

Gemelo digital de un centro de mecanizado asistido por brazo robótico

Autor: Jimeno Presas, Javier

Director: Javier Manini Gumz

Co-Director: José Antonio Rodríguez Mondejar

Entidad Colaboradora: Universidad Pontificia Comillas

ABSTRACT La emergencia de la Industria 4.0 ha expandido enormemente el horizonte de posibilidades de implementación de tecnologías punteras en los modelos de producción tradicionales. El presente proyecto trata de realizar una demostración de esto mediante el desarrollo de un gemelo digital de una célula flexible constituida por un centro de mecanizado asistido por un brazo robótico. Para ello, se ha hecho uso de una arquitectura de simulación en la que conectan varios programas mediante acoplamientos software con SIMIT. Por una parte, se utiliza RobotStudio con el propósito de obtener el comportamiento cinemático exacto del brazo robótico y, por otra, se utiliza el motor de física de NX-MCD para simular la interacción del brazo con el centro de mecanizado. Este, a su vez, está programado para realizar una serie de operaciones de fresado en NX-CAM. Ambos programas contienen réplicas de la célula flexible que se encuentran gobernadas por un único PLC simulado programado en TIA Portal, con el objetivo de obtener una simulación lo más cercana posible a la realidad. Durante la simulación, el brazo robótico se servirá de una pinza doble con la que realizará sustituciones rápidas de la herramienta utilizada por el centro de mecanizado.

PALABRAS CLAVE Centro de Mecanizado, Brazo Robótico, Gemelo Digital, RobotStudio, NX, MCD CAD, CAM, SIMIT, Mecatrónica, TIA Portal

I. INTRODUCCIÓN

Los avances que trajo consigo la tercera revolución industrial originaron la aparición de dos sectores clave, el de las Tecnologías de la Información y la Comunicación (TIC) y el de la automatización industrial. Hasta ahora, estos campos se encontraban aislados, ocupando cada uno un sector distinto del mercado. Por su parte, la ingeniería de automatización revolucionó los modelos de producción tradicionales. Las cadenas de ensamblaje se complementaron con brazos robóticos capaces de coordinarse mediante controladores o autómatas programables. En cuanto a los talleres artesanales, estos quedaron reconvertidos en células flexibles, equipadas con máquinas CNC, Vehículos Guiados Automáticos (AGVs) y cintas transportadoras.

Más tarde, con el advenimiento de la Industria 4.0, se introdujo la posibilidad de digitalizar estos modelos productivos con el propósito de desarrollar fábricas inteligentes, en las que se priorizaría la flexibilidad por encima de la cantidad producida. En este campo, destacan especialmente las células flexibles que, como se comprobará más tarde, sirven como una infraestructura excelente para la implantación de las nuevas tecnologías.



Figura 1. Diseño conceptual de una célula flexible [1]

II. ESTADO DE LA CUESTIÓN

El proceso de digitalización de los sistemas de producción tradicionales está desvelando su verdadero potencial.

a) CNNs

Un ejemplo de ello son los algoritmos de visión por ordenador o *Computer Vision* que, gracias a las capacidades

de interpretación de imágenes de las redes neuronales convolucionales (CNN), pueden dotar de una gran autonomía a los brazos robóticos, confiriéndoles la capacidad de identificar la posición de aquellos objetos que deban manipular sin necesidad de que estos se hallen siempre en un lugar fijo, para el que los robots estén rígidamente programados. Un ejemplo de CNN es la arquitectura VGG-16 que aparece en la Fig. 2.

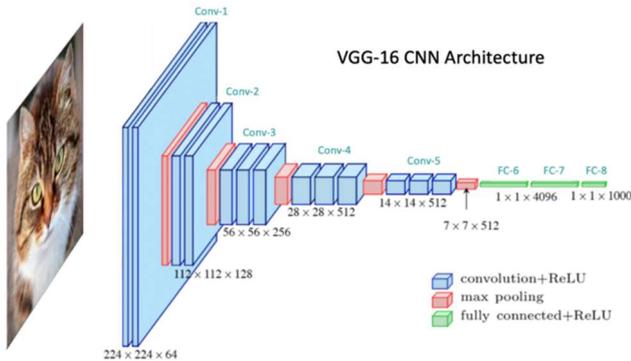


Figura 2. Arquitectura VGG-16 [2]

b) AMRs

Por otra parte, los AGVs, que suelen encontrarse restringidos a desplazarse sobre una trayectoria preestablecida, ahora pueden sustituirse por Robots Móviles Autónomos (AMR), como el ABB Flexly Mover que aparece en la Fig. 3. Estos vehículos pueden utilizar algoritmos de mapeo y de localización o “SLAM”, que les permiten desenvolverse con soltura en entornos industriales sin necesidad de haberles suministrado previamente un mapa de la instalación. De esta manera, los AMRs pueden transportar cargas por trayectorias nuevas y complejas sin requerir grandes esfuerzos de programación y desarrollo.



Figura 3. ABB Flexly Mover AMR P304 [3]

c) Gemelos Digitales

Una de las tecnologías que más impacto está teniendo en la industria son los gemelos digitales. Esto es debido a su capacidad para transformar todas las etapas del ciclo de vida de los equipos industriales modernos mediante una técnica denominada *Virtual Commissioning* o puesta en marcha virtual. Como se comprobará más tarde, implementar este procedimiento conlleva múltiples ventajas, como el ahorro en

costes de inversión, la reducción del tiempo de puesta en marcha o la optimización del rendimiento.

Un gemelo digital está compuesto por tres partes fundamentales: los sistemas de control, los modelos de comportamiento de los componentes electromecánicos y un motor de física con el que puedan simularse las interacciones que dichos componentes experimentan con el entorno que les circunda.

Existen diversas maneras de desarrollar un gemelo digital dependiendo del estado de avance del sistema o producto que se requiera virtualizar. En caso de que el proyecto se halla en una fase preliminar, resulta deseable realizar una puesta en marcha virtual. Como puede observarse en la Fig. 4, esta consta de varias fases de desarrollo mediante las que el sistema sufre una transición desde una configuración puramente digital a la puesta en marcha real del sistema.

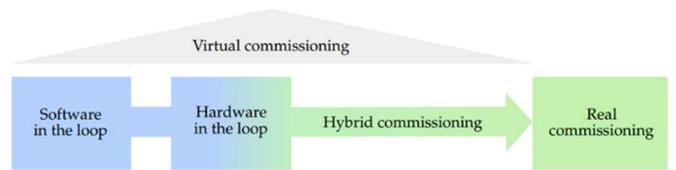


Figura 4. Etapas de una puesta en marcha virtual [4]

- i) **Software in the Loop.** La primera fase de una puesta en marcha virtual se denomina *Software in the Loop* (SiL). En este esquema, tanto los sistemas de control, como los modelos de comportamientos y el entorno de simulación son digitales y generalmente se encuentran implementados mediante varias soluciones *software*. Esto permite, de manera preliminar, planificar el sistema mediante iteraciones en su diseño, utilizando el entorno *software* como campo de pruebas. Así, aparece en la Fig. 5, donde se presenta un ejemplo de gemelo digital SiL implementado con productos de SIEMENS.



