

Gemelos funcionales para validar el software de control

María Luz Álvarez*, Isabel Sarachaga, Arantzazu Burgos, Nagore Iriondo

*Departamento de Ingeniería de Sistemas y Automática, Escuela de Ingeniería de Bilbao, UPV/EHU, Plaza Ingeniero Torres Quevedo, nº 1,
48013, Bilbao, España.*

To cite this article: Álvarez, M.L., Sarachaga, I., Burgos, A., Iriondo, N. 2024. Functional twins to validate control software. Revista Iberoamericana de Automática e Informática Industrial 21, 159-170. <https://doi.org/10.4995/riai.2024.20830>

Resumen

La innovación y los retos tecnológicos de la fabricación inteligente han provocado un incremento notable en la complejidad del software de control de los sistemas de producción automatizados (aPS) integrados en un entorno global interconectado. Una herramienta de pruebas muy potente para su validación es emplear plantas virtuales (uno de los pilares de la digitalización en la industria). En este contexto, este artículo contribuye con una metodología de diseño e implementación de gemelos funcionales construidos a partir de componentes funcionales básicos de biblioteca, que no precisa herramientas comerciales de desarrollo de plantas virtuales. Como representación virtual de la funcionalidad de una entidad del sistema de producción, el gemelo funcional se empleará como herramienta de pruebas para probar la reacción del sistema de control tanto en producción normal como ante la inyección de fallos. La metodología se ha aplicado en la construcción de los gemelos funcionales que permiten validar el sistema de control de una célula de ensamblaje.

Palabras clave: Control de sistemas y células de fabricación flexible e inteligente, Automatización y control con autómatas programables (PLC), Supervisión y pruebas.

Functional twins to validate control software

Abstract

The innovation and technological challenges of smart manufacturing have led to a significant increase in the complexity of control software for automated production systems (aPS) integrated in an interconnected global environment. A very powerful testing tool for their validation is to use virtual plants (one of the pillars of digitization in industry). In this context, this article contributes with a methodology for the design and implementation of functional twins built from basic functional components of library, which does not require commercial virtual plant development tools. As a virtual representation of the functionality of a production system entity, the functional twin will be used as a test tool to test the reaction of the control system both in normal production and with fault injection. The methodology has been applied in the construction of the functional twins to validate the control system of an assembly cell.

Keywords: Control of flexible and intelligent manufacturing systems and cells, Automation and control with Programmable logic controllers (PLC), Supervision and testing.

1. Introducción

La fabricación inteligente requiere integrar las Tecnologías de la Información y Comunicación (TIC) con las Tecnologías de Operación del entorno industrial. Esto redundará en una mayor complejidad del software de control de los sistemas de

producción automatizados (aPS) que, bajo el prisma de Industry 4.0, deben estar interconectados a nivel global. Por tanto, es primordial que durante su proceso de desarrollo se realicen pruebas exhaustivas que permitan evidenciar que dicho software y los productos asociados cumplen con los requisitos (funcionales y no funcionales), satisfacen los estándares, prácticas y convenciones, y responden a sus

objetivos comerciales y las expectativas de las partes interesadas. Con tal fin, el empleo de una planta virtual del sistema real constituye una herramienta de pruebas muy potente para probar el software de control antes de conectarlo a dicho sistema real. En este contexto, la tecnología de los gemelos digitales (Digital Twin – DT) es una de las más empleadas (Ferko et al., 2022), siendo una herramienta adecuada para la validación de sistemas de control (Prados et al., 2022).

No obstante, pese a las numerosas referencias a esta tecnología, el concepto de DT tiene distintas interpretaciones por parte de los autores. Con objeto de paliar esta situación, (Kritzinger et al., 2018) propone una clasificación atendiendo al nivel de integración entre el objeto virtual y el objeto físico. Así, se refiere a un Modelo Digital (Digital Model) como representación digital de un objeto físico sin intercambio automatizado de datos entre ambos; a la Sombra Digital (Digital Shadow) cuando existe un flujo de datos automatizado unidireccional del estado del objeto físico a su modelo digital; y al Gemelo Digital (Digital Twin) si el flujo es en ambos sentidos.

Por tanto, parece interesante evaluar cuál de estas opciones es la más adecuada a la hora de abordar la validación de sistemas de control, puesto que no siempre es necesario construir complejos sistemas mecatrónicos. En este sentido, construir modelos digitales sería la opción necesaria para validar el código de los PLC (Programmable Logic Controller) (Konstantinov et al., 2022). Esta opción no sólo supone un ahorro de tiempo, coste y esfuerzo de desarrollo, sino también permite acercar al personal que construye las simulaciones a la terminología utilizada por el personal que realiza las automatizaciones para reducir el esfuerzo a la hora de conectar los modelos digitales con el PLC. Pero este gap entre ambos resulta muchas veces insalvable debido al uso de herramientas propietarias o comerciales que precisan la formación de personal especializado en aspectos relacionados con el enfoque que se emplea para el desarrollo del software y/o con la infraestructura necesaria para realizar dicho desarrollo, aspectos ajenos a la automatización del proceso.

Aparte del uso de buenas prácticas de la automatización, el desarrollo de este tipo de software debe contemplar los retos identificados en la revisión bibliográfica reciente: incorporación de aspectos de Ingeniería del Software (IS) para abordar la reutilización, escalabilidad, mantenibilidad, etc. (Qamsane et al., 2021), y un enfoque modular con directrices para crear bibliotecas de componentes y metodologías genéricas que usen dichas bibliotecas para facilitar la construcción de nuevas aplicaciones (Shao et al., 2023).

En este contexto, este trabajo aporta una metodología, que contempla buenas prácticas de automatización e IS, para construir una representación funcional del proceso real que permite simular el comportamiento del aPS sin intercambio automatizado de datos con dicho aPS. Este Modelo Digital se puede construir con distintos niveles de complejidad de acuerdo con el Modelo Físico del estándar IEC 61512 (IEC, 2002) de la arquitectura RAMI (Reference Architecture Model Industry) 4.0 (IEC, 2017), que organiza jerárquicamente el aPS en células de proceso, unidades de trabajo, módulos de equipo y módulos de control. Así, el Gemelo Funcional (GF) es un Modelo Digital asociado a una unidad de trabajo, que se

construye agregando Modelos Digitales de los equipos de los niveles inferiores de la jerarquía, denominados Componentes Funcionales Básicos (CFB), en una estructura jerárquica. Estos CFB son representaciones funcionales de módulos de control y módulos de equipo que están almacenados en bibliotecas.

Asimismo, cuando se considera un enfoque jerárquico en la estrategia de control de la célula de proceso, se pueden construir GF de los sistemas de control de las unidades de trabajo, los cuales se construyen a partir de los CFB de operaciones de decisión, producción y auxiliares identificadas en el Modelo de Proceso que se implementan con los procedimientos recogidos en el Modelo de Control Procedural (ambos modelos del estándar IEC 61512).

Además, los CFB pueden incluir inyección de fallos. De este modo, no sólo se valida el sistema de control, sino que se puede comprobar su reactividad ante acontecimientos inciertos. La implementación de estos Modelos Digitales se realiza siguiendo el estándar IEC 61131-3 (IEC, 2013) en el mismo entorno en el que se desarrolla el software de control, no requiriendo herramientas comerciales de desarrollo de plantas virtuales para su construcción.

El artículo se estructura en seis apartados. El apartado 2 está dedicado a la revisión de la bibliografía reciente con objeto de trasladar las últimas tendencias al desarrollo de los GF. En el apartado 3 se expone el escenario de aplicación de los GF y los CFB para realizar las distintas pruebas a las que se va a someter el sistema de control. El apartado 4 se centra en la construcción de estas herramientas de prueba, describiendo la metodología de desarrollo de los CFB con ejemplo de aplicación a un módulo de control de un cilindro de doble efecto, así como la metodología de desarrollo de los GF como composición de CFB reutilizables almacenados en biblioteca. En el caso de estudio, que se presenta en el apartado 5, se aplica la metodología de construcción de GF en una célula de ensamblaje, desarrollando tanto el GF de una unidad de trabajo como el GF del sistema de control de dicha unidad. Por último, el apartado 6 recoge las conclusiones de este trabajo.

2. Revisión bibliográfica

En los últimos años, se han publicado numerosos trabajos relacionados con el estado del arte de los DT (Jones et al., 2020), (Semeraro et al., 2021), (Mihai et al., 2022), la evaluación de las tecnologías habilitadoras, los retos y la investigación (Fuller et al., 2020), (Rasheed et al., 2020), (Qi et al., 2021), o las mejores prácticas de empresas punteras en diversos campos, como la ingeniería aeroespacial, la red eléctrica, la fabricación de automóviles, la industria petrolera, la sanidad, etc. (Tao et al., 2019). Pero gran parte de la investigación se enmarca en el ámbito de la industria manufacturera (Kritzinger et al., 2018), (Cimino et al., 2019), (Mylonas et al., 2021). En este ámbito, su finalidad principal consiste en la mejora de la operación de los procesos, realizando el control y la optimización de los mismos, aunque también se esperan otras capacidades avanzadas de planificación de escenarios, apoyo a la toma de decisiones y validación durante el ciclo de vida. Precisamente estas características marcan la diferencia respecto a las simulaciones convencionales (de Prada et al., 2022).

