

**ESCUELA TÉCNICA SUPERIOR DE INGENIEROS
INDUSTRIALES Y DE TELECOMUNICACIÓN
UNIVERSIDAD DE CANTABRIA**



Trabajo Fin de Grado

**Diseño de célula de fabricación flexible
mediante la utilización de herramientas
software de digitalización 3D**

**(Design of a flexible workcell using 3D
digitalization software)**

Para acceder al Título de
**GRUADO EN INGENIERÍA EN
TECNOLOGÍAS INDUSTRIALES**

Autor: Calvo Ruiz, Rubén

Febrero - 2019

Índice general

1- Introducción	3
2- Contexto: Industria 4.0	5
2.1- Antecedentes y definición de 'Industria 4.0'	5
2.2- Bucle PDP	7
2.3- Ingeniería concurrente y sistema PLM	9
2.4- La cuarta revolución industrial en Europa y España: problemática	10
3- Plataforma 3DExperience	13
3.1- Descripción y justificación de uso	13
3.2- Aspectos iniciales de la herramienta	14
3.3- Aplicaciones de la plataforma	16
3.4- Entidades esenciales de trabajo	19
4- Diseño de una planta robótica con 3DExperience	22
4.1- Planteamiento, función y composición de la planta	22
4.2- Modelado 3D: CATIA Part Design	27
4.3- Posicionamiento de recursos: DELMIA Plant Layout	29
4.4- Preparar la simulación: DELMIA Robot Simulation	30
4.5- Aspectos de programación: DELMIA Robot Programming	52
4.6- Vinculación de las aplicaciones en tiempo real	61
5- Simulación	63
6- Conclusiones	66
7- Anexo: programas robot	67
8- Lista de figuras, lista de tablas, referencias y bibliografía	73

1. Introducción

El presente proyecto comprende el estudio y utilización de una herramienta software de digitalización para diseñar, programar y simular una célula de fabricación flexible robotizada dentro de un sistema de producción PLM. La característica principal de esta herramienta es la capacidad para integrar simultáneamente la información de todos los elementos de vida del producto en sus distintas fases. De esta manera se posibilita el trabajo paralelo del diseño del producto y sus procesos relacionados, aumentando la eficiencia y la calidad de los mismos aplicando el enfoque de la ingeniería concurrente y las ideas de la llamada 'Industria 4.0'.

El software empleado es la plataforma colaborativa 3DExperience, que combina las tecnologías de modelado dinámico en 3D con las técnicas de la automática y la robótica industrial para conseguir la virtualización de los procesos automatizados de producción.

La motivación principal para la realización de este proyecto es la adquisición de un conocimiento más profundo de este tipo de tecnologías novedosas, que considero tienen un fuerte interés tanto académico como profesional. En un mundo cada vez más globalizado y en plena revolución industrial, es un requisito de creciente importancia desarrollar competencias de alto valor añadido y adquirir habilidades de forma continua para adaptarse a la transformación de los procesos de producción. Los conocimientos aportados por el estudio de esta herramienta proporcionan una ayuda incomparable en el posicionamiento y la diferenciación del profesional de ingeniería, cualificándole para desempeñar labores de creciente demanda en el sector industrial y tecnológico.

Ha sido necesario un intenso ejercicio de aprendizaje de las aplicaciones contenidas en la plataforma colaborativa utilizada mediante la práctica autodidacta y con la ayuda de cursos online. Gracias a este esfuerzo se ha conseguido un hábil manejo de estas, sin el cual hubiese sido imposible el desarrollo de este trabajo. De igual manera se ha requerido un trabajo de

investigación considerable para la correcta comprensión de la magnitud de la transformación digital que se está llevando a cabo en el presente y que previsiblemente continuará imparable en el futuro.

En la célula de fabricación que se ha diseñado se emplean dos robots cuya función consiste en soldar dos piezas metálicas que formarán parte de un vehículo militar blindado y posteriormente paletizar el producto obtenido. Además de los robots, se utilizarán otros componentes programables tales como cintras transportadoras y controladores. Estos desempeñarán su función manipulando el producto en un entorno con elementos estáticos como el vallado o la mesa de trabajo.

La programación y simulación de la planta persigue optimizar los recursos disponibles y estudiar su comportamiento en diferentes situaciones, verificando su corrección antes de construirla físicamente. El software empleado debe ser lo más completo posible si se quiere representar fielmente la realidad. Para ello, además de una correcta representación del robot, debe permitir añadir elementos con los que interactúe, controlar la cinemática del robot para impedir movimientos imposibles respetando sus restricciones, tener en cuenta la dinámica en las trayectorias, permitir un buen sistema de renderizado sin sufrir ralentización, contar con sistemas de modelado, diseño y programación propios, calcular el tiempo de simulación... Se analizarán estos aspectos en las diferentes aplicaciones que ofrece 3DExperience para estos fines.

En este documento se recoge el contexto actual que promueve el desarrollo del proyecto en un capítulo dedicado a la Industria 4.0 y la transformación digital. Posteriormente se describe de forma general la herramienta software empleada, 3DExperience, mencionando las características generales más importantes y las ventajas que comporta su uso. A continuación, se detalla el diseño de la célula y se desgranar los pasos en que se divide su creación a través de las apps concretas de robótica y diseño 3D, resaltando sus características. Por último, se consideran detalles de la simulación, buscando extraer resultados y conclusiones de la misma.

2. Contexto: Industria 4.0

2.1. Antecedentes y definición de 'industria 4.0'

La historia de nuestra especie ha estado siempre condicionada por la tecnología y su desarrollo. Nuestra civilización marca esencialmente la era de la producción, en la que hacemos uso de tecnologías (herramientas, técnicas) para crear bienes y servicios. Su avance se relaciona íntimamente con los descubrimientos científicos y el crecimiento económico, de los que se nutre y a los que contribuye de forma recíproca (Kessler, 2006).

No es hasta la llegada de la primera revolución industrial en la segunda mitad del siglo XVIII, momento en que el proceso de transformación de los recursos naturales se sitúa en una escala superior, cuando se empieza a hablar propiamente de la industria. La industria (del latín industria – 'diligencia') puede definirse de la siguiente manera: "Actividad económica y técnica que consiste en transformar las materias primas hasta convertirlas en productos adecuados para satisfacer las necesidades del hombre". (Selecciones del Reader's Digest, 1981). Con el transcurso de la edad contemporánea, la aceleración del progreso tecnológico ha llevado aparejada cambios profundos en la industria, hallándonos inmersos en la actualidad en la llamada industria 4.0 o cuarta revolución industrial.

El nombre 'industria 4.0' se acuñó por primera vez en Alemania en 2011 y es el que se utilizará principalmente en este trabajo; no obstante, se entenderá como sinónimos expresiones tales como 'fábrica inteligente' (Japón), 'transformación digital' (Estados Unidos) o 'Industria del futuro' (otras partes de Europa). Aunque la terminología difiera, tanto el concepto de fondo como las tecnologías y aplicaciones en que se basa son fundamentalmente las mismas. (Deutsche Welle, 2014).

En la figura 1 se puede observar un resumen de la evolución industrial en los dos últimos siglos, a través de las consecutivas revoluciones, y conceptos clave que conforman la industria 4.0.

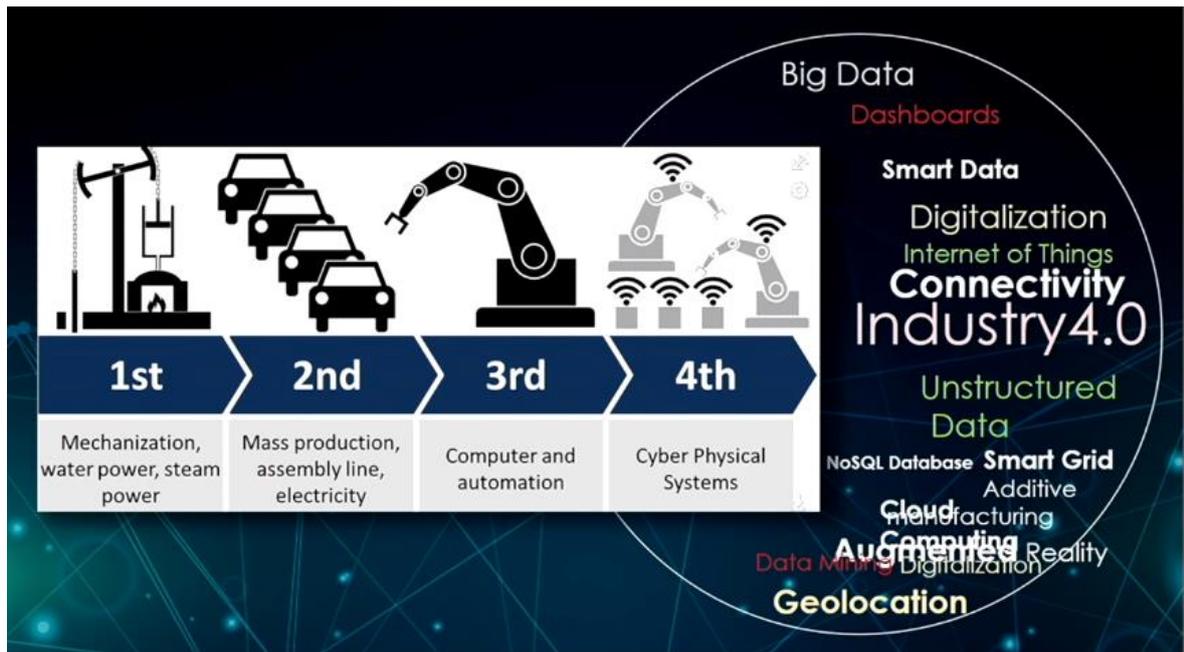


Figura 1: Revoluciones industriales y conceptos propios de industria 4.0 [1]

El término 'industria 4.0' no se circunscribe únicamente a la cadena de suministro y al entorno de las fábricas, como podría inducirse del propio nombre de forma errónea. Tampoco es meramente una revolución digital como suele aludirse en Estados Unidos bajo la expresión 'digital supply network'. El fenómeno engloba tanto el hecho de usar las diversas tecnologías de nueva generación como la forma en que dichas tecnologías se utilizan de forma conjunta y cómo las organizaciones pueden valerse de ellas para crecer y operar. Entender su naturaleza y la magnitud de la transformación que está comportando en el presente y que previsiblemente comportará para el futuro cercano es de vital importancia. Como las anteriores, esta revolución industrial conlleva un cambio de paradigma.

La transformación digital es una realidad que ha sido impulsada en gran medida por los comportamientos de los consumidores, cuyas expectativas han redefinido ya las cadenas de valor de múltiples sectores y reinventado modelos de negocio, desde el turismo hasta el entretenimiento. La conectividad gracias a internet y el acceso digital al cliente son dos de las palancas de cambio más importantes, unidas a la información digital que

se acumula y a la mejora de la automatización, cada vez más autónoma y flexible. Estas palancas se relacionan inequívocamente con el flujo de información del que se hablará a continuación, el PDP. (Berger, 2016)

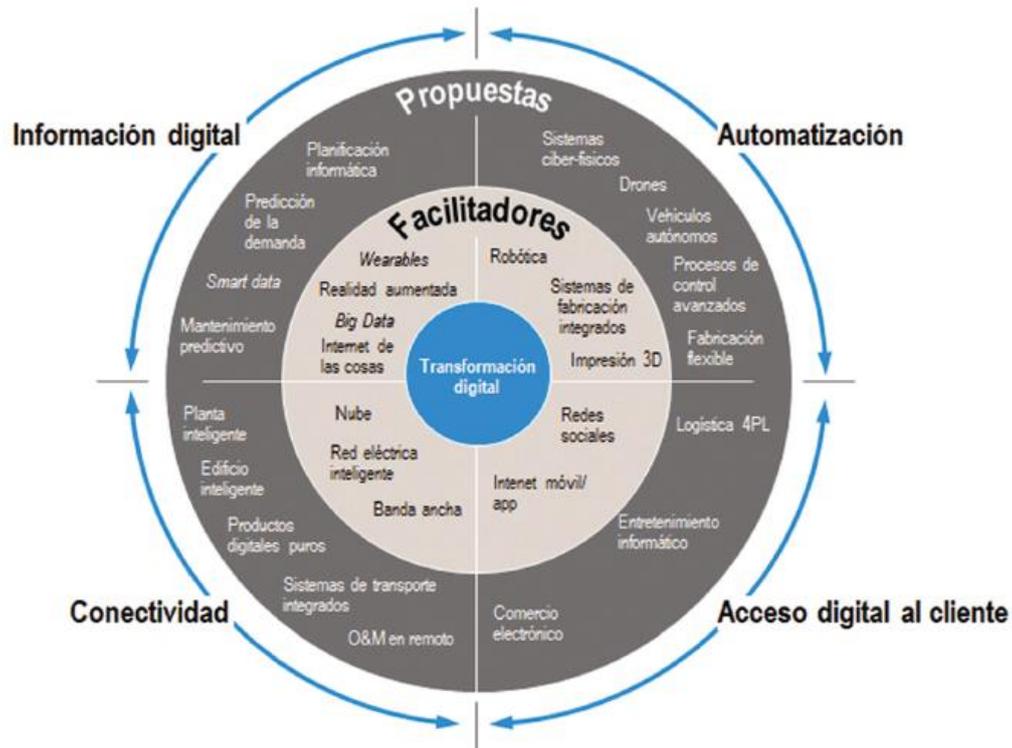


Figura 2: Palancas y facilitadores de la transformación digital. [2]

2.2. Bucle PDP

El rasgo más importante que define la cuarta revolución industrial es el acceso a datos en tiempo real de un flujo continuo de información e interacciones entre el mundo digital y el físico. Este flujo es un ciclo cerrado formado por series iterativas de tres etapas, conocido en inglés como lazo 'physical-to-digital-to-physical' o PDP. La consecución de este proceso no sería posible sin la participación de tecnologías avanzadas en diversos campos tales como análisis de datos, impresión 3D, robótica, computación de alta eficiencia, procesadores de lenguaje, inteligencia artificial y computación cognitiva, ingeniería de materiales o realidad aumentada (Cotteleer, & Sniderman, 2017).

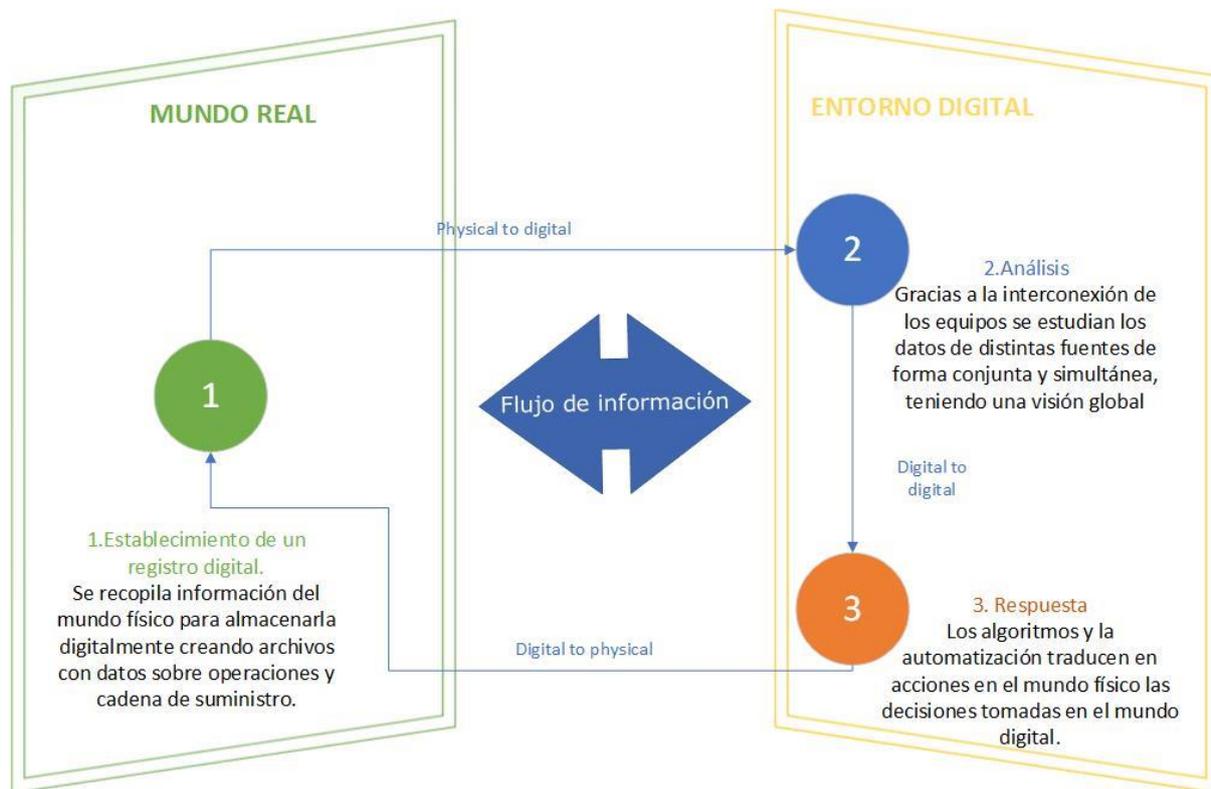


Figura 3: Bucle PDP y tecnologías relacionadas.

Esta característica afecta a todos y cada uno de los pasos del proceso productivo: de qué se hacen las cosas, cómo y dónde se hacen, cómo llegan al consumidor, a dónde van cuando necesitan repararse o cuando el propietario desea deshacerse de ellas. Es decir, afecta a todas las etapas del ciclo de vida del producto. Por esa razón, todos los participantes del ciclo de vida del producto (fabricantes, intermediarios, clientes, consumidores, vendedores, transportistas, empresas, accionistas, administraciones...) verán el rol que juegan modificado de alguna manera. (Laaper, Dollar, Mussomeli, Sniderman, & Mariani, 2017)

Así, la interconexión e intercambio de información de las distintas fases ofrece la oportunidad de actuar en tiempo real para ajustar operaciones de acuerdo con la información que fluye incesantemente. La diferencia sustancial con los procesos monitorizados hasta ahora es que la reacción frente a posibles problemas o necesidades de cambio pasa de ser reactiva a proactiva. Con ello, las decisiones que se pueden tomar serán más

inteligentes, al estar más informadas, mejorando la calidad de los productos, servicios y sistemas. De igual manera, los recursos pueden ser empleados más eficientemente. Incluso se puede prever necesidades futuras con un correcto análisis de datos; esto es, anticipar la demanda.

2.3. Ingeniería concurrente y sistema PLM

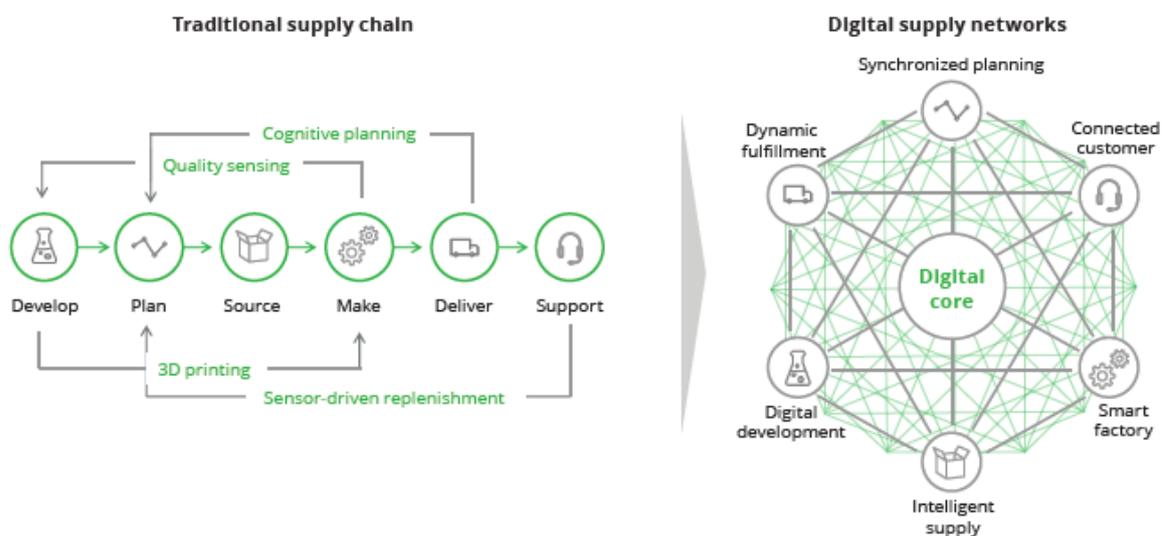


Figura 4: Cambio de la cadena de suministros tradicional a la digital [3]

En el ámbito de la ingeniería desde hace tiempo se viene desarrollando una concepción muy similar a esta nueva forma de producción: la ingeniería concurrente. Una de las definiciones comúnmente aceptadas es la siguiente: 'La ingeniería concurrente es un enfoque sistemático al diseño integrado, paralelo de productos y sus procesos relacionados, incluyendo su fabricación y soporte. Dicho enfoque busca considerar desde el principio todos los elementos del ciclo de vida del producto, desde su concepción a su eliminación, incluyendo su calidad, coste, tiempo y requerimientos del usuario' (Winner, Pennell, Bertrand, & Slusarczyk, 1991). Los avances tecnológicos vistos potencian su práctica.

