

Jornadas de Automática

Control ROSCO en PLC mediante arquitectura Software-in-the-Loop para Turbinas Eólicas

Germán Corredera^a, Ignacio López^b, Francisco Vázquez^b, Juan Garrido^b, Mario L. Ruz^{a,*}

^aDepartamento de Mecánica, Universidad de Córdoba, N-IV, km 396, 14014 Córdoba, España

^bDepartamento de Ingeniería Eléctrica y Automática, Universidad de Córdoba, N-IV, km 396, 14014 Córdoba, España

To cite this article: Corredera, G, López, I, Vázquez, F, Garrido, J, Ruz, M.L. 2026. ROSCO Control on PLC via Software-in-the-Loop Architecture for Wind Turbines Jornadas de Automática, 47.
<https://doi.org/10.17979/ja-cea.2026.47.13854>

Resumen

En este trabajo se presenta la implementación y validación preliminar del controlador de referencia de código abierto ROSCO, empleado en turbinas eólicas, en un PLC industrial de Siemens virtualizado. Para la traducción del algoritmo de control se ha empleado el lenguaje estructurado SCL. La validación se lleva a cabo mediante una arquitectura de co-simulación *Software-in-the-Loop* (SIL) que enlaza PLCSim Advanced a través de Simulink con el modelo aerodinámico del aerogenerador, implementado en OpenFAST. El desempeño del controlador implementado en el PLC virtual se evalúa frente al controlador nativo utilizando escenarios de viento determinista y turbulento. Asimismo, se analiza el comportamiento del módulo de Control de Paso Individual (IPC) y su efecto sobre la fatiga estructural mediante el cálculo de la carga de daño equivalente (DEL, *Damage Equivalent Load*). Los resultados demuestran una buena correspondencia, alcanzándose en la mayoría de los casos un índice VAF (*Variance Accounted For*) superior al 90 %.

Palabras clave: Control de recursos energéticos renovables, Software-in-the-Loop, Turbinas eólicas, Simulación, Controlador lógico programable.

ROSCO Control on PLC via Software-in-the-Loop Architecture for Wind Turbines

Abstract

This work presents the implementation and preliminary validation of the open-source reference controller ROSCO for wind turbines on a virtualized Siemens industrial PLC. The control algorithm was translated using the Structured Control Language (SCL). Validation is carried out through a Software-in-the-Loop (SIL) co-simulation architecture that integrates PLCSim Advanced, via Simulink, with the aerodynamic wind turbine model implemented in OpenFAST. The performance of the controller implemented on the virtual PLC is evaluated against the native controller under deterministic and turbulent wind scenarios. In addition, the behavior of the Individual Pitch Control (IPC) module and its influence on structural fatigue is analyzed through the calculation of the Damage Equivalent Load (DEL). The results demonstrate good agreement, achieving a VAF (*Variance Accounted For*) index of over 90 % in most cases.

Keywords: Control of renewable energy resources, Software-in-the-Loop, Wind turbines, Simulation, Programmable logic controller.

1. Introducción

La lucha contra el cambio climático es una prioridad para la Unión Europea (UE). Prueba de ello es el hecho de haber al-

canzado los tres objetivos climáticos que se propusieron para 2020 de reducir las emisiones de gases de efecto invernadero en un 20 % respecto a los niveles de 1990, aumentar la cuota

*Autor para correspondencia: mario.ruz@uco.es
Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0)

de uso de energías renovables hasta el 20 % y mejorar la eficiencia energética en un 20 %. La UE continúa haciendo grandes esfuerzos en la misma dirección con el objetivo de reducir las emisiones de gases de efecto invernadero en un 55 % para 2030, con el propósito a largo plazo presentado en el Pacto Verde Europeo de alcanzar la neutralidad climática en 2050.

Dentro de las energías renovables, la eólica constituye una de las claves para la transformación energética a nivel nacional, europeo e internacional. Uno de los objetivos fundamentales de control de las turbinas eólicas, es conseguir que la extracción de energía o la potencia obtenida sea lo más cercana a la máxima extraíble. La principal perturbación del sistema es el viento, del cual se obtiene la energía. En las turbinas modernas, un controlador se encarga de modificar las acciones de control para cumplir ciertos objetivos, entre los que se destacan la maximización de la energía extraída según el punto de trabajo del sistema (Muñoz-Palomeque et al., 2024), y aspectos relacionados con la seguridad de la estructura del aerogenerador y la fatiga que sufre.

