

MÁSTER UNIVERSITARIO EN INGENIERÍA INDUSTRIAL

TRABAJO FIN DE MÁSTER

***GEMELO DIGITAL Y PUESTA EN
MARCHA VIRTUAL CON (HiL) DE
CÉLULA DE SOLDADURA ROBOTIZADA***

**Alumno/Alumna
Director/Directora
Departamento
Curso académico**

*Sánchez, García, Roberto
Orive, Revillas, Darío
Ingeniería de Sistemas y Automática
2018/2019*

Bilbao, 28, 05, 2019

DATOS BÁSICOS DEL TRABAJO FIN DE MASTER

Alumno: Roberto Sánchez García

Director del proyecto: Darío Orive Revillas

Departamento: Departamento de Ingeniería y Sistemas de Automática

Título del trabajo: Gemelo digital y puesta en marcha virtual con (HiL) de célula de soldadura robotizada

Resumen: Este documento presenta el modelado de un gemelo digital de una célula de soldadura robotizada y su puesta en marcha virtual. El gemelo digital se modela con la herramienta de ingeniería Tecnomatix Process Simulate V14.1. Para el código de control de la célula se hace uso de la herramienta de ingeniería TIA Portal V15.1, con la que también se realiza el diseño de un HMI. La arquitectura implementada en la puesta en marcha virtual es con "Hardware in the Loop" (HiL). Se hace uso de un PLC S7-1500 como controlador I/O y se emplea una tarjeta SIMIT Unit para realizar la simulación de los dispositivos IO de periferia de la célula en PROFINET IO. El mapeado de las señales es realizado con SIMULATION Unit V9.1.

Palabras clave: Gemelo digital, PLC, PROFINET-IO, Hardware In the Loop, Industria 4.0, SIMIT Unit

Izenburua: Soldadurarako errobotizatutako zelularen biki digitala eta martxan ipintze birtuala (HiL)-ekin

Laburpena: Dokumentu honek soldadurazko zelula robotizatu baten biki digitalaren modelaketa eta haren abiarazte digitala aurkezten du. Biki digitala Tecnomatix Process Simulate V14.1 ingeniaritza tresnarekin modelatu da. Zelulazko kontrol koderako TIA Portal V15.1 ingeniaritza tresna erabili da, baita HMI bateko diseinua egiteko ere. Martxa ipintze birtualean inplementatutako arkitektura "Hardware in the Loop"-ekin (HiL) da. Kontrolagailu bezala PLC S7-1500 erabiltzen da, eta zelularen periferiako IO gailuen simulazioa PROFINET IO-n, SIMIT Unit txartela batekin egiten da. Seinaleen mapeaketa SIMULATION Unit V9.1 erabiliz gauzatu da.

Gako hitzak: Biki digital, PLC, PROFINET-IO, Hardware In the Loop, Industria 4.0, SIMIT Unit

Title: Digital twin and virtual commissioning with (HiL) for robotized welding cell

Abstract: This document presents the modelling of the digital twin of a robotized welding cell and its virtual commissioning. The digital twin is modelled using the Tecnomatix Process Simulate V14.1 engineering tool. TIA Portal V15.1 engineering tool is used for the development of the control code of the cell, as well as for the design of a HMI. The architecture implemented in the virtual commissioning is with “Hardware in the Loop” (HiL). A PLC S7-1500 is used as a I/O controller and a SIMIT Unit card is employed in order to carry out the cell’s peripheral IO devices’ simulation in PROFINET IO. The mapping of the signals is performed by means of SIMULATION Unit V9.1.

Key words: Digital twin, PLC, PROFINET-IO, Hardware In the Loop, Industry 4.0, SIMIT Unit

ÍNDICES

Tabla de contenido

1. INTRODUCCIÓN	8
2. CONTEXTO	10
3. OBJETIVOS Y ALCANCE DEL TRABAJO	16
3.1. Modelado de la célula de soldadura	16
3.2. Diseño del sistema de control	17
3.3. Virtual commissioning	18
4. BENEFICIOS QUE APORTA EL TRABAJO	20
4.1. Beneficios sociales	20
4.2. Beneficios económicos	20
4.3. Beneficios técnicos	21
4.3.1. Red de comunicaciones	21
4.3.2. Hardware	21
4.3.3. Software	21
4.3.4. HMI	23
5. ANÁLISIS DE ALTERNATIVAS	24
5.1. Alternativas del proyecto	24
5.1.1. Arquitectura de puesta en marcha virtual	24
5.1.2. Modelado de la célula de soldadura	24
5.1.3. Sistema de control	26
5.1.4. Red de comunicaciones	26
5.1.5. HMI	27
5.2. Decisión a las alternativas	28
6. ANÁLISIS DE RIESGOS	30
7. DESCRIPCIÓN DE LA SOLUCIÓN: DISEÑO Y SELECCIÓN DE LOS ELEMENTOS DE LA CÉLULA	33
7.1. Diseño de la célula	33
7.2. Selección de elementos de la célula	35
7.3. Selección de los robots	38
7.4. Diseño de los objetos	41
7.5. Diseño del sistema de control	42
8. METODOLOGÍA. DISEÑO	43
8.1. Entorno de trabajo	43

8.2. Modelado de la célula de soldadura.....	44
8.2.1. Definición e inserción de los componentes.....	44
8.2.2. Cinemática de los elementos.....	45
8.2.3. Diseño de las herramientas de los robots.....	49
8.2.4. Operaciones de los elementos de la célula.....	52
8.2.5. Programas de los robots.....	57
8.2.6. Sensorización.....	60
8.2.7. Flujo de materiales.....	61
8.3. Control desde PLC.....	64
8.3.1. Hardware de Control.....	64
8.3.2. Red de comunicaciones.....	64
8.3.3. Software de Control.....	65
8.4. Puesta en Marcha Virtual mediante HiL.....	76
8.4.1. Configuración Hardware.....	76
8.4.2. Configuración Software.....	78
9. DESCRIPCIÓN DE LAS TAREAS.....	84
10. DESCARGO DE GASTOS.....	88
11. CONCLUSIONES.....	91
12. FUENTES DE INFORMACIÓN.....	93
ANEXO I: LISTA COMPLETA DE VARIABLES DE LA CÉLULA ROBOTIZADA.....	94
ANEXO II: CÓDIGO DE CONTROL DEL PLC.....	96
ANEXO III: CÓDIGO DEL ROBOT MANIPULADOR.....	112
ANEXO IV: CÓDIGO DEL ROBOT SOLDADOR.....	128
ANEXO V: PRESUPUESTO DE LA PUESTA EN MARCHA VIRTUAL Y REAL.....	143
ANEXO VI: DIAGRAMA DE GANTT Y DE RED DEL PROYECTO.....	144

Listado de figuras

Fig. 1 Línea de montaje completa	8
Fig. 2 Tecnologías de la Industria 4.0	11
Fig. 3 Smart Plataform.....	12
Fig. 4 Gemelo digital.....	12
Fig. 5 Virtual Commissioning.....	13
Fig. 6 Niveles de puesta en marcha virtual	14
Fig. 7 SiL vs. HiL.....	15
Fig. 8 Calderín de agua caliente sanitaria.....	16
Fig. 9 PLC (izquierda) y PC industrial (derecha).....	17
Fig. 10 Ejemplo panel operador HMI	18
Fig. 11 Arquitecturas para puesta en marcha virtual.....	19
Fig. 12 Alzado y planta línea de producción.....	34
Fig. 13 Cinta transportadora	35
Fig. 14 Diseño simplificado alimentador de tubos.....	36
Fig. 15 Mesa de trabajo.....	37
Fig. 16 Layout de la célula	37
Fig. 17 Robot KUKA KR30-3 y espacio de trabajo.....	39
Fig. 18 Robot KUKA KR8 R2100 arc HW y espacio de trabajo	40
Fig. 19 Controlador KUKA KR C4.....	40
Fig. 20 Base y tubos del calderín	41
Fig. 21 Red de comunicaciones de la célula	42
Fig. 22 Demostrador.....	43
Fig. 23 Directorio del proyecto.....	44
Fig. 24 Cinemática del robot manipulador.....	46
Fig. 25 Conjunto de elementos "Base".....	47
Fig. 26 Cinemática de la mesa giratoria	48
Fig. 27 Cinemática del vallado de seguridad	49
Fig. 28 Herramienta doble de manipulación (izquierda) y boquilla de soldadura (derecha)	50
Fig. 29 Cinemática de la garra de manipulación	51
Fig. 30 Movimiento cinemático garra para bases	51
Fig. 31 Movimiento relativo garra para tubos.....	52
Fig. 32 Tipos de operaciones para robots	53
Fig. 33 Panel configuración de operación de movimiento de un objeto	53
Fig. 34 Nueva Operación de un elemento.....	54
Fig. 35 Panel de configuración de operación de Pick & Place.....	54
Fig. 36 Path Editor	55
Fig. 37 Editor de operaciones de proceso continuo.....	56
Fig. 38 Árbol de operaciones de la célula.....	57
Fig. 39 Conjunto de operaciones del robot manipulador (izquierda) y soldador (derecha).....	58
Fig. 40 Ventana de configuración del MAIN del robot manipulador (izquierda) y soldador (derecha)	58
Fig. 41 Diagramas de flujo de los programas de los robots	59
Fig. 42 Panel de configuración de un sensor fotoeléctrico	60
Fig. 43 Posición de los sensores dentro de la célula de soldadura	61
Fig. 44 Visor del flujo de material.....	63

Fig. 45 Red de comunicaciones	65
Fig. 46 Secuencia GRAFCET	67
Fig. 47 Pantalla principal HMI	73
Fig. 48 Pantalla célula de soldadura robotizada	74
Fig. 49 Pantalla GRAFCET del HMI.....	75
Fig. 50 Pantalla SIMULACIÓN del HMI	75
Fig. 51 Tarjeta SIMIT Unit PN 128	76
Fig. 52 Configuración hardware de la simulación con dos PCs	77
Fig. 53 Configuración Hardware de la simulación con un PC.....	78
Fig. 54 Hardware importado en SIMULATION Unit.....	80
Fig. 55 Ventana de configuración de arquitectura de simulación	81
Fig. 56 Panel de señales de la célula de soldadura	82
Fig. 57 Panel de simulación de Tecnomatix PS	83
Fig. 58 Diagrama de Gantt simplificado	87

Listado de tablas

Tabla 1 Matriz de clasificación de riesgos.....	31
Tabla 2 Características para selección de robots.....	38
Tabla 3 Características KR30-3	39
Tabla 4 Características KR8 R2100 arc HW	40
Tabla 5 Características KR C4	41
Tabla 6 Sensores de la célula de soldadura.....	61
Tabla 7 Parámetros entrada salida FB2.....	66
Tabla 8 Programas de los robots.....	71
Tabla 9 Variables I/O de la célula de soldadura	72
Tabla 10 Compatibilidad entre herramientas de ingeniería para mapeado de señales.....	79
Tabla 11 Tareas principales del proyecto.....	84
Tabla 12 Listado de tareas críticas	85
Tabla 13 Presupuesto del proyecto.....	89

1. INTRODUCCIÓN

Este documento presenta el modelado de gemelo digital y puesta en marcha virtual de una célula robotizada para soldar los tubos de agua caliente y fría a la base de un calderín de agua caliente sanitaria. Esta célula formará parte de una línea de montaje del calderín completo (figura 1).

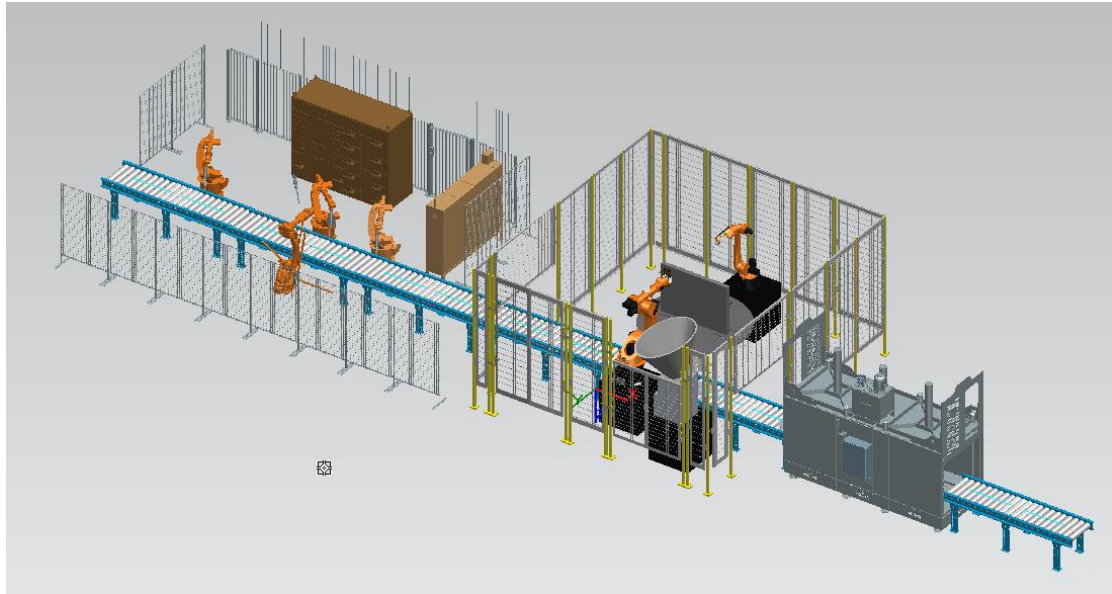


Fig. 1 Línea de montaje completa

La puesta en marcha virtual de la célula se divide en tres partes.

Primeramente, se realiza el modelado del gemelo digital (*Digital twin*) de la célula con la herramienta de ingeniería Tecnomatix Process Simulate. Esta herramienta permite la optimización inicial del proceso de ensamble, es decir, mediante la creación de un gemelo digital se pueden verificar la correcta posición de todos los elementos, que estos cumplan los requisitos del proyecto, análisis de colisiones de los mismos e incluso tiempos de fabricación. Todo ello antes de poner en marcha la célula real, lo que da como resultado una puesta en marcha real más rápida y fiable. Por tanto, se escogen los elementos necesarios y se configuran para poder realizar las operaciones necesarias para obtener el producto final.

Por otra parte, se diseña e implementa el programa de control de la célula. Para ello se hace uso de la herramienta de ingeniería TIA Portal. Esta herramienta hace que puedan llevarse a cabo todas las tareas de ingeniería, puesta en marcha, configuración de todos los componentes de automatización y accionamientos a través de una única

plataforma. Se escoge como Controlador IO un PLC de la gama S7-1500 de Siemens. Este se encarga de controlar los Dispositivos IO de periferia, tales como los controladores de los robots y un módulo ET 200M. Los dos primeros ejecutan los programas de control de los dos robots existentes en la célula. El ET 200M, por otro lado, se encarga de la sensorización de toda la célula y resto de componentes de la misma. La arquitectura de comunicaciones por la que se comunicarán todos los dispositivos será PROFINET-IO.

Se realiza también el diseño de un sistema de supervisión SCADA. Además de las funciones habituales de un SCADA como son la captura y presentación de variables, se le han añadido otras adicionales que ayudan a la realización de la simulación. Esta implementación permite una mayor versatilidad en la fase de simulación, ya que se dispone de una pantalla para poder realizar la generación de nuevas piezas.

Finalmente, se realiza la puesta en marcha virtual del gemelo digital. Actualmente existen dos tipos. Por un lado, se tiene el "Software in the Loop" (SiL), con el que se realiza la simulación del gemelo digital con un Controlador Virtual (VR). Por otro lado, está el "Hardware in the Loop" (HiL), el cual hace uso de un PLC real para realizar el control contra el gemelo digital. En este proyecto se implementa la arquitectura HiL para comprobar el correcto funcionamiento del mismo. Para la simulación de los diferentes dispositivos IO de periferia que controlará el PLC se hace uso de una tarjeta SIMIT Unit PN, la cual emulará el comportamiento de los dispositivos anteriormente citados. Así mismo, para cargar la configuración del hardware en esta tarjeta se utiliza la herramienta de ingeniería SIMULATION Unit, perteneciente también a la plataforma de productos de Tecnomatix. Este software permite a su vez realizar el mapeado y comunicación entre la herramienta de modelado Tecnomatix PS y la tarjeta SIMIT Unit PN.

2. CONTEXTO

La historia de la industria es un claro reflejo de avance y evolución de la sociedad, limitándose en sus comienzos al trabajo humano o animal. Sin embargo, es a mediados del siglo XVIII cuando comienza a desarrollarse el concepto de la industria, pues a partir de ese momento la manera realizar los trabajos varía, dando lugar al comienzo de las revoluciones industriales.

Todo comienza pasado el año 1760 cuando aparece la máquina de vapor, dando comienzo a la Primera Revolución Industrial. Este invento, sumado al descubrimiento de nuevos minerales como el hierro y el carbón, generan un cambio en la tecnología del momento en cuanto a la producción de energía se refiere, modificando la forma de trabajar en muchos sectores. Gracias a esto nace el uso de vapor en la mecanización de las tareas, eliminando a los humanos la necesidad de realizar las mismas, lo que se tradujo en un aumento de la calidad de vida para los mismos.

A medida que la producción crecía, resultaba necesario buscar nuevas formas de trabajo basadas en un menor coste y tiempo de operación. Fue a mediados del siglo XIX cuando Henry Ford desarrolla la idea de producción en serie y división de tareas cambiando por completo los procesos de operación en las industrias. Al mismo tiempo se descubre como fuente de energía la electricidad, junto a otras como el gas o el petróleo, y se consiguen grandes avances en el sector de la comunicación (radio y teléfono), dando lugar a lo que se ha denominado la Segunda Revolución Industrial.

A mediados del siglo XX el desarrollo de la electrónica, la informática y las telecomunicaciones permitía crear lo que hoy día se conoce como Tecnologías de la Información y Comunicación (TIC) e iniciar la Tercera Revolución Industrial. Esto provoca cambios en todo el mundo y da lugar al nacimiento de la globalización, la automatización de los procesos e internet.

Tras casi medio siglo de avances de la mano de la tercera revolución, es en la Feria Tecnológica de Hannover del 2011 donde aparece el concepto de la Industria 4.0. Nace como una corriente filosófica, en la cual van apareciendo nuevos conceptos para la aplicación de nuevas tecnologías al sector industrial.

La Cuarta Revolución Industrial, también conocida como Industria 4.0, está cambiando la forma en que las líneas de producción operan y, por lo tanto, los entornos en los se ven obligados a competir.



Fig. 2 Tecnologías de la Industria 4.0

La filosofía 4.0 trae consigo la idea de una nueva revolución que combine técnicas avanzadas de producción y operaciones con tecnologías inteligentes que se integrarán en las organizaciones, las personas y los activos. Esta pretende promover la aparición de tecnologías como la analítica de grandes cantidades de datos, la inteligencia artificial, las tecnologías cognitivas, la nanotecnología y el Internet of Things (IoT), entre otras (figura 2). Todas ellas tienen en común la adquisición, almacenamiento y procesamiento de datos.

A nivel industrial, esta revolución está encaminada hacia la creación de fábricas inteligentes (Smart Factory). Estas surgen de la necesidad de poder adaptar el producto a los requisitos de cada cliente, por lo que se trata de fábricas con un gran grado de flexibilidad de fabricación orientadas siempre al cliente y no al producto, como venía a ser la tendencia de los últimos años. Con este tipo de fábricas inteligentes se consigue un mayor grado de personalización y configuración del producto final, sustituyendo a la actualmente implantada producción en masa.

Para lograr esto, los sistemas de producción deben estar conectados entre sí, tanto horizontal como verticalmente (figura 3). Esto será posible introduciendo, junto a las existentes formas de comunicación, nuevas tecnologías de la Industria 4.0 como son la IoT o la introducción de la emergente computación en la niebla (fog computing) para el procesamiento de datos.

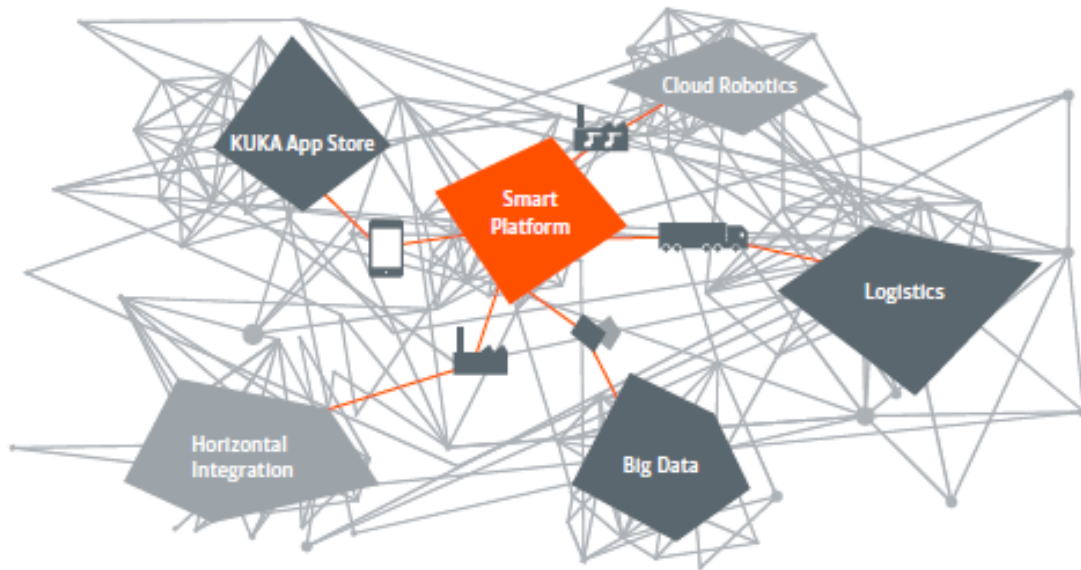


Fig. 3 Smart Plataform

En estas instalaciones los sistemas de fabricación, robots, sistemas logísticos, productos y sus componentes son en gran parte capaces de organizarse de forma autónoma. Los productos inteligentes, componentes, herramientas y máquinas son identificables y, pueden localizarse en todo momento, siendo conscientes de su historial, su estado actual y futuro para alcanzar el objetivo deseado.

Otra tecnología relacionada con la digitalización de los procesos que nace con esta filosofía es el gemelo digital (*digital twin*). Este surge a raíz de la búsqueda de mejorar las puestas en marcha de los equipos, desde un punto de vista económico y temporal, permitiendo realizar un análisis tanto físico como cinemático de cualquier elemento de una línea de producción, así como evaluar tiempos de producción, análisis de colisiones, etc., todo ello antes de realizar el montaje final de la célula.

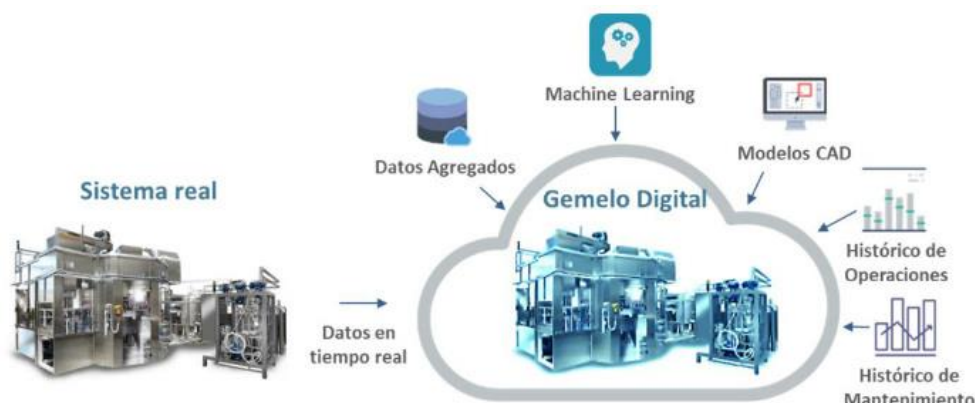


Fig. 4 Gemelo digital

La etapa de operación de un proyecto de automatización industrial comienza con el proceso de puesta en marcha, donde todos los ajustes necesarios son realizados para asegurar el correcto funcionamiento del software desarrollado y de los dispositivos involucrados en la secuencia de automatización. Este proceso es muy importante, ya que cualquier aspecto no previsto en el software podría retrasar el inicio del proyecto. De hecho, cualquier aspecto de software, hardware, comunicación o montaje no considerado da lugar a retrasos y pérdidas. Luego esta tecnología digital permite reducir el *"time to market"* o tiempo de lanzamiento del producto, ya que se consiguen reducir los tiempos de la puesta en marcha real.

La tecnología de simulación disponible permite desarrollar pruebas de validación simultáneamente al diseño de sistemas productivos. Se trata de realizar el llamado *"Virtual Commissioning"* (Puesta en marcha Virtual). Este consiste en la puesta en marcha de líneas de producción, parte de las mismas o una maquina individual mediante la conexión entre el hardware de control, que gobernará la planta real, (PLC) y el gemelo virtual del mundo real (diseño 3D virtual) para acortar los tiempos de verificación y reducir errores de manera significativa. Con esto se consigue una mejora de la calidad general en el diseño e implementación de los sistemas de producción.

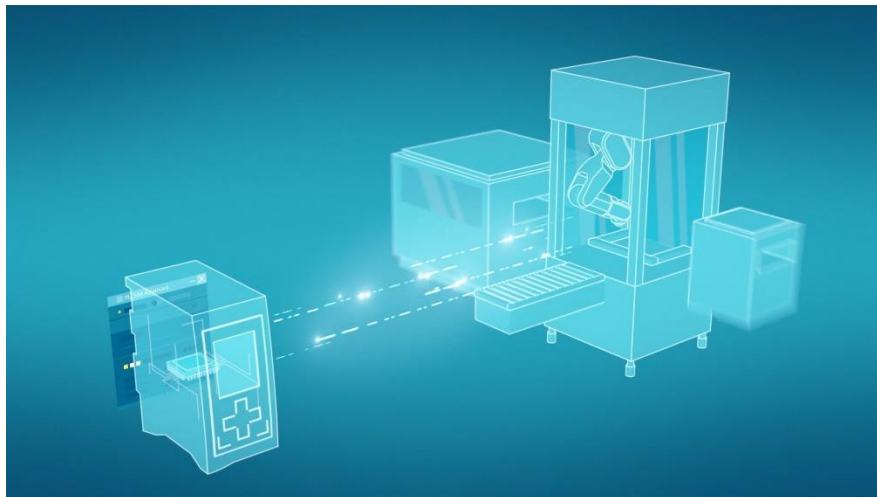


Fig. 5 Virtual Commissioning

Una de las empresas con impacto en la Industria 4.0 es Siemens. Siemens cuenta con soluciones para el diseño digital de productos, softwares para la gestión de fábricas, tecnologías de automatización y accionamientos. El PLM (*Product Lifecycle Management*) de Siemens cuenta con una amplia cartera de productos para administrar el ciclo de vida completo del producto. A su vez, la filosofía de Siemens distingue entre dos tipos de puesta en marcha virtual. Por un lado, se encuentra la

puesta en marcha de modelos virtuales de máquinas, la cual se encarga de analizar el diseño mecánico y eléctrico desde un punto de vista físico. Esta se desarrolla mediante la herramienta de ingeniería NX-MCD, software de Siemens que dispone de soluciones para el diseño, simulación y fabricación.

Por otro lado, se encuentra la puesta en marcha de células o líneas de producción, en donde el proceso de producción es el objeto de estudio. Como ya se ha mencionado, se está buscando implementar una producción enfocada al cliente, donde es necesario poder realizar un producto final en función de las necesidades de cada uno. A su vez, debe estar garantizado el correcto funcionamiento de una línea de producción a la que se le realice una modificación en la configuración, quedando garantizada la calidad del producto final. Siemens desarrolla este tipo de puesta en marcha mediante su herramienta de ingeniería Tecnomatix. Tecnomatix es una completa cartera de soluciones de fabricación digital que permiten digitalizar los procesos de fabricación, consiguiendo sincronizar la ingeniería de productos, la ingeniería de fabricación, la producción y las operaciones del servicio.

La puesta en marcha de procesos virtuales se puede utilizar en dos niveles de la estructura de los sistemas de producción: en el nivel del subsistema (célula), el cual se implementa con la herramienta de ingeniería Tecnomatix Process Simulate; y en el nivel del sistema (planta), con el software Plant Simulation. Ambas herramientas perteneces a la plataforma Tecnomatix. En la figura 6 se recoge un resumen con los diferentes tipos de puesta en marcha virtual.

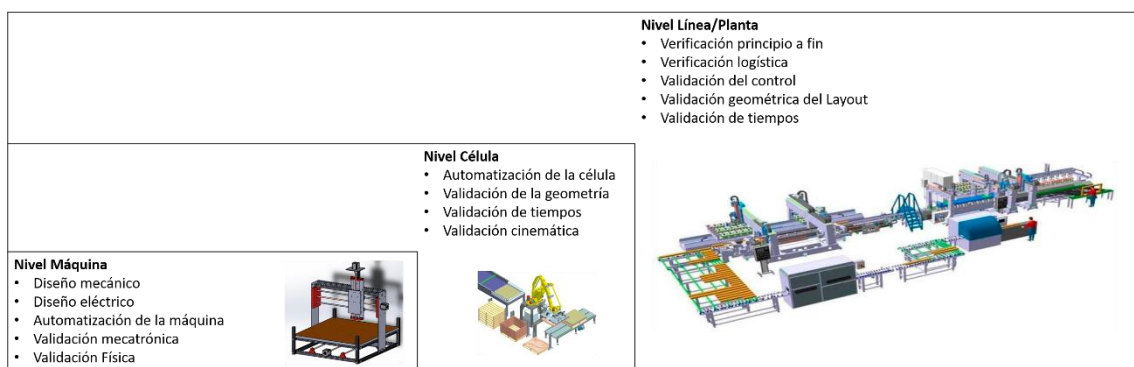


Fig. 6 Niveles de puesta en marcha virtual

Mediante la puesta en marcha virtual se consigue la validación de la cinemática, validación del código del PLC, validación de los programas de los robots, verificación del HMI, realización de pruebas de seguridad, análisis y diagnóstico del sistema, tiempos de ejecución, entre otras.

Para lograr la puesta en marcha virtual es necesario realizar un enlace o conexión entre la herramienta de ingeniería donde se ha modelado el sistema físico real y el controlador lógico programable que realiza el control del gemelo digital. Se distinguen dos tipos de arquitecturas para realizar la puesta en marcha virtual. Por un lado, el “Software in the Loop” (SiL) consiste en la puesta en marcha virtual entre un gemelo digital y un PLC virtual, sin ningún tipo de conexión física a hardware. Este método es un primer acercamiento para verificar el código de control de PLC, las operaciones de los robots o análisis de colisiones.

Por otro lado, el método conocido como “Hardware in the Loop” (HiL) hace uso de un controlador real, el cual se conecta con el modelo virtual, lo que hace que se consiga un acercamiento mayor a la realidad. Con esta arquitectura se pueden analizar de manera más precisa tiempos de ciclo y el comportamiento real del controlador, yendo un paso más allá en la puesta en marcha. A continuación, se muestra un esquema de ambas arquitecturas de puesta en marcha virtual.

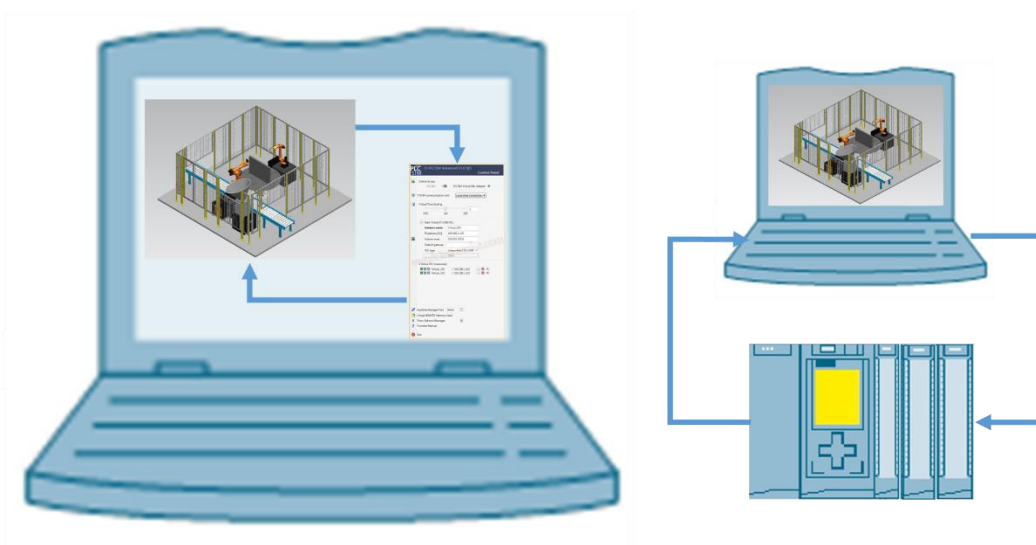


Fig. 7 SiL vs. HiL

Por tanto, la arquitectura SiL permite realizar una comprobación rápida, sin riesgos, del programa de control para una detección temprana de errores sin necesidad de disponer de hardware real y, por el contrario, la arquitectura HiL permite realizar un análisis más exhaustivo y detallado, con un mayor grado de acercamiento al sistema real en caso de disponer del hardware físico, consiguiendo una validación de las comunicaciones y tiempos de ejecución más preciso.

3. OBJETIVOS Y ALCANCE DEL TRABAJO

El objetivo del proyecto es realizar el modelado de un gemelo digital de una célula de soldadura robotizada y su puesta en marcha virtual mediante "Hardware in the Loop". El alcance del proyecto contiene tres etapas diferenciadas.

3.1. Modelado de la célula de soldadura

Se realiza el gemelo virtual de una célula de soldadura. Esta realiza la soldadura de los tubos de entrada de agua fría y salida de agua caliente a la base de un calderín de agua sanitaria (figura 8).

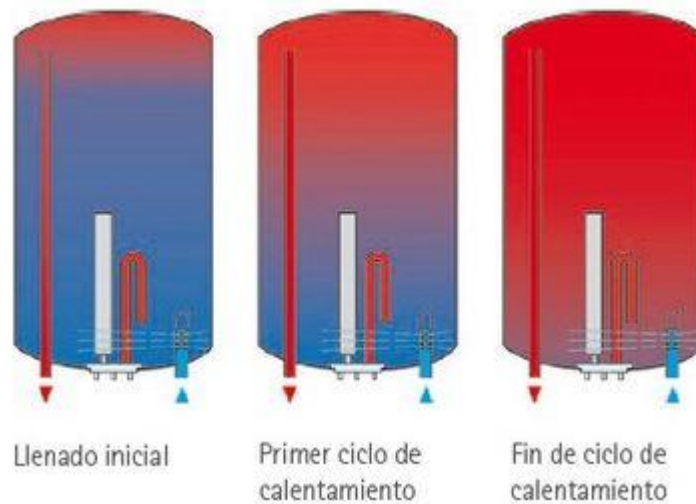


Fig. 8 Calderín de agua caliente sanitaria

La célula robotizada forma parte de la línea de montaje completa, la cual comienza con el corte de chapa metálica por prensado. Se tiene una chapa circular, con dos orificios para los tubos. Estos también llegan a la célula y un robot se encarga de su colocación en el puesto de trabajo. A su vez, existe otro robot que se encarga de la soldadura de los tubos a la base, obteniendo así el conjunto montado disponible para ser enviado a la siguiente estación de montaje. Ambos robots disponen de su controladora, la cual se comunica con el PLC que controla la célula.

Así mismo, la célula cuenta con diferentes elementos para poder llevar a cabo la correspondiente tarea tales como un alimentador de tubos, una mesa de trabajo, las correspondientes cintas de entrada y salida de piezas y un vallado de seguridad, característica indispensable en cualquier célula robotizada.

Por otra parte, se implementan todos los sensores necesarios para saber el estado actual de la célula en todo momento y que el controlador IO pueda realizar las órdenes correspondientes.

3.2. Diseño del sistema de control

El control de la célula se realiza desde un controlador IO, el cual puede ser un PLC (Programmable Logic Controller) o un PC industrial (figura 9).



Fig. 9 PLC (izquierda) y PC industrial (derecha)

Este dispositivo se encarga de mandar las ordenes a los diferentes elementos de la célula, los cuales le envían su situación en todo momento.

Como dispositivos IO de periferia se implementa, por un lado, una tarjeta de entradas/salidas. Esta se encarga de manejar la comunicación entre los diferentes elementos sensorizados de la célula y el controlador de E/S. Esta tarjeta debe tener la suficiente capacidad para poder conectar todos los sensores de la célula a sus entradas, así como disponer de las suficientes salidas como para mandar las ordenes a los accionamientos de los diferentes elementos de la célula.

A su vez, se implementan los dos controladores de los robots. La selección de estos dependerá de los robots escogidos para realizar las tareas de manipulación y soldadura, ya que se tendrán que seleccionar controladores del mismo fabricante que los robots seleccionados con capacidad suficiente para realizar el control.

Para la red de comunicaciones se busca una arquitectura común a todos los dispositivos, capaz de soportar la comunicación entre el controlador IO y los diferentes dispositivos de periferia. A su vez, este proyecto está enmarcado en las tecnologías de la Industria 4.0, por lo que se busca una red de comunicaciones capaz de implementar nuevas aplicaciones en futuras modificaciones de la célula.

Por otra parte, se debe diseñar e implementar el código de control de la célula. El lenguaje del mismo depende del dispositivo de control seleccionado. El programa de control debe de ofrecer las prestaciones suficientes para tener un buen control sobre todo el proceso de montaje y soldadura, así como cumplir con las medidas de seguridad de cualquier estación robotizada.

Por último, se diseña e implementa un interfaz de usuario HMI (figura10).



Fig. 10 Ejemplo panel operador HMI

Los HMI ofrecen visualizar con las pantallas táctiles estados, alarmas, funcionamientos y accionar bombas, válvulas y cualquier otro dispositivo a través de una interfaz gráfica y táctil.

3.3. Virtual commissioning

Finalmente, se realiza la puesta en marcha virtual del gemelo digital de la célula de soldadura.

La puesta en marcha virtual consiste en una serie de métodos que, mediante un gemelo digital, es decir, un modelo ejecutable de la máquina o proceso, permiten la realización de pruebas anticipadas del software, independientemente de si la máquina física ya está disponible o no. El control durante la simulación puede realizarse desde un PLC también virtual o con un PLC real, acercando la puesta en marcha virtual un paso más hacia la puesta en marcha real.

Basándose en un modelo tridimensional en CAD, se puede asignar a este todas las propiedades del producto real, funciones y parámetros del proceso, creando el

denominado gemelo virtual. Las posibles arquitecturas para realizar la puesta en marcha virtual del gemelo se presentan en la siguiente gráfica.