Se ha referido al ciclo de vida completo del producto y cómo actualmente la nueva perspectiva que surge comprende y se ocupa de todas sus fases. Esta nueva forma de gestión denominada PLM (Product Lifecycle Management) integra profesionales, datos, procesos y sistemas de negocio para tener la información necesaria con la que acometer la optimización de la producción. Los beneficios de los PLM están extensamente documentados (Day, 2002) (Hill, 2006) habiendo sido fortalecidos con los últimos avances tecnológicos. Atendiendo a la definición de industria 4.0 señalada al principio del capítulo, se puede observar que es un buen ejemplo para ilustrar esta revolución. En la imagen siguiente se resumen las fases principales del ciclo de vida del producto. En general, el PLM hace referencia al ciclo de vida físico, siendo el económico denominado 'product life-cycle management (marketing)' (PLCM).

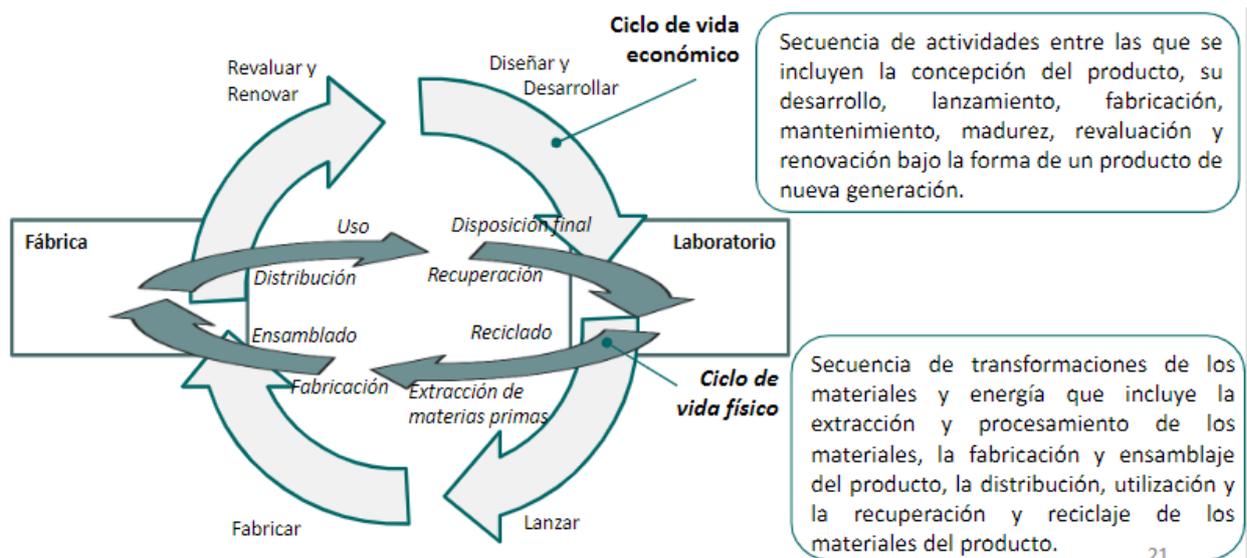


Figura 5: Ciclo de vida de un producto [4]

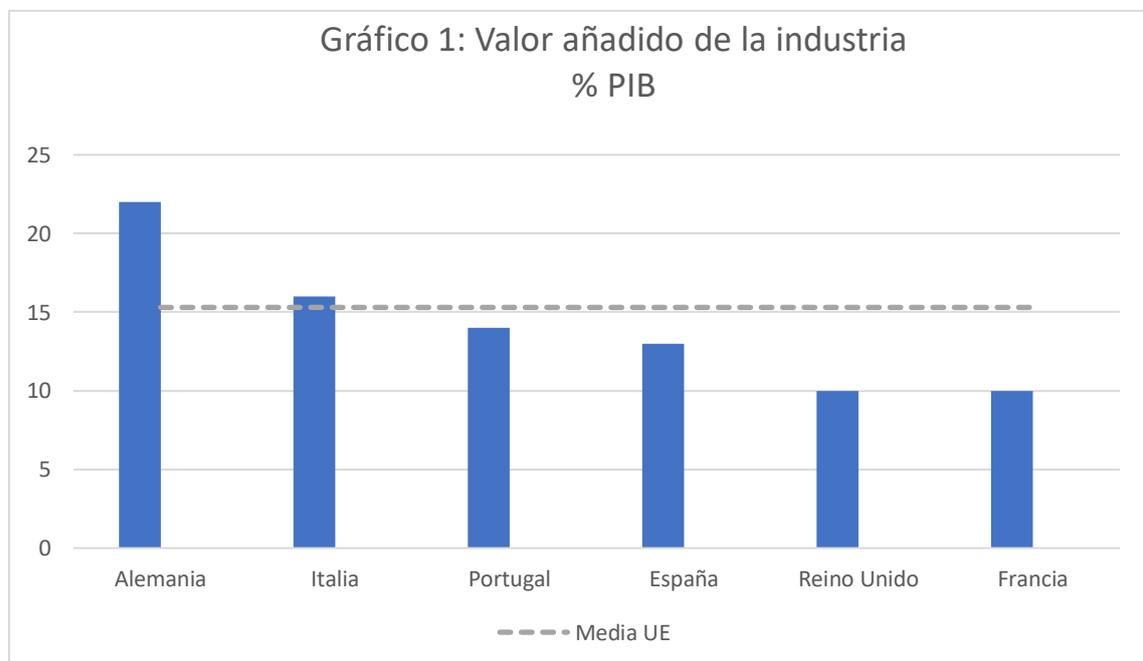
2.4. La cuarta revolución industrial en Europa y España: problemática

La cuarta revolución industrial se originó principalmente en el ámbito de la empresa privada, con empresas pioneras desarrollando soluciones innovadoras a problemas cada vez más complejos. No obstante, en algunos

países europeos los gobiernos tomaron la iniciativa para apoyar la transformación digital. Estos impulsos nacionales han cristalizado en una corriente de respaldo a nivel europeo. De esta forma, la Comisión Europea impulsará inversiones público-privadas en todos los sectores industriales y tratará de ayudar a la transformación digital creando un marco jurídico seguro. Los intereses son principalmente dos: desarrollar un mercado digital único entre todos los miembros de la Unión Europea para adoptar una regulación conjunta y recuperar una mayor participación de la industria en el PIB. Las ventajas de alcanzar tales objetivos son múltiples:

- Aumento de la eficiencia con un ahorro calculado en 415.000 millones de euros al año.
- Simplificación burocrática a nivel europeo.
- Creación de cientos de miles de nuevos trabajos con alto valor añadido.
- Aumento de la competencia en el mercado global.

(Gabriel, 2017)



Fuente: Elaboración propia a partir de los datos de Eurostat (2017)

A pesar de todas las bondades que ofrece esta revolución, es necesario tener presente que cualquier cambio tiene retos inherentes. Algunos de ellos son los siguientes:

- Se incrementan los problemas de seguridad de datos al integrar más acceso a los nuevos sistemas.
- Se necesita una comunicación cibernética fiable y estable.
- Existe incertidumbre sobre la posible pérdida de empleos debido a las nuevas automatizaciones.
- En contraposición, puede que no haya suficiente obra de mano cualificada y experta para crear e implementar estos sistemas.
- Un problema técnico podría desencadenar un fallo de producción con altísimo coste, por lo que evitarlo es una preocupación constante.

(Marr, 2016)

Los esfuerzos de las administraciones se enfocan fundamentalmente en ayudar a solucionar estos problemas. A nivel nacional, España todavía tiene un largo recorrido para alcanzar a los países punteros, pues si bien las empresas líderes muestran una madurez digital comparable a sus competidores internacionales, hay sectores en los que aún no se ha extendido la digitalización. Buscando mejorar estos indicadores surgen las iniciativas gubernamentales 'Industria Conectada 4.0', 'Agenda Digital' y 'Agenda para el fortalecimiento del Sector Industrial' aprobadas por el Consejo de Ministros en 2014. (Ministerio de industria, Comercio y Turismo)

3. Plataforma 3DEXperience

3.1. Descripción y justificación de su uso

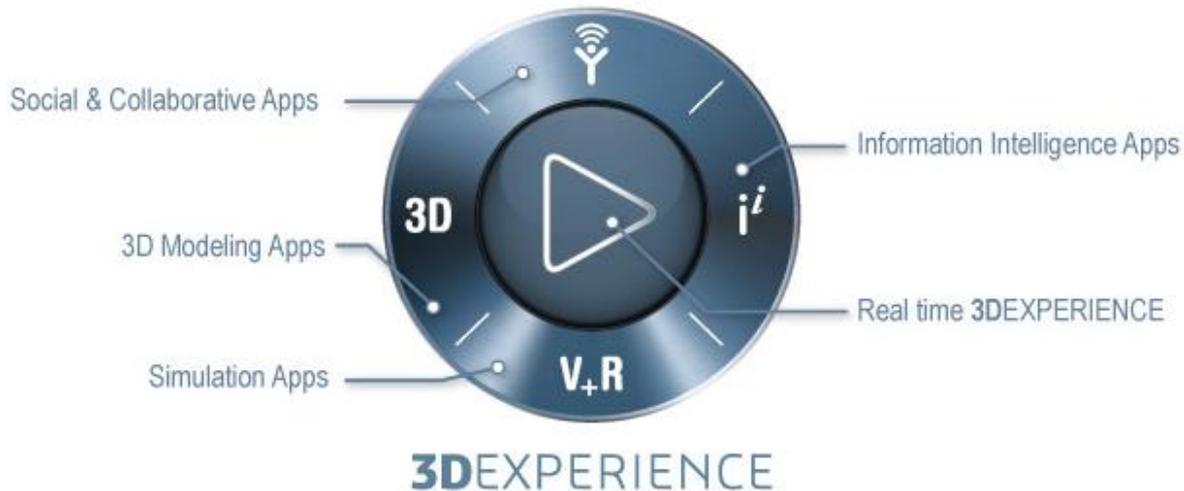


Figura 6: 'Compass' con acceso a las aplicaciones de 3DEXperience [5]

La plataforma colaborativa 3DEXperience de Dassault Systèmes es una potente solución PLM. Se puede considerar como un sistema de información que actúa de eje vertebrador en la gestión del proceso de diseño y desarrollo del producto, teniendo en cuenta todos los aspectos del ciclo de vida del producto. Lo consigue integrando los recursos y actores de dicho ciclo, de manera que dentro de este espacio de colaboración los distintos equipos actores -oficina de diseño, marketing, ventas, clientes, fabricación, logística, subcontratistas, servicios, mantenimiento, reciclaje...- comparten los datos de forma transparente en tiempo real.

El uso de esta herramienta tiene ventajas tangibles, entre las que se pueden destacar:

- La gestión de programas y la planificación de productos facilitan la toma de decisiones en tiempo real.
- Se disminuye al mínimo la posibilidad de problemas en la cadena de suministro durante el proceso de desarrollo del producto gracias a la colaboración con proveedores estratégicos.

- Se puede gestionar desde equipos remotos, garantizando simultaneidad en los datos y que estos estén en una única versión al poder trabajarse desde la nube.
- Conocer el impacto y coste de los cambios de diseño permite obtener mejores productos, adecuados a todo el proceso.

En definitiva, se aumenta la eficiencia global reduciendo tiempo y costes. De manera paralela, incrementa la productividad al identificar y resolver posibles contratiempos con rapidez y eficacia o ayudando a la innovación mediante la colaboración de los distintos departamentos involucrados (ingeniería, simulación, finanzas, etc.)

Las diversas funcionalidades que permite encajan con los modelos anteriormente citados y explicados, pues se incorpora la filosofía de la industria 4.0 y la ingeniería concurrente.

3.2. Aspectos iniciales de la herramienta

En la imagen inferior se observa la interfaz de inicio de la plataforma a la que nos redirige la página tras iniciar sesión. Desde aquí se accede a todas las aplicaciones mediante el gadget llamado *Compass* (esquina superior izquierda). De igual manera, se pueden realizar búsquedas unificadas para acceder a todos los contenidos de la plataforma (cuadro de búsqueda superior). En la esquina superior derecha se hallan diversas opciones de preferencias, creación y colaboración, que son las que conectan a los usuarios y permiten compartir proyectos, programas, métricas, resultados... (flecha *share*). Aparte de un repositorio conjunto propio de Dassault, existe la posibilidad de utilizar servicios de alojamiento de archivos en nube como Dropbox o Google Drive conectando con ellos desde la propia plataforma. Dentro de las distintas comunidades en las que se pueden intercambiar archivos existe un chat integrado, con opción de videollamada, para organizar y revisar conjuntamente el contenido.

Además, hay una función de notificaciones para mantenerse al corriente de cambios o actualizaciones (símbolo de campana).

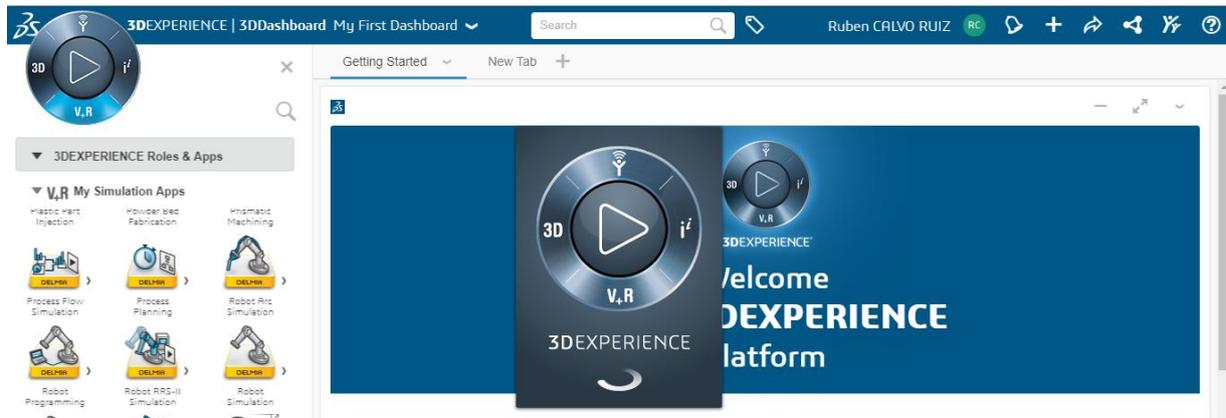


Figura 7: Página de inicio de 3DExperience con acceso a las aplicaciones.

En general se utiliza un formato denominado 'escritorio digital' o *Dashboard* como base para crear contenido y compartirlo. Este escritorio digital es un estilo de pagina web personalizable que permite recopilar información de diversos orígenes o reutilizarlo desde 3DExperience. Se asemeja a las llamadas 'entradas' o 'publicaciones' de redes sociales tales como Facebook o sitios web con configuración de tipo blog, razón por la cual el usuario está relativamente familiarizado.

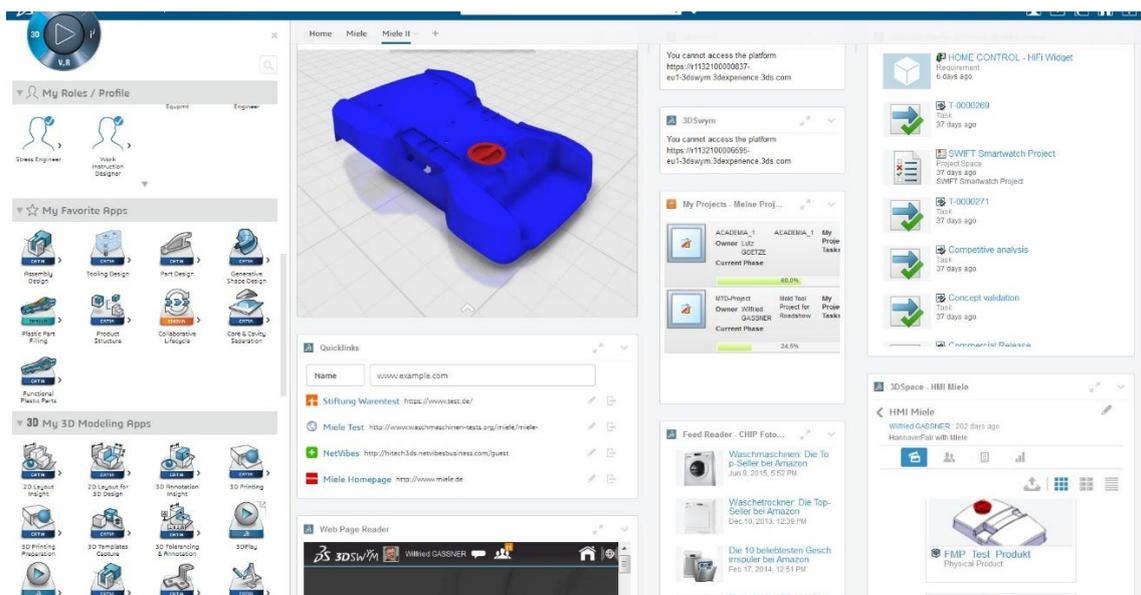


Figura 8: Ejemplo de escritorio digital [6]

La plataforma se compone de una numerosa cantidad de aplicaciones que estarán disponibles según la categoría del usuario. La identificación básica es el rol (imagen siguiente), que se constituiría de gente que comparte tipo o área de trabajo como puede ser diseñador mecánico o analista de simulaciones. De esta forma, las aplicaciones que cada trabajador debe usar se vinculan a su función facilitando la administración de aplicaciones.



Figura 9: Ejemplos de roles, activado el primero y desactivados los demás

Para el desarrollo de este proyecto se han activado los roles *3DEXPERIENCE Essentials Client* y *Manufacturing and Production*.

3.3. Aplicaciones de la plataforma

Una vez se tiene acceso a la aplicación deseada, habilitada por el rol correspondiente, esta se abre clicando sobre su icono. Hay dos clasificaciones en las que dividir las: según comportamiento y según dominio.

Atendiendo a su comportamiento, se tienen tres clases:

- Aplicaciones de *Dashboard*: agregan un widget al escritorio digital.
- Aplicaciones web: abre una pestaña nueva en el explorador.
- Aplicaciones nativas: necesitan de una instalación local en el equipo desde el que se trabaje, aunque es posible su instalación en la nube. El manejo remoto para trabajar en la nube tiene desventajas, como la dependencia de la calidad de la conexión de red, pero a cambio se garantiza la escalabilidad y las actualizaciones periódicas al gestionar

los servicios el proveedor de la nube, que serán en principio operadores certificados por Dassault Systèmes.

Por otro lado, atendiendo a su dominio se clasifican en los siguientes cuatro tipos, cada uno asociado a un cuadrante de *Compass*:

- Aplicaciones sociales y de colaboración (cuadrante norte)
- Aplicaciones de modelado 3D (cuadrante oeste)
- Aplicaciones de simulación (cuadrante sur)
- Aplicaciones de inteligencia de la información (cuadrante este)

A continuación, se detallan brevemente las características principales de cada conjunto.

Aplicaciones sociales y de colaboración

Las aplicaciones sociales y de colaboración se enfocan principalmente en las funcionalidades para los procesos de negocio listadas más abajo, buscando que los usuarios de la empresa, desde planificadores de producto, diseñadores y directores de programa hasta planificadores de fabricación, agentes de compras, proveedores y directores de cumplimiento, puedan colaborar entre sí en el contexto global de los datos de producto.

Funciones:

- Planificación de productos y administración de programas
- Administración de configuraciones
- Administración de diseño global
- Ingeniería global y administración de fabricación
- Administración estratégica de relaciones con los proveedores
- Administración de cumplimiento y calidad
- Clasificación de IP y administración de seguridad
- Administración estratégica de relaciones con los clientes

Aplicaciones de modelado 3D

En este apartado se encuentran aplicaciones de Catia (*computer-aided three dimensional interactive application*) y Solidworks, que ofrecen un amplio espectro de soluciones de diseño, fabricación e ingeniería asistida por ordenador. No es únicamente software CAD, sino que incluye entre otros modelado de sistemas y arquitectura, sistemas eléctricos y electrónicos, análisis de sistemas fluidos o captura y reutilización de conocimientos.