Para evaluar el desempeño de un controlador hay diferentes entornos que permiten simular aerogeneradores. Uno de ellos es OpenFAST (NREL, 2026a), un software de código abierto que se utiliza para el modelado y simulación de turbinas eólicas de eje horizontal, tanto fijas como flotantes. Permite simular el comportamiento de dichas turbinas en una variedad de condiciones de viento, incluyendo vientos extremos y cambios en la dirección. Para ello, utiliza modelos matemáticos y algoritmos que permiten reproducir la dinámica de la turbina. Este artículo toma como base la turbina de referencia fija de NREL de 5MW (NREL, 2026b), desarrollada por el Laboratorio Nacional de Energías Renovables de EE. UU. Este *software* de código abierto se ha consolidado como estándar en la investigación eólica.

Existe bastante literatura disponible sobre la implementación del control de un aerogenerador, tanto a nivel teórico, como mediante *software* disponibles en el ámbito académico y de investigación, como por ejemplo MATLAB o el propio OpenFAST. En lo que respecta a OpenFAST, es de destacar que proporciona una interfaz que permite conectarle un controlador externo. A este respecto hay ejemplos disponibles de código abierto, siendo destacable ROSCO (*Reference OpenSource Controller*), implementado en lenguaje Fortran (NREL, 2026c), y configurable desde Python para su sintonía. Sin embargo, no hay tanta información disponible de la programación en controladores industriales reales (Vestas, Siemens Gamesa, etc.), implementados en PLC o controladores embebidos propietarios, que no se publican en detalle por motivos comerciales, de seguridad y de certificación. En este trabajo se ha llevado a cabo la traducción del código de los módulos principales para turbinas fijas del controlador ROSCO a lenguaje de programación de PLC de Siemens. Para el modelado del aerogenerador se ha utilizado OpenFAST, empleando Simulink como plataforma intermedia para el intercambio de datos de forma sincronizada.

El resto del artículo se organiza de la siguiente manera: en la sección 2 se describe cómo se ha realizado la implementación, utilizando el paradigma *Software-in-the-Loop* (SIL). Se destacan los bloques principales empleados para la comunicación entre el controlador implementado en el PLC y OpenFAST. En la sección 3 se entra en detalle en cómo se ha llevado

a cabo la traducción de ROSCO a lenguaje PLC de Siemens. En la sección 4 se valida el controlador mediante una serie de perfiles de viento. Finalmente, las conclusiones del trabajo se resumen en la sección 5.

2. Arquitectura *Software-in-the-Loop*

Las herramientas de simulación son fundamentales para estudiar el sistema de control de un aerogenerador (Kenko and Gambier, 2022; Martínez-Inchusta et al., 2023; Puleva et al., 2016). Uno de los posibles enfoques es la metodología SIL, donde todo el sistema (planta y controlador) se implementa en *software*. En este trabajo se utiliza MATLAB/Simulink, integrando el *software* de código abierto OpenFAST mediante una S-Function, permitiendo el modelado físico de la turbina eólica y la monitorización de sus variables en tiempo de simulación. El lazo de control se cierra mediante una segunda S-Function desarrollada por Siemens que, entre otras opciones, permite la comunicación con una instancia virtual de un PLC mediante PLCSim Advanced, donde reside la lógica de control. Se comentan a continuación cada uno de estos elementos.

2.1. PLCSim Advanced

El flujo de trabajo comienza en TIA Portal, donde se desarrolla el control en lenguaje de texto estructurado y se definen las variables del sistema (momentos en la base de las palas y la torre, velocidad del generador, potencia generada, entre otras) y las señales de control (ángulo *pitch* de cada una de las palas y par electromagnético). Posteriormente se compila el código y se carga en instancia virtual del PLC generada con PLCSim Advanced. En Simulink, se referencia esta instancia en la S-Function correspondiente, tal y como se muestra en la Figura 1.

