiene salidas con transistor con una corriente de salida de 0,5A a 24V DC.

<b>Tipo</b>	Módulo de salida digital
<b>Clase</b>	SM 1222
<b>Salidas digitales</b>	8 (Transistor)
<b>Voltaje de entrada</b>	24 V DC
<b>Rango admisible</b>	20,4...28,8 V DC
<b>Salida de corriente permanente</b>	0,5 A
<b>Conexión</b>	Conexión roscada
<b>Montaje</b>	Carril DIN 35 mm
<b>Dimensiones</b>	45 x 100 x 75 mm
<b>Rango Temp. ambiente</b>	-20 – 60 °C
<b>Protección</b>	IP20

Tabla 5.3: Características técnicas del módulo de salida

H



Figura 5.7: Vista frontal SM 1222 DC



Figura 5.8: Vista trasera SM 1222 DC



Figura 5.9: Vista lateral SM 1222 DC

#### 5.1.4. Controlador Robótico

El controlador empleado es el IRC5 de ABB. La empresa ha optado por éste en específico por la tecnología de control de movimiento que emplea, ya que le da más rendimiento al robot en términos de precisión, rapidez, programación y sincronización con algún dispositivo externo.



Figura 5.10: Controlador ABB IRC5

En cuanto a seguridad, ofrece una protección total. El sistema de seguridad está basado en un circuito de doble canal que se controla continuamente. Si cualquiera de los componentes falla, la alimentación eléctrica de los motores se interrumpe y se aplican los frenos.

El panel de control está formado por los siguientes 9 módulos que se describen a continuación.

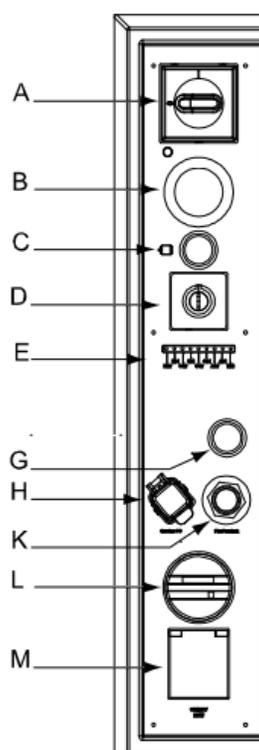


Figura 5.11: Módulos del controlador IRC5

Posición	Nombre
A	<i>Interruptor principal y control remoto de la alimentación de los módulos de accionamiento.</i>
B	<i>Paro de emergencia. Si está introducido, tire para liberarlo.</i>
C	<i>MOTORS ON.</i>
D	<i>Selector de modo de funcionamiento.</i>
E	<i>LEDs de la cadena de seguridad (opción).</i>
G	<i>Pulsador de hot plug de FlexPendant (opción).</i>
H	<i>Conexión de PC de servicio.</i>
K	<i>Conexión de FlexPendant.</i>
L	<i>Contador de tiempo de funcionamiento (opción).</i>

Tabla 5.4: Panel de control

### 5.1.5. FlexPendant

El FlexPendant es una unidad de operación manual que se utiliza para realizar tareas que implican al brazo robótico: ejecutar programas, activar el manipulador, modificar programas y así sucesivamente. A pesar de incluirlo en este apartado de hardware, se compone tanto de este último como de software, ya que es una computadora completa por si mismo. Se conecta al controlar mediante un cable y un conector integrados.



Figura 5.12: FlexPendant

Posición	Nombre
A	<i>Conector</i>
B	<i>Pantalla táctil</i>
C	<i>Botón de parada de emergencia</i>
D	<i>Palanca de mando</i>
E	<i>Puerto USB</i>
F	<i>Dispositivo habilitador</i>
G	<i>Lápiz óptico</i>
H	<i>Botón de reinicio</i>

Tabla 5.5: FlexPendant

Una de las ventajas que ofrece el FlexPendant es que es muy intuitivo, por lo que no se requiere ningún nivel previo de programación para poder utilizarlo. Además, con la palanca de mando o “joystick” se puede mover el manipulador hasta la posición deseada. Esto permite amenizar mucho los procesos, ya que no hay que preprogramar el robot para que vaya a un punto inicial por ejemplo, sino que simplemente llevándolo a mano y guardando esa posición es suficiente.

Para su uso, hay que presionar el dispositivo habilitador. Esto es un botón con tres posiciones: reposo, intermedia y final. Para que el FlexPendant funcione, hay que mantener este botón en la posición intermedia, quedando desactivado en cualquier otra posición. Esto es una medida de seguridad, ya que el FlexPendant está diseñado para ser utilizado en planta a lado de los robots. Esto es porque ante una situación de riesgo, se ha demostrado que el operador puede reaccionar de dos formas distintas. Una es apretando más fuerte de forma involuntaria el botón, mientras que la otra es soltándolo. Con esta medida de seguridad nos aseguramos de que pase lo que pase, el robot se parará.

### 5.1.6. Brazo robótico

En cuanto al brazo robótico, en la propia empresa se utilizan diferentes modelos, dependiendo de la tecnología que utiliza cada uno y sobre todo del peso que tiene que desplazar. En este proyecto hemos utilizado para realizar las pruebas el mismo que efectúa la acción de soldadura en fábrica, el IRB6600 de ABB. Este modelo puede soportar una carga de hasta 150 Kg y posee un alcance de 2.2 metros. Es ágil, teniendo un largo recorrido en vertical, y muy rápido en ciclos cortos, el cual lo convierte en el más adecuado para el proceso de soldadura. Tiene la capacidad de doblarse completamente hacia atrás, ampliando aún más su rango de trabajo.



Figura 5.13: Brazo robot ABB IRB6600

<i>Exactitud de pose, APa, (mm)</i>	0,09
<i>Repetibilidad de pose, RP (mm)</i>	0,18
<i>Tiempo de estabilización de pose, PSt (s)</i>	0,03
<i>Exactitud de trayectoria, AT (mm)</i>	2,33
<i>Repetibilidad de trayectoria, RT (mm)</i>	1,05

Tabla 5.6: Características generales del IRB6600

A continuación vemos el desglose de los 6 ejes de libertad del brazo robótico, mostrando el signo correspondiente según hacia que lado gire cada eje. Se observa que coinciden con la comparación realizada en la sección 2.1.3 con un brazo humano. Para añadir el séptimo eje tan solo hay que añadir una cinta o corredera para que el robot se pueda desplazar de su posición.

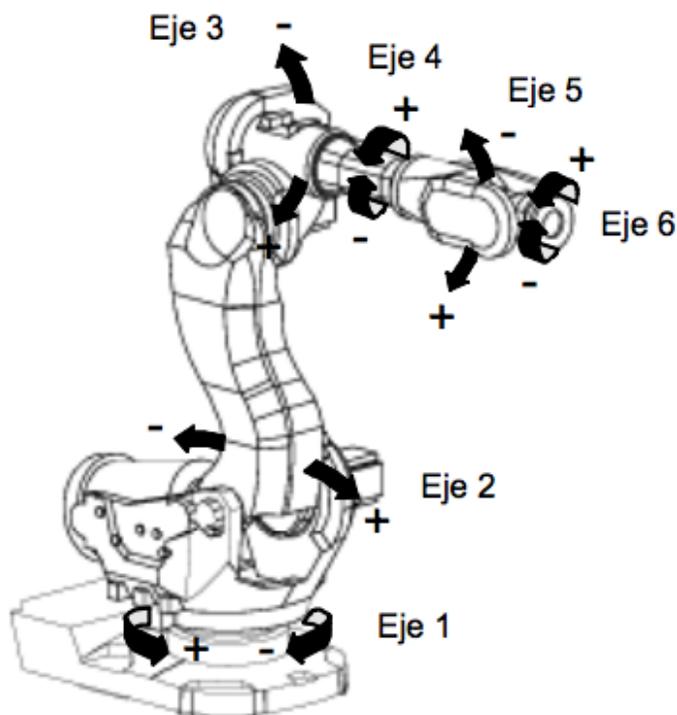


Figura 5.14: Ejes robot ABB IRB6600

Por último se muestra el rango y la capacidad de movimientos del modelo de brazo robótico utilizado en este proyecto.

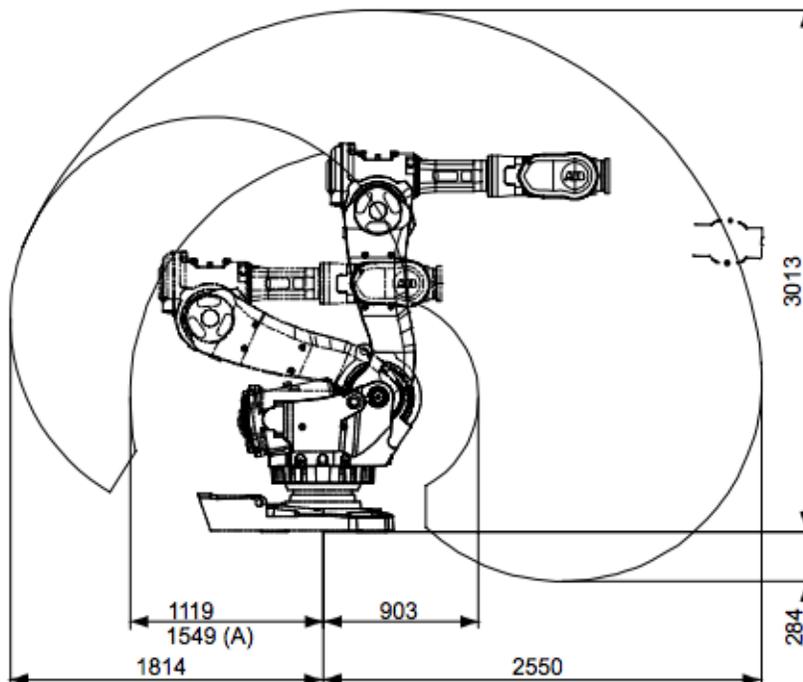


Figura 5.15: Rango de movimientos robot ABB IRB6600

Eje	Tipo de movimiento	Área de movimiento
1	Movimiento de rotación	De $+180^{\circ}$ a $-180^{\circ}$ De $+220^{\circ}$ a $-220^{\circ}$ (opcional)
2	Movimiento del brazo	De $+85^{\circ}$ a $-65^{\circ}$
3	Movimiento del brazo	De $+70^{\circ}$ a $-180^{\circ}$
4	Movimiento de la muñeca	De $+300^{\circ}$ a $-300^{\circ}$
5	Movimiento de doblado	De $+120^{\circ}$ a $-120^{\circ}$
6	Movimiento de giro	De $+360^{\circ}$ a $-360^{\circ}$ predeterminado $\pm 96$ rev.

Tabla 5.7: Capacidad de movimientos del IRB6600

### 5.1.7. Leica AT402 laser tracker

Los laser trackers son instrumentos de miden de forma muy precisa grandes objetos mediante la posición de un reflector óptico situado sobre éstos. La precisión de estos lasers es del orden de  $0,025mm$  en una distancia de varios metros. Entre sus aplicaciones destacan la alineación de las alas de los aviones durante su ensamblaje y la medición de los robots en los procesos industriales, sobre todo en el sector automovilístico.

Para su uso, se coloca encima del trípode con una visión ininterrumpida con el objeto a medir. Después se coloca el objetivo óptico o reflector en la posición a medir, siendo su mo-

vimiento suave una vez detectado para que el laser no pierda el objetivo. Con un controlador remoto, el técnico mide en la posición deseada y estos puntos se importan a un software para poder trabajar con ellos.

El Leica laser tracker AT402 es un tracker 3D utilizado para tareas de inspección tradicional o procesos de medida completamente guiados. Puede funcionar con su propia batería interna y está capacitado para trabajar en cualquier ambiente de trabajo, siempre manteniendo la precisión en cada medida. La empresa ha seleccionado este modelo en concreto ya que es más robusto y tiene una vida útil más larga que su antecesor. Esto hace que sea el indicado para las labores de medición y calibración de los robots en fábrica, donde puede sufrir algún golpe inesperado.

El tracker puede medir la ubicación de un reflector espacial con interferometría laser con gran precisión y velocidad.



Figura 5.16: Leica AT402 laser tracker

### 5.1.8. Reflector Bola

Con este reflector somos capaces de medir hasta 3 grados de libertad. El diámetro de la bola varía en función de la necesidad, siendo en nuestro caso de  $40,2mm$ . El origen de la bola, es decir, el punto real de medición, esta situado en el centro de la esfera.

Para su uso, simplemente se coloca en el punto exacto que se desea medir. El tracker se mueve automáticamente una vez detecta el sensor situado en el centro del reflector, por lo que tan solo hay que ir moviéndolo por los puntos deseados para realizar las mediciones.



Figura 5.17: Reflector bola

## 5.2. Software

A continuación se va a analizar la solución de software adoptada en este proyecto. Empezaremos por analizar el método utilizado para la programación del PLC, para luego describir detalladamente cada uno de los softwares utilizados.

### 5.2.1. GRAFCET

Para la descripción del funcionamiento del automatismo se ha utilizado la herramienta gráfica GRAFCET, dada su sencillez para elaborar el conjunto de funciones lógicas que regirán el funcionamiento del robot.