No obstante, se trata de un área con muchos subcampos, desde pequeña maquinaria como robots (Prados et al., 2022),

(Montenegro-Bravo et al., 2023) hasta fábricas inteligentes en las que algunos escenarios de aplicación se centran en minimizar el impacto del tiempo de inactividad de los equipos, la optimización de la planificación de la producción, la supervisión del rendimiento de los nuevos equipos durante la puesta en servicio (Shao et al., 2020), la monitorización del consumo de energía (Cimino et al., 2019) o el mantenimiento predictivo (Chen et al., 2023), entre otros, con objeto de permitir la optimización del sistema y la mejora continua. Otros escenarios son más ambiciosos proponiendo una fabricación conducida por los DT con objeto de proporcionar una supervisión más transparente del proceso de fabricación, un análisis más exhaustivo de dicho proceso y un control más preciso de la fabricación (Liu et al., 2023).

También existen numerosos trabajos relacionados con la infraestructura necesaria para crear y mantener los DT garantizando su operatividad en plantas de fabricación de motores diesel marinos (Hu et al., 2021), de piezas mecánicas (Zhang et al., 2020) o de turbinas de vapor (Cheng et al., 2020), entre otros. Destacar el trabajo de (Moyné et al., 2020), ya que su framework contempla gran parte de los requisitos derivados de la visión actual y futura de los DT aplicándolo en un caso de estudio de planificación y control para la fabricación de obleas de semiconductores.

Otros autores integran TIC avanzadas en sus frameworks con objeto de propiciar fabricación inteligente: Internet Industrial para integrar tres perspectivas (el ciclo de vida del producto, el nivel intraempresarial y el nivel interempresarial) (Cheng et al., 2020); CAM (Computer Aided Manufacturing) para acoplar los DT de la pieza, la herramienta de corte, la sujeción, la máquina herramienta y del proceso de mecanizado digital (Plakhotnik et al., 2021); análisis y aprendizaje de Big Data e Internet Industrial para propiciar fabricación de equipos de conmutación de potencia (Wang et al., 2021); servicios de fabricación inteligentes basados en Inteligencia Artificial (IA) y tecnologías de contenedores de software en una infraestructura Cloud, Fog y dispositivos IoT (Internet of Things) en planta para crear gemelos del producto, la maquinaria y el proceso de fabricación (Hung et al., 2022); esquemas de Deep Learning que integran redes neuronales combinados con un algoritmo de aprendizaje para el reconocimiento en tiempo real de maquinaria, producto y personal de planta (Zhou et al., 2022); o capacidades cognitivas y de aprendizaje para pasar de la digitalización a la intelectualización de procesos de fabricación conducidos por los DT (Liu et al., 2023).

Dada la cantidad de frameworks con diferentes objetivos y distintos dominios de aplicación, parece necesaria su normalización. Así surge el estándar ISO 23247-2 "Automation systems and integration - Digital twin framework for manufacturing" (ISO, 2021), que proporciona un framework genérico basado en IoT para desarrollar los DT en el dominio de la fabricación discreta. Ejemplos de su aplicación se describen en (Ganser et al., 2021) que presenta un framework basado en IIoT (Industrial IoT) con dispositivos localizados en el Edge, Fog o Cloud para mecanizado, o en (Farhadi et al., 2022) que contempla el desarrollo de los DT para la máquina, el producto y el proceso, así como algoritmos inteligentes basados en el aprendizaje automático y las redes

neuronales artificiales para la toma de decisiones aplicándolo en el proceso de taladrado robotizado.

Desde una perspectiva software, el análisis de todos estos trabajos ha puesto de manifiesto la carencia de arquitecturas y metodologías de desarrollo que contemplen aspectos de IS y, en particular, aquellas que consideren reutilización, interoperabilidad, mantenibilidad, escalabilidad, etc. En este sentido, (Shao et al., 2023) analiza el estándar ISO 23247 y destaca la necesidad de realizar extensiones para incluir más capacidades, entre las que cabe destacar el abordar directrices para crear bibliotecas de componentes y plantillas para organizar la información y los modelos, así como metodologías y arquitecturas genéricas que usen dichas bibliotecas. Asimismo, destacan la importancia de incorporar los aspectos de verificación y validación en dichas metodologías.

En lo que respecta a arquitecturas, (Ferko et al., 2022) analiza los trabajos relacionados en un sentido amplio, pues contempla modelos de referencia, arquitecturas de referencia, patrones arquitectónicos y arquitectura de software. Estos autores debaten sus resultados con expertos en arquitecturas software para concluir que la gran mayoría de los estudios proponen modelos de referencia funcionales específicos de un dominio que se validan con prototipos o casos de uso simulados en lugar de usar sistemas industriales reales. Con objeto de abordar la falta de interoperabilidad, flexibilidad y escalabilidad de los enfoques previos, (Bowman et al., 2022) presenta un enfoque unificado con interfaces estandarizados transversales a varios sectores industriales. Se trata de una arquitectura horizontal que proporciona flexibilidad, facilidad de extensión y escalabilidad para integrar los DT de plantas nucleares. Pero, por el momento, las soluciones arquitectónicas no están maduras y se requiere mayor esfuerzo para formalizar arquitecturas de referencia.

En cuanto a metodologías de desarrollo, (Aivaliotis et al., 2019) guía la construcción de un modelo basado en la estructura física de la máquina real que combinan con modelos de degradación para su uso en aplicaciones de mantenimiento predictivo. Su metodología establece las acciones para definir el modelo del comportamiento cinemático y dinámico, realizar el modelo de los sensores virtuales para capturar los datos y establecer los parámetros que representa el estado real.

Sin embargo, este enfoque basado en la estructura física no es el único que adoptan otras metodologías. (Adamenko et al., 2020) describe las ventajas e inconvenientes de los métodos de diseño basados en sus datos estructurados según determinados criterios (como funcionalidades o montaje del objeto físico) y los basados en el propio sistema que se centran en los detalles técnicos del objeto físico, para concluir que los resultantes de la combinación de ambos métodos proporcionan mayor valor añadido.

Con independencia del enfoque, algunas metodologías emplean tecnologías cercanas al dominio de la automatización. (Negri et al., 2019) propone una metodología basada en una máquina de estados para determinar el consumo energético de las operaciones de un sistema de producción (cadena de montaje de un laboratorio). Los estados se asemejan a los contemplados en el diagrama procedural de estados del estándar IEC 61512 (IEC, 2002) de la arquitectura RAMI (IEC, 2017), que no son tan intuitivos para el personal de

planta como los identificados en la guía GEMMA (Guide des Modes d'Etude et d'Arrêts Marches) (ADEPA, 1981). (Schroeder et al., 2021) presenta una metodología basada en ingeniería conducida por modelos (MDE – Model Driven Engineering) que utiliza como tecnología subyacente AutomationML para realizar los modelos del dispositivo físico y del DT como composición de componentes básicos, y su despliegue, mediante servidor web, en una refinería de petróleo con diferentes topologías de composición para probar los aspectos de reutilización y flexibilidad. Pese a usar AutomationML (AutomationML, 2023) que es un lenguaje de modelado de datos orientado a objetos basado en XML (eXtensible Markup Language) para el intercambio de datos en el sector industrial, las tecnologías de MDE, que son adecuadas para el desarrollo de sistemas complejos, están alejadas del ámbito de la automatización. También se aplican estas últimas tecnologías en (Qamsane et al., 2021). Su metodología presenta un ciclo de desarrollo bien estructurado, según las etapas típicas de un desarrollo software, que mapea los requisitos en un modelo jerárquico orientado a objetos, el cual define el modo en el que los componentes del DT se combinan para obtener el modelo final por agregación. Posteriormente se aplican técnicas de modelado para transformar el modelo en un gemelo operativo.

Otros autores proponen disminuir la granularidad para identificar patrones que capturen el comportamiento de sistemas independientemente del dominio de aplicación e integrar los constructores disponibles en bibliotecas para construir el DT (Semeraro et al., 2023a). Este enfoque se ha validado en un proceso de fabricación de herrajes para marcos de ventanas civiles y en (Semeraro et al., 2023b) en sistemas de almacenamiento de energía en baterías que se ha convertido en un componente crucial en el sector energético. Sin embargo, este tipo de enfoques requieren conocimientos sobre estas prácticas de IS, lo que conlleva una escasa aceptación en el ámbito de la automatización industrial.

Uno de los escenarios más relevantes para el presente trabajo es precisamente la puesta en marcha virtual o Virtual Commissioning (VC). Así, (Orive et al., 2019) presenta una metodología siguiendo los pasos establecidos en la fase de fallos de MeIA (MEthodology for Industrial Automation) (Álvarez et al., 2013) que se extiende con un módulo de inyección de fallos de proceso para probar la reacción del sistema de control en su puesta en marcha virtual. Estos autores ven la necesidad de introducir estas tecnologías en los planes de estudios universitarios relacionados con la automatización, y en (Iriando et al., 2020) describen los pasos tanto para construir un DT con una herramienta de modelado industrial, como para realizar la puesta en marcha virtual usando Hardware in the Loop (HiL) con objeto de validar el sistema de automatización. En (Barbieri et al., 2021) se presenta una metodología basada en VC para diseñar y verificar DT que emplea la definición de los distintos niveles de integración entre el espacio físico y el cibernético de (Kritzinger et al., 2018). (Konstantinov et al., 2022) analizan los métodos actuales de VC para construir una metodología con dicho enfoque que dé soporte a todo el ciclo de vida de un sistema de fabricación desde las perspectivas del fabricante de maquinaria, el integrador de sistemas y el usuario final. Pero en todos estos trabajos se usan herramientas propietarias o

comerciales más o menos cercanas a la automatización, que requieren formar a personal especializado.