Figura 5. Software in the Loop [5]

- ii) **Hardware in the Loop.** En la segunda fase, denominada *Hardware in the Loop* (HiL), los sistemas de control digitales se sustituyen por PLCs y controladores reales, que se acoplan mediante módulos de entrada y salida a aquellos equipos que contengan los modelos de comportamiento y el entorno de simulación.

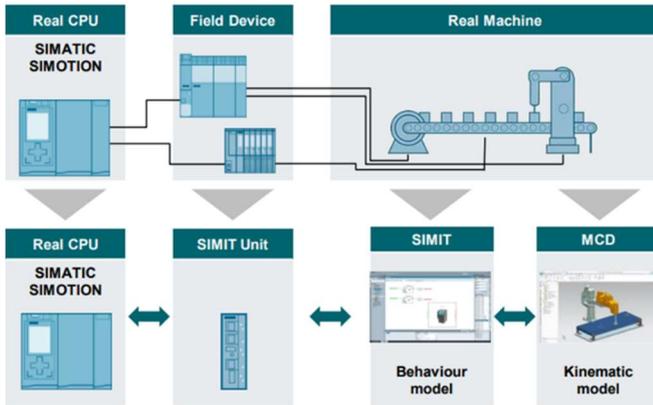


Figura 6. Hardware in the Loop [6]

III. MOTIVACIÓN DEL PROYECTO

Este proyecto tiene su enfoque en la digitalización de una máquina CNC procedente del fabricante Hartford, modelo HV-35 que se halla en el Laboratorio de Sistemas Integrados de Fabricación (LSITF).



Figura 7. Centro de mecanizado Hartford HV-35

Los trabajos de virtualización de la máquina en sí ya fueron realizados por un alumno anterior en su TFG, en el que se desarrolló un modelo CAD y CAM de la misma en el software NX, así como la programación de una serie de operaciones de fabricación.

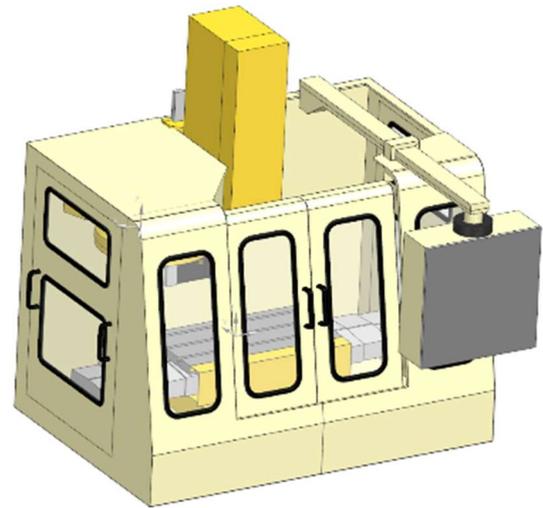


Figura 8. Modelo CAD del centro de mecanizado [7]

Aunque la máquina ya contaba con un sistema de cambio automático de herramientas, esta tenía ciertas limitaciones en cuanto a su capacidad y el tamaño de fresa que era capaz de albergar. Por estas razones, se propuso la utilización de un brazo robótico que pudiera suministrarla con un amplio abanico de herramientas de distintos tamaños con el objetivo de aumentar su versatilidad.

Dicho robot, para realizar el cambio de herramienta de forma ágil, debería estar equipado con una pinza doble. De esta manera, podría transportar la herramienta solicitada en una de sus pinzas al centro de mecanizado y, al llegar a él, desacoplar la herramienta a sustituir mediante la otra pinza.

A continuación, el robot acoplaría la herramienta solicitada a la máquina y se retiraría con la herramienta sustituida.

IV. PLANIFICACIÓN DEL SISTEMA

Para desarrollar un gemelo digital de la máquina CNC Hartford y del brazo robótico que la asiste, se decidió implementar la primera fase de una puesta en marcha virtual, es decir, un gemelo digital de tipo *Software in the Loop*.

Para los sistemas de control, se decidió utilizar un PLC simulado que se programaría en TIA Portal. Este quedaría encargado de gobernar todos los procesos que se llevarían a cabo en NX, que es el software donde se encontraba definida la máquina CNC. A continuación, para conectar ambos softwares se haría uso del *software* SIMIT, que manejaría los intercambios de señales entre ellos.

Esta arquitectura aparece representada en detalle en la Fig. 9, donde se detallan los roles que adopta cada uno de los *softwares* involucrados.

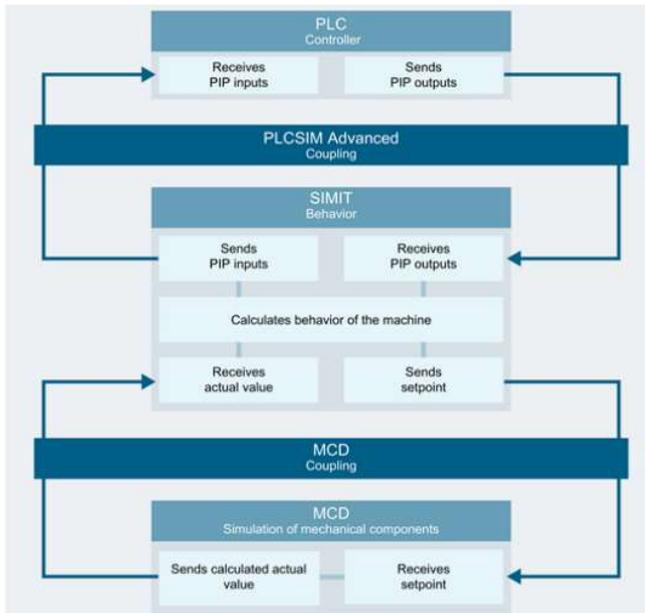


Figura 9. Arquitectura PLC-SIMIT-MCD [8]

En cuanto al brazo robótico, si bien es cierto que su modelo de comportamiento podría programarse en SIMIT o en NX, esto conllevaría varios problemas en el momento de arrancar la simulación, ya que su comportamiento cinemático no se correspondería con el establecido por el fabricante. Por tanto, cabría la posibilidad de que las trayectorias programadas en el robot no pudieran llevarse a cabo en la realidad debido a que el robot podría estar adoptando configuraciones imposibles o incluso experimentar impactos entre sus eslabones.

