XLIV Jornadas de Automática 2023

Entorno PIL para la validación de controladores de turbinas eólicas basados en IEC-61131

Martínez-Inchusta, A.[a, *], Sierra-García, J. E.[b], Santos, M.[c], Leija, L[a,d].

[a] *Facultad de Informática, Universidad Complutense de Madrid, Calle del Prof. José García Santesmases, 9, 28040, España.*
[b] *Escuela Politécnica Superior, Universidad de Burgos, Avda. Cantabria, s/n, 09006 Burgos, España.*
[c] *Instituto de Tecnología del Conocimiento, Universidad Complutense de Madrid, Calle del Prof. José García Santesmases, 9, 28040, Madrid*
[d] *Sección de Bioelectrónica del Departamento de Ingeniería Eléctrica, CINVESTAV, IPN, México*

To cite this article: Martínez-Inchusta, A., Sierra-García, J. E., Santos, M., Leija, L., 2023. PIL environment for validation of wind turbine controllers based on IEC-61131. XLIV Jornadas de Automática, 807-812. https://doi.org/10.17979/spudc.9788497498609.807

Resumen

Cada día los sistemas de control son más complejos; los controladores de turbinas eólicas son un claro ejemplo de ello. Esto conlleva una demanda creciente de nuevas técnicas de verificación y validación que minimicen los posibles riesgos durante la fase de implementación en el dispositivo final. Las pruebas MIL (Model In the Loop), SIL (Software In the Loop), PIL (Processor In the Loop) y HIL (Hardware In the Loop) se realizan en momentos específicos del desarrollo del producto con el objetivo de obtener un sistema de control robusto y confiable de cara a su integración en el mundo real. El método de validación PIL ejecuta el software de control en el hardware real y comunica éste con la planta virtual, todo ello en tiempo real y de forma determinística. Esta técnica de validación permite corroborar que el software de control se ejecuta dentro de los tiempos de procesado requeridos (valida el hardware) y que las salidas generadas son las correctas para llevar al sistema a su estado objetivo (valida el software de control). En este trabajo se ha demostrado la posibilidad de implementar un PIL mediante un controlador desarrollado en IEC-61131 ejecutado en tiempo real basado en PC y una planta virtualizada bajo Simulink Desktop Real-Time®.

Palabras clave: Control, Tiempo real, Processor In the Loop (PIL), Validación hardware/software, Turbina eólica.

PIL environment for wind turbine controller validation

Control systems are becoming more complex every day; wind turbine controllers are a clear example of this. This leads to an increasing demand for new verification and validation techniques that minimize potential risks during the implementation phase in the final device. MIL (Model In the Loop), SIL (Software In the Loop), PIL (Processor In the Loop) and HIL (Hardware In the Loop) tests are performed at specific points during product development with the objective of obtaining a robust and reliable control systems for real-world integration. The PIL validation method runs the control software on the real hardware and communicates this with the virtual plant, all in real time and deterministically. This validation technique allows corroborating that the control software runs within the required processing times (validates the hardware) and that the generated outputs are the correct ones to take the system to its target state (validates the control software). In this work we have demonstrated the possibility of implementing a PIL using a controller developed in IEC-61131 running in real time based on PC and a virtualized plant under Simulink Desktop Real-Time®.

Key words: Control, Tiempo real, Processor In the Loop (PIL), Validación hardware/software, Turbina eólica.

1. Introducción

El mundo se encuentra inmerso en un proceso de descarbonización. La energía eólica ha demostrado ser una energía limpia muy eficiente. Sin embargo, las turbinas eólicas son sistemas con una dinámica compleja que hacen del diseño de sus controladores un gran reto (Sierra-García & Santos, 2021). Esto ha llevado a investigar técnicas novedosas de

simulación que permitan realizar controladores robustos y confiables previo a su despliegue en el dispositivo final.

Este artículo aborda el problema de cómo implementar una simulación Processor In the Loop (de ahora en adelante PIL) (Arof, et al., 2019) enfocada en el ámbito de las turbinas eólicas. Específicamente, busca obtener un sistema de control (planta + controlador) que trabaje de una manera determinística y en tiempo real, ayudando al ingeniero en el desarrollo de controladores.

