

Interconexión de gemelo digital de célula de fabricación flexible con PLC, para el aprendizaje automático de parámetros.

Alejandro Torres, Javier Gómez, Juan M. Escaño
Departamento de Ingeniería de Sistemas y Automática. Universidad de Sevilla
aletormun@alum.us.es, jgomezj@us.es, jescano@us.es

Resumen

Se presenta la implementación de un sistema de comunicaciones que permita el intercambio de datos en tiempo real entre el sistema físico y su Gemelo Digital. El objetivo de este trabajo ha sido proponer una solución capaz de garantizar no sólo el intercambio de datos entre ambos sistemas; sino también la adaptación del gemelo digital a la realidad física, mediante rutinas de aprendizaje automático.

Palabras clave: Gemelo Digital, Comunicación entre planta y su gemelo, Adaptación de parámetros.

1. Introducción

El término "Gemelo Digital", o también conocido, en inglés, como *Digital Twin* (DT), se refiere a la representación virtual de cualquier sistema físico, incorporando, en tiempo real, datos sobre los parámetros de dicho sistema [1].

Dicho término fue expuesto por primera vez por el Dr. Michael Grieves en 2003 en la Universidad de Florida, siendo un concepto demasiado revolucionario para la tecnología disponible en aquellos años, ya que en el mismo se establecía que un producto virtual debía ser virtualmente indistinguible respecto al producto físico, y las tecnologías presentes en aquella época no permitían desarrollar en su plenitud este concepto. [2].

Sin embargo, la estructura de dicho concepto se ha mantenido constante a lo largo de este tiempo, siendo la siguiente:

1. Un sistema físico en el espacio real.
2. Un sistema virtual en un entorno digital.
3. Una conexión que permita la comunicación para el intercambio de datos entre los dos componentes anteriores.

Dicha terminología fue puesta en práctica por primera vez por la NASA, con el objetivo de mejorar las simulaciones de los modelos físicos de sus

naves espaciales. Esta no fue realmente acuñada hasta 2012, cuando el concepto de DT fue redefinido como: *Una simulación multifísica, multiescala, probabilística y de extrema fidelidad ejecutada en tiempo real; basándose en un histórico de datos, así como información recogida por sensores y el modelo físico del sistema* [3].

Además, dicha definición no debe confundirse con una simple simulación de un proceso. Lo que diferencia a un DT son las múltiples ventajas que posee gracias a sus múltiples aplicaciones. Entre ellas, se deben destacar las siguientes:

1. Monitorización del proceso : Detección de problemas y averías.
2. Aplicación de modelos de control alternativo, evitando poner en riesgo el sistema físico.
3. Modelo de entrenamiento para el control sobre plantas automatizadas, con el objetivo de predicción de futuras averías.
4. Mejora de los parámetros de diseño a través de la realimentación de los datos intercambiados entre producto físico y producto virtual [4].
5. Aumento de la efectividad de las líneas de producción en un entorno industrial [5].

Por tanto, el flujo de datos entre ambos sistemas se debe realizar de manera bidireccional:

Desde el sistema físico hacia el DT, recogiendo datos de los distintos sensores y actuadores del sistema, con el objetivo de replicar la realidad en la simulación. Dichos datos pueden ser almacenados para realizar un procesamiento posterior, con el objetivo de analizar el comportamiento del sistema y tratar de mejorar sus prestaciones.

Desde el DT hacia el sistema físico, tratando de aplicar los valores recogidos dentro de la simulación a la realidad. Dicha configuración puede ser útil para estimar de manera más precisa el comportamiento de un nuevo sistema de control que se desee implementar sobre la realidad y que haya sido desarrollado sobre la simulación.

Más allá de esto, las posibilidades que ofrecen los DT en el ámbito del Big Data y la Inteligencia Artificial, hacen de ellos unas de las tecnologías más emergentes en la Industria 4.0, por lo que se debe considerar como una de las herramientas a ser más utilizadas por la industria en los próximos años.