Por tanto, del análisis de la bibliografía reciente se desprende que no hay un consenso en la definición de DT debido quizás a la amplia variedad de campos de aplicación. Su proceso de desarrollo, por lo general ad-hoc, es complejo, costoso, requiere mucho tiempo, carece de metodologías que aporten directrices a lo largo de todo su ciclo de vida con fundamentos tanto de IS como de automatización, y está sujeto a aplicaciones propietarias o comerciales que no sólo van a condicionar su evolución sino también van a precisar personal especializado para su mantenimiento. Además, este proceso se ve limitado por la infraestructura, ya que, pese a la existencia de productos comerciales, todavía existen dificultades técnicas por subsanar, como por ejemplo el control del proceso en tiempo real (Liu et al., 2023), que precisan alguna solución para poder consolidar esta tecnología en el ámbito de la fabricación.

La puesta en marcha virtual ofrece un apoyo significativo para la validación de los sistemas de control, pero precisa mayor atención la creación de un DT adecuado para VC (Konstantinov et al., 2022). Además, la creación de estos DT adolece de metodologías que integren tanto buenas prácticas de IS como de automatización y de herramientas de implementación familiares para el personal que realiza las automatizaciones con objeto de facilitar la interconexión del DT y el PLC. Este es precisamente el alcance del trabajo que se presenta en este artículo.

3. Escenario de pruebas

Un sistema automatizado consta de la parte operativa (unidad de trabajo del aPS a controlar, panel de operación y HMI - Human Machine Interface) controlada por la parte de mando (software de control), tal y como se observa en la Figura 1. Este software recibe información de los sensores del aPS y órdenes de los paneles de operación y del HMI, y activa los accionamientos, que llevan a cabo las operaciones en el aPS, y los elementos de visualización y de aviso en el panel operación y en el HMI. En este escenario, un modelo digital que simule el comportamiento del aPS a controlar permitirá realizar las pruebas al software de control.

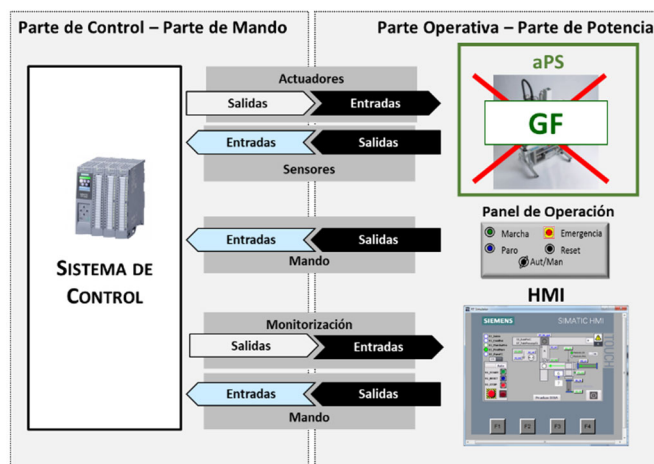


Figura 1: Escenario de aplicación del GF. El GF reemplaza al aPS real, simulando su comportamiento, para ejecutar las pruebas del sistema de control.

El sistema de control del aPS se construye de acuerdo con la metodología MeiA 4.0 (Burgos et al., 2021), que combina la madurez de las disciplinas de IS, usando los procesos del estándar ISO/IEC/IEEE 12207 “Procesos del ciclo de vida del software” (ISO/IEC/IEEE, 2017), con los métodos y estándares del campo de la automatización industrial contemplados en distintas iniciativas de la fabricación inteligente (IEC 61512 e IEC 61131, entre otros).

Así, el sistema de control consta de tantos componentes MeiA (Iriondo et al., 2022) como unidades de trabajo del Modelo Físico del estándar IEC 61512 de RAMI 4.0, dado que cada componente MeiA encapsula el sistema de control de una unidad de trabajo. Además, cuando el control del aPS global sigue una estrategia jerárquica, se requiere un componente MeiA adicional para realizar el control general. Estos componentes proporcionan diferentes modos de mando del sistema (automático, manual, pruebas, fallos, emergencia y reconfiguración) e interactúan con otros componentes y/o procesos de gestión empresarial ofreciendo su funcionalidad a través de una interfaz que oculta sus aspectos de implementación.

Cada componente MeiA se implementa en un proyecto de automatización IEC 61131-3 (IEC, 2013) siguiendo una serie de pasos, los cuales incluyen la construcción de los GF a partir de componentes de biblioteca, de acuerdo al modelo en V de la figura (Figura 2). Como se puede observar en dicha figura, la rama descendente aborda los procesos que transforman los requisitos en código, realizando también pruebas estáticas a los entregables para detectar errores tempranos. En la rama ascendente se distinguen los procesos necesarios para realizar las distintas pruebas sobre el software desarrollado que ya incluyen técnicas dinámicas. Este último tipo de técnicas deben ir acompañadas de las correspondientes pruebas de regresión para comprobar que no existen nuevos errores derivados de la modificación, actualización o corrección de los fallos detectados al realizar las pruebas unitarias, pruebas de integración, y pruebas de verificación y validación.

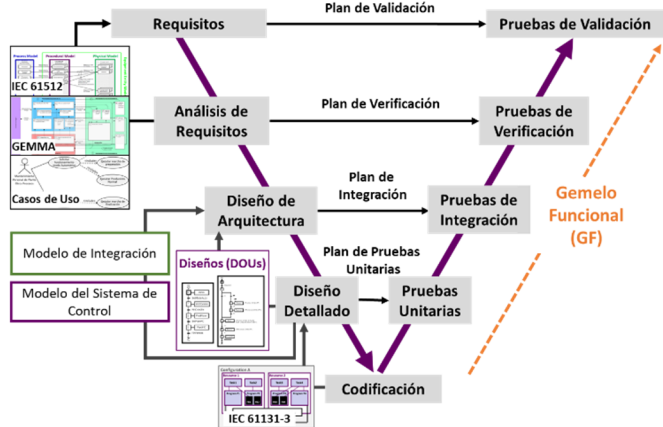


Figura 2: Modelo en V para la construcción de cada uno de los componentes MeiA. Este modelo establece la secuencia de los procesos del estándar ISO/IEC/IEEE 12207 (Procesos del ciclo de vida del software), que usan métodos habituales en la automatización industrial, para transformar los requisitos en un proyecto de automatización conforme IEC 61131-3.

Destacar que las ramas de este modelo en V no son independientes, sino que existe una relación horizontal por niveles. Por ejemplo, una vez que se dispone de la arquitectura

del componente, ya se está en disposición de establecer las pruebas que se llevarán a cabo en el momento de la integración.

En los siguientes subapartados se detallan las pruebas a realizar a los componentes MeiA que controlan las unidades de trabajo, como paso previo a la posterior integración de estos componentes MeiA y el componente MeiA del control general. Dado el área en el que está inmerso el presente trabajo, en todas ellas se propone el GF como herramienta de pruebas.

3.1. Pruebas unitarias

El objetivo de las pruebas unitarias consiste en comprobar el correcto funcionamiento de todas y cada una de las unidades de programa codificadas (POU - Program Organization Unit). Con tal fin, se realizan pruebas estructurales y pruebas funcionales a cada POU.

Las pruebas estructurales, que se aplican a la parte secuencial, se centran en probar que todos los caminos de evolución se recorren de forma adecuada, comprobando todas las transiciones, así como la correcta activación y desactivación de cada etapa.

Las pruebas funcionales, que se aplican a la parte combinatorial, permiten comprobar que las acciones asociadas a la activación de las etapas se desempeñan de forma correcta. También se realizan pruebas funcionales de operación para comprobar que la funcionalidad del POU se realiza correctamente.

Habitualmente, los sistemas de simulación que ofrecen los entornos de programación de PLCs y/o el HMI son las herramientas utilizadas en la realización de las pruebas unitarias. Ambas herramientas permiten tanto activar las entradas al software (salidas de la unidad de trabajo del aPS asociadas con los sensores y los controles de mando), como visualizar el resto de las señales.

3.2. Pruebas de integración

El objetivo de las pruebas de integración es comprobar la correcta coordinación y sincronización entre los POU. Para ello, se precisan pruebas estructurales y pruebas funcionales.

En lo que respecta a las primeras, las pruebas de sincronización permiten testear las señales de coordinación entre los distintos POU. Concretamente, se coge el POU de mando de cada modo de funcionamiento (automático, manual, fallos, emergencia, ...) y se comprueba que se activan correctamente tanto los procedimientos auxiliares como las señales de mando.

En relación con las segundas, las pruebas de operación consisten en poner la unidad de trabajo en todos los estados identificados y validar todas las operaciones involucradas en cada uno de dichos estados (arranque, ejecución y parada correcta de cada operación). Además, también se realizan pruebas específicas para operaciones en paralelo.

3.3. Pruebas de verificación y de validación

El objetivo de las pruebas de verificación y validación es comprobar que el componente MeiA satisface tanto los requisitos funcionales y no funcionales como las expectativas de las partes interesadas.

Como herramienta para realizar las pruebas de verificación y validación se propone el uso del GF con inyección de fallos.

4. Herramientas de prueba

Como ya se ha comentado, el sistema de control consta de tantos componentes MeIA como unidades de trabajo, que encapsulan el sistema de control de cada una de ellas, más otro componente que realiza el control general siguiendo una estrategia jerárquica. Por lo tanto, para realizar las pruebas al sistema de control se precisan modelos digitales que simulen el comportamiento de cada unidad de trabajo del aPS para probar su software de control, así como modelos digitales de los propios componentes MeIA para probar el software del control general. Consecuentemente, dado que el GF simula el comportamiento de la entidad controlada, se distinguen dos tipos: el GF de las unidades de trabajo del aPS (Álvarez et al., 2023) y el GF de los componentes MeIA que controlan estas unidades de trabajo.