La simulación de turbinas eólicas se ha empleado en trabajos anteriores. Por ejemplo, en (Kenko & Gambier, 2022) se presenta un entorno de simulación HiL con el que poder probar diferentes controles avanzados sobre turbinas de grandes dimensiones. En (Puleva et al., 2016) se implementa un HiL basado en un controlador S7-300 de Siemens con el objetivo de simular el aerogenerador en diferentes regímenes de trabajo. En (Maheshwari et al., 2023) se presenta una revisión completa de los simuladores de turbinas eólicas. En (Li et al., 2009) se muestra como un entorno HiL puede ayudar a suplir las carencias de aquellos laboratorios de pruebas de turbinas eólicas que no poseen la posibilidad de generar viento durante el test. En (Martínez et al., 2014) se presenta un emulador de turbinas eólicas capaz de simular las curvas de potencia de las turbinas sin utilizar un sistema de control de lazo cerrado.

La principal contribución de este trabajo es la implementación de un entorno de simulación PIL mediante la integración de herramientas comerciales como la plataforma de control TwinCAT 3® (conforme con el estándar industrial internacional IEC 61131) (Cruz et al., 2022) y el entorno de simulación Simulink®.

La estructura del trabajo es la siguiente. La sección II muestra la arquitectura del entorno PIL. La sección III describe la implementación de la turbina eólica virtual. La sección IV presenta el controlador en tiempo real basado en PC. En la sección V se muestra la comunicación entre la planta y el controlador. Por último, en la sección VI se presentan y discuten los resultados.

2. Arquitectura entorno validación PIL

Un entorno PIL básicamente esta compuesto por dos subsistemas; un procesador sobre el que se ejecuta el algoritmo de control y una planta virtualizada.

La Figura 2 muestra una imagen general del PIL implementado en el presente trabajo. En ella se observa la estructura típica de un lazo de control cerrado; el control se implementa en el lado del procesador, mientras que el modelo de la turbina eólica se simula bajo la plataforma Simulink®. Los datos de consigna hacia el modelo y los valores actuales empleados para cerrar el lazo se intercambian mediante el protocolo de comunicación UDP.

Para la implementación de la planta virtual y sus comunicaciones se ha empleado Simulink Desktop Real-Time.

Así mismo, para la implementación del algoritmo de control, así como el desarrollo de las librerías de comunicación con la planta virtual, se ha seleccionado el estándar internacional de programación IEC61131.

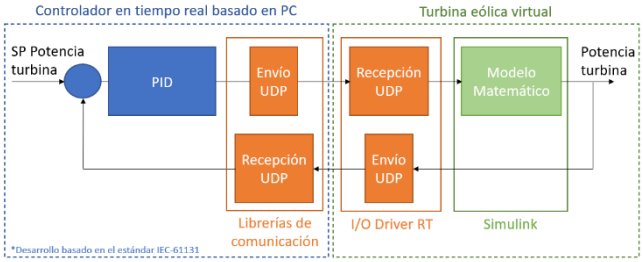

Figura 1: Arquitectura entorno PIL.

Los lenguajes de programación ofrecidos por el entorno de desarrollo TwinCAT 3® XAE (eXtended Automation Engineering) del fabricante Beckhoff® están alineados con este estándar. Así mismo, TwinCAT 3® XAR (eXtended Automation Runtime) ofrece la posibilidad de crear un entorno de ejecución determinístico sobre un sistema operativo Microsoft Windows®. Por todas estas bondades se ha seleccionado Beckhoff® como marca comercial sobre la que desarrollar el controlador en tiempo real basado en PC.

La comunicación entre el modelo virtual y el controlador en tiempo real basado en PC se realiza bajo el protocolo UDP (User Data Protocol). Mediante este protocolo la transmisión de datagramas se realiza sin establecimiento de la conexión, se reduce el ancho de banda y tiene una mayor compatibilidad con los PLCs industriales. La figura 2 muestra una captura de pantalla del controlador y la planta simulada funcionando de forma conjunta.