El objetivo de este trabajo ha sido el desarrollo de la interacción entre los sistemas físicos disponibles en una célula de fabricación flexible, ubicada en los laboratorios del departamento de Sistemas y Automática de la Universidad de Sevilla, y el DT de la misma desarrollado por el grupo del proyecto DENIM [6]. De una manera esquemática se enuncian aquí las distintas partes del desarrollo:

- Programación y coordinación de tareas entre distintos PLCs de la planta.
- Comunicación y coordinación de tareas entre distintos PLCs de la planta.
- Creación de una librería de comunicación que pueda ser usada de manera genérica PLC-computador a través de sockets, basado en el protocolo de comunicación MODBUS TCP/IP.
- Traducción de datos entre componentes del DT y elementos de memoria de los PLCs.
- Lectura de elementos de temporización de los PLCs para realizar correcciones sobre algunas variables del DT, tales como la velocidad.
- Mejora de la experiencia del usuario dentro de la simulación.

Para el desarrollo de este trabajo han sido empleadas, principalmente, las siguientes dos herramientas software:

- EcoStruxure Control Expert, para la programación y configuración de los PLCs.
- Unity3D, como entorno de desarrollo del DT.

La organización del resto del artículo es la siguiente. En la sección 2 se hace una descripción de la célula de fabricación flexible y los DT. En la sección 3 se muestra la arquitectura interna del DT. En la sección 4 se desarrolla toda la comunicación con el sistema real. En la sección 5, se exponen los resultados de la evaluación de la comunicación hecha. Por último, se describen algunas conclusiones y trabajos futuros en la sección 6.

2. Descripción del sistema físico

El entorno físico donde se ha desarrollado el trabajo y en el cual se ha desarrollado el DT corresponde con la planta situada en la planta baja de los laboratorios del departamento de Sistemas y Automática. Dicha planta está compuesta por distintos PLCs encargados de gestionar distintos elementos de la planta.



Figura 1: Vista general de la planta física

El objetivo de esta planta se basa en controlar el movimiento de cajas a través de la misma, así como la manipulación de las mismas a través de las distintas máquinas disponibles. En la fecha de publicación de este documento, se ha implementado el uso de dos PLCs:

1. **Main PLC:** PLC encargado de gestionar el movimiento de las cajas a través de las cintas, haciendo uso de los sensores inductivos y pistones neumáticos disponibles en las mismas.
2. **Stacker PLC:** PLC encargado de controlar la máquina denominada como “apilador”, con el objetivo de almacenar o servir nuevas bandejas sobre las cintas.

3. Arquitectura del gemelo digital

El DT diseñado en el grupo de investigación, para llevar a cabo las tareas del proyecto DENiM,



Figura 2: PLC principal.

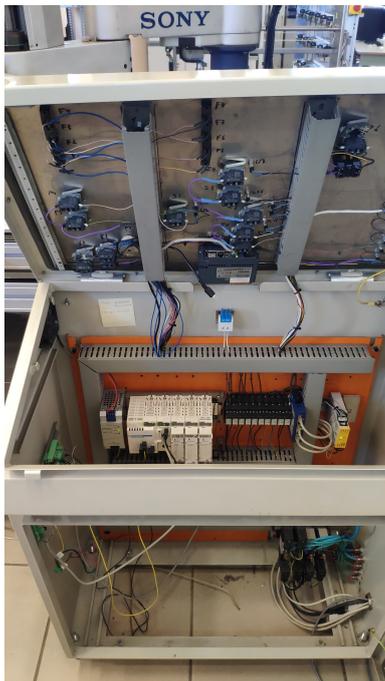


Figura 3: Cuadro del PLC del apilador

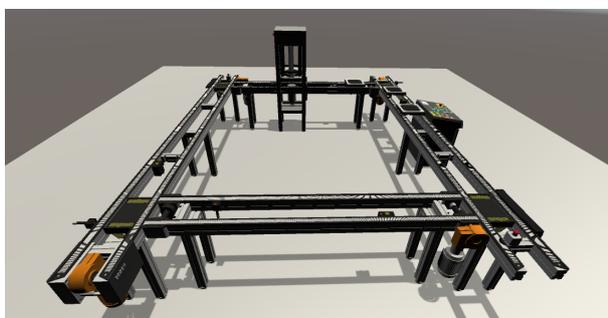


Figura 4: Imagen del DT desarrollado en el proyecto DENIM.