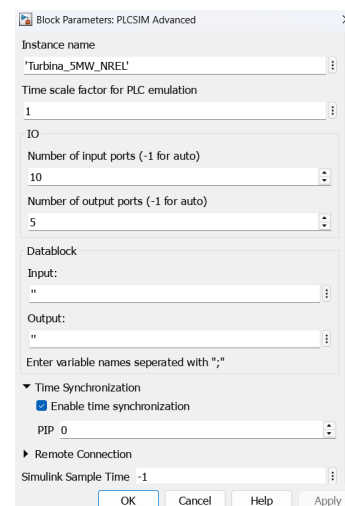


Figura 1: S-Function que llama a la instancia de PLCSim Advanced.

2.2. OpenFAST

Como se ha comentado, la dinámica de la turbina eólica se ha modelado mediante OpenFAST. El modelo empleado corresponde al aerogenerador de referencia de 5 MW de NREL (Jonkman et al., 2009). Como características principales, tiene unas velocidades de corte, nominal y de salida de 3 m/s,

11.4 m/s y 25 m/s, respectivamente, y un rango de velocidades del rotor de 6.9-12.1 rpm. Para su uso en Simulink, una S-Function integra el motor de OpenFAST como un bloque funcional, mostrado en color verde en la Figura 2. Esto permite diseñar la estrategia de control dentro del propio Simulink. Sin embargo, a diferencia del flujo de trabajo convencional de OpenFAST, donde el controlador se carga como una librería dinámica externa o se desarrolla en Simulink, en este trabajo se ha implementado un esquema de co-simulación síncrona en tiempo de ejecución. En este contexto, el tiempo de integración del modelo aeroelástico y el tiempo de ciclo del PLC se encuentran sincronizados y acoplados mediante la API de PLCSim Advanced desarrollada por Siemens.

2.3. Simulink

Simulink actúa como el núcleo de la arquitectura SIL, funcionando como intermediario en la co-simulación entre PLC-Sim Advanced y OpenFAST. En este entorno se gestiona el flujo de señales de forma síncrona, enviando al PLC los momentos flectores calculados por OpenFAST, el azimut y el resto de parámetros operativos. A partir de estos datos, el algoritmo de control calcula las consignas de par electromagnético y de *pitch* de las palas, enviándolas de vuelta a OpenFAST.

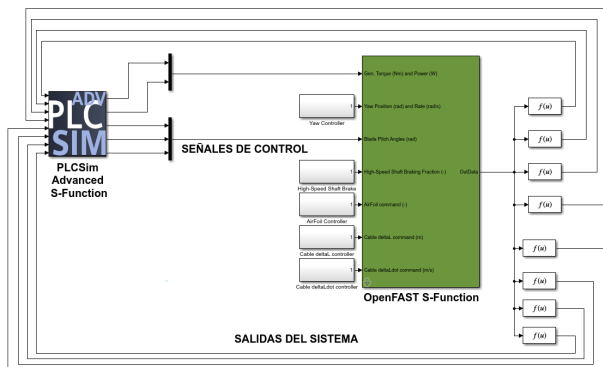


Figura 2: Esquema completo de Simulink.

3. Programación de ROSCO en PLC

El controlador ROSCO se desarrolló para proporcionar un módulo de control genérico para OpenFAST (Abbas et al., 2022). La arquitectura del controlador ROSCO se caracteriza por un diseño modular basado en bloques funcionales configurables que gestionan los distintos regímenes operativos del aerogenerador. Entre sus componentes principales se encuentran el control de par en la región de carga parcial, y el control colectivo de *pitch* (CPC) para la regulación de potencia, implementado mediante un controlador PI sobre la velocidad.

Adicionalmente, ROSCO permite incorporar funcionalidades avanzadas como el control individual de *pitch* (IPC), orientado a la mitigación de cargas asimétricas, así como estrategias de amortiguamiento estructural (por ejemplo, *tower damping*) y diversas lógicas de supervisión y seguridad destinadas a garantizar una operación estable del sistema.

El objetivo principal de este trabajo ha sido la migración y posterior validación del controlador de referencia ROSCO del NREL en un PLC de Siemens. Para ello, se ha llevado a cabo un proceso de integración de software, traduciendo la lógica

del código originalmente desarrollado en Fortran hacia un entorno de automatización industrial como lo es TIA Portal.