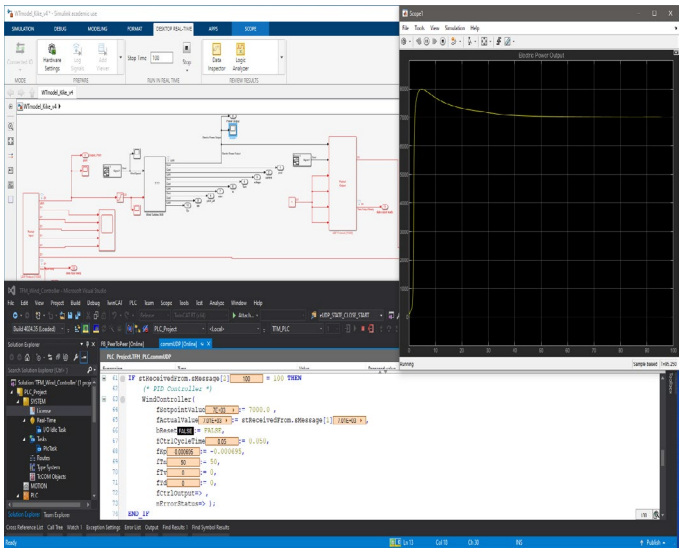

Figura 2: Modelo virtual y controlador real trabajando conjuntamente.

3. Implementación de la turbina eólica virtual

La virtualización de la turbina eólica se fundamenta en Simulink Desktop Real-Time®, el cual incluye un kernel en tiempo real para ejecutar modelos Simulink® sobre un sistema operativo Microsoft Windows®. Este kernel ejecuta en tiempo real los drivers de comunicación con el hardware y en paralelo ejecuta un modelo Simulink® en modo normal.

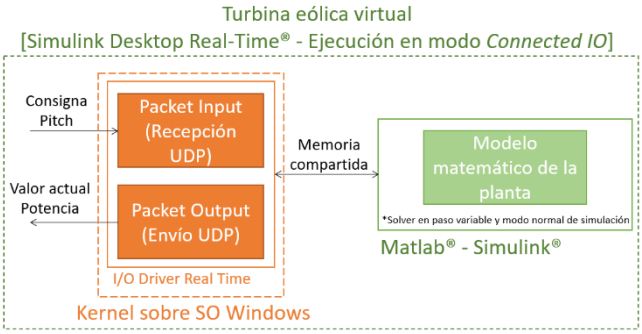

Figura 3: Vista en detalle de la turbina eólica virtual.

Simulink® y el kernel en tiempo real se ejecutan sobre el mismo PC e intercambian datos a través de una zona de memoria compartida. Ver Figura 3.

Simulink Desktop Real-Time® permite trabajar en dos modos diferentes; *Connected IO* y *Run in Kernel*.

El segundo de los modos emplea Simulink Coder para generar código en C++ del modelo matemático y los drivers de comunicación. Posteriormente, ese código será ejecutado sobre el kernel mencionado anteriormente o desplegado sobre un hardware de prototipado real. Esta es una solución idónea para entornos de pruebas HIL (Hardware In the Loop).

En el modo Connected IO (empleado en este trabajo), el algoritmo matemático que modela la turbina eólica no se ejecuta en tiempo real, sino en modo normal o acelerado sobre Simulink, no siendo necesario generar el modelo en C++ para su simulación. Esto último es la principal razón por la que se empleará este modo de trabajo, ya que de esta forma se evitan modificaciones indeseadas de las ecuaciones diferenciales a causa de la compilación de C++.

En (Sierra-Garcia et al., 2022) se desarrollan las ecuaciones diferenciales de una turbina eólica de 7kW empleadas en este trabajo. A continuación, se expone un resumen de las mismas:

$$\dot{I}_a = \frac{1}{L_a}\left(K_g \cdot K_\phi \cdot w - (R_a + R_L)I_a\right) \quad (1)$$

$$\lambda_i = \left[\left(\frac{1}{\lambda + c_8}\right) - \left(\frac{c_9}{\theta^3 + 1}\right)\right]^{-1} \quad (2)$$

$$C_p(\lambda_i, \theta) = c_1\left[\frac{C_2}{\lambda_i} - c_3\theta - c_4\theta^{c_5} - c_6\right]e^{-\frac{c_7}{\lambda_i}} \quad (3)$$

$$\dot{w} = \frac{1}{2 \cdot J \cdot w}\left(C_p(\lambda_i, \theta) \cdot \rho\pi R^2 \cdot v^3\right) - \frac{1}{J}\left(K_g \cdot K_\phi \cdot I_a + K_f w\right) \quad (4)$$

$$\ddot{\theta} = \frac{1}{T_\theta}\left[K_\theta\left(\theta_{ref} - \theta\right) - \dot{\theta}\right] \quad (5)$$

$$P_{out} = R_L \cdot {I_a}^2 \quad (6)$$

4. Controlador en tiempo real basado en PC

El controlador en tiempo real basado en PC se apoya en la plataforma de control TwinCAT 3® desarrollada por la compañía Beckhoff Automation GmbH.

