

Arquitecturas de Control a Tiempo Real de un Gemelo Virtual

Heber Hernández-Vázquez¹, Irma Y. Sánchez^{2,3}, Fernando Martell⁴, José E. Guzmán⁵, Raúl A. Ortiz⁶

¹ Universidad Politécnica de Aguascalientes, Dirección de Posgrado e Investigación, Calle Paseo San Gerardo No. 207, Fracc. San Gerardo C.P.20342 Aguascalientes, Ags., México, mc190007@alumnos.upa.edu.mx

² Universidad Politécnica de Aguascalientes, Dirección de Posgrado e Investigación, Calle Paseo San Gerardo No. 207, Fracc. San Gerardo C.P.20342 Aguascalientes, Ags., México, irma.sanchez@upa.edu.mx

³ Ingeniería Mecatrónica, S.A. de C.V., Investigación y Desarrollo, Córdoba No. 204, Fracc. El Dorado C.P.20235 Aguascalientes, Ags., México, isanchez@ingmt.com

⁴ Centro de Investigaciones en Óptica, Departamento de Ingeniería, Calle Prol. Constitución 607, Fraccionamiento Reserva Loma Bonita C.P.20200 Aguascalientes, Ags., México, fmartell@cio.mx

⁵ Universidad Politécnica de Aguascalientes, Ingeniería en Energía, Calle Paseo San Gerardo No. 207, Fracc. San Gerardo C.P.20342 Aguascalientes, Ags., México, jose.guzman@upa.edu.mx

⁶ Universidad Politécnica de Aguascalientes, Ingeniería en Energía, Calle Paseo San Gerardo No. 207, Fracc. San Gerardo C.P.20342 Aguascalientes, Ags., México, raul.ortiz@upa.edu.mx

Resumen

El gemelo virtual expedita el diseño de estrategias de control por permitir la interacción con un modelo de la máquina real correspondiente fuera de línea, lo cual permite tanto el manejo de tiempo de simulación para pruebas rápidas, como la operación en tiempo real prescindiendo de la máquina física. La operación a tiempo real del gemelo virtual facilita la incorporación del equipo de control con la implementación de los algoritmos a utilizar para automatizar y operar el sistema ciberfísico en una fase de desarrollo posterior. En este trabajo se exponen alternativas para el funcionamiento del gemelo virtual en un esquema de simulación hardware-in-the-loop de un escáner tridimensional (3D). Se plantean arquitecturas que permiten el uso de dos plataformas de control básicas en la automatización industrial, microprocesador y controlador lógico programable (PLC) interconectados directamente con el gemelo virtual. Esta investigación tecnológica contribuye con lineamientos para resolver la conectividad que posibilita la implementación de dichas arquitecturas utilizando protocolos Modbus. Se plantea la continuación de este trabajo en un esquema de hardware en cascada o serie para comunicarse con el gemelo virtual en una automatización multinivel y potencialmente distribuida, así como la integración del sistema ciberfísico para el escáner 3D.

Palabras clave— Virtual Twin, Modbus TCP/IP, Modbus RTU, Hardware-in-the-loop.

I. INTRODUCCIÓN

LOS SISTEMAS ciberfísicos constituyen una de las tecnologías potencializadoras de la industria 4.0 o también llamada Industria Inteligente, es la cuarta revolución industrial que viene a cambiar el estilo de fabricación y gestión de procesos industriales implementando nuevas tecnologías innovadoras que permitan alcanzar altos márgenes de calidad y producción, basándose en una visualización del proceso en operación, con acceso a la información del mismo.

La Industria 4.0 se caracteriza por un mayor nivel de interacción con los procesos, lo que permite el acceso a la información y el funcionamiento de los equipos de la planta desde ubicaciones remotas, bajo el objetivo de la fabricación integrada [1]. Los sistemas ciberfísicos constituyen un pilar tecnológico para lograr los principios de diseño y funcionamiento de la industria 4.0. El sistema ciberfísico integra los sistemas físicos con sus correspondientes gemelos digitales o virtuales [2] (Fig.1). El gemelo digital de una máquina física es un elemento esencial de un sistema de producción ciberfísico para mejorar la eficiencia y la seguridad del proceso y la operación, incluso en los procedimientos de configuración o puesta en servicio [3], [4].