El GRAFCET está compuesto, entre otros, por etapas y transiciones. A cada etapa se le asocia generalmente una o varias acciones, que solo se ejecutarán cuando la etapa esté activada. Cada una de estas etapas se activa cuando se cumple la condición de transición y esta activa la etapa anterior. Esto implica que cuando se cumple la condición de transición, se activa la etapa siguiente y la precedente se desactiva. Además nunca pueden haber dos etapas o condiciones de forma consecutiva, deben de ir colocadas de forma alterna para que se puedan cumplir las normas anteriores.

Una vez obtenida la representación del GRAFCET, es fácil obtener las funciones lógicas que se implementarán en el lenguaje de programación para controlar todo el proceso.

A continuación se muestran los elementos que conforman un GRAFCET:

- **Etapas.** Representan la evolución del proceso. Se representan mediante un cuadrado con el número de etapa correspondiente, empezando por  $En$  donde  $n$  es el número de etapa, y en orden creciente. La etapa inicial, la que activa el GRAFCET, se representa con un recuadro de línea doble. Finalmente, las acciones se asocian a cada una de las etapas.



Figura 5.18: Etapas inicial y primera

- Acciones.** Están asociadas a las etapas, y representan lo que hay que hacer mientras la etapa está activa. Se representan mediante un rectángulo con la correspondiente etiqueta, situadas horizontalmente con respecto a la etapa a la que van asociadas.

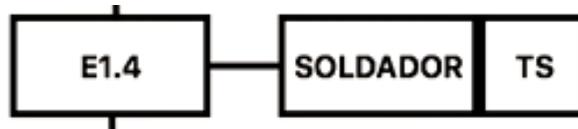


Figura 5.19: Acciones asociadas a las etapas

- Transiciones.** Representan la posibilidad de evolución de una etapa a la siguiente. Se representan mediante líneas horizontales sobre las barras verticales que unen las etapas.

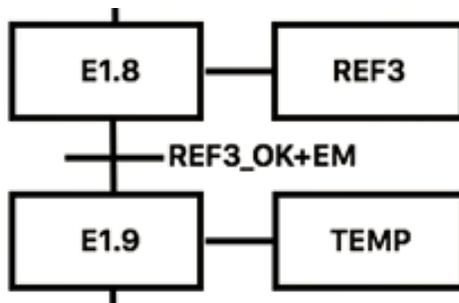


Figura 5.20: Transiciones entre etapas

A continuación se adjunta el GRAFCET que gobierna el proceso de la acción de soldadura con el brazo robótico.

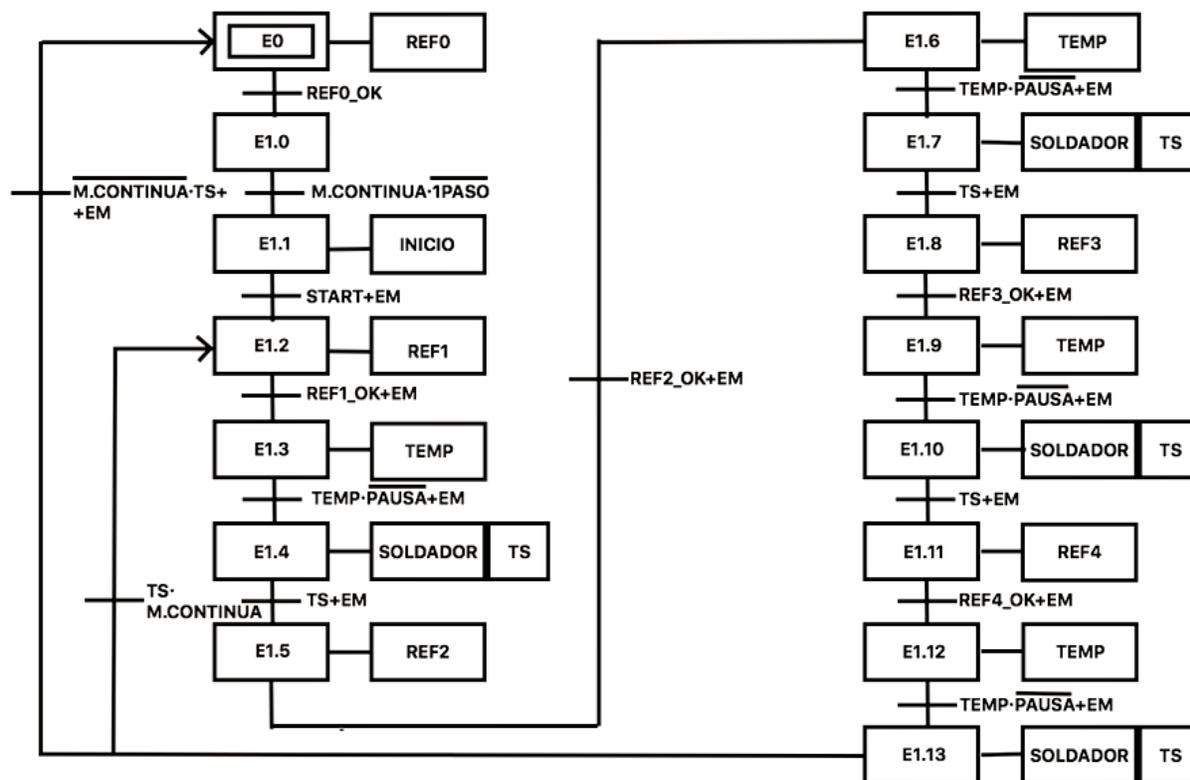


Figura 5.21: GRAFCET del proceso

Acción	Descripción
<i>Ref0</i>	<i>Lleva el robot a la posición inicial.</i>
<i>Ref1</i>	<i>Lleva el robot a la posición de referencia 1.</i>
<i>Ref2</i>	<i>Lleva el robot a la posición de referencia 2.</i>
<i>Ref3</i>	<i>Lleva el robot a la posición de referencia 3.</i>
<i>Ref4</i>	<i>Lleva el robot a la posición de referencia 4.</i>
<i>Inicio</i>	<i>Activa los sistemas del robot.</i>
<i>temp</i>	<i>Activa el temporizador de establecimiento.</i>
<i>ts</i>	<i>Activa el temporizador de soldadura.</i>
<i>soldadura</i>	<i>Realiza la acción de soldadura.</i>

Tabla 5.8: Acciones del GRAFCET

Transición	Descripción (en valor 1, conectado)
<i>Ref0_OK</i>	<i>El robot ha llegado a la Ref0.</i>
<i>Ref1_OK</i>	<i>El robot ha llegado a la Ref1.</i>
<i>Ref2_OK</i>	<i>El robot ha llegado a la Ref2.</i>
<i>Ref3_OK</i>	<i>El robot ha llegado a la Ref3.</i>
<i>Ref4_OK</i>	<i>El robot ha llegado a la Ref4.</i>
<i>M.CONTINUA</i>	<i>Se ha activado la marcha continua.</i>
<i>1PASO</i>	<i>Se ha activado 1 paso (1 ciclo del proceso).</i>
<i>START</i>	<i>Se ha puesto en marcha el proceso.</i>
<i>PAUSA</i>	<i>Realiza una pausa en el proceso.</i>
<i>EM</i>	<i>Se ha activado la opción de emergencia.</i>
<i>TEMP</i>	<i>Se ha cumplido el tiempo de establecimiento.</i>
<i>TS</i>	<i>Se ha cumplido el tiempo de soldadura.</i>

Tabla 5.9: Transiciones del GRAFCET

Se procede a explicar los elementos que componen el GRAFCET de este proyecto.

- La etapa “0” es la inicial. Se pone en funcionamiento el sistema llevando a cabo la acción “*Ref0*”. Esta lleva el brazo robótico a la posición inicial, independientemente de donde se encuentre en ese instante. Una vez los sensores del robot detectan que ha llegado al punto de referencia, se cumple la condición de la primera transición, “*Ref0\_OK = 1*”, y se activa el estado 1.
- La etapa “1” no contiene ninguna acción. Simplemente se utiliza como apoyo para poder ejecutar las siguientes condiciones de forma más sencilla e intuitiva. Para poder avanzar a la siguiente etapa, el operario debe seleccionar si desea realizar tan solo un ciclo, o si quiere que el proceso no se detenga. Para ello, el sistema está diseñado de forma que en este punto, debe estar “*MarchaContinua = 1*” y “*1Paso = 0*” para que el programa se ejecute. De esta forma, y como veremos más adelante, podemos optar a ambas opciones con tan solo 1 rama en el grafcet, por lo que el proceso es más eficiente y rápido.
- En la etapa “1,1” se ejecuta la acción de Inicio. Esto es un pulsador que activa los sistemas del robot para estar a punto para cuando el operario decida iniciar el proceso. A partir de esta etapa, cada transición estará compuesta de 2 cosas: la primera, la condición para poder avanzar a la siguiente etapa; la segunda, la opción de paro de emergencia “*EM*” por si algo falla y hay que detener el proceso rápidamente. En esta etapa, se debe poner “*Start = 1*” para poder avanzar a la siguiente.
- La etapa “1,2” contiene la acción de *Ref1*. Como en la etapa inicial, el robot se mueve hasta la primera posición donde realizará un punto de soldadura. Cuando los sensores detecten que el robot ha llegado, es decir, “*Ref1\_OK = 1*”, se avanzará a la siguiente etapa.
- Esta etapa, la “1,3”, se añade para asegurarse de que el proceso se realizará correctamente. Para ello, se obliga al robot a estar un cierto tiempo parado, mediante la acción “*temp*”, para que se establezca y realice la acción de soldadura correctamente. Cuando se cumpla que “*temp = 1*” y además que “*pausa = 0*”, se cumplirá la transición. En este caso se deben cumplir ambas cosas porque aquí el operario tiene la opción de pausar el proceso para ver que todo está cumpliendo las expectativas. Esta opción de pausa se puede ejecutar antes de cada acción de soldadura, para comprobar que la anterior ha salido correctamente. Una vez desactivada la opción, las condiciones se cumplirán y se avanzará a la siguiente etapa.
- La etapa “1,4” es la encargada de ejecutar la acción de soldar. En ella, además, se activa la acción “*ts*”, la cual activa un temporizador de soldadura. Cuando se cumplen los 0.5 segundos de soldadura, “*ts = 1*” y se avanza a la siguiente etapa.
- Una vez llegados a la etapa “1,5”, el proceso anterior se repite para los 4 puntos de referencia distintos en los cuales hay que realizar la acción de soldar. Por ello, avanzamos a la última etapa de todas.
- En la etapa final, la “1,13”, hay dos caminos que se pueden seguir. Después de realizar la soldadura, se puede ir tanto a la etapa inicial como a la “1,2”. La diferencia reside en lo mencionado anteriormente dependiendo de si se quiere realizar uno o más ciclos del proceso.

- Si después de la etapa final “ $mc = 0$ ” y “ $ts = 1$ ”, o “ $EM = 1$ ”, el sistema vuelve a la etapa inicial, permaneciendo en reposo a partir de aquí hasta que alguien lo vuelva a activar. Esta sería la opción de “1Paso”, es decir, tan solo se realizaría un ciclo del proceso. El hecho de que la parada de emergencia aparezca en esta rama es debido a que se está obligando al sistema a volver a la posición inicial tras la parada, es decir, el sistema no puede seguir funcionando hasta que el operario haya comprobado que todo está bien. Esto se suele hacer para verificar que todo funciona correctamente antes de ponerse a producir en cadena.
- En cambio, si solo se cumple que “ $ts = 1$ ” y “ $mc = 1$ ”, es decir, no se desactiva la marcha continua, el proceso regresa a la etapa “1,2” y se repite todo de nuevo, hasta que se desactive la Marcha Continua.

### 5.2.2. TIA Portal

El Totally Integrated Automation Portal, más conocido como TIA Portal, es un software que permite configurar de forma intuitiva y eficiente todos los procesos de planificación y producción. Optimiza todos sus procedimientos de procesamiento, operación de máquinas y planificación para ofrecer al usuario todas la herramientas que pueda necesitar durante su uso.

En este proyecto específicamente, ha sido elegido por su intuitiva sencillez de usuario, ya que cualquiera que no tenga experiencia con el programa puede llegar a programar controladores en poco tiempo. Además, la sencillez de sus funciones hace que sea un programa fácil de utilizar.

El procedimiento empleado para el uso de este software ha sido la construcción previa del GRAFCET y posteriormente las ecuaciones y diagramas de contactos en papel. Una vez obtenidas, se han implementado en el software. Para ello, se ha utilizado el mismo PLC y las mismas tarjetas de entradas y salidas que las que utiliza el robot, para cerciorarse que el programa funcionará en las pruebas reales. A continuación se añade una captura mostrando los dispositivos implementados en el software.