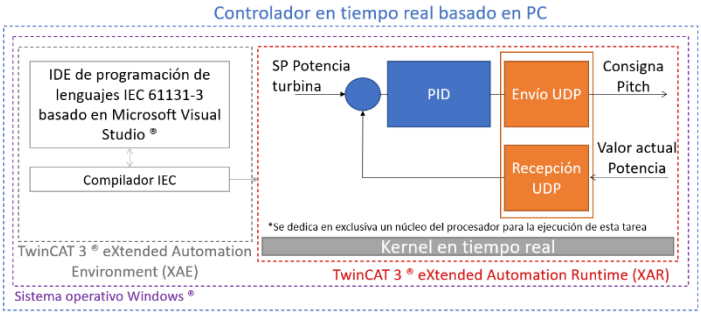

Figura 4: Vista en detalle del controlador en tiempo real basado en PC .

TwinCAT 3® principalmente se divide en dos componentes; eXtended Automation Environment (de ahora en adelante XAE) y eXtended Automation Runtime (de ahora en adelante XAR).

En la Figura 4 se representan ambos componentes. El componente XAE basado en Microsoft Visual Studio® (a la izquierda de la imagen) permite programar, depurar y compilar el software de control basado en los lenguajes de programación del estándar IEC 61131-3. El componente XAR (a la derecha de la imagen) permite ejecutar deterministamente la aplicación de control volcada desde el IDE de desarrollo.

TwinCAT 3® debe tener acceso directo a los recursos del hardware, ya que el kernel en tiempo real que incluye hará uso de estos durante la ejecución del software de control. Así mismo, al menos uno de los núcleos del procesador debe de estar dedicado en exclusiva y de manera aislada (es decir, no comparte tareas con Windows) al kernel de TwinCAT 3®.

4.1. Algoritmo de control basado en PID

Como se muestra en la Figura 4, el algoritmo de control está basado en un controlador PID básico con la función de transferencia representada en (7).

$$G(s) = K_p\left(1 + \frac{1}{T_n s} + \frac{T_v s}{1 + T_d s}\right) \quad (7)$$

La librería Tc2_Utilities de TwinCAT 3® pone a disposición del desarrollador diferentes POUs (Unidades de Organización del Programa) entre los que se encuentra el bloque de función FB_BasicPID empleado para la implementación del PID básico.

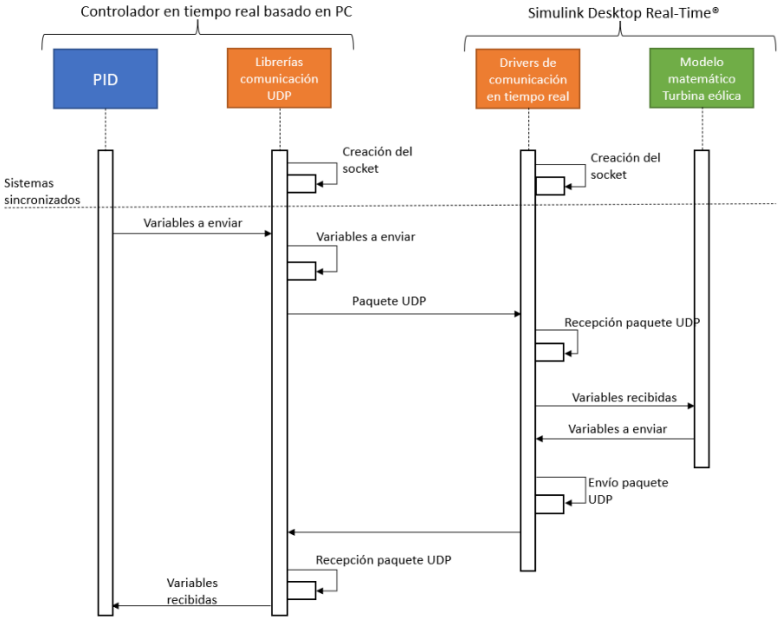

Figura 5: Secuencia de comunicación entre planta y controlador.