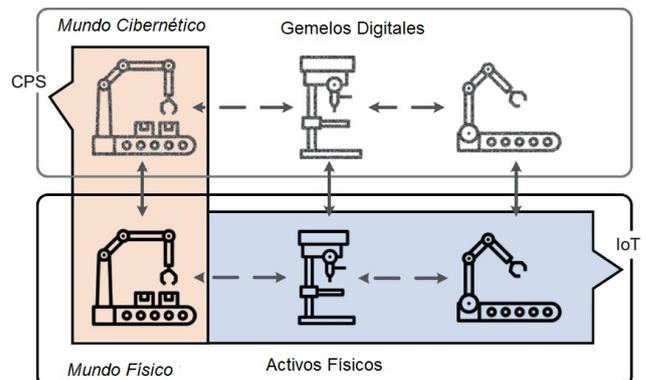


Fig. 1. Sistema Ciberfísico que integra máquinas físicas con sus correspondientes gemelos digitales. CPS: Cyber-Physical System. IoT: Internet of Things. Adaptado de [2].

Los sistemas ciberfísicos pueden entenderse como sistemas mecatrónicos con funciones avanzadas de automatización y que cumplen los criterios de diseño de la industria 4.0, de interoperabilidad y transparencia de la información. Los gemelos digitales son sistemas mecatrónicos virtuales que reproducen el comportamiento del sistema físico, tanto sus aspectos mecánicos y estructurales que son representados

mediante modelación 3D, como la respuesta dinámica del propio mecanismo, de los sensores y actuadores que son reproducibles mediante simulaciones numéricas computacionales. Los sistemas mecatrónicos virtuales se utilizan cada vez más para obtener un mejor desempeño de los sistemas físicos en estrategias avanzadas de control [5]-[7], y también con fines educativos, como la formación a distancia en laboratorios virtuales [8], [9].

En la conformación general del sistema ciberfísico se integra además un sistema de control, comunicación bidireccional y la máquina real como su gemelo digital, pueden resaltarse dos componentes que pueden aprovecharse en el diseño y pruebas de estrategias de control fuera de línea, es decir, sin requerir el paro de la máquina: el gemelo digital y el controlador.

Las metodologías propuestas para el desarrollo de los sistemas ciberfísicos contemplan el diseño de las comunicaciones como un paso importante, por el motivo que es el medio en donde la máquina real y su gemelo virtual interactúan en tiempo real. La simulación en tiempo real que requiere el sistema ciberfísico es un reto que se soluciona en este trabajo mediante el uso del software de modelado 3D y animación llamado Blender®.

En distintas aplicaciones, se ha solucionado el problema de conectividad utilizando distintos protocolos de comunicación como lo son Modbus TCP (Protocolo de Control de Transmisión, por sus siglas en inglés) y Modbus RTU (Unidad Terminal Remota, por sus siglas en inglés). Destaca el protocolo Modbus por su gran versatilidad en la industria haciéndolo una herramienta eficaz y segura, y además, es un protocolo de comunicación de código abierto para el usuario que requiera utilizarlo.

En el diseño mecatrónico, se emplean distintos enfoques de simulación que permiten incorporar el elemento controlador en forma preliminar a su puesta en marcha para la operación automática del proceso real antes de ser implementados, con el fin de obtener una base de datos de funcionamiento que permitan tener una noción clara de posibles problemas al estar sometido a ciertas condiciones de trabajo. El enfoque “hardware-in-the-loop” (HiL) es inherente al funcionamiento del gemelo virtual en el sistema ciberfísico, y constituye una gran parte de su desarrollo. La comunicación del hardware del controlador con el gemelo virtual debe resolverse para lograr la integración con el gemelo real. El presente trabajo aborda esta cuestión para el gemelo virtual de un escáner 3D, considerando dos plataformas básicas de control representadas por el microprocesador Arduino® y la plataforma CoDeSys® para programación de controladores programables.