3.1. Integración del código

La estructura modular ha resultado clave para su implementación en el PLC, ya que ha permitido descomponer el controlador en bloques funcionales equivalentes dentro de TIA Portal, manteniendo la coherencia con la arquitectura original y facilitando tanto la validación como la futura extensibilidad del sistema. Se resume a continuación la correspondencia entre el código Fortran y SCL:

- **De Subroutines a Function Blocks (FB):** Las subrutinas de ROSCO, como el lazo de control CPC o ciertas funciones de filtrado, se han implementado como bloques de función. Esta decisión responde a la necesidad de preservar el estado de las variables entre ciclos de ejecución.
- **De Functions a FC:** Para los cálculos matemáticos puros, incluyendo transformaciones trigonométricas y operaciones de normalización, se han empleado funciones (FC). Al tratarse de operaciones sin memoria, su implementación mediante FC permite optimizar el uso de los recursos del PLC.
- **De DISCON a bloque de organización (OB):** Un aspecto crítico de la migración ha sido la adaptación de la interfaz de comunicación. En la implementación original de OpenFAST, el controlador se ejecuta a través de la subrutina *DISCON*, que actúa como punto de entrada único para el intercambio de señales entre el modelo aeroelástico y el controlador.

3.2. Módulo IPC

Debido a la cizalladura del viento, las palas experimentan cargas asimétricas, provocando una fatiga en el eje principal y en la base de las palas. Con el objetivo de mitigarlas, una de las soluciones más extendidas es la implementación del Control Individual de Pitch (IPC) (Bossanyi, 2003; Lara et al., 2025; Garrido Jurado et al., 2026). A diferencia del enfoque tradicional basado en el CPC, el cual regula la potencia y la velocidad del rotor operando en la región nominal mediante un ángulo de paso idéntico para todas las palas, las turbinas modernas integran cada vez más estrategias IPC. La principal ventaja de este esquema radica en su capacidad para atenuar el daño estructural acumulado en componentes críticos como las palas y la torre, manteniendo prácticamente inalterada la producción eléctrica. Para su ejecución, el algoritmo requiere la lectura de los momentos flectores en la raíz de las palas a través de sensores dedicados. No obstante, dado que estas mediciones se encuentran ligadas al sistema de coordenadas rotatorio de cada pala, es necesario aplicar la Transformación Multi-Pala (MBC) para proyectar dichas señales sobre un marco de referencia fijo (Mulders and van Wingerden, 2019). Mediante esta transformación, las cargas se desacoplan en componentes virtuales de cabeceo (*tilt*) y guiñada (*yaw*), permitiendo al IPC actuar directamente sobre ellas para su minimización, si bien en entornos operacionales reales suele persistir un leve acoplamiento residual entre ambos ejes.

Para la implementación de este módulo, primero se estiman los momentos en la base de las palas (M_1, M_2, M_3). Mediante la transformada MBC, estas tres señales se proyectan sobre un marco de referencia fijo, obteniéndose dos componentes: el momento de inclinación, M_{tilt} , y el momento de guiñada, M_{yaw} , donde ψ_i representa el ángulo de azimut de cada pala, desplazadas 120° entre sí:

$$\begin{bmatrix} M_{tilt} \\ M_{yaw} \end{bmatrix} = \frac{2}{3} \begin{bmatrix} \cos(\psi_1) & \cos(\psi_2) & \cos(\psi_3) \\ \sin(\psi_1) & \sin(\psi_2) & \sin(\psi_3) \end{bmatrix} \begin{bmatrix} M_1 \\ M_2 \\ M_3 \end{bmatrix} \quad (1)$$

Una vez determinadas estas dos componentes, el bloque de control IPC obtiene las correspondientes componentes de *pitch* en el marco de referencia fijo, β_{tilt} y β_{yaw} . Estas señales se convierten nuevamente al sistema de coordenadas rotacional, aplicando la transformación MBC inversa, obteniéndose de esta manera los valores adicionales de *pitch* para cada pala, β_{IPC1} , β_{IPC2} y β_{IPC3} :

$$\begin{bmatrix} \beta_{IPC1} \\ \beta_{IPC2} \\ \beta_{IPC3} \end{bmatrix} = \begin{bmatrix} \cos(\psi_1) & \sin(\psi_1) \\ \cos(\psi_2) & \sin(\psi_2) \\ \cos(\psi_3) & \sin(\psi_3) \end{bmatrix} \begin{bmatrix} \beta_{tilt} \\ \beta_{yaw} \end{bmatrix} \quad (2)$$

Finalmente, la consigna de *pitch* total para cada pala se obtiene sumando la componente individual a la consigna de paso colectivo CPC.