5. Comunicación entre el controlador y la planta

En la Figura 5 se expone un diagrama de secuencia de la comunicación e intercambio de datos entre los dos subsistemas del entorno de validación PIL.

El primer paso en ambos lados es la creación del socket. El segundo paso es la sincronización de ambos subsistemas; para ello el controlador basado en PC se encontrará en modo RUN a la espera de la señal de sincronización desde la turbina eólica virtual. Esta señal de sincronización será enviada una vez se arranque la simulación. Cuando el controlador basado en PC reciba esta señal, procesara los valores de entrada, ejecutara el algoritmo de control (PID) y generara los datos a enviar a la simulación. Desde este punto, se entra en un modo cíclico de intercambio de datos tal y como se expone en la Figura 6.

5.1. Librería de comunicación UDP basada en IEC61131

Las librerías de comunicación para el protocolo UDP implementadas bajo el estándar internacional de programación IEC61131 se apoyan en la función TF6310-TwinCAT 3® TCP/IP.

El comportamiento de la librería IEC61131 de comunicación UDP es el descrito en la Figura 7. Como se ha mencionado en la introducción de esta sección, el primer paso es la creación del socket. Una vez creado este, la librería revisa si el buffer de envío tiene datos (hasta que no llegue la señal de sincronización este buffer siempre estará vacío), en caso de estar lleno, se envían los datos. A la fase de lectura de datos se puede llegar por dos vías, o bien porque el buffer de envío está vacío o bien porque el envío se ha realizado de manera satisfactoria. Si la lectura se lleva a cabo correctamente, se comienza el ciclo de nuevo. El socket puede ser cerrado o bien porque se ha producido algún error durante la comunicación, o bien porque se ha dado orden de parar el control.

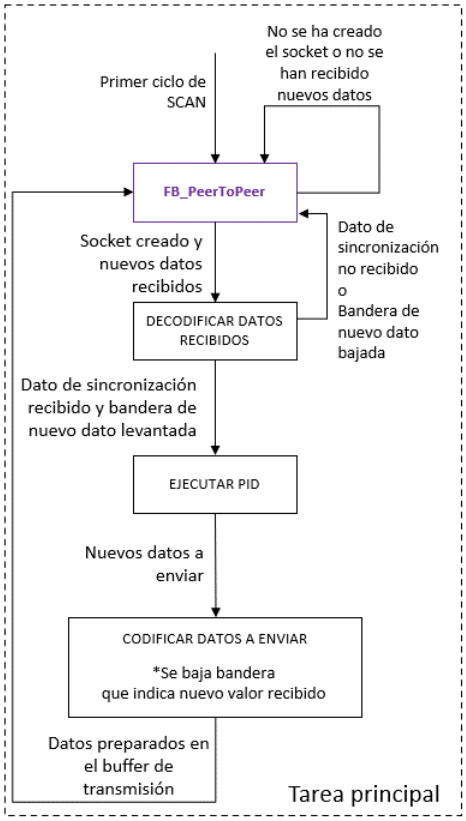

Figura 6: Tarea principal del controlador basado en PC .

La funcionalidad descrita en líneas anteriores ha sido encapsulada en un FB (Bloque de Función) llamado FB_PeerToPeer. Para su desarrollo se ha empleado texto estructurado (ST). Al tratarse de un bloque de función, es muy fácil convertir cualquier controlador basado en PC en un punto de comunicación UDP, simplemente es necesario el instanciamiento del bloque de función FB_PeerToPeer.