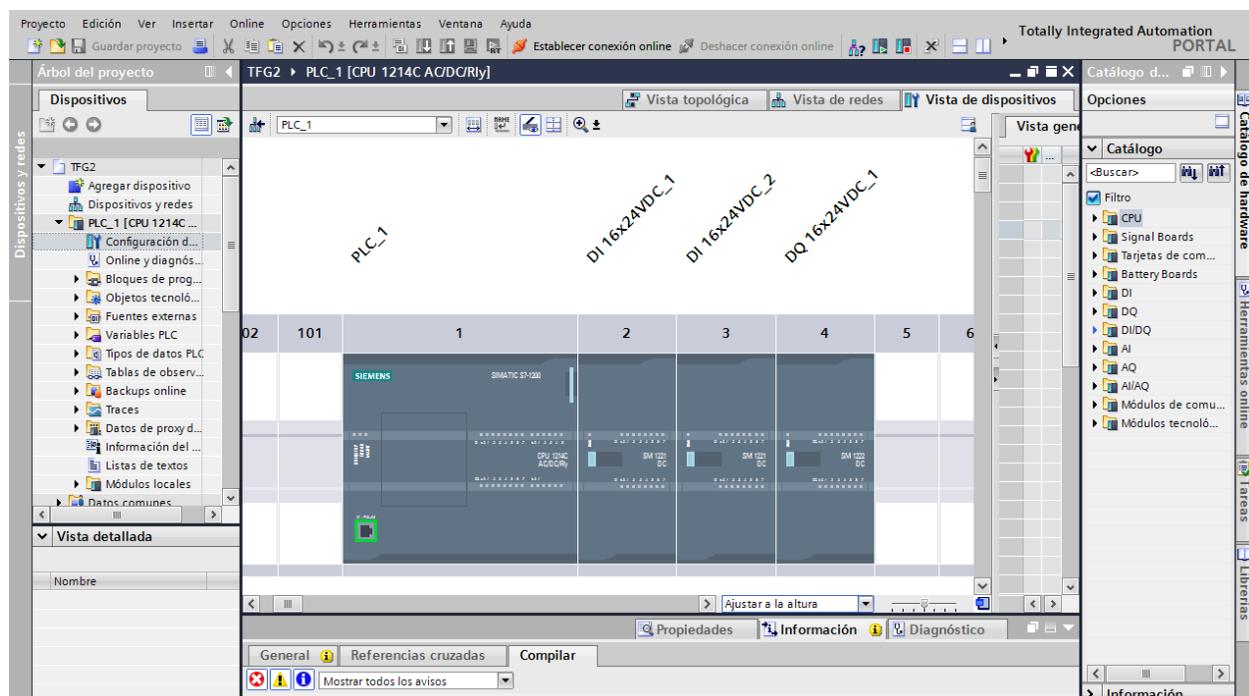


Figura 5.22: Dispositivos implementados en TIA Portal

Se procede a mostrar tres ejemplos de programación para los distintos módulos: estados, temporizadores y salidas del PLC.

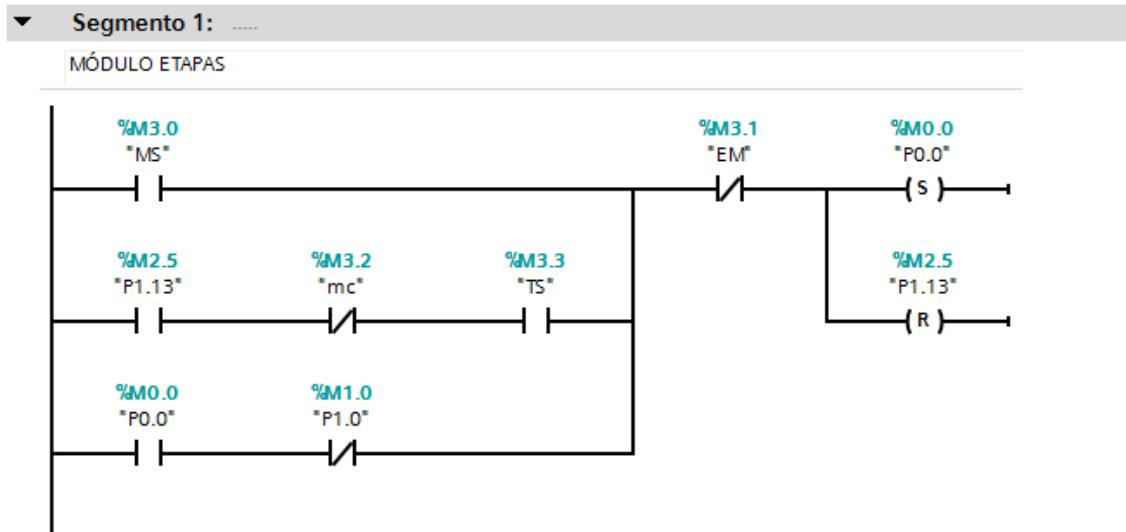


Figura 5.23: Estado inicial del GRAFCET

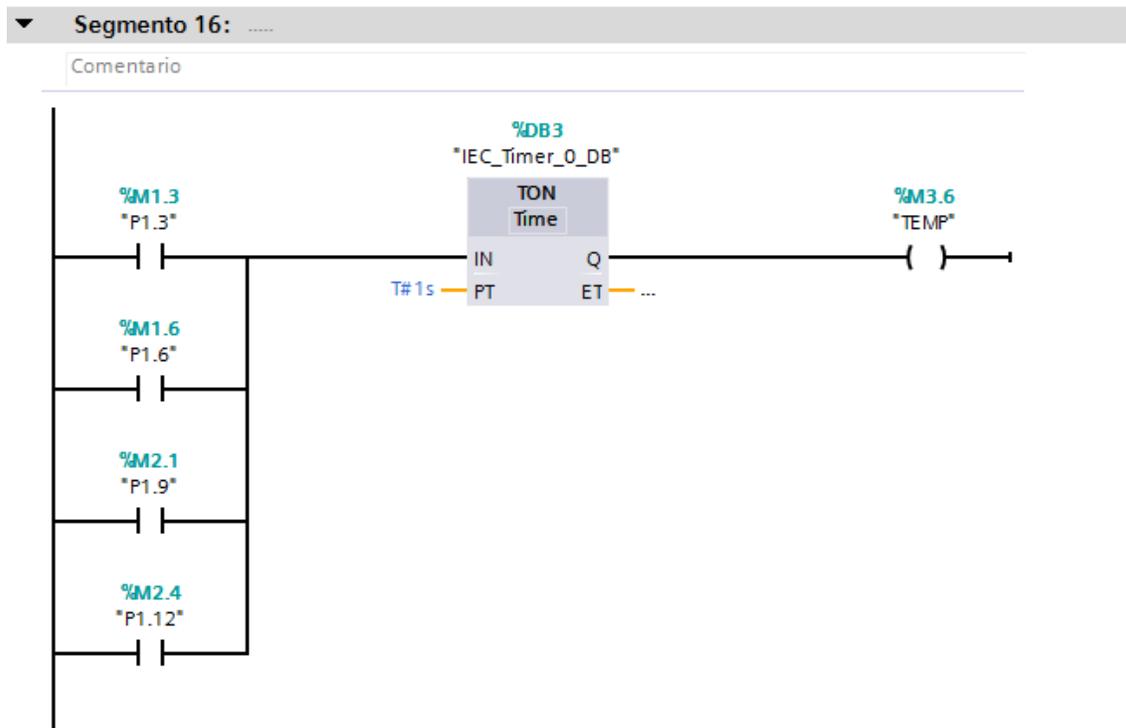


Figura 5.24: Temporizador de establecimiento del GRAFCET

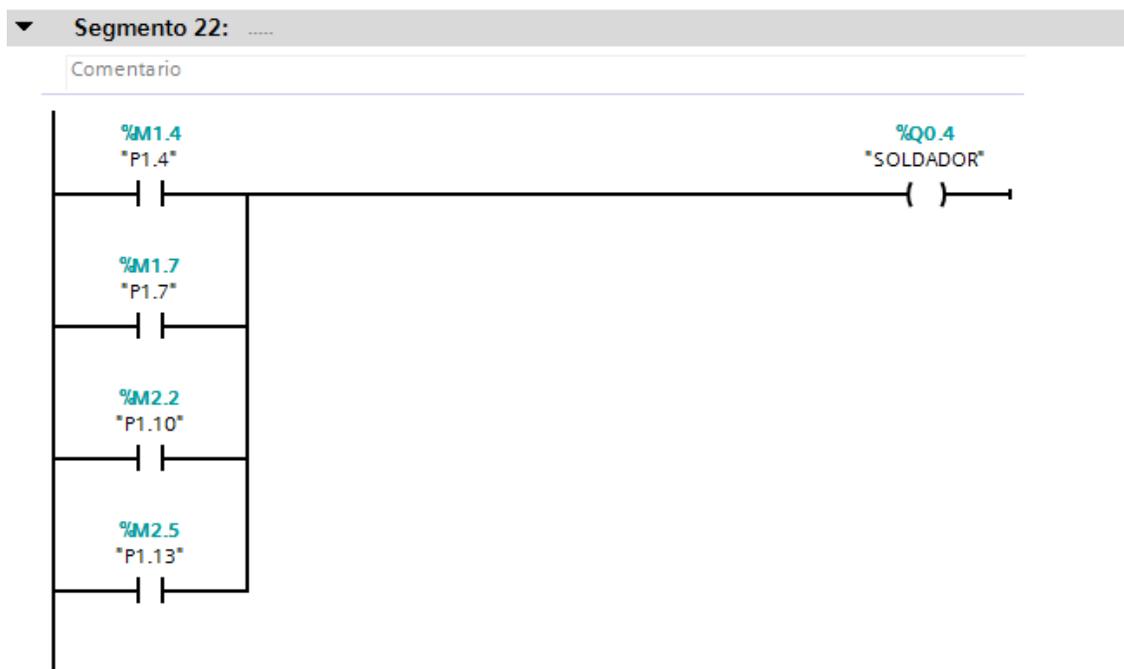


Figura 5.25: Salida de soldadura del GRAFCET

### 5.2.3. RobotStudio

RobotStudio es un software que te permite trabajar eficientemente con datos de controladores y robots reales creados por la empresa ABB. En él, puedes configurar tanto la parte teórica de la programación del robot, como la práctica en cuanto al montaje de la estación se refiere, lo que le convierte en un simulador perfecto para realizar pruebas con cualquiera de los robots que ofrece. Posee también el FlexPendant mencionado en la sección 5.1.5, por lo que con este software se puede programar un proceso de automatización completo.

El FlexPendant tiene como fines principales el movimiento del robot con el joystick, la programación de posiciones, el manejo y el ajuste de movimientos, mientras que RobotStudio resulta ideal para el manejo de datos de configuración, gestión de programas, documentación en línea y acceso remoto. Como en el caso real, RobotStudio sólo puede tomar el control del robot si el FlexPendant le autoriza esa operación.

RobotStudio actúa directamente sobre los datos activos del controlador. La conexión al controlador puede hacerse localmente a través de la conexión para PC de servicio y, si el controlador cuenta con la opción de RobotWare PC Interface, a través de una conexión de red.

Para este proyecto se ha creado una pequeña estación compuesta por los siguientes elementos: el robot IRB6600, cargado desde la librería que ofrece ABB; el controlador IRC5, sincronizado con el robot; la herramienta que simula la acción de soldadura; un pedestal donde se sitúa el robot; una mesa de trabajo donde se sitúa la pieza que requiere la soldadura y finalmente la pieza a soldar. Se muestra a continuación una captura de la estación.

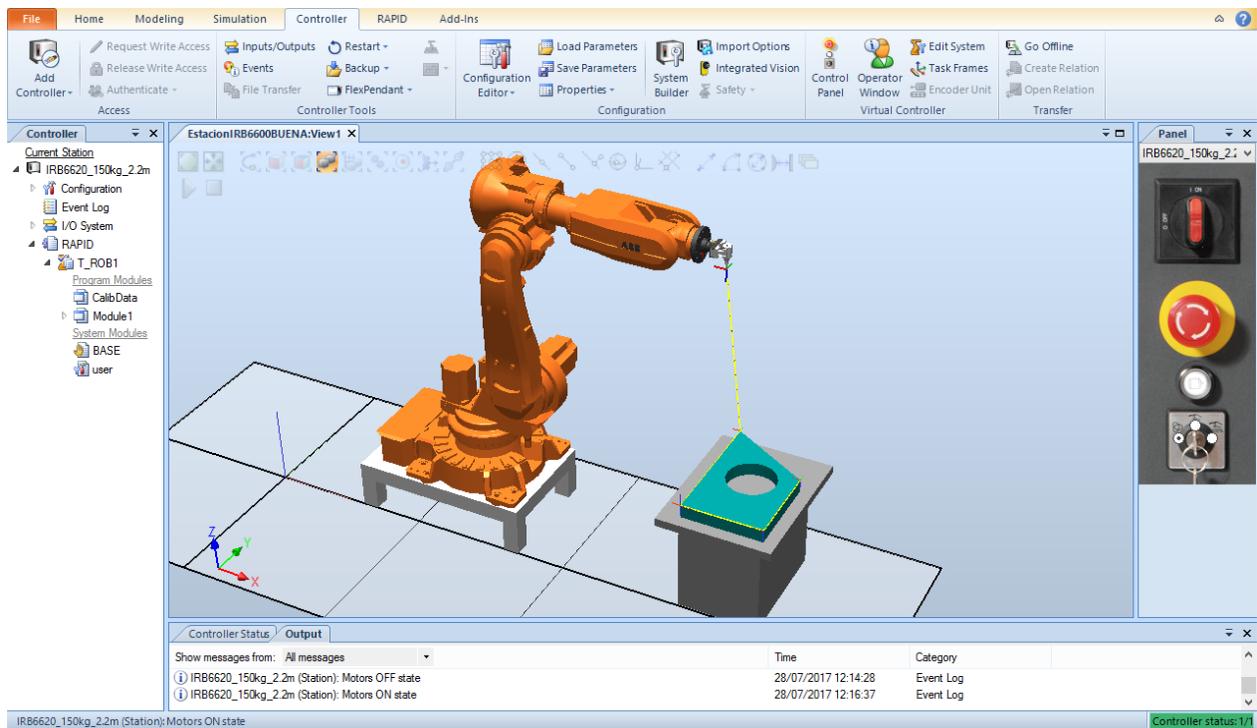


Figura 5.26: Estación con RobotStudio

### 5.2.4. OpenIndy

La razón principal del cambio de software para la medición de la mesa es la corrección del error debido al TCP. Además, con el OpenIndy no es necesario visualizar las mediciones que se van realizando en el ordenador, por lo que la medición se realiza con mucha más rapidez.