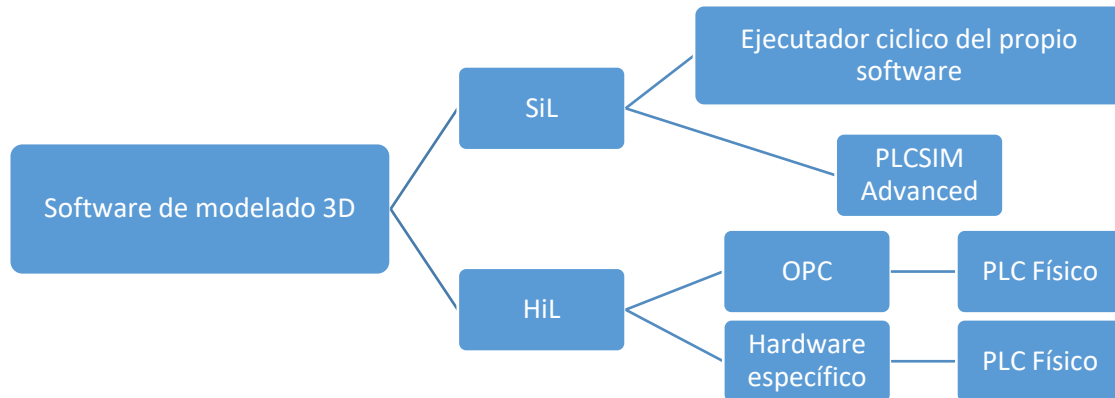


Fig. 11 Arquitecturas para puesta en marcha virtual

Como puede apreciarse, se distinguen dos tipos de arquitectura anteriormente citados que son “Software in the Loop” y “Hardware in the Loop”. En el primero se tienen el simulador propio del que disponen las herramientas de ingeniería en las que se realiza el modelo del gemelo digital y cualquier tipo de software que sea capaz de simular el comportamiento de un controlador de forma virtual. A modo de ejemplo se tiene la herramienta de ingeniería PLCSIM Advanced, el cual tiene capacidad para simular los controladores S7-1500 y ET 200SP.

Las otras dos opciones hacen uso de un PLC real para realizar la simulación, el ya mencionado “Hardware in the Loop”, utilizando la primera de estas un servidor OPC para realizar la comunicación del PLC con el modelo virtual; y la segunda haciendo uso de hardware específico, el cual se encarga de simular los diferentes dispositivos de periferia del modelo 3D, como pueden ser esclavos PROFIBUS o dispositivos IO en PROFINET. En cualquiera de estas dos opciones de arquitectura HiL, se necesita una herramienta de ingeniería para realizar el mapeado entre el gemelo digital y el hardware real. Con ambas arquitecturas se podrá realizar la simulación del modelo antes de la puesta en marcha de la máquina real.

Se hará un estudio posteriormente analizando ambas arquitecturas de “Virtual Commissioning” descritas en este apartado para elegir con cual se realizará la simulación.

4. BENEFICIOS QUE APORTA EL TRABAJO

En este apartado se exponen los diferentes beneficios que aporta este proyecto. Se van a dividir en tres grupos.

4.1. Beneficios sociales

La aportación de las nuevas tecnologías hace que los trabajos que requieren de esfuerzo físico queden a cargo de máquinas que sólo requieren supervisión, disminuyendo la cantidad de accidentes laborales en el sector industrial. Además, también eliminan trabajos rutinarios en plantas u oficinas, alcanzando una mayor flexibilidad de las fabricas con la ayuda de las nuevas tecnologías de la Industria 4.0.

4.2. Beneficios económicos

El proyecto de modelado y puesta en marcha virtual se ha realizado en un laboratorio con fines de investigación, por lo que no recibe ningún beneficio económico directo. Sin embargo, la aplicación de este proyecto en un entorno industrial real supone un importante beneficio económico.

Por un lado, el desarrollo de este proyecto consigue acercar a la célula a las nuevas tecnologías referentes a la Industria 4.0, las cuales, una vez implantadas en una línea de producción real, hacen que esta sea mucho más eficiente y competitiva.

Por otro lado, la simulación de una planta real hace que no se tenga que realizar el consiguiente descargo de gastos en la obtención de todos los componentes de la célula para poder realizar las pruebas de puesta en marcha. Por tanto, se consigue realizar una validación del diseño de la célula y del control antes de obtener los elementos de la misma, pudiendo rechazar un proyecto que no es viable sin haber realizado su correspondiente implantación. Además, la programación fuera de línea es la mejor manera de maximizar la rentabilidad de inversión de los sistemas de robots. La puesta en marcha virtual permite efectuar la programación del robot en un ordenador en la oficina sin interrumpir la producción.

Por otra parte, la industria actualmente está enfocada al producto. Cada vez son más las necesidades que existen por adecuar el producto a las características que el cliente demanda. Por tanto, se vuelve necesaria la capacidad para realizar cambios en líneas de montaje y fabricación en cuestión de horas. Mediante la puesta en marcha virtual es

posible realizar modificaciones y pruebas en tan sólo unas horas sin necesidad de paradas en la línea de producción real.

4.3. Beneficios técnicos

4.3.1. Red de comunicaciones

Uno de los beneficios de este proyecto son las prestaciones del sistema de comunicaciones PROFINET. PROFINET es el estándar abierto de Ethernet industrial según la norma IEC 61784-2. Con PROFINET es posible conectar diferentes dispositivos, mejorando tanto la velocidad como la seguridad de sus comunicaciones. A su vez, una red de comunicaciones PROFINET en el nivel E/S mejora la escalabilidad en las infraestructuras, el acceso a los dispositivos de campo a través de la red y ofrece una ejecución de las tareas de mantenimiento y prestación de servicio desde cualquier lugar. Con la implantación de la arquitectura PROFINET-IO se busca un acercamiento a las futuras tecnologías basadas en la industria 4.0, como son la adquisición, almacenamiento y procesamiento de datos, visión artificial, entre otras.

4.3.2. Hardware

El diseño de la célula de soldadura se realiza utilizando un controlador de última generación, un PLC S7-1516F. Esta es una CPU de seguridad, idónea para aquellas aplicaciones de seguridad en las que se requieren gran memoria de datos y programa. Esta gama de controladores ofrece el más alto nivel de eficiencia y de usabilidad para aplicaciones de rango medio y alto en máquinas de sistemas de automatización. Estas CPUs tienen suficiente memoria para cada aplicación y cuentan con un eficiente análisis de fallo desde el display que tienen incorporado en la propia CPU. Por otra parte, para la realización de puesta en marcha virtual de un gemelo digital es necesaria la utilización de un PLC de esta gama o de la gama S7-1200. Es por todo esto que este controlador ofrece un gran beneficio al proyecto.

4.3.3. Software

Se distinguen dos apartados. Por un lado, se encuentran las ventajas que ofrece la herramienta de ingeniería Tecnomatix Process Simulate, perteneciente a la cartera de productos de Tecnomatix. Tecnomatix PS permite que los usuarios verifiquen la factibilidad de un proceso de ensamblaje. Esto permite que los ingenieros de manufactura determinen la secuencia de ensamblaje más eficiente, evitando las posibles colisiones e identificando el tiempo de ciclo más corto. Tecnomatix PS proporciona la capacidad para seleccionar las herramientas más apropiadas para el

proceso, gracias a la búsqueda en una biblioteca de herramientas clasificada, efectuando pruebas de alcance virtuales y análisis de colisión, y simulando el proceso completo de ensamblaje del producto y la herramienta en conjunto.

Los principales beneficios que aporta esta herramienta son:

- Reducción de costo de cambio con la detección y comunicación temprana de problemas de diseño de producto
- Reducción del número de prototipos físicos gracias a la validación virtual adelantada
- Optimización de tiempos de ciclo mediante simulación
- Aseguramiento de procesos ergonómicamente seguros
- Reducción de costo gracias al re-empelo de herramientas estándares e instalaciones
- Minimización de riesgos de producción al simular varios escenarios de manufactura
- Validación temprana de los procesos de producción mecánicos y eléctricos integrados (PLC y robótica)
- Validación temprana de comisionamiento de producción en un ambiente virtual
- Incremento de calidad de proceso al emular procesos reales a través del ciclo de vida de proceso

Por otro lado, TIA Portal es la innovadora herramienta de ingeniería que permite configurar de forma intuitiva y eficiente todos los procesos de planificación y producción. Totally Integrated Automation unifica todas las tareas de control, visualización y accionamiento. Esta herramienta incorpora las últimas versiones de software de ingeniería SIMATIC STEP 7, WinCC y Startdrive. Permite realizar la configuración de red, programación del PLC, realizar análisis del estado online y diagnóstico, programación de la tecnología, programación del HMI y configurar los accionamientos. Todas estas características hacen que esta herramienta tenga cuantiosos beneficios respecto a la anterior plataforma de Siemens.

Por último, dentro del programa de control, la introducción de la opción de poder generar las piezas nuevas que llegan a la célula hace que haya una mayor flexibilidad a la hora de realizar las correspondientes simulaciones. Esta función es programada en un FC, diferente al usado en el control de la célula para su posible eliminación o no uso del mismo en el caso de implementar el código en una célula real.

4.3.4. HMI

Como ya se ha indicado, con la herramienta de ingeniería TIA Portal se puede implementar en la célula de manipulación una pantalla HMI. Esta ofrece la capacidad de visualizar mediante pantallas táctiles o CPUs estados, alarmas, funcionamientos y accionar bombas, válvulas o cualquier tipo de dispositivo, lo que da más flexibilidad al manejo de la célula.

5. ANÁLISIS DE ALTERNATIVAS

Se presentan a continuación varias alternativas de solución que pueden cumplir las especificaciones impuestas por el propio proyecto.

5.1. Alternativas del proyecto

5.1.1. Arquitectura de puesta en marcha virtual

En primer lugar, se decide el tipo de arquitectura a implementar para la realización de la puesta en marcha virtual del gemelo digital. Se establecen dos opciones principales.

Por un lado, una opción consiste en realizar la puesta en marcha virtual mediante "Software in the Loop" (SiL). Esta arquitectura realiza la simulación del modelo del gemelo digital con un PLC virtual, es decir, un PLC que se ejecuta en el mismo PC que la herramienta de ingeniería utilizada para realizar el modelado del gemelo. Con esta arquitectura se consiguen validar la correcta definición de las diferentes operaciones de los robots, así como realizar un análisis de las colisiones que se pueden ocasionar. Se puede realizar una validación completa de la cinemática del gemelo virtual y del programa de control del mismo y obtener un primer acercamiento a los tiempos de ciclo de operación, todo ello sin la necesidad de un PLC real, con el coste económico que este dispositivo supone.

La otra opción consiste en la arquitectura denominada "Hardware in the Loop" (HiL). Esta arquitectura va un paso más allá y hace uso de un PLC real. Con ello se consiguen validar todos los requisitos funcionales anteriores y, además, se consigue realizar una validación de las comunicaciones y de los tiempos de ciclo de una forma más precisa. Con HiL se da un paso más hacia la puesta en marcha real, obteniendo una visión más real del proyecto final.

5.1.2. Modelado de la célula de soldadura

Una de las principales alternativas depende de la elección de la herramienta de ingeniería para realizar el modelado de la célula de soldadura. Se proponen tres opciones dentro del mundo de modelado 3D en el ámbito de la robótica.

La primera opción es utilizar para dicho propósito Tecnomatix Process Simulate. Tecnomatix PS es una solución de fabricación digital para la verificación de procesos de fabricación en un entorno 3D. Permite validar virtualmente los conceptos de

manufactura por adelantado, a lo largo del ciclo de vida de las introducciones de nuevos productos. La capacidad de poder implementar los modelos 3D de productos y recursos facilita la validación virtual, la optimización y la puesta en servicio de procesos de fabricación complejos, lo que resulta un lanzamiento más rápido y una mayor calidad de producción. Esta herramienta pertenece al grupo de productos de Tecnomatix. Tecnomatix es una plataforma de soluciones de fabricación que ayudan a digitalizar la fabricación y el proceso de transformación de ideas innovadoras y materias primas en productos reales. Con el software Tecnomatix, se logra la sincronización entre la ingeniería de producto, la ingeniería de fabricación, la producción y las operaciones de servicio para maximizar la eficiencia de producción.

Otra alternativa es RobotStudio. RobotStudio es una aplicación de PC destinada al modelado, la programación fuera de línea y la simulación de células de robot. RobotStudio le permite trabajar con un controlador fuera de línea, que constituye un controlador IRC5 virtual que se ejecuta localmente en su PC. Este controlador fuera de línea también se conoce como el controlador virtual (VC). RobotStudio también le permite trabajar con un controlador IRC5 físico real, que simplemente se conoce como el controlador real. Cuando RobotStudio se utiliza con controladores reales, se conoce como el modo online. Al trabajar sin conexión a un controlador real o mientras está conectado a un controlador virtual, se dice que RobotStudio se encuentra en el modo fuera de línea. RobotStudio proporciona las herramientas para incrementar la rentabilidad de su sistema robotizado mediante tareas como programación y optimización, sin afectar la producción, lo que proporciona numerosas ventajas, como reducción de riesgos, arranque más rápido, transición más corta e incremento de la productividad. RobotStudio se ha construido en el VirtualController de ABB, una copia exacta del software real que hace funcionar su robot en producción. Ello permite simulaciones muy realistas, con archivos de configuración y programas de robot reales e idénticos a los utilizados en su instalación.

Finalmente, KUKA.Sim Pro es un software diseñado para programar robots de KUKA fuera de línea. Este producto permite realizar de manera virtual el control, el análisis de ciclos y la generación de programas para el robot. Se trata de un programa con un manejo intuitivo, con numerosas funciones y módulos, permitiendo planificar instalaciones de robots de forma rápida y sencilla antes de realizar la puesta en marcha real. Gracias a esta herramienta de ingeniería se puede diseñar y ajustar los tiempos de ciclo para una mayor seguridad de planificación ya que permite realizar diferentes pruebas de validación ya sean de colisiones o tiempos de ciclo.

En este programa es posible introducir datos CAD de otros sistemas o elaborar componentes con las herramientas CAD predefinidas en el programa. También es posible el uso de los amplios catálogos electrónicos que suministra la marca, o también se pueden descargar los modelos de Internet. La mayoría de componentes del catálogo electrónico son variables en cuanto a su dimensionado. Con lo que es posible por ejemplo cargar una valla de protección y adaptar la altura y anchura de ésta a las necesidades. Este modelo modular evita que los componentes tengan que ser dibujados de nuevo y supone por lo tanto un ahorro considerable de tiempo.

A su vez, cuenta con una interfaz OPC-UA para realizar la conexión PLC de Beckhoff TwinCAT, CodeSys o Siemens PLCSIM Advanced.

5.1.3. Sistema de control

Otra alternativa es la elección del elemento de control de la célula. Este podría tratarse de un PC industrial o de un PLC. El PC hace uso del SoftPLC, que es un software para usar un ordenador como PLC. La ventaja de este tipo de solución es principalmente el poder aprovechar el PC para realizar el software de control y a su vez funcionar como controlador del gemelo digital. Sin embargo, este hardware es costoso y la codificación del código de control se programa en lenguaje de alto nivel.

El PLC por otra parte es un dispositivo diseñado específicamente para realizar el control de este tipo de aplicaciones. Utiliza lenguajes que son más sencillos que el de un PC, pero solo podrá reutilizarse para proyectos relacionados con la automatización.

5.1.4. Red de comunicaciones

La siguiente alternativa pasa por la elección del tipo de red de comunicaciones. Las dos principales opciones son PROFIBUS-DP y PROFINET-IO, ya que son dos de los tipos de redes más usados en PLCs de la marca Siemens. Cabe mencionar que existen otro tipo de redes como Ethercat o PowerLink, sin embargo, la gama de controladores escogidos para la realización del proyecto hace que las dos opciones más viables sean las dos primeras descritas.

PROFIBUS-DP es un estándar de red de campo abierto e independiente de proveedores, el cual permite una amplia aplicación en procesos, fabricación y automatización. Actualmente, se calculan por encima de 20 millones de nudos instalados con tecnología PROFIBUS-DP. Son 23 organizaciones regionales (RPAs) y 33 Centros de Capacidad en PROFIBUS (PCCs), ubicados estratégicamente en varios países, con la posibilidad de

proveer soporte a sus usuarios. Es por todo esto que mantener esta tecnología en la célula es una opción a tener en cuenta. No obstante, vale la pena recordar que la tecnología es estable, pero no estática.

Por otro lado, Ethernet se está afianzando en el sector industrial. Ordenadores personales, impresoras y demás equipos periféricos con tarjetas de interfaz de red Ethernet se están utilizando cada vez más en el ambiente industrial y la aceptación de Ethernet va en aumento.

PROFINET está basado en Ethernet Industrial y en algunos estándares de comunicación pertenecientes al mundo de la tecnología de la información (TI). Entre sus características destaca que es Ethernet en tiempo real, donde los dispositivos que se comunican por el bus de campo acuerdan cooperar en el procesamiento de solicitudes que se realizan dentro del bus. Con PROFINET es posible conectar diferentes dispositivos y sistemas, mejorando tanto la velocidad como la seguridad de sus comunicaciones. Por sus características, PROFINET permite la compatibilidad con comunicaciones Ethernet más propias de entornos TI, aprovechando todas las características de éstas, salvo la diferencia de velocidad que posee una comunicación Ethernet situada en una red corporativa frente al rendimiento en tiempo real que necesita una red industrial. Adicionalmente el uso del estándar PROFINET en el nivel E/S pueden proporcionar las siguientes ventajas:

1. Mejora la escalabilidad en las infraestructuras.
2. Acceso a los dispositivos de campo a través de la red. PROFINET al ser un protocolo que utiliza Ethernet en su comunicación facilita acceder a dispositivos de campo desde otras redes de una forma más fácil.
3. Ejecución de tareas de mantenimiento y prestación de servicio desde cualquier lugar. Es posible acceder a dispositivos de campo mediante conexiones seguras como por ejemplo VPN para realizar mantenimientos remotos.

5.1.5. HMI

Otra alternativa del proyecto reside en la aplicación HMI. Se tiene que decidir el dispositivo que se usará como HMI. La Interfaz Humano Máquina permite la comunicación entre estos dos, transmitiéndose información, órdenes y datos en ambos sentidos. La importancia de una buena interfaz entre usuario y máquina se basa en poder controlar las anomalías que se puedan dar en un tiempo determinado, pudiendo también ajustar los diversos parámetros seleccionados para el control de procesos.

Pese al gran desarrollo de esta tecnología en el campo de la realidad aumentada, las dos opciones factibles económicamente para este proyecto son un panel de operador o un PC.

El panel de operador es un dispositivo que ha sido fabricado explícitamente para realizar las funciones de HMI, sin embargo, son dispositivos relativamente caros. EL PC, por otra parte, requiere de la adquisición de una licencia runtime para su implementación, pero este puede usarse para otras funciones que ha sido diseñado.

5.2. Decisión a las alternativas

- Se escoge como arquitectura de puesta en marcha virtual la denominada "Hardware in the Loop", ya que se dispone de un PLC de la gama S7-1500 y de una tarjeta SIMIT Unit para realizar la simulación de los dispositivos IO de periferia, obteniendo así un acercamiento mayor al proceso real.
- La herramienta de ingeniería escogida para realizar el modelado de la célula es Tecnomatix PS. Esta herramienta, perteneciente al conjunto de productos de Tecnomatix, hace que sea la mejor opción, ya que puede realizarse la sincronización entre la ingeniería de producto, la ingeniería de fabricación, la producción y las operaciones de servicio para maximizar la eficiencia de su producción y lograr la innovación. A su vez, esta herramienta permite la implementación de robots de diferentes marcas, ventaja competitiva que hace que la etapa de diseño sea más flexible. Además, se decide realizar la puesta en marcha virtual del gemelo digital mediante "Hardware in the Loop", por lo que la herramienta que permite esta opción con más prestaciones es Tecnomatix PS.
- El proyecto se basa en el modelado de gemelo digital y puesta en marcha virtual con HiL de una célula de soldadura robotizada, por lo que el dispositivo más adecuado para llevar a cabo esta tarea será un PLC, ya que un PLC se acerca más a una solución de ingeniería real, donde las células de fabricación están gobernadas por este. Además, permite usar un lenguaje más intuitivo a la hora de programar procesos y la inversión económica necesaria será mejor que si se escoge un PC.

- Se elige como red de comunicaciones del sistema una red PROFINET-IO. Pese al amplio uso de la red PROFIBUS-DP en el sector industrial, se opta por el primer tipo de red de comunicaciones para encaminar a la célula hacia las tecnologías de la industria 4.0.
- Se escoge como HMI un PC de sobremesa, pudiendo reutilizar el mismo PC donde se encuentra el gemelo digital de la célula. Sin embargo, no se descarta la implementación de un panel de operador en la célula de manipulación en futuros proyectos.

6. ANÁLISIS DE RIESGOS

Los principales riesgos que se pueden generar en el proyecto son los que derivan de fallos en el hardware y el software necesario para realizar la puesta en marcha virtual del gemelo digital.

El gemelo digital de la célula de soldadura robotizada debe asemejarse con la mayor precisión posible al modelo real, para así tener una visión lo más cercana posible al producto final. Para ello, el modelado realizado del mismo debe seguir las medidas establecidas en el layout. Sin embargo, siempre existe un cierto error entre el modelo y el proceso real.

Por otra parte, el elevado coste de los dispositivos hardware necesarios para realizar la puesta en marcha virtual mediante "Hardware in the Loop", así como todas las herramientas de ingeniería software usadas para diseñar e implementar tanto el gemelo digital como el código de control hace que un error durante la fase de diseño derive en sobrecostes que no se producirían en una puesta en marcha real.

A continuación, se van a exponer los cuatro posibles riesgos más importantes a los que se enfrenta este proyecto junto con la probabilidad de aparición de los mismos.

- a) Error en el diseño del gemelo virtual: Existe un cierto grado de error entre el gemelo digital y el proceso real. Sin embargo, un error grave en el diseño del gemelo digital supondría un error en el proceso de la puesta en marcha real del modelo, con la consecuente inversión necesaria para subsanar dicho error. Un error en el diseño puede ocasionar defectos en la fabricación del producto final, colisiones entre elementos o errores a nivel hardware. Esto supone una clara disminución de los beneficios del proyecto, por lo que supone un riesgo considerable. **Riesgo: Alto**
- b) Error en el código de control: Un error en el código de control, posiblemente causado por la realización de una puesta en marcha virtual y no real, supondría la necesidad de realizar un rediseño del código de control, con su correspondiente coste económico. **Riesgo: Alto**
- c) Error durante la puesta en marcha virtual con HiL: La realización de la puesta en marcha virtual mediante "Hardware in the Loop" requiere el uso de un PLC real, así como de una tarjeta que simule los dispositivos IO de periferia de la célula. La

rotura de uno de estos dos dispositivos y su necesario reemplazamiento hace necesaria una nueva inversión en material, junto con un aumento del plazo de realización del proyecto. **Riesgo: Bajo**

- d) Selección de las herramientas de ingeniería: En este proyecto se escogen las herramientas de ingeniería Tecnomatix PS para realizar el modelado del gemelo digital y TIA Portal para el diseño del código de control. Se tiene que asegurar que ambas herramientas tengan la suficiente potencialidad para poder realizar el alcance del proyecto y que ambas sean compatibles para realizar la puesta en marcha virtual, disponiendo de los mecanismos necesarios para la realización de la misma. **Riesgo: Muy alto**

Tabla 1 Matriz de clasificación de riesgos

Probabilidad		Impacto				
		Insignificante	Menor	Moderado	Mayor	Catastrófico
Casi seguro	81-100%	Alto	Alto	Muy alto	Muy alto	Muy alto
Probable	61-80%	Medio	Alto	Alto	Muy alto	Muy alto
Posible	41-60%	Bajo	Medio	Alto B	Muy alto A	Muy alto
Improbable	21-40%	Bajo	Bajo	Medio	Alto	Muy alto D
Raro	1-20%	Bajo	Bajo C	Medio	Medio	Alto

Por tanto, se deberá prestar atención al modelado del gemelo digital, ya que un error significativo en el mismo hacer perder la ventaja competitiva que la puesta en marcha virtual ofrece, perdiendo el ahorro de tiempo en la realización de la puesta en marcha real. Para ello, se deberán realizar las diferentes validaciones de toda la cinemática de la

misma mediante simulaciones previas con las herramientas disponibles dentro de la propia herramienta de ingeniería Tecnomatix PS.

El código de control deberá ser correctamente analizado y verificado, no sólo desde un punto de vista funcional sino también desde el punto de vista de seguridad. Todos los mecanismos de seguridad deberán responder correctamente a las diferentes problemáticas que puedan suceder durante el funcionamiento de la célula.

Se deberá realizar un estudio previo de las herramientas de ingeniería necesarias para la realización del proyecto, ya que se trata de herramientas con un elevado coste.

7. DESCRIPCIÓN DE LA SOLUCIÓN: DISEÑO Y SELECCIÓN DE LOS ELEMENTOS DE LA CÉLULA

7.1. Diseño de la célula

Dentro de una célula robotizada se distinguen dos grandes grupos de elementos. Por un lado, los elementos activos, donde se encuentran robots, máquinas CN, manipuladores secuenciales, etc. Por otro lado, están los elementos pasivos, donde destacan mesas, alimentadores, utillajes, etc.

Las especificaciones impuestas en el proyecto detallan el uso de dos robots: uno para realizar tareas de manipulación y otro para realizar la soldadura de los diferentes elementos. El modelo y características de estos queda sujeta a la elección del diseñador de la célula. A su vez, para aumentar rendimientos de la célula y, en vistas a realizar estudios de tiempos de producción, se establece como norma que ambos robots deberán trabajar en paralelo, por lo que se opta por integrar en la célula una mesa con la capacidad necesaria para habilitar y presentar las piezas a ambos robots.

Por otra parte, la célula formará parte de una línea de producción, por lo que su tamaño queda acotado por el espacio disponible de la misma. Se dispone de 8000x8100mm en planta para poder implantar la célula. Por tanto, y por motivos de seguridad, se especifica que la mesa tendrá capacidad de giro para poder situar un robot a cada lado de la misma, separando ambos lados de trabajo por una mampara de seguridad. A continuación se muestra una imagen del alzado y planta de la línea de producción sin la célula de soldadura.

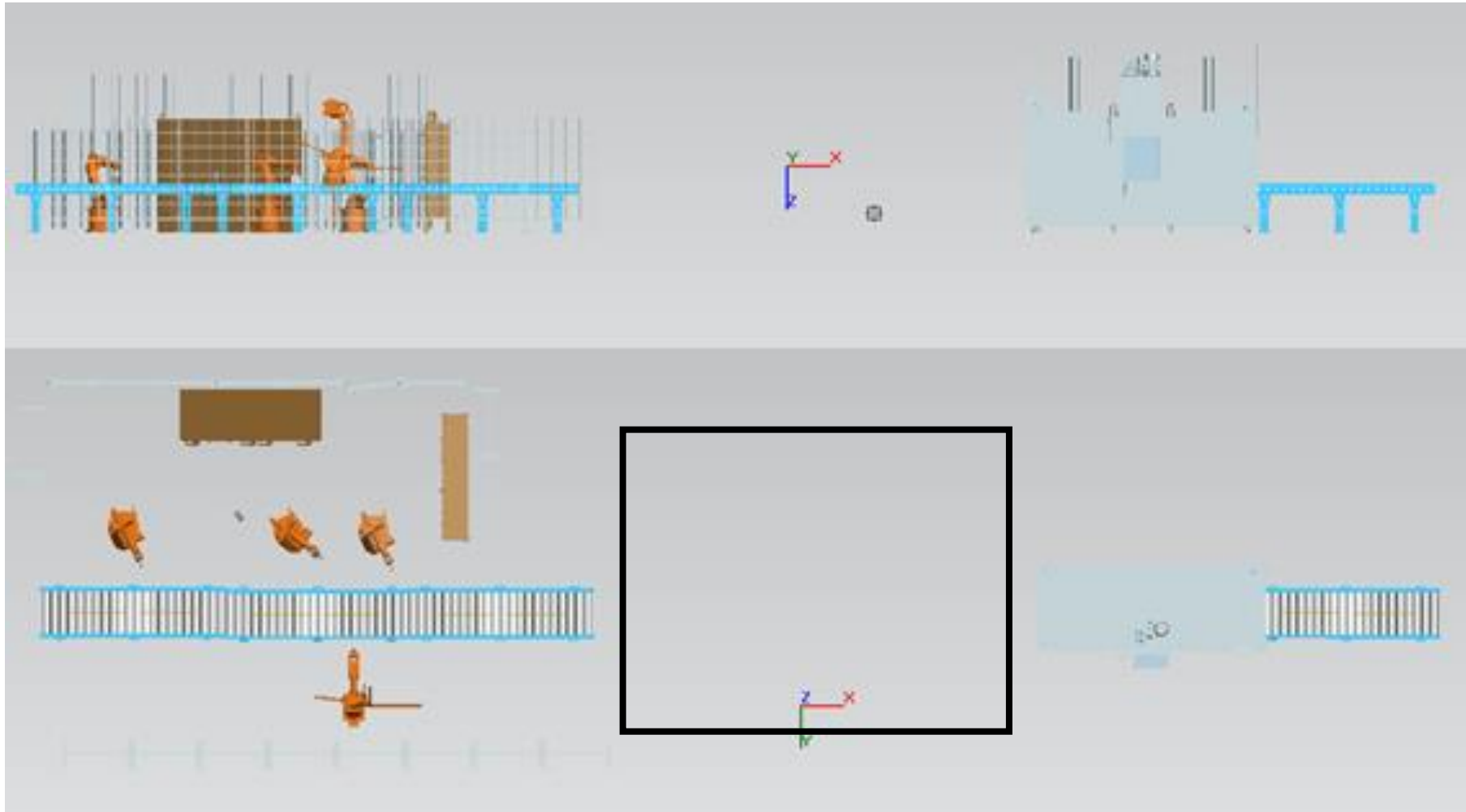


Fig. 12 Alzado y planta línea de producción

7.2. Selección de elementos de la célula

Una vez distinguidos los dos principales grupos que componen una célula y detalladas las especificaciones del proyecto, se comienza con la selección de los mismos. En primer lugar, y como ya se ha comentado en apartados anteriores, esta célula se encarga de la soldadura de los tubos de agua caliente y fría a la base de un calderín de agua caliente sanitaria. Este proceso forma parte de la línea de montaje del calderín completo, por tanto, el primer elemento dentro del grupo de elementos pasivos es el alimentador de bases, las cuales llegarán en cinta transportadora. A su vez, los conjuntos soldados también serán retirados en el mismo tipo de cinta. El modelo escogido de librerías es el mostrado en la figura 13.

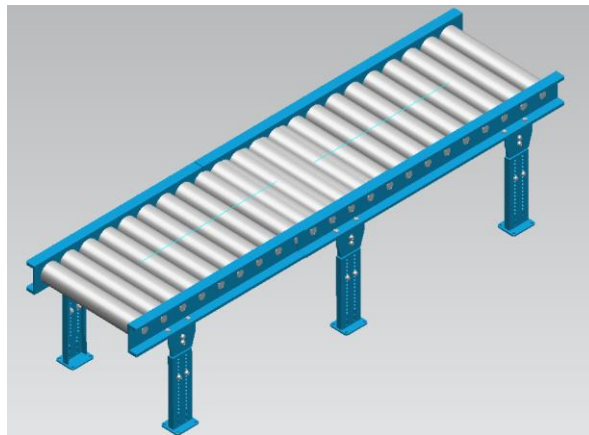


Fig. 13 Cinta transportadora

Se trata de una cinta de rodillos accionada mediante un motor asíncrono. Las bases llegan posicionadas del proceso de producción anterior (Corte en serie de bases de chapa metálica por prensado). Existen tres tamaños posibles de base.

El siguiente elemento pasivo a elegir es el dispensador de tubos. Este se encarga de regular la entrada de tubos de manera uniforme y sincronizada. Debido a que se trata de una aplicación específica, no existe un alimentador específico para este tipo de tubos, por lo que se diseña mediante la herramienta de ingeniería Solid Edge, un modelo conceptual simplificado de un alimentador de tubos vibratorio. Si se quisiera realizar el análisis de esta máquina debería usarse otra herramienta de ingeniería como es NX-MCD, con la cual se pueden analizar físicas y comportamientos mecánicos de una máquina específica. Sin embargo, este análisis queda fuera del alcance de este proyecto, por lo que sólo se realiza el diseño conceptual sin dotar al mismo de un

modelo cinemático. El mecanismo consta de una tolva de alimentación por donde se introducen los tubos e irán dispensándose de forma homogénea por los dos orificios señalados en la figura 14. El robot manipulador los irá seleccionando de dos en dos por cada nuevo conjunto a montar.

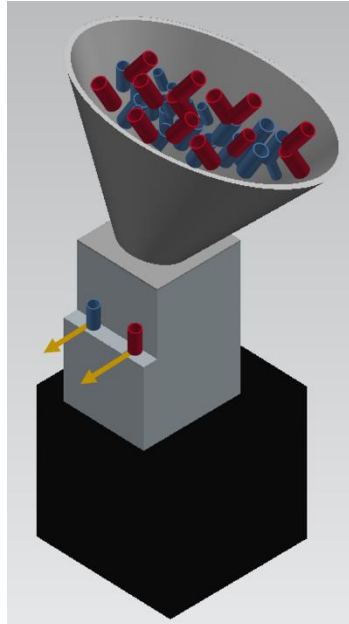


Fig. 14 Diseño simplificado alimentador de tubos

Otro de los elementos pasivos es la mesa donde se realizan los trabajos de montaje y soldadura. La elección de la misma pasa por disponer de la capacidad de tener el robot de manipulación y el robot de soldadura funcionando simultáneamente como se ha especificado, permitiendo así un mayor número de ensambles por hora. La mesa ha sido escogida de librería y es la que se muestra en la figura 15. Se trata de una mesa circular de 2500 mm de diámetro la cual es accionada mediante un motor asíncrono trifásico. Tiene capacidad de giro de 180°, mediante el cual se presentan las piezas a los dos robots.

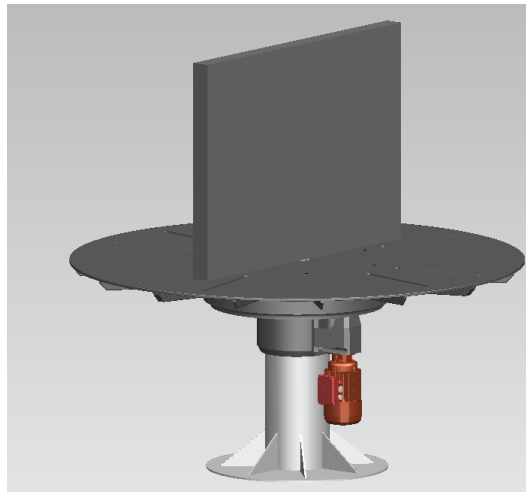


Fig. 15 Mesa de trabajo

En esta célula como elementos activos se dispone únicamente de robots, uno de manipulación y otro de soldadura. La elección de estos se verá en el siguiente apartado.

Por último, se adquiere también de librería el vallado de seguridad, elemento indispensable en cualquier célula robotizada. El layout final de los elementos es el siguiente:

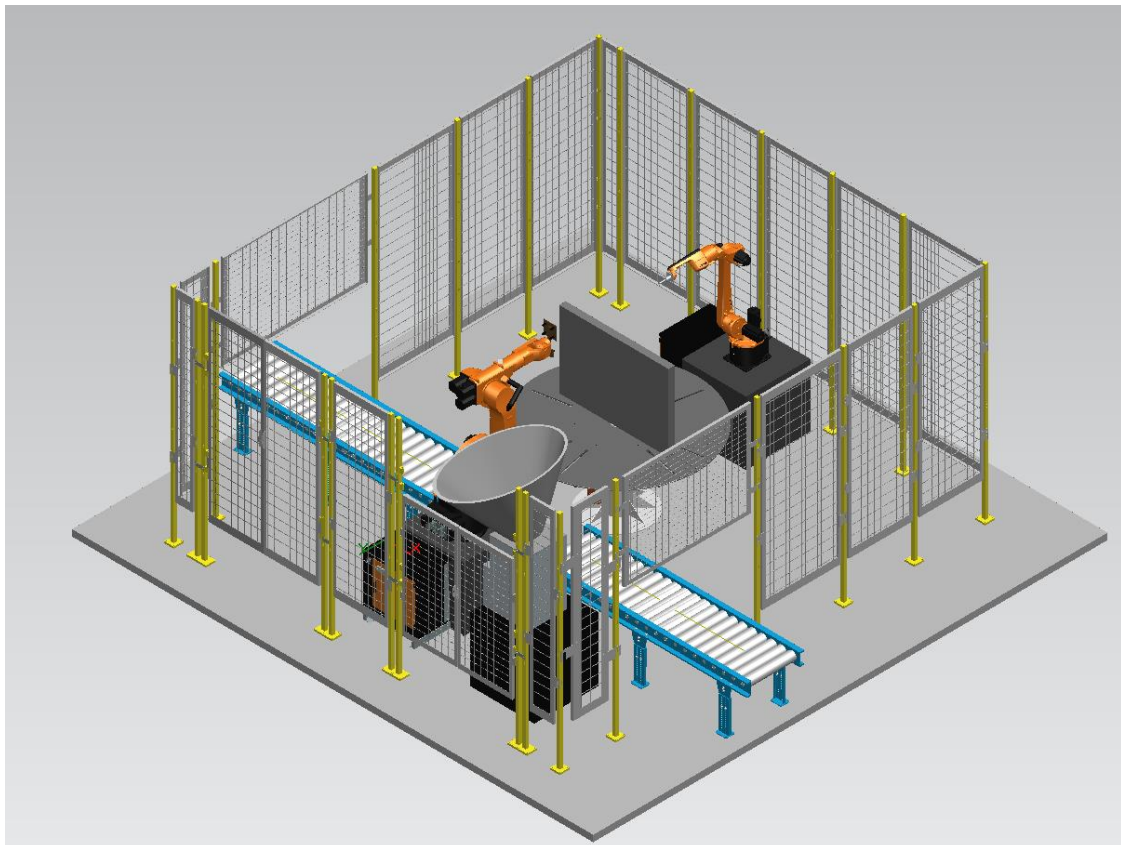


Fig. 16 Layout de la célula

7.3. Selección de los robots

Una vez definidas las características de los robots necesarios para el proyecto, se procede a seleccionar el modelo específico para cada uno. A continuación, se muestra una tabla con las características a tener en cuenta para la selección de un robot.

Tabla 2 Características para selección de robots

Características generales	Área de trabajo Grados de libertad Errores de posicionamiento: distancia tras emergencia, repetitividad, resolución. Errores en el seguimiento de trayectorias (calidad de una línea recta, arco, ...) Precisión cuando se mueve el mínimo incremento posible
Características cinemáticas	Velocidad nominal Aceleración y deceleración
Características dinámicas	Fuerza (de agarre, carga máxima, control de fuerza-par, ...) Frecuencia de resonancia
Tipo movimientos	Movimientos punto a punto Movimientos coordinados Trayectorias continuas
Modo de Programación	Enseñanza (guiado) Textual
Tipo de accionamiento	Eléctrico (c. alterna o c. continua) Neumático Hidráulico
Comunicaciones	E/S Digitales/Analógicas Comunicaciones línea serie
Servicio proveedor	Mantenimiento, Servicio Técnico, Cursos de formación
Coste	

Una vez recogidas las características necesarias para realizar la selección de un robot y, habiendo hecho un estudio del mercado actual, se escogen robots de la marca KUKA. KUKA AG es una empresa presente a nivel internacional siendo uno de los líderes globales en soluciones de automatización inteligentes. KUKA ofrece una gama completa de servicios: desde componentes de robots hasta células, pasando por las instalaciones completamente automatizadas.

Por tanto, se trata de una marca con un gran prestigio en el mercado, en la que la calidad de sus robots queda más que justificada. Además, su transparencia a la hora de ofrecer modelos cinemáticos de sus robots, así como de modelos CAD de sus controladores hace que este haya sido el fabricante escogido.

El modelo escogido para el robot de manipulación es el KR30-3 (figura 17).