consta del transporte y el apilador de bandejas de la célula de fabricación flexible antes citada.

El gemelo digital diseñado, sigue el esquema presentado en la Figura 3 Existe una capa superior

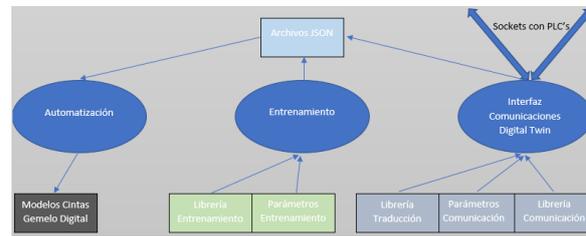


Figura 5: Esquema interno dentro del Gemelo Digital.

formada por las rutinas de **Comunicaciones, Entrenamiento y Automatización**. Dichas rutinas son ejecutadas dentro de las propias del entorno de Unity: Awake, Start, Update y LateUpdate. Los objetivos principales de dichas rutinas serán:

- **Comunicaciones** : Establecer los canales de comunicación con los PLCs y servir como elemento intermedio para la correcta adaptación de los datos entre planta física y simulación digital.
- **Entrenamiento** : Cuando sea activada, ejecutar la rutina necesaria para, a través de datos recogidos de los PLCs, corregir parámetros del Gemelo Digital.
- **Automatización** : Recoger los datos de entrenamiento para corregir los parámetros necesarios sobre los objetos correctos del Gemelo Digital. *Se debe destacar que dicho archivo ya fue creado originalmente por el equipo de DENIM. Dentro de la realización de este trabajo, dicho archivo sólo ha sido modificado para el correcto funcionamiento con los nuevos módulos creados.*

Tanto para la rutina de comunicaciones, como para la rutina de entrenamiento, se han creados archivos donde son almacenados los parámetros necesarios para su correcta ejecución : "Parámetros Comunicaciónz "Parámetros Entrenamiento", respectivamente. Dichos archivos son ejecutados en modo *lectura* desde ambas rutinas.

De manera análoga, han sido creados otros dos archivos que funcionan como librerías tanto para las comunicaciones como para el entrenamiento. En dichas librerías se almacenan las clases con los métodos necesarios para la realización del proyecto, así como las estructuras de datos necesarias

para la agilización de los algoritmos. Dichos archivos son ejecutados en modo *lectura* desde ambas rutinas.

Para la rutina de comunicaciones, se tiene acceso a otra librería denominada "Traducción". En ella se almacenan las clases con los métodos y estructura de datos necesarias para poder realizar la conversión entre los datos de memoria de ambos PLCs y los datos de los objetos presentes en la simulación. Dicho archivo es ejecutado en modo *lectura* desde dicha rutina.

De manera global, tanto las rutinas de comunicaciones y entrenamiento como la de automatización tienen acceso a archivos JSON donde son almacenados valores de velocidad para el Gemelo Digital. Es por ello que la rutina de automatización acceda a ellos en modo *lectura*, para poder actuar a continuación sobre los parámetros de los objetos presentes en la simulación.