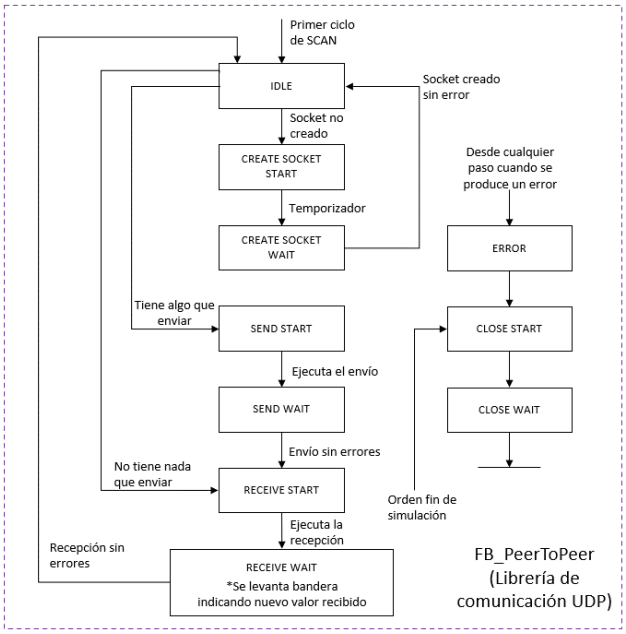

Figura 7: Secuencia de comunicación entre planta y controlador .

El bloque de función se activa a través de un nivel ascendente en la entrada bEnable. Tras ello, un nuevo socket se crea, y el intercambio de datos comienza. Este socket se crea en base a los datos de entrada sLocalHost y nLocalPort (dirección y puerto local). El buffer de datos a enviar se pasa al FB (VAR_IN_OUT) como una instancia al bloque de función FB_FrameFifo, el cual implementa una serie de acciones de manejo del buffer. Así mismo, los datos recibidos se sirven al exterior mediante el uso de otra instancia del bloque de función FB_FrameFifo. Ambos buffers son declarados como una estructura (STRUCT), la cual está formada por la dirección remota de tipo ST_SockAddr y el array de LREAL a intercambiar con el punto remoto.

El bloque de función ha sido instanciado en el PRG (programa) MAIN. Este programa está asociado a la tarea de ejecución principal PlcTask. Además del instanciamiento, en este PRG se lee el buffer de entrada datos, se ejecuta el PID y se rellena el buffer de datos a enviar al modelo virtual (ver Figura 6). El lenguaje de programación empleado en MAIN, es el mismo que en el bloque de función FB_PeerToPeer.

5.2. Integración drivers I/O de Simulink Desktop Real-Time®

La librería Simulink Desktop Real-Time® proporciona bloques que permiten conectar en tiempo real el modelo en Simulink® con el hardware real. En este trabajo se utilizarán los bloques Packet Input y Packet Output que leen y escriben datos binarios sin formato de un puerto.

6. Resultados

En la presente sección se presentarán los resultados obtenidos tras la implementación del entorno PIL presentado a lo largo del documento.

Como se ha mencionado anteriormente, existen numerosos propósitos a la hora de implementar un control de turbinas eólicas, este trabajo se focalizará en controlar la salida de la potencia eléctrica de la turbina a través del control del ángulo de pitch. Los parámetros de la turbina usados durante la simulación se muestran en (Sierra-Garcia et al., 2022).

Tanto el controlador basado en PC como el modelo en Simulink® almacenarán los datos intercambiados en un buffer con el objeto de evidenciar la correcta comunicación entre ambos subsistemas.

La Figura 8 muestra como está configurado el controlador basado en PC. De los ocho núcleos del PC, únicamente uno se ha dedicado de manera aislada al kernel de TwinCAT 3®. Así mismo, la tarea de lectura y escritura del mapa de entradas y salidas se ejecuta con una frecuencia de 1 ms (esta tarea no es relevante para este trabajo ya que no se está controlando una planta real). Por su parte, la tarea principal, encargada de las comunicaciones con la turbina eólica virtual, así como de la ejecución del algoritmo de control, se ejecuta con una frecuencia de 10 ms. En base a lo reflejado en la Figura 6, la comunicación UDP se ejecutará de manera completa cada 50 ms. Por su parte, el bloque de control PID se ejecutará cada 50 ms, coincidiendo con la recepción de un nuevo dato de potencia eléctrica desde el modelo virtual.