En lo que respecta al primer tipo, considerando el escenario de la Figura 1, para realizar las pruebas al software de control se precisa un modelo digital que simule el comportamiento del aPS a controlar. Así, el GF es un componente software que responde a las órdenes del software de control (entradas al aPS) con la activación de los sensores (salidas del aPS), simulando el comportamiento del aPS.

La construcción del GF se basa en el Modelo Físico de la unidad de trabajo del aPS, que organiza los equipos jerárquicamente atendiendo a las operaciones que realizan, teniendo en cuenta los principios de subdivisión aportados por el estándar IEC 61512. Dado que la efectividad de este enfoque modular se puede ver comprometida si se realizan subdivisiones de equipos inapropiadas, se aconseja la delimitación gráfica del perímetro de los equipos sobre el layout de la unidad de trabajo del aPS para facilitar la identificación de: (1) las operaciones principales, los equipos que las realizan y los procedimientos que las comandan, así como su orden de ejecución; (2) las relaciones entre los procedimientos y las medidas de seguridad para evitar choques, caídas de piezas, bloqueos, etc.; (3) las operaciones auxiliares de preparación (cargas, llenados, calentamiento de equipos, etc.), de finalización (vaciado, limpieza de equipos, calibrados, etc.), entre otras; y (4) las señales de coordinación entre operaciones.

El GF se construye a partir de los CFB que son modelos digitales de los módulos de control o de equipo de su Modelo Físico y simulan la funcionalidad (opcionalmente, con inyección de fallos) de equipos tales como cilindros, pinzas, cintas transportadoras, elevadores, ventosas, almacenes, garras, mesas giratorias, etc. La agregación de estos CFB permite obtener modelos digitales que simulan elementos de nivel superior en la jerarquía del Modelo Físico (módulos de control o módulos de equipo compuestos) hasta obtener los GF de unidades de trabajo (Figura 3). Por tanto, el diseño del GF requiere identificar los módulos de control o equipamiento de la unidad de trabajo del aPS y encontrar los CFB correspondientes a dichos módulos en la biblioteca de componentes. Cabe destacar que estos CFB trabajan con el tiempo empleado en realizar acciones en la unidad de trabajo. Por ejemplo, el tiempo de avance o retroceso de un cilindro, el tiempo de desplazamiento de una pieza de una posición a otra dentro de la unidad de trabajo, etc.

Por otra parte, al considerar un componente MeIA que controla una unidad de trabajo, su GF simula las operaciones de decisión, producción y auxiliares que debe realizar la unidad de trabajo del aPS. El GF se construye como combinación de los CFB de las operaciones identificadas en su Modelo de Proceso que se ejecutan con los procedimientos recogidos en el Modelo de Control Procedural (ambos modelos del estándar IEC 61512). Estos CFB trabajan con el tiempo en realizar las operaciones de decisión, producción y auxiliares.

Con independencia del tipo, un GF se construye a partir de CFB reutilizables disponibles en bibliotecas. Por ello, en los siguientes subapartados se describe tanto el proceso de desarrollo de un CFB como el proceso para combinarlos con objeto de construir un GF.

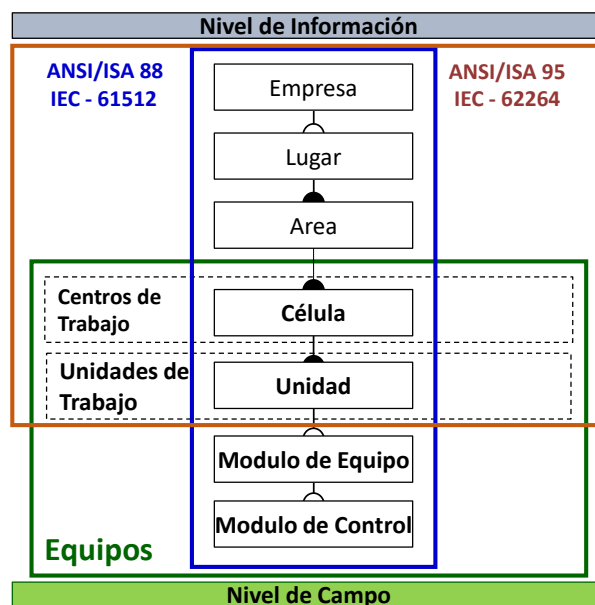


Figura 3: Modelo Físico (estándar IEC 61512) para Sistemas Discretos.

4.1. Desarrollo de un Componente Funcional Básico (CFB)

La Figura 4 muestra un sinóptico de los pasos para desarrollar un CFB, los cuales se describen a continuación:

1. Identificación de las señales de entrada y salida del equipo cuando se trata de un CFB para un módulo de control o un módulo de equipo. En el caso de un CFB de una operación de decisión, producción o auxiliar, identificación de las señales de activación y desactivación.
2. Análisis de la funcionalidad del módulo de control o módulo de equipo, o bien de la operación de decisión, producción o auxiliar del Modelo de Proceso correspondiente.
3. Análisis dinámico para establecer los parámetros de tiempo necesarios para simular su comportamiento: tiempos de respuesta de los módulos ante una variación en sus entradas (tiempos de reacción, tiempos muertos, constantes de tiempo, tiempos de pico o tiempos de establecimiento) o tiempo necesario para realizar la operación de decisión, producción o auxiliar. Estos tiempos pueden ser adaptados para la realización de las pruebas con objeto de facilitar la inspección de la ejecución de las operaciones.

4. Diseño detallado de la funcionalidad en lenguaje de modelado GRAFCET (GRAPhe Fonctionnel de Commande, Etapes, Transitions).
5. Análisis de fallos del módulo (opcional). Si el CFB se va a preparar para simular fallos, se analizan los posibles fallos que pueden producirse en el funcionamiento habitual.
6. Diseño detallado de la funcionalidad con fallos (opcional). Para simular el comportamiento ante fallos, se añaden dos señales que permiten seleccionar el tipo de fallo (*CodFallo*) y activar la simulación con fallos (*ActivarFallo*).
7. Implementación del CFB conforme al estándar IEC 61131-3. Cada componente se implementa como un POU de tipo FB (Functional Block) en lenguaje estructurado (ST) para que los componentes tengan el mayor grado de reutilización. La parte secuencial de los Grafquets se implementa en lenguaje ST siguiendo el método propuesto en (Burgos et al., 2020) y la parte combinacional se programa tras la parte secuencial.
8. Pruebas del CFB tanto estructurales (caja blanca) como funcionales (caja negra).
9. Documentación del CFB tomando como ejemplo la plantilla de la Tabla 1.
10. Inclusión del CFB en la biblioteca.

A modo de ejemplo, se describe la construcción del CFB para el módulo de control de un cilindro. La Figura 5 presenta el Modelo Físico de dicho módulo de control que consta de un cilindro de doble efecto con válvula 5/3, muelle mecánico y centro cerrado, y de dos sensores de final de recorrido. Los pasos de desarrollo son los siguientes:

1. *Identificación de señales de entrada y salida.*

Las entradas al CFB serán las señales de Extender (*CilExt*) y Recoger (*CilRec*) asociadas a la válvula biestable. Las salidas serán las señales de los sensores que detectan si el cilindro se encuentra extendido (*senCilExt*) o recogido (*senCilRec*).

2. *Análisis de la funcionalidad del cilindro.*

Cuando se activa la señal de *Extender* del cilindro y éste no se encuentra *extendido*, comienza la operación de extender. Cuando se activa la señal de *Recoger* del cilindro y éste no se encuentra *recogido*, comienza la operación de recoger. Los sensores de final de recorrido se activarán y desactivarán en función de la posición del cilindro. Si la señal de *Extender* o *Recoger* se desactiva en mitad del recorrido, el cilindro se detendrá en una posición intermedia (ninguno de los sensores de final de recorrido estará activado).

3. *Análisis dinámico del cilindro.*

Para simular el comportamiento del cilindro se identifican tres parámetros de tiempo: (1) el tiempo que transcurre desde que se da la orden de Extender o Recoger y se desactiva el sensor correspondiente que señala su posición de recogido o extendido (*TiempoQuitarSensor*); (2) el tiempo que tarda en realizar el recorrido de avance (*TiempoExtender*); y (3) el tiempo que tarda en realizar el recorrido de retroceso (*TiempoRecoger*).

4. *Diseño detallado de la funcionalidad del CFB.*

En la Figura 6 se muestra el grafcet con el diseño detallado de la funcionalidad incluidos los fallos.

5. *Análisis de Fallos (opcional).*

En la Tabla 2 se describen cuatro posibles situaciones de fallo en el cilindro.

6. *Diseño detallado de la funcionalidad con fallos (opcional).*

En la Figura 6 se muestra el grafcet correspondiente.

7. *Implementación del CFB según el estándar IEC-61131-3.*

El CFB se ha implementado en un POU tipo FB en lenguaje estructurado (ST).

Por último, una vez realizadas las fases 8 y 9 (pruebas y documentación), el CFB se incluye en la biblioteca de componentes reutilizables independiente de plataforma. Estos componentes son los elementos básicos a partir de los cuales se construirá el GF.