Otra manera de visualizarlo consiste en dividir el sistema en un sistema de *tres niveles*, que puede ser útil para comprender la jerarquía dentro del sistema virtual. Se puede aproximar de la manera mostrada en la Figura 3 Así, lo desarrollado en

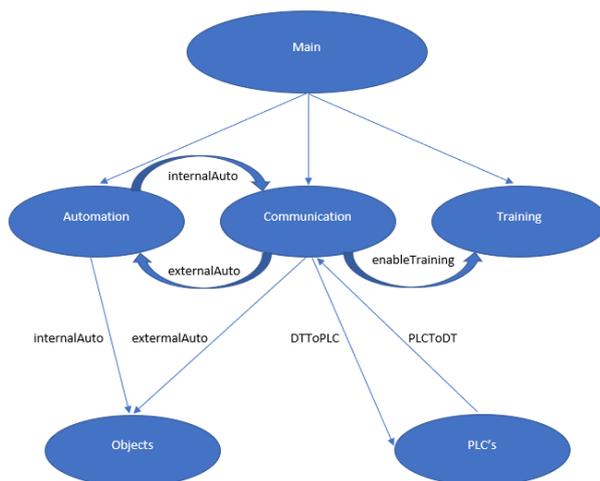


Figura 6: Estructura interna DT

cada bloque es:

Primer nivel: Se corresponde a la primera capa de abstracción del sistema desarrollado. Está formado por :

- *Main* : Se encarga de instanciar las clases del nivel inferior.

Segundo nivel: Se corresponde a la segunda capa de abstracción del sistema desarrollado. Está formado por :

- *Automation* : Se encarga de ejecutar la **automatización interna** en caso de que haya

sido escogida; o bien respeta la ejecución de la **automatización externa**.

- *Communication* : Se encarga de realizar la comunicación con los PLCs del sistema físico, bien **escribiendo** sobre ellos; o bien **leyendo** de ellos.
- *Training* : Se encarga de ejecutar el algoritmo de entrenamiento en base a los datos aportados por el algoritmo de comunicación, siempre que el entrenamiento haya sido habilitado.

Tercer nivel: Se corresponde a la tercera capa de abstracción del sistema desarrollado. Está formado por :

- *Objects* : Se encarga de gestionar las físicas y cinemáticas de los objetos presentes en el sistema virtual.
- *PLCs* : Se corresponden con los PLCs presentes en la planta. Son algo externo al propio sistema virtual.

4. Interacción de los PLCs con el DT

Respecto a la arquitectura que se ha seguido para la comunicación de los distintos elementos de la planta, se planteó originalmente un esquema se establecía un sólo canal de comunicación del DT con el "Main PLC", mientras que dicho PLC se encargaría además de gestionar la comunicación con el "Stacker PLC". Sin embargo, dicho esquema fue descartado por las siguientes razones:

- En primer lugar, supondría una mayor carga computacional sobre el "Main PLC", que debería gestionar, aparte de la escritura/lectura de datos con el Gemelo Digital; la escritura/lectura de datos con el "Stacker PLC"; así como la gestión de sus propios elementos de entrada y salida.
- En segundo lugar, este esquema suponía un gran riesgo en caso de fallo, ya que en caso de hubiera algún fallo en el "Main PLC", el Gemelo Digital no podría acceder a los datos recogidos por el "Stacker PLC". Esto podría suponer un fallo crítico en el sistema planteado.
- En tercer lugar, la cantidad de buffers que tendría que recorrer un dato que saliera desde el "Stacker PLC", pasando por un total de 4:
 1. Búfer de salida del "Stacker PLC".
 2. Búfer de entrada del "Main PLC".