La reducción del error se debe a que este software calcula el TCP del robot utilizando 10 puntos repartidos por toda la zona de trabajo. El resultado es que, a diferencia del Polyworks donde obtenemos  $0mm$  de desviación en el punto medido y  $5mm$  en zonas lejanas a este, obtenemos un error de  $1mm$  distribuido por toda la zona de trabajo, el cual es aceptable.

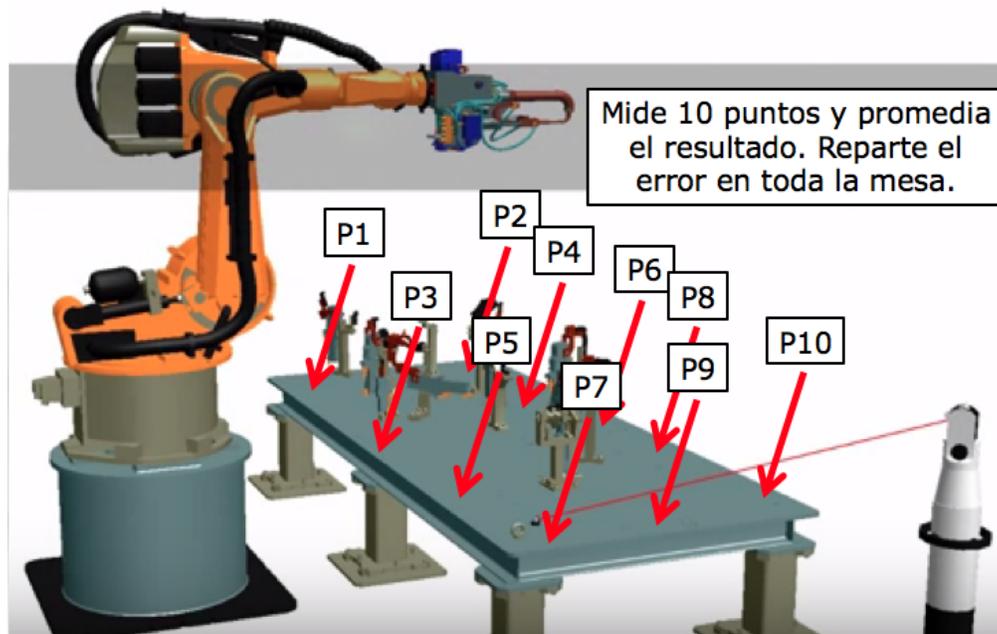


Figura 5.27: Distribución de puntos medidos con OpenIndy





---

# Capítulo 6

## Tests y resultados

En este capítulo se presentan los tests y análisis realizados para comprobar el correcto funcionamiento de los softwares diseñados. Se realizarán comprobaciones tanto de posición como de velocidad del robot, estudiados según los distintos tramos que éste efectúa, analizando los resultados obtenidos.

### 6.1. Test del primer tramo

El primer tramo se compone del movimiento del robot desde el punto de referencia inicial (Ref0) hasta el primer punto de referencia (Ref1). Aquí, el robot se mueve en los 3 ejes de coordenadas, pero solo analizaremos el plano  $x - y$  dado que la pieza a soldar está toda a la misma altura. Se procede a mostrar gráficamente el movimiento del robot en este tramo, junto con la vista cenital del punto de referencia inicial.

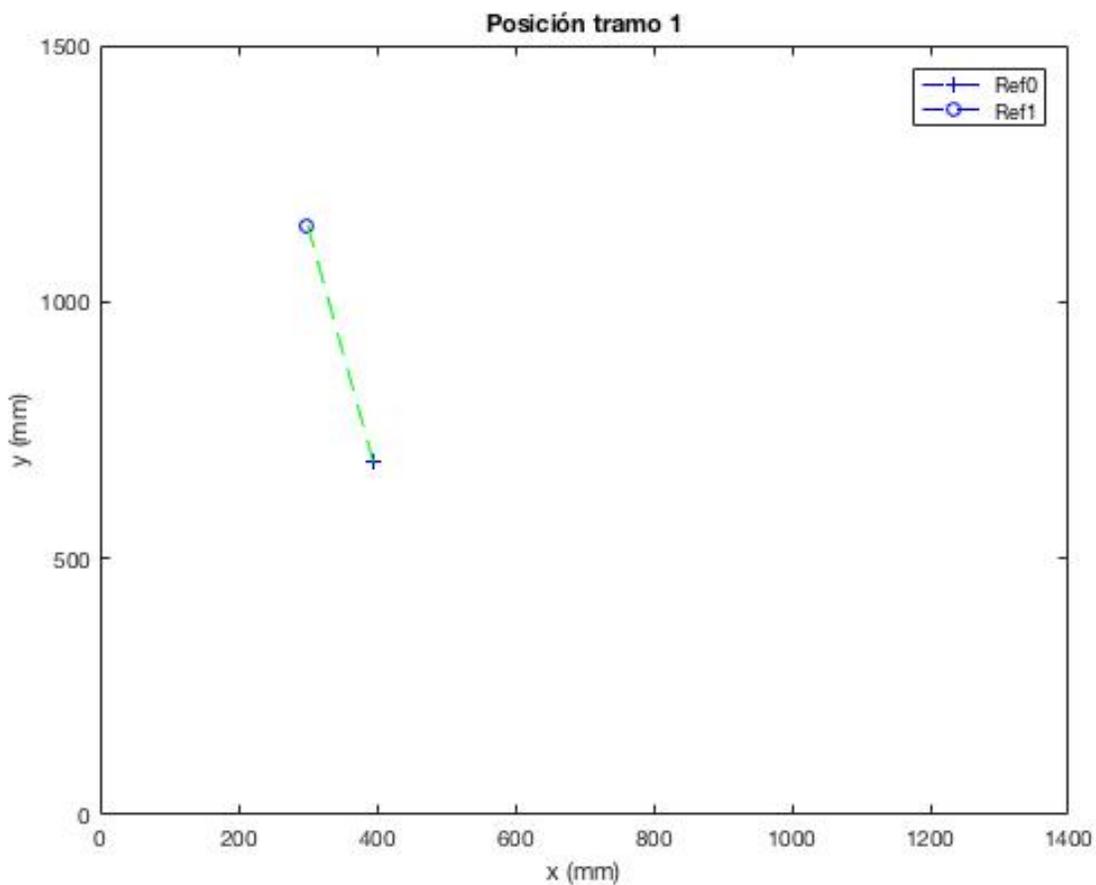


Figura 6.1: Posición en el tramo 1

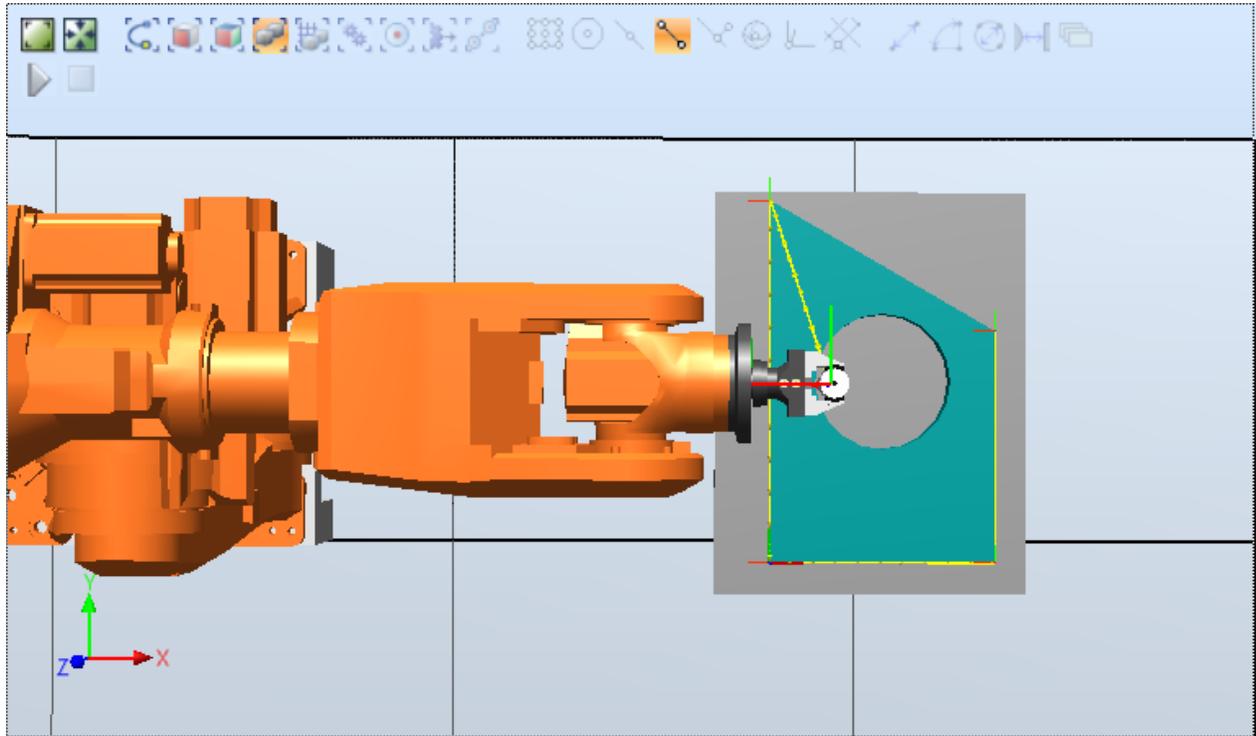


Figura 6.2: Vista cenital en Ref0

Como veremos en el siguiente gráfico, la velocidad inicial del robot es nula. Esto es lógico ya que en este punto el robot se encuentra parado esperando a recibir las órdenes del PLC. Al recibir la instrucción, acelera hasta llegar a la velocidad de movimiento predeterminada, establecida en  $1\text{m/s}$ . Entonces sigue moviéndose hasta estar cercano al punto de referencia 1, donde empieza a decelerar hasta quedarse quieto otra vez. El tiempo total, como se puede ver en la siguiente imagen, es de 2,284 segundos.

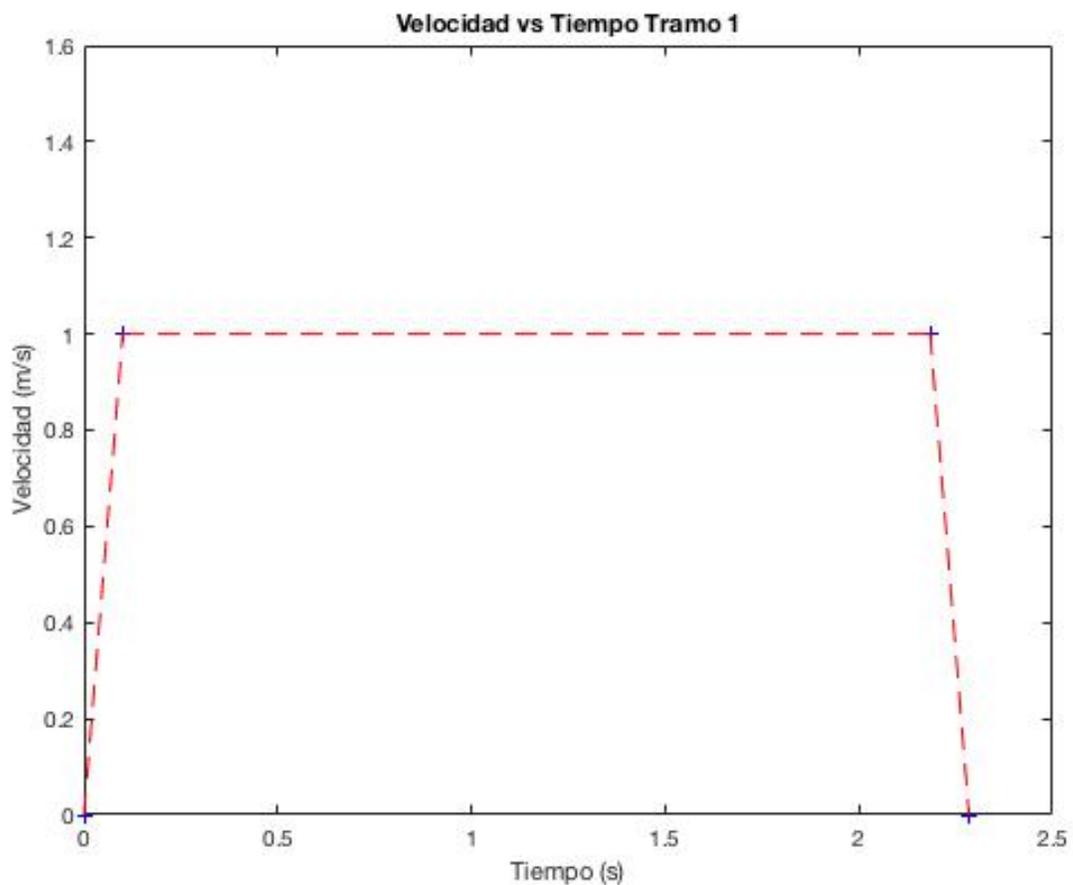


Figura 6.3: Velocidad en el tramo 1

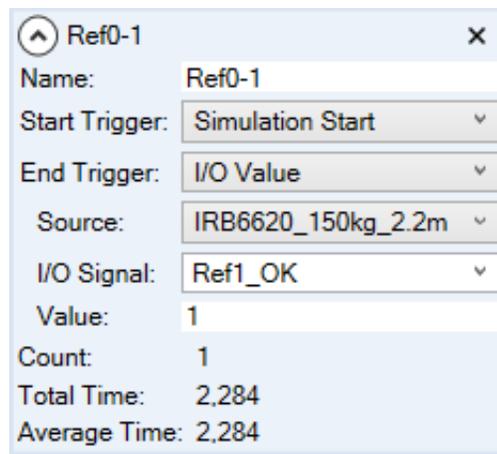


Figura 6.4: Tiempo total en realizar el tramo 1

## 6.2. Test del tramo 2

Una vez el robot llega al primer punto de referencia, el robot debe avanzar linealmente por el borde de la pieza hasta llegar al siguiente punto donde debe efectuar la soldadura.