El artículo consta de las siguientes secciones: la Sección II describe las entidades gemelas virtual y física del router o escáner 3D, los recursos de software y hardware, y protocolos Modbus utilizados; la Sección III plantea las arquitecturas para la operación a tiempo real del gemelo digital con Arduino® y CoDeSys®, la configuración correspondiente de las comunicaciones, así como el desempeño observado; en la Sección IV se proponen extensiones de las arquitecturas presentadas para esquemas avanzados de control; finalmente,

en la Sección V se concluye sobre las ventajas y desventajas de las configuraciones exploradas y las propuestas en base a la combinación de las primeras.

II. RECURSOS Y MÉTODOS

El gemelo virtual o simulador de un router real interactúa con una plataforma de automatización. En este trabajo se maneja la opción de un Controlador Lógico Programable (PLC, por sus siglas en inglés) a través de la plataforma de desarrollo CoDeSys®, al que podemos referirnos como un SoftPLC (PLC emulado en software). CoDeSys® se ha vuelto popular porque implementa de manera versátil el estándar IEC-61131-3 para la programación de PLCs, y actualmente se utiliza para la programación de distintas marcas comerciales de PLC. Como alternativa, se propone la comunicación con un microcontrolador Arduino®. El alcance de este artículo es la definición de un sistema que implementa una simulación en tiempo real del tipo HiL como se muestra en la Fig. 2. Este sistema es previo a la implementación de la interacción simultánea con el router físico y con el router virtual.

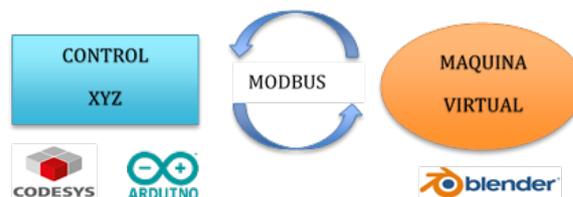


Fig. 2. Diagrama conceptual de la arquitectura básica de control a tiempo real de un gemelo virtual.

A. Gemelos Físico y Virtual, Controladores, Protocolo

La máquina física es un router comercial de tres ejes modelo CNC3 3018 PRO (Fig. 3). Las dimensiones de la máquina son 330 x 400 x 240 mm (13.0" × 15.7" × 9.4") y tiene un área de trabajo efectiva de 300 x 180 x 45 mm (11.8" × 7.1" × 1.8"). La Tabla I muestra la lista de partes del CNC3 3018 PRO.

TABLA I
LISTA DE PARTES DELCNC3 3018 PRO

Nombre	Tipo	Cantidad
Extrusión de aluminio	20×40×290 mm	2
	20×20×360 mm	2
	15×180×300mm	1
Guías lineales	10×290 mm	2
	10×360 mm	2
Tornillos T8-4	T8*4, 295 mm	1
	T8*4, 365 mm	1
Baquelita	Eje Y	2
	Eje X	2
Partes XZ		1
Tarjeta electrónica	GRBL 1.1	1
Huso	775-ER11	1
Motor de pasos	42×34 mm	4
Deslizador eje Y	10 mm	4
Soporte eje Y	T8*4	1



Fig. 3. Sistema físico del router CNC3 3018 PRO.

La máquina virtual o gemelo digital del router es un desarrollo en Blender® v.2.79 [10], software gratuito de código abierto en lenguaje Python. El gemelo virtual consiste en gráficos 3D animados que muestran con precisión suficiente la estructura y los actuadores la máquina física. La animación refleja la dinámica de movimiento sobre los tres ejes. El gemelo virtual, a semejanza del físico, recibe como entrada la señal del controlador o manipulación sobre cada actuador y produce como salida la posición alcanzada en las tres coordenadas correspondientes a los tres ejes de movimiento.

Las plataformas de control son CoDeSys® y Arduino®. CoDeSys® permite el desarrollo de programas de PLC según el estándar IEC-61131-3, ya sea en SoftPLC o en PLCs de diferentes fabricantes. Arduino® es una plataforma de desarrollo basada en una tarjeta electrónica de hardware libre con un microcontrolador y puertos de entrada y salida. Tanto en el caso del SoftPLC como del Arduino, las entradas a manejar consisten en los desplazamientos en tres direcciones de la herramienta del router virtual, y las salidas son las señales a enviar a los tres actuadores correspondientes.