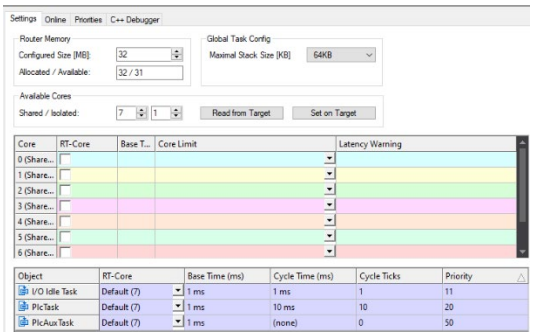

Figura 8: Configuración del controlador basado en PC .

La configuración del solver de Simulink® para la resolución del modelo matemático que simula la turbina eólica presenta los siguientes valores destacados; se ha seleccionado un tipo de solver automático (es decir Simulink® escoge el mejor en cada momento) de ejecución en paso variable. Así mismo, el mayor tiempo de paso será de 10 ms con una tolerancia de 1e-3. Con este paso de simulación se garantiza que la simulación en tiempo real no se ralentice.

Simulink Desktop Real-Time® será ejecutado en modo Connected IO por las razones mencionadas en líneas anteriores.

Los drivers de comunicación UDP en el lado de Simulink®, tanto el de envió como el de recepción trabajarán con una frecuencia de 50 ms.

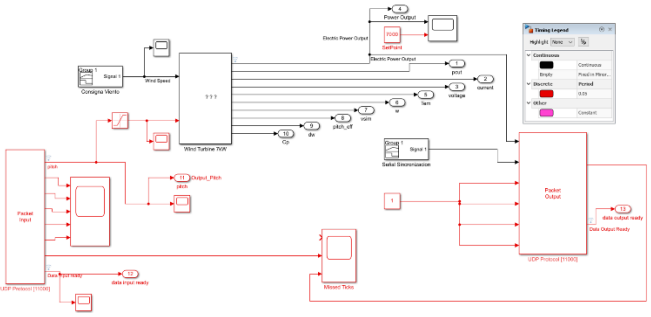

Figura 9: Modelo virtual implementado en Simulink® .

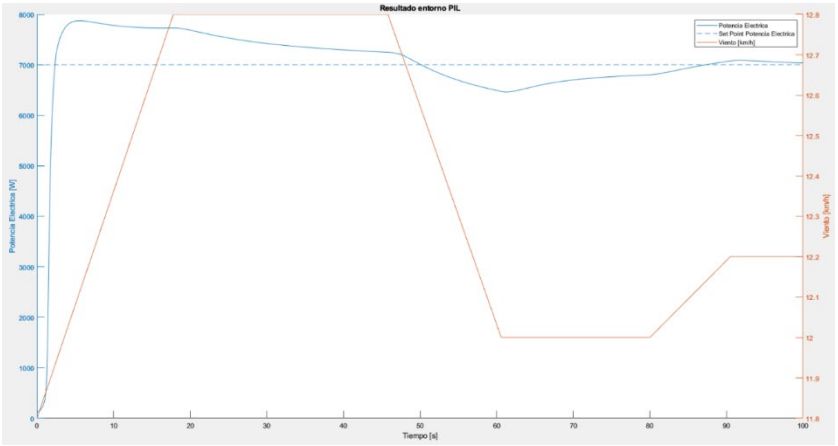

Figura 10: Potencia eléctrica de salida .

En la Figura 9 se muestra una imagen general de la implementación realizada en Simulink. En ella se puede observar que el modelo virtualizado de la turbina eólica se ejecutara en tiempo continuo, mientras que los bloques de comunicación tendrán un periodo de trabajo de 50 ms.

Por su parte, la Figura 10 muestra la potencia eléctrica generada por la turbina eólica tras ejecutar el entorno PIL. Se trata de un experimento de 100 s, realizado en las condiciones descritas a lo largo del trabajo. El control PID consigue estabilizar la turbina eólica en 7 kW en mitad del experimento.

SIMULINK -> CONTROLADOR BASADO EN PC (Potencia)			
SIMULINK		CONTROLADOR	
TIEMPO[ms]	DATO ENVIADO	TIEMPO[ms]	DATO RECIBIDO
0	0.000008000000	Tiempo de comunicación UDP	
50	136.611177362142	0	0.000008000000
100	129.538770093265	50	136.611177362142
150	124.408134602619	100	129.538770093265
200	124.924235728612	150	124.408134602619

CONTROLADOR BASADO EN PC -> SIMULINK (Pitch)			
CONTROLADOR		SIMULINK	
TIEMPO[ms]	DATO ENVIADO	TIEMPO[ms]	DATO RECIBIDO
0	0	0	0
20	-4.86743249443722	50	-4.86743249443722
70	-4.89175749440942	100	-4.89175749440942
120	-4.82600804000697	150	-4.82600804000697
170	-4.85345046734273	200	-4.85345046734273

Figura 11: Primeras muestras de intercambio entre planta y controlador.