3.2.1. Implementación del Módulo IPC en TIA Portal

Para implementar este módulo, se ha encapsulado la lógica original de ROSCO dentro de un único Bloque de Función (FB) programado en SCL. En la Figura 3 se muestran las variables de entrada de la FB, que son los momentos en la base de las palas (M_1, M_2, M_3), el ángulo de rotación azimut y el viento. La función realiza la transformada inversa, generando las consignas *pitch* que se suman a la señal de control CPC.

```
"Inst_IPC"(i_rootMyc1 := "RootMyc1",
i_rootMyc2 := "RootMyc2",
i_rootMyc3 := "RootMyc3",
i_genSpeed := #temp_genSpeed_flt_rads,
i_azimuth := "azimuth",
i_wind_flt := "Wind F",
q_pitchComIPC_1P => "IPC".PitchComIPC_1P,
i_enable := NOT "DatosGlobales".g_bldPitchAutoMode
OR NOT "DatosGlobales".g_torqAutoMode
OR NOT "DatosGlobales".g_IPC_activo );
```

Figura 3: Llamada al módulo IPC.

De forma similar, se han implementado otros módulos de control avanzados, como el *Fore-Aft Damping*, basado en la actuación del *pitch* para reducir la carga estructural en la base de la torre.

4. Validación del controlador con ROSCO

Esta sección presenta el análisis comparativo del desempeño entre el controlador de referencia ROSCO de NREL (rojo), y que se implementa mediante una librería dinámica externa (.dll) y la arquitectura propuesta basada en el PLC (azul). La validación se ha realizado mediante el índice VAF (*Variance Accounted For*), utilizado para evaluar la similitud de las señales:

$$\text{VAF} = \left(1 - \frac{\text{var}(y - \hat{y})}{\text{var}(y)} \right) \times 100 \quad (3)$$

donde la variable y representa cualquier señal obtenida de la implementación original de ROSCO e $\hat{y}$ las correspondientes a la traducción del controlador a SCL con el esquema SIL descrito anteriormente. En todas las simulaciones realizadas están activos los módulos de control IPC y *Fore-Aft Damping*.

4.1. Viento uniforme

Se han generado dos perfiles de viento uniforme, en carga parcial y en régimen nominal, con una duración de 660 s. En los análisis posteriores se descartan los primeros 60 s con el fin de evitar los transitorios asociados al inicio de la simulación. El perfil incluye cambios bruscos para evaluar el comportamiento del controlador en un amplio rango de velocidades. Las Figuras 4 y 5 muestran la evolución temporal del par electromagnético, la velocidad angular del rotor y el ángulo de *pitch*. En carga parcial, los resultados son superiores al 99 % para el par electromagnético y la velocidad del rotor, y del 98.46 % para el *pitch*. En la región nominal (Figura 5), se obtienen resultados aceptables, con un VAF mínimo del 89.98 %.

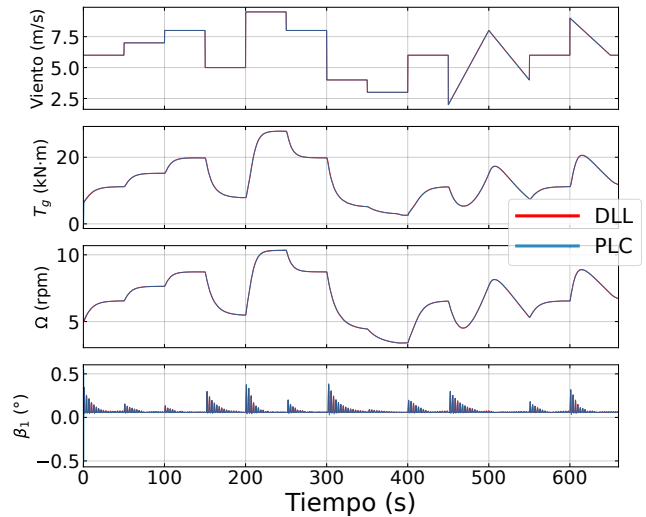


Figura 4: Comparación de las respuestas temporales en carga parcial con perfil de viento uniforme.