Se utiliza el protocolo de comunicación Modbus en sus versiones para puerto serial o Modbus RTU, y sobre Ethernet TCP/IP (Protocolo de Control de Transmisión/Protocolo de Internet, por sus siglas en inglés), denominado Modbus TCP. El protocolo de comunicación entre el gemelo virtual y CoDeSys® es Modbus TCP, y la comunicación con Arduino® es por Modbus TCP y Modbus RTU. El protocolo Modbus es público y de amplio uso y conocimiento industrial.

B. Métodos de Implementación de Arquitecturas

El procedimiento para implementar la comunicación bidireccional entre el gemelo virtual y la plataforma CODESYS®, mediante Modbus TCP, consistió en

1. Configurar gemelo virtual en plataforma Blender® como cliente.
2. Configurar plataforma CoDeSys® como servidor.
3. Realizar pruebas funcionales de la comunicación.

Para establecer la comunicación del gemelo virtual con el microcontrolador Arduino®, se utilizó una librería que implementa el cliente o maestro, según el tipo de protocolo Modbus, en el gemelo virtual, por lo que se siguieron los siguientes pasos:

1. Configurar el gemelo virtual ya sea como cliente de Modbus TCP o maestro de Modbus RTU.
2. Configurar Arduino® como servidor o esclavo de la comunicación Modbus TCP o RTU, respectivamente.
3. Realizar pruebas funcionales de la comunicación.

Las arquitecturas básicas que se proponen mediante los métodos expuestos se ilustran en Fig. 4 y Fig. 7.

III. ARQUITECTURAS BÁSICAS

A. Arquitectura con PLC

El protocolo de comunicación por red Ethernet, Modbus TCP, define un servidor y un cliente. La Fig. 4 esquematiza la arquitectura basada en PLC.

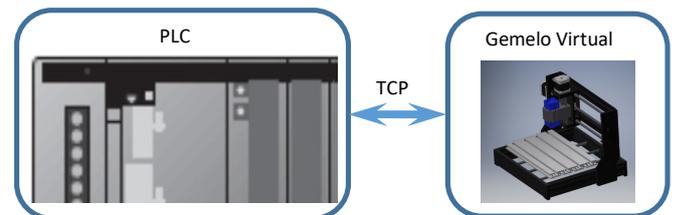


Fig. 4. Arquitectura básica para PLC.

El gemelo virtual en Blender® es designado como cliente utilizando la librería pymodbus, de acuerdo, al algoritmo de la Fig. 5 [10].

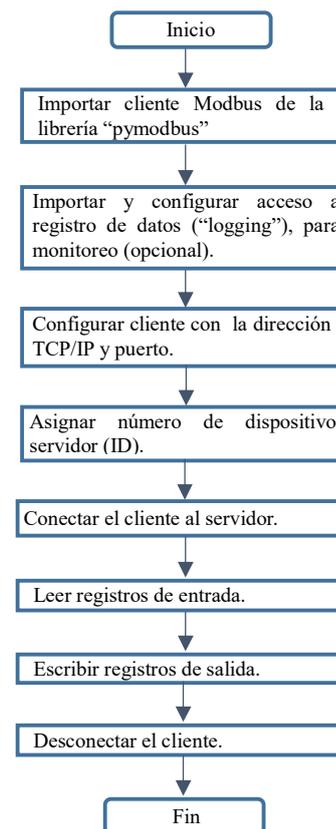


Fig. 5. Configuración de gemelo virtual como cliente Modbus TCP.

Como puede apreciarse, el cliente es el encargado de realizar las operaciones de lectura y escritura. El PLC está preconfigurado como servidor, y se dedica a la ejecución de funciones de control.

El tiempo de comunicación bajo esta configuración, como se indica en la Fig. 6, es de 29 ms para la transferencia de 9 registros: 6 registros para manipular actuadores (información del controlador a gemelo virtual, 2 registros por actuador para magnitud y dirección de manipulación), y 3 registros de posición en los tres ejes (información del gemelo virtual al controlador).