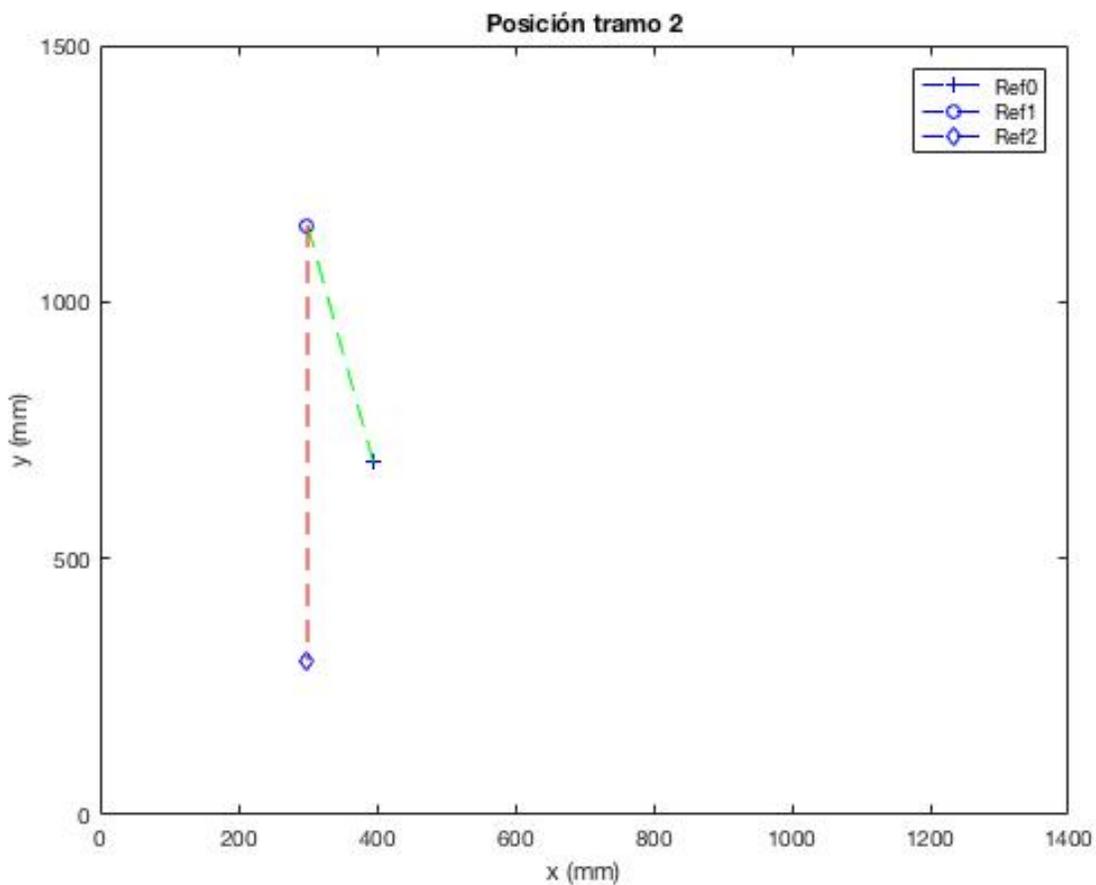


Figura 6.5: Posición en el tramo 2

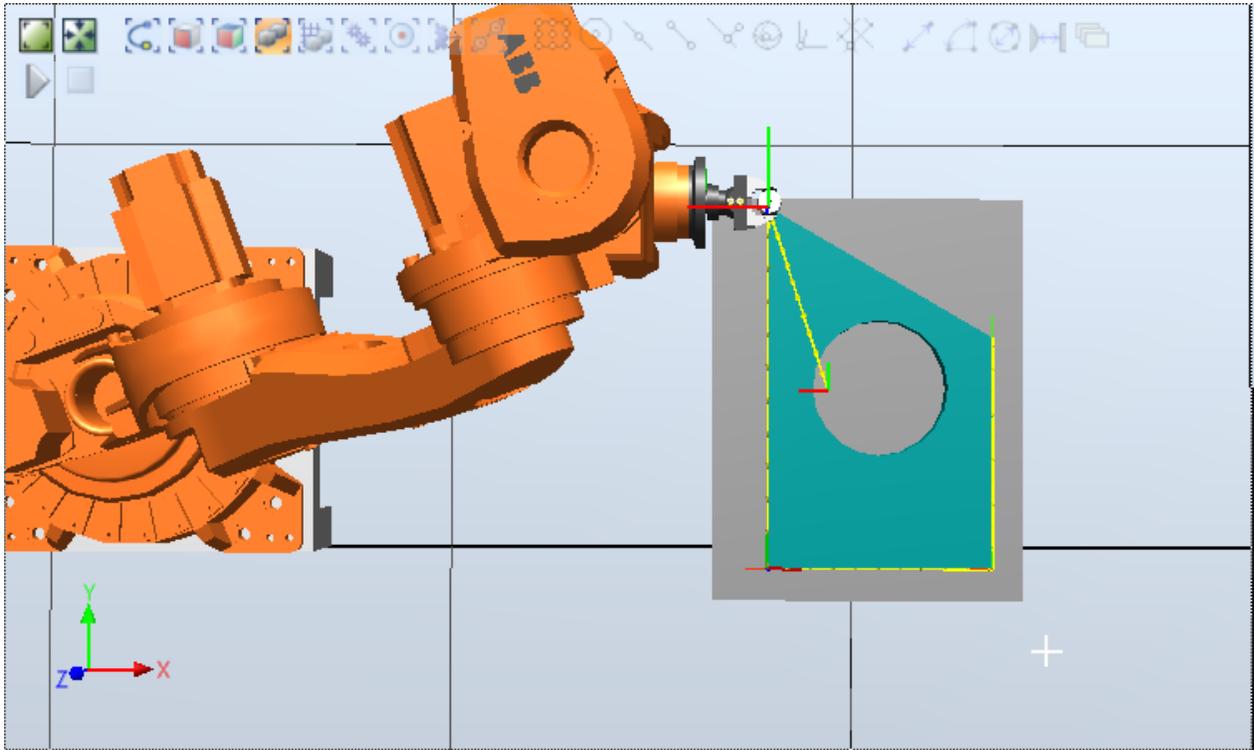


Figura 6.6: Vista cenital en Ref1

Estando en este punto, la orden del PLC es que se espere cierto tiempo para estabilizarse y, una vez transcurrido esto, realice la soldadura antes de avanzar al siguiente punto de referencia.

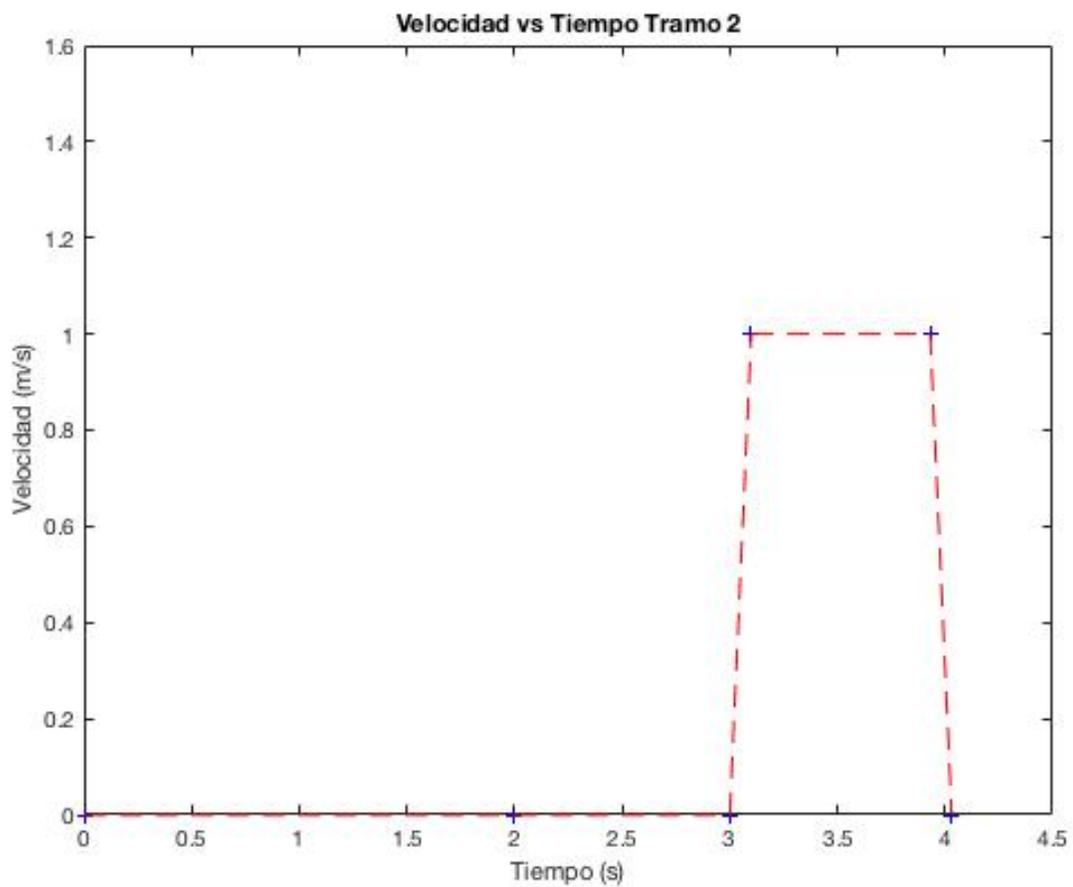


Figura 6.7: Velocidad en el tramo 2

Esta vez, vemos como el robot espera 3 segundos antes de empezar a moverse. Esto es correcto, ya que se espera los 2 segundos de estabilización antes de realizar la soldadura, y 1 segundo mientras realiza ésta última. Una vez iniciado el movimiento, el tiempo en recorrer

este tramo es de 1,031 segundos que, sumados a los tres que permanece en reposo, hacen un total de 4,031 segundos.

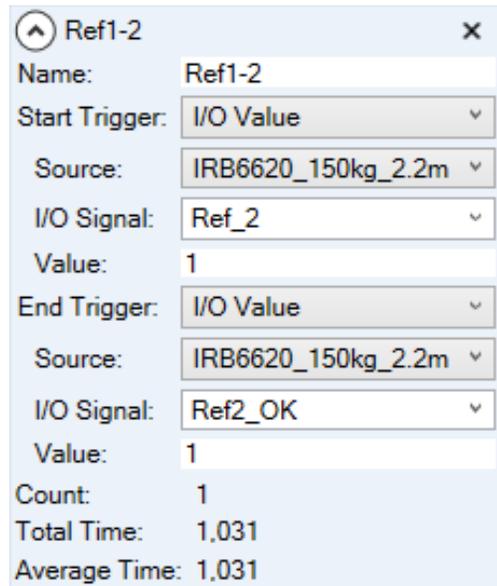


Figura 6.8: Tiempo total en realizar el tramo 2

### 6.3. Test del tramo 3

En este tramo el robot debe recorrer según el eje x el lado horizontal de la pieza.