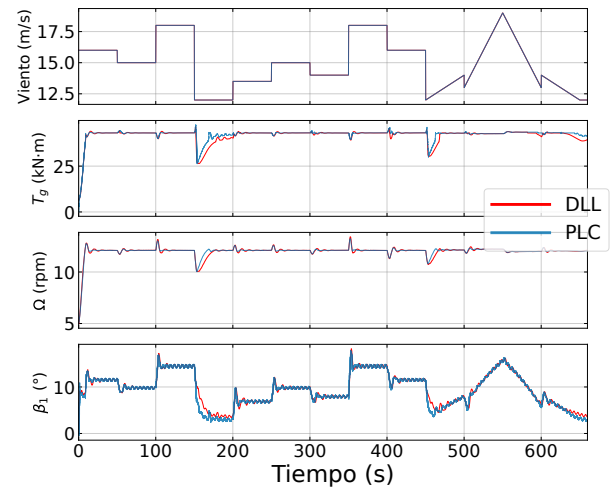


Figura 5: Comparación entre las respuestas temporales en carga nominal con perfil de viento uniforme.

4.2. Viento turbulento

Se evalúa el rendimiento del sistema bajo condiciones de viento turbulento, con una velocidad media de 15 m/s en régimen nominal y de 5 m/s en carga parcial. En ambos casos se emplea un espectro de turbulencia de Kaimal, con una intensidad de turbulencia del 26.18 % a la altura del buje y un exponente de cizalladura de 0.2. Los resultados obtenidos en la región de carga parcial son satisfactorios (Figura 6), alcanzándose un VAF del 99.99 % para la velocidad del rotor, del 99.62 % para el par electromagnético y del 93.58 % para el *pitch*. Cabe destacar que, al estar activo el módulo de *Fore-Aft Damping*, se observan pequeñas variaciones en el ángulo de *pitch* incluso en esta región de operación. Este comportamiento introduce amortiguamiento adicional en la dinámica de la torre, reduciendo la amplitud de las oscilaciones en dirección *fore-aft* y contribuyendo a la disminución de las cargas estructurales, si bien incrementa la actividad del actuador de *pitch*.

En cuanto a la región nominal (Figura 7) se obtienen valores de VAF superiores al 85 % en el par electromagnético, *pitch* y velocidad angular. En la Tabla 1 se resume el índice VAF obtenido en cada simulación.

Tabla 1: Resultados VAF (%)

Variable	Nominal Turb.	Nominal Uniforme
	VAF (%)	VAF (%)
T_g (kN·m)	86.72 %	89.98 %
Ω (rpm)	96.39 %	94.44 %
β (°)	85.05 %	96.37 %

Variable	Parcial Turb.	Parcial Uniforme
	VAF (%)	VAF (%)
T_g (kN·m)	99.62 %	99.98 %
Ω (rpm)	99.99 %	99.99 %
β (°)	93.58 %	98.46 %

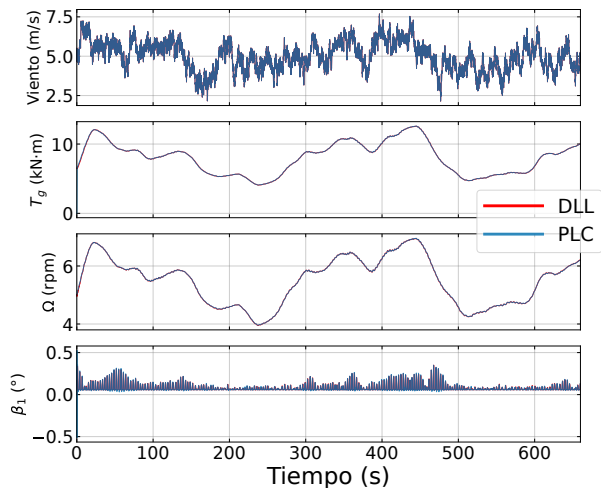


Figura 6: Comparación de las respuestas temporales en la zona parcial con perfil de viento turbulento.