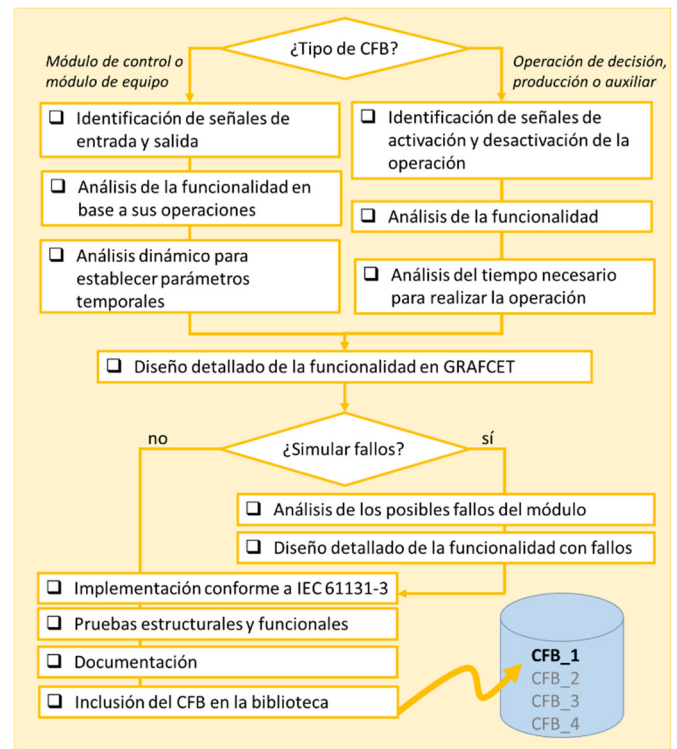


Figura 4: Pasos para el desarrollo de un CFB.

Tabla 1: Documentación de un CFB

Id CFB	Identificación del CFB
Metadatos	Datos sobre el hardware y software compatible, versión del lenguaje, actualizaciones del CFB,...
Palabras clave	Términos de búsqueda del CFB
Descripción Funcional	Descripción del funcionamiento del CFB
Modelo de Proceso	Operaciones a nivel de acciones
Módulo Físico	Módulo de Control o Módulo de Equipo simple o compuesto
Fallos	Identificación y descripción de fallos
Señales	Entradas (acciones), salidas (sensores), temporizaciones, activación fallos,...
Interfaz	Parametrización de las señales
Diseño Detallado	GRAFCET
HMI	Representación del elemento en el HMI
Esquemas	Eléctricos, neumáticos, etc.
Referencias a módulos comerciales	Referencias técnicas de fabricantes

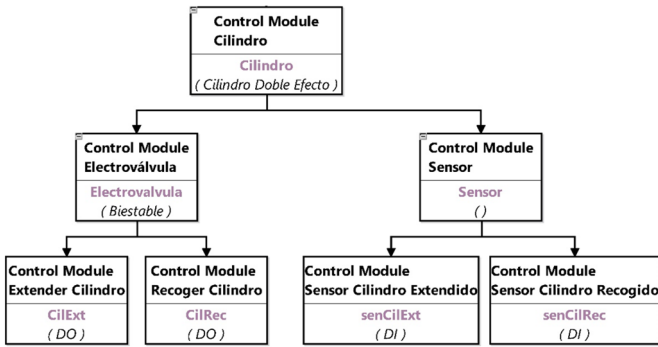


Figura 5: Modelo Físico del módulo de control de un cilindro de doble efecto. La descomposición se lleva hasta la identificación de las señales de entrada y salida del módulo de control.

4.2. Desarrollo de un Gemelo Funcional (GF)

La Figura 7 muestra un sinóptico de los pasos para desarrollar un GF, los cuales se describen a continuación:

1. Identificación de los módulos de equipo y/o de control de la unidad de trabajo en el Modelo Físico del aPS, para un GF de una unidad de trabajo. En el caso de un GF de un componente MeiA, identificación de las operaciones de decisión, producción y auxiliares recogidas en su Modelo de Proceso, las cuales se ejecutan con los procedimientos descritos en el Modelo de Control Procedural.
2. Búsqueda de los CFB identificados en la biblioteca de componentes. En caso de no existir el CFB, se deberá desarrollar siguiendo los pasos descritos.

Tabla 2: Análisis de fallos del cilindro

Código fallo	Descripción	Detección
Fallo 1	Cilindro atascado o fallo del sensor de cilindro extendido	Tras la orden de Extender y transcurrido el tiempo de avance, la señal de sensor extendido no se activa
Fallo 2	Cilindro atascado o fallo del sensor de cilindro recogido	Tras la orden de Recoger y transcurrido el tiempo de retroceso, la señal de sensor recogido no se activa
Fallo 3	La electroválvula no responde a la señal de Extender	Tras la orden de Extender, el sensor de recogido no se desactiva y el sensor de extendido no se activa transcurrido el tiempo de avance
Fallo 4	La electroválvula no responde a la señal de Recoger	Tras la orden de Recoger, el sensor de extendido no se desactiva y el sensor de recogido no se activa transcurrido el tiempo de retroceso

3. Implementación del GF conforme al estándar IEC 61131-3. El GF se implementa como un POU de tipo FB en lenguaje estructurado (ST) o de contactos (LD). En la Interfaz de Bloque se definen las entradas y salidas del GF. Para el GF de la unidad de trabajo, las entradas se corresponden con las salidas del software de control que ejecutan las órdenes sobre la entidad controlada. Las entradas/salidas del GF se corresponden con las entradas del software de control que

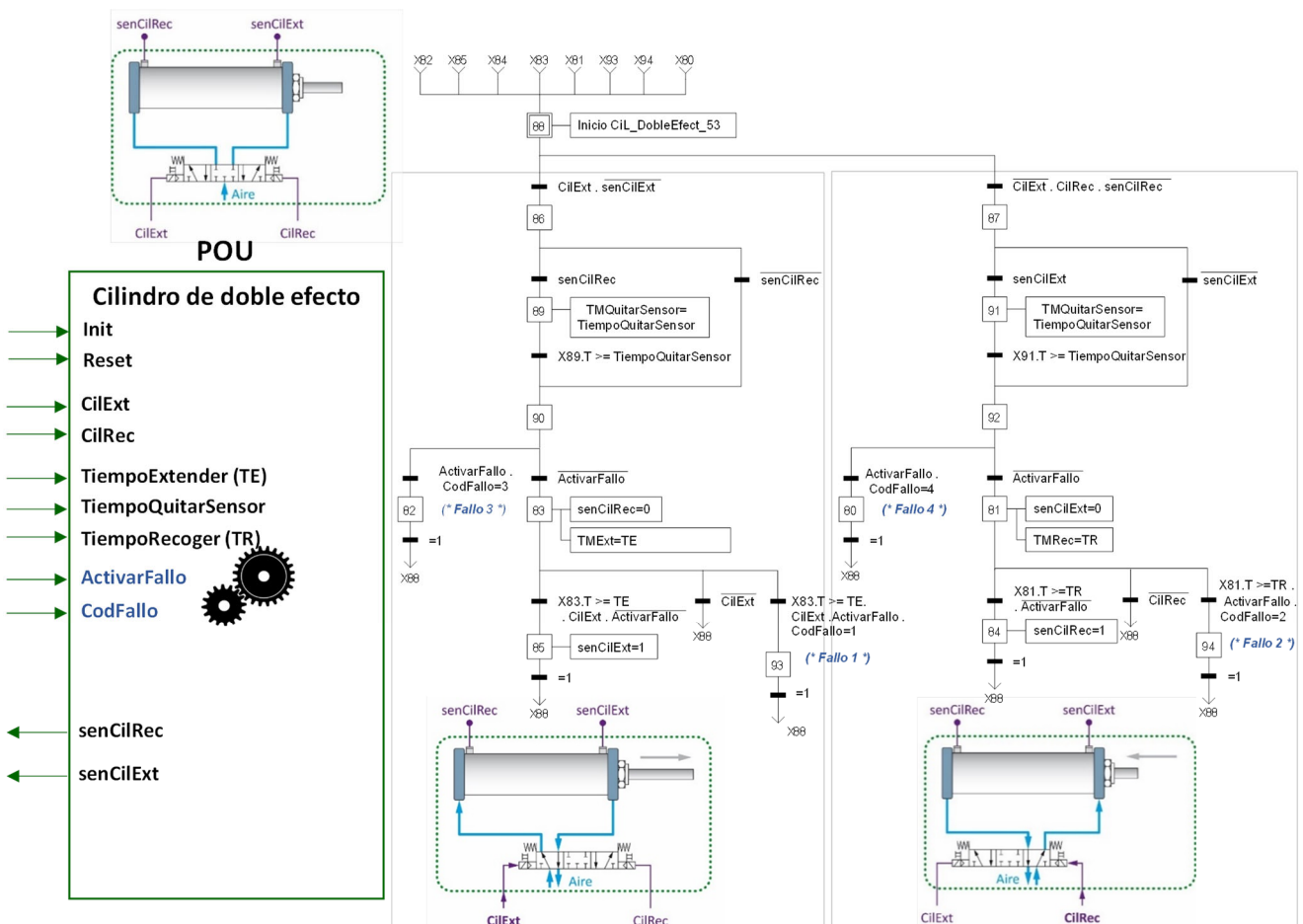


Figura 6: Graficet del diseño detallado de la funcionalidad con fallos del módulo de control de un cilindro de doble efecto.

informan de la situación de la entidad controlada. En el caso del GF del componente MeIA se puede utilizar la misma interface de bloque del componente en el que se suprimen las entradas y salidas de la unidad de trabajo.

En el *Cuerpo de Bloque* se instancian los FB de los CFB identificados, se conectan con las entradas y salidas del GF y se asignan los tiempos de simulación acorde con el análisis dinámico de cada CFB.