Aplicaciones de simulación

La principal meta que buscan cumplir estas aplicaciones es mejorar el conocimiento del comportamiento en el mundo real dentro de un entorno virtual. Así se reducen los prototipos físicos, incrementa la confianza en el desempeño del producto y aceleran las decisiones de diseño, buscando siempre la eficiencia que otorga la posibilidad de anticipar diferentes procesos antes de realizarlos en la realidad.

De nuevo aquí tenemos dos subtipos. Por un lado, las aplicaciones de DELMIA, orientadas a los procesos de fabricación que incluyen la optimización del diseño de la planta y de las actividades de mecanizado, la investigación de problemas relacionados con factores humanos y la planificación de los procesos y los recursos. Por otro lado, las aplicaciones de SIMULIA, que proporcionan una gama de soluciones con las que simular el comportamiento de diseños en un entorno de modelado multifísica, es decir, análisis de elemento finito con la reproducción de problemas estructurales, térmicos y de dinámica de fluidos.

Aplicaciones de inteligencia de la información.

Estas aplicaciones permiten al usuario recopilar, conectar, descubrir, consolidar y entender de una manera holística la información de la plataforma 3DExperience o de cualquier otro sistema de la empresa o la web. Pueden mencionarse la creación de gráficos con *Metrics Reader* o el

uso de suministros de RSS mediante el *Feed Reader* con el fin de recopilar información en escritorios digitales.

3.4. Entidades esenciales de trabajo

Antes de profundizar más, es necesario mencionar la unidad básica de trabajo que se emplea para generar procesos: el contexto PPR (*Process Product Resource*). Bajo esta denominación se agrupan las distintas entidades requeridas para la creación o manufactura de productos. Cualquier producción necesita recursos específicos con los que llevar a cabo los distintos procesos que elaboren el producto. Los productos y recursos pueden asignarse a procesos; de igual manera, ciertos procesos especializados (llamados tareas) pueden asignarse a recursos tales como los robots.

Las entidades principales que se agrupan bajo los contextos PPR son las cuatro que se listan a continuación:

- Producto ('Product'): la parte física de un objeto o ensamblaje (grupo de objetos) con propiedades y representaciones del mismo.
- Artículo-manufactura ('Manufactured Item'): describe cómo se transforma un producto.
- Sistema ('System'): la realización operacional del artículo, donde se define los flujos necesarios para una capacidad dada. Aquí se encontrarían las operaciones, su duración y las restricciones de tiempo si se tiene un ciclo de tiempo determinado.
- Recurso ('Resource'): hace referencia tanto a la disposición de la planta como a los componentes (robots, máquinas, herramientas) utilizados para la producción. La diferencia con los productos consiste en que los recursos crean, manipulan o ensamblan productos. Además, los recursos son programables.

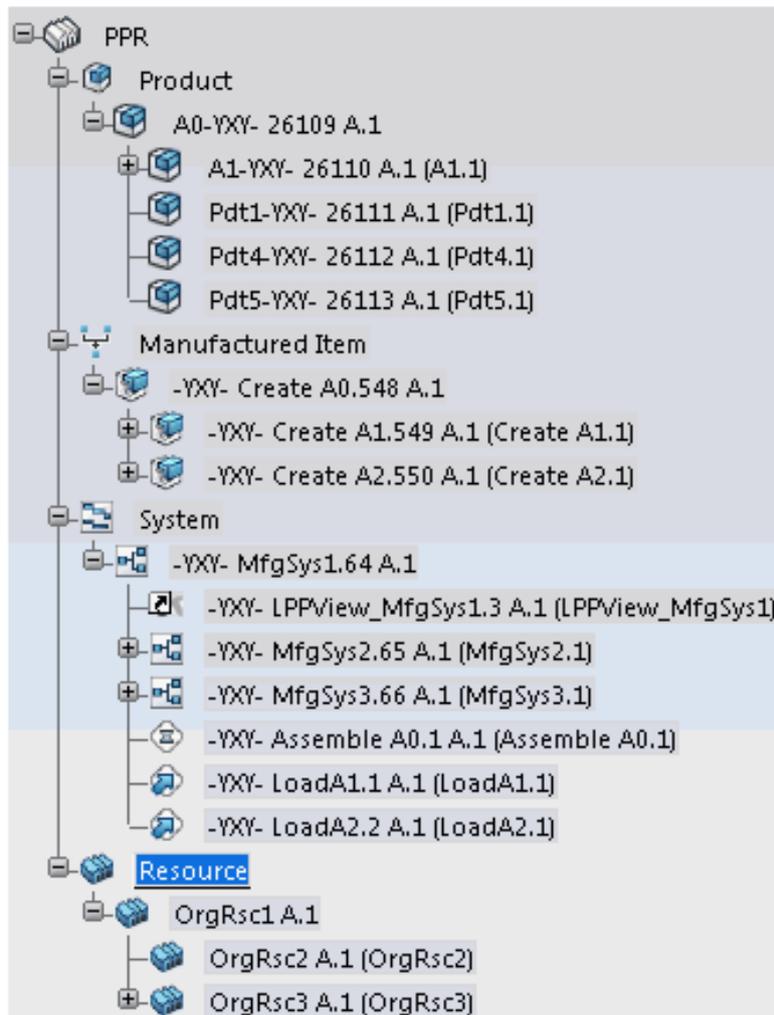


Figura 10: Muestra de 'tree' o árbol de recursos disponibles y relaciones de subordinación entre ellos.

La funcionalidad que se examina en este documento de forma más detallada es la correspondiente a las aplicaciones de robótica del ecosistema 3DEXperience. Particularmente dos: programación y simulación. Ambas se engloban en las llamadas *Simulation Apps*, en el subpartado correspondiente a DELMIA. Por otro lado, se complementa con otras aplicaciones orientadas al diseño, sea de piezas o de disposición de elementos de una planta, como CATIA 'Part Design' y DELMIA 'Plant Layout Design'.

En general, los programas de robótica proporcionan modelados en 3D a nivel de la celda de trabajo, incluidos estudios y programación de la

factibilidad robótica en un entorno donde los datos de productos y los recursos de fabricación se pueden integrar en un modelo 3D. Por tanto, permiten crear dispositivos, diseñar recursos y crear tareas para los dispositivos de la fábrica.

4. Diseño de una planta robótica con 3DExperience

4.1. Planteamiento, función y composición de la planta

Se pretende diseñar, programar y simular una célula de fabricación flexible robotizada mediante las aplicaciones de robótica y diseño 3D que se incluyen en el software 3DExperience de Dassault Systèmes. Debe tenerse en cuenta que no consiste en una estación de trabajo aislada, sino que se ubica dentro de un sistema de producción PLM, con lo que el diseño puede estar sujeto a variaciones por distintos requerimientos. Estas variaciones deberán llevarse a cabo de manera paralela al trabajo de otros profesionales de campos distintos, razón por la que el intercambio continuo en tiempo real de información y la rapidez serán factores capitales.

En este caso concreto se busca la construcción de un vehículo militar blindado. La célula de trabajo se encarga de la soldadura de dos piezas metálicas que formarán parte integral de la estructura, con una forma determinada para observar distintas prestaciones. Posteriormente deberán paletizarse para su traslado a una localización cercana donde se continuará con la producción. El tiempo de ciclo no debe superar el minuto para conseguir la eficiencia deseada.

Los robots utilizados para acometer estas tareas son un FANUC 100iC, encargado de la soldadura con una herramienta adecuada para tal fin, y un ABB IRB_4600_40_255, encargado de manipular las piezas y trasladarlas a las sucesivas posiciones mediante un electroimán.

La selección de los robots se ha hecho tomando los criterios de:

- variedad, utilizando dos robots de distintos fabricantes, algo que muchos otros programas de robótica no pueden conseguir al estar asociados a una marca;
- la función para la que están diseñados

- la carga que pueden soportar
- la envolvente de trabajo.

Estas características se recogen en la tabla siguiente:

Tabla 1. Características principales de los robots elegidos

Fabricante	Modelo	Función	Carga máxima	Alcance máximo
ABB	IRB_4600_40_255	Posicionador	60 kg	2552mm
FANUC	Mate 100 iC/ 6L	Soldador	7kg	1633mm

Las piezas están constituidas de un metal ferromagnético de baja densidad que recubre un polímero ligero y ultrarresistente. Su peso es de 15.32 kg y 26.11 kg, quedando lejos de los 60 kg de carga máxima del robot ABB incluso añadiendo el peso del electroimán. Para el FANUC tampoco hay problema: la pistola tiene una masa de 4.93kg.

La elección de la herramienta tampoco ha sido trivial. En un principio se valoró la utilización de una pinza mecánica para trasladar los objetos, pero dado que iban a ser productos magnéticos y que el robot tenía un puerto eléctrico, se optó por el electroimán. El elemento 3D que se observa en la planta simulada no tiene por qué ser la geometría real, como se aclarará posteriormente: el volumen del utilizado en la planta puede ser mayor y la forma distinta. El factor relevante es que pueda levantar las piezas, para lo cual la fuerza desarrollada debe ser lo suficiente. Esta fuerza viene dada por la expresión $F = \frac{B^2 * A}{2\mu}$ siendo F la fuerza en Newtons, B el campo magnético en teslas, A el área de las caras en m², μ_0 la permeabilidad magnética del medio. Para su diseño puede asumirse la simplificación del campo magnético cerrado donde el valor de B es directamente proporcional a la permeabilidad del material (μ), la corriente total abrazada que fluye por la bobina (N*I) e inversamente proporcional a la longitud del camino recorrido por el circuito magnético(L).

El alcance máximo no se logra en cualquier dirección, como es lógico. En la hoja de datos de los robots aparecen algunas especificaciones de estos aspectos, pero es más visual observar en el programa la envolvente de cada robot para percibir, aunque sea de una manera aproximada, los límites de cada uno:

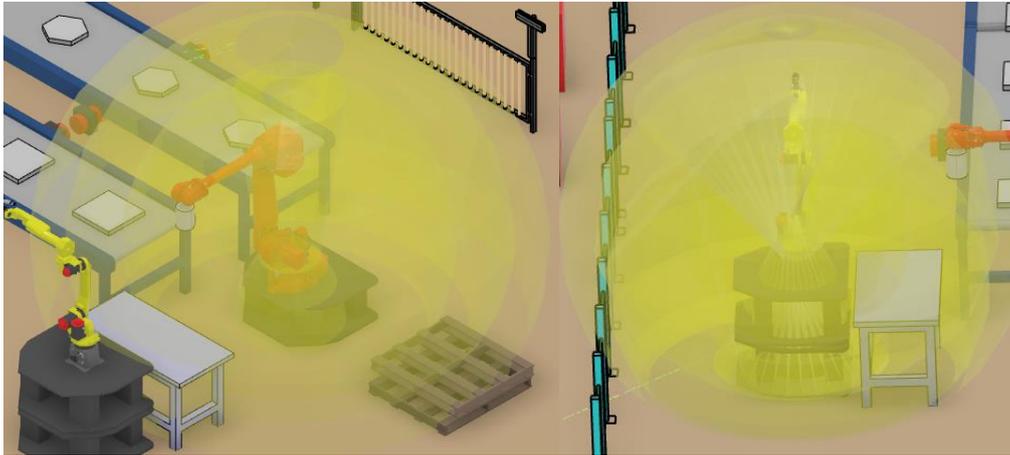


Figura 11: Espacio de trabajo o envolvente de los robots

El proceso que se lleva a cabo cuenta con los siguientes pasos:

- 1- Ambos robots están en posición de reposo. Por dos cintas transportadoras distintas llegan dos piezas metálicas, una de base cuadrada de dimensiones 500x500x50 mm y otra de base hexagonal de apotema 200mm y altura 50mm.
- 2- El robot ABB posiciona en la mesa de trabajo las piezas, colocando la hexagonal encima de la cuadrada y se separa volviendo a la posición de reposo.
- 3- El robot FANUC realiza la soldadura de ambas piezas y vuelve a su posición de reposo.
- 4- El robot ABB coloca la nueva pieza en un pallet.
- 5- Se repite el proceso, colocando la pieza terminada de forma contigua a la anterior hasta colocar cuatro, llenando un nivel.

Para conformar la celda de trabajo se han utilizado los elementos listados en la tabla siguiente:

Tabla 2. Cantidad y disposición de los elementos de la celda.

Elemento	Cantidad	Posiciones de los elementos (mm)		
		X	Y	Z
Suelo (floor)	1	0	0	0
Plataformas de robots	3	2100	3100	0
		2100	3100	500
		-1200	3100	0
Mesa de trabajo	1	1000	3200	0
Robot posicionador ABB	1	-1200	3100	500
Robot soldador FANUC	1	2100	3200	1000
Pallet	2	-1000	5000	0
		-1000	5000	200
Cinta transportadora	4	-1900	-4700	0
		-1900	-500	0
		-2900	-4700	0
		-2900	-500	0
Controlador robot	2	5000	1000	0
		5000	3000	0
Controlador cinta	2	4600	-3300	0
		4600	-1300	0
Valla seguridad	5	-5000	0	0
		-5000	-3000	0
		-5000	-6000	0
		3600	0	0

		3600	-6300	0
Pinza	1	-6400	5100	0
Electroimán	1	Herramienta robot posicionador		
Pistola de soldadura	1	Herramienta robot soldador		

En la imagen siguiente se puede apreciar la manera en que se disponen los distintos elementos. Excepto las piezas y las herramientas robot, se han podido obtener de catálogos incluidos en el programa (más sobre esto en los siguientes puntos).

Usar las aplicaciones de robótica de la plataforma implica moverse entre dos entornos diferentes dentro del software: el de diseño y el de simulación 3D. En el primero de ellos se pueden modelar las distintas partes que se quieran utilizar, construir los dispositivos y disponer los elementos de la celda de trabajo que queramos simular. El segundo entorno comprende lo relativo a la creación de tareas robot o programación, coordinación entre robot y herramienta, edición de las señales IO, los parámetros de movimiento y por supuesto, la simulación.

A continuación, se explicará la metodología con la que se ha constituido la celda de fabricación, mencionando las distintas aplicaciones de que se ha servido el autor y las funcionalidades más relevantes de cada una de ellas.

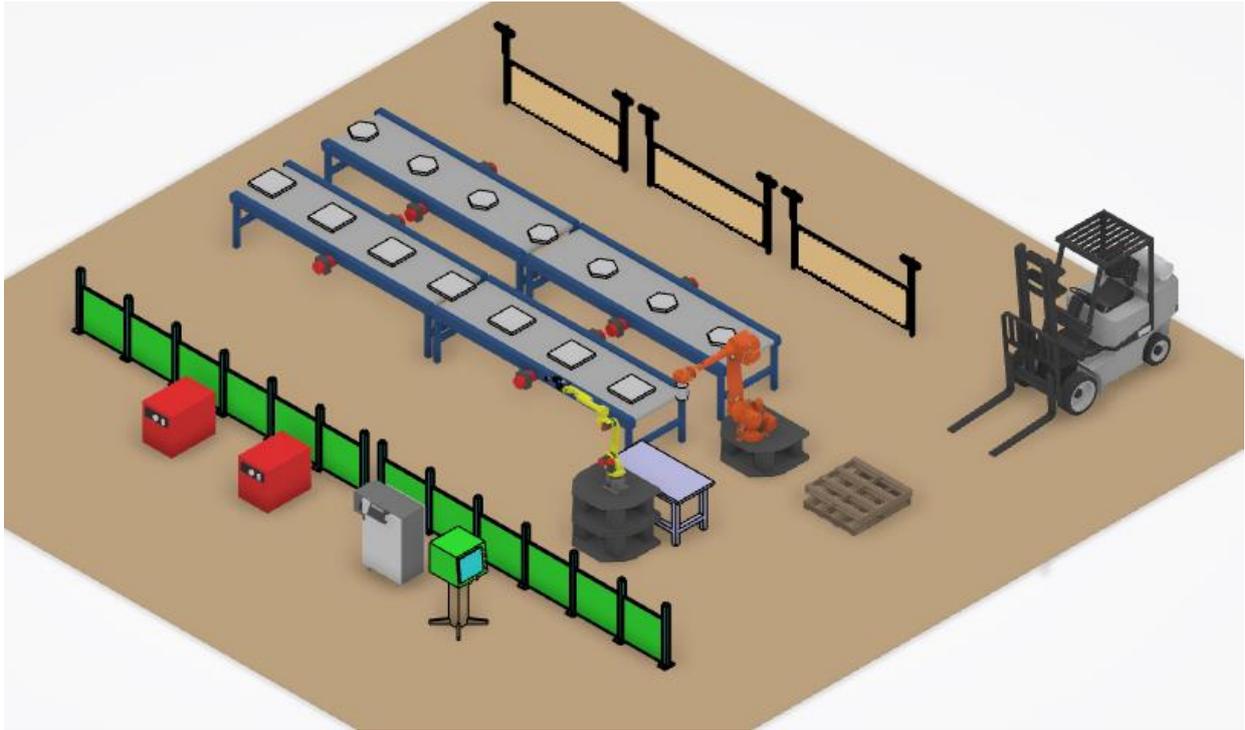


Figura 12: Vista isométrica de la célula de trabajo

4.2. Diseño 3D: CATIA PART DESIGN



Mediante esta aplicación se pueden diseñar sólidos en 3D, con toda la complejidad que se requiera. En general se utiliza con sólidos unitarios, siendo las principales operaciones definir la geometría y añadir o quitar material. Las funcionalidades empleadas para este trabajo han sido las básicas, ya que se han diseñado formas sencillas: boceto ('Sketch'), para dibujar la base en 2D de las piezas y extrusión ('Pad') para definir la tercera coordenada. La interfaz es muy parecida a programas como Inventor o NX, por lo que una persona con conocimientos mínimos de software de diseño puede usarlo con relativa facilidad.

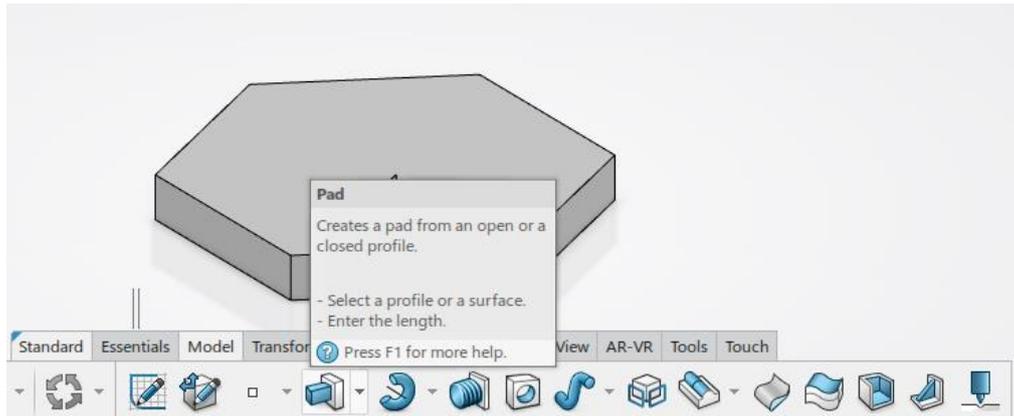


Figura 13: Operación de extrusión en CATIA PART DESIGN

Como puede deducirse del nombre, es un módulo de CATIA, un programa extensamente utilizado en sectores como el de automoción, naval o aeroespacial. Para diseñar entidades complicadas tal vez se imponga trabajar por superficie, para lo cual es indicado la aplicación 'Generative Shape Design' o incluso crear superficies racionales con el módulo 'ICEM', con el cual se generan, por ejemplo, las carrocerías de los coches. Otras posibilidades son el 'Sheet Metal Design', para diseñar chapa, y el 'Assembly Design', donde se ensamblan los distintos componentes básicos o partes creados en los otros módulos.

Para la preparación de la planta, se han tenido que fabricar en la aplicación CATIA DESIGN PART los componentes que no se encuentran en catálogo: el electroimán y las piezas que conforman el producto.

La geometría es sencilla: la de las piezas ya se ha mencionado y el electroimán es un cilindro de radio base 70mm y altura 230mm con otro pequeño cilindro concéntrico de dimensiones $R=30\text{mm}$ y $\text{altura}=12\text{mm}$.