Para evitar esto, es necesario utilizar una arquitectura de simulación robótica de alta fidelidad que admita la utilización de los *softwares* de simulación que corresponden al fabricante del robot en cuestión.

De esta manera, en caso de tratarse de un robot de ABB, su comportamiento cinemático del robot se transferiría a RobotStudio.

Por último, para el motor de física se utilizó la extensión de Diseño de Conceptos de Mecatrónica (MCD) en NX que, además de contener la máquina CNC, también incluiría una réplica del brazo robótico que se halla en RobotStudio.

Para gestionar los intercambios de señales entre los *softwares*, se hizo uso de la organización que aparece en la Fig. 10.

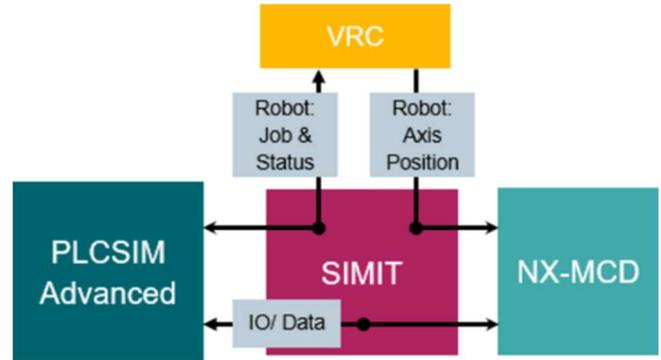


Figura 10. Arquitectura de simulación robótica de alta fidelidad [9]

Uno de los procesos más importantes consiste en la sincronización de las dos réplicas del brazo robótico. Según este esquema, la réplica que se halla en NX-MCD queda gobernada de manera externa a partir de una transferencia de valores de las posiciones angulares de las articulaciones del robot (*Robot: Axis Position*) desde el Controlador Virtual del Robot (VRC) ubicado en RobotStudio.

Por otra parte, la instancia de PLC simulado (PLCSIM Advanced) se dedica a adjudicar tareas al VRC y a monitorizar su estado de cumplimiento (*Robot: Job & Status*).

Finalmente, existe un intercambio de señales de entrada y salida (*IO/Data*) entre PLCSIM Advanced y NX-MCD mediante los que se gestiona la interacción del brazo robótico con su entorno y con la máquina herramienta. No obstante, como se comprobará más tarde, también se intercambiarán señales de este tipo entre el VRC y NX-MCD ya que, en ocasiones, no existe la necesidad de que estas sean gestionadas también por el PLC.

V. PINZA DE TRANSPORTE

Para que el brazo robótico sea capaz de manipular herramientas de diversos tamaños mediante una pinza doble, es necesario utilizar un diseño que permita realizar correctamente los agarres. Para ello, se recurrió a técnicas de *Digital Manufacturing* de modelado dinámico en conjunción con archivos CAD procedentes de sitios web de fabricantes de accesorios para robots.

En concreto, la base de la pinza se obtuvo del fabricante alemán SCHUNK, al que se solicitó el modelo de pinza angular: PWG-plus 64 [10]. Por su parte, los dedos se diseñaron mediante operaciones de modelado en NX-CAD, los cuales constan de 4 superficies oblicuas con las que pueden agarrarse fresas de diversos diámetros.

Para ensamblar la pinza a los dedos, se hizo uso de restricciones relacionales en la pestaña de Ensamblaje de NX. Finalmente, se obtuvo la pinza que puede apreciarse en la Fig. 11, que después se duplicaría para obtener la pinza doble.

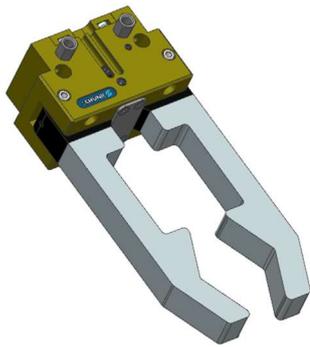


Figura 11. Ensamble de pinza

Una vez finalizada la pinza, se procedió a la importación desde SCHUNK de una placa adaptadora: A-AGE-F-XY-040 [11] que permitiera acoplar la pinza doble a la brida de la muñeca del brazo robótico. Dado que se trata de un robot ABB CRB1300-7/1.4 [12], dicha placa debe seguir un patrón de colocación de agujeros para uniones atornilladas según la norma: ISO 9409-1-40-4-M6.

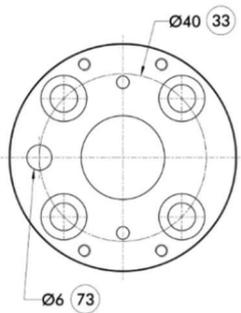


Figura 12. Norma ISO 9409-1-40-4-M6 [13]

A continuación, para acoplar ambas pinzas a la placa adaptadora y esta, a su vez, al robot, se hizo uso del marco que aparece en la Fig. 13, que también fue modelado en NX mediante la creación de planos de datum, bocetos, extrusiones, simetrías y redondeos. Como podrá comprobarse, la geometría que presenta permite insertar tornillos para unir el marco a la placa adaptadora y, a su vez, para unir la placa a la brida del robot.

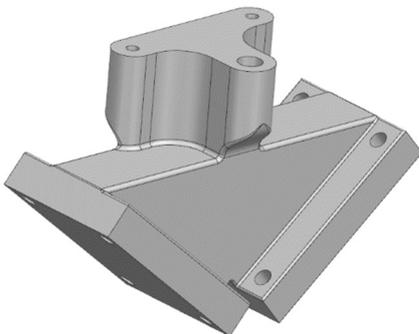


Figura 13. Marco de la pinza doble

Finalmente, se procedió a ensamblar todos estos componentes mediante restricciones relacionales de coincidencia planar y axial, resultando en el modelo CAD de pinza doble que puede apreciarse en la Fig. 14.

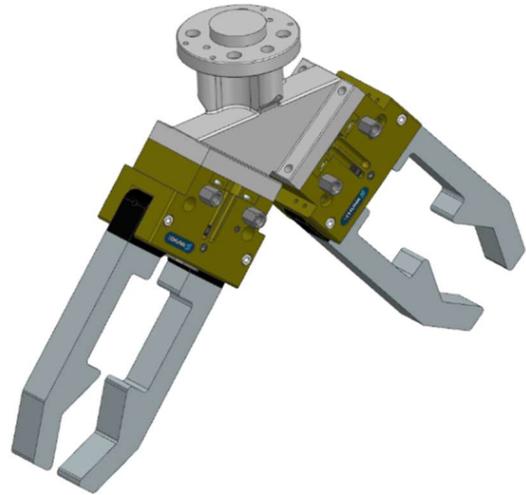


Figura 14. Ensamble de pinza doble

Acto seguido, este modelo se exportó a RobotStudio, donde cada pinza se configuró como un mecanismo independiente con dos articulaciones angulares, una por cada dedo. Estas articulaciones se programaron con diversas poses que correspondían al grado de cierre adecuado para el agarre de cada herramienta medido en grados (°).