3. Búfer de salida del "Main PLC".
4. Búfer de entrada del "Digital Twin".

Se decidió por tanto el esquema mostrado en la Figura 4 En dicha solución se plantea un esquema

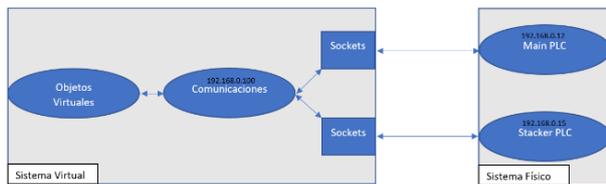


Figura 7: Esquema de comunicaciones implantado.

en "V", donde el Gemelo Digital es el encargado de establecer dos canales de comunicación con sendos PLCs. Las ventajas de usar este modelo, con el respecto al anterior, son las siguientes:

- En primer lugar, se ponía fin al problema de fallo en las comunicaciones. Con ello, en caso de que alguno de los PLCs entrara en bloqueo o sufriera algún fallo, el Gemelo Digital sería capaz de seguir comunicándose con el resto de elementos de la planta.
- En segundo lugar, se liberaba al "Main PLC" de cierta carga computacional. Sin embargo, ésta es transferida al Gemelo Digital.
- En tercer lugar, el establecimiento de las comunicaciones y gestión de datos desde el Gemelo Digital hacia el "Stacker PLC" se realiza de manera análoga a la forma en la que se realiza con el "Main PLC", por lo que resulta más sencilla su implementación.
- Y en cuarto lugar, se agilizaba la velocidad a la que un dato viajaría desde el "Stacker PLC" hacia el "Digital Twin", reduciéndose a un total de 2 búfer de recorrer.
 1. Búfer de salida del "Stacker PLC".
 2. Búfer de entrada del "Digital Twin".

Se ha desarrollado un espacio de nombres (*namespace*) de propósito general para la comunicación con cualquier PLC que siga el protocolo de comunicación MODBUS TCP/IP; así como el proceso de aplicación para el caso especial del Gemelo Digital desarrollado por DENIM. El *namespace* desarrollado posee tres elementos claves:

1. **Clase principal:** Contiene las propiedades y métodos necesarios para la comunicación. Los métodos que han sido desarrollados han sido los siguientes:

- a) **Constructor:** Se corresponde al método que se ha de llamar al crear una instancia de la clase, dando valores a sus distintas propiedades.
 - b) **Creación del socket:** Se ejecuta el proceso necesario para la creación del socket, así como el establecimiento de conexión con el PLC(servidor).
 - c) **Envío solicitud para lectura de coils :** Crea y envía el mensaje necesario para la lectura de coils discretos sobre el servidor.
 - d) **Envío solicitud para lectura de registros de 16 bits:** Crea y envía el mensaje necesario para la lectura de registros de 16 bits sobre el servidor.
 - e) **Lectura búfer de entrada para lectura de coils:** Lectura del búfer de entrada para la extracción de los datos de los coils del servidor.
 - f) **Lectura búfer de entrada para lectura de registros de 16 bits:** Lectura del búfer de entrada para la extracción de los datos de los registros de 16 bits del servidor.
 - g) **Proceso completo lectura coils:** Ejecuta todo el proceso completo para la extracción de datos de coils sobre el servidor.
 - h) **Proceso completo lectura registros de 16 bits:** Ejecuta todo el proceso completo para la extracción de datos de registros de 16 bits sobre el servidor.
 - i) **Envío solicitud para escritura de coils:** Crea y envía el mensaje necesario para la escritura sobre coils del servidor.
 - j) **Envío solicitud para escritura de registros de 16 bits:** Crea y envía el mensaje para la escritura sobre registros de 16 bits del servidor.
 - k) **Lectura búfer de entrada tras escritura:** Lectura del búfer de entrada para la obtención de la respuesta tras haber realizado la escritura sobre el servidor.
 - l) **Proceso completo escritura coils:** Ejecuta todo el proceso completo para la escritura de coils sobre el servidor.
 - m) **Proceso completo escritura registros de 16 bits:** Ejecuta todo el proceso completo para la escritura de registros de 16 bits sobre el servidor.
2. **Estructura de datos para la lectura de coils:** formada por la dirección de inicio y un array de booleanos. Es empleada para un mejor manejo de datos en las funciones que

lo requieran. Las funciones que harán uso de dicha estructura serán: $1c$, $1e$, $1g$, $1i$, $1l$.