Realmente no es necesario la total fidelidad geométrica para la simulación, razón por la cual no se ha fabricado algo muy complejo; lo relevante es la equivalencia dinámica (propiedades de masa, centro de gravedad y momentos de inercia iguales al que se use en la planta real que se quiera controlar, lo que puede definirse fácilmente) y la situación del TCP. La masa es 7.56 kg el cdg se sitúa en (117.027, 0, 0) mm -téngase en cuenta que

la base se sitúa en el centro del plano YZ y el eje de la pieza coincide con el X-. Los momentos de inercia obtenidos se muestran en la siguiente captura de pantalla:

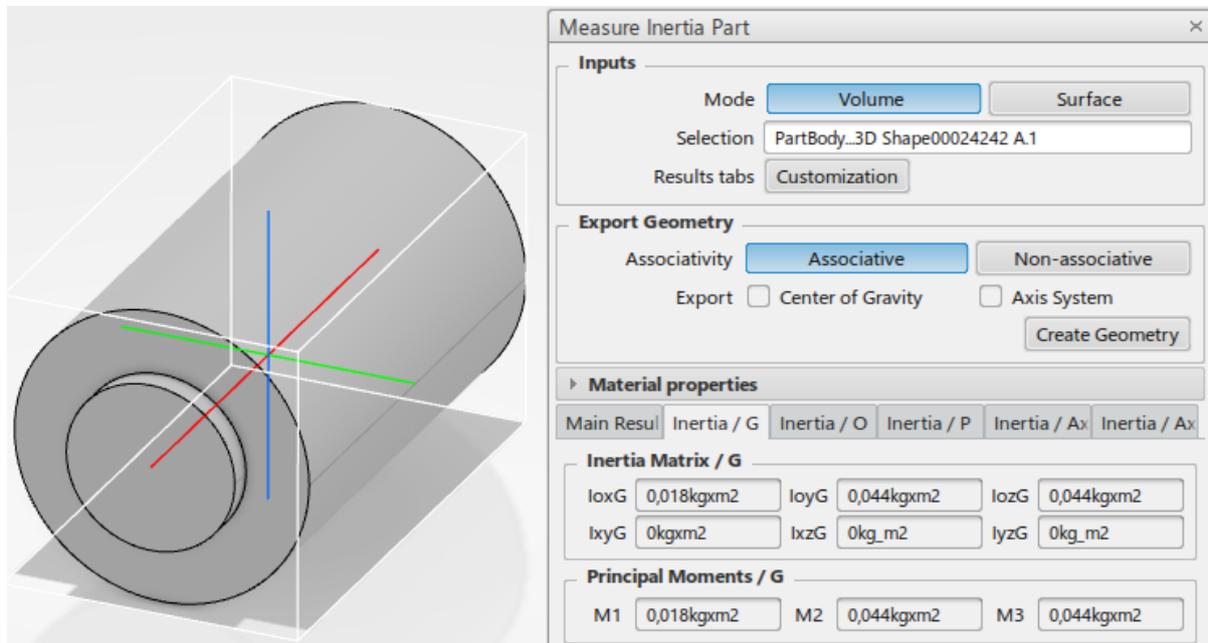


Figura 14: Cálculo de parámetros dinámicos en CATIA PART DESIGN

El programa lo calcula a la hora de simular, y en los recursos que se obtienen del catálogo (por ejemplo, los robots) ya están introducidos los datos necesarios. Este pequeño inciso sirve para cotejar las posibilidades de la aplicación, que permite controlar numerosísimos detalles, logrando así una más que fiable emulación de los procesos reales.

4.3. Plant Layout Design.



Esta aplicación proporciona al usuario la capacidad de diseñar una planta de trabajo en 3D, disponiendo los distintos elementos como se desee.

Solo se hará uso de tres comandos: 'Change current axis', 'Create Manufacturing Cell' y 'Create Manufacturing Product'. Con el primero se trasladan y se giran los elementos respecto a los ejes principales propios.

Con el segundo se crea el recurso fundamental 'Manufacturing Cell' (traducido libremente: célula de fabricación) entendido como un grupo cualquiera de dispositivos y elementos con una organización lógica. Con el tercero se da a los elementos incluidos en él la categoría de producto de manufactura, dado que serán los que se manipulen y de cuya unión se obtenga el producto final deseado de la planta. En nuestro caso, los prismas metálicos.



Figura 15: Situación de las entidades 'Manufacturing Cell' y 'Manufacturing Product' en el árbol de recursos

Aunque aquí se pueden llevar a cabo algunas acciones para preparar la planta con vistas a la simulación tales como generar el recurso herramienta, todo lo demás se hará en la aplicación robot simulation. Como se comprueba, las aplicaciones no son compartimentos cerrados o exclusivos, sino que muchas veces varias funcionalidades o comandos se comparten entre dos o más módulos.

4.4. Simulación Robot.



La aplicación 'Robot Simulation' proporciona todas las herramientas para llevar a cabo estudios de viabilidad y programación en un entorno donde se pueden integrar en un modelo 3D tanto recursos industriales como los productos (sean físicos o sus datos). Es la base para el desarrollo de procesos en los que intervienen robots: proporciona los instrumentos para comprobar el desempeño y alcance de robots en una célula de fabricación compleja, estudiando posibles interferencias en la ejecución de las tareas

según se posicionen los distintos recursos. Por supuesto, se habilita a programar robots, depurar trayectorias, establecer señales de entrada/salida (IO) entre los controladores robots y otros elementos y simular tareas, ya sea individualmente o de forma secuencial.

Dentro de este módulo también se emplea la aplicación 'DELMIA Robot Arc Simulation' para definir la trayectoria de soldadura. No obstante, al ser tan parecidas (es una extensión de 'Robot Simulation') he decidido incluir todos los aspectos bajo el capítulo de esta aplicación, ya que es la más importante de las que se estudian para cumplir el objetivo del proyecto.

En el siguiente diagrama se resumen los pasos para utilizar eficazmente las dos aplicaciones anteriores:

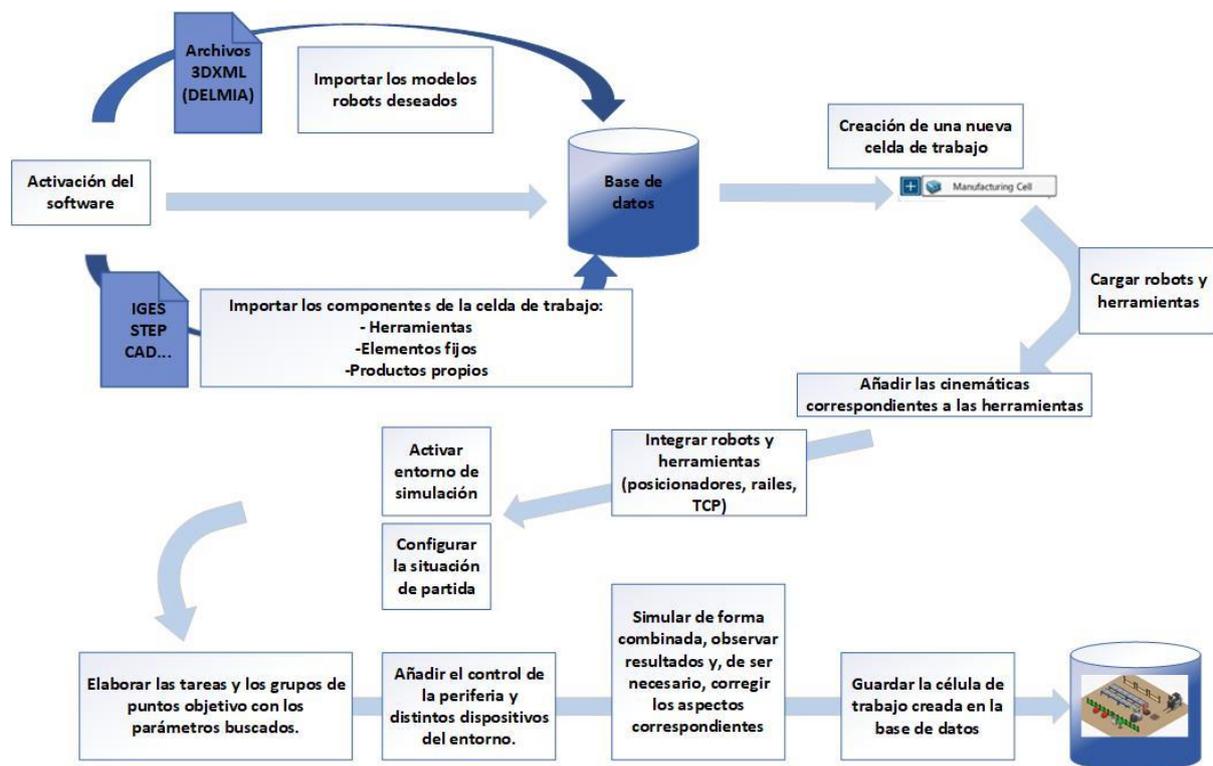


Figura 16: Descripción del proceso de creación de una estación de trabajo con 'Robot Simulation'.

Los robots se pueden importar desde catálogos de los fabricantes proporcionados por el propio programa, entre los que hay una gran

variedad. También hay algunas herramientas robot, aunque existe la posibilidad de diseñarlas en archivo CAD (CATIA Design) como se ha hecho. La aplicación utiliza el concepto de 'puertos' para montar las herramientas en los robots. Son unos marcos típicos de robótica (X, Y, Z, Yaw, Pitch, Roll) que aluden a los distintos grados de libertad existentes; son la base también para localizar y controlar los robots sobre raíles.

Hay que destacar que las herramientas tienen que establecerse como tal, no son solamente la geometría proveniente del archivo CAD. Para ello hay que atribuir características como el perfil de herramienta, los puertos de conexión con el robot y los TCP. Esto se consigue insertando la geometría en un recurso 'Tool Equipment' y con los comandos 'Create Port' y 'Set Tool' (imágenes siguientes). También permite seleccionar el controlador.

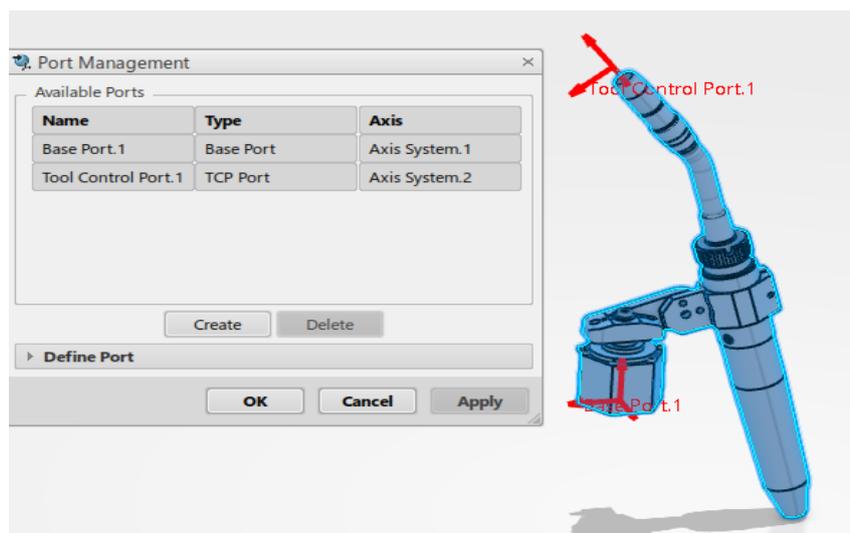


Figura 17: Creación de puertos en una herramienta robot

La herramienta de soldadura es del tipo pistola MIG TOUGH GUN ThruArm G2, de Tregarkiss. El cable vinculado a esta es flexible e incluye tecnología LowStressRobotic, que incorpora una conexión giratoria de la fuente de alimentación para aliviar el esfuerzo generado por la torsión del cable. Gracias a estas características no hay que preocuparse demasiado por los movimientos del robot soldador, pues deja absoluta libertad.

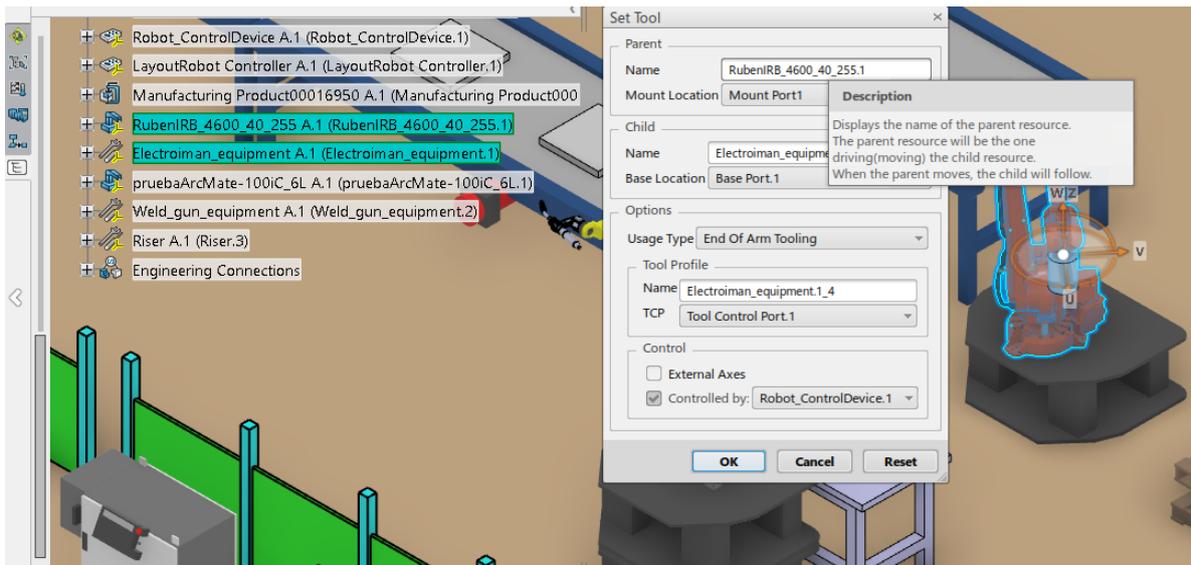


Figura 18: Fijación de la herramienta robot

Un aspecto importante de la unión herramienta-robot que se facilita en la aplicación es que consiste en una vinculación interna entre padre e hijo más compleja que una simple fijación. Es posible montar una herramienta sin articulación (esto es, sin cinemática), con articulaciones y posición inicial o 'home position', o incluso añadir un dispositivo de control para considerar el conjunto como uno solo. En este último caso, el control combinado se ocuparía no sólo de las seis articulaciones del robot, también de las de la herramienta (en caso de tenerlas, suelen ser como mucho tres). Se puede definir así cada operación (como, por ejemplo, activar y desactivar una pistola de soldadura) como una acción que ocurre cada vez que el robot se posiciona en el punto designado (en este caso sería punto de soldadura).

4.4.1. 'Simulation State' o estado de simulación y controladores de movimiento

Un estado de simulación o Simulation State define un estado transitorio de un producto o recurso, que puede guardarse y reusarse. Puede utilizarse como un estado inicial en una simulación para conseguir resultados consistentes en repeticiones sucesivas. A lo largo del trabajo se han ido guardando estados de diferentes momentos clave: el estado de reposo

inicial, el estado previo a la soldadura de las piezas en la mesa de trabajo, el estado posterior a la paletización de la pieza y el estado tras el movimiento de las cintas transportadoras. Reusándolos se han podido observar y corregir algunos aspectos de velocidad de la ejecución de tareas. Respecto a los controladores de movimiento, estos no se limitan a los robots, sino que deben crearse si tenemos varios recursos que necesitan de programación en un contexto de organización. Puede ser una planta o estación de trabajo. Un ejemplo sencillo consistiría en un robot y una línea transportadora en un área delimitada, cuyo funcionamiento se programa para que ambos recursos operen en la secuencia planeada.



Figura 19: Controladores de movimiento asociados a recursos de una planta.

En la célula de trabajo que nos ocupa se han añadido cuatro controladores para controlar los robots y las cintas transportadoras. En el punto 4.4.3 se explican más detalladamente las conexiones que se establecen entre los recursos.

4.4.2. Tag Points o puntos objetivos

Son los elementos constructivos fundamentales de una simulación con robots involucrados. Los 'Tags Points' son localizaciones en el espacio en coordenadas X, Y, Z, Yaw, Pitch, Roll, generalmente relativas a la base del robot. Los términos 'Target', 'Frame', 'Position' son utilizados también por otros fabricantes para referirse a ellos.

En este software los Tags deben crearse en relación con un grupo de puntos o 'tag group'. Estos conjuntos de puntos definen el camino en que deberá desplazarse el robot, siguiendo la programación. Por idoneidad suele nombrarse a cada Tag group con el mismo nombre que la tarea a la que se asocia, evitando así posibles errores.

Para la célula de trabajo, se creará un grupo de puntos relacionado con la actividad de soldadura ('Weldtags') y dos relacionados con la actividad de posicionamiento: una para colocar la base cuadrada y la pieza hexagonal sobre la mesa ('Pick_base') y otra para almacenar el producto unión de las dos partes anteriores en los palés ('Palletize'). En la imagen siguiente se muestran en cruces color negro. Sus nombres, posición y orientación se detallan en las siguientes tablas. Como se puede observar a la izquierda en el panel 'Behaviour' aparecen indicados los grupos de puntos creados.

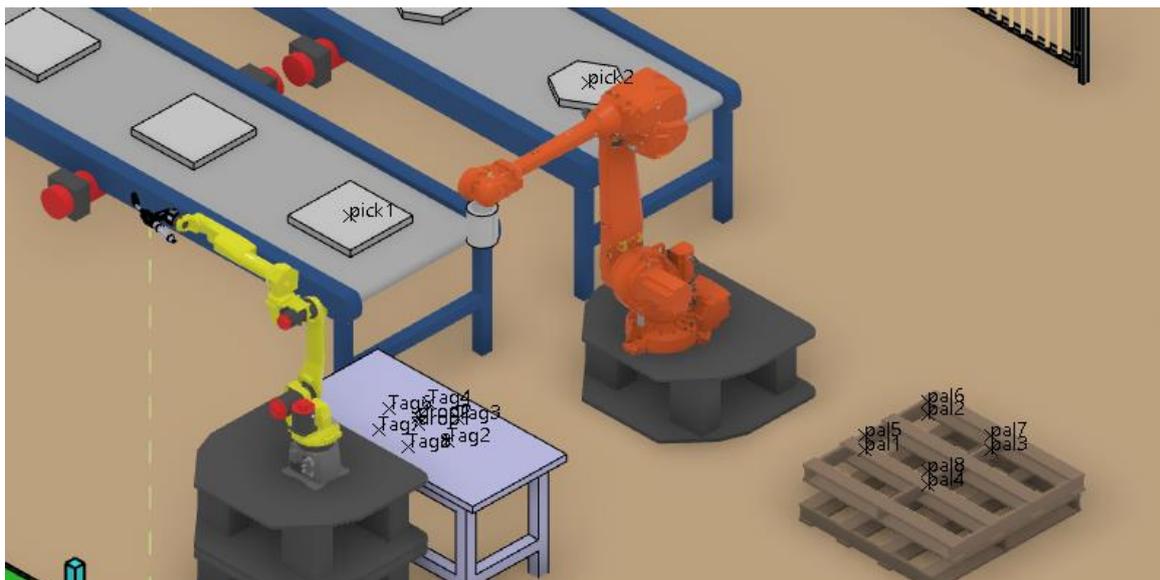


Figura 20: Visualización de los puntos de trabajo creados

La orientación de los puntos es de vital importancia, pues la dirección de acercamiento del robot a los mismos se hace a lo largo del eje W (ó Z) del puerto TCP de la herramienta que tenga ligada. En especial cobra relevancia con la soldadura, donde se debe tener especial cuidado en el movimiento de la herramienta, buscando el mejor ángulo de ataque posible para

conseguir una soldadura correcta. De igual manera, en casos en que se trabaje en el límite de la zona de trabajo se tendrá que considerar este aspecto.

Para variar sus parámetros en caso de necesidad, diversos comandos habilitan funcionalidades tales como copiar la alineación de un punto en otro, interpolar entre dos dados y proyectar sobre una superficie o producto puntos creados. También es posible su manipulación directa, girándolos o moviéndolos respecto a su propio sistema coordenado.

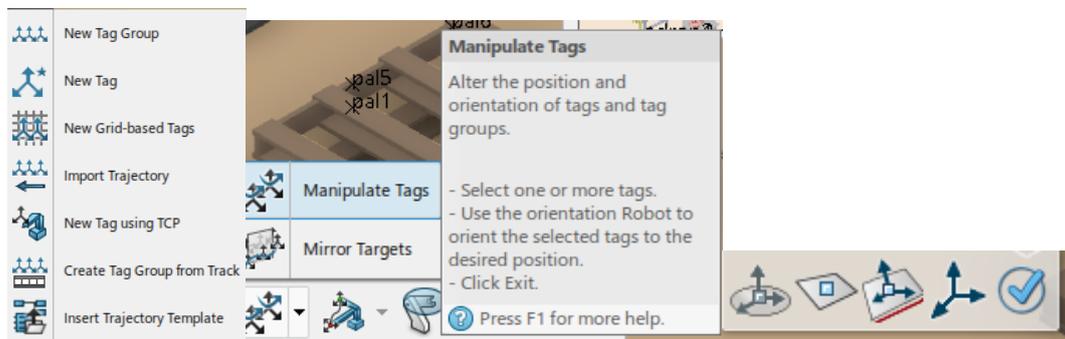


Figura 21: Comandos de manipulación de 'tags'

El formato de exportación e importación de tags es .xls (Excel).