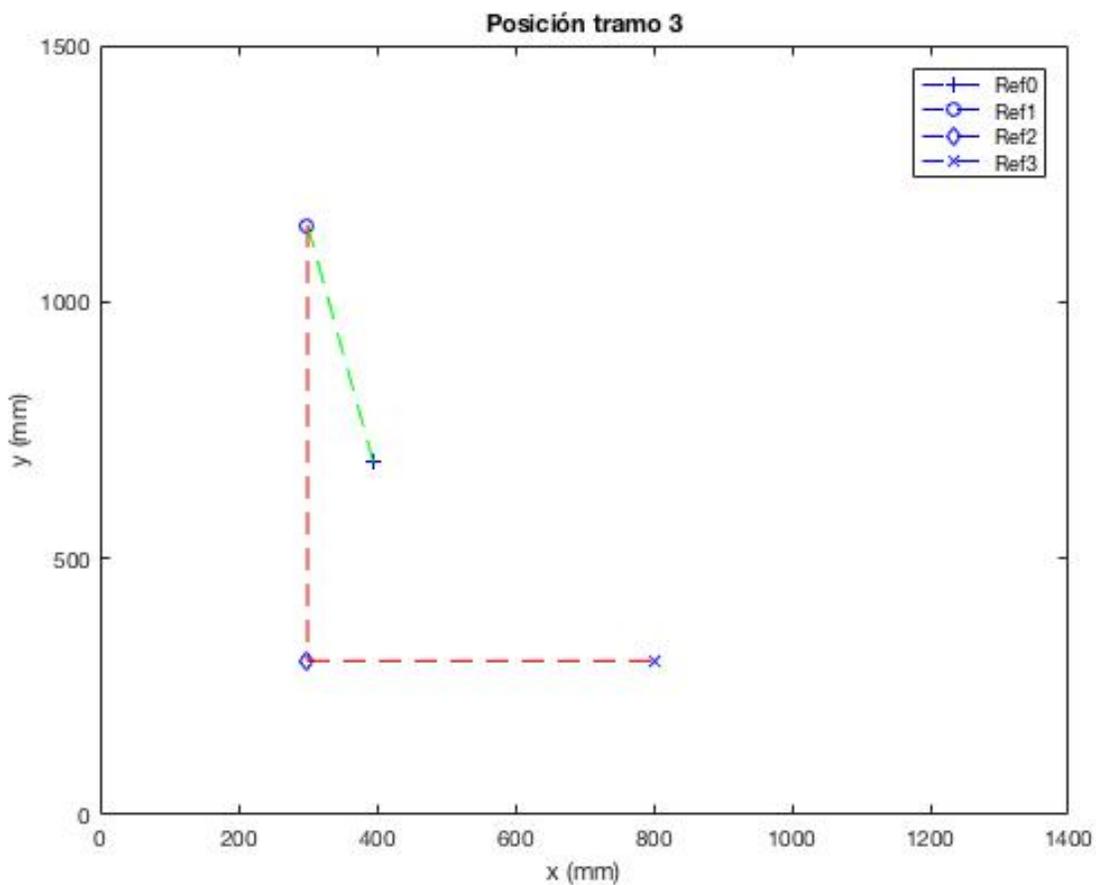


Figura 6.9: Posición en el tramo 3

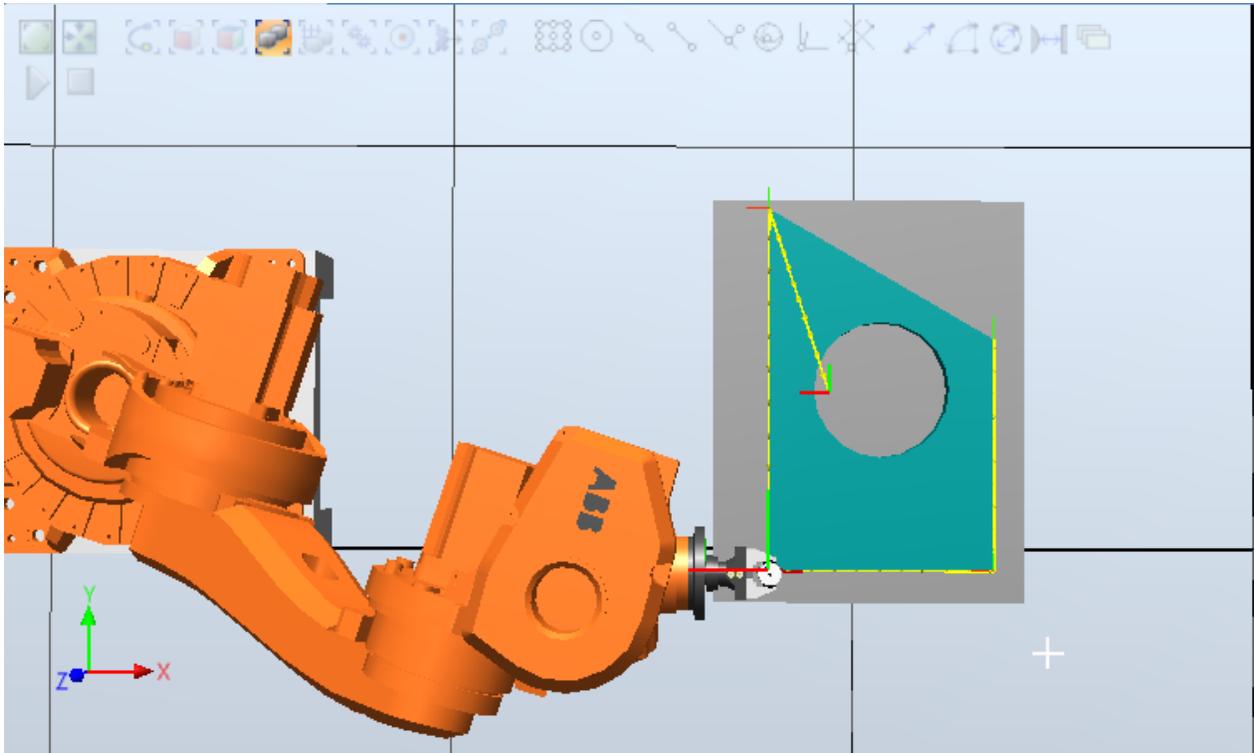


Figura 6.10: Vista cenital en Ref3

Desde este punto, las ordenes del PLC al robot se repiten a medida que avanza a lo largo de la pieza, es decir, le ordena que espere el tiempo de estabilización y que realice la soldadura. La diferencia entre cada punto es el tiempo total que tarda en llegar hasta el siguiente punto a soldar, ya que las dimensiones de la pieza varían a lo largo de los ejes. El tiempo total para este tramo es de 3,737 segundos.

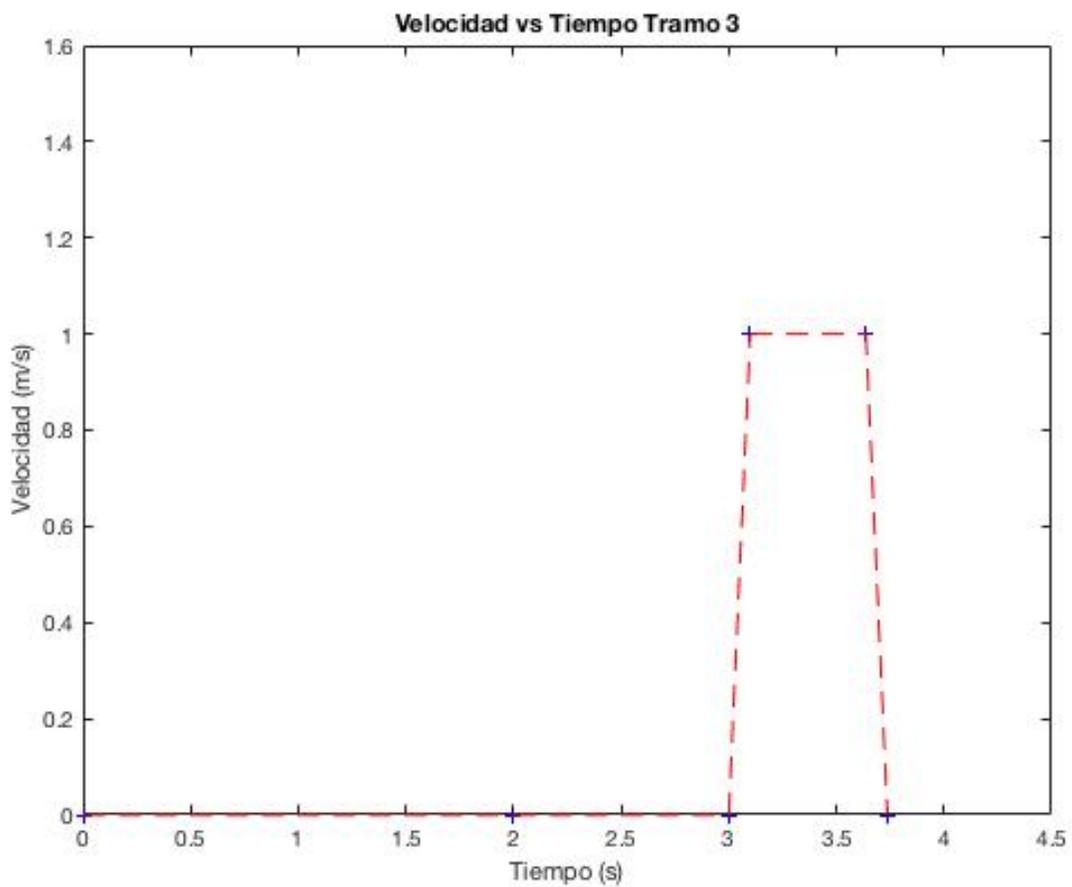


Figura 6.11: Velocidad en el tramo 3

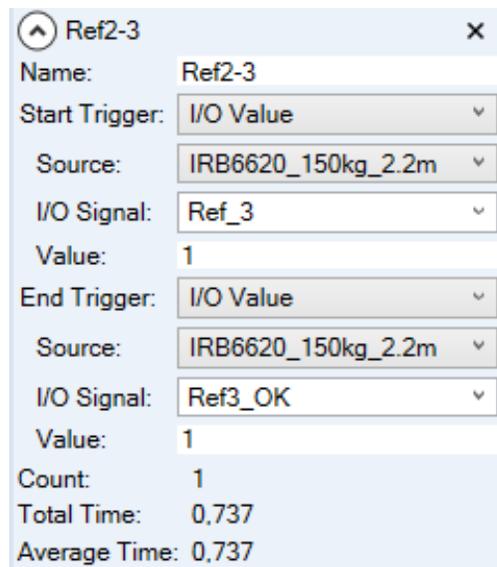


Figura 6.12: Tiempo total en realizar el tramo 3

## 6.4. Test del tramo 4

En este tramo el robot avanza hasta el último punto de referencia para efectuar un último punto de soldadura y terminar la pieza correspondiente. Es un movimiento paralelo al eje  $y$  de distancia 0.5 metros.