4. Modificación de las variables de entradas (%I) y salidas (%Q) a marcas de memoria (%M), permitiendo que las actualice el GF sin ser modificadas en cada ciclo de programa. Como alternativa, se pueden crear nuevas variables de memoria para las entradas y salidas, con objeto de preparar el software de control para conmutar entre el funcionamiento con el GF o con la entidad controlada real. En ambos casos, el GF dispone de una señal de activación.
5. Vinculación del GF con el software de control. Una vez realizada la instancia del FB del GF, se realiza la conexión con el software de control. Las entradas del GF se conectan con las salidas generadas por el software de control para actuar sobre la entidad controlada virtual. Las variables de entrada/salida se conectan con las entradas simuladas del software de control.
6. Verificación de todas las acciones de la unidad de trabajo/operaciones del componente MeIA y ajuste final de los tiempos de proceso. Para ello, cada una de las acciones/operaciones se activa manualmente y se analiza la coherencia de la respuesta. Posteriormente, el software de control coordinará las acciones de los distintos CFB que forman el GF de la misma forma que coordina las acciones de los módulos de equipo y módulos de control de la unidad de trabajo del aPS, o las operaciones de los componentes MeIA en un nivel jerárquico superior.

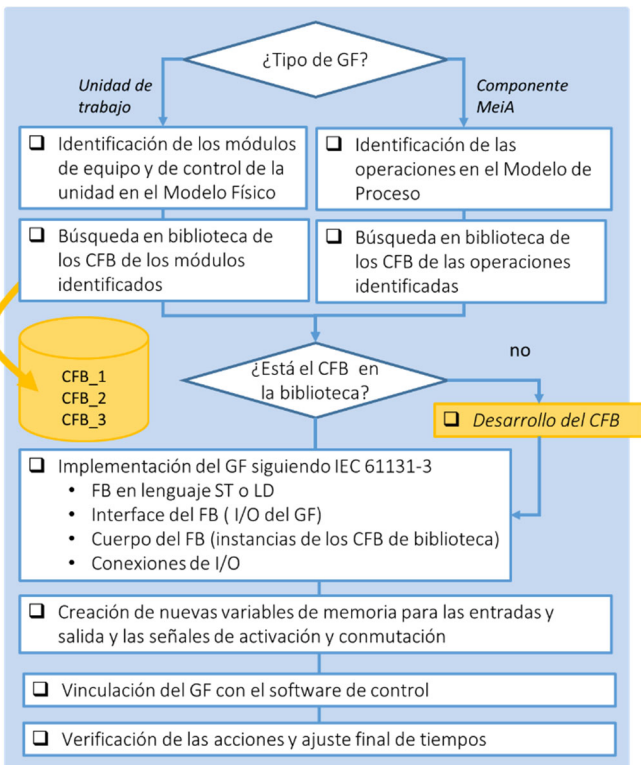


Figura 7: Pasos para el desarrollo de un GF.

5. Caso de estudio

El caso de estudio aborda el desarrollo de un GF para realizar las pruebas del software de control de una de las unidades de trabajo de la célula de ensamblaje DISA FMS200, así como el desarrollo de un GF del componente MeIA que controla dicha unidad de trabajo para realizar las pruebas del software de control de la célula de ensamblaje, tal y como se muestra en la Figura 8.

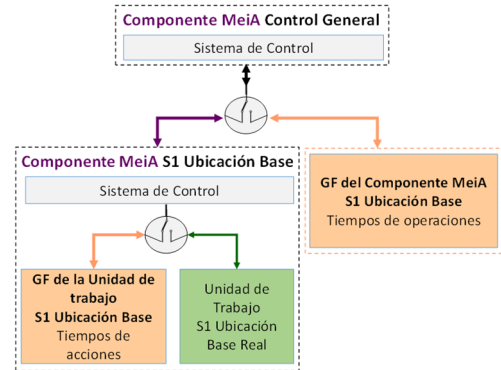


Figura 8: Los GF para la unidad de trabajo SI – Ubicación Base

Concretamente, la unidad de trabajo *SI Ubicación Base* extrae una base de un almacén vertical y comprueba si su posición es correcta. En caso afirmativo, la coloca sobre el palet situado en la posición de carga. En caso contrario, la rechaza por una rampa de evacuación.

El desarrollo del GF de la unidad de trabajo del aPS, empleado en las pruebas del componente MeIA a cargo de su control, se realiza según los siguientes pasos:

1. Identificación de los módulos de equipo o de control de la unidad de trabajo en el Modelo Físico del aPS.

En la Figura 9 se presenta el Modelo Físico de la unidad de trabajo *SI Ubicación Base* que consta de dos módulos de equipo: Manipulador y Alimentador de Bases. El Manipulador está constituido por los siguientes módulos de control: un cilindro de simple efecto, un cilindro de doble efecto (el utilizado como ejemplo para desarrollar el CFB) y una ventosa. El Alimentador de Bases está formado por cuatro cilindros de simple efecto y un almacén por gravedad.

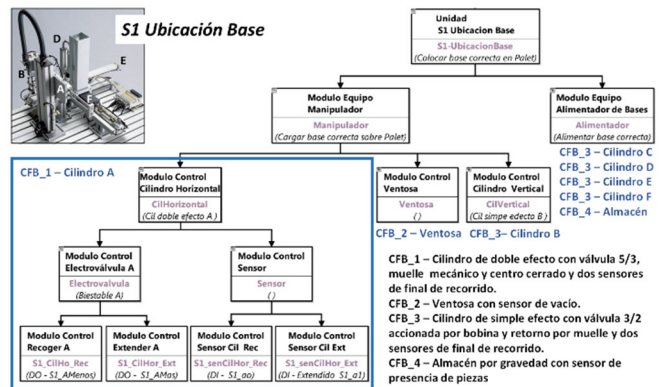


Figura 9: Modelo Físico de la unidad de trabajo SI – Ubicación Base.

2. Búsqueda de los CFB en la biblioteca de componentes.

Los CBF asociados a los módulos de control identificados se encuentran en la biblioteca de componentes.

Concretamente, serán necesarios cuatro tipos de CFB, tal y como se muestra en la Figura 10.



Figura 10: CFB seleccionados de la biblioteca de componentes.

3. Implementación del GF según el estándar IEC 61131-3.

Tras la creación del FB y la definición de la interfaz de bloque, en el cuerpo del FB se instancian los CFB identificados, se conectan sus entradas y salidas, y se ajustan sus tiempos. Este desarrollo se ha realizado en Tia Portal (Figura 11).

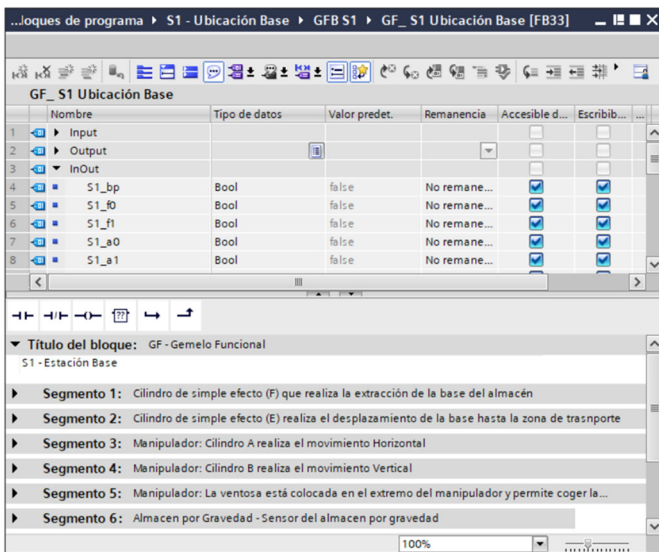


Figura 11: Implementación del FB del GF en Tia Portal.

4. Creación de nuevas variables de memoria para las entradas y salidas, y las señales de activación y conmutación necesarias.

En la Figura 12 se muestran los conmutadores para la activación del GF y la selección del modo de funcionamiento.

5. Vinculación del GF con el software de control.

En la misma Figura 12 se puede apreciar el GF de la unidad de trabajo en funcionamiento.

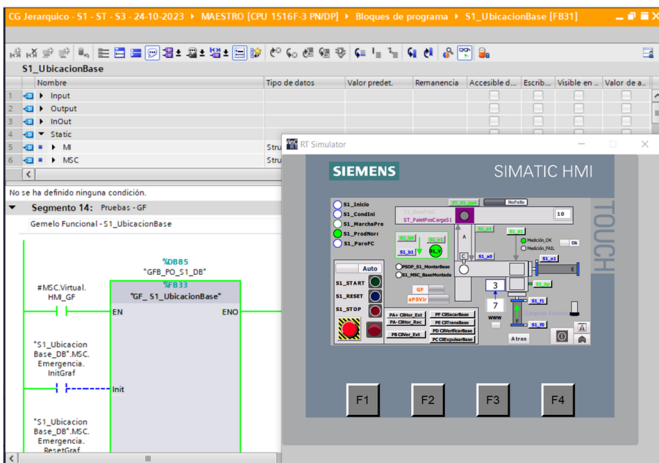


Figura 12: GF de la unidad de trabajo S1 Ubicacion Base en ejecución.

6. Verificación de las acciones y ajuste final de los tiempos de proceso.

Para la verificación de las acciones, los tiempos se aumentan de forma proporcional a los tiempos reales para ralentizar la ejecución y facilitar el seguimiento de las pruebas. Así, para el ajuste final ha sido necesario tener en cuenta las presiones de las válvulas de los módulos de control del sistema físico.

El GF de la unidad de trabajo S1 Ubicacion Base ha sido empleado en la realización de las pruebas unitarias, de integración y verificación & validación del sistema de control de dicha estación. Concretamente, se han validado los modos de operación: automático, manual, fallos y emergencia, representados en la Figura 13.