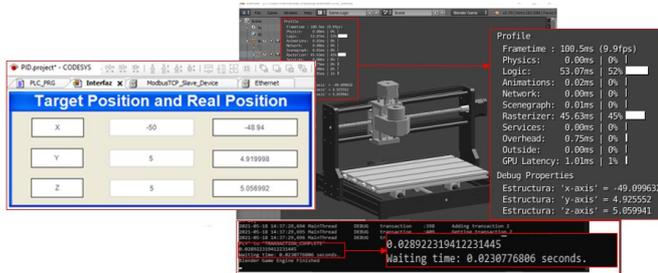


Fig. 6. Funcionamiento con PLC y Modbus TCP.

B. Arquitectura con Microcontrolador

La comunicación del microcontrolador Arduino® con el gemelo virtual puede implementarse por Modbus TCP o Modbus RTU. La Fig. 7 muestra la arquitectura básica de comunicación utilizando un microcontrolador para operar el gemelo virtual.

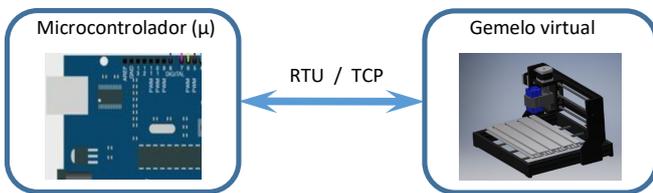


Fig. 7. Arquitectura básica para microcontrolador.

Bajo el protocolo Modbus TCP, el gemelo virtual se configura como cliente con el algoritmo de la Fig. 5, y el microcontrolador se configura como el servidor, según el procedimiento mostrado en la Fig. 8.

Con la configuración Modbus TCP se obtuvo un tiempo de 29 ms para la transferencia de 9 registros en total entre el microcontrolador y el gemelo virtual (6 de manipulación, 3 de posición X, Y y Z), como se muestra en la Fig. 9.

El gemelo virtual en Blender® y el microcontrolador Arduino pueden comunicarse con la arquitectura Modbus RTU, con el gemelo virtual como maestro y el microcontrolador como esclavo. Los algoritmos para esta configuración se muestran en las Fig. 10 y 11.

Con la arquitectura de Modbus RTU con el gemelo virtual y el microcontrolador, el tiempo de comunicación logrado es de 31 ms para el intercambio de 9 registros (6 para manipulación y 3 para posición), según la Fig. 12.

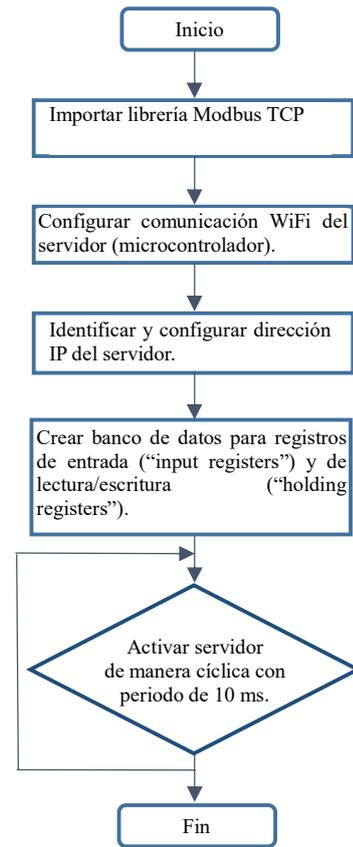


Fig. 8. Configuración del microcontrolador como servidor Modbus TCP.

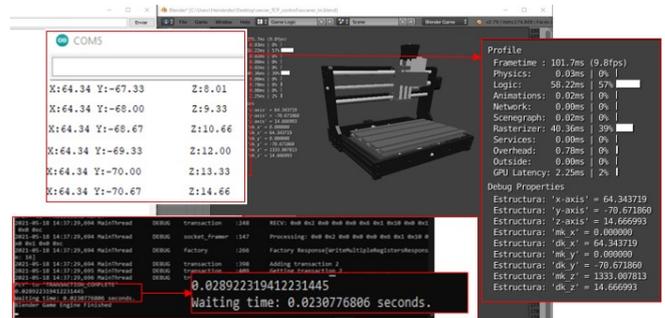


Fig. 9. Funcionamiento con microcontrolador y Modbus TCP.