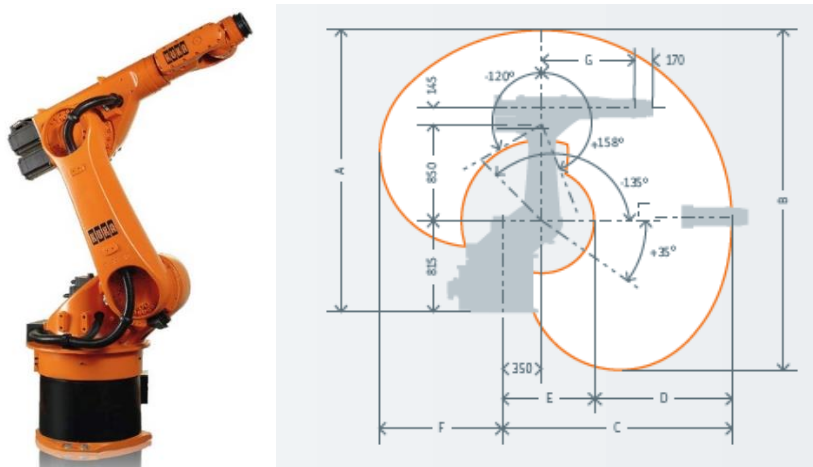


Fig. 17 Robot KUKA KR30-3 y espacio de trabajo

El KR 30-3 aporta la máxima calidad con los tiempos de ciclo más cortos y todo ello con una precisión de repetición máxima. De esta manera, el robot de seis ejes es perfecto para aplicaciones como la manipulación y medición de componentes o aplicaciones láser. Cuenta con una capacidad de carga de 30 kg en un alcance de hasta 2033 mm y una posición de montaje flexible (suelo, techo, pared o posición inclinada). En la tabla 3 se recogen las principales características de robot.

Tabla 3 Características KR30-3

KR30-3	
Alcance máximo	2033 mm
Carga máxima	30 kg
Repetitividad	+0.06 mm
Numero de ejes	6
Peso (sin controladora)	665 kg

Por otra parte, se escoge un robot modelo KR8 R2100 arc HW (figura 18), diseñado para realizar operaciones de soldadura.

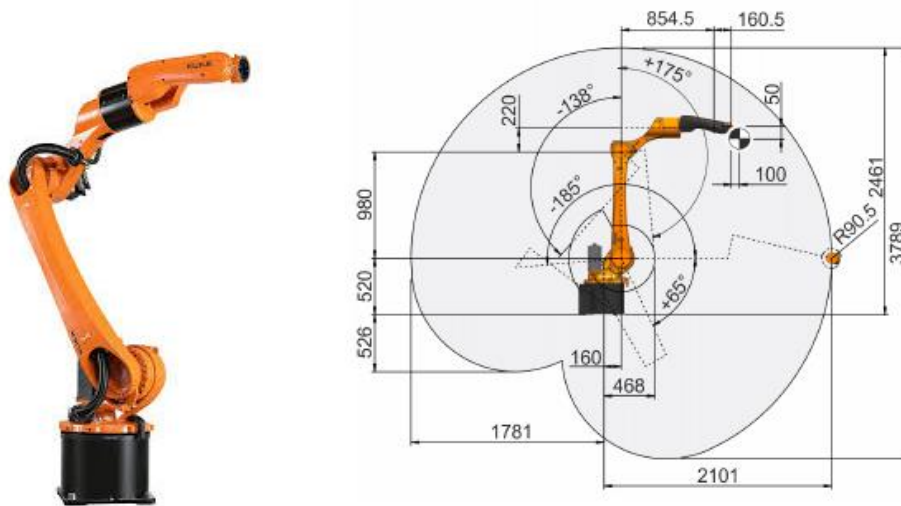


Fig. 18 Robot KUKA KR8 R2100 arc HW y espacio de trabajo

Este modelo, de mínimo mantenimiento, garantiza una correcta funcionalidad en operaciones de soldadura. Los robots industriales de la familia de productos KR CYBERTECH ARC son robots de procesos especializados en aplicaciones de trayectoria, como la soldadura, las aplicaciones de adhesivo y medios de obturación.

Tabla 4 Características KR8 R2100 arc HW

KR8 R2100 arc HW	
Alcance máximo	2101 mm
Carga máxima	-
Repetitividad	+/- 0.04 mm
Numero de ejes	6
Peso (sin controladora)	255 kg

Ambos robots son apropiados para los sistemas de control KR C4 y KR C4 extended. Se escoge la versión KR C4 (figura 19).



Fig. 19 Controlador KUKA KR C4

Además, la KR C4 permite posibilidades de aplicación inteligentes, flexibles y escalables. La opción High-End SoftPLC permite el acceso total a todo el sistema operativo del control y cuenta con un rendimiento elevado a largo plazo. Permite el manejo E/S de un robot o una célula o línea completa de robots mediante red de comunicaciones PROFINET. En la tabla 5 se recogen las principales características de este modelo de controlador.

Tabla 5 Características KR C4

KR C4	
Dimensiones	960x792x558 mm
Procesador	MultiCore-technology
Disco duro	SSD
Numero de ejes máximos a controlar	9
Peso	150 kg

Luego quedan definidos ambos robots con sus respectivas controladoras.

7.4. Diseño de los objetos

El conjunto a montar consta de dos tubos para el agua fría y para el agua caliente y la base del calderín. El diseño de los mismos viene también fijado por el propio proyecto, son especificaciones establecidas. Por tanto, mediante la herramienta de diseño NX se diseñan los diferentes objetos que se muestran en la siguiente figura.

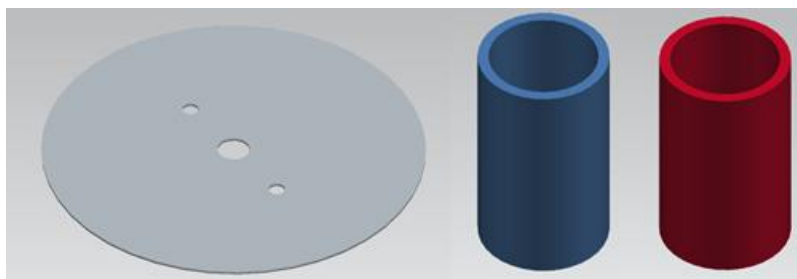


Fig. 20 Base y tubos del calderín

Los tubos del agua se han definido de color azul y rojo para distinguir su funcionalidad. Sin embargo, ambos tienen las mismas características. Una vez diseñados se exportan como un archivo *.jt.

7.5. Diseño del sistema de control

Por último, se realiza un diseño físico del sistema de control de la célula y de la red de comunicaciones de los mismos (figura 21).

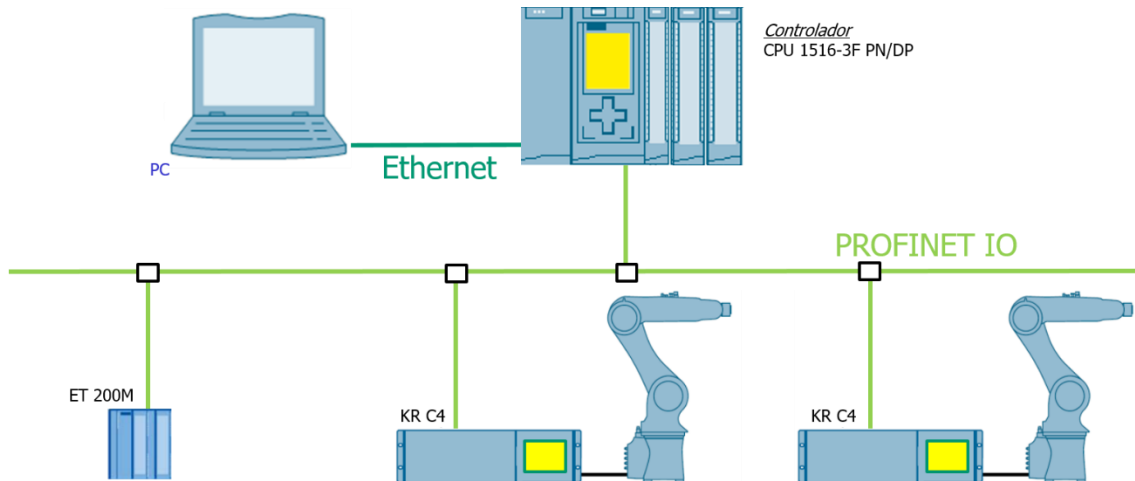


Fig. 21 Red de comunicaciones de la célula

En función de las especificaciones del proyecto se tiene un controlador de la gama S7-1500 para realizar el control y tres dispositivos I/O de periferia de la célula, siendo estos los controladores de la marca KUKA para los robots y una tarjeta ET 200M de Siemens para los sensores de la célula y resto de dispositivos de la misma.

8. METODOLOGÍA. DISEÑO

En este apartado se va a explicar en detalle el desarrollo del proyecto, los medios disponibles para llevar a cabo este y las valoraciones que se han tenido en cuenta.

8.1. Entorno de trabajo

Todo el proyecto se desarrolla en el laboratorio del Departamento de Ingeniería de Sistemas y Automática de la Escuela de Ingeniería de Bilbao.

En dicho laboratorio se dispone de todos los medios necesarios para llevar a cabo el proyecto. A nivel de Hardware se habilita un ordenador, un PLC S7-1500 CPU 1516F-3 PN/DP con 2 tarjetas de entradas y salidas digitales y dos tarjetas de entradas y salidas analógicas, una tarjeta SIMIT UNIT PN 128, que se encarga de simular los diferentes dispositivos IO de periferia de la célula y todo el cableado PROFINET necesario para realizar la instalación.

A nivel de Software, además de contar con Tecnomatix PS V14.1 para realizar el modelado del gemelo digital y TIA Portal V15.1 para realizar el código de control del PLC, también se instala el WinCC Advanced V15.1 para poder diseñar e implementar el HMI, así como una licencia runtime para poder aplicarlo a la célula desde el propio ordenador. También se dispone del software SIMULATION Unit V9.1 upd1 para realizar la carga de los dispositivos I/O de periferia configurados en el proyecto TIA Portal y realizar el enlace de entradas y salidas entre la herramienta de modelado Tecnomatix PS y la tarjeta SIMIT UNIT. En la figura 22 se muestra el demostrador.

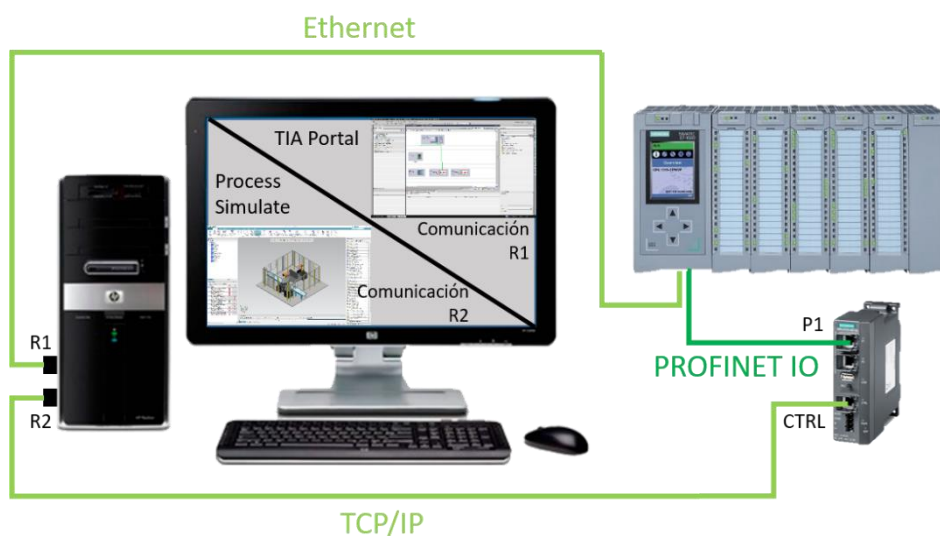


Fig. 22 Demostrador

8.2. Modelado de la célula de soldadura

A continuación, se procede a realizar el modelado del gemelo digital de la célula de soldadura. Para ello, se seguirán los siguientes pasos.

8.2.1. Definición e inserción de los componentes

El primer paso consiste en la “definición del tipo de componente” e “inserción” de los diferentes elementos que componen la célula. Ya que cada elemento tiene una función específica, Tecnomatix PS ofrece la capacidad de definir qué tipo de componente se trata para poder dotarle de capacidad de realizar unas funciones u otras. Para poder ser definidos e implementados en Tecnomatix PS, el modelo 3D de cada elemento, en formato *.jt, debe estar dentro de una carpeta con nombre (nombre_carpeta.cojt). Y todas estas carpetas deben encontrarse dentro del directorio principal establecido al principio del proyecto. Este suele ser por defecto C:\SYSROOT. Este requisito es necesario para que Tecnomatix PS pueda ver los diferentes elementos desde su entorno. A modo de ejemplo se muestra en la figura 23 el directorio del proyecto. En particular se muestra la carpeta que contiene los diferentes elementos de la célula, almacenados individualmente en su correspondiente carpeta de extensión *.cojt.

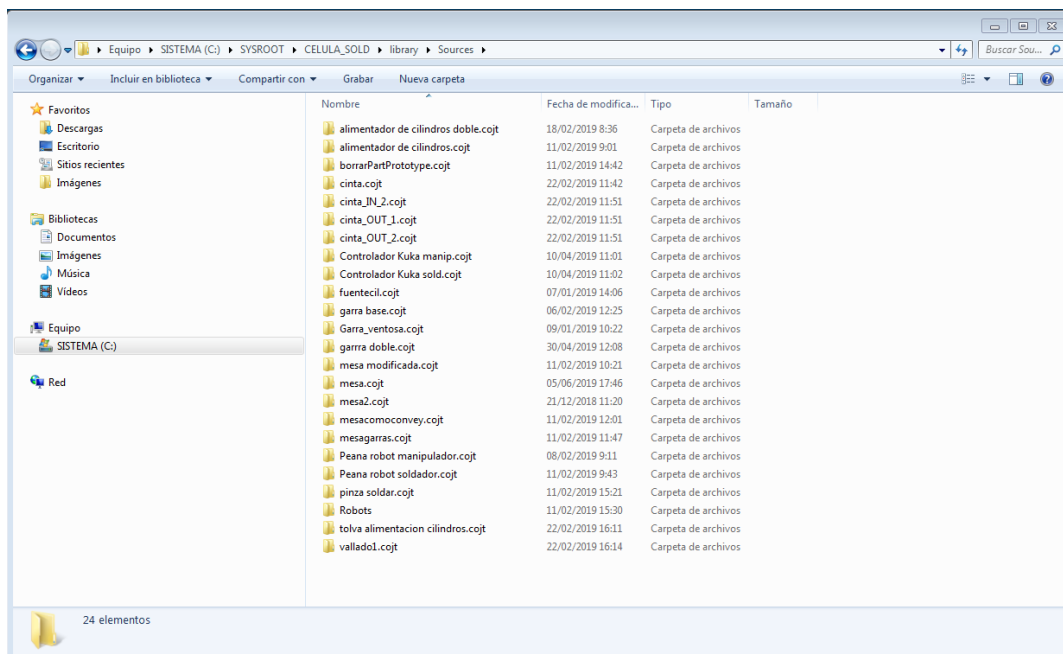


Fig. 23 Directorio del proyecto

A la hora de realizar la definición del componente, Tecnomatix PS ofrece la posibilidad de escoger entre 23 tipos de elementos tales como tipo garra, robot, humano, prototipo, entre otros. Se definen a ambos robots como tipo de componente “robot”, las cintas

transportadoras como "Conveyor", la mesa de trabajo como "Work_table" y el dispensador de tubos como "Equipment Prototype". La definición de un tipo u otro dota a ese elemento de unas características u otras. A continuación, se describen únicamente los tipos de elemento que se implementan en la célula.

Los robots cuentan con una amplia configuración de características. Se les asigna el sistema de coordenadas de la base y del punto extremo del robot o TCP. Se les asigna un controlador, se generan las señales típicas de robots industriales, las cuales se usan para realizar la comunicación con el controlador de la célula, y se emplea todo un apartado para configurar las diferentes operaciones que los robots realizarán durante la tarea de ensamblado del conjunto. También se les asocia la herramienta que les corresponde a cada uno de los dos. Además, Tecnomatix PS permite realizar análisis de colisiones, obtener posiciones específicas de la herramienta del robot de manera muy intuitiva y analizar posibles bloqueos en las articulaciones del robot entre otras, permitiendo un control total sobre la cinemática y control de los robots.

Por otro lado, a las cintas transportadoras se les dota de capacidad de mover objetos que se seleccionen como propios de las mismas (Define Conveyable Part), teniendo un mayor control sobre los objetos a la hora de realizar el diseño de la célula. Se puede configurar la trayectoria que seguirán los diferentes objetos por las mismas. La mesa de trabajo y el dispensador de tubos disponen de operaciones básicas para cualquier elemento de una célula de soldadura.

Finalmente, se importan las diferentes piezas que forman el conjunto. Estas se importan en el proyecto de Tecnomatix PS definiéndolas como piezas o "parts".

Una vez definidos se insertan en el entorno y se sitúan en la situación establecida como se indica en la figura 16.

8.2.2. Cinemática de los elementos

La cinemática de un elemento es la parte de la mecánica que define los movimientos del mismo. En un análisis cinemático la posición, velocidad y aceleración de cada uno de los elementos son calculados sin considerar las fuerzas que causan el movimiento. A su vez Tecnomatix PS es una herramienta de ingeniería que realiza el estudio de células o líneas de manipulación únicamente desde un punto de vista cinemático, sin tener en cuenta fuerzas ni esfuerzos de los diferentes elementos, es decir, sin que los elementos tengan las físicas propias del sistema real. Es por esto que se tiene que dotar de una

cierta cinemática a todos los elementos para que, a la hora de realizar la simulación, el gemelo virtual se comporte de igual modo que el sistema real.

Por otra parte, Tecnomatix PS permite la creación de modelos 3D básicos, únicamente para realizar modelos simplificados de un posible elemento real o para el diseño de piezas básicas. Permite realizar las operaciones básicas de extrusión, revolución y creación de figuras básicas, así como el escalado de objetos ya existentes, lo que permite adaptar ciertos elementos a las medidas de la célula diseñada. Un claro ejemplo es el vallado de seguridad, el cual se trata de un modelo generado para otra aplicación y el cual se escala a las necesidades de este. A su vez, Tecnomatix PS permite realizar la configuración de la cinemática de los elementos insertados en el modelo o los diseñados en el propio entorno.

Cabe destacar que los modelos de los robots descargados desde el centro de soporte técnico de KUKA son archivos de extensión *.jt, el cual ya contiene el modelo cinemático del robot. A continuación, se muestra a modo de ejemplo la cinemática del robot manipulador seleccionado para este proyecto.

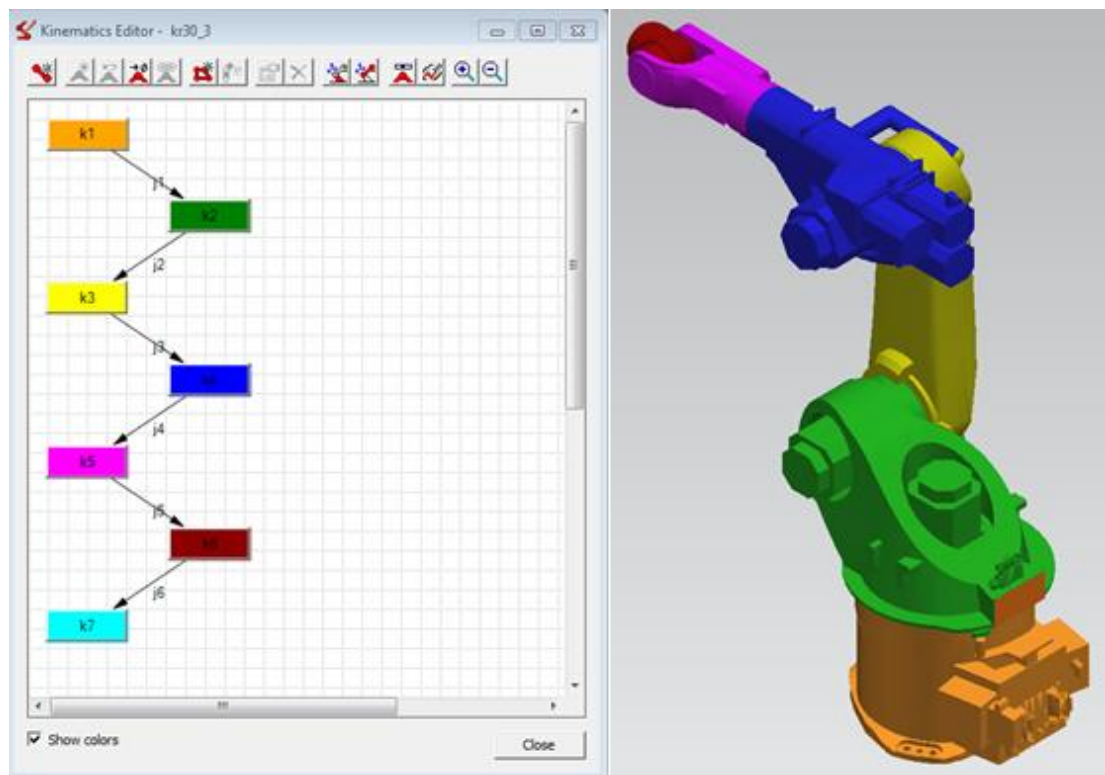


Fig. 24 Cinemática del robot manipulador

En la figura 24 se aprecia como cada elemento del robot (fijo o móvil) queda representado por un color diferente en el gráfico. Estos quedan unidos entre sí mediante el movimiento relativo que tienen unos respecto a otros. Así, la base fija del robot k1 (color naranja) es el primer elemento de esta cadena cinemática y la cual no tiene movimiento relativo respecto de ningún elemento anterior. Este se une al siguiente elemento del robot k2 (color verde), definiendo esta unión un movimiento relativo j_1 respecto el anterior. Este movimiento será de rotación alrededor de la articulación común a ambos realizado por el elemento k2 sobre el elemento k1. El elemento k2 se une al siguiente elemento k3 (color amarillo) mediante j_2 , representando este el giro de k3 alrededor de su eje común respecto k2. Esta cadena continua hasta finalizar en el elemento k7, elemento donde se sitúa la herramienta y, por tanto, donde se encuentra el TCP (Tool Center Point) del robot. Se trata del último punto del robot, siendo este un punto no físico y, por tanto, no observable mediante el color azul claro sobre el robot.

Sin embargo, se debe dotar de una cinemática al resto de componentes para poder simular los diferentes movimientos que todos los elementos realizan en la célula real. Para ello se debe abrir el editor cinemático de un elemento e ir seleccionando los diferentes elementos y unirlos mediante el movimiento relativo de un elemento respecto a otro como se muestra a continuación.

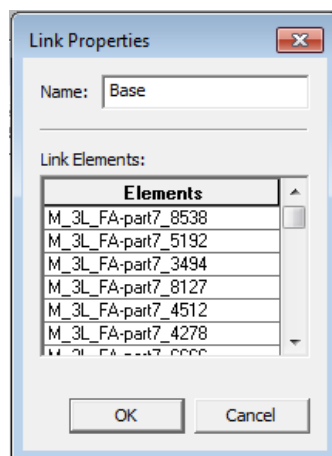


Fig. 25 Conjunto de elementos "Base"

En esta ventana de configuración se seleccionan todos los elementos del modelo mecánico que forman parte de elemento que definimos como "Base", es decir, los elementos fijos de la mesa de trabajo, como el soporte de la misma, el motor que acciona el giro de la mesa, pernos de sujeción, entre otros, creando lo que se denomina "link". Del mismo modo, se creará el "link" con los elementos de la mesa de trabajo que son móviles, como pueden ser la superficie de trabajo, la mampara de separación,

rodamientos, etc. A continuación, se muestra la cinemática implementada a la mesa giratoria de la célula.

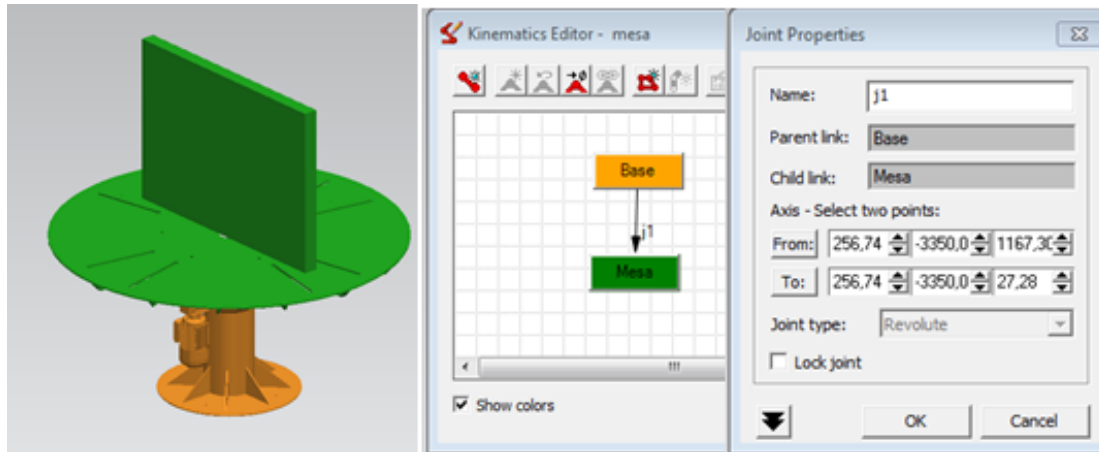


Fig. 26 Cinemática de la mesa giratoria

El modelo de esta mesa es adquirido de una página de modelos CAD (ver referencias). Se configura de tal modo que los elementos definidos como mesa (elementos verdes) tengan un movimiento relativo sobre la base (elementos en naranja) alrededor del eje central de revolución. Por tanto, se crean dos elementos denominados "links", uno con los elementos fijos de la mesa (Base) y otros con los elementos móviles (Mesa) como se ha explicado anteriormente. A continuación, se unen mediante un enlace (j1). Este es el que define el movimiento relativo que habrá entre ambos elementos. El enlace es una flecha que apunta al conjunto de elementos "link" que va a realizar el movimiento que se configure ("Child link") respecto del "link" del que sale la flecha denominado "Parent link". Se puede configurar movimientos de revolución y translación de manera muy intuitiva, con mas que definir el eje de giro o de desplazamiento por dos puntos. Además, con ayuda del "Pose Editor" se configura las dos posiciones posibles de la mesa, ya que sólo puede realizar giros de 180°, por lo que se configuran las posiciones "Recibir base", posición natural de la mesa, y la posición "Soldadura", la cual se corresponde con un giro de 180° anti horario, quedando definidas las dos únicas posiciones de trabajo de la mesa.

También se configura la cinemática de la puerta del vallado de seguridad (figura 27).

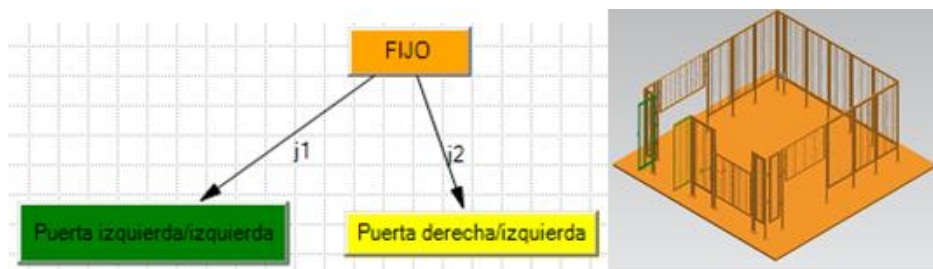


Fig. 27 Cinemática del vallado de seguridad

Esta se configura de tal modo que se asocia un "link" para la cada una de las dos puertas y otro para el resto de elementos fijos del vallado. Una vez creados se crean dos relaciones de movimiento relativo j_1 y j_2 . Cada una de las dos será un movimiento de revolución alrededor del eje de giro de cada puerta.

Finalmente, al alimentador de tubos y a las cintas de transporte de entrada de bases y salida de conjuntos no se les dota de cinemática.

Por un lado, el alimentador de tubos como ya se ha comentado se trata de un modelo simplificado, por lo que no se conoce información acerca de su cinemática real. Es por esto por lo que se dotan a los tubos de una operación de movimiento lineal, simulando la nueva aparición de estos en la recepción de tubos de igual manera que lo haría un alimentador real.

Por otro lado, respecto a las cintas de transporte, se asociará un movimiento rectilíneo a las bases nuevas que entran a la célula de soldadura y a los conjuntos soldados que abandonan la misma. Con ello se consigue simular la acción que realizarían las cintas sobre estos elementos. Sin embargo, como ya se ha mencionado anteriormente a las cintas transportadoras se les dota de tipo de objeto "conveyor". Esto permite realizar un manejo de la posición de los elementos que se desplazan por las cintas, permitiendo una mayor configuración de las operaciones de los robots.

Llegados a este punto los elementos quedan ubicados en la posición deseada y con su cinemática configurada para realizar los movimientos necesarios.

8.2.3. Diseño de las herramientas de los robots

Establecidos los robots con su cinemática, se procede a realizar el diseño de las herramientas de los mismos. El diseño de una herramienta de robot real conlleva un proceso de diseño que queda fuera del alcance de este proyecto. A su vez, se encuentran

dificultades para encontrar modelos 3D de las mismas. Es por esto que se añade este punto ya que, para poder realizar las correspondientes simulaciones, se realiza un diseño conceptual de una garra de manipulación y una boquilla de soldadura oxiacetilénica y se les dota de la cinemática correspondiente. Para ello, dentro de la propia herramienta de ingeniería Tecnomatix PS se dispone del apartado "Modeling", desde el cual además de poder definir la cinemática de los elementos, entre otras características, se encuentra la capacidad de definir geometrías. Luego, con dicha aplicación se diseña la garra del robot manipulador, de la cual se detallará luego su cinemática. La boquilla del robot soldador se trata de un diseño conceptual, la cual sólo describirá la trayectoria deseada. Por tanto, no es necesario dotarla de cinemática y se diseña mediante la herramienta Solid Edge ST9 como un único sólido. A continuación, se muestran ambas herramientas.

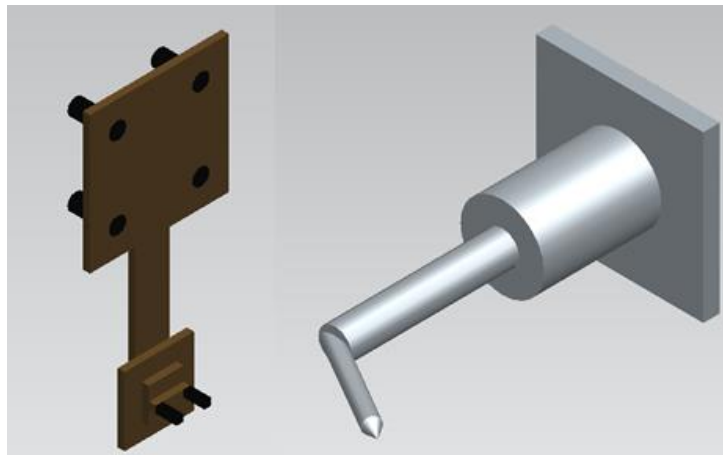


Fig. 28 Herramienta doble de manipulación (izquierda) y boquilla de soldadura (derecha)

Se aprecia como la herramienta del robot manipulador (izquierda) consta de dos partes. Por un lado, el diseño conceptual de una herramienta neumática (parte superior de la garra), la cual consta de 4 ventosas neumáticas que se mueven linealmente alrededor de su eje central para la colocación de los diferentes elementos. Se encargan de llevar las bases que llegan por la cinta y colocarlas en la mesa y también de retirar el conjunto una vez soldado y depositarlo en la cinta de salida. Por otro lado, esta herramienta consta de una pinza (parte inferior de la garra). Esta se diseña como dos prismas que tienen un movimiento rectilíneo transversal. Se encarga de coger los tubos del agua fría y caliente del dispensador de cilindros y los coloca en la posición de soldado.

Esta herramienta sí que tiene movimiento relativo entre elementos, por lo que se ha de dotar de cinemática. En la figura 29 se muestra la cinemática de dicha herramienta.

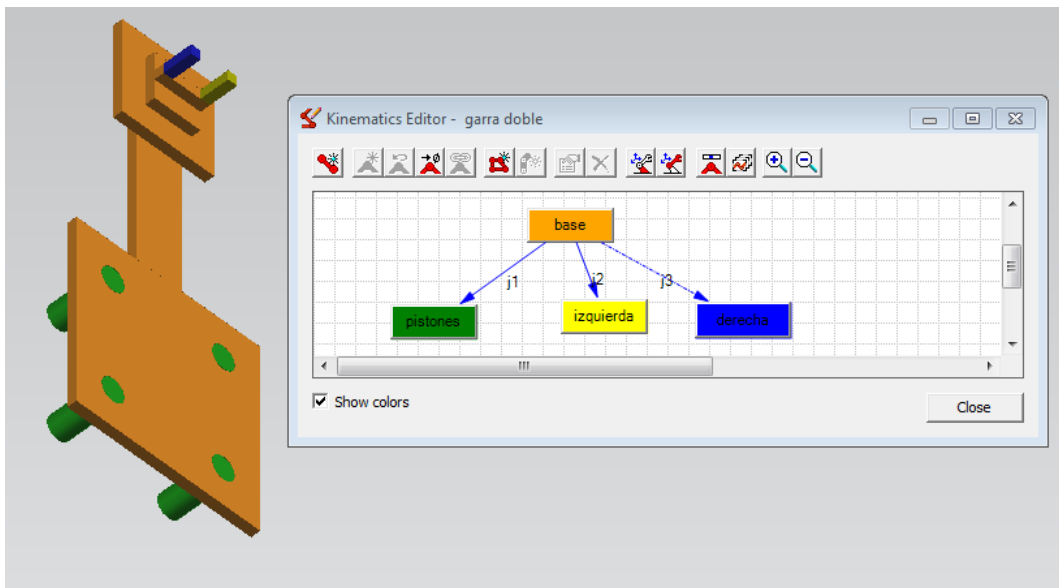


Fig. 29 Cinemática de la garra de manipulación

En la imagen se aprecia como la garra consta de un “link” denominado base (color naranja), que comprende los elementos fijos de la herramienta, y los elementos de ambos manipuladores explicados anteriormente. La parte correspondiente a las ventosas neumáticas (cilindros verdes) se dota de un movimiento relativo prismático, moviéndose hacia fuera en el momento de la recepción de la base (figura 30 derecha), momento en el cual se aplica el vacío neumático; y volviendo a la posición inicial (figura 30 izquierda) en el momento de colocación de la pieza. El movimiento de los cuatro cilindros tiene que ser simultáneo y por eso se configuran los 4 cilindros neumáticos como un solo “link”.

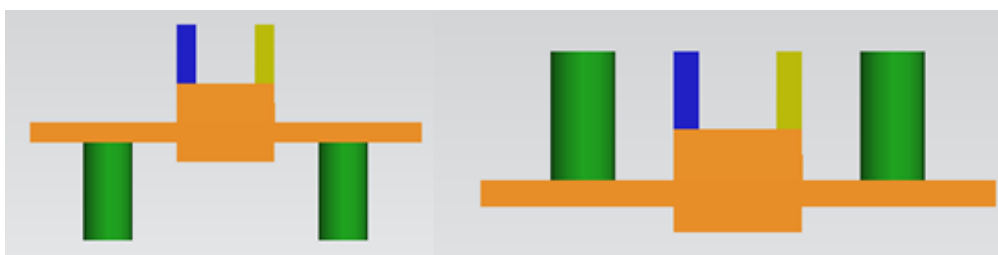


Fig. 30 Movimiento cinemático garra para bases

A su vez, la parte de la pinza se aprecia como consta de un movimiento para cada parte de la garra. Se define el movimiento prismático de una de las partes de la pinza (figura 31 prisma amarillo) y se configura el movimiento inverso a la otra parte de la pinza (figura 31 prisma azul). Así se consigue el efecto de apertura y cierre de la garra. Se define también el movimiento límite de los prismas, evitando así la colisión de los mismo o la separación de la parte fija de la garra doble.

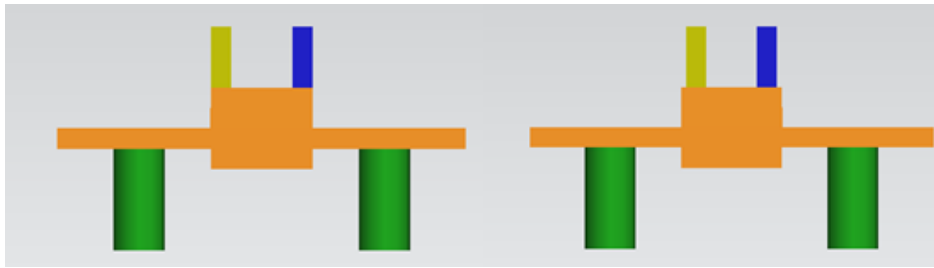


Fig. 31 Movimiento relativo garra para tubos

Finalmente se definen las posibles posiciones de operación de la herramienta. Tenemos por un lado la posición "HOME" (figura 30 o figura 31 izquierda). Esta posición es la posición de reposo de la herramienta. Mientras el robot manipulador esté a la espera de recibir una nueva orden de operación o apagado está será su posición. Se define también la posición "Funcionando". Esta posición se corresponde con la parte de las ventosas neumáticas en el momento en el que estén manipulando una pieza (figura 30 derecha). Luego, se encontrará en dicha posición cuando este robot esté colocando una nueva base en la mesa de trabajo o cuando este esté retirando un conjunto ya soldado. Finalmente se define la posición "garra cerrada". Como su nombre indica esta es la posición de la garra para coger tubos en posición cerrada (figura 31 derecha). La garra se encontrará en dicha posición cuando el robot esté cogiendo un tubo nuevo del alimentador y colocándolo en la posición correspondiente sobre la mesa de trabajo.

Por tanto, quedan ambas herramientas definidas con su cinemática y las posiciones en las que operarán para ser montadas sobre el correspondiente robot. Para ello, se hace uso de la opción "Mount Tool". Se escoge la herramienta y el robot al que se quiere asociar dicha herramienta. A su vez, se escogen los sistemas de coordenadas de ambos elementos que se relacionarán entre sí, es decir, se escoge el TCP del robot y el sistema de coordenadas central de la herramienta de tal modo que Tecnomatix PS hace coincidir sus orígenes de coordenadas y alinea sus diferentes ejes.

8.2.4. Operaciones de los elementos de la célula

Una vez establecidos los elementos y definida su cinemática se dispone a configurar las distintas operaciones que tienen que realizar. A continuación, se muestran las diferentes operaciones que pueden implementarse a los diferentes objetos, ya sean componentes de la célula o las piezas a montar.

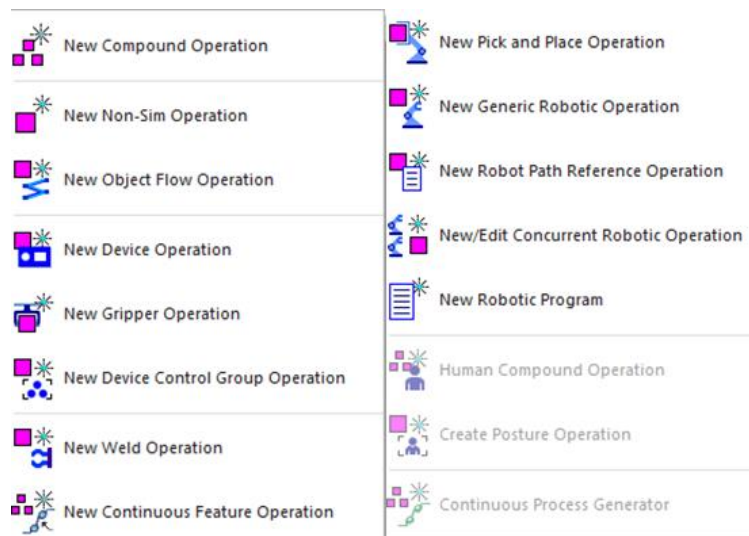


Fig. 32 Tipos de operaciones para robots

En la figura se aprecia como existen “grupos de operaciones”, “operaciones a objetos”, “operaciones de pinzas”, “operaciones de robots”, etc.