Poses	
Nombre de pose	Valores de pose
Pose de sincronización	[0.00: 0.00]
Pose inicial	[0.00: 0.00]
Left_UGT0202_031	[13.50: 13.50]
Opened_Left_Gripper	[18.00: 18.00]
Left_UGT0202_002	[9.00: 9.00]
Left_UGT0201_002	[1.00: 1.00]
Left_NXT0301_031	[-2.70: -2.70]
Left_WKZ_101004	[-2.70: -2.70]

Figura 15. Poses de la pinza izquierda

Finalizada la configuración de la pinza doble, esta se acopló a la brida del brazo robótico que se halla en Robotstudio.

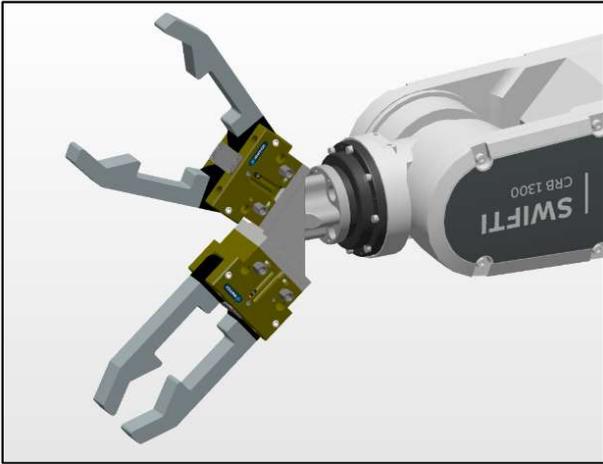


Figura 16. Pinza doble acoplada a la brida del robot

VI. CÉLULA FLEXIBLE EN ROBOTSTUDIO

El diseño en planta de la célula flexible se realizó en el *software* RobotStudio debido a la agilidad que este ofrecía para realizar pruebas con diferentes robots. De esta manera, se pudo determinar que el robot idóneo para esta aplicación era el robot colaborativo SWIFTI CRB1300-7/1.4 que, como su nombre indica, tiene un alcance de 1,4 m y es capaz de cargar con 7 kg, gozando además de una velocidad de TCP que ronda los 6 m/s [12]. Estas especificaciones le permiten cubrir la distancia entre la máquina CNC y la estantería de herramientas mientras carga conjuntamente con la pinza doble y la fresa más pesada.

En la Fig. 17 puede observarse el diseño final de la célula flexible, que incluye el centro de mecanizado, el brazo robótico escogido en posición invertida y la estantería de herramientas, que alberga todas las fresas que utilizará la máquina herramienta durante la ejecución de sus operaciones de mecanizado.

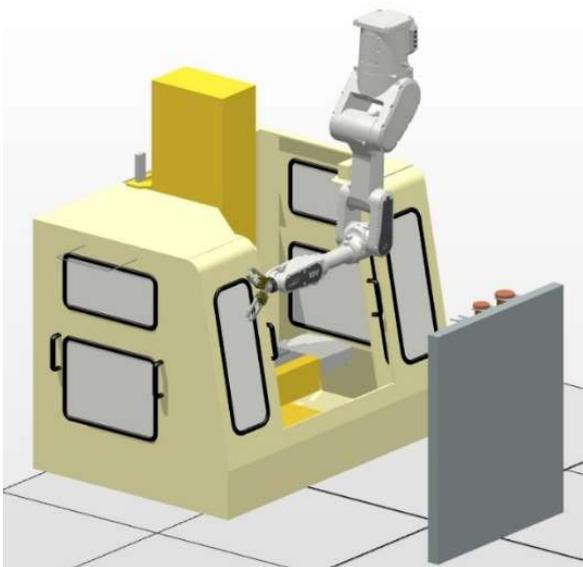


Figura 17. Célula flexible en RobotStudio

Habiendo finalizado la configuración en planta de la estación, se procedió a desarrollar los puntos-objetivo y las trayectorias correspondientes que el robot debería seguir para realizar las tareas de cambio de herramienta. Para ello, se hizo uso del lenguaje de programación RAPID, mediante el que se implementaron procedimientos y subrutinas que permitieron adaptar los desplazamientos del robot en función de la fresa solicitada.

VII. CÉLULA FLEXIBLE EN NX-MCD

a) Brazo Robótico

Según la arquitectura *software* que se comentó en la Fig. 10, también debe existir una réplica de la célula flexible en el entorno de NX-MCD. Para ello, primero se procedió a la importación del modelo CAD del robot en cuestión desde la página web de ABB. Después, mediante restricciones de coincidencia planar y axial, se ensamblaron todos sus eslabones entre sí y, finalmente, a la pinza doble, cuyo modelo CAD ya se encontraba en NX.

Una vez se obtuvo el ensamble completo del brazo robótico, este se colocó en la posición y orientación exactas que se habían adoptado en la réplica de la estación en RobotStudio, como puede observarse en la Fig. 18.

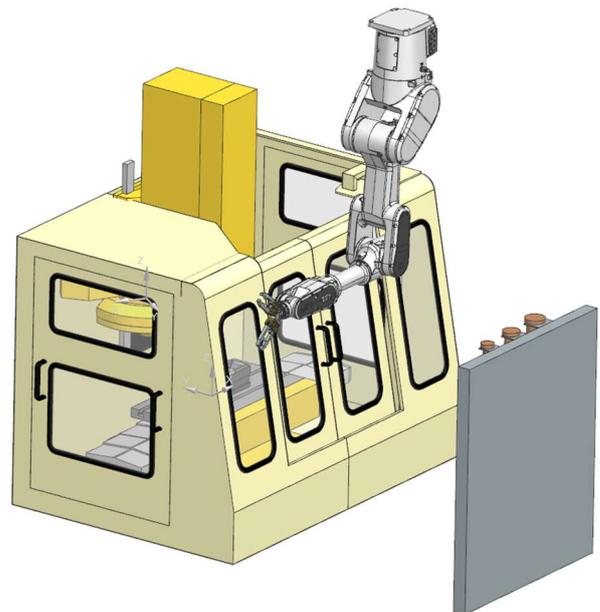


Figura 18. Célula flexible en NX-MCD

b) Preferencias de MCD

Como se ha comentado antes, la extensión MCD de NX consta de un motor de física que permite simular las interacciones entre los componentes del sistema. Para que la simulación transcurra de la manera deseada, existen varios parámetros que deben configurarse en el apartado de preferencias de MCD:

- i) **Algoritmo de control de velocidades.** Para que los brazos robóticos en RobotStudio y NX estén correctamente sincronizados, las articulaciones de la réplica que existe en NX son gobernadas de manera externa, recibiendo consignas de posición angular desde RobotStudio. Por otra parte, las velocidades con las que las juntas se desplazarán hasta alcanzar dichas posiciones se calcularán mediante el siguiente algoritmo, que se sirve de dos posiciones externas:

$$\text{Velocidad} = \frac{\text{Posición actual externa} - \text{Posición anterior externa}}{\text{tiempo en un paso hacia adelante}}$$

Resulta imprescindible escoger este algoritmo. En caso de seleccionar la opción predeterminada, donde las velocidades se calculan mediante una posición externa y una actual, el robot experimentará una gran inestabilidad en sus movimientos y no será capaz de realizar sus tareas.