Puede observarse que con Modbus TCP, los controladores, ya sea PLC o microcontrolador, asumen el rol de servidores. En Modbus RTU, el gemelo virtual se designa como maestro, lo cual corresponde a su función de cliente en el protocolo ModbusTCP.

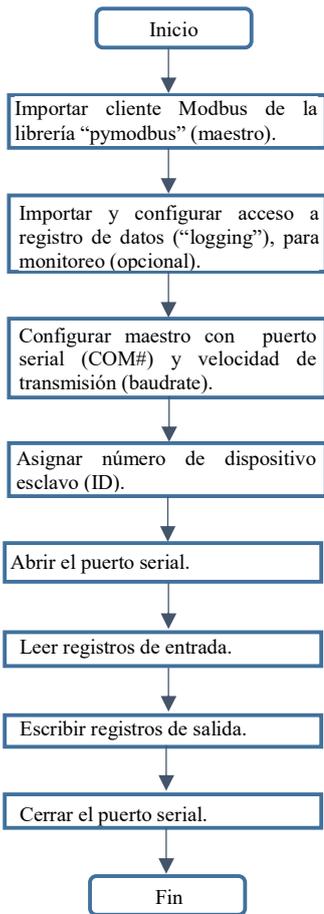


Fig 10. Configuración del gemelo virtual como maestro Modbus RTU.

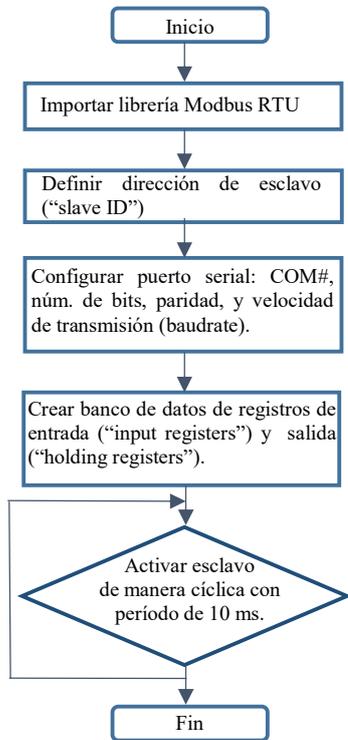


Fig 11. Configuración de microcontrolador como esclavo Modbus RTU

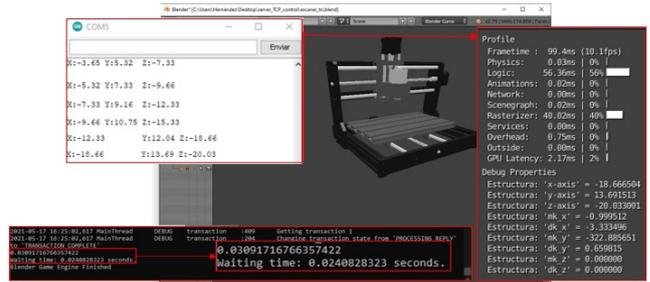


Fig. 12. Funcionamiento con microcontrolador y Modbus RTU.

IV. ARQUITECTURAS PROPUESTAS

A. Arquitectura en Cascada

Las comunicaciones en base al protocolo Modbus posibilitan la configuración de otras arquitecturas. En un esquema en cascada o serie puede proponerse el PLC como elemento primario de control, el microcontrolador como elemento secundario de control, y finalmente el gemelo virtual como proceso. En este esquema se propone la interacción directa entre el microcontrolador y el gemelo virtual, por lo que la información de los sensores de posición se recibiría en el microcontrolador, y éste enviaría las manipulaciones hacia los motores del gemelo virtual para producir movimiento. El PLC tiene una mayor jerarquía, y puede proporcionar al microcontrolador referencias de posición a partir de la implementación de un generador de trayectoria. En la Fig. 13 se muestra la arquitectura en cascada.