Tabla 3. Parámetros de los puntos del grupo 'Weldtags'.

TagGroup Name :	Weld_tags	Global					
Tag Prefix	Tag Index	X(mm)	Y(mm)	Z(mm)	Yaw(deg)	Pitch(deg)	Roll(deg)
Tag	1	1255	3315.47	930	0	0	-90
Tag	2	1055	3430.94	930	0	0	-30
Tag	3	855	3315.47	930	0	0	30
Tag	4	855	3084.53	930	0	0	90
Tag	5	1055	2969.06	930	0	0	150
Tag	6	1255	3084.53	930	0	0	-150
Tag	7	1255	3315.47	930	0	0	-90

Tabla 4. Parámetros de los puntos del grupo 'Pick_base'.

TagGroup Name :	Pick_base	Global					
Tag Prefix	Tag Index	X(mm)	Y(mm)	Z(mm)	Yaw(deg)	Pitch(deg)	Roll(deg)
pick	1	-100	1500	1000	0	90	0
drop	2	1055	3200	980	30	90	0
pick	2	-2000	1500	1000	0	90	0
drop	1	1055	3200	930	0	90	0

Tabla 5 Parámetros de los puntos del grupo 'palletize'.

TagGroup Name :	palletize	Global					
Tag Prefix	Tag Index	X(mm)	Y(mm)	Z(mm)	Yaw(deg)	Pitch(deg)	Roll(deg)
prod	1	1055	3200	980	0	0	0
pal	1	-1050	4666,56	430,339	0	0	0
pal	2	-1550	4666,56	430,339	-90	90	0
pal	3	-1550	5166,56	430,339	-90	90	0
pal	4	-1050	5166,56	430,339	-90	90	0
pal	5	-1050	4666,56	530,339	-90	90	0
pal	6	-1550	4666,56	530,339	-90	90	0
pal	7	-1550	5166,56	530,339	-90	90	0
pal	8	-1050	5166,56	530,339	-90	90	0

Una funcionalidad interesante, aunque no haya sido menester emplearla aquí, es la posibilidad de crear un equipo de grupos de puntos objetivo formando una cuadrícula en 2D. Varias de estas cuadrículas apiladas forman una malla. También es posible fijar un grupo a un elemento, de manera que cuando este varía su posición u orientación, los puntos se muevan solidarios con el mismo.

Para realizar la trayectoria de soldadura se emplea el comando 'create arc trajectory', donde especificamos el grupo de puntos 'WeldTags' y las superficies de intersección que queremos unir.

4.4.3. TASKS: Tareas

'Task' es el término utilizado para referirse a la secuencia que determina las características de orden y movimiento para cada desplazamiento y operación del robot.

El movimiento a una localización concreta es el componente principal de las tareas. Otras operaciones que se pueden definir son las relativas a soldadura, a coger o soltar un producto (grab/reléase activity) o incluso a la conexión-separación de la herramienta.

Un ejemplo de esto último puede consistir en utilizar dos pinzas para un mismo robot: según tenga que realizar unas operaciones u otras, puede usar la pinza indicada y dejar la otra preparada en una superficie cercana. Cuando haya que cambiar, el robot deja la actual y conecta la otra a la muñeca. Esto tiene limitaciones, como es obvio, tales como la necesidad de que ambas herramientas funcionen con la misma fuente o necesiten los mismos recursos (una ventosa de aire que utilice un circuito de aire comprimido no podría compartir el espacio con una pieza de accionamiento eléctrico) o que se posea sistemas de acoplamiento especializados con platos capaces de transmitir distintas señales entre sí -neumática, mecánica, hidráulica, eléctrica, etc.-.

Existe la opción de programar bucles y condiciones lógicas (if, then, else, wait) para ejecutar las tareas según se requiera, creando caminos alternativos. A continuación, se detallan los tipos de bucles que se pueden utilizar para construir una secuencia lógica:

- ➔ While-Do: se comprueba al inicio del bucle si la condición impuesta se cumple, en cuyo caso se ejecutan las instrucciones que haya

dentro; si la condición retorna un 'false', el conjunto de instrucciones en el bucle no se ejecutará.

- ➔ Do-While: en este caso la comprobación de la condición se lleva a cabo al final del bucle y en caso de ser positiva, vuelve a ejecutarlo. Esto significa que al menos una vez las instrucciones contenidas en el bucle son ejecutadas.
- ➔ For-Loop: para realizar una serie de instrucciones un número dado de veces.
- ➔ Break: Instrucción que sirve para salir de cualquier bucle.

Por supuesto, también es posible programar tareas en componentes como las cintas transportadoras, siendo generalmente estas tareas de posicionamiento de productos en las que no es preciso un grupo de puntos. Una vez se han definido y manipulado los tags points, se tiene soltura suficiente para utilizarlos y añadirlos a una tarea (generalmente se añade un grupo).

Para la generación de tareas se selecciona el recurso al que se quiera añadir una tarea para activarlo, convirtiéndolo en el ASO (Active Simulation Object) de la celda. Hay distintas formas: por ejemplo, con el comando 'Teach' se puede ir moviendo a las posiciones deseadas el robot o recurso, grabándolas en el orden deseado. Otro modo de conseguirlo es usando como cimiento los grupos de puntos mediante el comando 'Create Task From Trajectory'. Para este caso, no obstante, se producirán las distintas tareas a partir de los controladores, ya que interesa formular las tareas como servicios que puedan ser llamados en la tarea principal, la cual se definirá posteriormente. El comando indicado en esta ocasión es 'New Device Task', similar a 'Teach'. Es especialmente práctico en cuanto que permite abrir múltiples paneles para distintos robots al mismo tiempo. Además, posee un editor secuencial gráfico para programar recursos y tareas utilizando la lógica mencionada arriba.

Es una buena práctica nombrar las tareas y los puntos objetivo relacionados del mismo modo o parecido, pudiendo así manejarlos mejor y evitando

posibles errores diferenciándolos del resto. Si se quiere que estas tareas estén disponibles en el panel de recursos y puedan ser llamadas (en una rutina, por ejemplo), se pueden crear como servicio. Este servicio tendrá el mismo nombre que el de la tarea a la que alude.

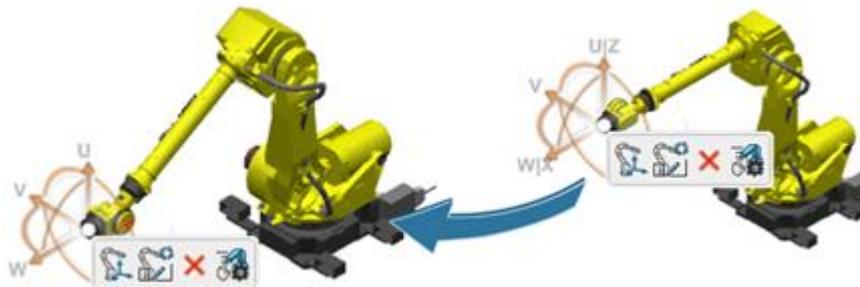


Figura 22: Arrastre de un robot a un punto con el comando 'Teach'

En esta estación de trabajo hay cuatro controladores que controlan ambos robots y las cuatro cintas transportadoras: el controlador verde se ocupa del robot soldador, el gris del posicionador, el rojo que sigue de las dos cintas por las que se desplazan las piezas de base cuadrada y el rojo del extremo de las cintas que transportan las piezas de base hexagonal.

Las relaciones se establecen mediante el comando 'Motion Controller', especificando para los robots el control de las articulaciones ('arm') y para las cintas que puedan posicionar piezas de trabajo ('Workpiece Positioner'). Utilizándolos como objeto activo consecutivamente se establecen las siguientes tareas-servicio:

- Move_Conveyor1, para el dispositivo de control de *Conveyor_ControlDevice1*. Mueve las piezas de base cuadrada por las cintas transportadoras 1.1 y 1.2. Hay tres órdenes principales para completar esta tarea: primero las cintas cogen los objetos, luego los desplazan en el eje 'y' 1250mm y los sueltan.
- Move_Conveyor2. Ídem que el anterior, pero siendo para el controlador *Conveyor_ControlDevice2* con los prismas hexagonales y las cintas transportadoras 2.1 y 2.2.

- *Pick_drop1*, para el controlador ligado al robot ABB. En esta tarea se recogen las órdenes para el robot posicionador de coger sendas piezas del final de las cintas transportadoras y situarlas en la mesa de trabajo. Después de ello se aparta, dejando espacio al robot soldador. La trayectoria se realiza con apoyo en los puntos pertenecientes al grupo *Pick_base*.
- *Weld*, para el controlador del robot FANUC. En esta tarea se realizan las órdenes de soldadura: la herramienta se desplaza por los bordes de unión entre ambas piezas. Una vez ha terminado, se retira a la posición de reposo. La trayectoria se apoya en los tags del grupo de puntos *Weldtags*.
- *Pick_drop_pal1*, para el controlador del robot ABB. Por último, el robot posicionador recoge el producto formado y lo coloca en el lugar correspondiente del palé. La base de esta tarea han sido los puntos objetivo del grupo *Palletize*.

A continuación, se muestran las trayectorias de las tareas correspondientes a los robots. En primer lugar, *Pick_drop1*:

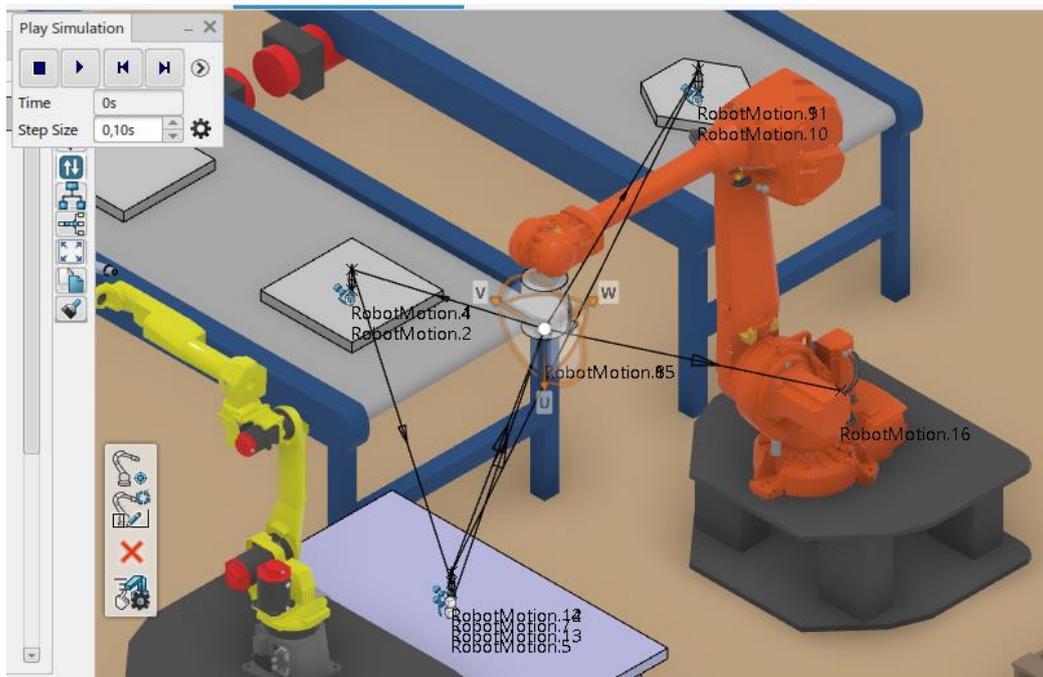


Figura 23: Trayectoria seguida por el robot posicionador (tarea '*Pick_drop1*')

Tras realizar esta primera tarea, se guardan los cambios en un nuevo estado de simulación denominado *Pre_weld*. En la siguiente imagen se muestra dicho estado y la trayectoria a seguir por el robot soldador.

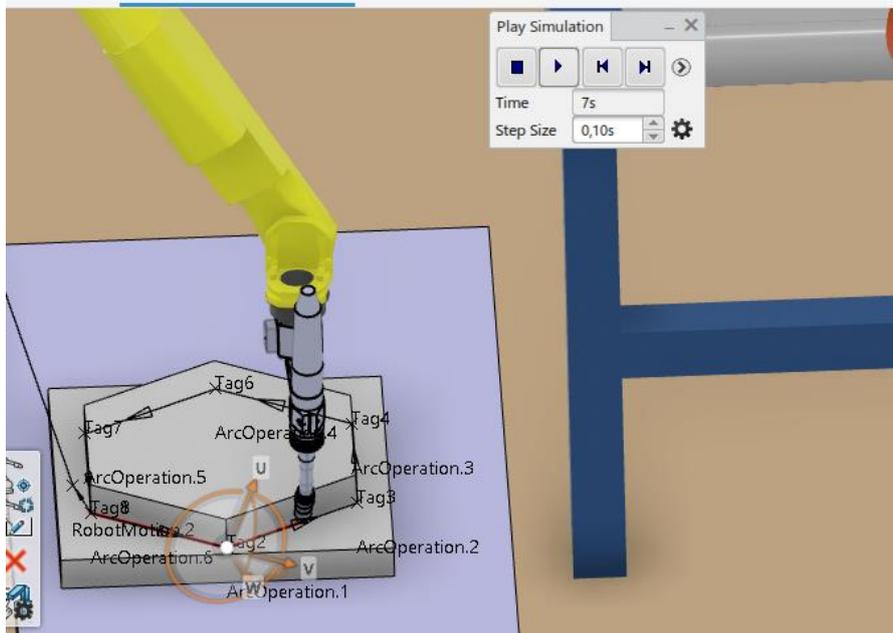


Figura 24: Detalle de la trayectoria seguida por el robot soldador (tarea 'Weld')

Posteriormente el robot ABB coloca el primer producto en el sitio correspondiente del pallet y vuelve a posición de reposo, donde comprobará que la señal proveniente de la primera cinta esté activada para realizar de nuevo la primera tarea con el siguiente par de piezas.

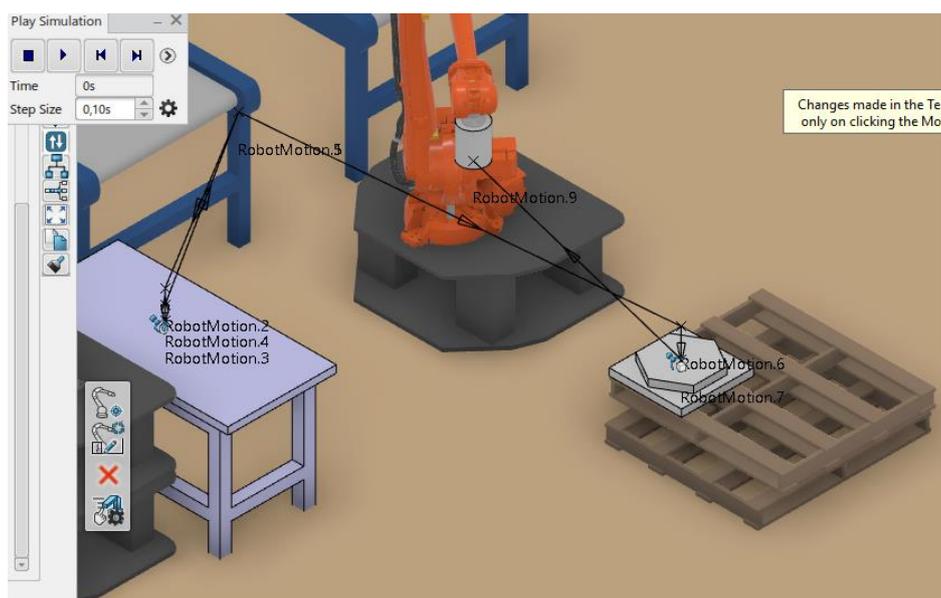


Figura 25: Trayectoria seguida por el robot posicionador (tarea 'Pick_drop_pal1')

Tras cuatro iteraciones se llega a cubrir el palé.

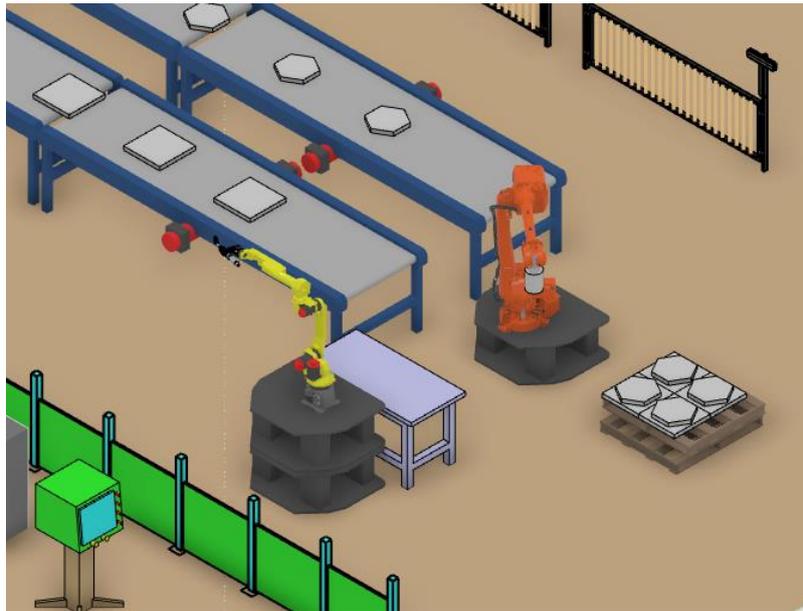


Figura 26: Detalle de la estación de trabajo tras cuatro ciclos

Mención aparte merece el comando 'Mirror targets', con el que podemos imitar una tarea o disponer una serie de puntos objetivo como si hubiesen sido reflejados en un espejo. Es valioso especialmente en casos en los que dos robots realizan una misma acción de forma simultánea. Ahorra mucho tiempo y trabajo, dado que estas acciones son complejas por su naturaleza – imagínese dos robots trasladando la misma pieza a la vez: la velocidad, puntos objetivo y precisión debe ser igual constantemente-.

Una vez se tienen definidos unos diversos puntos objetivo y tareas asociadas, debe analizarse si el robot propuesto puede cumplir eficientemente con su cometido y, si la respuesta es afirmativa, cuál es la mejor manera de llevar a cabo las sucesivas operaciones. Existen varios comandos que ayudan al usuario a verificarlo.

Una de ellas es 'Auto Place', función que analiza la posición correcta en la que situar un robot para alcanzar un grupo de puntos objetivo dado. También se puede utilizar en relación con una tarea. Considera entre otros parámetros los posibles choques o si los puntos son alcanzables ('Clash',

'Partial Reach' y Out of limit'). Por supuesto, es una opción útil en cualquier situación; sin embargo, puede argumentarse que no siempre se darán las condiciones idóneas de uso de esta función. En el caso de que el robot tenga que situarse en unos ciertos lugares en un espacio delimitado ya dado o si son varias tareas distintas que se prevé puedan cambiar, el uso de este comando puede ayudar al usuario, pero perderá eficacia. De esta forma, es responsabilidad final del ingeniero elegir la ubicación del robot.



Figura 27: Uso de la función 'Autoplace' en relación a las tareas del robot posicionador

Otra función muy conveniente asiste al usuario para remediar el problema de alcanzabilidad: 'Reach Function'. Este comando comprueba que los puntos-objetivo definidos para una tarea son alcanzados por un robot ya situado en el espacio elegido. En los otros grupos de puntos cuyo alcance a evaluar es por parte del robot posicionador también se cumple. Para satisfacer este requisito todos y cada uno de los puntos del grupo tienen que estar en el rango de alcance del robot; si uno sólo no lo está, el grupo entero da error. La potencia de dicho comando es servir de él antes de empezar a enseñar al robot las órdenes con el comando 'Teach'. De lo contrario, puede que algún punto no esté dentro del rango del robot y se descubra manualmente cuando se intenta llevar al mismo a la posición: en ese caso, tal vez se imponga empezar todo de nuevo.