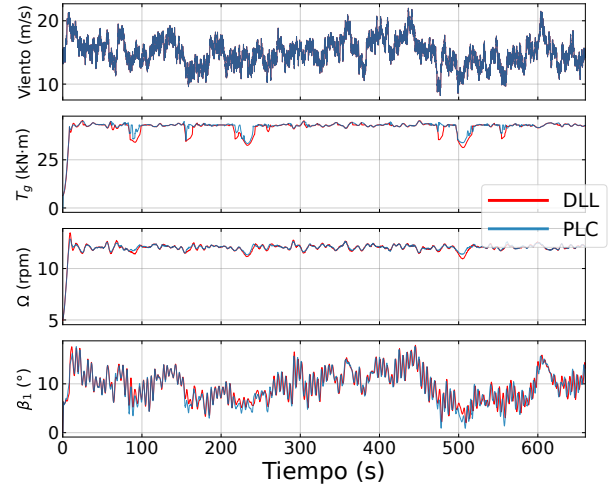


Figura 7: Comparación de las respuestas temporales en la zona nominal con perfil de viento turbulento.

4.3. Desempeño del módulo del IPC

En esta subsección se analiza el efecto del control IPC sobre las cargas en la base de las palas. Para cuantificar el daño estructural, se calcula la carga de daño equivalente (*Damage Equivalent Load*, DEL) en dicha región, dado que esta magnitud permite condensar el efecto de cargas cíclicas variables en un único indicador representativo de la fatiga. El análisis se ha llevado a cabo mediante la librería pCrunch de Python, basada en *MLife*, que implementa técnicas de conteo de ciclos y modelos de acumulación de daño ampliamente utilizados en el ámbito de la energía eólica. Los resultados obtenidos se presentan en la Tabla 2, donde se obtienen valores de DEL similares entre el controlador original y el traducido.

Se observa una reducción significativa en la componente M_y , también denominada momento *flapwise* de la pala, que constituye el principal objetivo de mitigación del módulo IPC, al ser la contribución dominante en la fatiga de las palas. Esta reducción es coherente con el principio de funcionamiento del IPC. Por el contrario, el efecto sobre las otras componentes de momento resulta más limitado, debido a la menor sensibilidad de dichas direcciones a la acción del control y a los acoplamientos aerodinámicos y estructurales del sistema.

Tabla 2: Error relativo DEL en la base de las palas entre controladores (DLL vs PLC).

Variable	DLL (kN·m)	PLC (kN·m)	Error (%)
DEL con IPC			
M_y	5030.63	5247.30	4.30 %
M_x	6390.38	6416.12	0.40 %
M_z	93.09	94.02	0.99 %
DEL sin IPC			
M_y	6232.74	6312.30	1.28 %
M_x	6533.50	6545.14	0.18 %
M_z	89.48	88.93	-0.61 %

A modo de ejemplo, las Figuras 8 y 9 muestran la evolución de los momentos en la base de las palas del controlador original y el traducido a código SCL. Se presenta un intervalo

de tiempo de 100 s con el fin de ampliar el nivel de detalle. Los perfiles de viento turbulento y determinista empleados son los mismos que en simulaciones anteriores, correspondientes a la zona nominal de operación.

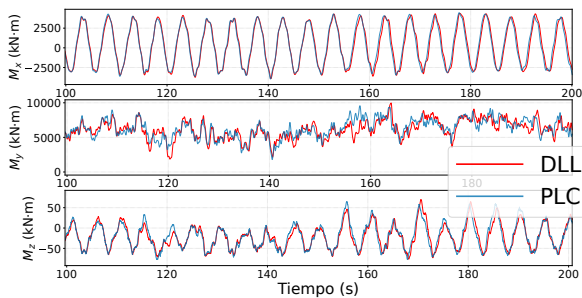


Figura 8: Momentos en la base de las palas con viento turbulento en zona nominal.

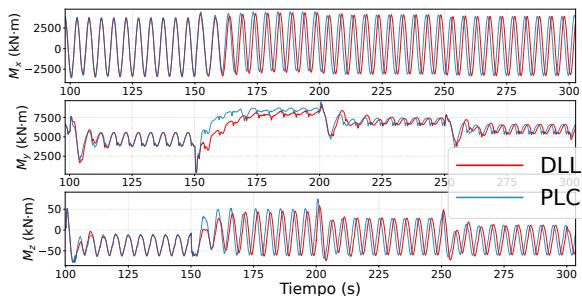


Figura 9: Momentos de las palas con viento uniforme en zona nominal.