- ii) **Motor de Física.** En el apartado de motor de física, debe escogerse “Modo fino” en las opciones de exactitud de la simulación. Esto le conferirá al robot la capacidad de realizar tareas de manipulación con sus pinzas de manera precisa debido a la disminución de la latencia en la entrada de señales externas. Por otra parte, a la fuerza de adherencia que se aplica a los cuerpos de colisión con capacidades adherentes se le ha adjudicado el valor de 1 N, siendo este el valor que más estabilidad producía durante el transcurso de la simulación.

En definitiva, se trata de configuraciones que aumentan el realismo y precisión del gemelo digital.

c) Propiedades físicas

A continuación, se procederá a describir las operaciones mediante las que se adjudicaron propiedades físicas a los componentes de la estación que se desea que participen en la simulación de MCD.

- i) **Sólidos Rígidos.** Las, fresas, eslabones del robot, componentes de la pinza, husillo y puertas de la Hartford se convirtieron en sólidos rígidos, lo cual permitió asignarles propiedades físicas y ubicar su centro de gravedad donde correspondiera.
- ii) **Cuerpos de Colisión.** Durante, la simulación, existen varias interacciones entre sólidos rígidos que deben gestionarse mediante la asignación de cuerpos de colisión en las superficies de contacto. Esto incluye a las superficies interiores de las pinzas y a la zona de agarre de las herramientas que, al ser configuradas con propiedades adherentes, podrían ser transportadas por el robot. También se asignaron cuerpos de colisión a la superficie inferior del extremo superior de las fresas y a

las superficies de sujeción correspondientes en la estantería de herramientas, lo que permitía que estas se mantuvieran sujetas sin utilizar restricciones relacionales.

Por último, se asignaron cuerpos de colisión adherentes a la zona inferior del husillo de la Hartford y a la superficie superior del extremo superior de las fresas. De esta manera, se simularía el acoplamiento con la máquina herramienta.

- iii) **Juntas.** Las uniones articuladas entre los eslabones del robot deben configurarse como juntas acharneladas, estableciendo así su comportamiento cinemático.

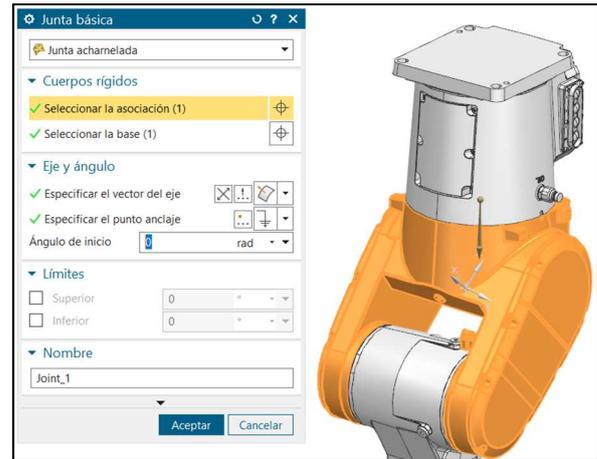


Figura 19. Ejemplo de definición de junta acharnelada

Por otra parte, los rieles sobre los que se desplazan las puertas de la Hartford se definirán como juntas deslizantes, lo que permitirá su apertura y cierre durante la simulación.

- iv) **Sensores y actuadores.** Para definir los accionamientos de las articulaciones del brazo robótico y de las puertas de la Hartford, es necesario asignarles controladores de posición. En caso del robot, estos deben ser controladores angulares que deben ser configurados con “Datos desde Externo”, cuyas consignas serán importadas desde RobotStudio.

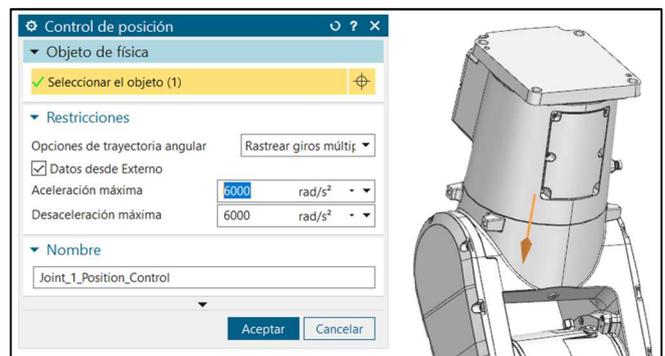


Figura 20. Ejemplo de controlador de posición angular

Finalmente, se asignaron sensores a cuerpos rígidos que se activarían en el momento de detectar interferencias con otros cuerpos en la simulación. Esto permitiría incrementar la robustez del acoplamiento entre las réplicas de la estación en RobotStudio y NX. En concreto, se colocaron sensores en el centro de la zona interior de los dedos de las pinzas, que indicarían de la presencia de una fresa; y otro en el husillo de la Hartford, que se activaría en caso de haber una fresa acoplada.

- v) **Cuerpos de Alineación.** Si dos cuerpos rígidos se definen como cuerpos de alineación, al encontrarse ambos a cierta distancia, uno de ellos se desplazará hasta adoptar la posición y orientación del otro. En la simulación, esto resulta de especial interés para asegurar que las fresas se posicionen correctamente con respecto a la pinza que las desplazará.
- vi) **Señales.** NX permite definir señales que pueden utilizarse para controlar procesos durante la simulación o para intercambiar información con *softwares* externos. Dichas señales pueden asignarse directamente a un parámetro de tiempo de ejecución o utilizarse en objetos de programación más complejos, como expresiones de tiempo de ejecución o bloques de expresiones. La Tabla 1 recoge las señales utilizadas, su tipo y su utilización en objetos de programación