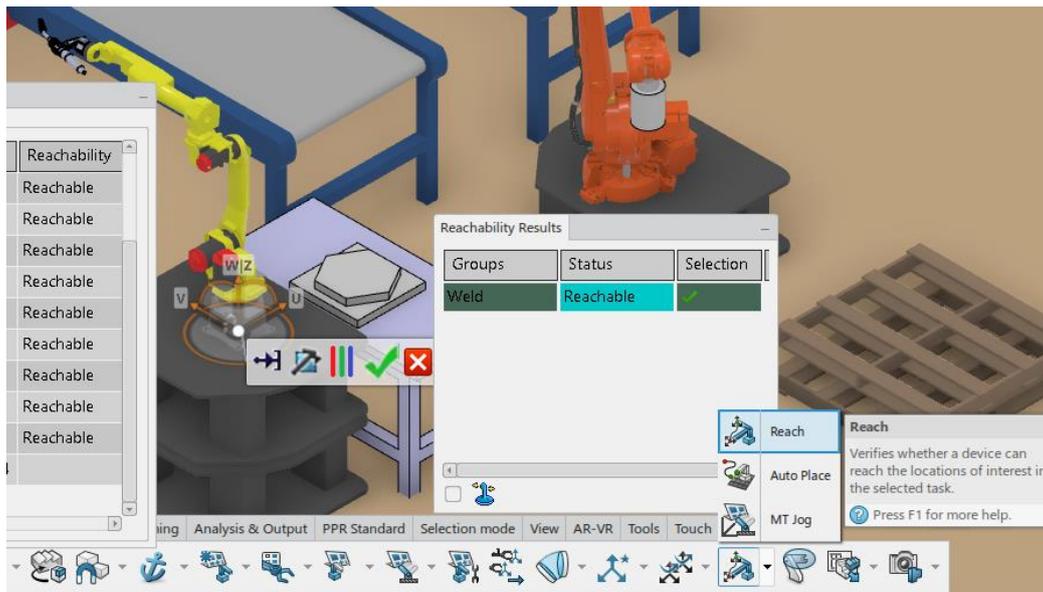


Figura 28: Comprobación de la accesibilidad de los puntos del grupo 'Weld' por parte del conjunto robot soldador - herramienta de soldadura.

El programa ofrece la posibilidad de actuar sobre cada articulación según los cálculos que surgen de resolver el problema cinemático inverso. La funcionalidad que se ocupa de este aspecto es 'Jog', a la que se puede acceder desde el panel del comando 'Teach a Device'. Con ella podemos tanto manipular el robot 3D para arrastrar el TCP donde deseemos, como editar los parámetros de desplazamiento y giro relativos numéricamente en un panel como el de la imagen o como editar los valores de giro de las articulaciones. Es evidente que los dos primeros casos pueden emplearse más bien para observar distintas posibilidades, más como método de ensayo y error, mientras que es la tercera la que propiamente puede aplicar los resultados del problema cinemático. Sin embargo, el software mismo es el encargado de aplicar los cálculos continuamente cuando se le ordena moverse a los distintos puntos objetivo, algo que ahorra una ingente cantidad de trabajo. Dentro del panel de 'Jog' se incluyen opciones para encontrar el punto más cercano posible a otro inalcanzable por el robot o definir el TCP, entre otros.

El control de las articulaciones en la realización de operaciones, más concretamente en la de soldadura, es indispensable, pues las zonas alejadas pueden dar problemas si el robot no alcanza a posicionarse correctamente. Para el robot posicionador no tenemos esa duda en principio, dado que el su envolvente cubría visiblemente los objetivos sin tener que llegar con una orientación desfavorable, aunque se comprueba de todos modos.

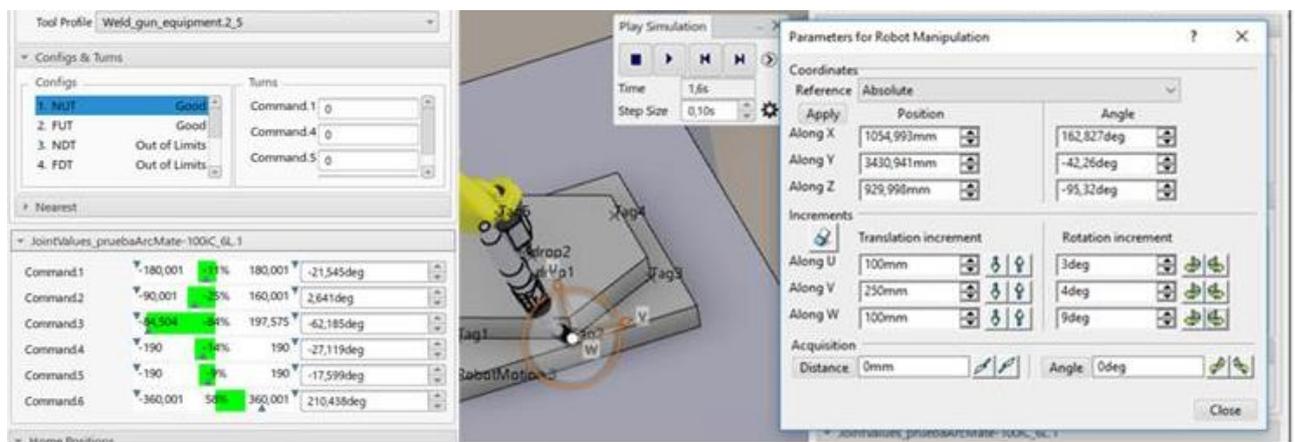


Figura 29: Control de las articulaciones del robot durante la tarea

En esta imagen se observa al robot soldador que ha sido arrastrado hasta la posición correspondiente al Tag2, con el giro de articulaciones que se muestra inmediatamente debajo. Los valores deben estar entre los límites permitidos. Para algunos movimientos esto no ocurría, lo que significa que la articulación ha sobrepasado su límite. En estos casos se adecuaba la orientación con el panel 'Parameters for Robot Manipulation' mostrado en la imagen hasta dar con la posición correcta. Para mayor afinamiento, puede intentar alejarse lo máximo posible los valores de las articulaciones de los extremos, para que el robot sufra lo mínimo posible.

En cuanto a la eficiencia, debe entenderse en este contexto como el número mínimo de rotaciones de los ejes del robot y el mínimo recorrido indispensable para realizar la operación propuesta. La función 'Set Turn

Number' se ocupa de analizar las distintas configuraciones para alcanzar esta eficiencia.

4.4.4 Señales de entrada y salida de los controladores

Las conexiones IO (Input/Output) son señales que entran o salen del controlador del robot y de las que el usuario se puede valer para controlar la secuencia de funcionamiento en una operación. Son por tanto la clave en la organización de tareas y recursos en aquellos medios que necesiten una sucesión de operaciones como pueden ser líneas de producción, celdas de fabricación, estaciones de trabajo...

Pueden crearse los siguientes tipos de IO globales: booleanas, enteras, doubles y cadena de caracteres. Cuando se crean debe declararse de igual forma su dirección y su valor por defecto. De la misma forma y con el mismo rango de posibilidades se pueden generar variables para el robot.

Para asignar instrucciones IO hay que relacionarlas con algún servicio-tarea, sea antes o después de realizar el movimiento de la tarea.

Las señales IO generadas se listan a continuación:

Tabla 6. Señales IO del controlador del robot soldador

Name	Direction	Type	Default Value	Port Number
Readyweld	In	Boolean	false	
Finishweld	Out	Boolean	false	

Tabla 7. Señales IO del controlador del robot posicionador

Name	Direction	Type	Default Value	Port Number
------	-----------	------	---------------	-------------

isconvready	In	Boolean	false	
isconvready2	In	Boolean	false	
weldfinish	In	Boolean	false	
ABBisready	Out	Boolean	false	
ABBisready2	Out	Boolean	false	
weldready	Out	Boolean	false	

Tabla 8. Señales IO del controlador de las cintas #1.1 y #1.2

Name	Direction	Type	Default Value	Port Number
isABBready	In	Boolean	false	
convready	Out	Boolean	false	

Tabla 9. Señales IO del controlador de las cintas #2.1 y #2.2

Name	Direction	Type	Default Value	Port Number
isABBready2	In	Boolean	false	
convready2	Out	Boolean	false	

El número de puerto se deja vacío, pero se le daría el valor correspondiente si se va a cargar el programa en un controlador físico real.

Una vez están las señales creadas y asignadas a los distintos recursos de la planta, el siguiente paso es mapearlas para sincronizar todos los elementos que participan en la secuencia de tareas. Así se da lugar a 'IO states' que permiten usar una señal de sincronización para el proceso, donde múltiples programas asociados a distintos recursos se ejecutan en paralelo. El entorno de mapeado es gráfico, lo que facilita la programación. Como es obvio, una señal de salida en un elemento debe conectarse a otra

de entrada del siguiente y viceversa, siendo incorrecto el mapeo de dos de entrada o dos de salida.

El mapeado que se establece entre ellas es el siguiente:

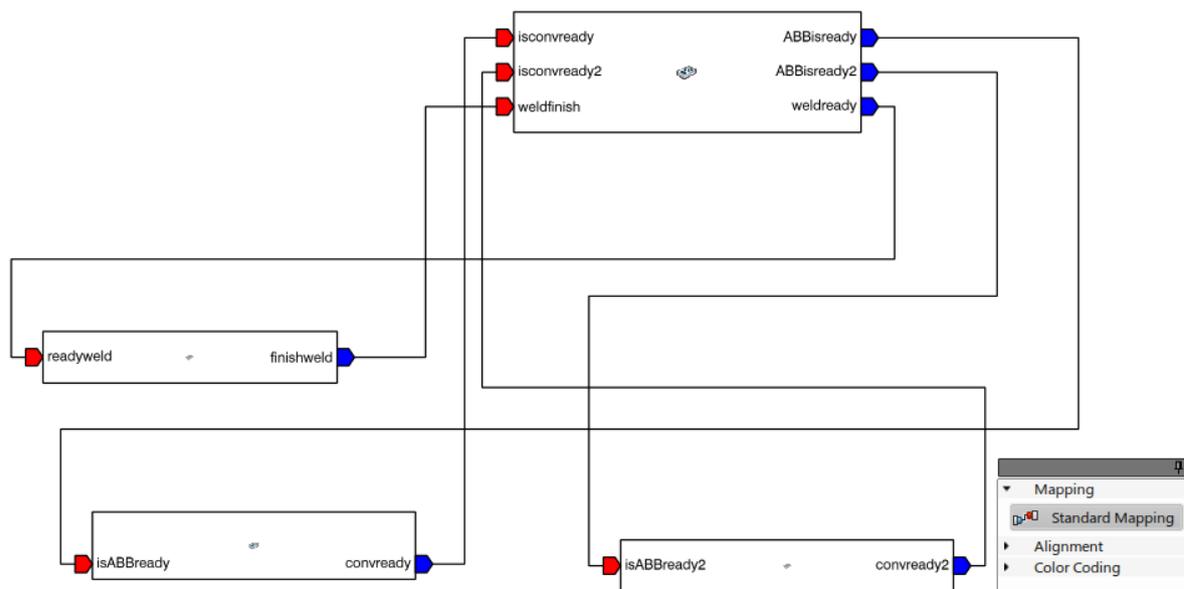


Figura 30: Mapeado de las señales IO de la planta

4.4.5. Secuenciación

Posteriormente al mapeado de las señales y su asociación al servicio-tarea correspondiente, debe crearse una tarea principal que esté encima de las tareas-servicios en la jerarquía de organización. Esta tarea principal no puede ser llamada por otra y se utiliza para llamar a las otras; puede definirse como la tarea 'por defecto' asociada a un recurso de la planta. Con ella, posicionando las tareas de los recursos subordinados de la planta en la forma correcta se consigue el funcionamiento de la tarea principal. A forma de comparación, las otras tareas pueden considerarse subrutinas del código principal que da cuerpo al programa o actividad completa.

En el momento adecuado, esas señales se activan indicando al controlador que las recibe que puede ordenar la tarea indicada, combinadas con el

contador var. Para conseguir esto se utiliza la instrucción *wait* y la asignación de valores a las señales en el momento oportuno del árbol de órdenes de tareas:

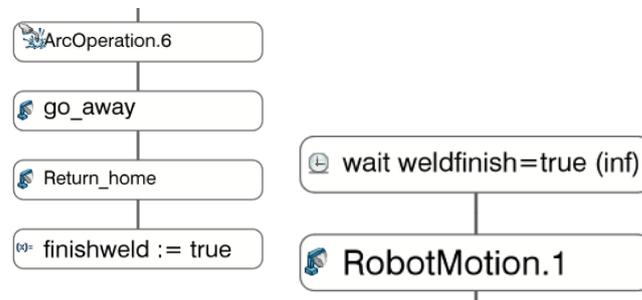


Figura 31: Muestra de órdenes de espera y asignación incluidas en las tareas

En la imagen se muestran las instrucciones añadidas para la tarea Pick_drop_pal1. A la izquierda la orden inicial: la tarea solo comenzará si la señal weldfinish tiene valor *true*. Atendiendo al mapeado mostrado, es una señal de entrada del controlador *Robot_ControlDevice.1* conectada a la señal de salida 'finishweld' del controlador *LayoutRobot Controller.1*. De la misma manera que aquí se utiliza la asignación de las señales que van a los controladores de las cintas (derecha de la imagen), en la tarea 'Weld' se cambia momentáneamente el valor una vez se ha finalizado de la señal weldfinish a valor *true*, y luego vuelve a su valor por defecto, *false*.

Tal es la forma de controlar la secuencia de la planta: antes de ejecutar una tarea, se comprueba que las señales correspondientes están activadas, lo que quiere decir que la acción previa ha finalizado. Al terminar, se envía la señal para que active la siguiente tarea.

Todo esto tiene que hacerse llamando a las tareas servicio en la tarea principal. Esta tarea se genera para toda la celda de trabajo. La secuencia es directa, pudiendo establecer ramas paralelas si se desea, debido a que la jerarquía está controlada por las conexiones IO. En la imagen se muestra un ejemplo de disposición con forma mixta:

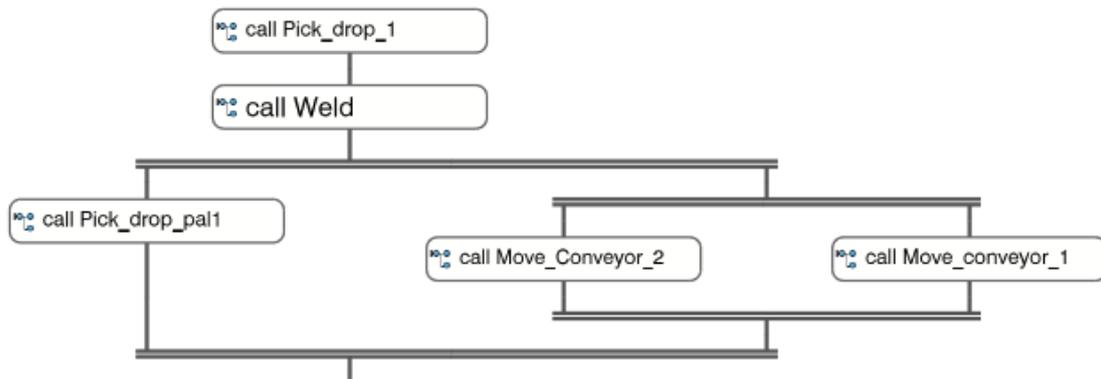


Figura 32: Composición de la tarea principal

Estas son las tareas base, con las que se consigue un ciclo de trabajo de célula. Para seguir con los siguientes tres ciclos y llenar el pallet, se puede utilizar un bucle añadiendo una variable que avise del número de piezas que hay en el pallet para acomodar el suelte de la pieza donde corresponda. También pueden generarse tareas a continuación copiando las anteriores y cambiando las operaciones necesarias.

La retirada del pallet completo se realizará mediante una carretilla elevadora. El proceso se repetiría inicializando la simulación de nuevo una vez se ha llenado el pallet, colocando otro en el mismo lugar y dando el tiempo prudencial necesario para activar la planta de nuevo (mediante una señal externa, por ejemplo).

Con esto la célula de fabricación está prácticamente acabada. Únicamente restan algunos detalles que se tratarán con la siguiente aplicación.

4.5. Programación Robot



Esta herramienta permite traducir las distintas tareas V6 del robot a programas específicos en lenguaje de control de robots y viceversa. Con ella se pueden cargar programas creados offline a controladores robots para aplicaciones de planta real o descargar uno ya existente para validarlo o editarlo. Habilita de igual manera a calibrar la celda de trabajo virtual empleando el formato Visual Basic para importar o exportar los programas, de manera que coincida con la celda de trabajo real.

4.5.1. Traductores

El elemento fundamental para conseguirlo son los Traductores OLP. Existen distintos tipos de traductores asociados a distintos controladores según qué tipo de robots se controlen (ABB, FANUC, ACMA, MOTOMAN...). Por otro lado, no es necesario usar los traductores predeterminados de los fabricantes, sino que la arquitectura abierta permite crear traductores propios. Una vez son cargados en la base de datos permanecerán ahí para posteriores sesiones y para todos los usuarios, pudiendo establecer las preferencias en el menú de configuración.

La acción de traducir las tareas del robot en el lenguaje nativo del robot (NRL) se denomina 'descarga', mientras que su antónimo sería la 'carga'. Esas tareas tendrán unas etiquetas asociadas en coordenadas x, y, z. A continuación, se ofrece una lista de los tipos de actividades que pueden descargarse o cargarse:

- ➔ Movimientos del robot (centro coordenado auxiliar, fija TCP (Tool Center Point), movimiento circular, centro de coordenadas en la muñeca, 'home position').
- ➔ Operaciones de arco y punto
- ➔ Instrucciones de agarre o suelte

- ➔ Asignaciones de variables o señales IO
- ➔ Instrucciones de espera o pulso
- ➔ Instrucciones de ejecución de tareas
- ➔ GOTO and label instructions ¿?
- ➔ Instrucciones de condicionales y bucle (if, for, while loop)
- ➔ Instrucciones Return and break
- ➔ Comentarios
- ➔ Instrucciones de tiempo
- ➔ Solo para FANUC y MOTORMAN, macros

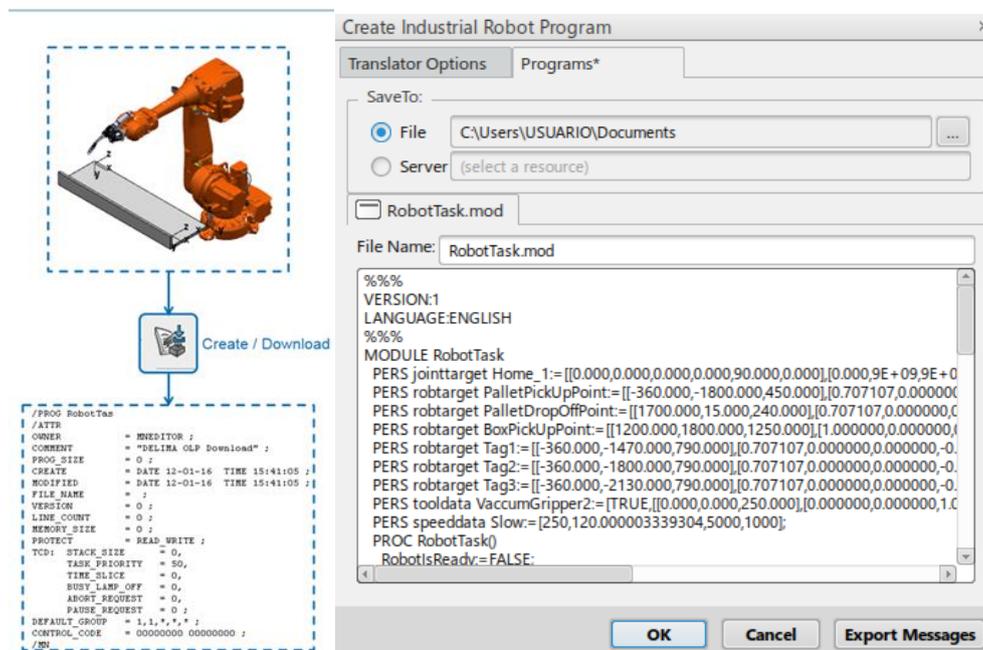


Figura 33: Descarga de un programa robot

El propio traductor emite mensajes durante la descarga si hay algún problema. Si se da el caso de cargar en el programa instrucciones que no se reconocen para la simulación, serán ignoradas. Téngase en cuenta que también se pueden traducir atributos de estas actividades tales como la precisión deseada o la velocidad de movimiento entre otras.

En el anexo se han añadido los programas en lenguaje NRL. Los traductores empleados para descargar esos programas han sido el DELMIA RAPID

Translator R2018 para el robot ABB y FANUC para el robot FANUC, el DELMIA FANUC Translator 2018x.

4.5.2. 'Device profiles' o perfiles de los dispositivos.