Las primeras operaciones implementadas en la célula son las operaciones sobre las nuevas piezas de los robots, denominadas “Object Flow Operation”. Se les dota de un nombre, se selecciona el objeto que realiza el movimiento, el punto inicial y final del movimiento y la duración del mismo (ver figura 33). De este modo se simula las acciones que realiza la cinta transportadora y el dispensador de tubos como se ha comentado anteriormente.

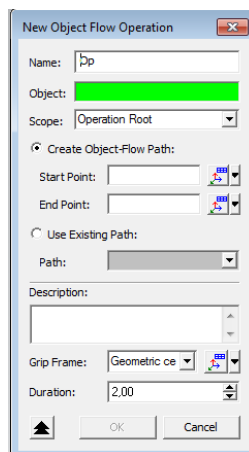


Fig. 33 Panel configuración de operación de movimiento de un objeto

A continuación, se pasa a definir las diferentes operaciones de los elementos de la célula. La mesa de trabajo realiza únicamente giros de 180°, conmutando entre un giro antihorario y un giro horario en cada secuencia. Este tipo de operación se denomina

“New Device Operation” (figura 34), ya que se trata de la operación de un dispositivo de la célula.

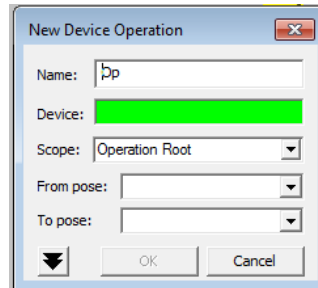


Fig. 34 Nueva Operación de un elemento

La diferencia de este tipo de operación respecto a la anterior es que esta no realiza un movimiento del elemento como tal, sino una variación en la posición del mismo. Por tanto, se dota de un nombre a la operación, se selecciona el elemento y se elige de que posición parte el movimiento y cuál será la posición final. Las posiciones deben ser previamente establecidas con ayuda del “Pose Editor” como se ha explicado en el apartado de cinemática de los elementos. De igual manera se definen las operaciones de apertura y cierre de la puerta del vallado de seguridad.

El siguiente tipo de operación configurada son las operaciones de los robots. El robot manipulador realiza operaciones de Pick & Place. En la figura 35 se muestra el panel de configuración de una operación de pick & place, operación típica en células robotizadas.

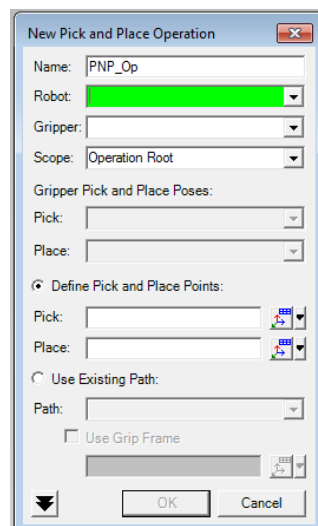
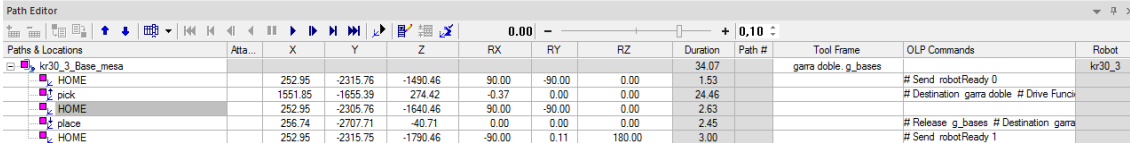


Fig. 35 Panel de configuración de operación de Pick & Place

Se aprecia cómo se ha de escoger el robot, la garra correspondiente, la posición de la garra en los momentos de coger y dejar el objeto y los puntos donde se realizan estas operaciones. Una vez configurada la operación el robot ya está en disposición de realizar la tarea de Pick & Place, sin más que configurar en las propiedades de la garra que elementos son los que va a coger. Sin embargo, aunque el robot realiza la operación establecida esta primera configuración no tiene en cuenta la posición del resto de elementos y, por tanto, no tiene en cuenta las posibles colisiones que se dan con el resto de elementos de la célula. Por tanto, se configuran las posiciones del robot con el "Path Editor" (figura 36).



Paths & Locations	Atta...	X	Y	Z	RX	RY	RZ	Duration	Path #	Tool Frame	OLP Commands	Robot
kr30_3_Base_mesa								34.07		gara_doble_g_bases		kr30_3
HOME		252.95	-2315.76	-1490.46	90.00	-90.00	0.00	1.53			# Send robotReady 0	
pick		1581.85	-1655.39	274.42	-0.37	0.00	0.00	24.46			# Destination gara_doble # Drive Funci	
HOME		252.95	-2305.76	-1640.46	90.00	-90.00	0.00	2.63				
place		256.74	-2707.71	-40.71	0.00	0.00	0.00	2.45			# Release g_bases # Destination gara	
HOME		252.95	-2315.75	-1790.46	-90.00	0.11	180.00	3.00			# Send robotReady 1	

Fig. 36 Path Editor

Este es un editor de Tecnomatix PS, el cual permite configurar de forma precisa las operaciones que se implementan en la célula, así como una primera simulación de las mismas. Por tanto, se configuran al principio y al final de la operación dos posiciones, referentes a la posición "HOME" del robot, para que este vuelva a la posición de reposo una vez finalizada la operación y no se quede en el lugar donde ha dejado la pieza. Además, se configuran las posiciones de recogida y dejada de los elementos y se insertan nuevas posiciones auxiliares si fuera necesario o se realiza la configuración de velocidades del robot, entre otras características.

Siguiendo con las operaciones de los robots, para la realización de la soldadura de los tubos a la base se requiere de operaciones de proceso continuo. Para ello se define una nueva operación denominada "Continuous Process Generator". El editor de operaciones es el mostrado en la figura 37.

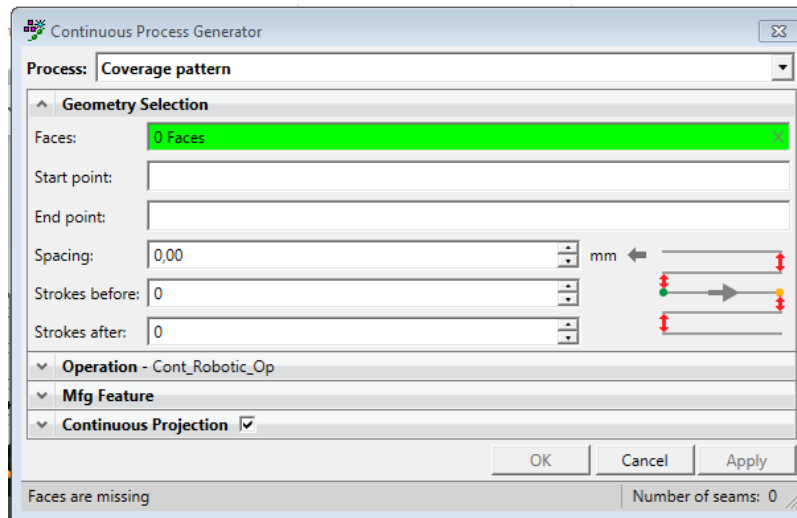


Fig. 37 Editor de operaciones de proceso continuo

Se puede elegir como proceso “arc” o “Coverage pattern”. Se escoge este último ya que la soldadura a realizar tiene un patrón circular. En el apartado “Face Sets” se marcan las dos caras involucradas en la soldadura, así como el punto inicial y final de la soldadura. En el apartado “Operation” se selecciona el robot que va a realizar la operación y la correspondiente herramienta. En “Mfg Feature” se elige el tipo de operación continua que se va a realizar. Se tiene desde soldadura por arco, operaciones de lavado, pintado, pegado o simplemente se puede definir como operación continua. Finalmente, en el apartado “Locations Distribution” se configura el tipo de orientación tangente o en zigzag de la soldadura, tolerancias dimensionales y otros parámetros, quedando la operación de soldadura continua perfectamente definida. Este editor de operaciones tampoco tiene en cuenta posibles colisiones o problemas en las articulaciones del robot, por lo que para configurar las posibles posiciones incompatibles se hace uso del “Path Editor” del mismo modo que en la operación de Pick & Place,

Llegados a este punto se tienen todas las operaciones que se realizan durante la simulación del gemelo digital. En la figura 38 se muestra el árbol de operaciones con todas las operaciones definidas a ambos robots y diferentes elementos de la célula.

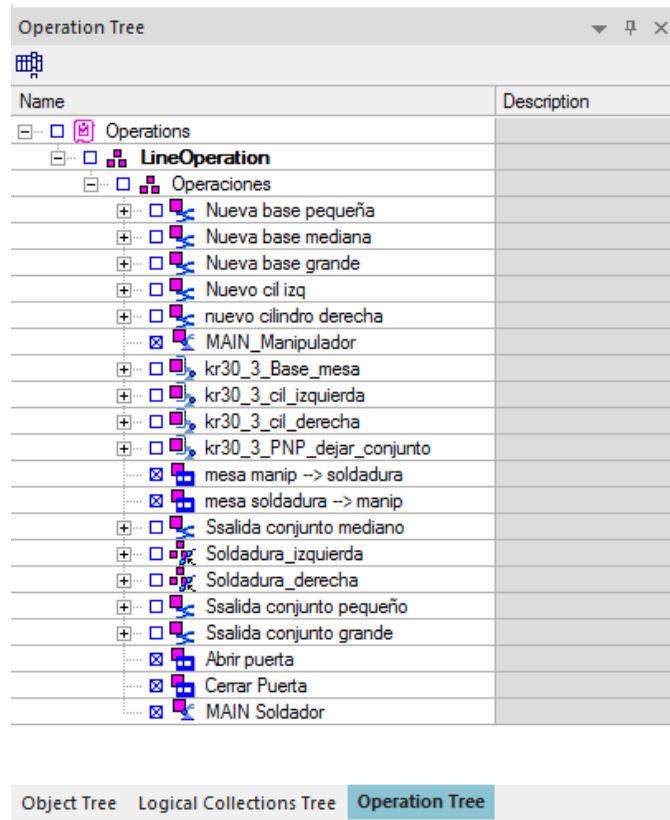


Fig. 38 Árbol de operaciones de la célula

Se definen todas las operaciones de movimiento de las diferentes piezas, diferenciando entre bases pequeñas, medianas y grandes y tubos izquierdos y derechos. A su vez, todas las operaciones del robot manipulador vienen precedidas del nombre del robot (kr30_3). El robot soldador por el contrario únicamente cuenta con las operaciones soldadura izquierda y derecha, por lo que se prescinde de su nombre. Se cuenta también con dos programas principales "MAIN", uno por cada robot, necesarios para realizar el programa de los robots. La programación de los mismos se muestra en el apartado siguiente. Finalmente, se dispone de las operaciones de movimiento del resto de elementos de la célula como son el giro de la mesa de trabajo y la apertura y cierre de la puerta del vallado de seguridad.

8.2.5. Programas de los robots

Una vez creadas todas las operaciones se tiene que diseñar el código de control de cada robot. Desde el panel "Robotic Program Inventory" de cada robot se crea un programa principal y se asocia al robot correspondiente como programa por defecto. Este se abre en el "Path Editor" y es ahí donde se incluyen las operaciones asociadas a cada robot y se crea una operación "New Generic Robotic Operation" por cada robot donde se codifica el programa principal que realiza las correspondientes llamadas a cada operación en el momento que se necesite. El resultado es el mostrado en la figura 39.

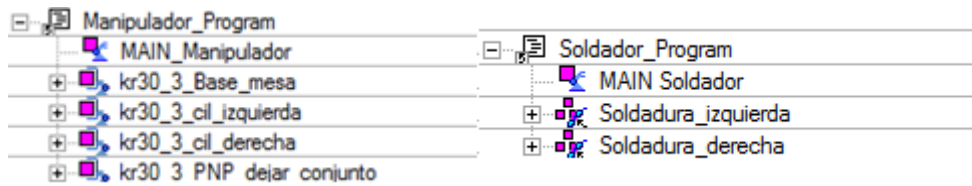


Fig. 39 Conjunto de operaciones del robot manipulador (izquierda) y soldador (derecha)

Una vez asociadas y creada la operación MAIN, se programa esta última en cada robot como un bucle en el que mientras el robot esté funcionando (señal Start_program correspondiente a cada robot) se ejecute un bucle que esté a la espera de la señal del controlador PLC, el cual mandará el número correspondiente a cada operación. Esta programación se realiza usando la pestaña "OLP Commands" del "Path Editor". Esta ventana (figura 40) permite la programación de código de robot en un lenguaje genérico, el cual es adaptado a cada fabricante si se dispone de la Unidad de Control Virtual correspondiente. En esta ventana es donde se configura el menú principal en bucle que ejecutará cíclicamente cada controlador de robot a la espera de recibir el número del programa a ejecutar por parte del controlador I/O.

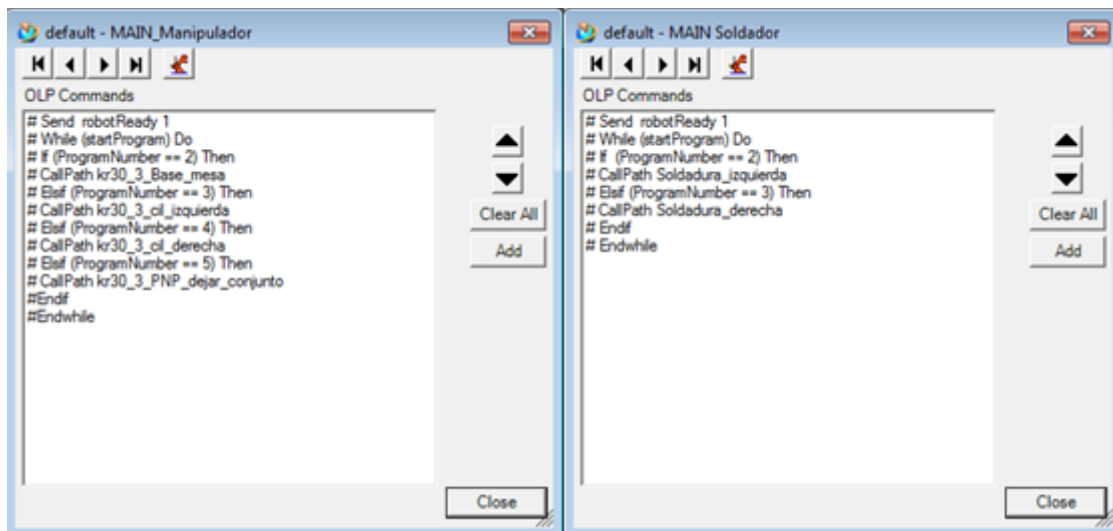


Fig. 40 Ventana de configuración del MAIN del robot manipulador (izquierda) y soldador (derecha)

Ambos bucles de los robots han sido programados a partir de diagrama de flujo diseñado para cada uno. En la figura 41 se muestra el diagrama de flujo de ambos programas.



Fig. 41 Diagramas de flujo de los programas de los robots

Se aprecia como en ambos casos, el programa se ejecuta cíclicamente mientras la variable "Start_Program" de cada robot esté activa. En el momento en el que el PLC envíe un número, el controlador del robot ejecutará la operación correspondiente.

Finalmente, terminada la codificación, los programas están listos para ser descargados. Tecnomatix PS tiene un controlador por defecto, el cual genera un código genérico de robots. Si se dispone del controlador del fabricante, conocido como Unidad de Control Virtual, este puede configurarse en los parámetros del robot y generar el fichero en el lenguaje del controlador real. Así, si las medidas de la célula y los objetos han sido respetadas, el programa puede ser implementado en la controladora del robot real sin más que realizando la descarga del programa del control desde la herramienta de ingeniería Tecnomatix PS y realizando la correspondiente carga en el controlador real del robot.

8.2.6. Sensorización

Una parte indispensable en cualquier célula robotizada es la sensorización de la misma. El saber en todo momento donde se encuentran los objetos, el estado de robot o cualquier servicio de seguridad hace que los sensores sean dispositivos indispensables. Los dos tipos de sensor principales dentro de Tecnomatix PS son el de proximidad y el fotoeléctrico. En este proyecto se usan exclusivamente los últimos, los cuales se parametrizan a través de la siguiente ventana.

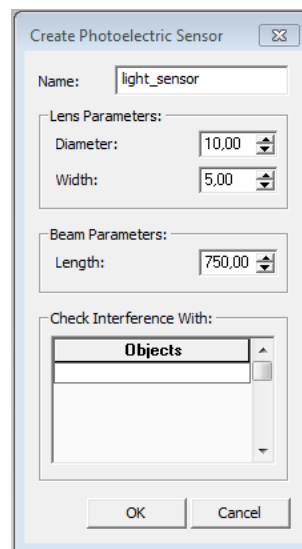


Fig. 42 Panel de configuración de un sensor fotoeléctrico

Se define el nombre del sensor, el diámetro y longitud del haz del sensor y los objetos que, atravesando el haz de luz, tiene que detectar. El sensor queda definido y únicamente es necesaria su colocación en el lugar de la célula deseado. A continuación, se muestran una imagen con la posición de los sensores en la célula de soldadura, junto a una tabla con una breve descripción.

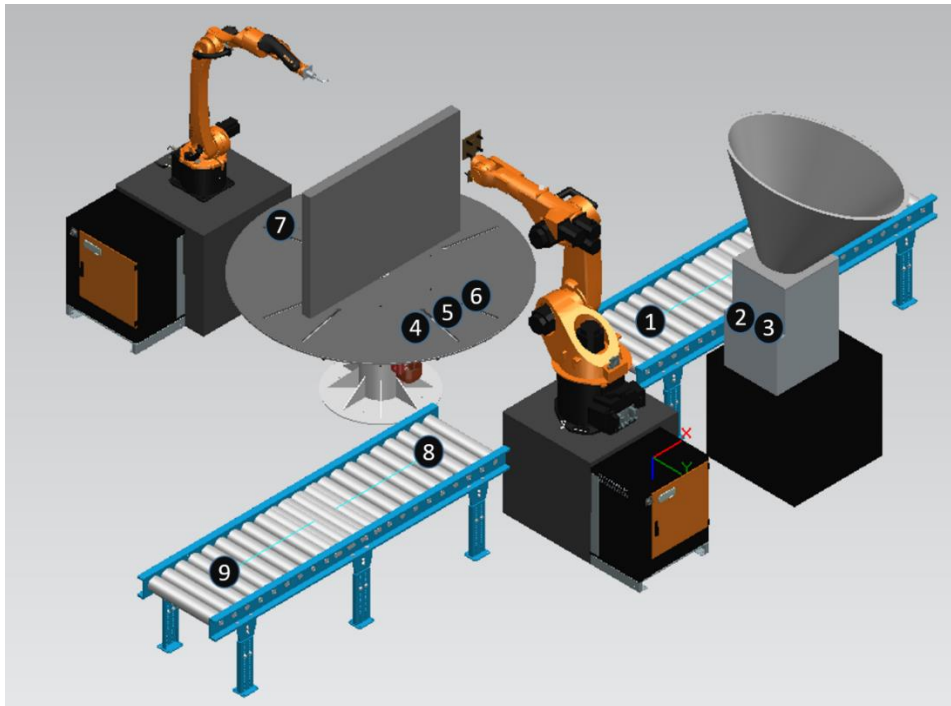


Fig. 43 Posición de los sensores dentro de la célula de soldadura

Tabla 6 Sensores de la célula de soldadura

Numero	Descripción
1	Sensor presencia nueva base en la cinta de recepción para ser colocada en la mesa de trabajo
2	Sensor que detecta un nuevo tubo izquierdo en la recepción del dispensador para ser colocado en la mesa de trabajo
3	Sensor que detecta un nuevo tubo derecho en la recepción del dispensador para ser colocado en la mesa de trabajo
4	Sensor presencia tubo izquierdo en la mesa de trabajo
5	Sensor presencia base en la mesa de trabajo
6	Sensor presencia tubo derecho en la mesa de trabajo
7	Sensor presencia conjunto en soldadura
8	Sensor presencia conjunto soldado en la cinta de salida
9	Sensor presencia conjunto soldado final cinta de salida. Este sensor se implementa por necesidad en el programa de control

De este modo se consigue la completa información del estado de la célula en todo momento.

8.2.7. Flujo de materiales

Por último, se ha de definir el flujo o camino que siguen los materiales durante todo el proceso de soldadura. Los objetos o piezas sobre los que se realizan las operaciones de manipulación o soldadura se definen en el software Tecnomatix PS como "parts". Estas se diseñan o importan en el modo "Standard Mode". Sin embargo, a la hora de realizar simulaciones del modelo se tiene que activar el modo "Line Simulation Mode". Es en este

modo donde aparece la ventana "Material Flow Viewer", en la cual se define el flujo o camino que siguen los diferentes elementos, es decir, los movimientos u operaciones sobre los mismos. Este es un requisito indispensable para que, a la hora de realizar la simulación, los objetos aparezcan durante la misma.

El Visor de flujo de materiales muestra operaciones que utilizan elementos (bases y tubos en este caso) y enlaces de flujo de materiales entre las operaciones. El enlace de operación 1 con la operación 2, por ejemplo, indica que las partes utilizadas por operación 1 están disponibles para la operación 2. Incluso si una parte utilizada por operación 1 se pasa a través de un enlace a la operación 2 pero no es usada por operación 2, todavía pasa por el siguiente enlace a la operación 3 y queda disponible para ella. Esto le permite ver y controlar la forma en que se pasan las partes entre las operaciones en el modo de simulación basada en eventos.

El Visor de flujo de material muestra la estructura de flujo de material de todo el estudio, no tiene un alcance específico. Siempre muestra todas las operaciones en el estudio que tienen enlaces de flujo de material entrantes / salientes.

Inicialmente, el sistema evalúa las condiciones de inicio de las operaciones. Cuando se ejecutan las operaciones, utiliza las piezas disponibles de las operaciones anteriores (de acuerdo con los enlaces de flujo de material) incluso si todavía faltan algunas de las piezas necesarias y comienza la operación. En los siguientes puntos críticos de la operación, el sistema verifica si las operaciones requeridas restantes han sido suministradas por las operaciones anteriores (la operación continúa incluso si todavía faltan las partes).

La ejecución de las operaciones está controlada por el "Editor de secuencias", pero el flujo de las partes es independiente del orden de las operaciones. En la figura 44 se muestra el Visor de flujo de materiales de la célula de soldadura.

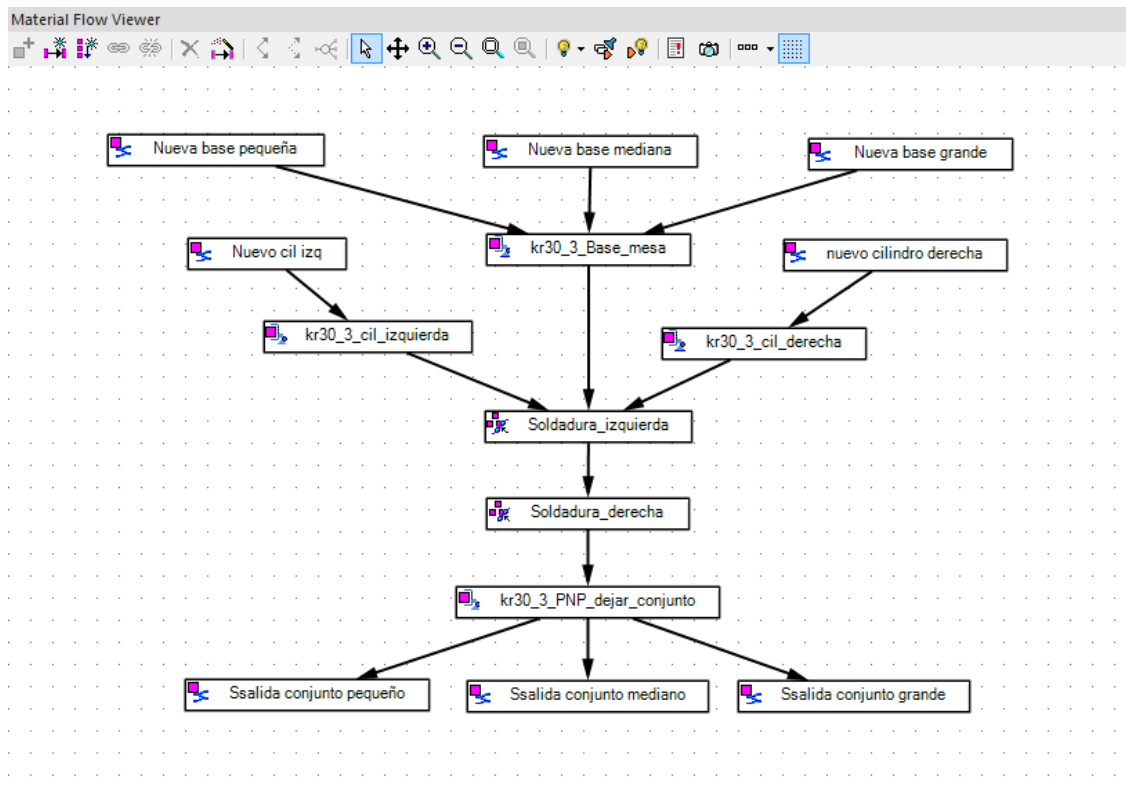


Fig. 44 Visor del flujo de material

Se trata de una estructura jerarquizada de flujo descendente, donde las primeras operaciones son operaciones de generación de nuevos objetos, ya que las operaciones en la célula comienzan por la generación de una base, sea pequeña, mediana o grande, y la generación de los tubos izquierdo y derecho. A continuación, tenemos la colocación de la base y de sendos tubos, realización de la soldadura de los tubos y la retirada y salida del conjunto ya soldado, quedando definido completamente el flujo de los materiales usados en la célula de soldadura.

Terminada esta fase de modelado, se tiene el gemelo digital de la célula de soldadura, es decir, se tiene el modelo virtual de la célula real. Llegados a este punto se pueden realizar simulaciones sobre el propio modelo con el “evaluador cíclico” disponible en el propio Tecnomatix PS para comprobar la correcta definición de todos los apartados detallados en este punto y poder realizar modificaciones antes de pasar a la programación del código de control mediante un PLC. Existe también la opción de poder implementar un humano dentro de la célula de soldadura para analizar el comportamiento de esta ante tareas de mantenimiento o posibles descuidos humanos. Sin embargo, esta opción queda fuera del alcance de este proyecto.

8.3. Control desde PLC

En este punto se abarca la realización del proyecto de control de la célula de soldadura mediante PLC, abordando tanto la parte hardware y comunicaciones como software.

8.3.1. Hardware de Control

A continuación, se pasa a configurar la periferia descentralizada de la célula de soldadura dentro del software de control. Para ello, en la herramienta de ingeniería TIA Portal, se agrega primeramente un nuevo dispositivo cuya referencia es SIMATIC ET 200M 153-4AA01-0XB0, contando con una tarjeta de entradas digitales de referencia DI 16xDC24V 321-1BH02_0AA0 y una tarjeta de salidas digitales DO 16xDC24V/0.5A 322-1BH01-0AA0. Este módulo se encarga de recibir y mandar señales de todos los dispositivos de la célula encargados de la sensorización, controlando el estado de los diferentes elementos de la célula; así como activación de los diferentes accionamientos de cintas transportadoras, mesa de trabajo o vallado de seguridad.

Por otra parte, se agregan los dos controladores de los robots. Estos son el modelo KR C4, pertenecientes a la gama de productos de KUKA. Estos controladores no se encuentran en la biblioteca de Siemens y, por tanto, se deberá instalar el archivo GSDML correspondiente al modelo de controlador escogido. Los archivos GSDML son archivos GSD escritos en formato XML para identificar dispositivos PROFINET. Este tipo de archivos contienen información acerca de las características y prestaciones básicas de un dispositivo. Por lo general todos los equipos se entregan con un archivo GSD y, en este caso, se ha descargado desde el sitio web del fabricante. Con el archivo GSDML, TIA Portal puede determinar tanto los datos básicos, como las opciones de comunicación y los diagnósticos disponibles.

8.3.2. Red de comunicaciones

Una vez introducidos todos los dispositivos en el proyecto, se conectan a una red PROFINET-IO. Para ello se conectan todos los módulos a la red PROFINET-IO creada y se asignan al controlador S7-1500. Finalmente, se les asigna la correspondiente dirección IP y se les fija las direcciones de entradas y salidas, manteniendo el orden acordado con el director del proyecto. El esquema prefijado es el mostrado en la figura 45.

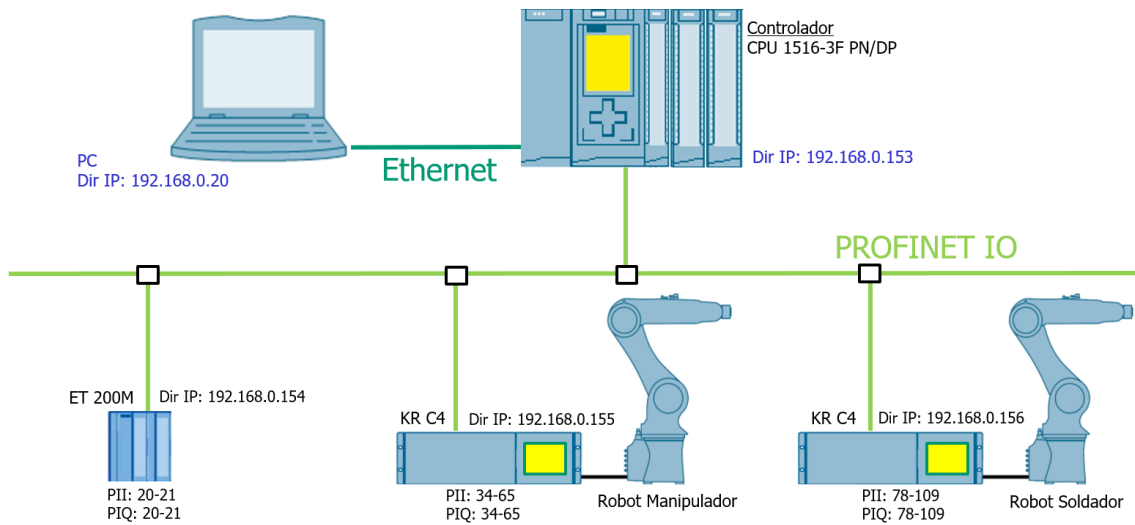


Fig. 45 Red de comunicaciones

Puede apreciarse que se trata de un controlador con funciones de seguridad ya que los controladores de los robots incluyen estas funciones. El controlador y los dispositivos I/O de periferia tienen las direcciones IP en orden ascendente, al igual que las direcciones E/S.

Terminada la configuración del hardware y de la red de comunicaciones de la célula se inicia el proceso de diseño del código de control de la célula de manipulación.

8.3.3. Software de Control

8.3.3.1. Diseño del código de control

Una vez terminada la configuración del hardware se procede a realizar diseño del software.

El código de la secuencia principal de la célula se implementa en el FC4, el cual es llamado desde el OB1 (Bloque de organización), que contiene la estructura del programa de control. En él se realiza la llamada al FB que ejecuta la secuencia principal (FB2) y, además, contiene la lógica necesaria para la comunicación con los robots. A continuación, se muestra una tabla con los parámetros de entrada y salida del FB2.

Tabla 7 Parámetros entrada salida FB2

Input	Tipo	Funcionamiento
SW_AUTO	Bool	Activación del modo automático de la secuencia
SW_TAP	Bool	Activación modo manual (paso a paso) de la secuencia
T_PUSH	Bool	Botón paso a paso de la secuencia manual
InOut	Tipo	Funcionamiento
Start	Bool	Activación de la célula desde botonera
HMI_Start	Bool	Activación de la célula desde HMI
Base en mesa	Bool	Presencia de base en la mesa de trabajo
Base en recepción	Bool	Presencia de base en la cinta de entrada
Conjunto salida	Bool	Presencia conjunto soldado en la cinta de salida
Conjunto soldado	Bool	Conjunto en mesa soldado
Fin programa Manip	Bool	Finalización operación de robot manipulador
Fin programa Soldador	Bool	Finalización operación de robot soldador
Tubo izq Recep	Bool	Presencia tubo izquierdo en el dispensador
Tubo der Recep	Bool	Presencia tubo derecho en el dispensador
Tubo izq Mesa	Bool	Presencia tubo izquierdo en la mesa de trabajo
Tubo der Mesa	Bool	Presencia tubo derecho en la mesa de trabajo
Conjunto en Soldadura	Bool	Presencia de conjunto listo para ser soldado
Sentido de giro	Bool	Indicador de sentido de giro de la mesa de trabajo
Fin Soldadura → Manip	Bool	Fin del movimiento horario de la mesa (180°)
Fin Manip → Soldadura	Bool	Fin del movimiento anti horario de la mesa (180°)
Manipulador_READY	Bool	Robot manipulador en reposo
Soldador_READY	Bool	Robot soldador en reposo
Soldadura_izq	Bool	Soldadura del tubo izquierdo
Soldadura_der	Bool	Soldadura del tubo derecho
Retirar conjunto	Bool	Permiso para poder retirar el conjunto

La secuencia principal de la célula de soldadura contenida en dicho FB se muestra en la figura 46.

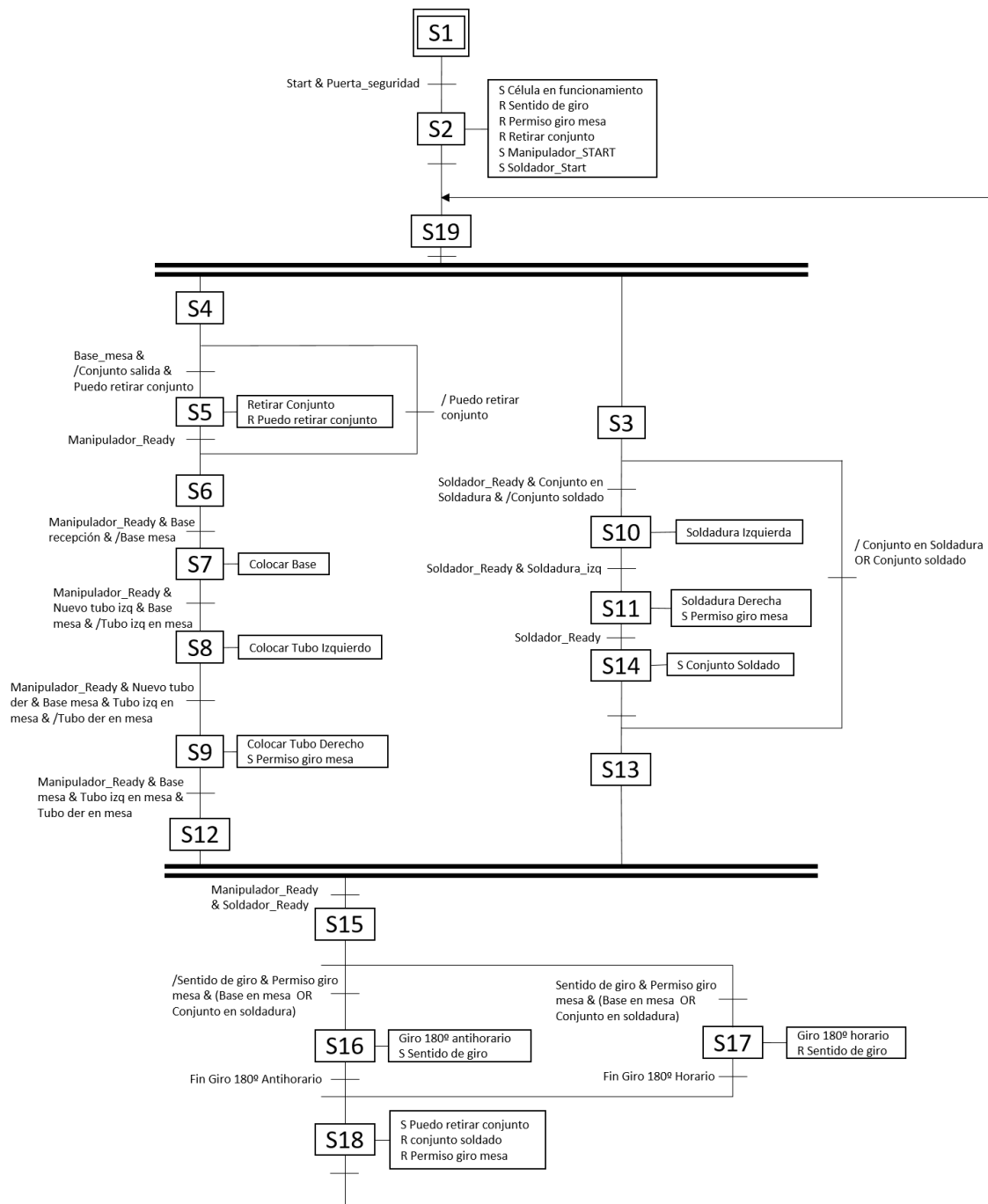


Fig. 46 Secuencia GRAFCET

A continuación, se explica el proceso que se realiza en cada etapa.

Etapa S1: Esta es la etapa inicial. Una vez seleccionada la forma de mando de la estación (modo automático o modo manual), esta etapa queda activada, quedando el FB a la espera que se cumpla la transición, la cual depende de la puerta de seguridad de la célula y el botón de arranque START.

Etapa S2: Esta etapa se encarga de mandar a ambos robots activar el programa número 1, el cual hace que entren dentro del bucle de espera de una nueva operación a cada robot mientras el START de cada robot esté activo. Además, activa y desactiva ciertas marcas necesarias para el funcionamiento de la célula como son las variables START de cada robot y el permiso de giro de la mesa.

Etapa S3: Etapa de transición.

Etapa S4: Etapa de transición.

Etapa S5: Etapa en la que se manda el número 5 al robot manipulador, el corresponde con la operación retirar conjunto. Esta etapa se cumple si existe en la mesa de trabajo un conjunto ya soldado para retirar. De lo contrario saltará a la siguiente etapa. Por tanto, esta etapa desactiva la marca de puedo retirar conjunto.

Etapa S6: Etapa de transición.

Etapa S7: Etapa que manda el número 2 al robot manipulador. Este programa se encarga de colocar una nueva base sobre la mesa de trabajo.

Etapa S8: Etapa que manda el número 3 al robot manipulador. Este programa se encarga de colocar el tubo izquierdo en la posición correspondiente en la nueva base sobre la mesa de trabajo.

Etapa S9: Etapa que manda el número 3 al robot manipulador. Este programa se encarga de colocar el tubo derecho en la posición correspondiente en la nueva base sobre la mesa de trabajo. Una vez realizada esta operación, el conjunto ya dispone de todas las piezas para pasar al robot de soldadura y, por tanto, también se activa el permiso de giro de mesa.

Etapa S10: Etapa que manda el número 2 al robot soldador. Este programa se encarga de realizar la soldadura del tubo izquierdo a la base si se cumplen las condiciones necesarias. De lo contrario, el robot soldador seguirá en posición de "ready".

Etapa S11: Etapa que manda el número 3 al robot soldador. Este programa se encarga de realizar la soldadura del tubo derecho a la base. Terminada la soldadura del tubo

derecho, el conjunto queda en condiciones de ser retirado, por lo que se habilita el giro de la mesa para que el robot manipulador pueda depositarlo en la cinta de salida.

Etapa S12: Etapa de transición.

Etapa S13: Etapa de transición.

Etapa S14: En esta etapa se activa la marca de conjunto soldado, lo que ayudará a saber al robot manipulador cuando tiene que retirar un conjunto.

Etapa S15: Etapa de transición.

Etapa S16: Etapa que se encarga de realizar el giro antihorario de la mesa de trabajo.

Etapa S17: Etapa que se encarga de realizar el giro horario de la mesa de trabajo.