Por último, la Figura 11 expone los cinco primeros datos intercambiados entre la turbina eólica y el controlador basado en PC (en ambos sentidos).

El controlador basado en PC recibirá el primer dato con el retardo propio de la comunicación UDP. De tal manera, que si la comunicación mantiene un tiempo de ciclo estable no deberían de producirse pérdidas de datos.

Así mismo la primera consigna de pitch que recibirá la turbina eólica será de cero, ya que aún no se habrá ejecutado el PID. A los 20 ms enviará la siguiente consigna de pitch, la cual será procesada a los 50 ms por la turbina eólica. Al trabajar con un margen de 30 ms entre envío y recepción tampoco deberían producirse pérdidas de datos.

Para finalizar este apartado cabe mencionar las posibilidades de reutilización de este entorno PIL en trabajos futuros. Prácticamente la mayoría del entorno puede ser empleado en otros casos de estudio ya que se ha concebido con este propósito. En la parte del modelo virtual bastaría con sustituir el bloque del modelo matemático de la turbina eólica por el nuevo modelo a estudiar. Toda la gestión de comunicaciones puede ser mantenida y dado que es fácil su configuración, bastaría con cambiar el tamaño de los paquetes a intercambiar, así como las direcciones IPs y puertos a emplear. En el lado del controlador basado en PC, sucede algo similar, bastaría con sustituir el bloque PID por otro tipo de controlador, por ejemplo, un controlador basado en aprendizaje por refuerzo. Como la librería de comunicación es un FB, es instanciable y únicamente requiere configurarlo correctamente.

Agradecimientos

Este trabajo ha sido parcialmente financiado por el Ministerio de Ciencia e Innovación español en el marco del proyecto MCI/AEI/FEDER PID2021-123543OB-C21.

Referencias

Arof, S., Diyanah, N. H., Yaakop, N. M., Mawby, P. A., & Arof, H. (2019). Processor in the loop for testing series motor four quadrants drive direct current chopper for series motor driven electric car: part1: chopper operation modes testing. Advanced Engineering for Processes and Technologies, 59-76.

Cruz, E. M., Carrillo, L. G., Salazar, L. C., & Rojo, B. H. (2022, September). Comparación de estándares IEC 61131-3 e IEC 61499 para implementar sistemas de control distribuido. In: XXVIII Congreso Internacional Anual de la SOMIM.

Kenko, David & Gambier, Adrian. (2022). Real-time wind turbine simulation for pitch control purposes by using a hardware-in-the-loop approach. 10.46354/i3m.2022.mas.026.

Li, Jichen & Zhou, Bo & Guo, Honghao. (2009). Hardware-in-loop simulation of wind turbine based on BLDCM. 1 - 5. 10.1109/WNWEC.2009.5335825.

Maheshwari, Z., Kengne, K., & Bhat, O. (2023). A comprehensive review on wind turbine emulators. Renewable and Sustainable Energy Reviews, 180, 113297.

Martinez, F., Herrero, L. C., & de Pablo, S. (2014). Open loop wind turbine emulator. Renewable Energy, 63, 212-221.

Puleva, Teofana & Ruzhekov, Georgi & Slavov, Tsonyo & Rakov, B.. (2016). Hardware in the loop (HIL) simulation of wind turbine power control. 64 (8 .)-64 (8 .). 10.1049/cp.2016.1053.

Sierra-García, J. E., & Santos, M. (2021). Redes neuronales y aprendizaje por refuerzo en el control de turbinas eólicas. Revista Iberoamericana de Automática e Informática industrial, 18(4), 327-335.

Sierra-Garcia, J. E., Santos, M., & Pandit, R. (2022). Wind turbine pitch reinforcement learning control improved by PID regulator and learning observer. Engineering Applications of Artificial Intelligence, 111, 104769.