Estos perfiles dan información sobre el movimiento de un robot o elemento. Los controladores de los robots tienen todos por defecto perfiles de movimiento, herramienta, precisión u objetos. Además de estos, el usuario puede crear otro como un perfil práctico. A continuación, se detallan brevemente:

- Perfiles de precisión (Accuracy profiles): se utiliza para definir la precisión de una trayectoria determinada. Si se desea una precisión distinta para distintas trayectorias se deben definir distintos perfiles. La precisión puede especificarse en función de la distancia o de la velocidad.
- Perfiles de movimiento (Motion profiles): permiten definir el tipo, velocidad y aceleración del robot al que se asigne. La velocidad se puede especificar en valor absoluto, en porcentaje de la velocidad máxima de la que puede desarrollar el robot o en unidades de tiempo.
- Perfil de herramienta (Tool profile): con él se definen características de la herramienta que se acopla al robot, como pueden ser el peso o el desfase del TCP. Estas características afectan el comportamiento del robot.
- Perfil objeto (Object or reference profile): permite definir el marco usado como base del robot, eligiendo unos valores iniciales distintos a cero según la posición inicial del robot y resolviendo con esos datos el problema cinemático inverso.
- Perfil práctico (Applicative profile): contienen información requerida para el proceso de soldadura que, al aplicarse a una tarea de robot de este tipo, definen los retrasos y velocidad propios de la acción de soldar. Estas características se reflejan en la simulación.

Los más relevantes que se han empleado para esta ocasión han sido los de precisión, los de movimiento y uno práctico para la soldadura. Los perfiles de herramienta utilizados son los creados por defecto al generar la unión robot-herramienta, ya que el TCP queda definido con el puerto elegido. De igual manera no se realizan cambios en la posición inicial de reposo del robot que necesiten saberse para el cálculo del problema cinemático inverso, con lo que los perfiles objeto serán también los creados por defecto.

En cuanto al perfil práctico, se ha supuesto un retardo al empezar la soldadura y terminarla de medio segundo y un segundo, respectivamente. La velocidad con la que avanza la pistola soldando es de 100mm/s y se especifica una precisión de distancia de 0mm durante la trayectoria. La velocidad de los movimientos para posicionarse antes y después de la operación de soldadura se ha dejado en un 50% de la velocidad máxima del robot (1m/s), el valor por defecto.

Para el robot posicionador ha resultado suficiente con una precisión más basta en general excepto en los movimientos que involucran la recogida o el suelte de las piezas, donde se ha especificado que la precisión en la distancia sea total, con una tolerancia de error de 0mm. Se definen tres perfiles de movimiento: 'slow', 'medium', 'fast'. El primero desarrolla el 10% de la velocidad máxima (0.15m/s) y se utiliza en el acercamiento a los puntos de recogida y suelte de piezas. El segundo, el 50% de la velocidad máxima (1m/s) y se utiliza cuando el robot tiene carga, para no cargar excesivamente las articulaciones. El tercero aumenta al 90% (1.8m/s), utilizándose en todos los demás movimientos.

Para las cintas transportadoras solo se emplea el perfil de movimiento. Se especifica un tiempo de dos segundos para mover las piezas.

4.5.3. NRL Teach.

Este comando permite ver y editar programas robot escritos en NRL, posibilitando ver la sintaxis y crear entidades como las tareas del robot. Es una funcionalidad útil y cómoda de usar, ya que moviendo el robot a los puntos deseados lo 'enseñamos', añadiendo este punto objetivo al programa y pudiendo luego añadir la información complementaria necesaria.

A partir de aquí se tratan aspectos sobre la simulación teniendo la celda real. Debido al coste que supone realizarla, no ha podido practicarse lo descrito a continuación. No obstante, considero importante reflejar al menos de forma teórica el funcionamiento de diversos asuntos clave que complementan la simulación de la planta conseguida.

4.5.4. Aspectos relativos a una simulación realista (RRS- Realistic Robot Simulation)



Figura 34: Concepto de control del robot real mediante simulación por ordenador de su gemelo virtual [7]

Este comando habilita una mejor precisión tanto en trayectorias como en el tiempo de ciclo utilizado, conectando la simulación con el módulo RCS (Robot Controller Simulation) y con el software de planificación de movimiento del fabricante del robot. La realimentación ofrece datos de análisis facilitando la obtención de perfiles de movimiento precisos al tiempo que el robot real realiza sus tareas. Mediante la integración del software RCS con el de simulación, podemos conseguir:

- ◆ Mejorar los análisis en tiempo real de ciclos de trabajo y detectar posibles colisiones más eficazmente.
- ◆ Controlar y simular de forma simultánea múltiples robots y recursos, como pueden ser posicionar distintos dispositivos o la modelización de personas.
- ◆ Mantener robots con ejes externos. Soporta los estandarizados de 4 y 6 ejes incluyendo los de ejes externos – eje de la herramienta y sistemas de robot sobre raíl.
- ◆ Estudios más precisos del alcance del robot utilizando el cálculo de cinemática inversa del controlador.

4.5.5. Métodos de calibración.

Las características que ofrece el software para calibrar permiten modificar aspectos de la planta virtual y la simulación para ser un reflejo preciso de la planta real. Identifica las causas de imprecisión en la posición de los elementos para corregirlas, sea por parte de dispositivos, robots, herramientas-robot u otros. Adicionalmente, estos aspectos permiten utilizar una misma simulación en plantas de trabajo a grandes rasgos iguales, en las que varían ligeramente las posiciones de elementos o los offsets del robot, por ejemplo.

La calibración del TCP ('Tool Point Calibration') es un método para calibrar el perfil de herramienta de un robot basado en la medida de múltiples puntos. Estos puntos, que formarán parte de un 'tag group' (véase más

adelante) han sido en principio generados en el soporte de fijación (0,0,0,0,0,0) mientras se movía el extremo de la herramienta a un punto concreto del espacio con diferentes orientaciones de la muñeca del robot. Si se desea se puede aportar una aproximación de los parámetros del TCP, aunque se debe tener la seguridad de que son valores cercanos al correcto, con errores de pocos centímetros. Posteriormente se especifican los componentes que deben ser calibrados (X, Y, Z) y se aporta una estimación de ruido, es decir, la confianza o incertidumbre de las medidas durante el experimento. En base a esto el perfil es ajustado para conseguir que el extremo de la herramienta se mueva al mismo punto del espacio en cada una de las diferentes orientaciones de la muñeca introducidas.

Los resultados que arroja una calibración son los siguientes:

- Número de iteraciones requeridas por el método de identificación numérica.
- Número de puntos usados por el método de mínimos cuadrados.
- Distancia media cuadrática mínima (RMS), es decir, la raíz cuadrada de la media del error de posición entre los puntos seleccionados y el resultado final, supuesto como la mejor aproximación posible.
- Incertidumbre máxima de la estimación.

Un valor elevado de este último parámetro (del orden de $\pm 10\%$ o más) indicaría que la estrategia seguida en el experimento no es correcta, incluso si el RMS es pequeño.

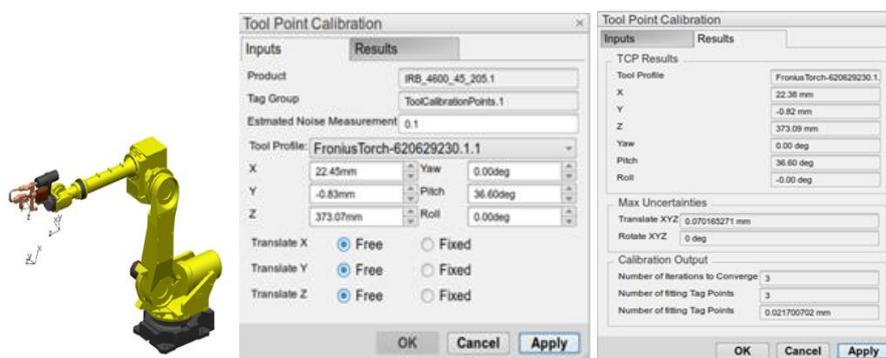


Figura 35: Ejemplo de calibración del TCP.

El método de mínimos cuadrados se utiliza para ajustar los valores de los seis grados de libertad de un dispositivo o elemento en la simulación basándose en datos obtenidos de la estación de trabajo. Se asume que el elemento no tiene propiedades dinámicas a efectos de la planta, siendo objetos fijos tales como una mesa de trabajo. El propósito de la calibración es localizarlo en la simulación de manera que sus parámetros (X, Y, Z, Yaw, Pitch, Roll) relativos respecto al robot coincidan con los reales. Las tareas desarrolladas posteriormente a la calibración contendrán las posiciones corregidas.

Los datos de entrada son similares al anterior método, siendo las dos diferencias que ahora los datos de orientación no son fijados, sino que pueden calcularse y que se toman dos grupos de puntos como base: los 'puros' que localizan los archivos CAD (es decir, los de la planta simulada) y los experimentales de la planta real, que se cargan al programa. El método trata de minimizar el error de posición manteniendo lo más constante posible cada parámetro.

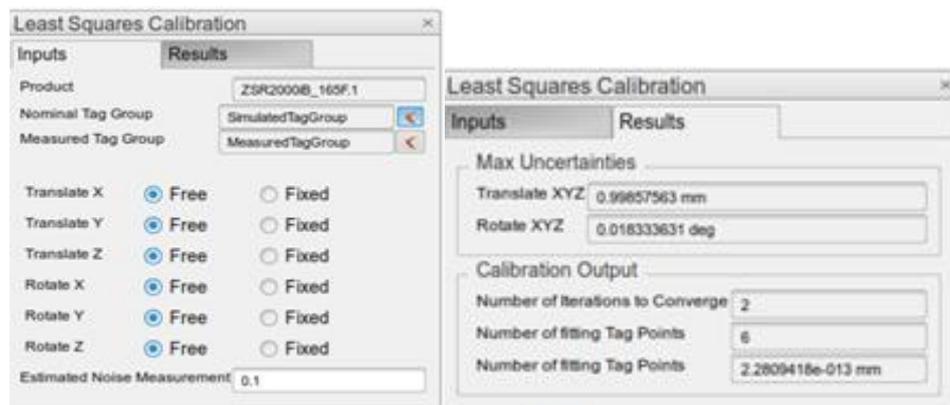


Figura 36: Ejemplo del método de calibración de mínimos cuadrados

Otro procedimiento de calibración es llamado en inglés Six Point Calibration Method. Es indicado cuando el producto tiene tres superficies ortogonales (como puede ser una forma cúbica) y no se tienen al menos tres puntos significativos que pueden ser fácilmente medidos. Para emplearlo, deben

introducirse seis puntos (tres para definir el primer plano, dos para el segundo y otro para el restante) que especifiquen la posición en el mundo virtual del objeto y seis que sean los medidos experimentalmente.

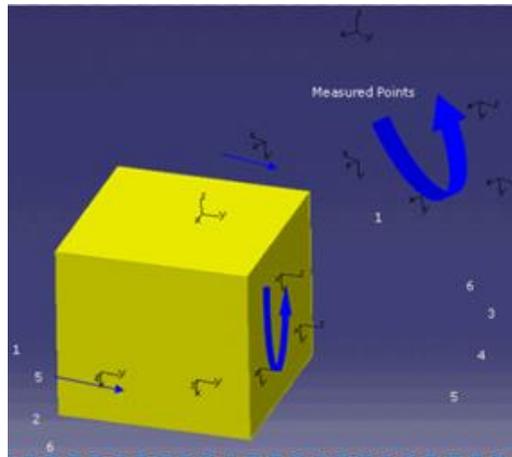


Figura 37: 'Six Point Calibration Method'

El comando traslada el objeto haciendo coincidir los primeros seis puntos con los del segundo grupo

De igual manera se puede calibrar tanto la posición base de un dispositivo posicionador de objetos como su herramienta (de tenerla) respecto a un robot. Se necesita para ello el grupo de puntos a corregir que especifican esos parámetros, el grupo de puntos correctos del entorno real, el robot, el posicionador que se quiere calibrar y un objeto fijo auxiliar.

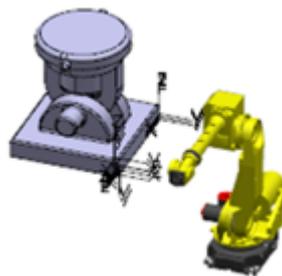


Figura 38: Robot y posicionador calibrados

También es posible calibrar o corregir la dimensión que ocupa el robot en la planta virtual (robot signature), ya que a veces sus valores longitudinales

o los offset en las uniones no son correctos, lo que llevaría a resultados erróneos de la simulación. Para mayor corrección se pueden introducir valores de error de movimiento mediante los vectores Delta, de manera que la imprecisión en el movimiento por los cálculos de la cinemática directa y/o inversa sea minimizada.

4.6- Vinculación de las aplicaciones en tiempo real

Supóngase que en un estadio más avanzado del ciclo de vida del producto se llega a la conclusión de que la base cuadrada debe cambiar y en lugar de tener vértices, debe realizarse un empalme. La persona encargada del diseño de piezas realizaría las transformaciones necesarias, que se trasladarían automáticamente, reflejándose allí donde esa pieza se estuviese empleando. Los nuevos parámetros se actualizan y se vuelve a comprobar si el diseño cumple los requisitos necesarios; de no ser así, se reformula el producto.

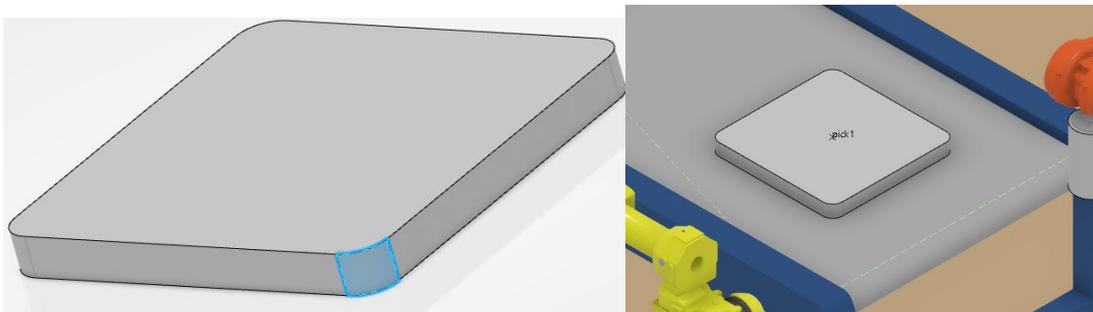


Figura 39: Muestra de la conexión entre aplicaciones de 3DExperience.

Incluso puede llevarse un paso más allá: en equipos multidisciplinares, a la vez que se está diseñando el producto, otro profesional trabaja en la nube analizando, por ejemplo, su estética, y aporta sus ideas y puntos de vista. Esto llega tanto a la persona encargada del diseño como al encargado de garantizar que el producto cumple con las prestaciones mecánicas que debe desempeñar, quien matiza un aspecto que, aunque embellece y hace más

atractivo el producto, compromete su resistencia a la compresión. Por otro lado, se barajan distintos materiales que parecen similares en cuanto al coste y características, pero la persona de la empresa de reciclaje en la que acabará el producto tras su vida útil analiza las opciones y concluye que es mejor utilizar uno sobre los demás, pues el impacto ambiental se reduce.

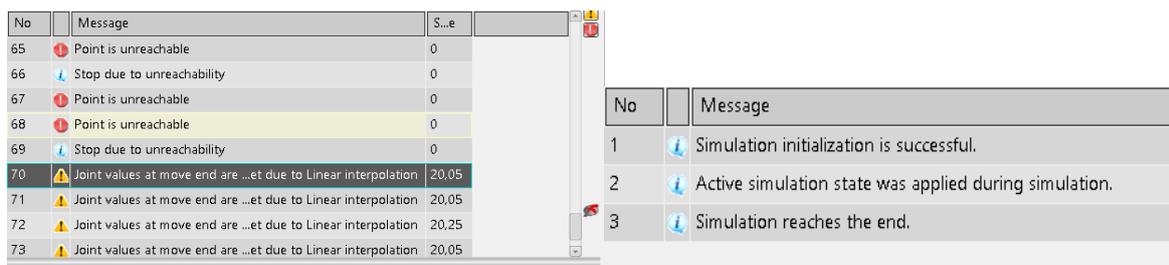
La ficción descrita es un ejemplo de ingeniería concurrente posibilitada y potenciada por las virtudes de la plataforma cooperativa. La sencillez del problema que aquí se ha tratado no limita sus eventuales aplicaciones; más bien al contrario, la diferencia con la ingeniería secuencial será considerablemente superior con productos más complejos. La consecuencia es clara: mayor calidad del producto en todas sus fases del ciclo de vida y menor tiempo y coste empleado, lo que se traduce en un aumento de la eficiencia.

5. SIMULACION

El paso final es simular la planta con objeto de comprobar su comportamiento para anticipar de la forma más fidedigna posible lo que ocurriría con el sistema real.

Para ello se utiliza, una vez seleccionada la tarea que queramos observar, el 'Simulation Player', al alcance de un click en el botón central del 'Compass'. Según se desee, se puede analizar una tarea específica de un robot, una parte de ella o la tarea principal completa de la célula. Esta última es la que reviste mayor interés, puesto que mostrará si la lógica y relaciones entre los recursos es adecuada. De confirmarse puede utilizarse la programación desarrollada en el entorno virtual en la célula de fabricación física que se construya, ya que el lenguaje de programación NRL (Native Robot Language) es universal.

Los distintos estados de simulación que se han guardado con el avance de la construcción de la celda sirven para sentar los escenarios iniciales de la simulación. Gracias a ellos se pueden realizar distintas pruebas, comprobando los resultados con pequeñas variaciones de las tareas. Estas pruebas se han ido realizando conforme avanza la programación de tareas, probando los módulos de operaciones que se van generando. De esta manera se va verificando y corrigiendo en caso de ser necesario, teniendo además la información del tiempo de cada acción. Cuando están todas las tareas construidas se simula el conjunto para chequear la lógica de la célula.



The image shows a screenshot of a simulation log window. It contains two tables of messages. The left table shows error messages related to unreachable points and joint values, while the right table shows success messages for simulation initialization, state application, and completion.

No	Message	S...e
65	Point is unreachable	0
66	Stop due to unreachability	0
67	Point is unreachable	0
68	Point is unreachable	0
69	Stop due to unreachability	0
70	Joint values at move end are ...et due to Linear interpolation	20,05
71	Joint values at move end are ...et due to Linear interpolation	20,05
72	Joint values at move end are ...et due to Linear interpolation	20,25
73	Joint values at move end are ...et due to Linear interpolation	20,05

No	Message
1	Simulation initialization is successful.
2	Active simulation state was applied during simulation.
3	Simulation reaches the end.

Figura 40: Mensajes de error por punto inalcanzable y de éxito de simulación

Son cinco las comprobaciones fundamentales que deben examinarse en estas simulaciones. Primero debe revisarse si realmente todas las tareas son factibles, es decir, si los robots pueden llegar a todos los puntos de las trayectorias. En segundo lugar, hay que examinar que no hay choque entre elementos durante el desarrollo de las tareas. Además, hay que cerciorarse de que los robots no son forzados a sobrepasar su límite cinemático o dinámico (como por ejemplo, giro desmedido de las articulaciones o fuerzas que excedan el límite). Igualmente se tiene que constatar que la lógica de la estación es adecuada, evitando posibles bucles o tiempos muertos. Por último, se certifica el tiempo de ciclo.

Tras varias pruebas con errores, los cuatro primeros puntos se consiguieron cumplir, siendo la célula corregida la que se ha mostrado a lo largo de este documento. Con los perfiles de movimiento de los robots definidos (apartado 4.5) se consigue un tiempo de simulación algo inferior a 200 segundos, entendiendo como tal el proceso completo que culmina con cuatro productos soldados encima del pallet. Eso indica que el tiempo de ciclo es casi 50 segundos de media, con lo que la especificación se cumple al estar por debajo del minuto. Nótese que al ser paralelas las tareas de las cintas y la del robot posicionador, el tiempo que estas tarden no importa mientras esté por debajo del tiempo que emplee el robot.

Si el ciclo de trabajo de la célula estuviese por debajo de los requerimientos, habría que corregir los movimientos para conseguir el objetivo. Una forma puede ser aumentar la velocidad de movimiento. Otra disminuir de alguna forma el número de trayectorias si existiesen redundantes, con un cuidadoso estudio comprobando hasta qué punto puede ser esto posible. De no funcionar ninguna de estas opciones, podría considerarse un cambio de diseño: elegir otros robots y otra disposición de elementos que garantice una mejora del tiempo de ciclo.

Existe una función que permite actuar sobre robots individuales para observar cual sería el comportamiento de la planta si uno de ellos fallase, la pausa selectiva ('selective pause'). Una funcionalidad con la que se puede

poner el colofón a la comprobación y simulación de la celda de trabajo. Debido a que se han establecido las conexiones IO entre los controladores, la célula simplemente se para justo antes de la acción que tuviese que desempeñar el robot pausado.

6. CONCLUSIONES

Tras la realización de este proyecto podemos considerar demostrado que la herramienta software 3DExperience es totalmente válida para el diseño, programación y simulación de una célula de fabricación flexible robotizada en el entorno de un sistema PLM.