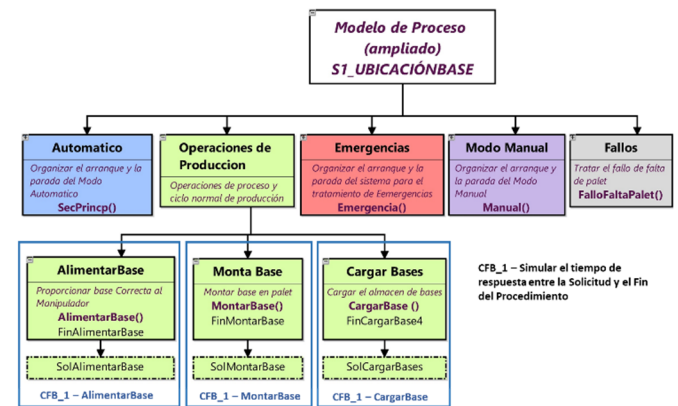


Figura 13: Modelo de Proceso extendido con procedimientos y señales de activación y desactivación

Por otra parte, el desarrollo del GF del componente MeIA que controla la unidad de trabajo S1 Ubicacion Base empleado en las pruebas del control general de la célula de ensamblado, se realiza según los siguientes pasos:

1. Identificación de las operaciones de decisión, producción y auxiliares.

En la Figura 13 se presenta el Modelo de Proceso extendido con los procedimientos asociados a las operaciones para poder identificar los CFB.

2. Búsqueda de los CFB en la biblioteca de componentes.

Los CBF asociados a los módulos de control identificados se encuentran en la biblioteca de componentes de operaciones, tal y como se observa en la Figura 14.



Figura 14: Los CFB de las operaciones para el componente MeIA.

3. Implementación del GF según el estándar IEC 61131-3.

Tras la creación del FB y la definición de la interfaz de bloque, en el cuerpo del FB se instancian los CFB identificados, se conectan sus entradas y salidas, y se ajustan sus tiempos.

4. Creación de las señales de activación y conmutación necesarias.

En la Figura 13 se observa que las señales de activación se corresponden con las señales de solicitud de operación y las señales de desactivación se corresponden con las señales de finalización de la operación. Además de estas señales se deben añadir las señales de conmutación entre el componente MeiA y su GF.

5. Vinculación del GF con el software de control.

La vinculación se realiza mediante la comunicación entre el GF del componente MeiA de la estación y el componente MeiA del control general, tal y como se puede apreciar en la Figura 8.

6. Verificación de las operaciones del componente MeiA y ajuste final de los tiempos de proceso.

En este caso, los tiempos de las operaciones se disminuyen de forma proporcional a los tiempos reales para agilizar la ejecución y el seguimiento de las pruebas.

Este GF junto con los correspondientes a las otras cuatro unidades de trabajo (S2 Robot-Eje/Rodamiento, S3 Ubicación Tapa, S4 Almacén y ST Sistema Transporte), que forman la célula de ensamblaje DISA FMS200, son necesarios para realizar la integración y las pruebas del componente MeiA de control general de la célula. Concretamente, se han validado los modos de operación automático y emergencia de dicha célula.

La integración de los componentes MeiA de las unidades de trabajo del aPS se lleva a cabo de forma escalonada mediante el procedimiento de pruebas que se realiza en el componente MeiA del control general, de acuerdo a los siguientes pasos: (1) probar el componente MeiA del control general incluyendo los GF de los componentes MeiA de las unidades de trabajo del aPS de forma escalonada; (2) probar el componente MeiA del control general sustituyendo, de forma escalonada, los GF de los componentes MeiA por los componentes MeiA que trabajan con los GF de las unidades de trabajo del aPS; y (3) probar el componente MeiA del control general conmutando progresivamente los componentes MeiA de las unidades de trabajo del aPS para que funcionen con la unidad de trabajo real.

6. Conclusiones

Este trabajo propone una metodología de desarrollo de modelos digitales como herramienta para realizar las pruebas a los sistemas de control que incluyen buenas prácticas de IS y de automatización. Estos GF se construyen combinando CFB reutilizables de biblioteca de acuerdo a los modelos propuestos en el estándar IEC 61512 (Modelo Físico o Modelo de Proceso/Modelo de Control Procedural según sean GF/CFB de unidades de trabajo del aPS o GF/CFB de componentes MeiA).

Estas herramientas permiten automatizar las pruebas, reduciendo el tiempo dedicado a su realización y/o incrementando la exhaustividad de las mismas. También disminuyen la probabilidad de error humano al ejecutarlas manualmente con herramientas convencionales que dependen, además, de la destreza del personal encargado. De este modo, se evita que el personal requiera un profundo conocimiento del comportamiento del aPS para activar y desactivar los sensores en función de las acciones realizadas por los actuadores, lo cual

se complica cuando el proceso requiere operaciones en paralelo.

El enfoque de desarrollo a partir de los CFB de biblioteca aporta reusabilidad, flexibilidad y versatilidad al permitir distintas posibilidades de composición, reduciendo considerablemente los costes de desarrollo, el tiempo y el nivel de conocimientos del sistema mecatrónico. Dado que estos CFB pueden incluir inyección de fallos, permiten analizar la reactividad del sistema antes eventos imprevistos sin poner en peligro los sistemas físicos.

Otro aspecto a destacar es que no se precisan herramientas comerciales de desarrollo de plantas virtuales para su construcción. Estos modelos digitales se implementan conforme al estándar IEC 61131-3 ampliamente aceptado en el área de automatización, favoreciendo la reutilización y portabilidad.

Agradecimientos

Este trabajo está financiado por UPV/EHU (proyecto PES18/48 y proyecto PES 22/77).

Referencias

- Adamenko, D., Kunnen, S., Pluhna, R., Loibl, A., Nagarajah, A., 2020. Review and comparison of the methods of designing the Digital Twin. *Procedia CIRP*, 91, 27-32. DOI: 10.1016/j.procir.2020.02.146
- ADEPA (Agence nationale pour le Développement de la Production Automatisée), 1981. *GEMMA (Guide d'Étude des Modes de Marches et d'Arrêts)*.
- Aivaliotis, P., Georgoulas, K., Arkouli, Z., Makris, S., 2019. Methodology for enabling Digital Twin using advanced physics-based modelling in predictive maintenance. *Procedia CIRP*, 81, 417-422. DOI: 10.1016/j.procir.2019.03.072
- Álvarez, M.L., Estévez, E., Sarachaga, I., Burgos, A., Marcos, M., 2013. A novel approach for supporting the development cycle of automation systems. *International Journal of Advanced Manufacturing Technology*, 68, 711-725. DOI: 10.1007/s00170-013-4793-4
- Álvarez, M.L., Sarachaga, I., Burgos, A., Iriondo, N., 2023. Gemelo funcional para pruebas del software de sistemas de producción automatizados. *XLIV Jornadas de Automática*, 831-836. DOI: 10.17979/spudc.9788497498609.831
- AutomationML, 2023. [Website] <https://www.automationml.org/>
- Barbieri, G., Bertuzzi, A., Capriotti, A., Ragazzini, L., Gutierrez, D., Negri, E., Fumagalli, L., 2021. A virtual commissioning based methodology to integrate digital twins into manufacturing systems. *Production Engineering*, 15, 397-412. DOI: 10.1007/s11740-021-01037-3
- Burgos, A., Iriondo, N., Álvarez, M.L., Sarachaga, I., 2021. MeiA 4.0 para abordar los retos actuales de formación en automatización. *XLII Jornadas de Automática*, 1-3 Sep., Castellón, España, 240-247. DOI: 10.17979/spudc.9788497498043.240
- Burgos, Álvarez, M.L., A., Iriondo, N., Sarachaga, I., 2020. Metodología para la transformación de diseños en GRAFCET a código IEC 61131-3. *Información Tecnológica*, 31(6). DOI: 10.4067/S0718-07642020000600133
- Cimino, Ch., Negri, E., Fumagalli, L., 2019. Review of digital twin applications in manufacturing. *Computers in Industry*, 113. DOI: 10.1016/j.compind.2019.103130
- Chen, Ch., Fu, H., Zheng, Y., Tao, F., Liu, Y., 2023. The advance of digital twin for predictive maintenance: The role and function of machine learning. *Journal of Manufacturing Systems*, 71, 581-594. DOI: 10.1016/j.jmsy.2023.10.010
- Cheng, J., Zhang, H., Tao, F., Juang, Ch.F., 2020. DT-II: Digital twin enhanced Industrial Internet reference framework towards smart manufacturing. *Robotics and Computer-Integrated Manufacturing*, 62, 101881. DOI: 10.1016/j.rcim.2019.101881
- de Prada, C., Galán-Casado, S., Pitarch, J. L., Sarabia, D., Galán, A., Gutiérrez, G., 2022. Gemelos Digitales en la Industria de Procesos. *Revista*