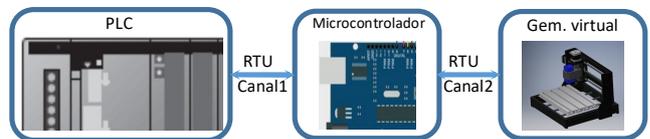


Fig. 13. Arquitectura cascada o en serie.

Las ventajas de la arquitectura en cascada son su sencillez, y el soporte de jerarquización de funciones, mientras que sus desventajas son el doble retraso en la comunicación al tener dos enlaces Modbus RTU, y por tanto el retraso en la detección de fallas del proceso por parte del elemento primario.

B. Arquitectura en Paralelo

Una arquitectura en paralelo se caracteriza por tener una ejecución simultánea de dos o más componentes de procesamiento. En la Fig. 14 se muestra una propuesta para implementar PLC y microcontrolador en paralelo con el gemelo virtual.

Como en la configuración en cascada se distribuyen las funciones de control, pero no necesariamente se asume que el PLC es un controlador de mayor jerarquía que el microcontrolador.

En esta arquitectura, algunos sensores pueden ser leídos de manera más rápida por el PLC y, por lo tanto, éste puede asumir las funciones de control que dependen de éstos.

El esquema en paralelo explota el concepto de IoT que se basa en la señalización accesible a todos los controladores,

para fines de supervisión, sin embargo, puede introducir asincronía en la comunicación y, por tanto, en el control.

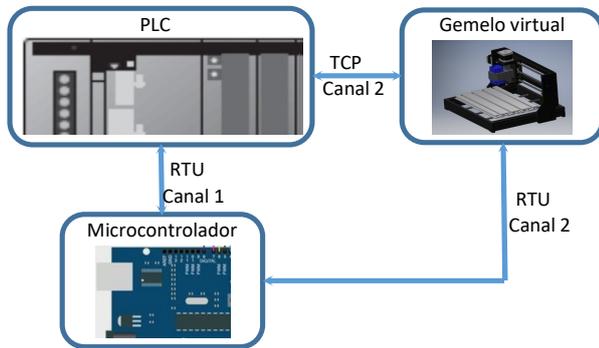


Fig. 14. Arquitectura en paralelo.

C. Arquitectura Distribuida

La arquitectura distribuida permite tener una configuración de un conjunto de componentes de procesamiento independientes que actúan como un único elemento. En la Fig. 15 se muestra la propuesta de un sistema distribuido.

El PLC es un controlador de mayor jerarquía que los microcontroladores. En el PLC se puede implementar el control cartesiano y de trayectoria del efector final de un manipulador robótico en general. En el caso de un manipulador con juntas rotacionales, el PLC puede determinar la posición deseada de la herramienta en coordenadas cartesianas, resolver la cinemática inversa para especificar los desplazamientos deseados en las juntas individuales, así como implementar los generadores de trayectoria correspondientes para una transición gradual entre puntos iniciales y finales.

Los microcontroladores pueden asignarse a cada uno de los ejes de movimiento, es decir, tendrían los algoritmos de control para operar los actuadores en base a los desplazamientos deseados o referencias proporcionadas por el PLC. La arquitectura distribuida favorece un mejor desempeño al permitir el control simultáneo o en paralelo de los ejes de movimiento.

La desventaja de la configuración distribuida es que requiere de más canales de comunicación y de una mejor sincronización.

V. CONCLUSIONES

El haber logrado la comunicación de dos plataformas distintas de control con el router virtual facilita la construcción de arquitecturas donde se puedan combinar distintas tecnologías de control. El PLC tiene mejores características de procesamiento, pero el microcontrolador permite implementar un control compacto, independiente y dedicado a ciertos elementos del sistema o máquina, es por eso que dichas plataformas son sometidas a prueba para conocer su desempeño y lograr su integración con el gemelo virtual, logrando tiempos similares de comunicación.

Las configuraciones básicas son consistentes en cuanto a la asignación del gemelo virtual como cliente o maestro, y al controlador como servidor o esclavo en Modbus TCP y RTU.