Las aplicaciones empleadas permiten no sólo la creación de cuerpos sólidos, sino el modelado dinámico 3D. Esto se combina con el control cinemático de los robots mediante las técnicas de la robótica industrial y la programación, tanto de los robots como de elementos del entorno interactivos, para secuenciar las tareas. Además, incluye la simulación mediante un controlador virtual interno con buena renderización y cálculo del tiempo de ciclo. El resultado es un análisis predictivo fiable del comportamiento de la célula física.

Asimismo, la conexión de todas las aplicaciones en la plataforma permite el trabajo paralelo en las distintas fases de vida del producto, ya que la vinculación de las diferentes tareas se hace en tiempo real, optimizando los procesos mediante el ahorro de tiempo y recursos. En consecuencia, se aumenta la calidad del producto y se reduce el impacto medioambiental.

Una peculiaridad notable es la capacidad de utilizar robots de distintos fabricantes, un hecho que muchos programas no consiguen, sea por limitaciones técnicas o porque están asociados a una única marca. Poder importar robots virtuales que existen en el mercado completamente funcionales desde catálogos del propio programa constituye una ventaja incomparable.

Una de las características clave de 3DExperience es su complejidad, condición que se explica en gran medida al ser tan completo. Dassault Systèmes actualiza periódicamente las aplicaciones de la plataforma, renovando y cambiando algunos aspectos, por lo que la formación continua es indispensable para su correcto uso. Aunque su estudio sea una tarea ardua, las competencias adquiridas merecen plenamente el esfuerzo.

7. ANEXO: programas robots:

Programación del robot FANUC:

```
/PROG Arc_weld
/ATTR
OWNER = MNEDITOR ;
COMMENT = "DELMIA OLP Download" ;
PROG_SIZE = 0 ;
CREATE = DATE 19-01-31 TIME 21:46:44 ;
MODIFIED = DATE 19-01-31 TIME 21:46:44 ;
FILE_NAME = ;
VERSION = 0 ;
LINE_COUNT = 0 ;
MEMORY_SIZE = 0 ;
PROTECT = READ_WRITE ;
TCD: STACK_SIZE = 0,
TASK_PRIORITY = 50,
TIME_SLICE = 0,
BUSY_LAMP_OFF = 0,
ABORT_REQUEST = 0,
PAUSE_REQUEST = 0 ;
DEFAULT_GROUP = 1,*,*,* ;
CONTROL_CODE = 00000000 00000000 ;
/MN
1: UFRAME_NUM = 0 ;
2: UTOOL_NUM = 1 ;
3: J P[1] 50% FINE Arc Start[1] ;
4: L P[2] 1000mm/sec FINE ;
5: L P[3] 1000mm/sec FINE ;
6: L P[4] 1000mm/sec FINE ;
7: L P[5] 1000mm/sec FINE ;
8: L P[6] 1000mm/sec FINE ;
9: L P[7] 1000mm/sec FINE Arc End[1] ;
10: L P[8] 1000mm/sec FINE ;
11: J PR[1:New Home.2] 50% FINE ;
/POS
P[1]{
GP1:
UF : 0, UT : 1, CONFIG : 'F U T, 0, 0, 0',
X = 1777.464 mm, Y = 1584.527 mm, Z = -450.001 mm,
```

```

W = -170.359 deg, P = -25.444 deg, R = 32.704 deg
};
P[2]{
GP1:
UF : 0, UT : 1, CONFIG : 'N U T, 0, 0, 0',
X = 1977.464 mm, Y = 1469.056 mm, Z = -449.999 mm,
W = 0.000 deg, P = -145.000 deg, R = -120.000 deg
};
P[3]{
GP1:
UF : 0, UT : 1, CONFIG : 'N U T, 0, 0, 0',
X = 2177.465 mm, Y = 1584.526 mm, Z = -449.998 mm,
W = 0.000 deg, P = -145.000 deg, R = -60.000 deg
};
P[4]{
GP1:
UF : 0, UT : 1, CONFIG : 'N U T, 0, 0, 0',
X = 2177.465 mm, Y = 1815.466 mm, Z = -449.998 mm,
W = 0.000 deg, P = -135.000 deg, R = 0.000 deg
};
P[5]{
GP1:
UF : 0, UT : 1, CONFIG : 'N U T, 0, 0, 0',
X = 1977.465 mm, Y = 1930.937 mm, Z = -450.000 mm,
W = 0.000 deg, P = -145.000 deg, R = 60.000 deg
};
P[6]{
GP1:
UF : 0, UT : 1, CONFIG : 'N U T, 0, 0, 0',
X = 1777.465 mm, Y = 1815.467 mm, Z = -450.002 mm,
W = 12.648 deg, P = -136.410 deg, R = 119.629 deg
};
P[7]{
GP1:
UF : 0, UT : 1, CONFIG : 'N U T, 0, 0, 0',
X = 1777.464 mm, Y = 1584.527 mm, Z = -450.001 mm,
W = -176.976 deg, P = -34.296 deg, R = -31.610 deg
};
P[8]{
GP1:
UF : 0, UT : 1,

```

```
J1 = -11.558 deg, J2 = -23.310 deg, J3 = -52.488 deg,  
J4 = -4.905 deg, J5 = -38.983 deg, J6 = 75.555 deg  
};  
/END
```

```
[*SYSTEM*]$MNUFRAME STORAGE: FAST_CMOS ACCESS: RW : ARRAY[1,0] OF POSITION
```

```
[*SYSTEM*]$MNUTOOL STORAGE: FAST_CMOS ACCESS: RW : ARRAY[1,1] OF POSITION
```

```
[1,1] =
```

```
Group: 1 Config: N D B, 0, 0, 0
```

```
X: 0.705 Y: 22.883 Z: 369.329
```

```
W: 0.000 P: 36.600 R: 90.000
```

```
[*POSREG*]$POSREG STORAGE: CMOS ACCESS: RW : ARRAY[1,1] OF POSITION
```

```
[1,1] = 'New Home.2' Group: 1
```

```
J1 = 90.000 deg J2 = 0.000 deg J3 = 0.000 deg
```

```
J4 = 0.000 deg J5 = 0.000 deg J6 = 0.000 deg
```

```
/PROG tooluser
```

```
/ATTR
```

```
OWNER = MNEDITOR ;
```

```
COMMENT = "DELMIA OLP Download" ;
```

```
PROG_SIZE = 0 ;
```

```
CREATE = DATE 19-01-31 TIME 21:46:44 ;
```

```
MODIFIED = DATE 19-01-31 TIME 21:46:44 ;
```

```
FILE_NAME = ;
```

```
VERSION = 0 ;
```

```
LINE_COUNT = 0 ;
```

```
MEMORY_SIZE = 0 ;
```

```
PROTECT = READ_WRITE ;
```

```
TCD: STACK_SIZE = 0,
```

```
TASK_PRIORITY = 50,
```

```
TIME_SLICE = 0,
```

```
BUSY_LAMP_OFF = 0,
```

```
ABORT_REQUEST = 0,
```

```
PAUSE_REQUEST = 0 ;
```

```
DEFAULT_GROUP = 1,*,*,* ;
```

```
CONTROL_CODE = 00000000 00000000 ;
```

```
/MN
```

```

1: ! WARNING THIS PROGRAM USES POSITION REGISTER 1 TO SET UFRAME AND UTOOL VALUES AT THE ROBOT ;
2: ! WARNING ONLY LOAD AND RUN THIS PROGRAM AT THE ROBOT IF THIS IS THE DESIRED RESULT ;
3: ! WARNING BE SURE TO SAVE SYSTEM VARIABLES FOLLOWING USE OF THIS PROGRAM OR VALUES WILL BE LOST ;
4: PR[1,1]= 0.70508658885956 ;
5: PR[1,2]= 22.8832952037592 ;
6: PR[1,3]= 369.328936748479 ;
7: PR[1,4]= 0 ;
8: PR[1,5]= 36.600009244439 ;
9: PR[1,6]= 90 ;
10: UTOOL[1]=PR[1] ;
/POS
/END

```

Programación del robot ABB.

```

%%%
VERSION:1
LANGUAGE:ENGLISH
%%%
MODULE Pick_drop_1
PERS jointtarget New_Home2:=[[0.000,0.000,0.000,0.000,90.000,0.000],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
PERS robtarget pick1:=[[-100.000,1500.000,1000.000],[0.707107,0.000000,0.707107,0.000000],[-1,0,0,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
PERS robtarget
drop1:=[[1055.000,3200.000,930.000],[0.707107,0.000000,0.707107,0.000000],[0,0,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
PERS robtarget pick2:=[[-2000.000,1500.000,1000.000],[0.707107,0.000000,0.707107,0.000000],[-2,0,-1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
PERS robtarget drop2:=[[1055.000,3200.000,980.000],[0.683013,0.183013,0.683013,-0.183013],[0,0,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
PERS tooldata Electroiman_equipment1:=[TRUE,[[0.000,0.000,242.000],[0.707107,0.000000,-0.707107,0.000000]],[0.000,[0.000,0.000,0.000],[1.000000,0.000000,0.000000,0.000000],0.000,0.000,0.000]];
PERS speeddata Fast:=[1350,324.000009016121,5000,1000];
PERS speeddata medium:=[750,360.000010017913,5000,1000];
PERS speeddata Slow:=[150,36.0000010017913,5000,1000];
PERS zonedata dist:=[TRUE,0,0,0,0,0];
PROC Pick_drop_1()
MoveAbsJ New_Home2,Fast,fine,Electroiman_equipment1;
MoveAbsJ [[-60.642,26.058,4.827,0.000,59.116,29.358],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],Fast,fine,Electroiman_equipment1;

```

```

MoveJ pick1,Slow,fine,Electroiman_equipment1;
Grab_square ;
Grab.18 ;
MoveAbsJ [[-
60.642,26.058,4.827,0.000,59.116,29.358],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],Slow,fine,Electroiman_equipment1;
MoveAbsJ [[2.786,38.421,-
10.222,0.000,61.801,92.786],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],medium,fine,Electroiman_equipment1;
MoveJ drop1,Slow,dist,Electroiman_equipment1;
Release_square ;
MoveAbsJ New_Home2,Fast,fine,Electroiman_equipment1;
MoveAbsJ [[-122.005,28.533,1.301,0.000,60.166,-
32.005],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],Fast,fine,Electroiman_equipment1;
MoveJ pick2,Slow,fine,Electroiman_equipment1;
Grab_hexagon ;
MoveAbsJ [[-122.005,28.534,1.301,0.000,60.166,-
32.005],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],Slow,fine,Electroiman_equipment1;
MoveAbsJ [[2.786,37.504,-
11.162,0.000,63.659,122.786],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],medium,fine,Electroiman_equipment1;
MoveJ drop2,Slow,dist,Electroiman_equipment1;
Release_hexagon ;
MoveAbsJ [[2.786,37.504,-
11.162,0.000,63.659,212.786],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],Fast,fine,Electroiman_equipment1;
MoveAbsJ New_Home2,Fast,fine,Electroiman_equipment1;
MoveAbsJ
[[50.000,0.000,0.000,0.000,90.000,0.000],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],Fast,fine,Electroiman_equipment1;
ENDPROC
PROC Pick_drop_pal1()
MoveAbsJ New_Home2,Fast,fine,Electroiman_equipment1;
MoveAbsJ [[2.786,35.913,-
13.385,0.000,67.473,92.786],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],Fast,fine,Electroiman_equipment1;
MoveJ
[[1054.977,3199.998,979.922],[0.707107,0.000000,0.707107,0.000000],[0,0,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]]
,Slow,fine,Electroiman_equipment1;
Grab_product ;
MoveAbsJ [[2.786,37.505,-
11.161,0.000,63.659,2.786],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],Slow,fine,Electroiman_equipment1;
MoveAbsJ New_Home2,medium,fine,Electroiman_equipment1;
MoveAbsJ
[[91.828,28.836,26.623,0.000,34.542,1.828],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],medium,fine,Electroiman equipme
nt1;
MoveJ [[-1050.000,4666.549,430.323],[0.500000,-

```

```
0.500000,0.500000,0.500000],[1,0,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],Slow,dist,Electroiman_equipment1;  
Release_product ;  
MoveAbsJ  
[[50.000,0.000,0.000,0.000,90.000,0.000],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],Fast,fine,Electroiman_equipment1;  
ENDPROC  
ENDMODULE
```

8. Lista de figuras, lista de tablas, referencias y bibliografía

Lista de figuras

Figura 1: Revoluciones industriales y conceptos propios de industria 4.0 [1] (página 6)

Figura 2: Palancas y facilitadores de la transformación digital [2] (página 7)

Figura 3: Bucle PDP y tecnologías relacionadas (página 8)

Figura 4: Cambio de la cadena de suministros tradicional a la digital [3] (página 9)

Figura 5: Ciclo de vida de un producto [4] (página 10)

Figura 6: 'Compass' con acceso a las aplicaciones de 3DExperience [5] (página 13)

Figura 7: Página de inicio de 3DExperience con acceso a las aplicaciones (página 15)

Figura 8: Ejemplo de escritorio digital [6] (página 15)

Figura 9: Ejemplos de roles, activado el primero y desactivados los demás (página 16)

Figura 10: Muestra de 'tree' o árbol de recursos disponibles y relaciones de subordinación entre ellos. (página 20)

Figura 11: Espacio de trabajo o envolvente de los robots (página 24)

Figura 12: Vista isométrica de la célula de trabajo (página 27)

Figura 13: Operación de extrusión en CATIA PART DESIGN (página 28)

Figura 14: Cálculo de parámetros dinámicos en CATIA PART DESIGN (página 29)

Figura 15: Situación de las entidades 'Manufacturing Cell' y 'Manufacturing Product' en el árbol de recursos (página 30)

Figura 16: Descripción del proceso de creación de una estación de trabajo con 'Robot Simulation'. (página 31)

Figura 17: Creación de puertos en una herramienta robot (página 32)

Figura 18: Fijación de la herramienta robot (página 33)

Figura 19: Controladores de movimiento asociados a recursos de una planta. (página 34)

Figura 20: Visualización de los puntos de trabajo creados (página 35)

Figura 21: Comandos de manipulación de 'tags' (página 36)

Figura 22: Arrastre de un robot a un punto con el comando 'Teach' (página 40)

Figura 23: Trayectoria seguida por el robot posicionador (tarea 'Pick_drop1') (página 41)

Figura 24: Detalle de la trayectoria seguida por el robot soldador (tarea 'Weld') (página 42)

Figura 25: Trayectoria seguida por el robot posicionador (tarea 'Pick_drop_pal1') (página 42)

Figura 26: Detalle de la estación de trabajo tras cuatro ciclos (página 43)

Figura 27: Uso de la función 'Autoplace' en relación a las tareas del robot posicionador (página 44)

Figura 28: Comprobación de la accesibilidad de los puntos del grupo 'Weld' por parte del conjunto robot soldador - herramienta de soldadura (página 45)

Figura 29: Control de las articulaciones del robot durante la tarea (página 46)

Figura 30: Mapeado de las señales IO de la planta (página 49)

Figura 31: Muestra de órdenes de espera y asignación incluidas en las tareas (página 50)

Figura 32: Composición de la tarea principal (página 51)

Figura 33: Descarga de un programa robot (página 53)

Figura 34: Concepto de control del robot real mediante simulación por ordenador de su gemelo virtual [7] (página 56)

Figura 35: Ejemplo de calibración del TCP (página 58)

Figura 36: Ejemplo del método de calibración de mínimos cuadrados (página 59)

Figura 37: 'Six Point Calibration Method' (página 60)

Figura 38: Robot y posicionador calibrados (página 60)

Figura 39: Muestra de la conexión entre aplicaciones de 3DExperience (página 61)

Figura 40: Mensajes de error por punto inalcanzable y de éxito de simulación (página 63)

Lista de tablas

Tabla 1. Características de los principales robots elegidos (página 23)

Tabla 2. Cantidad y disposición de los elementos de la celda (página 25)

Tabla 3. Parámetros de los puntos del grupo 'Weld_tags' (página 36)

Tabla 4. Parámetros de los puntos del grupo 'Pick_base' (página 37)

Tabla 5. Parámetros de los puntos del grupo 'palletize' (página 37)

Tabla 6. Señales IO del controlador del robot soldador (página 47)

Tabla 7. Señales IO del controlador del robot posicionador (página 47)

Tabla 8. Señales IO del controlador de las cintas #1.1 y #1.2 (página 48)

Tabla 9. Señales IO del controlador de las cintas #2.1 y #2.2 (página 48)

Referencias imágenes:

- [1] Gonzalez, B(2018). *Industria 4.0: una revolución para las personas*. Disponible en <https://www.youtube.com/watch?v=a0Ycxn-bZak>
- [2] Berger, R. (2016). *España 4.0: el reto de la transformación digital de la economía*. Roland Berger S.A. & SIEMENS
- [3] Cotteleer, M., Sniderman, B. (2017) *Forces of change: Industry 4.0*. Deloitte Insights.
- [4] Argos, B (2017) *Sostenibilidad ambiental en el diseño de proyectos, Lección cuatro*. Dpto. Transportes y Tecnología de Proyectos y Procesos UNICAN.
- [5] Página web de Dassault Systèmes sobre la plataforma 3DEXPERIENCE, URL: <https://www.3ds.com/es/acerca-de-3ds/plataforma-3dexperience/>
- [6] Página web de Dassault Systèmes sobre escritorios digitales, URL: <https://blogs.3ds.com/germany/3dexperience-das-dashboard-als-zentrale-anlaufstelle/>
- [7] Elaboración propia con imágenes de www.New.abb.com (controlador) y de https://an.wikipedia.org/wiki/Imachen:Robot_ABB_3.jpg (robots)

Bibliografía

- ABB. (2019). *ABB information systems LTD*. Obtenido de <https://new.abb.com/products/robotics/es/robots-industriales>
- Hecho en alemania (Deutsche Welle) (2014). *Industria 4.0: un desafío* [Documental].
- Altran. (2017). *Equipo Altran*. Obtenido de <https://equipo.altran.es/como-sera-la-industria-del-futuro/>
- Autómatas, B. (2014). *Infopl*. Obtenido de <http://www.infopl.net/blogs-automatizacion/item/102143-robots-colaborativos>
- Berger, R. (2016). *España 4.0: el reto de la transformación digital de la economía*. Roland Berger S.A. & SIEMENS.
- Dassault Systèmes. (2018). *DELMIA Robot Programming Essentials*. Obtenido de <https://peerlearningexperience.3ds.com/course/view.php?id=85>
- Dassault Systèmes. (2018). *DELMIA Robot Simulation*. Obtenido de <https://peerlearningexperience.3ds.com/course/view.php?id=126>
- Day, M. (2002). *What is PLM*. Cad Digest.
- FANUC. (2019). *Robots industriales FANUC*. Obtenido de <https://www.fanuc.eu/es/es/robots>
- Gabriel, C. M. (2017). *Digital Single Market*. Obtenido de https://ec.europa.eu/commission/priorities/digital-single-market_en
- Gonzalez, B. (2018). *Industria 4.0: una revolución para las personas*. Obtenido de <https://www.youtube.com/watch?v=a0Ycxn-bZak>
- H.Østergaard, E. (2018). *Welcome to industry 5.0*. Universal Robots. Obtenido de https://info.universal-robots.com/hubfs/Enablers/White%20papers/Welcome%20to%20Industry%205.0_Esben%20%C3%98stergaard.pdf?submissionGuid=00c4d11f-80f2-4683-a12a-e821221793e3

- Hill, S. (2006). *A winning strategy*. Manufacturing Business Technology.
- Kessler, J. (2006). CIVILIZING EVENTS AND CHRONOLOGY. Obtenido de <https://web.archive.org/web/20060209071757/http://www.revisedhistory.org/civilevents.htm>
- Laaper, S., Dollar, B., Mussomeli, A., Sniderman, B., & Mariani, J. (2017). *The digital supply network meets the future of work*. Deloitte Insights.
- Mark Cotteleer, B. S. (2017). *Forces of change: Industry 4.0*. Deloitte Insights.
- Marr, B. (2016). *What Everyone Must Know About Industry*. Forbes.
- Ministerio de Industria, C. y. (s.f.). *Industria Conectada 4.0*. Obtenido de <http://www.industriaconectada40.gob.es/Paginas/index.aspx#inicio>
- Selecciones del Reader's Digest. (1981). Gran Diccionario Enciclopédico Ilustrado (Doce Tomos). Selecciones del Reader's Digest.
- Winner, R. I., Pennell, J. P., Bertrand, H. E., & Slusarczuk, M. M. (1991). *The Role of Concurrent Engineering in Weapons System Acquisition*.
- World Economic Forum. (2018). *The Future of Jobs Report*.