Señal	Tipo	Objeto de programación
Right_Gripper_Sensor	Salida	Parámetro de Tiempo de Ejecución
Left_Gripper_Sensor	Salida	Parámetro de Tiempo de Ejecución
Spindle_Sensor	Salida	Parámetro de Tiempo de Ejecución
Right_Gripper_Control	Entrada	Bloque de expresiones
Left_Gripper_Control	Entrada	Bloque de Expresiones
Spindle_Connection	Entrada	Parámetro de Tiempo de Ejecución
Right_Gripper_Alignment	Entrada	Bloque de Expresiones
Left_Gripper_Alignment	Entrada	Bloque de Expresiones
Right_Gripper_Connection	Entrada	Bloque de Expresiones
Left_Gripper_Connection	Entrada	Bloque de Expresiones

Tabla 1. Señales en NX-MCD

VIII. ACOPLAMIENTOS SOFTWARE

Para interconectar los 3 *softwares*, se hizo uso de SIMIT, donde se crearon una serie de acoplamientos *softwares* que permitían el flujo de información entre cada programa y SIMIT.

a) Acoplamiento SHM

El acoplamiento por memoria compartida o *Shared Memory (SHM)* permite asignar una misma dirección de memoria a las señales que se requiere interconectar entre RobotStudio y SIMIT.

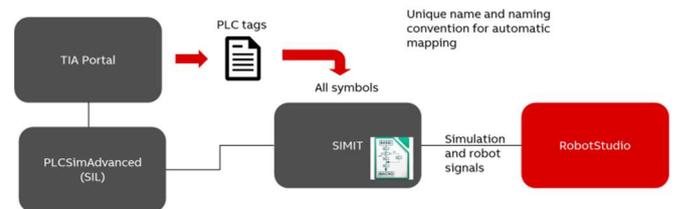


Figura 21. Acoplamiento SHM [14]

Para llevar esto a cabo, es necesario utilizar el componente inteligente: “SIMITConnection”, mediante el que se gestionarán las interacciones entre las señales externas y las señales de estación. Este componente debe incluir el nombre que se le ha asignado al acoplamiento SHM en su configuración en SIMIT, además de haber seleccionado la opción “Signal description in header”. De lo contrario, no podrá establecerse la conexión.

A continuación, se definieron las señales que se intercambiarían entre los 2 *softwares*, cuyo método de conexión dependerá del tipo de símbolo utilizado:

- Señales de Estación.** Este tipo de símbolo se gestiona mediante el apartado de lógica de estación en RobotStudio. Las señales que han sido previamente definidas en SIMIT, después de establecer la conexión con RobotStudio, pueden asignarse a señales de estación o a señales del controlador, según se necesite.
- Símbolos de Robot.** Los símbolos de robot son especialmente útiles para aquellas señales que no se requiere que interactúen con la estación en RobotStudio, ya que permiten realizar una conexión directa entre el controlador del robot y SIMIT. No obstante, para emplear este símbolo, es necesario crear un dispositivo PROFINET interno en el controlador del robot, el cual debe estar configurado con el nombre del acoplamiento en cuestión. Por otra parte, las señales en SIMIT que corresponden a las que se hayan definido en dicho dispositivo deben seguir una sintaxis determinada:

**Robot[Nombre del controlador]
[Nombre del dispositivo PROFINET]_Identificador [14]**

Robot[CRB1300_7_140__01][PN_Internal_Device]RS_AI_Tool_Request	MB1	BYTE
Robot[CRB1300_7_140__01][PN_Internal_Device]RS_DI_Tool_Status	M2.0	BOOL

Figura 22. Ejemplo de definición de símbolos de robot

Finalmente, es necesario asegurarse de que las señales que se hallen definidas en SIMIT se encuentren asignadas a direcciones de memoria contiguas, además de que la cantidad de memoria que ocupen debe ser idéntica a la que ocupen sus señales correspondientes en RobotStudio.

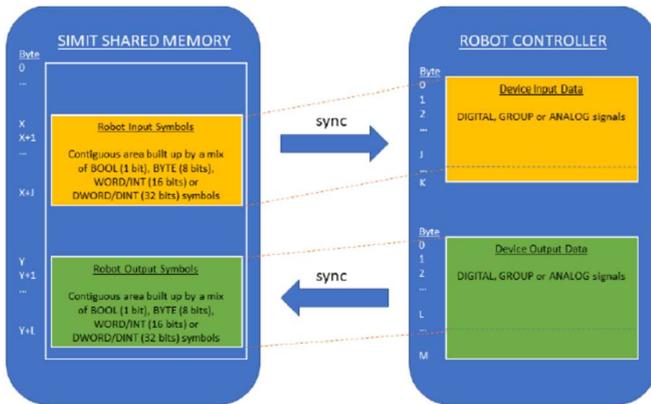


Figura 23. Conexión de señales mediante símbolo de robot [14]

iii) **Símbolos de Articulación.** Estos símbolos son de gran utilidad para extraer los valores de las posiciones angulares de las articulaciones del brazo robótico en RobotStudio. No obstante, debe utilizarse también una sintaxis determinada:

**Joint[Nombre del controlador]
[Nombre de la unidad mecánica][Número de eje] [14]**

Por otra parte, debe asegurarse que estas señales estén asignadas a direcciones de memoria de tipo: "REAL".

Joint[CRB1300_7_140__01][ROB_1][1]	MD16	REAL
Joint[CRB1300_7_140__01][ROB_1][2]	MD20	REAL
Joint[CRB1300_7_140__01][ROB_1][3]	MD24	REAL
Joint[CRB1300_7_140__01][ROB_1][4]	MD28	REAL
Joint[CRB1300_7_140__01][ROB_1][5]	MD32	REAL
Joint[CRB1300_7_140__01][ROB_1][6]	MD36	REAL

Figura 24. Definición de símbolos de articulación

b) Acoplamiento MCD

Para realizar el acoplamiento MCD, es necesario utilizar los comandos de la pestaña "Simit" mientras NX se halla funcionando en su extensión de MCD.

En concreto, debe seleccionarse el commando: "Allow connection" y, acto seguido, especificar el nombre del acoplamiento MCD que previamente se definió en SIMIT.

Después, en el entorno de SIMIT, seleccionar el comando: "Receive signals from MCD", completando el procedimiento mediante la selección del comando: "Send Signals to Simit" en el entorno de NX. De esta manera, se obtendrá en el entorno de SIMIT todas aquellas señales previamente definidas en NX.

c) Acoplamiento PLCSIM Advanced

Para crear un acoplamiento PLCSIM Advanced que interconecte el PLC simulado con SIMIT, es necesario importar la estación previamente creada en TIA Portal. Esto permitirá acceder a las variables de entrada y salida del PLC simulado.