3. **Estructura de datos para la lectura de registros de 16 bits:** Formada por la dirección de inicio y un array de palabras de 16 bits sin signo. Es empleada para un mejor manejo de datos en las funciones que lo requieran. Las funciones que harán uso de dicha estructura serán: $1d$, $1h$, $1j$, $1m$.

5. Evaluación de las comunicaciones

Más allá de tener la posibilidad de replicar la realidad en la simulación, y viceversa, se desarrolló un módulo de programación orientado a la obtención de datos de la realidad, para poder ser aplicados sobre el Gemelo Digital, para poder actualizar sus parámetros. Se ha creado una librería para la obtención de parámetros de la realidad y su aplicación sobre el Gemelo Digital. En concreto, se ha tratado de extraer las velocidades de las cintas en la realidad para su aplicación sobre el Gemelo Digital, a la escala correspondiente a su tamaño. Además de ello, se ha creado una rutina complementaria a la librería de comunicaciones y algoritmo de comunicaciones del Gemelo Digital capaz de calcular el retrasos de las mismas, con el objetivo de estimar la efectividad de la solución propuesta así como de ser útil para un posible uso en el futuro.

Se han realizado pruebas tanto para la comunicación PLC→DT así como para DT → PLC. Además, se ha trabajado con ambos PLCs y se ha calculado tanto para la lectura de coils así como de registros de 16 bits. El objetivo de la obtención de dichos resultados ha sido con el objetivo de que puedan ser útiles y a tener en cuenta en el caso de que se quisiera realizar un control que tuviera que hacer uso de las comunicaciones.

Tabla 1: Estimaciones de retrasos en las comunicaciones tras 30 ciclos de comunicación en operaciones de escritura.

DT→PLC (Operación de escritura)							
Main PLC				Stacker PLC			
Coils		Words		Coils		Words	
Avg	Desv	Avg	Desv	Avg	Desv	Avg	Desv
10.33	3.27	9.83	2.77	10.33	3.76	9.29	2.66

En la tabla 1 se presentan los valores de retrasos obtenidos tras 30 ciclos de comunicación entre Gemelo Digital y planta, en milisegundos, para la operación de escritura; mientras que en la tabla 2 se muestran los valores obtenidos para la operación de lectura. Se ha de destacar que el número

Tabla 2: Estimaciones de retrasos en las comunicaciones tras 30 ciclos de comunicación en operaciones de lectura.

PLC→DT (Operación de lectura)							
Main PLC				Stacker PLC			
Coils		Words		Coils		Words	
Avg	Desv	Avg	Desv	Avg	Desv	Avg	Desv
9.80	2.00	9.78	2.21	8.42	2.60	10.84	4.07

de elementos solicitados a leer/escribir para cada PLC no ha sido el mismo (ver tabla 3).

Tabla 3: Número de elementos de información.

DT→PLC (Operación de escritura)			
PLC→DT (Operación de lectura)			
Main PLC		Stacker PLC	
Coils	Words	Coils	Words
Num.Elmts	Num.Elmts	Num.Elmts	Num.Elmts
8	24	6	1

A su vez, en la tabla se puede observar el número de elementos de cada tipo que se han leído/escrito en cada uno de los ciclos de comunicación.

- **Para el Main PLC:** En lo que respecta a
 - **Coils:** La operación de **lectura es más rápida que la escritura**, un 5.13 %.
 - **Words:** La operación de **lectura es similar a la de escritura** en cuanto a rapidez.
- **Para el Stacker PLC:** En lo que respecta a
 - **Coils:** La operación de **lectura es más rápida que la escritura**, un 18.48 %.
 - **Words:** La operación de **escritura es ligeramente más veloz que la de lectura**, un 14.30 %.