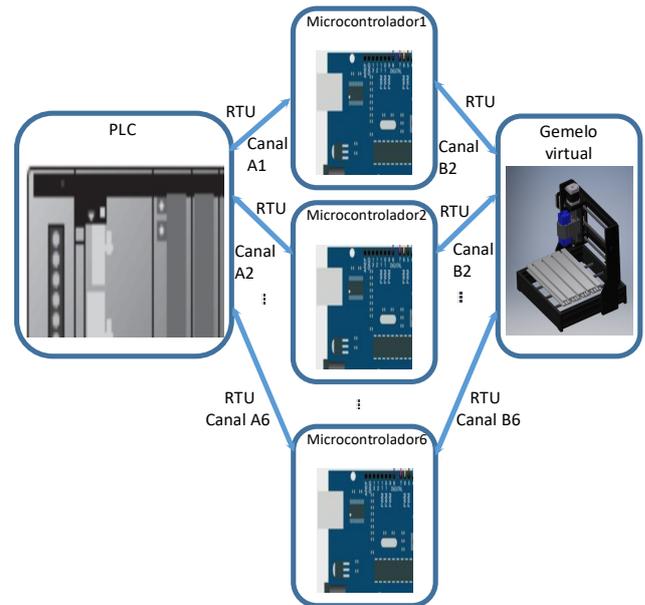


Fig. 15. Arquitectura distribuida.

Las arquitecturas propuestas permiten combinar dos plataformas de control comunes y permiten la jerarquización y/o la distribución de funciones de control. La implementación de las variantes propuestas de arquitecturas también ha de explorarse en el futuro utilizando el router virtual, e integrando la operación del router físico.

REFERENCIAS

- [1] Dalenogare, L.S., Benitez, G. B., Ayala, N. F., Frank, A. G.: The expected contribution of Industry 4.0 technologies for industrial performance. *Int. J. Prod. Econ.* 204, 383–394 (2018). doi: 10.1016/j.ijpe.2018.08.019
- [2] Lu, Yuqian & Liu, Chao & Wang, Kevin & Huang, Huiyue & Xu, Xun. (2019). Digital twin-driven smart manufacturing: Connotation, reference model, applications and research issues.
- [3] Xu, Y., Sun, Y., Liu, X., Zheng, Y.: A Digital-Twin-Assisted Fault Diagnosis Using Deep Transfer Learning. *IEEE Access*, 7, 19990–19999 (2019). doi: 10.1109/ACCESS.2018.2890566
- [4] Orive, D., Iriondo, N., Burgos, A., Saráchaga, I., Álvarez, M. L., Marcos, M.: Fault injection in Digital Twin as a means to test the response to process faults at virtual commissioning. In 2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), 1230–1234 (2019). doi: 10.1109/ETFA.2019.8869334
- [5] K Herbus, P. Ocieпка. Integration of the virtual 3D model of a control system with the virtual controller. *IOP Conf. Series: Materials Science and Engineering* 95 (2015) 012084
- [6] Krip Smerpitak. Et. Al. A Technique for mechanical adjustment of DC Motor by Virtual Model. *International Conference on Control, Automation and Systems* 2007. Oct.17-20, 2007 in COEX, Seoul, Korea
- [7] Vinzenz Bargsten, Pablo Zometa, Rolf Findeisen. Modeling Parameters Identification and Model-Based Control of a Lightweight Robotic Manipulator. *IEEE International Conference on Control Applications (CCA)*. 978-1-4799-1559-0/13/. 2007
- [8] Rina Familia. A Virtual Laboratory for Cooperative Learning of Robotics and Mechatronics. Dominican Republic; ITHET 6th Annual International Conference T2B-20. 0-7803-9141-1/05/. 2005 IEEE
- [9] Francisco A. Candelas. Et al. A Virtual Laboratory for Teaching Robotics. *Int. J. Engng Ed.* Vol. 19, No. 3, pp. 363–370, 2003.
- [10] H. Hernández-Vázquez, I. Y. Sanchez, F. Martell, J. E. Guzmán, R. A. Ortiz, “Development of Virtual Router Machine for Modbus Open Connection”, *International Conference on Applied Science and Advanced Technology (ICASAT) 2021*, submitted for publication.