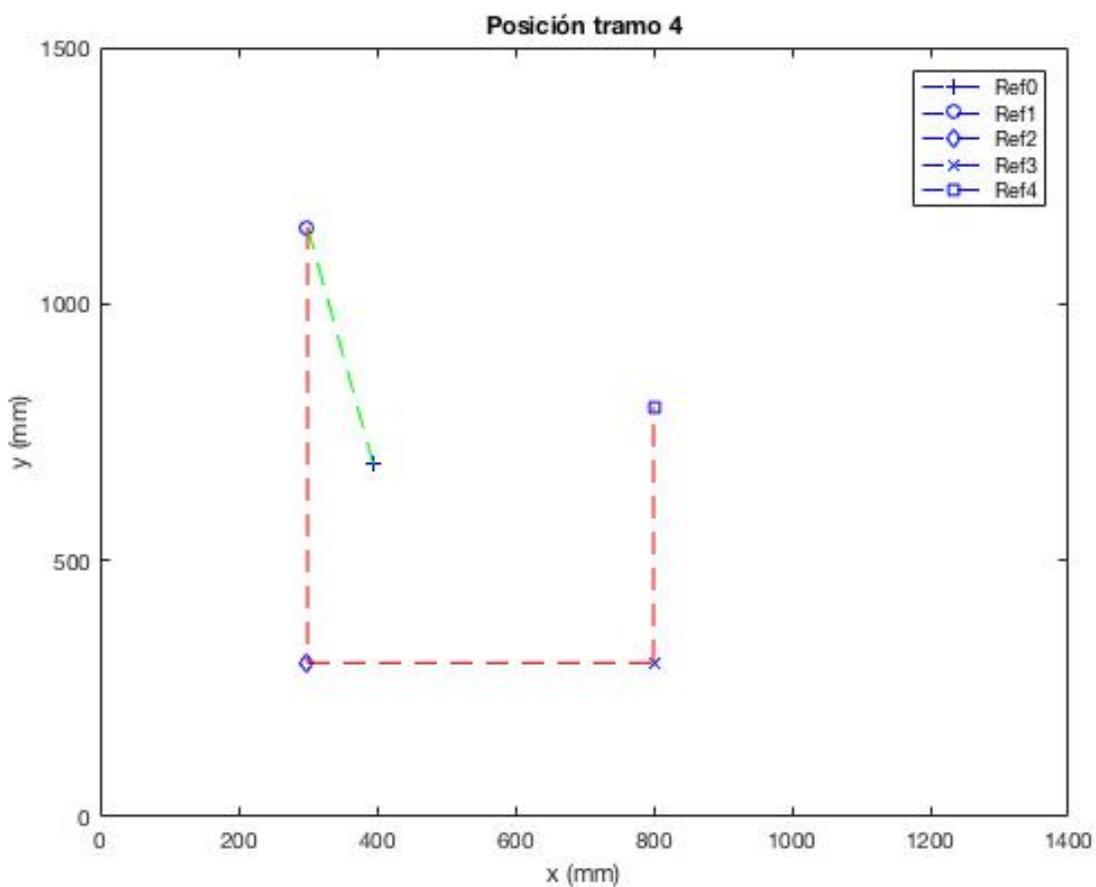


Figura 6.13: Posición en el tramo 4

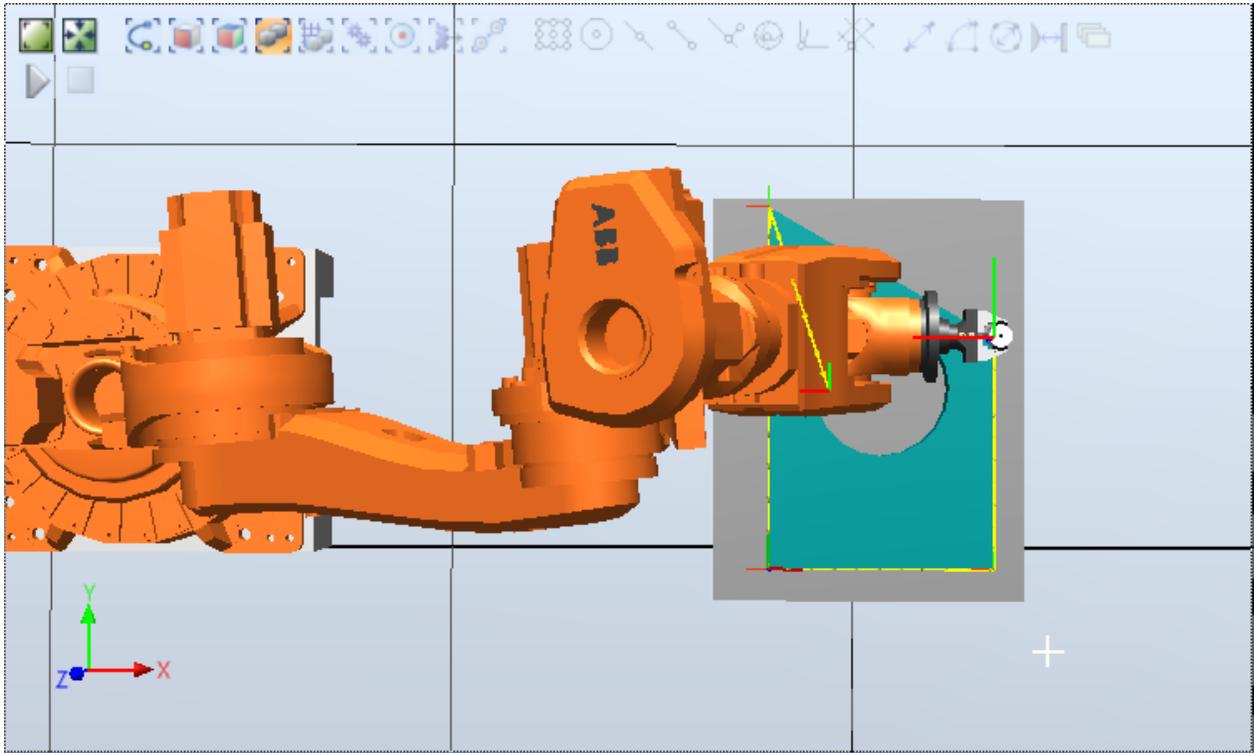


Figura 6.14: Vista cenital en Ref4

El proceso es análogo a los dos anteriores, tardando un tiempo total de 3,743 segundos en realizar el tramo completo.

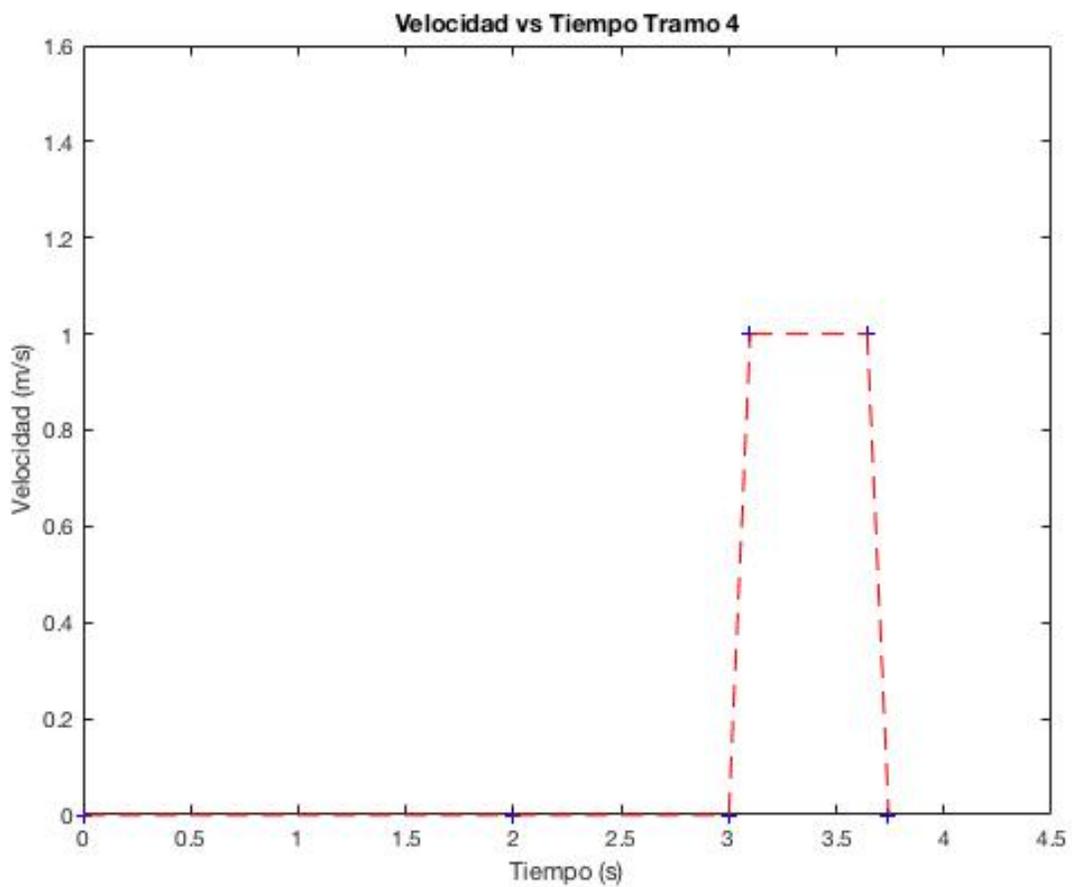


Figura 6.15: Velocidad en el tramo 4

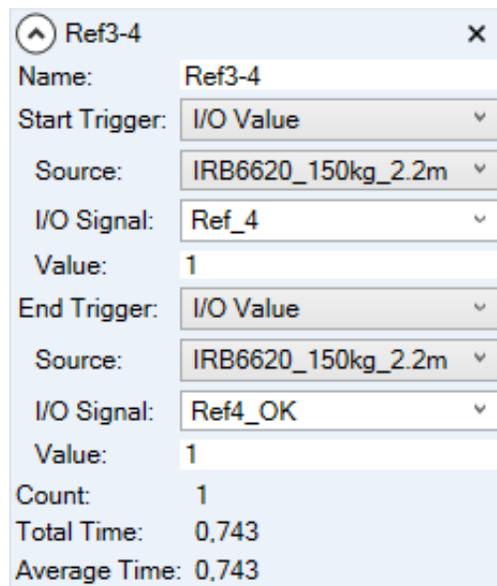


Figura 6.16: Tiempo total en realizar el tramo 4

## 6.5. Test del tramo final

El tramo final tiene dos opciones según el deseo del operario, que son analizadas a continuación.

La primera opción es escoger marcha continua, en la cual el robot debe volver a la posición de *Ref1* para efectuar la soldadura en la pieza siguiente. Esto es, debe realizar un movimiento lineal oblicuo a los ejes hasta llegar al primer punto de referencia.

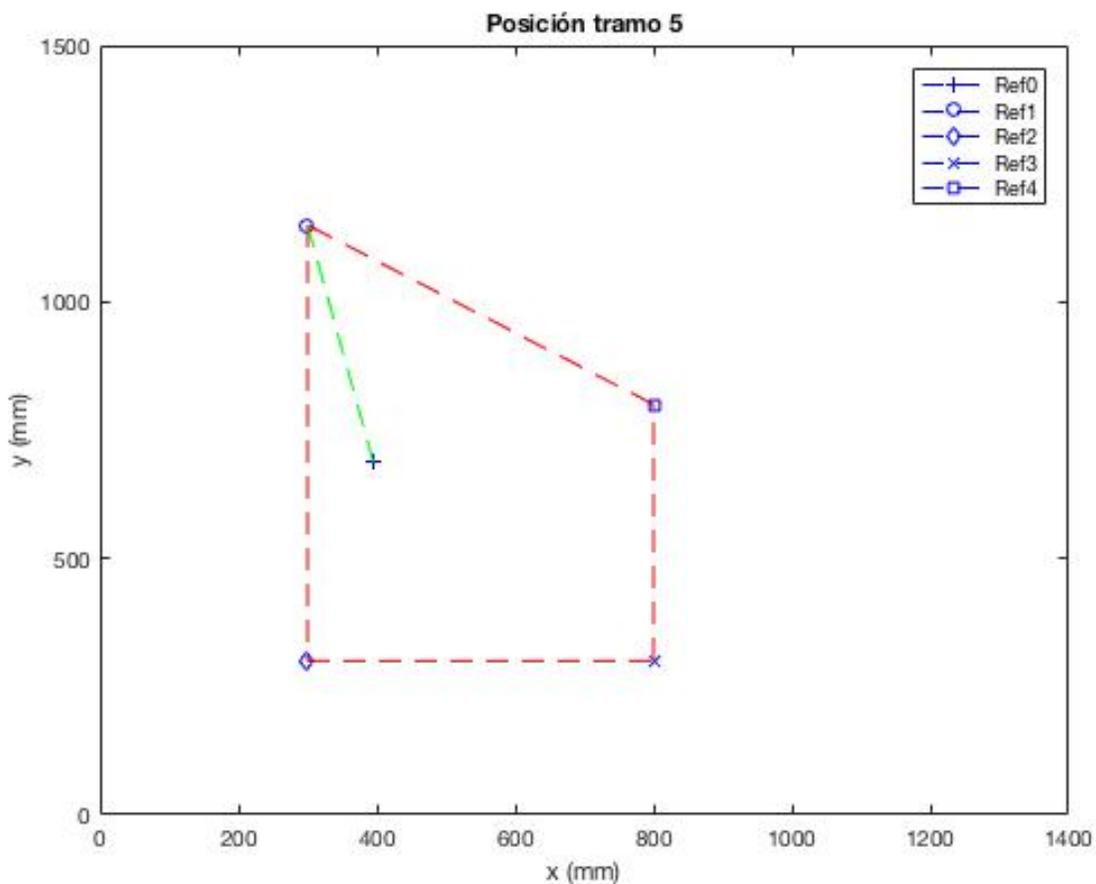


Figura 6.17: Posición en el tramo 5(I)

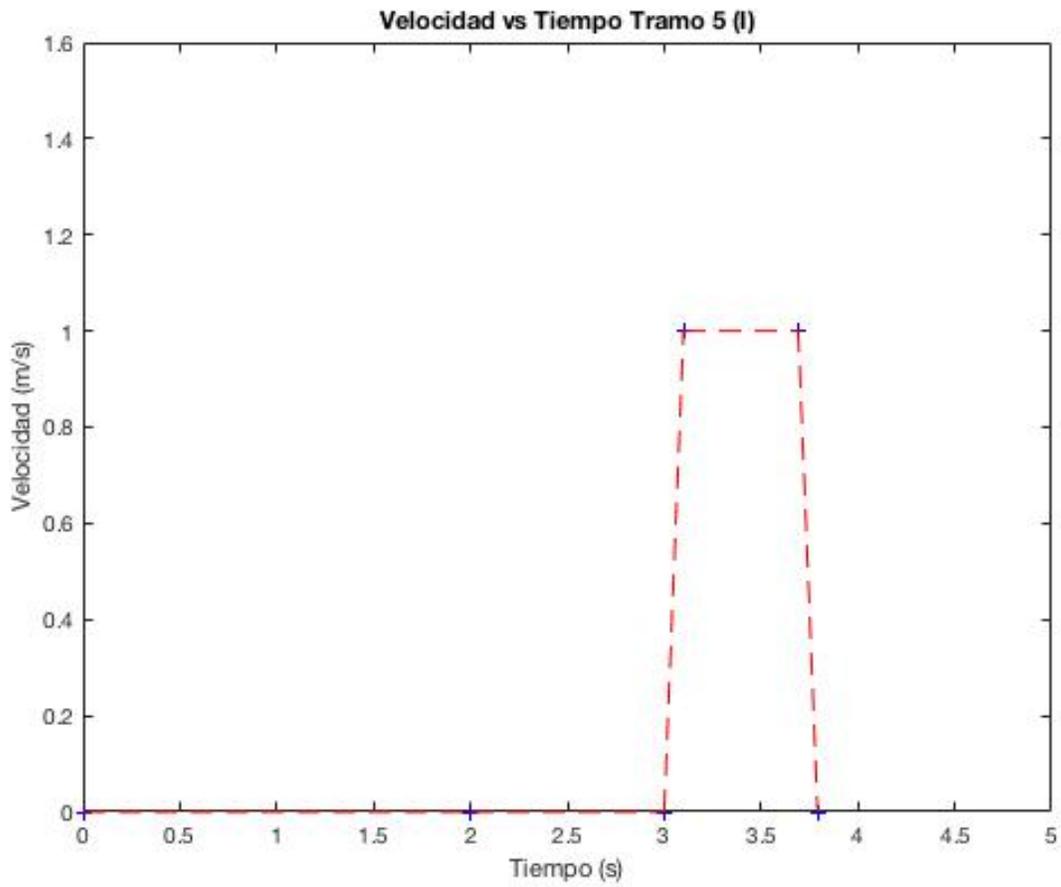


Figura 6.18: Velocidad en el tramo 5(I)

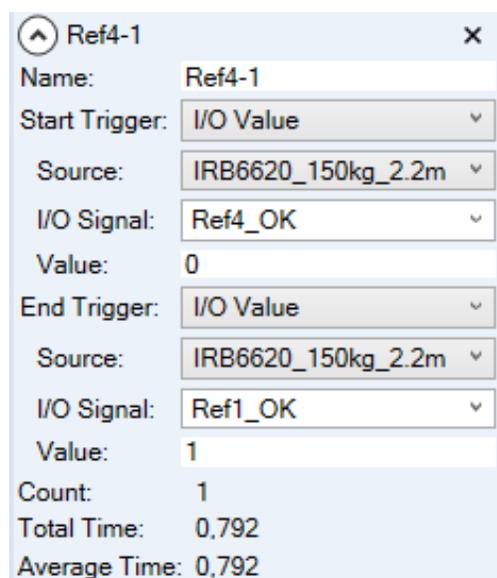


Figura 6.19: Tiempo total en realizar el tramo 5(I)

La segunda opción es elegir 1 paso, es decir, tan solo se realiza un ciclo del proceso. En este caso, una vez soldado el último punto, el robot debe regresar a la posición de referencia inicial hasta que el PLC le vuelva a dar alguna orden.