Etapa S18: Etapa de actualización de diferentes marcas.

Etapa S19: Etapa de transición.

Además del FC4, desde el OB1 se llaman a tres FC más, todos ellos programados mediante lenguaje FUP de Siemens. Estos se han implementado para cubrir diferentes necesidades de la célula que quedaban fuera de la cadena principal o que son necesarios para realizar una puesta en marcha virtual pero que en el modelo real no es necesaria su implementación.

Por un lado, el FC1 contiene toda la lógica necesaria para la apertura y cierre de la puerta de seguridad, la selección de los modos automático y manual, y la lógica binaria para activar señales de funcionamiento del robot, tales como "Emergency_STOP" o "Program_Pause". Si bien las funciones de seguridad que realiza este FC han sido programadas en un FC sin seguridad, cabe destacar la posible implementación en un FC con tal capacidad. Sin embargo, queda fuera del alcance del proyecto, pero se recoge como futura modificación de la célula. Se le asigna el nombre "Lógica puerta de seguridad y formas de mando".

Por otro lado, el FC2 se implementa para activar diferentes marcas que servirán de ayuda para implementar posteriormente el panel de operador HMI. Se trata de las

alarmas de la ausencia de nuevas bases o tubos, ya sea en la cinta transportadora en el caso de las bases o en el dispensador de tubos en el caso de los tubos de agua caliente y fría. Dicho FC queda disponible para la inserción de nuevas funciones al HMI. Tiene como nombre "Alarmas HMI".

Finalmente, el FC3 se implementa sólo con propósito de realizar la puesta en marcha virtual, ya que se encarga de activar las señales que hacen que se generen nuevas piezas en la célula. Si bien es necesario en el modelo virtual, en el modelo real las bases llegarán a la cinta de entrada del proceso de prensa anterior y los tubos los generará el dispensador de tubos, el cual en el modelo virtual es un modelo teórico sin físicas. Por tanto, este FC activa y desactiva las variables de salida que llegan al modelo virtual para activar las operaciones de generación de piezas, operaciones ficticias que no existen en el modelo real. A este FC se le asigna el nombre "Recepción y salida de piezas". Pese a que la generación de piezas se podría haber realizado desde Tecnomatix PS, se otorga esta funcionalidad al control desde PLC para poder realizar un control de la simulación más compacto y desde el sistema SCADA.

Como ya se ha mencionado anteriormente, los robots utilizados para manipulación y soldadura son de la marca KUKA, por lo que su programación queda fuera de TIA Portal. Esta se realiza en la herramienta de ingeniería Tecnomatix PS, donde se definen los movimientos que va a realizar cada robot y el propio software genera el código de cada operación. Estas son agrupadas en un programa principal de cada robot y se crea un menú para seleccionar cada operación, dotando a cada una de estas de un número, el cual mandará el PLC en función de las necesidades.

Por tanto, debe existir una comunicación entre el controlador PLC y los robots, la cual se realiza vía PROFINET-IO. La comunicación se realiza en ambos sentidos. El PLC manda órdenes a los robots y estos informan de su estado al PLC. A continuación, se detalla el protocolo entre Robot y PLC:

- Una vez arrancada la célula de soldadura, o iniciada la simulación en el caso de gemelo digital, y pulsado el botón "Start" de la célula, el PLC manda al robot un 1 en el byte "Program_Number", el cual hace que comience a ejecutarse el programa del robot.
- El robot manda al PLC la señal "Robot_Ready", quedando este a la espera de que el PLC le mande la operación correspondiente.

- En función de las necesidades de la célula, el PLC manda el número del programa correspondiente al Robot. Este comienza a realizar dicha operación. Una vez empiece a ejecutar la operación, el Robot manda al PLC en el byte "Program_mirror" la operación que contiene el byte "Program_Number".
- El PLC recibe el número de la operación. Si esta coincide con la enviada el PLC manda un 1 en el byte "Program_Number" para que, cuando el robot termine con la operación que esté realizando, vuelva al bucle y quede a la espera de una nueva operación.

Los números de cada programa se muestran en la tabla 8. Se reserva el número 1 para los programas principales correspondientes a cada robot, ya que es necesario enviar al robot este comando para que vuelva a entrar en el bucle principal del programa. Nótese que se repiten números entre las distintas operaciones. Esto es posible ya que, como se ha mencionado, cada robot dispone de su controlador y, por tanto, se trata de controles independientes.

Tabla 8 Programas de los robots

Robot	Nº	Programa
Manipulador	2	Colocar Base en la mesa de trabajo
	3	Colocar el tubo izquierdo
	4	Colocar el tubo derecho
	5	Retirar el conjunto soldado
Soldador	2	Soldadura Izquierda
	3	Soldadura derecha

8.3.3.2. Asignación de las variables de la célula

La siguiente tarea consiste en la asignación del nombre de cada variable respetando la terminología especificada por el director del proyecto. Una correcta asignación de nombres a las variables hace que el código tenga una comprensión más clara y sencilla, por lo que será más fácil su modificación en futuros proyectos. Para el caso de la célula la lista de variables de entrada y salida es la siguiente. La lista completa de variables de la célula robotizada se muestra en el anexo I.

Tabla 9 Variables I/O de la célula de soldadura

Nombre	Tipo	Dirección	Nombre	Tipo	Dirección
Start	Bool	%I0.0	soldadura --> manip	Bool	%Q20.3
Emergency Stop	Bool	%I0.1	NUEVO CILINDRO DERECHA	Bool	%Q20.4
Nuevo Cilindro izquierdo	Bool	%I0.2	Cinta_OUT_base mediana	Bool	%Q20.5
Nuevo Cilindro derecho	Bool	%I0.3	Cinta_IN_base pequeña	Bool	%Q20.6
Nueva base pequeña	Bool	%I0.4	Cinta_IN_base grande	Bool	%Q20.7
Nueva base mediana	Bool	%I0.5	Cinta_OUT_base pequeña	Bool	%Q21.0
Nueva base grande	Bool	%I0.6	Cinta_OUT_base grande	Bool	%Q21.1
Puerta de seguridad	Bool	%I0.7	Abrir Puerta	Bool	%Q21.2
Fin movimiento cinta	Bool	%I20.0	Cerrar Puerta	Bool	%Q21.3
Fin nuevo cilindro izquierda	Bool	%I20.1	Pulsador PASO a PASO	Bool	%I1.1
Base en Recepcion	Bool	%I20.2	Manip_PAUSE	Bool	%Q34.4
Base en mesa	Bool	%I20.3	Manip_emergencySTOP	Bool	%Q34.5
Tubo izq en recepcion	Bool	%I20.4	Fin_Prog_Manip	Bool	%I34.0
Tubo izquierdo	Bool	%I20.5	Sold_PAUSE	Bool	%Q78.2
Tubo derecho	Bool	%I20.6	Fin_Prog_Sold	Bool	%I78.0
Conjunto en soldadura	Bool	%I20.7	Num_Prog_Soldador	Byte	%QB79
Fin nuevo cilindro derecha	Bool	%I21.0	Num_Prog_Manipulador	Byte	%QB35
Fin manip --> sold	Bool	%I21.1	Manipulador_START	Bool	%Q34.6
Conjunto salida	Bool	%I21.2	Num_Prog_Manip_espejo	Byte	%IB35
Fin sold --> manip	Bool	%I21.3	Num_Prog_Sold_espejo	Byte	%IB79
Tubo der en recepcion	Bool	%I21.4	Robot_Manip_READY	Bool	%I34.5
Conjunto FIN	Bool	%I21.5	Robot_Sold_READY	Bool	%I78.1
Fin Abrir Puerta	Bool	%I21.6	Soldador_START	Bool	%Q78.0
Fin Cerrar Puerta	Bool	%I21.7	Sold_izq_OK	Bool	%I78.2
Cinta_IN_base mediana	Bool	%Q20.0	Sold_der_OK	Bool	%I78.3
NUEVO CILINDRO IZQUIERDA	Bool	%Q20.1	Sold_emergencySTOP	Bool	%Q78.1
manip --> soldadura	Bool	%Q20.2			

8.3.3.3. Diseño de un HMI

En este apartado se describe la implementación de un Interfaz Humano Maquina HMI. Al igual que para el estudio del código de control, se matiza la metodología del diseño del HMI para la célula de soldadura.

A modo de introducción se describe que es un HMI. El Interfaz Hombre-Máquina (HMI) es el interfaz entre el proceso y el operario. Es la principal herramienta utilizada por operarios y supervisores de línea para coordinar y controlar procesos industriales y de fabricación. El HMI traduce variables de procesos complejos en información útil y procesable.

La función de los HMI consiste en mostrar información operativa en tiempo real. Proporcionan gráficos de procesos visuales que aportan significado y contexto al estado de los diferentes dispositivos y otros parámetros del proceso. Suministran información operativa al proceso.

Una vez recopilada la información necesaria acerca del HMI, comienza su diseño. Se crea una pantalla principal (figura 47). Esta consta de un botón de salida del "runtime", un botón con el icono de Tecnomatix PS correspondiente a la parte implementada para realizar la generación de nuevas piezas dentro de la célula, una imagen del proceso de montaje del calderín completo y 3 botones que representan los tres principales procesos de los que se compone la línea. Desde estas se puede acceder a la imagen de cualquiera de los apartados y en particular a la célula de soldadura de este proyecto. Los apartados de prensado y montaje quedan fuera del alcance de este proyecto.



Fig. 47 Pantalla principal HMI

La pantalla correspondiente con el proceso de soldadura se muestra a continuación.



Fig. 48 Pantalla célula de soldadura robotizada

En ella se aprecia, por un lado, una botonera en la parte inferior, la cual dispone de un pulsador de paro de emergencia, un selector de modo de funcionamiento (automático o manual con su correspondiente botón de paso a paso), un botón de arranque de la célula y los botones para navegar a las pantallas de visor de Grafset y Simulación. Estas pantallas se muestran más adelante.

A su vez, se aprecia una imagen de cada robot de la célula con sus correspondientes operaciones. Los leds correspondientes a cada operación se activarán al estar realizándose cualquiera de ellas.

Por último, se tiene un interruptor correspondiente a las dos posiciones posibles de la célula (abierta y cerrada) y tres indicadores leds que se activan con la ausencia de bases o tubos.

A continuación, se muestra la pantalla correspondiente a la secuencia principal de control de la célula de soldadura (figura 49).

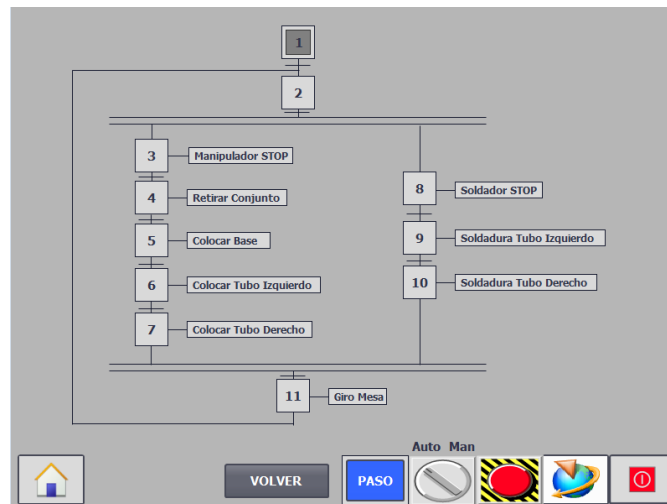


Fig. 49 Pantalla GRAFCET del HMI

Se aprecia cómo se trata de un diagrama simplificado, en el cual sólo se detallan las operaciones principales, omitiendo el resto de etapas auxiliares. Con ello se consigue un diagrama mucho más limpio y visual, características importantes en el diseño de un panel de operador. Esta secuencia GRAFCET se irá iluminando a medida que el proceso avance, simulando así el diagrama original visto en modo observador desde el entorno TIA Portal. Se aprecia cómo se mantiene la misma botonera inferior que en la pantalla anterior para poder realizar un cambio en el modo de operación de la célula o navegar a la ventana de simulación sin necesidad de tener que pasar por todas las ventanas, ofreciendo así una mayor comodidad y rapidez al usuario.

Finalmente, se muestra la pantalla de simulación.

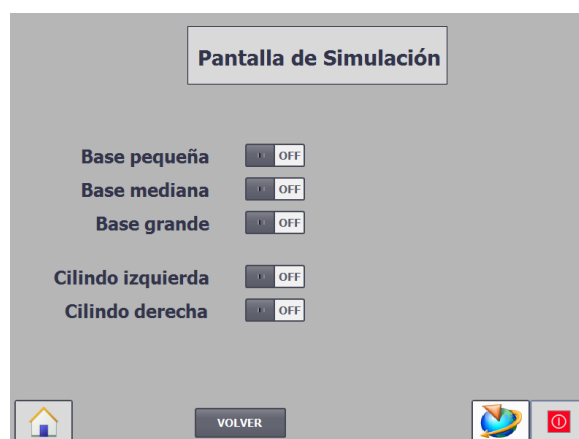


Fig. 50 Pantalla SIMULACIÓN del HMI

Se trata de una pantalla simple, la cual sólo es necesaria en simulación y no en la célula real. Consta de 5 interruptores para realizar la generación de bases y tubos. Destacar la

necesidad de generar tubos izquierda y derecha y un único tipo de base. Se programa de tal manera que, manteniendo el interruptor en la posición ON cada vez que en la posición donde el robot coge una nueva base o tubo conocidas como recepción no exista una pieza, se generará una nueva automáticamente.

De esta forma, la célula queda completamente controlada. Llegados a este punto se tiene el gemelo virtual de la célula de soldadura realizado en el apartado 6.4. y el control desde el PLC realizado en este apartado. Con estos dos se puede realizar la puesta en marcha virtual mediante "Software In the Loop" con un PLC virtual o mediante "Hardware In the Loop" con un PLC físico como es el caso de este proyecto. La configuración necesaria para la realización de dicha simulación se detalla en el siguiente apartado.

8.4. Puesta en Marcha Virtual mediante HiL

A continuación, se describe la metodología para realizar la puesta en marcha virtual de la célula de soldadura.

8.4.1. Configuración Hardware

Primeramente, se ha de distinguir entre la configuración hardware de la célula y de la configuración hardware de la puesta en marcha virtual. Como se ha establecido en el análisis de alternativas, se va a realizar la puesta en marcha virtual mediante "Hardware in the Loop". Se hace uso de un PLC de la gama S7-1500 y una tarjeta SIMIT UNIT PN128 (figura 51).



Fig. 51 Tarjeta SIMIT Unit PN 128

El SIMIT UNIT PN 128 puede simular hasta 128 dispositivos de campo PROFINET. Esta tarjeta se puede utilizar en modo mixto con dispositivos de campo reales existentes y como sistema de simulación puro. En esta versión, se puede simular una línea PROFINET. La parametrización y la operación se pueden hacer utilizando el software

SIMULATION Unit, estando conectada la tarjeta al PC a través de Ethernet (protocolo TCP/IP). Por consiguiente, la tarjeta se encargará de simular los 3 dispositivos IO de periferia existentes en la célula de soldadura. Se van a mostrar dos posibles configuraciones para realizar la simulación.

Una primera es el uso de dos PC diferentes (figura 53). Uno se conectará al PLC en una de sus bocas PROFINET. Este se destinará a la carga de programas en el PLC mediante el software TIA Portal. En el otro se ejecutarán SIMULATION Unit y Tecnomatix PS, los cuales siempre han de ejecutarse en el mismo ordenador. Este PC irá conectado vía Ethernet al puerto CTRL de la SIMIT Unit. Finalmente, la unidad SIMIT se conectará al PLC mediante su puerto PROFINET P1. El esquema correspondiente a esta configuración es el mostrado a continuación.

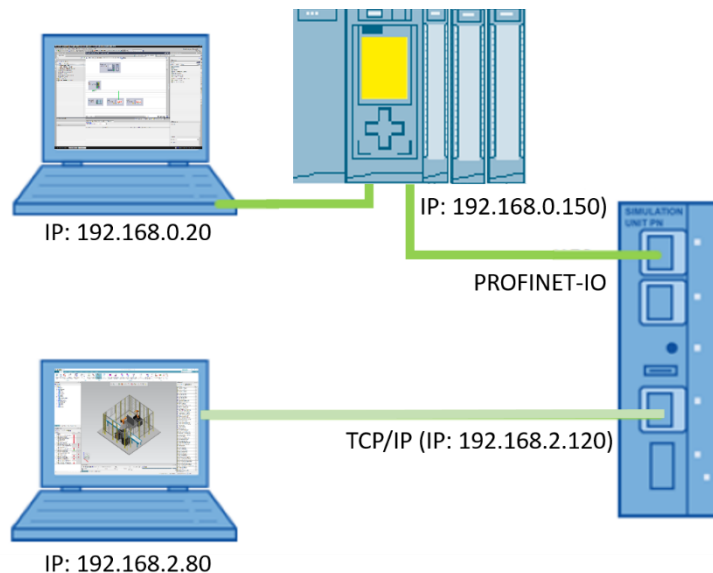


Fig. 52 Configuración hardware de la simulación con dos PCs

La segunda solución se basa en la utilización de dos tarjetas de red en el mismo PC, creando dos canales Ethernet en el mismo equipo. Esta solución ha sido comprobada con éxito, sin embargo, con un gran tráfico de datos podría generarse fallos durante la simulación. El esquema es el mostrado en la figura 53.

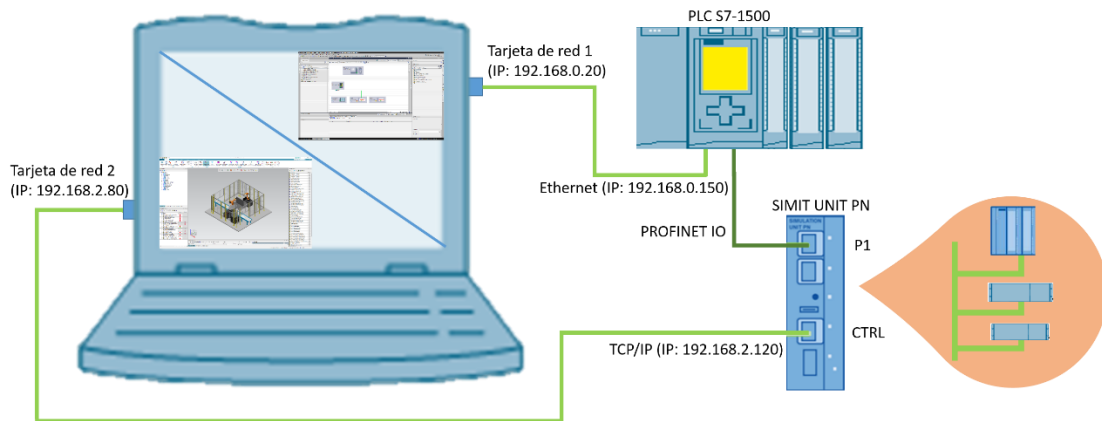


Fig. 53 Configuración Hardware de la simulación con un PC

Resaltar la importancia de que la unidad SIMIT y la tarjeta de red correspondiente pertenezcan a la misma subred (en la figura 192.168.2.xxx) ya que si no el software encargado del mapeado no dejará realizar la simulación.

Por tanto, la carga y descarga de programas al PLC se realiza mediante la tarjeta de red 1 vía Ethernet. La comunicación del controlador con la tarjeta SIMIT Unit se realiza mediante una red PROFINET-IO, misma red que estaría implementada en caso de tratarse de la célula real y se dispusiera de los dispositivos I/O de periferia que la tarjeta está simulando tal y como se muestra en la figura. Finalmente, la comunicación de la tarjeta SIMIT con el gemelo digital se realiza a través de la tarjeta de red 2 vía TCP/IP. Esta conexión simula la conexión real que hubiera en el modelo real de los dispositivos I/O con los diferentes componentes de la célula.

8.4.2. Configuración Software

Una vez conectado el hardware y establecidas sus direcciones IP se procede a realizar la configuración software. Para ello, se debe importar el hardware de los dispositivos de periferia I/O de la célula de soldadura en la herramienta de ingeniería SIMULATION Unit, la cual se encarga de cargar dicha configuración en la tarjeta SIMIT y de realizar el mapeado entre la aplicación de modelado y la tarjeta SIMIT. Para ello se debe generar desde TIA Portal un fichero de extensión *.oms. Este archivo contiene la información acerca del hardware del proyecto, así como la información acerca de la red de comunicaciones.

Siendo este un documento que trata de herramientas de Siemens, destacar que dos de sus grandes exponentes en simulación son NX y Tecnomatix PS, y para ambos la exportación del hardware es imprescindible.

Las herramientas de ingeniería que se usan para realizar dicho mapeado son Simulation Unit para Tecnomatix PS y SIMIT para NX. Sin embargo, la tendencia que están siguiendo las nuevas versiones de los diferentes softwares lleva a unificar la realización del mapeado de las señales de ambos programas con el uso de SIMIT. En la tabla 10 se muestran las posibles compatibilidades entre herramientas.

Tabla 10 Compatibilidad entre herramientas de ingeniería para mapeado de señales

	Simulation Unit V9.1 upd1	SIMIT V10	SIMIT V10 upd1
TIA Portal V15	Tecnomatix PS	NX	NX
TIA Portal V15.1	Desaparece	-	NX/Tecnomatix PS

Por tanto, la exportación del hardware desde la herramienta de control (TIA Portal) y la correspondiente importación a la herramienta que realiza el mapeado (Simulation Unit o SIMIT) se realiza mediante un archivo *.oms. Este archivo es generado siempre por TIA Portal al compilar el Hardware de un proyecto y automáticamente eliminado si no hay instalada ninguna aplicación que requiera de su uso. Para ello, es necesario lanzar un ejecutable (unlockHWconfig V3.1) que se encuentra en la ubicación donde se ha instalado la herramienta de ingeniería de modelado. Este fichero crea el directorio C:\TIAExports o C:\Users\Public\Documents\TIAExports, donde se almacenan los archivos *.oms de los diferentes proyectos.

Una vez generado el fichero e importado en el nuevo proyecto en SIMULATION Unit aparecen los diferentes dispositivos I/O de periferia con sus correspondientes tarjetas de entradas y salidas y sus correspondientes direcciones IP (ver figura 54). Llegados a este punto, una vez cargada esta configuración hardware en la unidad SIMIT, se puede cargar el proyecto de control en el PLC y ver que existe una correcta comunicación entre el controlador y la tarjeta SIMIT. Con esto se comprueba la correcta implementación del hardware del proyecto en la unidad SIMIT. Sin embargo, un error en la configuración del hardware desde TIA Portal no sería detectado hasta realizar la puesta en marcha real, ya que la configuración realizada en la tarjeta SIMIT es la exportada del propio proyecto de TIA Portal.

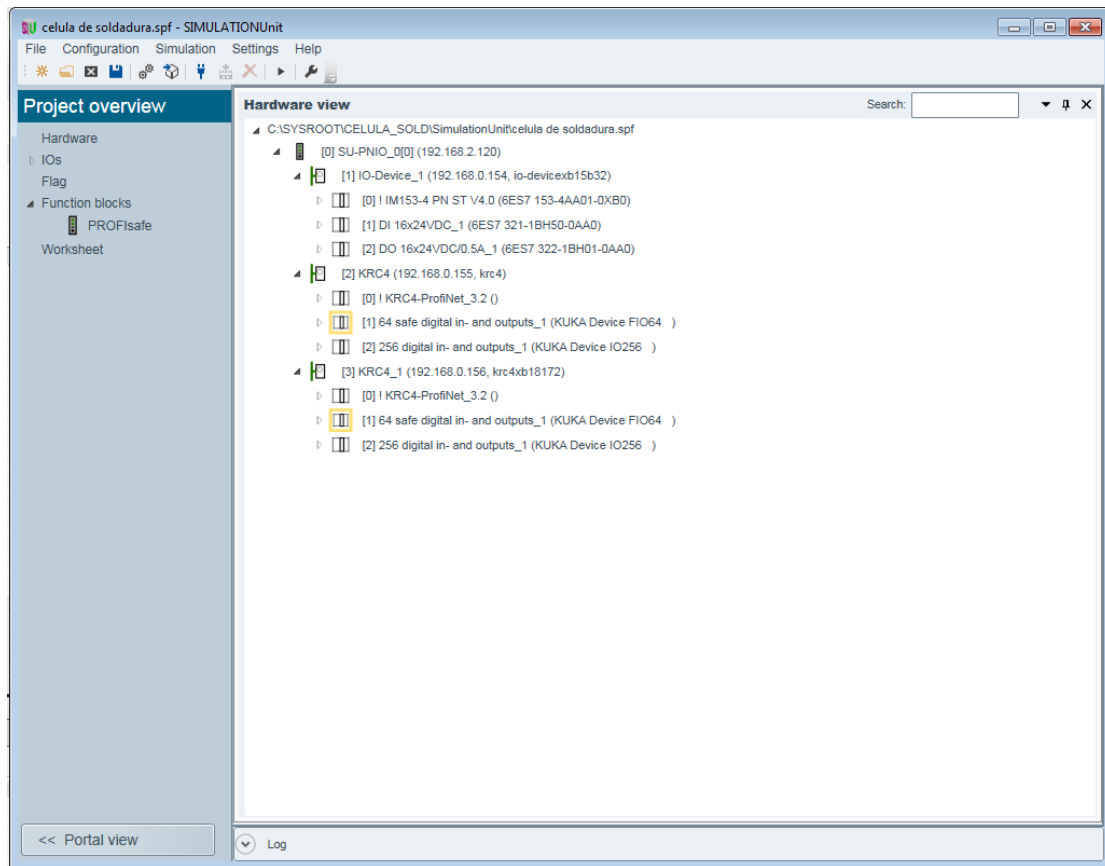


Fig. 54 Hardware importado en SIMULATION Unit

Por consiguiente, la tarjeta SIMIT simulará el dispositivo ET200 M y los dos controladores de KUKA.

Realizado el salvado del proyecto, se crea una carpeta con extensión *.spf, la cual contiene el mapeado de las E/S. Esta carpeta contiene diferentes archivos con los cuales se consigue realizar la comunicación entre la tarjeta SIMIT y Tecnomatix PS y así poder realizar la simulación del gemelo digital. El estado actual de las variables en Tecnomatix PS será actualizado en dicho directorio y leído por la tarjeta SIMIT, la cual enviará dicha información vía PROFINET-IO al controlador IO. A sí mismo, el controlador IO mandará la actualización de las salidas a los dispositivos IO simulados por la Unidad SIMIT, esta escribirá la información vía TCP/IP en el directorio anteriormente descrito y Tecnomatix PS leerá dicha información y realizará las operaciones convenientes.

Por todo esto, en Tecnomatix PS se tiene que configurar el tipo de simulación que se va a realizar. Existen dos grandes grupos (figura 55) explicados a continuación. Por un lado, el CCE (Cyclic Event Evaluation), el cual es un simulador del propio Tecnomatix PS basado en eventos. Para cada ciclo de la simulación, el CEE recopila y evalúa las señales

del PLC para determinar el flujo de la simulación. Este simulador funciona de forma cíclica, por lo que sólo finalizará cuando se pare la simulación. Este simulador es de gran ayuda para realizar un primer testeo de las operaciones configuradas.

Por otro lado, está la opción PLC, el cual puede ser un PLC virtual emulado con SIMIT, PLCSIM o PLCSIM Advanced o mediante un PLC real mediante una conexión externa. Con el primer método se consigue realizar una validación del código de control y un análisis de colisiones entre objetos, así como un primer acercamiento a tiempos de ciclo. El segundo, por el contrario, permite realizar una validación de las comunicaciones, ya que también se simula la red PROFINET-IO y el comportamiento del PLC real, el cual puede implementarse directamente en el modelo real. Como ya se ha mencionado, este último caso será el llevado a cabo en este proyecto.

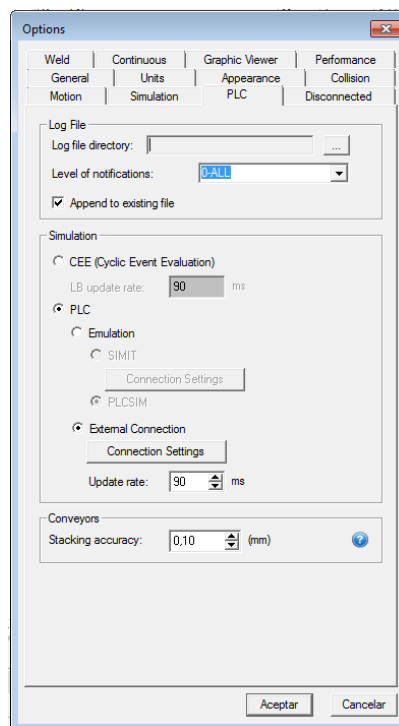
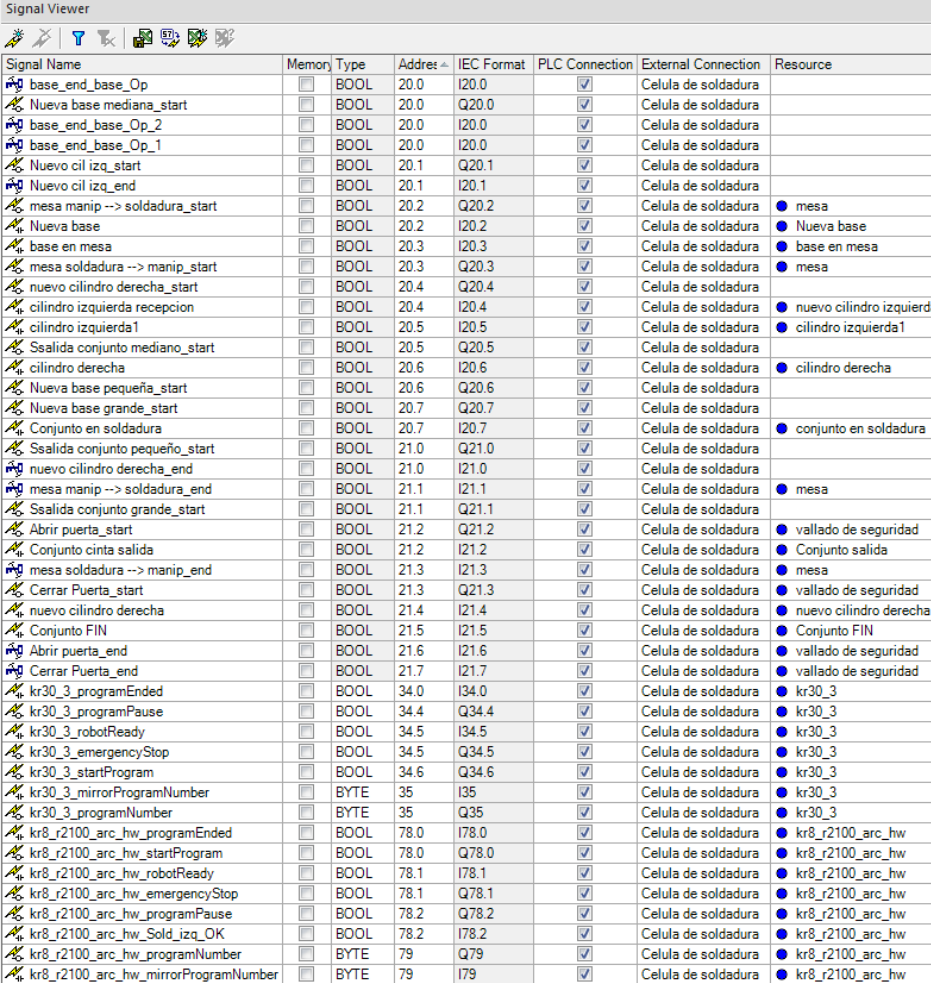


Fig. 55 Ventana de configuración de arquitectura de simulación

A su vez, en Tecnomatix PS se asigna la dirección a las variables de la célula de soldadura correspondiente con las direcciones establecidas en el código de control. Es importante respetar el mismo valor de señal de entrada o salida, ya que de lo contrario no se realizará la comunicación. Las señales correspondientes al robot se pueden generar en la propia configuración del robot como se ha mencionado antes. Estas son: "Start", "Program_Number", "Emergency_Stop", "Program_Mirror_Number", entre otras. A su vez, se tienen que generar las señales para realizar el movimiento de los diferentes

elementos de la célula cuando sea necesario. Para ello, desde el apartado de control de Tecnomatix PS se generan con ayuda de “Create All Flow Start Signals”, apareciendo así en el visor de señales. Las señales referentes a los sensores se crean automáticamente una vez generados los mismo. El visor de señales se muestra en la siguiente figura con la asignación realizada.



Signal Name	Memory	Type	Address	IEC Format	PLC Connection	External Connection	Resource
base_end_base_Op	<input type="checkbox"/>	BOOL	20.0	I20.0	<input checked="" type="checkbox"/>	Celula de soldadura	
Nueva base mediana_start	<input checked="" type="checkbox"/>	BOOL	20.0	Q20.0	<input checked="" type="checkbox"/>	Celula de soldadura	
base_end_base_Op_2	<input type="checkbox"/>	BOOL	20.0	I20.0	<input checked="" type="checkbox"/>	Celula de soldadura	
base_end_base_Op_1	<input checked="" type="checkbox"/>	BOOL	20.0	I20.0	<input checked="" type="checkbox"/>	Celula de soldadura	
Nuevo cil izq_start	<input checked="" type="checkbox"/>	BOOL	20.1	Q20.1	<input checked="" type="checkbox"/>	Celula de soldadura	
Nuevo cil izq_end	<input checked="" type="checkbox"/>	BOOL	20.1	I20.1	<input checked="" type="checkbox"/>	Celula de soldadura	
mesa manip --> soldadura_start	<input checked="" type="checkbox"/>	BOOL	20.2	Q20.2	<input checked="" type="checkbox"/>	Celula de soldadura	● mesa
Nueva base	<input checked="" type="checkbox"/>	BOOL	20.2	I20.2	<input checked="" type="checkbox"/>	Celula de soldadura	● Nueva base
base en mesa	<input checked="" type="checkbox"/>	BOOL	20.3	I20.3	<input checked="" type="checkbox"/>	Celula de soldadura	● base en mesa
mesa soldadura --> manip_start	<input checked="" type="checkbox"/>	BOOL	20.3	Q20.3	<input checked="" type="checkbox"/>	Celula de soldadura	● mesa
nuevo cilindro derecha_start	<input checked="" type="checkbox"/>	BOOL	20.4	Q20.4	<input checked="" type="checkbox"/>	Celula de soldadura	
cilindro izquierda recepcion	<input checked="" type="checkbox"/>	BOOL	20.4	I20.4	<input checked="" type="checkbox"/>	Celula de soldadura	● nuevo cilindro izquierda
cilindro izquierda1	<input checked="" type="checkbox"/>	BOOL	20.5	I20.5	<input checked="" type="checkbox"/>	Celula de soldadura	● cilindro izquierda1
Ssalida conjunto mediano_start	<input checked="" type="checkbox"/>	BOOL	20.5	Q20.5	<input checked="" type="checkbox"/>	Celula de soldadura	
cilindro derecha	<input checked="" type="checkbox"/>	BOOL	20.6	I20.6	<input checked="" type="checkbox"/>	Celula de soldadura	● cilindro derecha
Nueva base pequeña_start	<input checked="" type="checkbox"/>	BOOL	20.6	Q20.6	<input checked="" type="checkbox"/>	Celula de soldadura	
Nueva base grande_start	<input checked="" type="checkbox"/>	BOOL	20.7	Q20.7	<input checked="" type="checkbox"/>	Celula de soldadura	
Conjunto en soldadura	<input checked="" type="checkbox"/>	BOOL	20.7	I20.7	<input checked="" type="checkbox"/>	Celula de soldadura	● conjunto en soldadura
Ssalida conjunto pequeño_start	<input checked="" type="checkbox"/>	BOOL	21.0	Q21.0	<input checked="" type="checkbox"/>	Celula de soldadura	
nuevo cilindro derecha_end	<input checked="" type="checkbox"/>	BOOL	21.0	I21.0	<input checked="" type="checkbox"/>	Celula de soldadura	
mesa manip --> soldadura_end	<input checked="" type="checkbox"/>	BOOL	21.1	I21.1	<input checked="" type="checkbox"/>	Celula de soldadura	● mesa
Ssalida conjunto grande_start	<input checked="" type="checkbox"/>	BOOL	21.1	Q21.1	<input checked="" type="checkbox"/>	Celula de soldadura	
Abrir puerta_start	<input checked="" type="checkbox"/>	BOOL	21.2	Q21.2	<input checked="" type="checkbox"/>	Celula de soldadura	● vallado de seguridad
Conjunto cinta salida	<input checked="" type="checkbox"/>	BOOL	21.2	I21.2	<input checked="" type="checkbox"/>	Celula de soldadura	● Conjunto salida
mesa soldadura --> manip_end	<input checked="" type="checkbox"/>	BOOL	21.3	I21.3	<input checked="" type="checkbox"/>	Celula de soldadura	● mesa
Cerrar Puerta_start	<input checked="" type="checkbox"/>	BOOL	21.3	Q21.3	<input checked="" type="checkbox"/>	Celula de soldadura	● vallado de seguridad
nuevo cilindro derecha	<input checked="" type="checkbox"/>	BOOL	21.4	I21.4	<input checked="" type="checkbox"/>	Celula de soldadura	● nuevo cilindro derecha
Conjunto FIN	<input checked="" type="checkbox"/>	BOOL	21.5	I21.5	<input checked="" type="checkbox"/>	Celula de soldadura	● Conjunto FIN
Abrir puerta_end	<input checked="" type="checkbox"/>	BOOL	21.6	I21.6	<input checked="" type="checkbox"/>	Celula de soldadura	● vallado de seguridad
Cerrar Puerta_end	<input checked="" type="checkbox"/>	BOOL	21.7	I21.7	<input checked="" type="checkbox"/>	Celula de soldadura	● vallado de seguridad
kr30_3_programEnded	<input checked="" type="checkbox"/>	BOOL	34.0	I34.0	<input checked="" type="checkbox"/>	Celula de soldadura	● kr30_3
kr30_3_programPause	<input checked="" type="checkbox"/>	BOOL	34.4	Q34.4	<input checked="" type="checkbox"/>	Celula de soldadura	● kr30_3
kr30_3_robotReady	<input checked="" type="checkbox"/>	BOOL	34.5	I34.5	<input checked="" type="checkbox"/>	Celula de soldadura	● kr30_3
kr30_3_emergencyStop	<input checked="" type="checkbox"/>	BOOL	34.5	Q34.5	<input checked="" type="checkbox"/>	Celula de soldadura	● kr30_3
kr30_3_startProgram	<input checked="" type="checkbox"/>	BOOL	34.6	Q34.6	<input checked="" type="checkbox"/>	Celula de soldadura	● kr30_3
kr30_3_mirrorProgramNumber	<input checked="" type="checkbox"/>	BYTE	35	I35	<input checked="" type="checkbox"/>	Celula de soldadura	● kr30_3
kr30_3_programNumber	<input checked="" type="checkbox"/>	BYTE	35	Q35	<input checked="" type="checkbox"/>	Celula de soldadura	● kr30_3
kr8_r2100_arc_hw_programEnded	<input checked="" type="checkbox"/>	BOOL	78.0	I78.0	<input checked="" type="checkbox"/>	Celula de soldadura	● kr8_r2100_arc_hw
kr8_r2100_arc_hw_startProgram	<input checked="" type="checkbox"/>	BOOL	78.0	Q78.0	<input checked="" type="checkbox"/>	Celula de soldadura	● kr8_r2100_arc_hw
kr8_r2100_arc_hw_robotReady	<input checked="" type="checkbox"/>	BOOL	78.1	I78.1	<input checked="" type="checkbox"/>	Celula de soldadura	● kr8_r2100_arc_hw
kr8_r2100_arc_hw_emergencyStop	<input checked="" type="checkbox"/>	BOOL	78.1	Q78.1	<input checked="" type="checkbox"/>	Celula de soldadura	● kr8_r2100_arc_hw
kr8_r2100_arc_hw_programPause	<input checked="" type="checkbox"/>	BOOL	78.2	Q78.2	<input checked="" type="checkbox"/>	Celula de soldadura	● kr8_r2100_arc_hw
kr8_r2100_arc_hw_Sold_izq_OK	<input checked="" type="checkbox"/>	BOOL	78.2	I78.2	<input checked="" type="checkbox"/>	Celula de soldadura	● kr8_r2100_arc_hw
kr8_r2100_arc_hw_programNumber	<input checked="" type="checkbox"/>	BYTE	79	Q79	<input checked="" type="checkbox"/>	Celula de soldadura	● kr8_r2100_arc_hw
kr8_r2100_arc_hw_mirrorProgramNumber	<input checked="" type="checkbox"/>	BYTE	79	I79	<input checked="" type="checkbox"/>	Celula de soldadura	● kr8_r2100_arc_hw

Fig. 56 Panel de señales de la célula de soldadura

Finalmente, se ejecuta la simulación. Destacar que, pese a realizar la simulación mediante HiL, la propia herramienta de ingeniería Tecnomatix PS cuenta con un panel de simulación (figura 57) en el que se pueden incluir las diferentes señales de la célula sobre las que se quiera tener información o actuar durante la simulación. Este es el motivo por el cual el panel de simulación del HMI y toda la lógica implementada en el programa de control para poder realizar la generación de nuevas piezas no es estrictamente necesaria y se podría eliminar para simplificar el código.