- Iberoamericana de Automática e Informática Industrial, 19(3), 285-296. DOI: 10.4995/riai.2022.16901
- Farhadi, A., Lee, S.K.H., Hinchy, E.P., O'Dowd, N.P., McCarthy, C.T., 2022. The development of a digital twin framework for an industrial robotic drilling process. *Sensors* 22(19), 7232. DOI: 10.3390/s22197232
- Ferko, E., Bucaioni, A., Behnam, M., 2022. Architecting Digital Twins. *IEEE Access*, 10, 50335-50350. DOI: 10.1109/ACCESS.2022.3172964
- Fuller, A., Fan, Z., Day, C., Barlow, C., 2020. Digital Twin: Enabling Technologies, Challenges and Open Research. *IEEE Access*, 8, 108952-108971. DOI: 10.1109/ACCESS.2020.2998358
- Ganser, P., Venek, T., Rudel, V., Bergs, T., 2021. DPART - a digital twin framework for the machining domain. *MM Science Journal*, 5134-5141. DOI: 10.17973/MMSJ.2021_11_2021168
- Hung, M.H., Lin, Y.C., Hsiao, H.C., Chen, C.C., Lai, K.C., Hsieh, Y.M., Tieng, H., Tsai, T.H., Huang, H.C., Yang, H.C., Cheng, F.T., 2022. A novel implementation framework of digital twins for intelligent manufacturing based on container technology and cloud manufacturing services. *IEEE Transactions on Automation Science and Engineering*, 19(3), 1614-30. DOI: 10.1109/TASE.2022.3143832
- ISO 23247-2, 2021. Automation systems and integration — Digital twin framework for manufacturing — Part 2: Reference architecture. International Organization for Standardization. <https://www.iso.org/standard/78743.html>
- ISO/IEC/IEEE 12207, 2017. Systems and software engineering – Software life cycle processes. International Organization for Standardisation.
- IEC PAS 63088, 2017. Smart manufacturing – Reference architecture model industry 4.0 (RAMI 4.0).
- IEC 61131-3, 2013. IEC 61131-3, Programmable Controllers, Part 3: Programming Languages.
- IEC 61512-1, 2002. Batch Control-Part 1: Models and terminology.
- Iriondo, N., Orive, D., Casquero, O., Marcos, M., 2020. A proposal to introduce digitalization technologies within the automation learning process. *IFAC-PapersOnLine*, 53(2), 17592-17597. DOI: 10.1016/j.ifacol.2020.12.2674
- Iriondo, N., Álvarez, M.L., Sarachaga, I., Burgos, A., 2022. Unidades de control encapsuladas para sistemas de automatización. *XLIII Jornadas de Automática*, 7-9 Sep., Logroño, España, 892-899. DOI: 10.17979/spudc.9788497498418.0892
- Jones, D., Snider, Ch., Nassehi, A., Yon, J., Hicks, B., 2020. Characterising the Digital Twin: A systematic literature review. *CIRP Journal of Manufacturing Science and Technology*, 29, 36-52. DOI: 10.1016/j.cirpj.2020.02.002
- Konstantinov, S., Assad, F., Ahmad, B., Vera, D.A., Harrison, R., 2022. Virtual Engineering and Commissioning to Support the Lifecycle of a Manufacturing Assembly System. *Machines*, 10, 939. DOI: 10.3390/machines10100939
- Kritzinger, W., Karner, M., Traar, G., Henjes, J., Sihn, W., 2018. Digital Twin in manufacturing: A categorical literature review. *IFAC-PapersOnLine*, 51(11), 1016-1022. DOI: 10.1016/j.ifacol.2018.08.474
- Liu, S., Bao, J., Zheng, P., 2023. A review of digital twin-driven machining: From digitization to intellectualization. *Journal of Manufacturing Systems*, 67, 361-378. DOI: 10.1016/j.jmsy.2023.02.010
- Mihai, S., Yaqoob, M., Hung, D.V., Davis, W., Towakel, P., Raza, M., Karamanoglu, M., Barn, B., Shetve, D., Prasad, R.V., Venkataraman, H., Trestian, R., Nguyen, H.X., 2022. Digital Twins: A Survey on Enabling Technologies, Challenges, Trends and Future Prospects. *IEEE Communications Surveys & Tutorials*, 24(4), 2255-2291. DOI: 10.1109/COMST.2022.3208773
- Montenegro-Bravo, J. S., Ruiz-Flórez, J. D., Romero-Ante, J. D., Manrique-Córdoba, J., Vivas Albán, O. A., Sabater-Navarro, J. M., 2023. Generador 3D de trayectorias libres de colisiones para un manipulador UR3e con pinza blanda. *Revista Iberoamericana de Automática e Informática Industrial*. DOI: 10.4995/riai.2023.19332
- Moyné, J., Qamsane, Y., Balta, E. C., Kovalenko, I., Faris, J., Barton, K., Tilbury, D. M., 2020. A requirements driven digital twin framework: Specification and opportunities. *IEEE Access*, 8, 107781-107801. DOI: 10.1109/ACCESS.2020.3000437
- Mylonas, G., Kalogeras, A., Kalogeras, G., Anagnostopoulos, C., Alexakos, C., Muñoz, L., 2021. Digital Twins From Smart Manufacturing to Smart Cities: A Survey. *IEEE Access*, 9, 143222-143249. DOI: 10.1109/ACCESS.2021.3120843
- Negri, E., Assiro, G., Caioli, L., Fumagalli, L., 2019. A machine state-based digital twin development methodology. In *Summer School F. Turco-Industrial Systems Engineering*, 34-40.
- Plakhotnik, D., Curutiu, A., Zhulavskiy, A., Beudaert, X., Munoa, J., Stautner, M., 2021. Framework for coupled digital twins in digital machining. *MM Science Journal*, 5093-7. DOI: 10.17973/MMSJ.2021_11_2021158
- Shao, G., Helu, M., 2020. Framework for a digital twin in manufacturing: Scope and requirements." *Manufacturing Letters*, 24, 105-107. DOI: 10.1016/j.mfglet.2020.04.004
- Semeraro, C., Lezoche, M., Panetto, H., Dassisti, M., 2021. Digital twin paradigm: A systematic literature review. *Computers in Industry*, 130, 103469. DOI: 10.1016/j.compind.2021.103469
- Semeraro C, Lezoche M, Panetto H, Dassisti M., 2023a. Data-driven invariant modelling patterns for digital twin design. *Journal of Industrial Information Integration*, 31, 100424. DOI: 10.1016/j.jii.2022.100424
- Semeraro, C., Aljaghoub, H., Abdelkareem, M. A., Alami, A. H., Dassisti, M., Olabi, A.G., 2023b. Guidelines for designing a digital twin for Li-ion battery: A reference methodology. *Energy*, 284, 128699. DOI: 10.1016/j.energy.2023.128699
- Orive, D., Iriondo, N., Burgos, A., Sarachaga, I., Alvarez, M.L., Marcos, M., 2019. Fault injection in Digital Twins as a means to test the response to process faults at virtual commissioning. *24th IEEE Conference on Emerging Technologies and Factory Automation (ETFA)*, 1230-1234. DOI: 10.1109/ETFA.2019.8869334
- Prados, C., Hernando, M., Gambao, E., Brunete, A., 2022. ROMERIN: Organismo robótico escalador basado en patas modulares con ventosas activas. *Revista Iberoamericana de Automática e Informática Industrial*, 20(2), 175-186. DOI: 10.4995/riai.2022.18749
- Qamsane, Y., Moyné, J., Toothman, M., Kovalenko, I., Balta, E.C., Faris, J., Tilbury, D.M., Barton, K., 2021. A Methodology to Develop and Implement Digital Twin Solutions for Manufacturing Systems. *IEEE Access*, 9, 44247-44265. DOI: 10.1109/ACCESS.2021.3065971
- Qi, Q., Tao, F., Hu, T., Anwer, N., Liu, A., Wei, Y., Wang, L., Nee, A.Y.C., 2021. Enabling technologies and tools for digital twin. *Journal of Manufacturing Systems*, 58, 3-21. DOI: 10.1016/j.jmsy.2019.10.001
- Rasheed, A., San, O., Kvamsdal, T., 2020. Digital Twin: Values, Challenges and Enablers From a Modeling Perspective. *IEEE Access*, 8, 21980-22012. DOI: 10.1109/ACCESS.2020.2970143
- Schroeder, G.N., Steinmetz, C., Rodrigues, R.N., Henriques, R.V.B., Rettberg, A., Pereira, C.E., 2021. A Methodology for Digital Twin Modeling and Deployment for Industry 4.0. *Proceedings of the IEEE*, 109(4), 556-567. DOI: 10.1109/JPROC.2020.3032444
- Shao, G., Frechette, S., Srinivasan, V., 2023. An Analysis of the New ISO 23247 Series of Standards on Digital Twin Framework for Manufacturing. *Manufacturing Science and Engineering Conference*, 12-16 Junio, New Brunswick, USA. DOI: 10.1115/MSEC2023-101127
- Tao, F., Zhang, H., Liu, A., Nee, A. Y. C., 2019. Digital Twin in Industry: State-of-the-Art. *IEEE Transactions on Industrial Informatics*, 15(4), 2405-2415. DOI: 10.1109/TII.2018.2873186
- Wang, P., Luo, M., 2021. A digital twin-based big data virtual and real fusion learning reference framework supported by industrial internet towards smart manufacturing. *Journal of Manufacturing Systems*, 58, 16-32. DOI: 10.1016/j.jmsy.2020.11.012
- Zhang, Y. F., Shao, Y. Q., Wang, J. F., Li, S. Q., 2020. Digital Twin-based Production Simulation of Discrete Manufacturing Shop-floor for Onsite Performance Analysis. *IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, Singapore, 1107-1111. DOI: 10.1109/IEEM45057.2020.9309928
- Zhou, X., Xu, X., Liang, W., Zeng, Z., Shimizu, S., Yang, L.T., Jin, Q., 2022. Intelligent Small Object Detection for Digital Twin in Smart Manufacturing with Industrial Cyber-Physical Systems. *IEEE Transactions on Industrial Informatics*, 18(2), 1377-1386. DOI: 10.1109/TII.2021.3061419