Otra manera de analizarlo es:

- **Respecto a la operación de escritura:**
 - **Coils:** La velocidad del **Main PLC es similar a la del Stacker PLC**.
 - **Words:** La velocidad del **Stacker PLC es ligeramente superior a la del Main PLC**, un 5.49
- **Respecto a la operación de lectura:**
 - **Coils:** La velocidad del **Stacker PLC es superior a la del Main PLC**, un 14.08 %.
 - **Words:** La velocidad del **Main PLC es superior a la del Stacker PLC**, un 9.77 %.

6. Conclusiones y trabajos futuros

En este trabajo se ha creado una librería de comunicación de uso genérico en PLCs que hagan uso del protocolo MODBUS TCP/IP y la implementación de dicha librería a la arquitectura del Sistema Virtual desarrollado previamente, obteniendo un retrasos aceptables y una fiabilidad sólida; así como la creación de un sistema de traducción para la comunicación entre el Sistema Físico y el Sistema Virtual. Además se ha desarrollado una rutina de entrenamiento básica para la adaptación de las velocidades del Sistema Físico. Con este trabajo se ha creado un marco de trabajo que pretende estandarizar la inclusión de algoritmos nuevos de aprendizaje de parámetros. Se propone como trabajo futuro, hacer uso de técnicas de inteligencia artificial para el aprendizaje de los parámetros relevantes en la planta física y extender la misma metodología y arquitectura a otros gemelos digitales.

Agradecimientos

Los autores agradecen a la Comisión Europea la financiación de este trabajo en el marco del proyecto DENiM. Este proyecto ha recibido financiación del programa de investigación e innovación Horizonte 2020 de la Unión Europea en virtud del acuerdo de subvención nº 958339.

The authors thanks to the European Commission for funding this work under project DENiM. This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 958339.

English summary

Digital twin interfacing of flexible manufacturing cell with plc for automatic parameter learning.

Abstract

The implementation of a communication system that allows the exchange of data in real time between the physical system and its Digital Twin is presented. The aim of this work has been to propose a solution capable of guaranteeing not only the exchange of data between both systems, but also the adaptation of the digital twin to the physical reality, by means of automatic learning routines.

Keywords: Digital Twin, Communication between plant and its twin, parameter adaptation.

Referencias

- [1] E.Negri (2017). "A review of the roles of Digital Twin in CPS-based production systems". *Procedia Manufacturing*. 11: 939–948.
- [2] M.Grievies. "Digital twin: Manufacturing excellence through virtual factory replication.", 2014.
- [3] F. Tao, H. Zhang, A. Liu and A. Y. C. Nee. Digital Twin in Industry: State-of-the-Art, "IEEE Transactions on Industrial Informatics", vol. 15, no. 4, pp. 2405-2415, April 2019, doi: 10.1109/TII.2018.2873186 - Pag 2.
- [4] A. Canedo, "Industrial IoT lifecycle via digital twins", 2016 International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS), 2016, pp. 1-1.
- [5] H.Zhang, Q.Liu, X.Chen , D.Zhang y J.Leng, "A Digital Twin-Based Approach for Designing and Multi-Objective Optimization of Hollow Glass Production Line", IEEE Special Section On Key Technologies For Smart Factory Of Industry 4.0, 2017.
- [6] DENIM Project, <https://denim-fof.eu>
- [7] MODBUS Protocol Specification (2012, 26 de abril). Modbus ORG. MODBUS Application Protocol Specification V1.1b3.
- [8] MODBUS TCP/IP (2006, 24 de octubre). Modbus ORG. MODBUS Messaging on TCP/IP Implementation Guide V1.0b.
- [9] M. Xue and C. Zhu, "The Socket Programming and Software Design for Communication Based on Client/Server," 2009 Pacific-Asia Conference on Circuits, Communications and Systems, 2009, pp. 775-777, doi: 10.1109/PACCS.2009.89.



© 2022 by the authors. Submitted for possible open access publication under the terms and conditions of the Creative Commons Attribution CC-BY-NC-SA 4.0 license (<https://creativecommons.org/licenses/by-nc-sa/4.0/deed.es>).