Simulation Panel

Simulation	Inputs	Outputs	LB	Forced	Forced Value
celula					
kr30_3_end_kr30_3_Nueva base	■			<input type="checkbox"/>	■
Conjunto en soldadura	■			<input type="checkbox"/>	■
cilindro izquierda recepcion	■			<input type="checkbox"/>	■
cilindro derecha	■			<input type="checkbox"/>	■
kr30_3_cil_derecha_end	■			<input type="checkbox"/>	■
Conjunto cinta salida	■			<input type="checkbox"/>	■
Ssalida conjunto mediano_start		●		<input type="checkbox"/>	■
kr8_r2100_arc_hw_programPause		●		<input type="checkbox"/>	■
kr30_3_programPause		●		<input type="checkbox"/>	■
kr30_3_PNP_dejar_conjunto_end	■			<input type="checkbox"/>	■
kr30_3_programEnded	■			<input type="checkbox"/>	■
kr30_3_programNumber		0		<input type="checkbox"/>	2
kr30_3_mirrorProgramNumber	3			<input type="checkbox"/>	0
kr30_3_startProgram		●		<input type="checkbox"/>	■
kr30_3_robotReady	■			<input type="checkbox"/>	■
kr30_3_emergencyStop		●		<input type="checkbox"/>	■
kr30_3_cil_izquierda_end	■			<input type="checkbox"/>	■
kr8_r2100_arc_hw_programNum...		0		<input type="checkbox"/>	0
kr8_r2100_arc_hw_mirrorProgra...	0			<input type="checkbox"/>	0
kr8_r2100_arc_hw_Sold_izq_OK	■			<input type="checkbox"/>	■

Fig. 57 Panel de simulación de Tecnomatix PS

En conclusión, gracias a la potencia del software utilizado se tiene un gemelo digital de un proceso real, nuevo o ya existente, con el cual se pueden verificar códigos de control, tiempos de producción, colisiones por modificaciones en layouts, entre otras. Con ello se consigue un abaratamiento en puestas en marcha, comprobación de modificaciones o nuevas implementaciones en una línea de producción sin necesidad de parar la línea real, análisis de coste/beneficio con la implantación de nuevos dispositivos, comprobación de los mecanismos de seguridad de la célula entre otros.

Se obtiene un correcto funcionamiento de la célula con capacidad de ser implementada.

9. DESCRIPCIÓN DE LAS TAREAS

A continuación, se describen las tareas llevadas a cabo para la realización del proyecto. El desarrollo del proyecto comienza el jueves 01/11/18 y la duración de este es de un total de 148 días. El periodo laboral del proyecto es de lunes a viernes y se han excluido como días laborales el periodo de exámenes.

Las tareas a realizar en este proyecto se dividen en cinco grandes grupos como se muestra a continuación.

Tabla 11 Tareas principales del proyecto

EDT	Nombre	Duración	Predecesoras
1.1	Estudio Herramientas de Ingeniería	15 días	
1.2	Estudio técnicas Virtual Commissioning	15 días	
1.3	Especificación del Proyecto	6 días	2;3
1.4	Verificación de la propuesta de la solución	0 días	4
1.5	Modelado de la Célula de soldadura	60 días	4
1.6	Verificación del Modelado	0 días	9
1.7	Diseño del control de la célula	21 días	12
1.8	Verificación del control de la célula	0 días	13
1.9	Puesta en marcha Virtual	16 días	23
1.10	Verificación puesta en marcha virtual	0 días	24
1.11	Memoria	30 días	24
1.12	Entrega de la documentación y proyecto	0 días	34

Como se aprecia en la tabla 11, las tareas principales son Especificaciones del Proyecto, Modelado de la Célula de soldadura, Diseño del control de la célula, Puesta en marcha Virtual y Memoria. Estas se realizan de forma secuencial y no se da paso a la siguiente hasta la verificación del paso anterior. Según se avanza, se realizan pruebas de verificación específicas para cada apartado, ya que se tratan de tareas independientes entre sí, en las que también se comprueba en menor medida la compatibilidad con los niveles anteriores. Al final del proyecto se realizan todas las pruebas en conjunto a la par que se documentan, con el objetivo de validar el conjunto de la aplicación.

Se observa como después de cada tarea principal hay un hito. Estos indican que la tarea correspondiente ha sido finalizada y documentada para poder redactar después correctamente la memoria del proyecto (apartado final). En cada hito se celebra una reunión con el director del proyecto para mostrarle el estado actual del modelo virtual y del código de control de célula de soldadura.

Nótese que la tarea con mayor duración es la parte de Modelado de la célula de soldadura. Se hará especial hincapié ya que también se pueden realizar tareas de simulación del modelo sin necesidad de realizar un control mediante PLC. Es por estas pruebas de puesta a punto y validación que se tarda más en realizar esta.

La planificación contiene dieciséis tareas críticas mostradas en la tabla 12.

Tabla 12 Listado de tareas críticas

EDT	Nombre	Duración	Predecesoras
1.1	Estudio Herramientas de Ingeniería	15 días	
1.2	Estudio técnicas Virtual Commissioning	15 días	
1.3.1	Diseño de célula de soldadura	3 días	
1.3.2	Selección de los elementos de la célula	3 días	5
1.5.2	Modelado de Célula de soldadura	60 días	
1.6	Verificación del Modelado	0 días	9
1.7.1	Instalación herramienta de ingeniería TIA Portal	1 día	
1.7.4.1	Diseño del código de control	15 días	
1.7.4.2	Diseño de un HMI	5 días	20
1.7.5	Verificación del Software	0 días	19
1.8	Verificación del control de la célula	0 días	13
1.9.1	Instalación herramienta de ingeniería SIMULATION Unit	1 día	
1.9.4.1	Mapeado de las señales	15 días	
1.9.5	Verificación Software	0 días	30
1.11.1	Redacción de la memoria del trabajo	30 días	
1.12	Entrega de la documentación y proyecto	0 días	34

Pese a que este proyecto tiene bastante flexibilidad temporal se lleva una mayor supervisión de estas, ya que un retraso en cualquiera de ellas conlleva a un retraso en la planificación del proyecto, pudiendo no entregar a tiempo el gemelo digital, el diseño del código y la documentación correspondiente. El resto de tareas, en cambio, disponen de un margen, por lo que, a pesar de que no deben ser desatendidas, podrán sufrir retrasos en su realización.

A continuación, se muestra el diagrama de Gantt con las tareas principales del proyecto. En el anexo IV se muestra el diagrama de Gantt completo del proyecto.

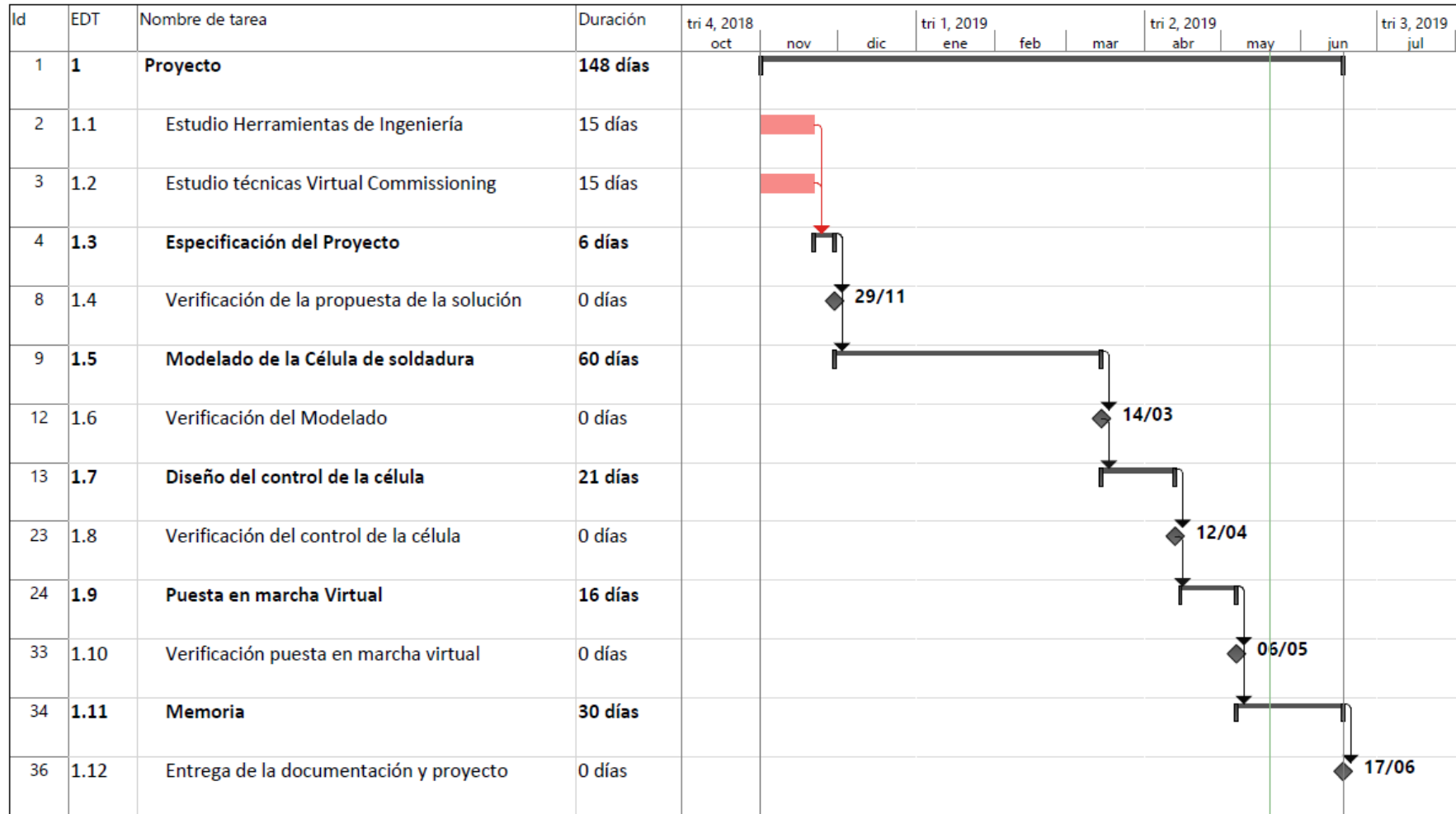


Fig. 58 Diagrama de Gantt simplificado

10. DESCARGO DE GASTOS

Para la realización de este proyecto se emplean un ordenador, un PLC y una tarjeta SIMIT Unit presentes en el laboratorio del Departamento de Ingeniería de Sistemas y Automática de la UPV/EHU. Con el ordenador se realizan el modelado de la célula de soldadura y el código de control del PLC y con ayuda del PLC y la SIMIT Unit se realiza la puesta en marcha virtual de la célula. Para este proyecto se emplea licencias y software de Tecnomatix PS V14.1, licencia de controlador KUKA virtual, TIA Portal V15.1 y licencia WinCC Advanced para el diseño y simulación del panel de operador.

El cálculo de las horas empleadas se realiza de modo aproximado, considerando el número de días dedicados a cada tarea en cuestión según refleja el diagrama de Gantt del proyecto. Se ha estimado que cada día se dedica una media de 5 horas de trabajo. Por otro lado, se tiene en cuenta la vida útil de los elementos empleados de cara a su amortización, tanto licencias (vida útil 2 años) como dispositivos (ordenador 4 años aproximadamente y PLC y SIMIT Unit 10 años).

Por otro lado, se hace el cálculo del coste del proyecto si se hubiera realizado una puesta en marcha real, por lo que se tiene en cuenta el coste de los diferentes elementos de la célula. Se ha estimado que las horas dedicadas por el alumno y director del proyecto son las mismas para ambos tipos de puesta en marcha, ya que las horas dedicadas al modelado de la célula de soldadura quedan compensadas con las de la puesta en marcha real. El cálculo del proyecto con puesta en marcha real se muestra en el anexo V.

Se han tenido en cuenta unos costes indirectos del 4% donde se incluyen el consumo, luz, internet, entre otros. Para la puesta en marcha real se han estimado unos costes indirectos del 10% ya que se tienen en cuenta costes de transporte de elementos, personas y resto de operaciones logísticas que se necesitan en una puesta en marcha real.

En la tabla 13 se muestra el presupuesto referente a la puesta en marcha virtual. En esta puede observarse que el precio total del proyecto es de 23.172,28 €.

A modo de conclusión, mencionar que el mayor coste se haya en lo que hubiera que pagar al director y al alumno por las horas dedicadas al proyecto. Por otra parte, el principal coste técnico es la amortización de la herramienta de ingeniera Tecnomatix PS, hecho natural debido a la relación del coste con la vida útil. Finalmente, la

amortización de los equipos o el consumo eléctrico de estos son costes menores en comparación.

Tabla 13 Presupuesto del proyecto

<u>Concepto</u>	<u>Unidad</u>	<u>nº de unidades</u>	<u>Tasa horaria</u>	<u>Coste</u>
Horas internas				18.300,00 €
Director de proyecto	h	70	50 €	3500
Alumno	h	740	20 €	14800
Amortizaciones				261,25 €
PLC S-1500	h	100	0,34 €	34,17 €
PC sobremesa	h	740	0,21 €	154,17 €
SIMIT Unit	h	350	0,21 €	72,92 €
Robot KUKA KR30	h	100	0,00 €	0,00 €
Robot KUKA kr8 r2100 arc hw	h	100	0,00 €	0,00 €
Controlador KUKA KR C4	h	100	0,00 €	0,00 €
Garras Robots	h	80	0,00 €	0,00 €
Resto elementos	h	80	0,00 €	0,00 €
Software				3.719,79 €
Software TIA Portal	h	185	0,83 €	154,17 €
Software WinCC	h	50	0,29 €	14,58 €
Runtime WinCC Advanced (512)	h	50	0,19 €	9,38 €
Tecnomatix PS V14.1	h	380	8,33 €	3.166,67 €
Controlador Virtual KUKA	h	180	2,08 €	375,00 €
Subtotal				22.281,04 €
Costes indirectos	4%			891,24 €
TOTAL				23.172,28 €

Por otra parte, la naturaleza de este proyecto hace que sea difícil calcular tiempos de amortización o el VAN/TIR correspondiente, ya que de antemano no se sabe qué impacto económico tendrá. Esto se debe a que este proyecto se trata de un trabajo de investigación, por lo que los beneficios no se ven reflejados en la capacidad de producción de la célula. Se espera que el beneficio de esta inversión sea el acercamiento a las tecnologías de la industria 4.0 para poder usarlas en proyectos de producción real. Es por esto que se realiza el presupuesto que sería necesario para realizar el mismo proyecto en una puesta en marcha real para ver el potencial real de las herramientas usadas en este proyecto.

Por tanto, como puede apreciarse en el anexo V, el coste del proyecto total realizando una puesta en marcha real es de 87.385,60 €. Este aumento del coste se debe en su totalidad a la inversión necesaria en los equipos para poder realizar la puesta en marcha, hecho que no sucede en la puesta en marcha virtual, donde la inversión necesaria en software es mucho más elevada pero rentable al mismo tiempo, ya que una vez

realizada la puesta en marcha y analizados tiempos de producción u otras variables se puede llegar a la conclusión de que el proyecto no es viable y rechazar la propuesta sin necesidad de realizar la inversión anteriormente citada.

11. CONCLUSIONES

Este proyecto se enmarca en las tecnologías de la Industria 4.0, en el cual se ha realizado la puesta en marcha virtual de una célula de soldadura mediante "Hardware in the Loop". El modelo del gemelo digital se ha realizado con la herramienta de ingeniería Process Simulate y el código de control desde PLC se codifica con TIA Portal.

Con el modelado del gemelo digital se consigue un entorno protegido para poder analizar modificaciones en proyectos actuales o realizar la puesta en marcha virtual de nuevos, optimizando el control de los parámetros de producción. Se consigue una verificación del funcionamiento de los modelos reales a nivel físico y cinemático incluso antes de producirlos.

La puesta en marcha virtual permite la simulación en tiempo real de extensos proyectos de ingeniería para diseñar y probar proyectos de automatización sin los riesgos de trabajar en un entorno con sistemas reales. En consecuencia, las pruebas de diseño de sistemas de automatización y la entrega de códigos de mayor calidad pueden acortar los tiempos de puesta en servicio, reducir el riesgo del proyecto, corregir errores humanos.

Se consigue una puesta en marcha más rápida, con una parte de la ingeniería probada en una etapa anterior, casi todos los errores funcionales y de diseño pueden detectarse antes de la puesta en servicio final. Esto reduce considerablemente las escalas de tiempo hasta un 30% y permite una puesta en marcha más rápida que la prevista previamente.

Se obtiene mayor calidad de ingeniería ya que es posible ampliar el alcance habitual de las pruebas y la simulación se puede realizar en paralelo a tareas de ingeniería que conducen a resultados de calidad de ingeniería superiores. El enfoque virtual puede ser un soporte altamente eficiente para la definición y aclaración de especificaciones funcionales, evitando la necesidad de costosas tareas de reingeniería.

Se consiguen unos costes más bajos, puesto que se consigue, por un lado, una reducción de posibles incidentes de reingeniería y, por otro lado, una reducción de los tiempos de puesta en marcha final, sin la necesidad de invertir en maquinaria u otros dispositivos hasta tener una validación del rendimiento real de la célula.

Con estas tecnologías de puesta en marcha virtual se consigue validar en simulación una parte mucho mayor del sistema de control (hasta un 50%), de forma que la puesta en marcha final requiera un tiempo mucho más reducido, del orden de un 30% menos.

Se reducen los riesgos con la puesta en marcha virtual utilizando herramientas no intrusivas como Tecnomatix Process Simulate o SIMIT, permitiendo realizar las correspondientes pruebas sin la participación de máquinas, equipos o personal reales, eliminando así los riesgos inherentes asociados con la puesta en servicio in situ. Cualquier posible mal funcionamiento no tiene consecuencias graves, y las empresas pueden evitar daños costosos a los activos o lesiones a las personas.

Con un marco de simulación flexible y escalable, los fabricantes tienen acceso al método más fácil, rápido y reduciendo riesgo para la finalización de proyectos de tecnología de automatización, ya sea un sistema de control, una línea de producción reconfigurada o, incluso, una nueva planta de fabricación.

12. FUENTES DE INFORMACIÓN

- <https://www.plm.automation.siemens.com/global/es/products/tecnomatix/assembly-simulation.html>
- https://en.wikipedia.org/wiki/Hardware-in-the-loop_simulation
- <https://www2.deloitte.com/es/es/pages/manufacturing/articles/que-es-la-industria-4.0.html#>
- <https://www.revistadyna.com/busqueda/puesta-en-marcha-virtual-de-automatizacion-de-una-planta-industrial>
- <https://www.engusa.com/es/product/siemens-tecnomatix-process-simulate>
- <https://avantek.es/productos/tecnomatix/process-simulate/>
- <https://w3.siemens.com/mcms/distributed-io/en/ip20-systems/et200m/interface-modules/pages/default.aspx>
- <https://www.kuka.com/es-es>

ANEXO I: LISTA COMPLETA DE VARIABLES DE LA CÉLULA ROBOTIZADA

Nombre	Tipo	Dirección	Nombre	Tipo	Dirección
Start	Bool	%I0.0	Tag_1	Bool	%M3.0
Emergency Stop	Bool	%I0.1	Tag_2	Bool	%M3.1
Nuevo Cilindro izquierdo	Bool	%I0.2	Num_Prog_Soldador	Byte	%QB79
Nuevo Cilindro derecho	Bool	%I0.3	Num_Prog_Manipulador	Byte	%QB35
Nueva base pequeña	Bool	%I0.4	Tag_3	Bool	%M3.2
Nueva base mediana	Bool	%I0.5	Tag_4	Bool	%M3.3
Nueva base grande	Bool	%I0.6	Tag_5	Bool	%M3.4
Puerta de seguridad	Bool	%I0.7	Tag_6	Bool	%M3.5
HMI_Start	Bool	%M0.0	Tag_7	Bool	%M3.6
HMI_Nuevo Cilindro izquierdo	Bool	%M0.2	Tag_8	Bool	%M3.7
HMI_Nuevo Cilindro derecho	Bool	%M0.3	Tag_9	Bool	%M4.0
HMI_Nueva base pequeña	Bool	%M0.4	Tag_10	Bool	%M4.1
HMI_Nueva base mediana	Bool	%M0.5	Tag_11	Bool	%M4.2
HMI_Nueva base grande	Bool	%M0.6	Tag_12	Bool	%M4.3
HMI_Puerta de seguridad	Bool	%M0.7	Tag_13	Bool	%M4.4
Celula en funcionamiento	Bool	%M1.0	Tag_14	Bool	%M4.5
Fin movimiento cinta	Bool	%I20.0	Tag_15	Bool	%M4.6
Fin nuevo cilindro izquierda	Bool	%I20.1	Tag_16	Bool	%M4.7
Base en Recepcion	Bool	%I20.2	Tag_17	Bool	%M5.0
Base en mesa	Bool	%I20.3	Tag_18	Bool	%M5.1
Tubo izq en recepcion	Bool	%I20.4	Tag_19	Bool	%M5.2
Tubo izquierdo	Bool	%I20.5	Tag_20	Bool	%M5.3
Tubo derecho	Bool	%I20.6	Tag_21	Bool	%M5.4
Conjunto en soldadura	Bool	%I20.7	Tag_22	Bool	%M5.5
Fin nuevo cilindro derecha	Bool	%I21.0	Biestable puerta	Bool	%M2.2
Fin manip --> sold	Bool	%I21.1	Permiso giro mesa	Bool	%M2.3
Conjunto salida	Bool	%I21.2	Manipulador_S TART	Bool	%Q34.6
Fin sold --> manip	Bool	%I21.3	Num_Prog_Manip_espejo	Byte	%IB35
Tubo der en recepcion	Bool	%I21.4	Tag_23	Bool	%M5.6
Conjunto FIN	Bool	%I21.5	Tag_24	Bool	%M5.7
Fin Abrir Puerta	Bool	%I21.6	Tag_25	Bool	%M6.0
Fin Cerrar Puerta	Bool	%I21.7	Tag_26	Bool	%M6.1
Cinta_IN_base mediana	Bool	%Q20.0	Tag_27	Bool	%M6.2

NUEVO CILINDRO IZQUIERDA	Bool	%Q20.1	Tag_28	Bool	%M6.3
manip --> soldadura	Bool	%Q20.2	Tag_29	Bool	%M6.4
soldadura --> manip	Bool	%Q20.3	Tag_30	Bool	%M6.5
NUEVO CILINDRO DERECHA	Bool	%Q20.4	Tag_31	Bool	%M6.6
Cinta_OUT_base mediana	Bool	%Q20.5	Tag_32	Bool	%M6.7
Cinta_IN_base pequeña	Bool	%Q20.6	Tag_33	Bool	%M7.0
Cinta_IN_base grande	Bool	%Q20.7	Tag_34	Bool	%M7.1
Cinta_OUT_base pequeña	Bool	%Q21.0	Num_Prog_Sold_espejo	Byte	%IB79
Cinta_OUT_base grande	Bool	%Q21.1	Tag_35	Bool	%M7.2
Abrir Puerta	Bool	%Q21.2	Tag_36	Bool	%M7.3
Cerrar Puerta	Bool	%Q21.3	Tag_37	Bool	%M7.4
HMI_Mesa girando	Bool	%M1.1	Tag_38	Bool	%M7.5
HMI_Faltan Bases	Bool	%M1.2	Tag_39	Bool	%M7.6
HMI_Falta tubo izq	Bool	%M1.3	Tag_40	Bool	%M7.7
HMI_Falta tubo der	Bool	%M1.4	Robot_Manip_READY	Bool	%I34.5
HMI_modos_OP	Bool	%M1.5	Robot_Sold_READY	Bool	%I78.1
MODO_AUTOMATICO	Bool	%M1.6	Soldador_START	Bool	%Q78.0
MODO_MANUAL	Bool	%M1.7	Sold_izq_OK	Bool	%I78.2
Pulsador PASO a PASO	Bool	%I1.1	Sold_der_OK	Bool	%I78.3
Conjunto Soldado	Bool	%M2.0	Retirar conjunto	Bool	%M2.4
Sentido de giro	Bool	%M2.1	HMI_paro emergencia	Bool	%M2.6
Manip_PAUSE	Bool	%Q34.4	Sold_emergencySTOP	Bool	%Q78.1
Manip_emergencySTOP	Bool	%Q34.5	HMI_Retirar Conjunto ON	Bool	%M8.0
Fin_Prog_Manip	Bool	%I34.0	HMI_Colocar Base	Bool	%M8.1
Sold_PAUSE	Bool	%Q78.2	HMI_Colocar Tubo Izquierdo	Bool	%M8.2
Fin_Prog_Sold	Bool	%I78.0	HMI_Colocar Tubo Derecho	Bool	%M8.3
			HMI_Soldadura Izquierda	Bool	%M8.5
			HMI_Soldadura Derecha	Bool	%M8.6

ANEXO II: CÓDIGO DE CONTROL DEL PLC

Main [OB1]

Main Propiedades					
General					
Nombre	Main	Número	1	Tipo	OB
Idioma	AWL	Numeración	Automático		
Información					
Título	"Main Program Sweep (Cycle)"	Autor		Comentario	
Familia		Versión	0.1	ID personalizado	

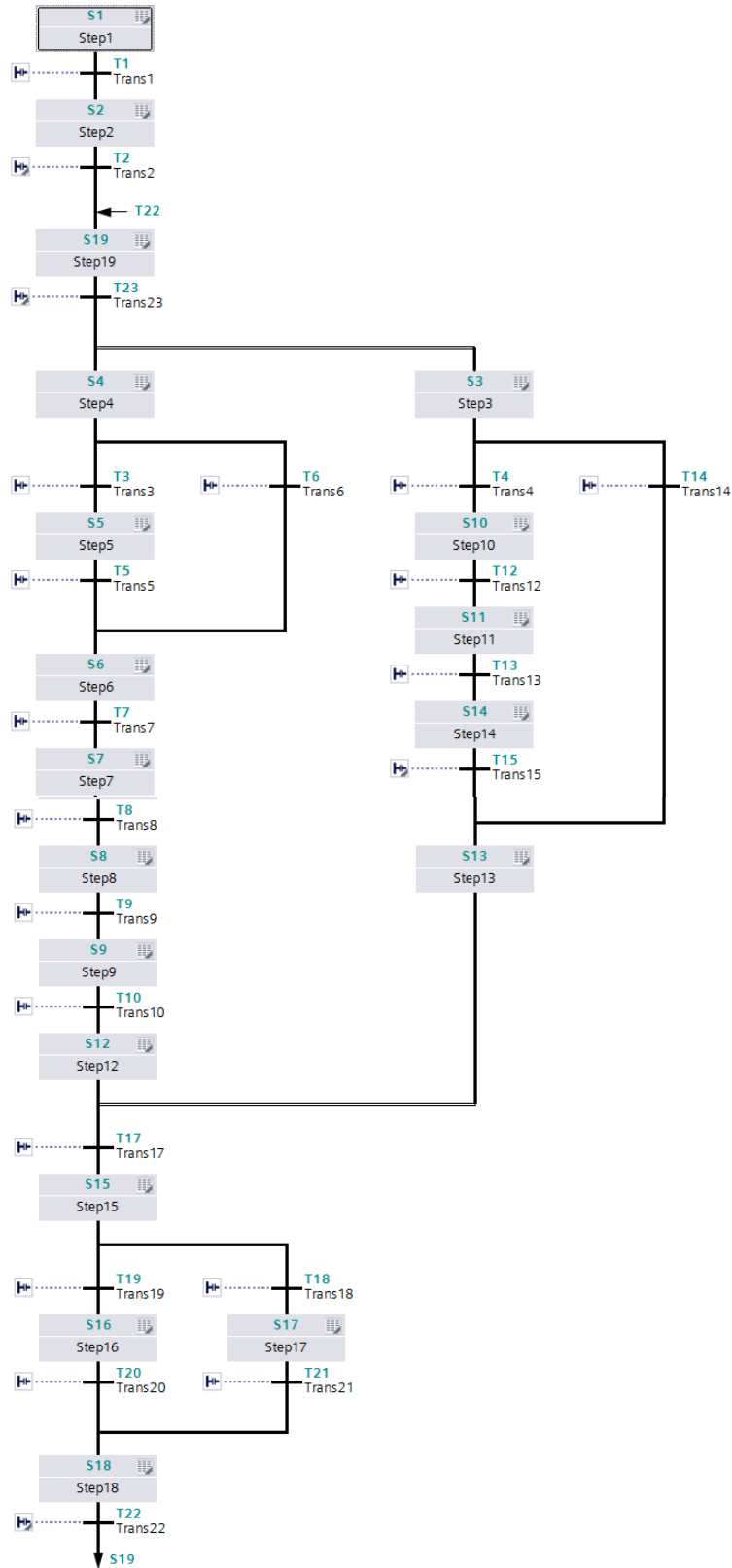
Nombre	Tipo de datos	Valor predet.	Comentario
▼ Input			
Initial_Call	Bool		Initial call of this OB
Remanence	Bool		=True, if remanent data are available
Temp			
Constant			

Segmento 1:

```

0001 CALL "FC Principal"
0002 CALL "Recepcion y salida de piezas"
0003 CALL "Logica Puerta de Seguridad y formas de mando"
0004 CALL "Alarmas HMI1"
  
```

Secuencia GRAF [FB2]



FC Principal [FC4]

FC Principal Propiedades					
General					
Nombre	FC Principal	Número	4	Tipo	FC
Idioma	AWL	Numeración	Automático		
Información					
Título		Autor		Comentario	
Familia		Versión	0.1	ID personalizado	
Nombre	Tipo de datos	Valor predet.	Comentario		
Input					
Output					
InOut					
Temp					
Constant					
▼ Return					
FC Principal	Void				

Segmento 1:

```

0001      CALL "Secuencia GRAF", "Secuencia GRAF_DB"
0002          OFF_SQ                :=
0003          INIT_SQ                :=
0004          ACK_EF                  :=
0005          S_PREV                  :=
0006          S_NEXT                  :=
0007          SW_AUTO                 := "MODO_AUTOMATICO"
0008          SW_TAP                  := "MODO_MANUAL"
0009          SW_TOP                   :=
0010          SW_MAN                   :=
0011          S_SEL                    :=
0012          S_ON                     :=
0013          S_OFF                    :=
0014          T_PUSH                   := "Pulsador PASO a PASO"
0015          S_NO                     :=
0016          S_MORE                   :=
0017          S_ACTIVE                 :=
0018          ERR_FLT                  :=
0019          AUTO_ON                  :=
0020          TAP_ON                   :=
0021          TOP_ON                   :=
0022          MAN_ON                   :=
0023          Start                    := "Start"
0024          HMI_Start                 := "HMI_Start"
0025          Base en mesa              := "Base en mesa"
0026          Base en recepcion          := "Base en Recepcion"
0027          Conjunto salida            := "Conjunto salida"
0028          Conjunto soldado           := "Conjunto Soldado"
0029          Fin programa Manip         := "Fin_Prog_Manip"
0030          Fin programa Soldador      := "Fin_Prog_Sold"
0031          Tubo izq Recep              := "Tubo izq en recepcion"
0032          Tubo der Recep              := "Tubo der en recepcion"
0033          Tubo izq Mesa               := "Tubo izquierdo"
  
```

```

0034      Tubo der Mesa           := "Tubo derecho"
0035      Conjunto en Soldadura   := "Conjunto en soldadura"
0036      Sentido de giro         := "Sentido de giro"
0037      Fin Soldadura --> Manip := "Fin sold --> manip"
0038      Fin Manip --> Soldadura := "Fin manip --> sold"
0039      Manipulador_READY       := "Robot_Manip_READY"
0040      Soldador_READY          := "Robot_Sold_READY"
0041      Soldadura_izq          := "Sold_izq_OK"
0042      Soldadura_der          := "Sold_der_OK"
0043      Retirar conjunto       := "Retirar conjunto"
  
```

Segmento 2:

```

0001      A      "Secuencia GRAF_DB".Step2.X
0002      FP     "Tag_1"
0003      =     "Tag_2"
0004      AN     "Tag_2"
0005      JC     RETIRAR_CONJUNTO
0006      L     1
0007      T     "Num_Prog_Manipulador"
0008      T     "Num_Prog_Soldador"
0009      SET
0010      R     "Sentido de giro"
0011      R     "Permiso giro mesa"
0012      R     "Retirar conjunto"
0013      S     "Celula en funcionamiento"
0014      S     "Manipulador_START"
0015      S     "Soldador_START"
0016
0017      RETIRAR_CONJUNTO : A "Secuencia GRAF_DB".Step5.X
0018      FP     "Tag_3"
0019      =     "Tag_4"
0020      AN     "Tag_4"
0021      JC     AUX_0
0022      L     5
0023      T     "Num_Prog_Manipulador"
0024      SET
0025      R     "Retirar conjunto"
0026      S     "HMI_Retirar Conjunto ON"
0027
0028      AUX_0: L "Num_Prog_Manip_espejo"
0029      L     5
0030      ==I
0031      =     "Tag_26"
0032      A     "Tag_26"
0033      FP     "Tag_27"
0034      =     "Tag_28"
0035      AN     "Tag_28"
0036      JC     COLOCAR_BASE
0037      L     1
0038      T     "Num_Prog_Manipulador"
0039
0040      COLOCAR_BASE : A "Secuencia GRAF_DB".Step7.X
0041      FP     "Tag_5"
0042      =     "Tag_6"
0043      AN     "Tag_6"
0044      JC     AUX_1
0045      L     2
  
```




```
0046 T "Num_Prog_Manipulador"
0047 SET
0048 R "HMI_Retirar Conjunto ON"
0049 S "HMI_Colocar Base"
0050
0051 AUX_1: L "Num_Prog_Manip_espejo"
0052 L 2
0053 ==I
0054 = "Tag_23"
0055 A "Tag_23"
0056 FP "Tag_24"
0057 = "Tag_25"
0058 AN "Tag_25"
0059 JC COLOCAR_TUBO_IZQ
0060 L 1
0061 T "Num_Prog_Manipulador"
0062
0063 COLOCAR_TUBO_IZQ : A "Secuencia GRAF_DB".Step8.X
0064 FP "Tag_7"
0065 = "Tag_8"
0066 AN "Tag_8"
0067 JC AUX_2
0068 L 3
0069 T "Num_Prog_Manipulador"
0070 SET
0071 R "HMI_Colocar Base"
0072 S "HMI_Colocar Tubo Izquierdo"
0073
0074 AUX_2: L "Num_Prog_Manip_espejo"
0075 L 3
0076 ==I
0077 = "Tag_29"
0078 A "Tag_29"
0079 FP "Tag_30"
0080 = "Tag_31"
0081 AN "Tag_31"
0082 JC COLOCAR_TUBO_DER
0083 L 1
0084 T "Num_Prog_Manipulador"
0085
0086 COLOCAR_TUBO_DER : A "Secuencia GRAF_DB".Step9.X
0087 FP "Tag_9"
0088 = "Tag_10"
0089 AN "Tag_10"
0090 JC AUX_3
0091 L 4
0092 T "Num_Prog_Manipulador"
0093 SET
0094 S "Permiso giro mesa"
0095 R "HMI_Colocar Tubo Izquierdo"
0096 S "HMI_Colocar Tubo Derecho"
0097
0098 AUX_3: L "Num_Prog_Manip_espejo"
0099 L 4
0100 ==I
0101 = "Tag_32"
0102 A "Tag_32"
0103 FP "Tag_33"
0104 = "Tag_34"
```



```
0105      AN      "Tag_34"
0106      JC      SOLDADURA_IZQ
0107      L        1
0108      T        "Num_Prog_Manipulador"
0109
0110 SOLDADURA_IZQ : A "Secuencia GRAF_DB".Step10.X
0111      FP      "Tag_11"
0112      =        "Tag_12"
0113      AN      "Tag_12"
0114      JC      AUX_4
0115      L        2
0116      T        "Num_Prog_Soldador"
0117      SET
0118      S        "HMI_Soldadura Izquierda"
0119
0120 AUX_4: L      "Num_Prog_Sold_espejo"
0121      L        2
0122      ==I
0123      =        "Tag_35"
0124      A        "Tag_35"
0125      FP      "Tag_36"
0126      =        "Tag_37"
0127      AN      "Tag_37"
0128      JC      SOLDADURA_DER
0129      L        1
0130      T        "Num_Prog_Soldador"
0131
0132 SOLDADURA_DER : A "Secuencia GRAF_DB".Step11.X
0133      FP      "Tag_13"
0134      =        "Tag_14"
0135      AN      "Tag_14"
0136      JC      AUX_5
0137      L        3
0138      T        "Num_Prog_Soldador"
0139      SET
0140      S        "Permiso giro mesa"
0141      R        "HMI_Soldadura Izquierda"
0142      S        "HMI_Soldadura Derecha"
0143
0144 AUX_5: L      "Num_Prog_Sold_espejo"
0145      L        3
0146      ==I
0147      =        "Tag_38"
0148      A        "Tag_38"
0149      FP      "Tag_39"
0150      =        "Tag_40"
0151      AN      "Tag_40"
0152      JC      CONJUNTO_SOLDADO
0153      L        1
0154      T        "Num_Prog_Soldador"
0155
0156 CONJUNTO_SOLDADO : A "Secuencia GRAF_DB".Step14.X
0157      FP      "Tag_15"
0158      =        "Tag_16"
0159      AN      "Tag_16"
0160      JC      GIRO_MESA
0161      S        "Conjunto Soldado"
0162      R        "HMI_Soldadura Derecha"
0163
```



```
0164 GIRO_MESA : A "Secuencia GRAF_DB".Step16.X
0165     FP      "Tag_17"
0166     =       "Tag_18"
0167     AN      "Tag_18"
0168     JC      MESA_GIRO
0169     SET
0170     S       "manip --> soldadura"
0171     S       "Sentido de giro"
0172     S       "HMI_Mesa girando"
0173     R       "HMI_Colocar Tubo Derecho"
0174
0175 MESA_GIRO : A "Secuencia GRAF_DB".Step17.X
0176     FP      "Tag_19"
0177     =       "Tag_20"
0178     AN      "Tag_20"
0179     JC      RESETS
0180     SET
0181     S       "soldadura --> manip"
0182     R       "Sentido de giro"
0183     S       "HMI_Mesa girando"
0184     R       "HMI_Colocar Tubo Derecho"
0185
0186 RESETS : A "Secuencia GRAF_DB".Step18.X
0187     FP      "Tag_21"
0188     =       "Tag_22"
0189     AN      "Tag_22"
0190     JC      FIN
0191     R       "manip --> soldadura"
0192     R       "soldadura --> manip"
0193     R       "HMI_Mesa girando"
0194     R       "Permiso giro mesa"
0195     A       "Conjunto Soldado"
0196     S       "Retirar conjunto"
0197     R       "Conjunto Soldado"
0198
0199 FIN:  NOP 0
0200
```