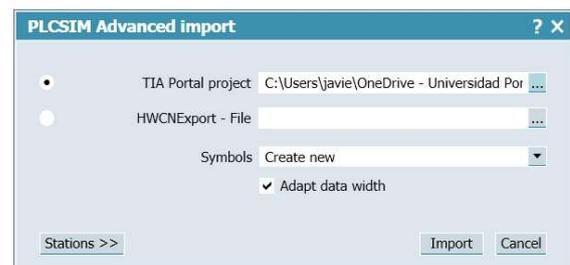


Figura 25. Importar proyecto desde TIA Portal

d) Gráficos de Conexiones

El entorno de SIMIT permite gestionar las señales transferidas entre los softwares mediante una interfaz gráfica.

i) **Gráfico de transferencia de posiciones de articulaciones del robot.** El gráfico que aparece en la Fig. 26 representa la conexión entre los símbolos de articulación provenientes de RobotStudio y los controladores de posición destinados a NX-MCD mediante conectores analógicos: "A".



Figura 26. Gráfico de conexiones de posiciones de articulaciones

ii) **Gráfico de señales I/O entre RobotStudio y NX.** En este gráfico se concentran todas las señales intercambiadas entre los dos softwares. Por un lado, las

señales de los sensores de las pinzas y del husillo de la Hartford provenientes del acoplamiento MCD se transfieren a las señales correspondientes en RobotStudio mediante conectores booleanos: “B”. Por otra parte, las señales de control de la apertura y cierre de las pinzas, la adherencia de sus dedos y el alineamiento de las fresas se destinan a las señales correspondientes en NX mediante los mismos conectores.

Además, el valor correspondiente a la fresa actualmente acoplada en el husillo de la Hartford se transfiere desde RobotStudio a NX mediante un símbolo de robot.



Figura 27. Gráfico de señales I/O entre RobotStudio y NX

iii) **Gráfico de señales I/O con PLCSIM Advanced.**

Finalmente, el gráfico que aparece en la Fig. representa todas las conexiones realizadas con el PLC simulado. Primero, el valor correspondiente a la herramienta que se solicite al brazo robótico es transferido tanto a RobotStudio como a NX desde el PLC. Adicionalmente, RobotStudio comunica el estado de disponibilidad del brazo robótico al PLC mediante un símbolo de robot.

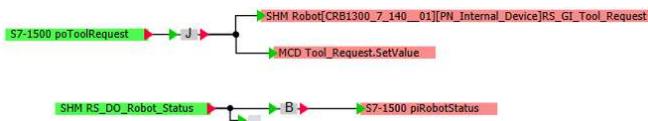


Figura 28. Gráfico de señales I/O con PLCSIM Advanced

IX. SIMULACIÓN

a) Procedimientos preliminares

El proceso de arranque de la simulación consta de varios pasos. Primero, en NX se permite la conexión con SIMIT mediante el comando: “Allow Connection” en la pestaña de “Simit”. Acto seguido, se procede a iniciar la simulación en SIMIT, lo que desencadena la creación de una instancia de

PLC simulado donde se aloja el acoplamiento PLCSIM Advanced entre SIMIT y el PLC [15]. Después de instalar el programa previamente realizado en TIA Portal y de arrancar el módulo, debe comprobarse que la instancia está activa, tal y como aparece en la Fig 29.



Figura 29. Instancia de PLC simulado: SIMIT_S7-1500

Por último, resta dirigirse a RobotStudio para seleccionar la opción: “Conectar” en el componente inteligente “SIMITConnection”. Esto completará la conexión entre los softwares.

b) Arranque de la simulación

Antes de ejecutar el programa realizado en el lenguaje RAPID en RobotStudio, es necesario acceder al apartado de “Configuración de la Simulación” e indicar que el componente inteligente: “SIMITConnection” participará en la simulación. De lo contrario, los intercambios de señales sólo se producirán en los momentos en que no se esté ejecutando el programa de RobotStudio.

A continuación, debe iniciarse la simulación del panel HMI desde TIA Portal (Véase Fig.), el cual contiene botones para solicitar las herramientas que se requiera acoplar en el husillo de la Hartford mediante el brazo robótico. En esta interfaz, también podrá comprobarse, mediante pilotos luminosos, el estado de disponibilidad del robot: “Robot Status” y la presencia o no de una herramienta acoplada en el husillo de la Hartford: “Tool Status”.

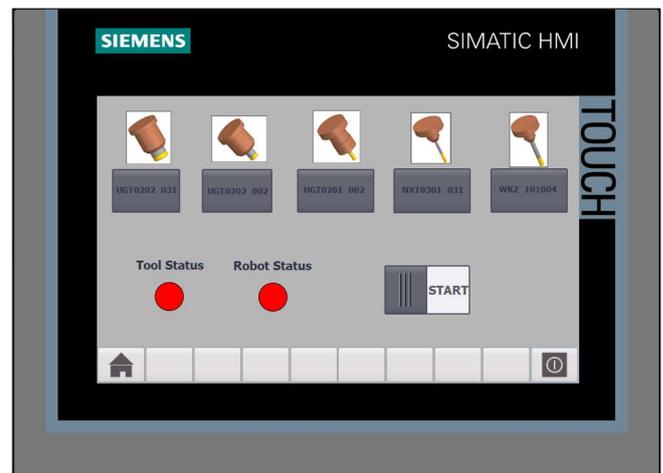


Figura 30. Panel HMI

Finalmente, resta ejecutar el programa desde RobotStudio mediante el botón de “Reproducir”, lo que dará comienzo a la simulación.

c) Transcurso de la simulación

i) **Solicitud de herramientas.** El desplazamiento del interruptor en el panel HMI hacia su posición “START” permitirá solicitar una de las 5 herramientas disponibles en la región superior del panel, según la operación de mecanizado que se desee llevar a cabo a continuación en la Hartford. El valor correspondiente a dicha herramienta se transmitirá a RobotStudio mediante un símbolo de robot, lo que iniciará las maniobras de manipulación del brazo robótico. El estado del robot se actualizará a en el panel HMI mediante una señal proveniente de RobotStudio que, además imposibilitará la solicitud de otra herramienta mientras el robot se encuentre llevando a cabo sus tareas.

ii) **Manipulación de herramientas.** Para realizar el agarre de las herramientas de forma correcta en NX-MCD, el programa ejecutado en RobotStudio se sirve de una serie de señales que se transfieren a NX para controlar ciertos fenómenos. En concreto:

- La adherencia de las superficies interiores de los dedos de las pinzas.
- El alineamiento de las herramientas con las pinzas.
- El ángulo de apertura y cierre de los dedos de las pinzas.
- La adherencia del husillo de la máquina herramienta.