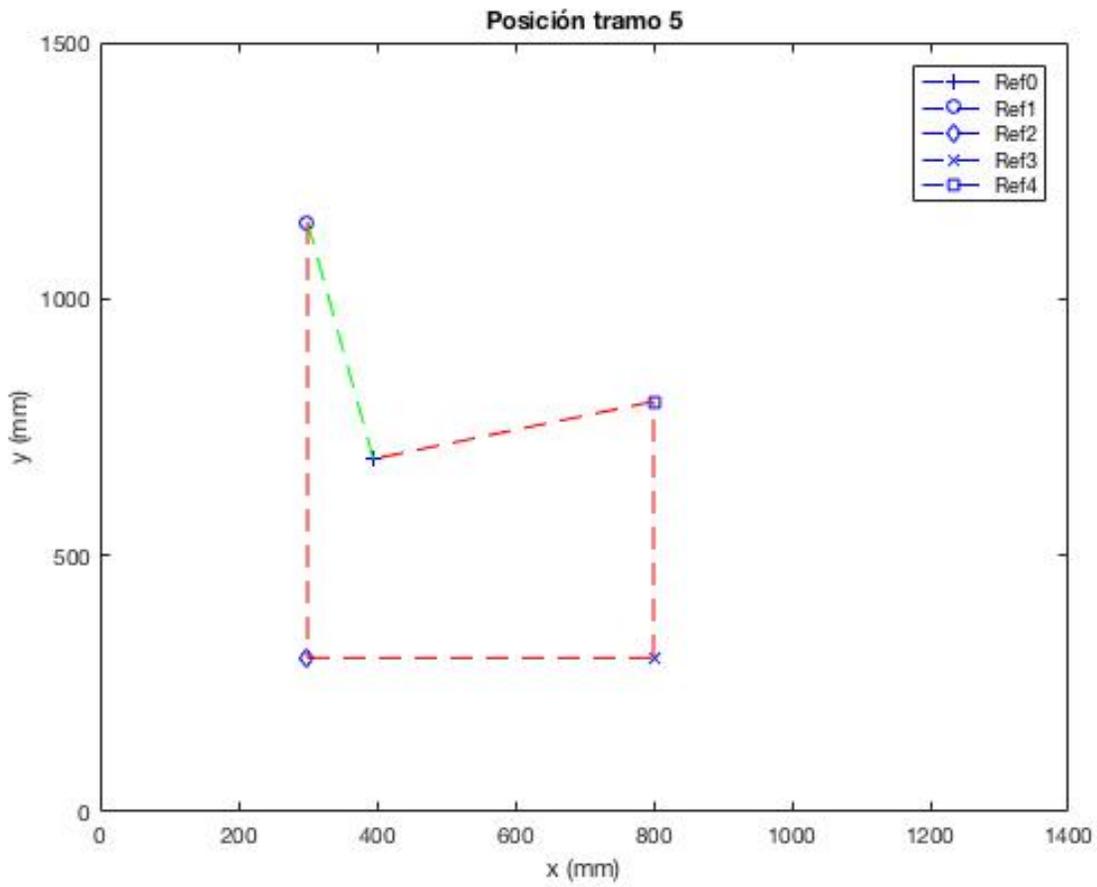


Figura 6.20: Posición en el tramo 5(II)

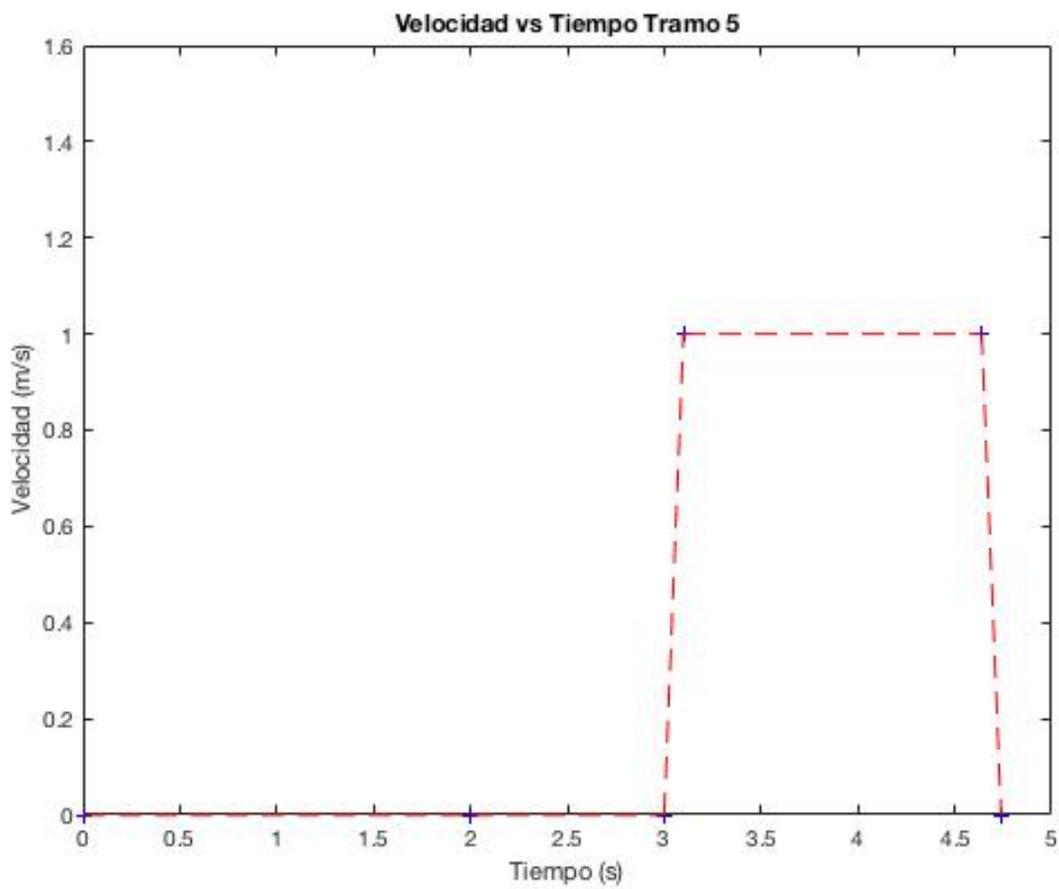


Figura 6.21: Velocidad en el tramo 5(II)

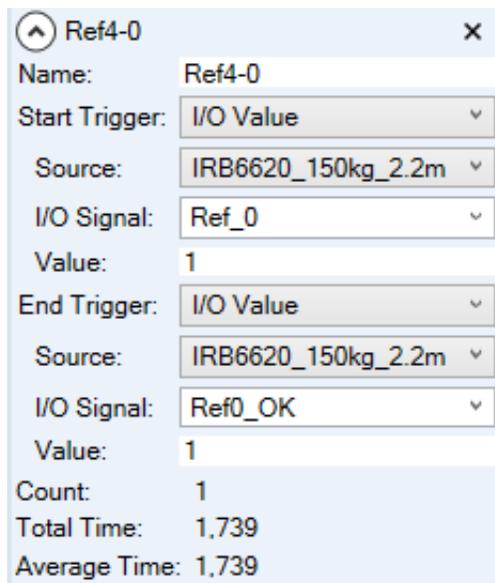


Figura 6.22: Tiempo total en realizar el tramo 5(II)

Comparando los dos movimientos, vemos que el segundo, el de regresar a la posición inicial de referencia, tarda casi un segundo más que el primero. A simple vista y analizando los gráficos de posición puede parecer incorrecto, ya que la distancia en el plano es más corta, pero hay que tener en cuenta como se ha mencionado antes que el Ref0 está a una cierta altura que no aparece en los gráficos.

---

# Capítulo 7

## Conclusiones

En este capítulo se van a analizar los objetivos que se expusieron al inicio del proyecto para confirmar que se han alcanzado con éxito. Esto es, se deben haber cumplido estos objetivos de forma satisfactoria, y se analizará si existe alguna forma de mejorarlos. Recordamos los objetivos iniciales:

- La correcta implementación del laser tracker para la medición de la mesa con la máxima precisión posible.
- Adaptación del sistema para albergar posibles modificaciones o mejoras.
- Implementación del PLC que gobernará todo el proceso.
- Diseño del mecanismo a través del cual el robot y el ordenador podrán transferirse las variables necesarias.
- Realización de pruebas con el software.
- Realización de pruebas con el robot.
- Validación y comprobación de resultados experimentales.

En términos globales se puede decir que se han cumplido los objetivos propuestos y el robot es capaz de realizar el proceso de soldadura de forma eficiente. A continuación se evaluarán los objetivos individualmente.

En primer lugar, las medidas realizadas con el laser tracker han sido satisfactorias. Esto se debe a que de no ser así, el robot no se movería a los puntos de referencia deseados, cuando claramente se muestra en el apartado de tests que si que lo cumple.

En segundo lugar, el sistema está abierto a posibles mejoras. Dada la sencillez de la estructura con la que se ha intentado implementar los programas, es sencillo hacer modificaciones sobre ellos. Además, se puede mejorar la pieza a soldar, haciéndola más realista al combinar puntos en distintos planos del eje z.

En cuanto a la implementación del PLC, puede verse en el apartado de tests y resultados que ha sido diseñado de forma correcta. El robot sigue paso a paso las órdenes que el PLC le manda, y el resultado es la soldadura completa de la pieza, junto con las opciones de realizar uno o varos ciclos según el interés del operario.

Una vez todo el proceso programado, se ha simulado tanto en TIA Portal como en RobotStudio para comprobar su funcionamiento. Las señales de salida del PLC se activan correctamente a medida que se van activando las condiciones necesarias, y el robot realiza

los movimientos sólo al recibir las órdenes del PLC.

Por último, se puede ampliar el proyecto a pesar de haber cumplido los objetivos. Una ampliación del proyecto podría ser el diseño de un PLC que cambie la pieza a soldar tras realizarse la soldadura completa de la pieza anterior. Sería necesario un robot adicional para efectuar este proceso, pero se podría diseñar de igual manera en RobotStudio comprobar su correcto funcionamiento.

# Bibliografía

- [1] Emilio García Moreno. *Automatización de procesos industriales*. Alfaomega Valencia, 2001.
- [2] Dolores Blanco, Salah Al Ansari, Cristina Castejón Sisamón, Beatriz López Boada, and Luis E Moreno. Manfred: robot antropomórfico de servicio fiable y seguro para operar en entornos humanos. 2005.
- [3] Martha Rodriguez. *Inteligencia artificial*. 2010.
- [4] Fernando Reyes. *Robótica-Control de robots manipuladores*. Alfaomega Grupo Editor, 2011.
- [5] Josep Balcells, José Luis Romeral, and José Luis Romeral Martínez. *Autómatas programables*, volume 1089. Marcombo, 1997.
- [6] Pilar Mengual, Carlos Astorga Vega, Juan Luis Rodríguez Parga, and María Cristina Montoya Rivero. *Step 7: una manera fácil de programar PLC de Siemens*. Alfaomega, 2010.
- [7] Rafael Aracil, Roque J Saltafén, José Ma Sabater, and Oscar Reinoso. Robots paralelos: Máquinas con un pasado para una robótica del futuro. *RIAI*, 3(1):16–28, 2006.
- [8] Larry Jeffus. *Soldadura: principios y aplicaciones*, volume 3. Editorial Paraninfo, 2009.
- [9] MA Pacheco, A Arnanz, and JR Perán. Monitorización y detección de fallos en robots de soldadura. *Actas de las I Jornadas de Diagnósis, Razonamiento Cualitativo y Sistemas Socioeconómicos, Valladolid*, 2001.
- [10] Sunyoon Choi and Susan P Ashdown. 3d body scan analysis of dimensional change in lower body measurements for active body positions. *Textile Research Journal*, 81(1):81–93, 2011.
- [11] Andrés Vivas. *Robótica paralela: aplicaciones industriales, modelado y control*. Universidad del Cauca. Popayán, 2005.
- [12] M Sc Kryscia Ramírez Benavides. *Cinemática directa del robot*.