Logica Puerta de Seguridad y formas de mando [FC1]

Logica Puerta de Seguridad y formas de mando Propiedades					
General					
Nombre	Logica Puerta de Seguridad y formas de mando	Número	1	Tipo	FC
Idioma	FUP	Numeración	Automático		
Información					
Título		Autor		Comentario	
Familia		Versión	0.1	ID personalizado	
Nombre	Tipo de datos	Valor predet.	Comentario		
Input					
Output					
InOut					
Temp					
Constant					
▼ Return					
Logica Puerta de Seguridad y formas de mando	Void				

Segmento 1:

Logica Puerta



Segmento 2:

Logica puerta



Segmento 3:

logica puerta



Segmento 4:

Formas de mando

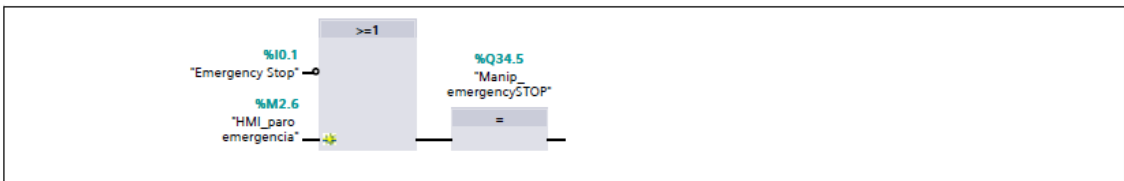


Segmento 5:

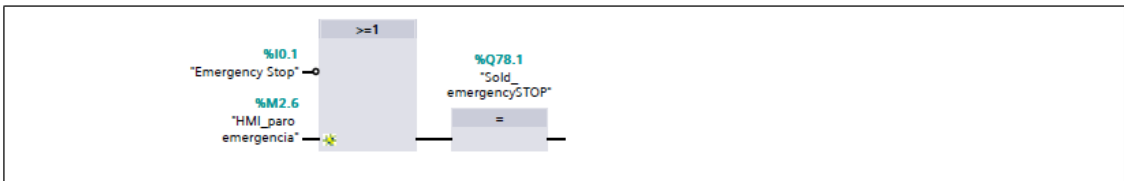
Formas de mando



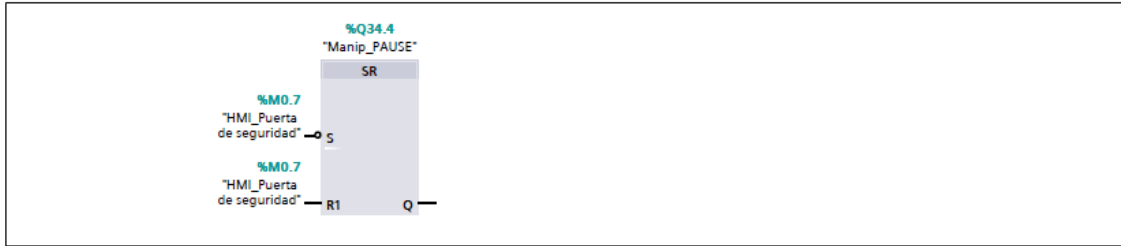
Segmento 6:



Segmento 7:



Segmento 8:



Segmento 9:



Alarmas HMI1 [FC2]

Alarmas HMI1 Propiedades					
General					
Nombre	Alarmas HMI1	Número	2	Tipo	FC
Idioma	FUP	Numeración	Automático		
Información					
Título		Autor		Comentario	
Familia		Versión	0.1	ID personalizado	

Nombre	Tipo de datos	Valor predet.	Comentario
Input			
Output			
InOut			
Temp			
Constant			
▼ Return			
Alarmas HMI1	Void		

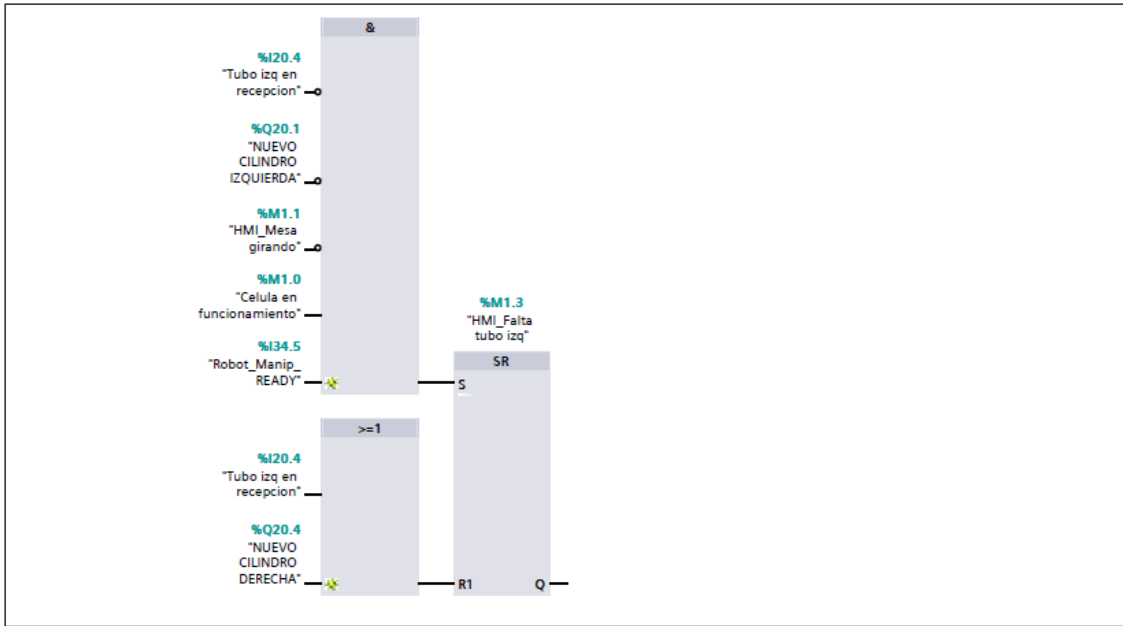
Segmento 1:

Alarma de que no llegan bases a la célula



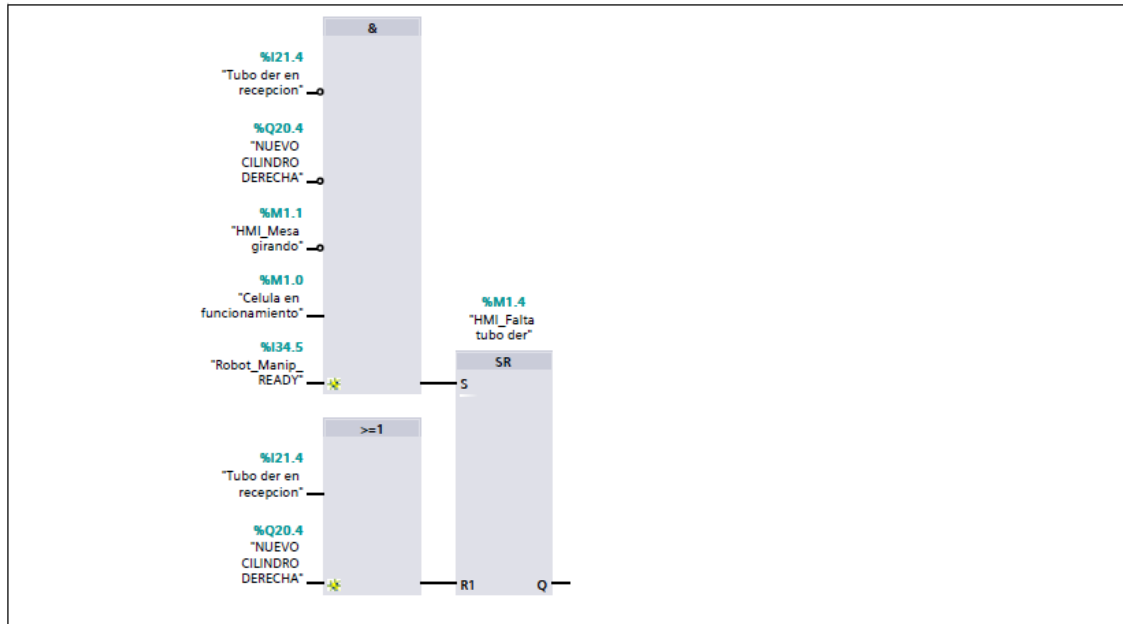
Segmento 2:

Alarma faltan tubos izquierdos en el generador de tubos



Segmento 3:

Alarma faltan tubos derechos en el generador de tubos



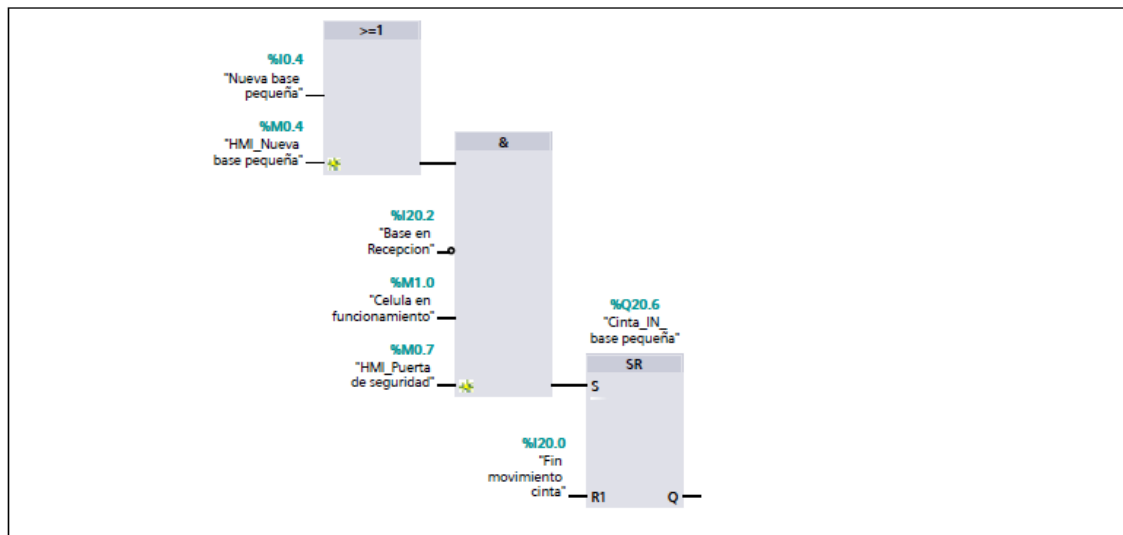
Recepcion y salida de piezas [FC3]

Recepcion y salida de piezas Propiedades					
General					
Nombre	Recepcion y salida de piezas	Número	3	Tipo	FC
Idioma	FUP	Numeración	Automático		
Información					
Título		Autor		Comentario	
Familia		Versión	0.1	ID personalizado	

Nombre	Tipo de datos	Valor predet.	Comentario
Input			
Output			
InOut			
Temp			
Constant			
▼ Return			
Recepcion y salida de piezas	Void		

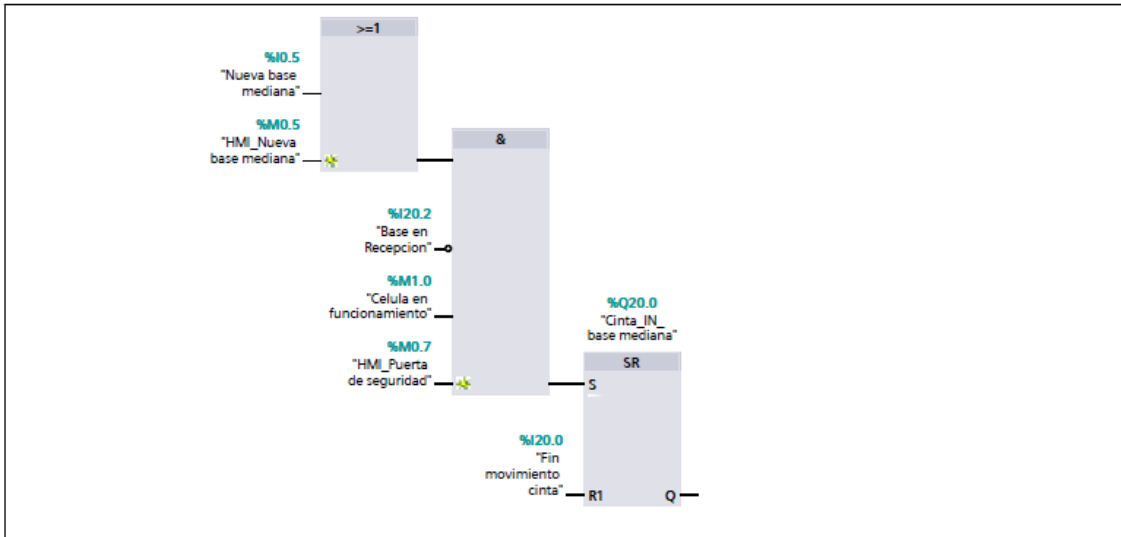
Segmento 1:

Generar base pequeña



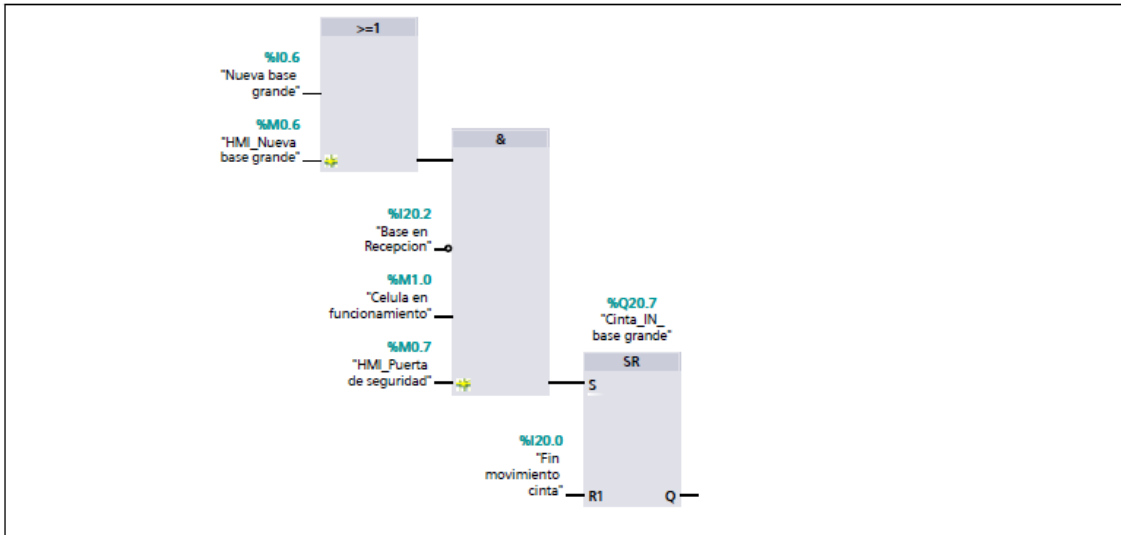
Segmento 2:

Generar base mediana



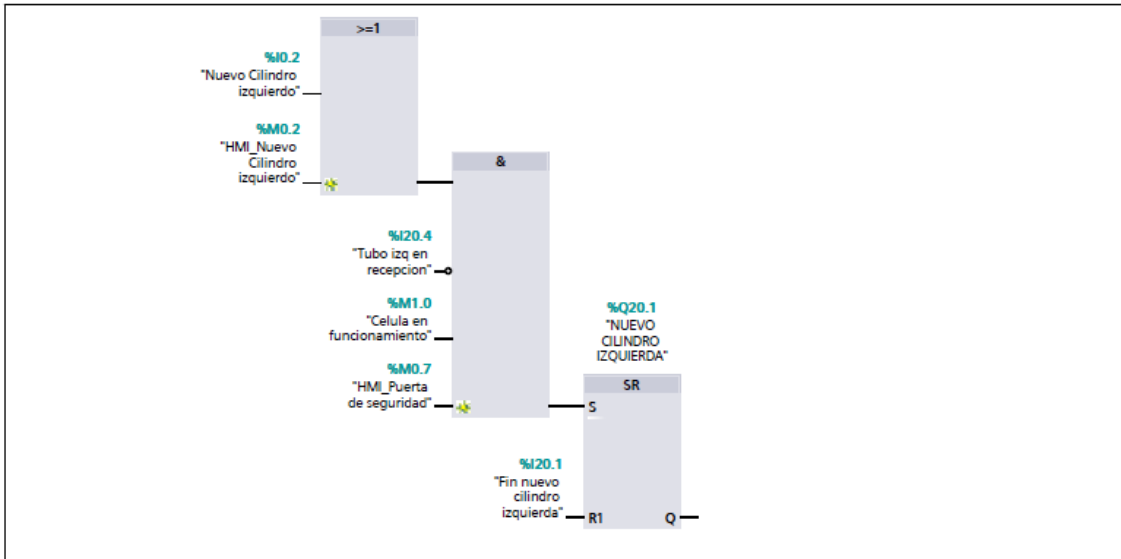
Segmento 3:

Generar base grande



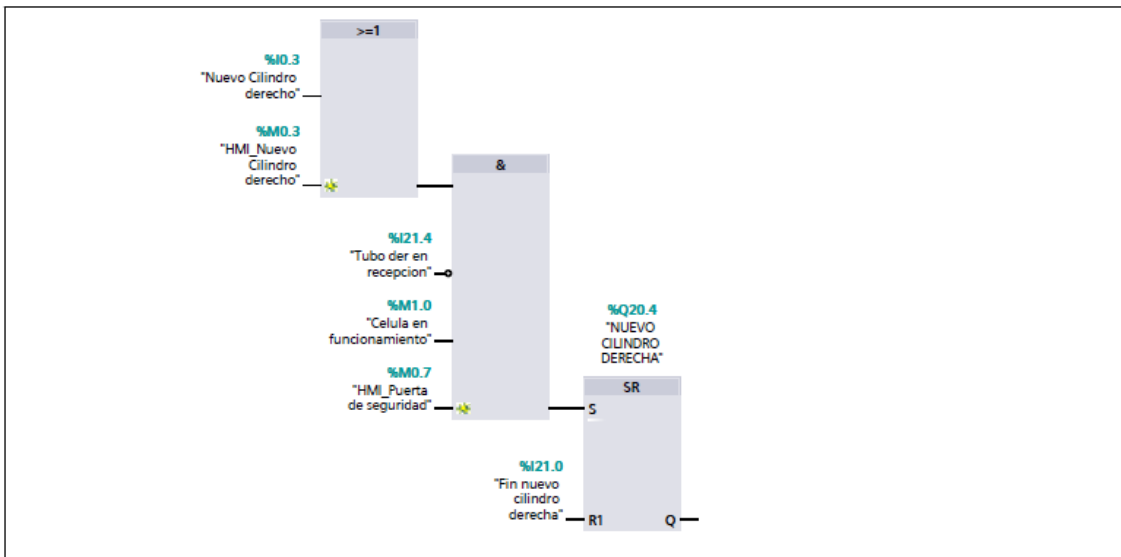
Segmento 4:

Generar nuevo cilindro izquierda



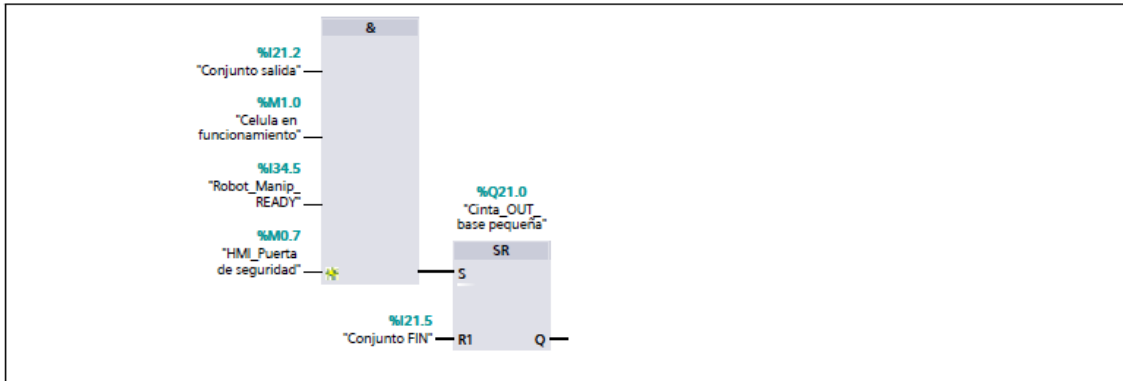
Segmento 5:

Generar nuevo cilindro derecha



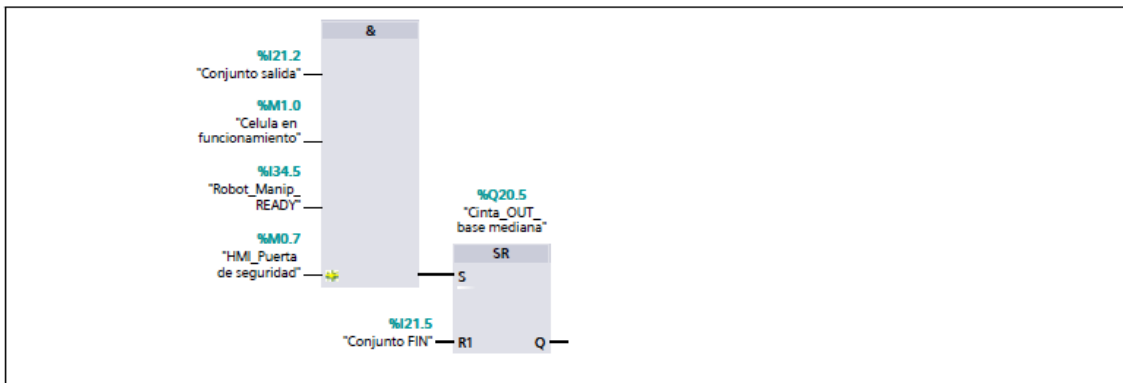
Segmento 6:

Salida conjunto base pequeña



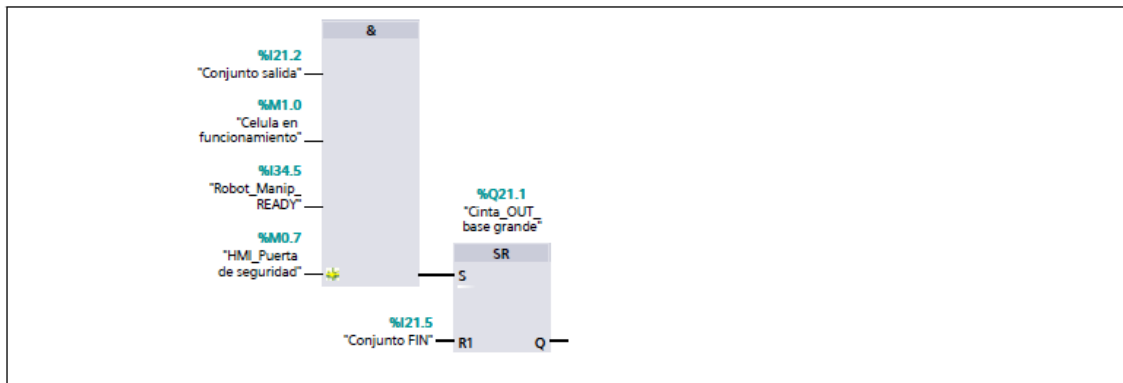
Segmento 7:

Salida conjunto base mediana



Segmento 8:

Salida conjunto base grande



ANEXO III: CÓDIGO DEL ROBOT MANIPULADOR

```
// File: Manipulador_Program.src  
//  
// Generated with: Process Simulate Disconnected 14.1  
//  
// Study: celula  
// Robot: kr30_3  
//  
// Author: User ()  
// Date: 22/05/2019 - 11:09:53  
//  
DATA
```

END_DATA

PROG Manipulador_Program

```
START_OPERATION MAIN_Manipulador  
  // # Send robotReady 1  
  // # While (startProgram) Do  
  // # If (ProgramNumber == 2) Then  
  // # CallPath kr30_3_Base_mesa  
  // # Elsif (ProgramNumber == 3) Then  
  // # CallPath kr30_3_cil_izquierda  
  // # Elsif (ProgramNumber == 4) Then  
  // # CallPath kr30_3_cil_derecha  
  // # Elsif (ProgramNumber == 5) Then  
  // # CallPath kr30_3_PNP_dejar_conjunto  
  // # Endif  
  // # Endwhile  
END_OPERATION
```

```
START_OPERATION kr30_3_Base_mesa  
  Tool Frame = g_bases;  
  [OBJ]RRS_TOOL_FRAME = g_bases;  
  START_MOTION PTP HOME  
  process = HANDLING;  
  loc_type = VIA;  
  x = 1349.999;  
  y = -0.002;  
  z = 1660;  
  rx = 0;  
  ry = 90;  
  rz = 0;  
  j1 = 7.306;  
  j2 = -90.211;  
  j3 = 100.564;  
  j4 = 35.505;  
  j5 = -12.648;  
  j6 = 55.263;
```

```
Speed = 100;
Zone = fine;
Gun Wait = No Wait;
[DOUBLE]RRS_JOINT_SPEED = 100;
[INT]RRS_MOTION_TYPE = 1;
[STRING]RRS_ZONE_NAME = fine;
[INT]SW_GUN_STATE = 4;
[INT]SW_GUN_WAIT = 0;
[STRING]TP_SPEED_TYPE = Joint Speed;
END_MOTION
//# Send robotReady 0
START_MOTION PTP pick
process = HANDLING;
loc_type = PICK;
x = 689.633;
y = -1298.901;
z = -104.876;
rx = 179.634;
ry = 0;
rz = -90;
j1 = 57.079;
j2 = -1.082;
j3 = 76.469;
j4 = 181.203;
j5 = -14.816;
j6 = 56.021;
Speed = 100;
Zone = fine;
Config = OH- J3- J5-, T1:0 T4:0 T6:0;
Gun Wait = No Wait;
[DOUBLE]RRS_JOINT_SPEED = 100;
[INT]RRS_MOTION_TYPE = 1;
[STRING]RRS_ZONE_NAME = fine;
[STRING]TP_SPEED_TYPE = Joint Speed;
END_MOTION
//# Destination garra doble
//# Drive Funcionando
//# Destination garra doble
//# WaitDevice Funcionando
//# Grip g_bases
START_MOTION PTP HOME
process = HANDLING;
loc_type = VIA;
x = 1340;
y = 0;
z = 1809.999;
rx = 0;
ry = 90;
rz = 0;
j1 = 7.368;
j2 = -11.034;
j3 = -69.945;
```



```
j4 = 172.541;
j5 = -81.054;
j6 = -88.728;
Speed = 100;
Zone = fine;
Config = OH- J3+ J5-, T1:0 T4:0 T6:-1;
Gun Wait = No Wait;
[DOUBLE]RRS_JOINT_SPEED = 100;
[INT]RRS_MOTION_TYPE = 1;
[STRING]RRS_ZONE_NAME = fine;
[INT]SW_GUN_STATE = 4;
[INT]SW_GUN_WAIT = 0;
[STRING]TP_SPEED_TYPE = Joint Speed;
END_MOTION
START_MOTION PTP place
process = HANDLING;
loc_type = PLACE;
x = 1741.954;
y = -3.787;
z = 210.249;
rx = 180;
ry = 0;
rz = -90;
j1 = 0.106;
j2 = 33.316;
j3 = -26.188;
j4 = 180;
j5 = -82.872;
j6 = 0.212;
Speed = 100;
Zone = fine;
Config = OH- J3+ J5-, T1:0 T4:0 T6:0;
Gun Wait = No Wait;
[DOUBLE]RRS_JOINT_SPEED = 100;
[INT]RRS_MOTION_TYPE = 1;
[STRING]RRS_ZONE_NAME = fine;
[STRING]TP_SPEED_TYPE = Joint Speed;
END_MOTION
//# Release g_bases
//# Destination garra doble
//# Drive HOME
//# Destination garra doble
//# WaitDevice HOME
//# Attach Mesa
//# Attach Mesa
//# Attach Mesa
START_MOTION PTP HOME
process = HANDLING;
loc_type = VIA;
x = 1349.998;
y = -0.001;
z = 1960.002;
```

```
rx = 90;
ry = -0.106;
rz = 90;
j1 = 0;
j2 = -90;
j3 = 90;
j4 = 0;
j5 = 0;
j6 = 0;
Speed = 100;
Zone = fine;
Config = OH- J3-, T1:0 T4:0 T6:0;
Gun Wait = No Wait;
[DOUBLE]RRS_JOINT_SPEED = 100;
[INT]RRS_MOTION_TYPE = 1;
[STRING]RRS_ZONE_NAME = fine;
[INT]SW_GUN_STATE = 4;
[INT]SW_GUN_WAIT = 0;
[STRING]TP_SPEED_TYPE = Joint Speed;
END_MOTION
//# Send robotReady 1
END_OPERATION

START_OPERATION kr30_3_cil_izquierda
START_MOTION PTP HOME
process = HANDLING;
loc_type = VIA;
x = 1350.019;
y = -0.003;
z = 1959.991;
rx = 90;
ry = -0.106;
rz = 90;
j1 = 0;
j2 = -89.999;
j3 = 90;
j4 = 0;
j5 = -0.001;
j6 = 0;
Speed = 100;
Zone = fine;
Gun Wait = No Wait;
[DOUBLE]RRS_JOINT_SPEED = 100;
[INT]RRS_MOTION_TYPE = 1;
[STRING]RRS_ZONE_NAME = fine;
[INT]SW_GUN_STATE = 4;
[INT]SW_GUN_WAIT = 0;
[STRING]TP_SPEED_TYPE = Joint Speed;
END_MOTION
//# Send robotReady 0
START_MOTION PTP via2
process = HANDLING;
```

```
loc_type = VIA;
x = -461.006;
y = -1031.848;
z = 914.998;
rx = 180;
ry = 0;
rz = -90;
j1 = 106.777;
j2 = -82.338;
j3 = 134.85;
j4 = 0;
j5 = 37.488;
j6 = -73.117;
Speed = 100;
Zone = fine;
Gun Wait = No Wait;
[DOUBLE]RRS_JOINT_SPEED = 100;
[INT]RRS_MOTION_TYPE = 1;
[STRING]RRS_ZONE_NAME = fine;
[INT]SW_GUN_STATE = 4;
[INT]SW_GUN_WAIT = 0;
[STRING]TP_SPEED_TYPE = Joint Speed;
END_MOTION
START_MOTION PTP via1
process = HANDLING;
loc_type = VIA;
x = -461.006;
y = -1231.849;
z = 714.998;
rx = 180;
ry = 0;
rz = -90;
j1 = 104.172;
j2 = -60.783;
j3 = 123.449;
j4 = 0;
j5 = 27.334;
j6 = -75.722;
Speed = 100;
Zone = fine;
Gun Wait = No Wait;
[DOUBLE]RRS_JOINT_SPEED = 100;
[INT]RRS_MOTION_TYPE = 1;
[STRING]RRS_ZONE_NAME = fine;
[INT]SW_GUN_STATE = 4;
[INT]SW_GUN_WAIT = 0;
[STRING]TP_SPEED_TYPE = Joint Speed;
END_MOTION
START_MOTION PTP pick4
process = HANDLING;
loc_type = PICK;
x = -461.006;
```

```
y = -1231.848;  
z = 614.999;  
rx = 180;  
ry = 0;  
rz = -90;  
j1 = 113.08;  
j2 = -58.168;  
j3 = 131.226;  
j4 = 0;  
j5 = 16.942;  
j6 = -156.92;  
Speed = 100;  
Zone = fine;  
Tool Frame = g_cil;  
Gun Wait = No Wait;  
[DOUBLE]RRS_JOINT_SPEED = 100;  
[INT]RRS_MOTION_TYPE = 1;  
[OBJ]RRS_TOOL_FRAME = g_cil;  
[STRING]RRS_ZONE_NAME = fine;  
[STRING]TP_SPEED_TYPE = Joint Speed;  
END_MOTION  
//# Destination garra doble  
//# Drive garra cerrada  
//# Destination garra doble  
//# WaitDevice garra cerrada  
//# Grip g_bases  
START_MOTION PTP via7  
process = HANDLING;  
loc_type = VIA;  
x = -461.006;  
y = -1231.849;  
z = 714.997;  
rx = 180;  
ry = 0;  
rz = -90;  
j1 = 104.173;  
j2 = -60.783;  
j3 = 123.449;  
j4 = 0;  
j5 = 27.334;  
j6 = -75.722;  
Speed = 100;  
Zone = fine;  
Gun Wait = No Wait;  
[DOUBLE]RRS_JOINT_SPEED = 100;  
[INT]RRS_MOTION_TYPE = 1;  
[STRING]RRS_ZONE_NAME = fine;  
[INT]SW_GUN_STATE = 4;  
[INT]SW_GUN_WAIT = 0;  
[STRING]TP_SPEED_TYPE = Joint Speed;  
END_MOTION  
START_MOTION PTP via6
```

```
process = HANDLING;
loc_type = VIA;
x = -461.007;
y = -1031.849;
z = 914.998;
rx = 180;
ry = 0;
rz = -90;
j1 = 106.777;
j2 = -82.338;
j3 = 134.85;
j4 = 0;
j5 = 37.488;
j6 = -73.117;
Speed = 100;
Zone = fine;
Gun Wait = No Wait;
[DOUBLE]RRS_JOINT_SPEED = 100;
[INT]RRS_MOTION_TYPE = 1;
[STRING]RRS_ZONE_NAME = fine;
[INT]SW_GUN_STATE = 4;
[INT]SW_GUN_WAIT = 0;
[STRING]TP_SPEED_TYPE = Joint Speed;
END_MOTION
START_MOTION PTP via9
process = HANDLING;
loc_type = VIA;
x = 338.993;
y = -1031.849;
z = 914.997;
rx = 180;
ry = 0;
rz = -90;
j1 = 64.638;
j2 = -78.529;
j3 = 130.172;
j4 = 0;
j5 = 38.358;
j6 = -115.257;
Speed = 100;
Zone = fine;
Gun Wait = No Wait;
[DOUBLE]RRS_JOINT_SPEED = 100;
[INT]RRS_MOTION_TYPE = 1;
[STRING]RRS_ZONE_NAME = fine;
[INT]SW_GUN_STATE = 4;
[INT]SW_GUN_WAIT = 0;
[STRING]TP_SPEED_TYPE = Joint Speed;
END_MOTION
START_MOTION PTP via8
process = HANDLING;
loc_type = VIA;
```

```
x = 1739.997;  
y = 133.713;  
z = 539.564;  
rx = 180;  
ry = 0;  
rz = -90;  
j1 = -4.055;  
j2 = -19.325;  
j3 = 56.251;  
j4 = 0;  
j5 = 53.074;  
j6 = 176.051;  
Speed = 100;  
Zone = fine;  
Gun Wait = No Wait;  
[DOUBLE]RRS_JOINT_SPEED = 100;  
[INT]RRS_MOTION_TYPE = 1;  
[STRING]RRS_ZONE_NAME = fine;  
[INT]SW_GUN_STATE = 4;  
[INT]SW_GUN_WAIT = 0;  
[STRING]TP_SPEED_TYPE = Joint Speed;  
END_MOTION  
START_MOTION PTP place4  
process = HANDLING;  
loc_type = PLACE;  
x = 1737.997;  
y = 136.068;  
z = 231.083;  
rx = 180;  
ry = 0;  
rz = -90;  
j1 = -9.347;  
j2 = -12.96;  
j3 = 68.549;  
j4 = 0;  
j5 = 34.412;  
j6 = 80.653;  
Speed = 100;  
Zone = fine;  
Tool Frame = g_cil;  
Config = OH- J3- J5+, T1:-1 T4:0 T6:0;  
Gun Wait = No Wait;  
[DOUBLE]RRS_JOINT_SPEED = 100;  
[INT]RRS_MOTION_TYPE = 1;  
[OBJ]RRS_TOOL_FRAME = g_cil;  
[STRING]RRS_ZONE_NAME = fine;  
[STRING]TP_SPEED_TYPE = Joint Speed;  
END_MOTION  
///  
Release g_bases  
Destination garra doble  
Drive HOME  
Destination garra doble
```

```
//# WaitDevice HOME
//# Attach Mesa
START_MOTION PTP HOME
process = HANDLING;
loc_type = VIA;
x = 1340;
y = 0.001;
z = 1809.999;
rx = 0;
ry = 90;
rz = 0;
j1 = 0;
j2 = -90;
j3 = 90;
j4 = 0;
j5 = 0;
j6 = 0;
Speed = 100;
Zone = fine;
Tool Frame = TOOLFRAME;
Config = OH- J3-, T1:-1 T4:0 T6:0;
Gun Wait = No Wait;
[DOUBLE]RRS_JOINT_SPEED = 100;
[INT]RRS_MOTION_TYPE = 1;
[OBJ]RRS_TOOL_FRAME = TOOLFRAME;
[STRING]RRS_ZONE_NAME = fine;
[INT]SW_GUN_STATE = 4;
[INT]SW_GUN_WAIT = 0;
[STRING]TP_SPEED_TYPE = Joint Speed;
END_MOTION
//# Send robotReady 1
END_OPERATION

START_OPERATION kr30_3_cil_derecha
START_MOTION PTP HOME
process = HANDLING;
loc_type = VIA;
x = 1350;
y = -0.003;
z = 1659.999;
rx = 0;
ry = 90;
rz = 0;
j1 = 7.306;
j2 = -90.211;
j3 = 100.564;
j4 = 35.505;
j5 = -12.648;
j6 = 55.263;
Speed = 100;
Zone = fine;
Gun Wait = No Wait;
```



```
[DOUBLE]RRS_JOINT_SPEED = 100;  
[INT]RRS_MOTION_TYPE = 1;  
[STRING]RRS_ZONE_NAME = fine;  
[INT]SW_GUN_STATE = 4;  
[INT]SW_GUN_WAIT = 0;  
[STRING]TP_SPEED_TYPE = Joint Speed;  
END_MOTION  
//# Send robotReady 0  
START_MOTION PTP via3  
process = HANDLING;  
loc_type = VIA;  
x = -226.872;  
y = -1019.428;  
z = 895.664;  
rx = 180;  
ry = 0;  
rz = -90;  
j1 = 94.314;  
j2 = -84.601;  
j3 = 139.312;  
j4 = 0;  
j5 = 35.289;  
j6 = -85.581;  
Speed = 100;  
Zone = fine;  
Gun Wait = No Wait;  
[DOUBLE]RRS_JOINT_SPEED = 100;  
[INT]RRS_MOTION_TYPE = 1;  
[STRING]RRS_ZONE_NAME = fine;  
[INT]SW_GUN_STATE = 4;  
[INT]SW_GUN_WAIT = 0;  
[STRING]TP_SPEED_TYPE = Joint Speed;  
END_MOTION  
START_MOTION PTP via  
process = HANDLING;  
loc_type = VIA;  
x = -759.35;  
y = -1234.859;  
z = 671.589;  
rx = 180;  
ry = 0;  
rz = -90;  
j1 = 116.269;  
j2 = -53.651;  
j3 = 114.774;  
j4 = 0;  
j5 = 28.877;  
j6 = -63.625;  
Speed = 100;  
Zone = fine;  
Gun Wait = No Wait;  
[DOUBLE]RRS_JOINT_SPEED = 100;
```

```
[INT]RRS_MOTION_TYPE = 1;
[STRING]RRS_ZONE_NAME = fine;
[INT]SW_GUN_STATE = 4;
[INT]SW_GUN_WAIT = 0;
[STRING]TP_SPEED_TYPE = Joint Speed;
END_MOTION
START_MOTION PTP pick1
process = HANDLING;
loc_type = PICK;
x = -760.496;
y = -1234.81;
z = 629.344;
rx = 180;
ry = 0;
rz = -90;
j1 = 125.032;
j2 = -53.441;
j3 = 119.23;
j4 = 0;
j5 = 24.211;
j6 = -144.968;
Speed = 100;
Zone = fine;
Tool Frame = g_cil;
Config = OH- J3- J5+, T1:0 T4:0 T6:-1;
Gun Wait = No Wait;
[DOUBLE]RRS_JOINT_SPEED = 100;
[INT]RRS_MOTION_TYPE = 1;
[OBJ]RRS_TOOL_FRAME = g_cil;
[STRING]RRS_ZONE_NAME = fine;
[STRING]TP_SPEED_TYPE = Joint Speed;
END_MOTION
//# Destination garra doble
//# Drive garra cerrada
//# Destination garra doble
//# WaitDevice garra cerrada
//# Grip g_bases
START_MOTION PTP via5
process = HANDLING;
loc_type = VIA;
x = -759.35;
y = -1234.859;
z = 671.588;
rx = 180;
ry = 0;
rz = -90;
j1 = 116.269;
j2 = -53.651;
j3 = 114.774;
j4 = 0;
j5 = 28.877;
j6 = -63.625;
```

```
Speed = 100;
Zone = fine;
Gun Wait = No Wait;
[DOUBLE]RRS_JOINT_SPEED = 100;
[INT]RRS_MOTION_TYPE = 1;
[STRING]RRS_ZONE_NAME = fine;
[INT]SW_GUN_STATE = 4;
[INT]SW_GUN_WAIT = 0;
[STRING]TP_SPEED_TYPE = Joint Speed;
END_MOTION
START_MOTION PTP via4
process = HANDLING;
loc_type = VIA;
x = -226.872;
y = -1019.429;
z = 895.663;
rx = 180;
ry = 0;
rz = -90;
j1 = 94.314;
j2 = -84.601;
j3 = 139.312;
j4 = 0;
j5 = 35.288;
j6 = -85.581;
Speed = 100;
Zone = fine;
Gun Wait = No Wait;
[DOUBLE]RRS_JOINT_SPEED = 100;
[INT]RRS_MOTION_TYPE = 1;
[STRING]RRS_ZONE_NAME = fine;
[INT]SW_GUN_STATE = 4;
[INT]SW_GUN_WAIT = 0;
[STRING]TP_SPEED_TYPE = Joint Speed;
END_MOTION
START_MOTION PTP place1
process = HANDLING;
loc_type = PLACE;
x = 1739.997;
y = -142.055;
z = 244.894;
rx = 180;
ry = 0;
rz = 90;
j1 = 9.528;
j2 = -13.572;
j3 = 68.673;
j4 = 0;
j5 = 34.899;
j6 = -80.472;
Speed = 100;
Zone = fine;
```

```
Tool Frame = g_cil;
Config = OH- J3- J5+, T1:0 T4:0 T6:-1;
Gun Wait = No Wait;
[DOUBLE]RRS_JOINT_SPEED = 100;
[INT]RRS_MOTION_TYPE = 1;
[OBJ]RRS_TOOL_FRAME = g_cil;
[STRING]RRS_ZONE_NAME = fine;
[STRING]TP_SPEED_TYPE = Joint Speed;
END_MOTION
//# Release g_bases
//# Destination garra doble
//# Drive HOME
//# Destination garra doble
//# WaitDevice HOME
//# Attach Mesa
START_MOTION PTP HOME
process = HANDLING;
loc_type = VIA;
x = 1350;
y = -0.003;
z = 1659.999;
rx = 0;
ry = 90;
rz = 0;
j1 = 7.306;
j2 = -90.211;
j3 = 100.564;
j4 = 35.505;
j5 = -12.648;
j6 = 55.263;
Speed = 100;
Zone = fine;
Gun Wait = No Wait;
[DOUBLE]RRS_JOINT_SPEED = 100;
[INT]RRS_MOTION_TYPE = 1;
[STRING]RRS_ZONE_NAME = fine;
[INT]SW_GUN_STATE = 4;
[INT]SW_GUN_WAIT = 0;
[STRING]TP_SPEED_TYPE = Joint Speed;
END_MOTION
//# Send robotReady 1
END_OPERATION

START_OPERATION kr30_3_PNP_dejar_conjunto
Tool Frame = g_bases;
[OBJ]RRS_TOOL_FRAME = g_bases;
//# Attach
//# Attach
//# Attach
//# Attach
//# Attach
//# Attach
```

```
START_MOTION PTP HOME
process = HANDLING;
loc_type = VIA;
x = 1340.045;
y = 0.002;
z = 1809.98;
rx = 0;
ry = 90;
rz = 0;
j1 = 7.368;
j2 = -90.017;
j3 = 89.999;
j4 = 90.139;
j5 = -7.368;
j6 = -0.034;
Speed = 100;
Zone = fine;
Config = OH- J3- J5-, T1:0 T4:0 T6:-1;
Gun Wait = No Wait;
[DOUBLE]RRS_JOINT_SPEED = 100;
[INT]RRS_MOTION_TYPE = 1;
[STRING]RRS_ZONE_NAME = fine;
[INT]SW_GUN_STATE = 4;
[INT]SW_GUN_WAIT = 0;
[STRING]TP_SPEED_TYPE = Joint Speed;
END_MOTION
//# Send robotReady 0
START_MOTION PTP pick3
process = HANDLING;
loc_type = PICK;
x = 1739.997;
y = -3.787;
z = 207.563;
rx = 180;
ry = 0;
rz = 90;
j1 = 0.146;
j2 = -19.3;
j3 = 87.611;
j4 = 0;
j5 = 21.688;
j6 = 0.252;
Speed = 100;
Zone = fine;
Config = OH- J3- J5+, T1:0 T4:0 T6:0;
Gun Wait = No Wait;
[DOUBLE]RRS_JOINT_SPEED = 100;
[INT]RRS_MOTION_TYPE = 1;
[STRING]RRS_ZONE_NAME = fine;
[STRING]TP_SPEED_TYPE = Joint Speed;
END_MOTION
//# Destination garra doble
```

```
//# Drive Funcionando
//# Destination garra doble
//# WaitDevice Funcionando
//# Grip g_bases
START_MOTION PTP HOME
process = HANDLING;
loc_type = VIA;
x = 1340.045;
y = 0.002;
z = 1809.98;
rx = 0;
ry = 90;
rz = 0;
j1 = 7.368;
j2 = -90.017;
j3 = 89.999;
j4 = 90.139;
j5 = -7.368;
j6 = -0.034;
Speed = 100;
Zone = fine;
Config = OH- J3- J5-, T1:0 T4:0 T6:-1;
Gun Wait = No Wait;
[DOUBLE]RRS_JOINT_SPEED = 100;
[INT]RRS_MOTION_TYPE = 1;
[STRING]RRS_ZONE_NAME = fine;
[INT]SW_GUN_STATE = 4;
[INT]SW_GUN_WAIT = 0;
[STRING]TP_SPEED_TYPE = Joint Speed;
END_MOTION
START_MOTION PTP place3
process = HANDLING;
loc_type = PLACE;
x = 658.468;
y = 1363.942;
z = -104.184;
rx = -179.634;
ry = 0;
rz = 90;
j1 = -69.508;
j2 = -3.621;
j3 = 85.638;
j4 = 177.568;
j5 = -8.119;
j6 = -246.995;
Speed = 100;
Zone = fine;
Config = OH- J3- J5-, T1:-1 T4:0 T6:-1;
Gun Wait = No Wait;
[DOUBLE]RRS_JOINT_SPEED = 100;
[INT]RRS_MOTION_TYPE = 1;
[STRING]RRS_ZONE_NAME = fine;
```

```
[STRING]TP_SPEED_TYPE = Joint Speed;
END_MOTION
//# Release g_bases
//# Destination garra doble
//# Drive HOME
//# Destination garra doble
//# WaitDevice HOME
START_MOTION PTP HOME
process = HANDLING;
loc_type = VIA;
x = 1350.016;
y = -0.005;
z = 1959.992;
rx = 90;
ry = -0.106;
rz = 90;
j1 = 0;
j2 = -89.999;
j3 = 90;
j4 = 0;
j5 = 0;
j6 = 0;
Speed = 100;
Zone = fine;
Config = OH- J3-, T1:0 T4:0 T6:0;
Gun Wait = No Wait;
[DOUBLE]RRS_JOINT_SPEED = 100;
[INT]RRS_MOTION_TYPE = 1;
[STRING]RRS_ZONE_NAME = fine;
[INT]SW_GUN_STATE = 4;
[INT]SW_GUN_WAIT = 0;
[STRING]TP_SPEED_TYPE = Joint Speed;
END_MOTION
//# Send robotReady 1
END_OPERATION

END_PROG
```