Estas señales se coordinan con los sensores descritos anteriormente para maximizar la fluidez y sincronización entre las dos réplicas de la célula flexible

X. CONCLUSIÓN Y TRABAJOS FUTUROS

Como se ha comprobado antes en la Fig. 4, este proyecto se enfoca únicamente en la primera fase de un proceso de *Virtual Commissioning*. En efecto, se ha logrado desarrollar un gemelo digital de tipo *Software in the Loop* mediante la implementación de una simulación robótica de alta fidelidad. De esta manera, ya se han finalizado la mayoría de las tareas de investigación, planificación y diseño de la célula flexible, además de gran parte de las fases de pruebas, gracias a la alta fidelidad de la simulación.

Posibles futuros trabajos deberían estar dirigidos a desarrollar un gemelo digital de tipo *Hardware in the Loop* con el objetivo de dar un paso más hacia una puesta en marcha real. Como se ha comentado antes, se trata de la siguiente fase en un proceso de puesta en marcha virtual, en la que se sustituyen los sistemas de control por *hardware* real:

a) PLC real

El primer paso consistiría en la utilización de un PLC real, que se conectaría al dispositivo que contiene el gemelo digital SiL, así como aparece en la Fig. 31. Para esta aplicación, lo más lógico sería utilizar un PLC SIMATIC S7-1500, siendo este el mismo modelo que el PLC que se ha simulado en TIA Portal.



Figura 31. PLC SIEMENS SIMATIC S7-1500

Para llevar esto a cabo, se utilizaría el diagrama que aparece representado en la Fig. 32, donde se utiliza un módulo SIMIT Unit en una red industrial PROFINET para conectar el PLC real con el dispositivo donde se encuentra el *software* SIMIT.

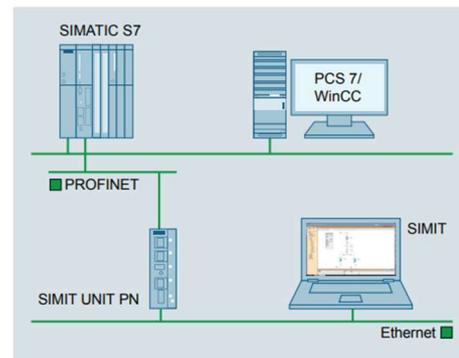


Figura 32. Conexión con PLC real mediante red PROFINET [16]

b) Controlador del robot real

Finalmente, se sustituiría el controlador virtual del robot ubicado en RobotStudio por un controlador real. Según la hoja de características del robot ABB CRB1300-1.4/7, podría utilizarse el modelo: “Omnicores C30” o el “Omnicores C90XT”.



Figura 33. Controladores ABB Omnicore C30 y C90XT [17]

BIBLIOGRAFÍA

- [1] *Células flexibles | Automatización | Liebherr.* (s. f.). Liebherr. <https://www.liebherr.com/es/col/productos/t%C3%A9cnica-de-engranajes-y-automatizaci%C3%B3n/sistemas-de-automatizaci%C3%B3n/maquinas-herramienta/celulas-de-robot-flexibles/celulas-de-robot-flexibles.html>
- [2] Kromydas, B., & Kromydas, B. (2024, 24 junio). Convolutional Neural Network: A complete guide. LearnOpenCV – Learn OpenCV, PyTorch, Keras, Tensorflow With Code, & Tutorials. <https://learnopencv.com/understanding-convolutional-neural-networks-cnn/>
- [3] ABB AMR Flexley Mover. (2023). ABB. <https://new.abb.com/products/robotics/robots/autonomous-mobile-robots/products/flexley-mover>
- [4] Noga, M., Juhás, M., & Gulán, M. (2022). Hybrid Virtual Commissioning of a Robotic Manipulator with Machine Vision Using a Single Controller. *Sensors*, 22(4), 1621. <https://doi.org/10.3390/s22041621>
- [5] Simulation for Automation. (s. f.). siemens.com Global Website. <https://www.siemens.com/global/en/products/automation/topic-areas/use-cases/simulation.html>
- [6] SIMATIC/SIMOTION Virtual Commissioning with Hardware in the Loop. (2018). Siemens Industry Online Support. <https://support.industry.siemens.com/cs/document/109758739/simatic-simotion-virtual-commissioning-with-hardware-in-the-loop?dti=0&lc=en-ES>
- [7] Javier, M. G., Mariano, J. C., & De Ingeniería, U. P. C. E. T. S. (2023). Simulación virtual de una MHCN (CMV-Centro de Mecanizado Vertical) y su integración en la aplicación Siemens NX. <http://hdl.handle.net/11531/70608>
- [8] Virtual Commissioning of machines with S7-PLCSIM Advanced, SIMIT & NX MCD – Getting Started. (2024). Siemens Industry Online Support. <https://support.industry.siemens.com/cs/document/109758943/virtual-commissioning-of-machines-with-s7-plcsim-advanced-simit-and-nx-mcd?dti=0&lc=en-ES>
- [9] Robotic Simulation with PLCSIM Advanced – SIMIT – NX-MCD. (2023). Siemens Industry Online Support. <https://support.industry.siemens.com/cs/document/109795246/simit-couplings-applicative-expansions-of-the-interface?dti=0&lc=en-ES>
- [10] SCHUNK Hand in hand for tomorrow. (s. f.). <https://schunk.com/br/en/gripping-systems/angular/radial-gripper/pwg-plus/pwg-plus-64/p/000000000000311620>
- [11] SCHUNK Hand in hand for tomorrow. (s. f.-b). <https://schunk.com/br/en/gripping-systems/accessories/adapter-plate/a-age-f-xy-040/p/000000000000324923>
- [12] SWIFTI CRB 1300. (2024). ABB. <https://new.abb.com/products/robotics/robots/collaborative-robots/swifti/crb-1300-swifti>
- [13] ISO 9409-1:2004. (s. f.). ISO. <https://www.iso.org/standard/36578.html>
- [14] RobotStudio. Manual del operador. (2023). ABB. <https://library.e.abb.com/public/f753d51b3abc4a26b633292af34ac734/3HAC032104%20OM%20RobotStudio-es.pdf?x-sign=+ML+uUvpsbe0ni1sykMztqvAhBeqr3WGIUQ4hY5yCe/+v4jW7cUSHtLJ8R226NBj>
- [15] SIMIT Simulation Platform (V11). (2022). Siemens Industry Online Support. https://cache.industry.siemens.com/dl/files/085/109812085/att_1109061/v1/SIMIT_enUS_en-US.pdf
- [16] SIMIT UNIT. Hardware platform for real time simulation via PROFINET and PROFIBUS. (2018). SIEMENS. <https://assets.new.siemens.com/siemens/assets/api/uuid:2ecd8e07-5017-4941-954a-c616ceff11ffe/pafl-00162-1217-simit-unit-usa.pdf>
- [17] Omnicore C30 and C90XT controllers. (2024). ABB. <https://new.abb.com/products/robotics/controllers/omnicore/omnicore-c30-and-c90xt>