ANEXO IV: CÓDIGO DEL ROBOT SOLDADOR

```
// File: Soldador_Program.src
//
// Generated with: Process Simulate Disconnected 14.1
//
// Study: celula
// Robot: kr8_r2100_arc_hw
//
// Author: User ()
// Date: 22/05/2019 - 11:10:22
//
DATA

END_DATA

PROG Soldador_Program

START_OPERATION MAIN Soldador
  //# Send robotReady 1
  //# While (startProgram) Do
  //# If (ProgramNumber == 2) Then
  //# CallPath Soldadura_izquierda
  //# ElseIf (ProgramNumber == 3) Then
  //# CallPath Soldadura_derecha
  //# Endif
  //# Endwhile
END_OPERATION

START_OPERATION Soldadura_izquierda
START_MOTION LIN Cont_Robotic_Op_1_Is1
  process = GLUE/DISP;
  loc_type = SEAM_START;
  x = 4865.638;
  y = 1326.211;
  z = 548.491;
  rx = 180;
  ry = 7.947;
  rz = 121.765;
  Zone = fine;
  [BOOL]OLP_CIRCLE_START = True;
  [DOUBLE]RRS_JOINT_SPEED = 100;
  [INT]RRS_MOTION_TYPE = 2;
  [STRING]RRS_ZONE_NAME = fine;
  [STRING]TP_SPEED_TYPE = Joint Speed;
END_MOTION
  //# Send robotReady 0
START_MOTION CIRC Cont_Robotic_Op_1_Is2 Cont_Robotic_Op_1_Is3
  process = GLUE/DISP;
  loc_type = SEAM_MIDDLE;
  x_via = 4864.361;
```



```
y_via = 1327.724;  
z_via = 548.215;  
rx_via = 180;  
ry_via = 7.926;  
rz_via = 132.353;  
x_end = 4862.928;  
y_end = 1329.081;  
z_end = 547.942;  
rx_end = 180;  
ry_end = 7.598;  
rz_end = 142.941;  
Zone = fine;  
[DOUBLE]RRS_JOINT_SPEED = 100;  
[INT]RRS_MOTION_TYPE = 4;  
[STRING]RRS_ZONE_NAME = fine;  
[STRING]TP_SPEED_TYPE = Joint Speed;  
END_MOTION  
START_MOTION LIN Cont_Robotic_Op_1_Is4  
process = GLUE/DISP;  
loc_type = SEAM_MIDDLE;  
x = 4861.865;  
y = 1329.883;  
z = 547.77;  
rx = 180;  
ry = 7.355;  
rz = 142.941;  
Zone = fine;  
[BOOL]OLP_CIRCLE_START = True;  
[DOUBLE]RRS_JOINT_SPEED = 100;  
[INT]RRS_MOTION_TYPE = 2;  
[STRING]RRS_ZONE_NAME = fine;  
[STRING]TP_SPEED_TYPE = Joint Speed;  
END_MOTION  
START_MOTION CIRC Cont_Robotic_Op_1_Is5 Cont_Robotic_Op_1_Is6  
process = GLUE/DISP;  
loc_type = SEAM_MIDDLE;  
x_via = 4858.957;  
y_via = 1331.465;  
z_via = 547.364;  
rx_via = 180;  
ry_via = 6.534;  
rz_via = 153.529;  
x_end = 4855.756;  
y_end = 1332.307;  
z_end = 547.019;  
rx_end = 180;  
ry_end = 5.132;  
rz_end = 174.706;  
Zone = fine;  
[BOOL]OLP_CIRCLE_START = True;  
[DOUBLE]RRS_JOINT_SPEED = 100;  
[INT]RRS_MOTION_TYPE = 4;
```

```
[STRING]RRS_ZONE_NAME = fine;
[STRING]TP_SPEED_TYPE = Joint Speed;
END_MOTION
START_MOTION CIRC Cont_Robotic_Op_1_Is7 Cont_Robotic_Op_1_Is8
process = GLUE/DISP;
loc_type = SEAM_MIDDLE;
x_via = 4851.77;
y_via = 1332.364;
z_via = 546.723;
rx_via = 180;
ry_via = 3.488;
rz_via = -174.706;
x_end = 4849.381;
y_end = 1331.891;
z_end = 546.596;
rx_end = 180;
ry_end = 1.597;
rz_end = -164.118;
Zone = fine;
[DOUBLE]RRS_JOINT_SPEED = 100;
[INT]RRS_MOTION_TYPE = 4;
[STRING]RRS_ZONE_NAME = fine;
[STRING]TP_SPEED_TYPE = Joint Speed;
END_MOTION
START_MOTION LIN Cont_Robotic_Op_1_Is9
process = GLUE/DISP;
loc_type = SEAM_MIDDLE;
x = 4844.438;
y = 1329.429;
z = 546.528;
rx = 180;
ry = -0.133;
rz = -142.941;
Zone = fine;
[BOOL]OLP_CIRCLE_START = True;
[DOUBLE]RRS_JOINT_SPEED = 100;
[INT]RRS_MOTION_TYPE = 2;
[STRING]RRS_ZONE_NAME = fine;
[STRING]TP_SPEED_TYPE = Joint Speed;
END_MOTION
START_MOTION CIRC Cont_Robotic_Op_1_Is10 Cont_Robotic_Op_1_Is11
process = GLUE/DISP;
loc_type = SEAM_MIDDLE;
x_via = 4840.493;
y_via = 1324.962;
z_via = 546.774;
rx_via = 180;
ry_via = -3.658;
rz_via = -121.765;
x_end = 4839.706;
y_end = 1323.382;
z_end = 546.897;
```

```
rx_end = 180;
ry_end = -4.44;
rz_end = -111.176;
Zone = fine;
[BOOL]OLP_CIRCLE_START = True;
[DOUBLE]RRS_JOINT_SPEED = 100;
[INT]RRS_MOTION_TYPE = 4;
[STRING]RRS_ZONE_NAME = fine;
[STRING]TP_SPEED_TYPE = Joint Speed;
END_MOTION
START_MOTION CIRC Cont_Robotic_Op_1_Is12 Cont_Robotic_Op_1_Is13
process = GLUE/DISP;
loc_type = SEAM_MIDDLE;
x_via = 4838.516;
y_via = 1318.019;
z_via = 547.45;
rx_via = 180;
ry_via = -6.786;
rz_via = -90;
x_end = 4838.7;
y_end = 1315.153;
z_end = 547.815;
rx_end = 180;
ry_end = -7.553;
rz_end = -79.412;
Zone = fine;
[BOOL]OLP_CIRCLE_START = True;
[DOUBLE]RRS_JOINT_SPEED = 100;
[INT]RRS_MOTION_TYPE = 4;
[STRING]RRS_ZONE_NAME = fine;
[STRING]TP_SPEED_TYPE = Joint Speed;
END_MOTION
START_MOTION CIRC Cont_Robotic_Op_1_Is14 Cont_Robotic_Op_1_Is15
process = GLUE/DISP;
loc_type = SEAM_END;
x_via = 4839.752;
y_via = 1311.539;
z_via = 548.328;
rx_via = 180;
ry_via = -7.946;
rz_via = -68.824;
x_end = 4840.241;
y_end = 1310.485;
z_end = 548.491;
rx_end = 180;
ry_end = -8.027;
rz_end = -58.235;
Zone = fine;
[DOUBLE]RRS_JOINT_SPEED = 100;
[INT]RRS_MOTION_TYPE = 4;
[STRING]RRS_ZONE_NAME = fine;
[STRING]TP_SPEED_TYPE = Joint Speed;
```

```
END_MOTION
START_MOTION LIN Cont_Robotic_Op1_1_Is1
process = GLUE/DISP;
loc_type = SEAM_START;
x = 4845.938;
y = 1304.577;
z = 546.365;
rx = 180;
ry = 2.917;
rz = -26.471;
Zone = fine;
[DOUBLE]RRS_JOINT_SPEED = 100;
[INT]RRS_MOTION_TYPE = 2;
[STRING]RRS_ZONE_NAME = fine;
[STRING]TP_SPEED_TYPE = Joint Speed;
END_MOTION
START_MOTION LIN Cont_Robotic_Op1_1_Is2
process = GLUE/DISP;
loc_type = SEAM_MIDDLE;
x = 4847.941;
y = 1303.579;
z = 546.251;
rx = 180;
ry = 2.925;
rz = -26.471;
Zone = fine;
[BOOL]OLP_CIRCLE_START = True;
[DOUBLE]RRS_JOINT_SPEED = 100;
[INT]RRS_MOTION_TYPE = 2;
[STRING]RRS_ZONE_NAME = fine;
[STRING]TP_SPEED_TYPE = Joint Speed;
END_MOTION
START_MOTION CIRC Cont_Robotic_Op1_1_Is3 Cont_Robotic_Op1_1_Is4
process = GLUE/DISP;
loc_type = SEAM_MIDDLE;
x_via = 4850.927;
y_via = 1302.754;
z_via = 546.096;
rx_via = 180;
ry_via = 2.806;
rz_via = -5.294;
x_end = 4853.593;
y_end = 1302.533;
z_end = 545.969;
rx_end = 180;
ry_end = 2.611;
rz_end = 5.294;
Zone = fine;
[BOOL]OLP_CIRCLE_START = True;
[DOUBLE]RRS_JOINT_SPEED = 100;
[INT]RRS_MOTION_TYPE = 4;
[STRING]RRS_ZONE_NAME = fine;
```

```
[STRING]TP_SPEED_TYPE = Joint Speed;
END_MOTION
START_MOTION CIRC Cont_Robotic_Op1_1_Is5 Cont_Robotic_Op1_1_Is6
process = GLUE/DISP;
loc_type = SEAM_MIDDLE;
x_via = 4859.176;
y_via = 1303.685;
z_via = 545.739;
rx_via = 180;
ry_via = 1.933;
rz_via = 26.471;
x_end = 4861.953;
y_end = 1305.224;
z_end = 545.646;
rx_end = 180;
ry_end = 1.313;
rz_end = 37.059;
Zone = fine;
[DOUBLE]RRS_JOINT_SPEED = 100;
[INT]RRS_MOTION_TYPE = 4;
[STRING]RRS_ZONE_NAME = fine;
[STRING]TP_SPEED_TYPE = Joint Speed;
END_MOTION
START_MOTION LIN Cont_Robotic_Op1_1_Is7
process = GLUE/DISP;
loc_type = SEAM_MIDDLE;
x = 4863.922;
y = 1306.835;
z = 545.592;
rx = 180;
ry = 1.058;
rz = 47.647;
Zone = fine;
[BOOL]OLP_CIRCLE_START = True;
[DOUBLE]RRS_JOINT_SPEED = 100;
[INT]RRS_MOTION_TYPE = 2;
[STRING]RRS_ZONE_NAME = fine;
[STRING]TP_SPEED_TYPE = Joint Speed;
END_MOTION
START_MOTION CIRC Cont_Robotic_Op1_1_Is8 Cont_Robotic_Op1_1_Is9
process = GLUE/DISP;
loc_type = SEAM_MIDDLE;
x_via = 4865.763;
y_via = 1309.03;
z_via = 545.559;
rx_via = 180;
ry_via = 0.41;
rz_via = 58.235;
x_end = 4867.149;
y_end = 1311.53;
z_end = 545.552;
rx_end = 180;
```

```
ry_end = -0.121;  
rz_end = 68.824;  
Zone = fine;  
[BOOL]OLP_CIRCLE_START = True;  
[DOUBLE]RRS_JOINT_SPEED = 100;  
[INT]RRS_MOTION_TYPE = 4;  
[STRING]RRS_ZONE_NAME = fine;  
[STRING]TP_SPEED_TYPE = Joint Speed;  
END_MOTION  
START_MOTION CIRC Cont_Robotic_Op1_1_Is10 Cont_Robotic_Op1_1_Is11  
process = GLUE/DISP;  
loc_type = SEAM_MIDDLE;  
x_via = 4867.948;  
y_via = 1313.783;  
z_via = 545.568;  
rx_via = 180;  
ry_via = -0.627;  
rz_via = 79.412;  
x_end = 4868.389;  
y_end = 1316.141;  
z_end = 545.604;  
rx_end = 180;  
ry_end = -1.153;  
rz_end = 79.412;  
Zone = fine;  
[BOOL]OLP_CIRCLE_START = True;  
[DOUBLE]RRS_JOINT_SPEED = 100;  
[INT]RRS_MOTION_TYPE = 4;  
[STRING]RRS_ZONE_NAME = fine;  
[STRING]TP_SPEED_TYPE = Joint Speed;  
END_MOTION  
START_MOTION CIRC Cont_Robotic_Op1_1_Is12 Cont_Robotic_Op1_1_Is13  
process = GLUE/DISP;  
loc_type = SEAM_MIDDLE;  
x_via = 4868.388;  
y_via = 1318.052;  
z_via = 545.647;  
rx_via = 180;  
ry_via = -1.398;  
rz_via = 90;  
x_end = 4868.194;  
y_end = 1319.943;  
z_end = 545.696;  
rx_end = 180;  
ry_end = -1.673;  
rz_end = 100.588;  
Zone = fine;  
[BOOL]OLP_CIRCLE_START = True;  
[DOUBLE]RRS_JOINT_SPEED = 100;  
[INT]RRS_MOTION_TYPE = 4;  
[STRING]RRS_ZONE_NAME = fine;  
[STRING]TP_SPEED_TYPE = Joint Speed;
```

```
END_MOTION
START_MOTION CIRC Cont_Robotic_Op1_1_Is14 Cont_Robotic_Op1_1_Is15
process = GLUE/DISP;
loc_type = SEAM_MIDDLE;
x_via = 4867.229;
y_via = 1323.304;
z_via = 545.819;
rx_via = 180;
ry_via = -2.213;
rz_via = 111.176;
x_end = 4866.216;
y_end = 1325.278;
z_end = 545.909;
rx_end = 180;
ry_end = -2.451;
rz_end = 121.765;
Zone = fine;
[BOOL]OLP_CIRCLE_START = True;
[DOUBLE]RRS_JOINT_SPEED = 100;
[INT]RRS_MOTION_TYPE = 4;
[STRING]RRS_ZONE_NAME = fine;
[STRING]TP_SPEED_TYPE = Joint Speed;
END_MOTION
START_MOTION CIRC Cont_Robotic_Op1_1_Is16 Cont_Robotic_Op1_1_Is17
process = GLUE/DISP;
loc_type = SEAM_END;
x_via = 4862.671;
y_via = 1329.275;
z_via = 546.162;
rx_via = 180;
ry_via = -2.864;
rz_via = 142.941;
x_end = 4859.253;
y_end = 1331.317;
z_end = 546.365;
rx_end = 180;
ry_end = -2.934;
rz_end = 153.529;
Zone = fine;
[DOUBLE]RRS_JOINT_SPEED = 100;
[INT]RRS_MOTION_TYPE = 4;
[STRING]RRS_ZONE_NAME = fine;
[STRING]TP_SPEED_TYPE = Joint Speed;
END_MOTION
START_MOTION PTP HOME
process = GLUE/DISP;
loc_type = VIA;
x = 1389.991;
y = 60.62;
z = 1719.998;
rx = -135;
ry = -90;
```

```
rz = 0;
j1 = 0;
j2 = -90;
j3 = 89.999;
j4 = 44.569;
j5 = 0;
j6 = -44.569;
Zone = fine;
Gun Wait = No Wait;
[INT]RRS_MOTION_TYPE = 1;
[STRING]RRS_ZONE_NAME = fine;
[INT]SW_GUN_STATE = 4;
[INT]SW_GUN_WAIT = 0;
[STRING]TP_SPEED_TYPE = Joint Speed;
END_MOTION
//# Send robotReady 1
//# Send Sold_izq_OK 1
END_OPERATION

START_OPERATION Soldadura_derecha
START_MOTION LIN Cont_Robotic_Op_1_Is1
process = GLUE/DISP;
loc_type = SEAM_START;
x = 5139.402;
y = 1310.781;
z = 547.06;
rx = 180;
ry = 2.038;
rz = -58.235;
Zone = fine;
[BOOL]OLP_CIRCLE_START = True;
[DOUBLE]RRS_JOINT_SPEED = 100;
[INT]RRS_MOTION_TYPE = 2;
[STRING]RRS_ZONE_NAME = fine;
[STRING]TP_SPEED_TYPE = Joint Speed;
END_MOTION
//# Send robotReady 0
START_MOTION CIRC Cont_Robotic_Op_1_Is2 Cont_Robotic_Op_1_Is3
process = GLUE/DISP;
loc_type = SEAM_MIDDLE;
x_via = 5143.862;
y_via = 1305.551;
z_via = 546.818;
rx_via = 180;
ry_via = 1.903;
rz_via = -37.059;
x_end = 5148.172;
y_end = 1303.307;
z_end = 546.668;
rx_end = 180;
ry_end = 1.569;
rz_end = -15.882;
```



```
Zone = fine;
[BOOL]OLP_CIRCLE_START = True;
[DOUBLE]RRS_JOINT_SPEED = 100;
[INT]RRS_MOTION_TYPE = 4;
[STRING]RRS_ZONE_NAME = fine;
[STRING]TP_SPEED_TYPE = Joint Speed;
END_MOTION
START_MOTION CIRC Cont_Robotic_Op_1_Is4 Cont_Robotic_Op_1_Is5
process = GLUE/DISP;
loc_type = SEAM_MIDDLE;
x_via = 5159.612;
y_via = 1304.229;
z_via = 546.482;
rx_via = 180;
ry_via = 0.162;
rz_via = 26.471;
x_end = 5165.614;
y_end = 1309.849;
z_end = 546.54;
rx_end = 180;
ry_end = -0.914;
rz_end = 58.235;
Zone = fine;
[BOOL]OLP_CIRCLE_START = True;
[DOUBLE]RRS_JOINT_SPEED = 100;
[INT]RRS_MOTION_TYPE = 4;
[STRING]RRS_ZONE_NAME = fine;
[STRING]TP_SPEED_TYPE = Joint Speed;
END_MOTION
START_MOTION CIRC Cont_Robotic_Op_1_Is6 Cont_Robotic_Op_1_Is7
process = GLUE/DISP;
loc_type = SEAM_MIDDLE;
x_via = 5167.199;
y_via = 1313.352;
z_via = 546.617;
rx_via = 180;
ry_via = -1.39;
rz_via = 68.824;
x_end = 5167.565;
y_end = 1315.24;
z_end = 546.667;
rx_end = 180;
ry_end = -1.566;
rz_end = 79.412;
Zone = fine;
[BOOL]OLP_CIRCLE_START = True;
[DOUBLE]RRS_JOINT_SPEED = 100;
[INT]RRS_MOTION_TYPE = 4;
[STRING]RRS_ZONE_NAME = fine;
[STRING]TP_SPEED_TYPE = Joint Speed;
END_MOTION
START_MOTION CIRC Cont_Robotic_Op_1_Is8 Cont_Robotic_Op_1_Is9
```

```
process = GLUE/DISP;
loc_type = SEAM_MIDDLE;
x_via = 5167.732;
y_via = 1318.599;
z_via = 546.766;
rx_via = 180;
ry_via = -1.806;
rz_via = 90;
x_end = 5167.182;
y_end = 1321.733;
z_end = 546.871;
rx_end = 180;
ry_end = -2.004;
rz_end = 111.176;
Zone = fine;
[DOUBLE]RRS_JOINT_SPEED = 100;
[INT]RRS_MOTION_TYPE = 4;
[STRING]RRS_ZONE_NAME = fine;
[STRING]TP_SPEED_TYPE = Joint Speed;
END_MOTION
START_MOTION LIN Cont_Robotic_Op_1_Is10
process = GLUE/DISP;
loc_type = SEAM_END;
x = 5164.8;
y = 1326.506;
z = 547.06;
rx = 180;
ry = -2.041;
rz = 121.765;
Zone = fine;
[DOUBLE]RRS_JOINT_SPEED = 100;
[INT]RRS_MOTION_TYPE = 2;
[STRING]RRS_ZONE_NAME = fine;
[STRING]TP_SPEED_TYPE = Joint Speed;
END_MOTION
START_MOTION LIN Cont_Robotic_Op1_1_Is1
process = GLUE/DISP;
loc_type = SEAM_START;
x = 5162.755;
y = 1328.716;
z = 547.733;
rx = 180;
ry = 5.412;
rz = 142.941;
Zone = fine;
[BOOL]OLP_CIRCLE_START = True;
[DOUBLE]RRS_JOINT_SPEED = 100;
[INT]RRS_MOTION_TYPE = 2;
[STRING]RRS_ZONE_NAME = fine;
[STRING]TP_SPEED_TYPE = Joint Speed;
END_MOTION
START_MOTION CIRC Cont_Robotic_Op1_1_Is2 Cont_Robotic_Op1_1_Is3
```

```
process = GLUE/DISP;  
loc_type = SEAM_MIDDLE;  
x_via = 5158.81;  
y_via = 1331.211;  
z_via = 547.291;  
rx_via = 180;  
ry_via = 5.289;  
rz_via = 153.529;  
x_end = 5155.48;  
y_end = 1332.271;  
z_end = 546.978;  
rx_end = 180;  
ry_end = 4.934;  
rz_end = 174.706;  
Zone = fine;  
[BOOL]OLP_CIRCLE_START = True;  
[DOUBLE]RRS_JOINT_SPEED = 100;  
[INT]RRS_MOTION_TYPE = 4;  
[STRING]RRS_ZONE_NAME = fine;  
[STRING]TP_SPEED_TYPE = Joint Speed;  
END_MOTION  
START_MOTION CIRC Cont_Robotic_Op1_1_Is4 Cont_Robotic_Op1_1_Is5  
process = GLUE/DISP;  
loc_type = SEAM_MIDDLE;  
x_via = 5152.619;  
y_via = 1332.504;  
z_via = 546.747;  
rx_via = 180;  
ry_via = 4.246;  
rz_via = -174.706;  
x_end = 5150.707;  
y_end = 1332.327;  
z_end = 546.614;  
rx_end = 180;  
ry_end = 3.834;  
rz_end = -174.706;  
Zone = fine;  
[DOUBLE]RRS_JOINT_SPEED = 100;  
[INT]RRS_MOTION_TYPE = 4;  
[STRING]RRS_ZONE_NAME = fine;  
[STRING]TP_SPEED_TYPE = Joint Speed;  
END_MOTION  
START_MOTION LIN Cont_Robotic_Op1_1_Is6  
process = GLUE/DISP;  
loc_type = SEAM_MIDDLE;  
x = 5149.765;  
y = 1332.186;  
z = 546.548;  
rx = 180;  
ry = 3.729;  
rz = -164.118;  
Zone = fine;
```

```
[BOOL]OLP_CIRCLE_START = True;
[DOUBLE]RRS_JOINT_SPEED = 100;
[INT]RRS_MOTION_TYPE = 2;
[STRING]RRS_ZONE_NAME = fine;
[STRING]TP_SPEED_TYPE = Joint Speed;
END_MOTION
START_MOTION CIRC Cont_Robotic_Op1_1_Is7 Cont_Robotic_Op1_1_Is8
process = GLUE/DISP;
loc_type = SEAM_MIDDLE;
x_via = 5145.298;
y_via = 1330.472;
z_via = 546.31;
rx_via = 180;
ry_via = 2.163;
rz_via = -153.529;
x_end = 5143.854;
y_end = 1329.484;
z_end = 546.253;
rx_end = 180;
ry_end = 1.598;
rz_end = -142.941;
Zone = fine;
[BOOL]OLP_CIRCLE_START = True;
[DOUBLE]RRS_JOINT_SPEED = 100;
[INT]RRS_MOTION_TYPE = 4;
[STRING]RRS_ZONE_NAME = fine;
[STRING]TP_SPEED_TYPE = Joint Speed;
END_MOTION
START_MOTION CIRC Cont_Robotic_Op1_1_Is9 Cont_Robotic_Op1_1_Is10
process = GLUE/DISP;
loc_type = SEAM_MIDDLE;
x_via = 5140.687;
y_via = 1326.336;
z_via = 546.194;
rx_via = 180;
ry_via = -0.09;
rz_via = -121.765;
x_end = 5139.232;
y_end = 1323.853;
z_end = 546.226;
rx_end = 180;
ry_end = -1.376;
rz_end = -111.176;
Zone = fine;
[DOUBLE]RRS_JOINT_SPEED = 100;
[INT]RRS_MOTION_TYPE = 4;
[STRING]RRS_ZONE_NAME = fine;
[STRING]TP_SPEED_TYPE = Joint Speed;
END_MOTION
START_MOTION LIN Cont_Robotic_Op1_1_Is11
process = GLUE/DISP;
loc_type = SEAM_MIDDLE;
```

```
x = 5138.365;
y = 1321.606;
z = 546.292;
rx = 180;
ry = -1.803;
rz = -100.588;
Zone = fine;
[BOOL]OLP_CIRCLE_START = True;
[DOUBLE]RRS_JOINT_SPEED = 100;
[INT]RRS_MOTION_TYPE = 2;
[STRING]RRS_ZONE_NAME = fine;
[STRING]TP_SPEED_TYPE = Joint Speed;
END_MOTION
START_MOTION CIRC Cont_Robotic_Op1_1_Is12 Cont_Robotic_Op1_1_Is13
process = GLUE/DISP;
loc_type = SEAM_END;
x_via = 5139.608;
y_via = 1310.448;
z_via = 547.023;
rx_via = 180;
ry_via = -4.938;
rz_via = -58.235;
x_end = 5144.753;
y_end = 1304.878;
z_end = 547.733;
rx_end = 180;
ry_end = -5.443;
rz_end = -37.059;
Zone = fine;
[DOUBLE]RRS_JOINT_SPEED = 100;
[INT]RRS_MOTION_TYPE = 4;
[STRING]RRS_ZONE_NAME = fine;
[STRING]TP_SPEED_TYPE = Joint Speed;
END_MOTION
START_MOTION PTP HOME
process = GLUE/DISP;
loc_type = VIA;
x = 1389.991;
y = 60.62;
z = 1719.998;
rx = -135;
ry = -90;
rz = 0;
j1 = 0;
j2 = -90;
j3 = 89.999;
j4 = 44.569;
j5 = 0;
j6 = -44.569;
Zone = fine;
Gun Wait = No Wait;
[INT]RRS_MOTION_TYPE = 1;
```

```
[STRING]RRS_ZONE_NAME = fine;  
[INT]SW_GUN_STATE = 4;  
[INT]SW_GUN_WAIT = 0;  
[STRING]TP_SPEED_TYPE = Joint Speed;  
END_MOTION  
//# Send robotReady 1  
//# Send Sold_izq_OK 0  
END_OPERATION  
  
END_PROG
```

ANEXO V: PRESUPUESTO DE LA PUESTA EN MARCHA VIRTUAL Y REAL

	coste	Uso proyecto (días)	vida útil (años)	Nº de dispositivos		Tasa Horaria					
				Real	Gemelo Digital	Real	Gemelo Digital				
PLC S-1500	4.100 €	20	10	1	1	0,342 €	0,342 €				
Software TIA Portal V15.1	2.000 €	37	2	1	1	0,833 €	0,833 €				
Software WinCC Advanced	700 €	10	2	1	1	0,292 €	0,292 €				
Runtime WinCC Advanced (512)	450 €	10	2	1	1	0,188 €	0,188 €				
PC sobremesa	1.000 €	148	4	1	1	0,208 €	0,208 €				
Process Simulate V14.1	20.000 €	76	2	0	1	0,000 €	8,333 €				
Controlador Virtual KUKA	5.000 €	36	2	0	1	0,000 €	2,083 €				
Robot KUKA KR30	20.000 €	20	10	1	0	1,667 €	0,000 €				
Robot KUKA kr8 r2100 arc hw	20.000 €	20	10	1	0	1,667 €	0,000 €				
Controlador KUKA KR C4	5.000 €	20	10	1	0	0,417 €	0,000 €				
Garras Robots	60.000 €	16	10	1	0	5,000 €	0,000 €				
Resto elementos	60.000 €	16	10	1	0	5,000 €	0,000 €				
SIMIT Unit	2.500 €	70	10	0	1	0,000 €	0,208 €				
Horas laborales/día	5										

Real					Gemelo digital				
Concepto	Unidad	nº de unidades	Tasa horaria	Coste	Concepto	Unidad	nº de unidades	Tasa horaria	Coste
Horas internas				18.300,00 €	Horas internas				18.300,00 €
Director de proyecto	h	70	50	3500	Director de proyecto	h	70	50	3500
Alumno	h	740	20	14800	Alumno	h	740	20	14800
Amortizaciones				60.963,33 €	Amortizaciones				261,25 €
PLC S-1500	h	100	0,34 €	34,17 €	PLC S-1500	h	100	0,34 €	34,17 €
PC sobremesa	h	740	0,208 €	154,17 €	PC sobremesa	h	740	0,21 €	154,17 €
SIMIT Unit	h	350	0,000 €	0,00 €	SIMIT Unit	h	350	0,21 €	72,92 €
Robot KUKA KR30	h	100	1,667 €	166,67 €	Robot KUKA KR30	h	100	0,00 €	0,00 €
Robot KUKA kr8 r2100 arc hw	h	100	1,667 €	166,67 €	Robot KUKA kr8 r2100 arc hw	h	100	0,00 €	0,00 €
Controlador KUKA KR C4	h	100	0,417 €	41,67 €	Controlador KUKA KR C4	h	100	0,00 €	0,00 €
Garras Robots	h	-	-	60.000,00 €	Garras Robots	h	80	0,00 €	0,00 €
Resto elementos	h	80	5,000 €	400,00 €	Resto elementos	h	80	0,00 €	0,00 €
Software				178,13 €	Software				3.719,79 €
Software TIA Portal V15.1	h	185	0,83 €	154,17 €	Software TIA Portal	h	185	0,83 €	154,17 €
Software WinCC Advanced	h	50	0,29 €	14,58 €	Software WinCC	h	50	0,29 €	14,58 €
Runtime WinCC Advanced (512)	h	50	0,19 €	9,38 €	Runtime WinCC Advanced (512)	h	50	0,19 €	9,38 €
Process Simulate V14.1	h	380	0,00 €	0,00 €	Process Simulate V14.1	h	380	8,33 €	3.166,67 €
Controlador Virtual KUKA	h	180	0,00 €	0,00 €	Controlador Virtual KUKA	h	180	2,08 €	375,00 €
Subtotal				79.441,46 €	Subtotal				22.281,04 €
Costes indirectos	10%			7.944,15 €	Costes indirectos	4%			891,24 €
TOTAL				87.385,60 €	TOTAL				23.172,28 €

ANEXO VI: DIAGRAMA DE GANTT Y DE RED DEL PROYECTO