5. Conclusiones

En este trabajo se ha presentado un esquema de simulación SIL basado en la implementación del controlador ROSCO en código SCL, compatible con equipos PLC de Siemens. Se ha llevado a cabo una validación preliminar mediante perfiles de viento tanto deterministas como turbulentos, evaluando la correspondencia entre las señales de control del controlador original y las del código implementado en SCL mediante el índice VAF. Los resultados obtenidos muestran una alta similitud entre ambas implementaciones.

Las posibilidades de la metodología SIL propuesta son amplias y ofrecen ventajas significativas, entre las que destacan la validación de algoritmos de control de cara a su posterior implementación en campo, así como el desarrollo de sistemas SCADA orientados al entrenamiento y capacitación de operarios.

Como aspecto relevante del controlador, se ha analizado el efecto del módulo de *Individual Pitch Control* (IPC) sobre la fatiga de las palas, para lo cual se ha empleado el indicador DEL, observándose valores comparables entre el controlador original y su implementación en SCL. No obstante, persisten ciertas discrepancias entre el controlador original y el traducido que requieren un análisis más detallado.

El trabajo desarrollado será de utilidad para la evaluación de estrategias de control avanzadas desarrolladas por el equipo investigador. Como línea futura, se plantea la validación mediante un controlador real utilizando el paradigma *Hardware-in-the-Loop* (HIL).

Agradecimientos

Este trabajo ha sido financiado por el proyecto PID2023-149181OB-I00/AEI/10.13039/501100011033/FEDER, UE, del Ministerio de Ciencia, Innovación y Universidades.

Referencias

- Abbas, N. J., Zalkind, D. S., Pao, L., Wright, A., 2022. A reference open-source controller for fixed and floating offshore wind turbines. *Wind Energy Science* 7 (1), 53–73.
URL: <https://wes.copernicus.org/articles/7/53/2022/>
DOI: 10.5194/wes-7-53-2022
- Bossanyi, E. A., 2003. Individual blade pitch control for load reduction. *Wind Energy* 6 (2), 119–128.
DOI: 10.1002/we.76
- Garrido Jurado, J., Lara Ortiz, M., Vázquez Serrano, F., Ruz Ruiz, M. L., 2026. Limitación del esfuerzo de los actuadores en el control de paso individual de turbinas eólicas. *Revista Iberoamericana de Automática e Informática Industrial*.
URL: <https://doi.org/10.4995/riai.2026.25395>
DOI: 10.4995/riai.2026.25395
- Jonkman, J., Butterfield, S., Musial, W., Scott, G., 2009. Definition of a 5-MW reference wind turbine for offshore system development. Technical report, National Renewable Energy Laboratory, Golden, CO, USA.
- Kenko, D., Gambier, A., Sep. 2022. Real-time wind turbine simulation for pitch control purposes by using a hardware-in-the-loop approach. In: *Proceedings of the 21st International Conference on Modelling and Applied Simulation (MAS)*. Rome, Italy, p. 026, 19th International Multidisciplinary Modeling & Simulation Multiconference.
DOI: 10.46354/i3m.2022.mas.026
- Lara, M., Ruz, M., Vázquez, F., Garrido, J., 2025. Enfoque anticipativo de control de paso individual en turbinas eólicas. *Jornadas de Automática* 46.
URL: <https://doi.org/10.17979/ja-cea.2025.46.12058>
DOI: 10.17979/ja-cea.2025.46.12058
- Martínez-Inchusta, A., Sierra-García, J. E., Santos, M., Leija, L., 2023. PIL environment for validation of wind turbine controllers based on IEC-61131. In: *XLIV Jornadas de Automática*. pp. 807–812.
DOI: 10.17979/spudc.9788497498609.807
- Mulders, S. P., van Wingerden, J.-W., 2019. On the importance of the azimuth offset in a combined 1P and 2P SISO IPC implementation for wind turbine fatigue load reductions. In: *2019 American Control Conference (ACC)*. pp. 3506–3511.
DOI: 10.23919/ACC.2019.8814829
- Muñoz-Palomeque, E., Sierra-García, J. E., Santos, M., 2024. Técnicas de control inteligente para el seguimiento del punto de máxima potencia en turbinas eólicas. *Revista Iberoamericana de Automática e Informática Industrial* 21 (3), 193–204.
DOI: 10.4995/RIAI.2024.21097
- NREL, 2026a. OpenFAST documentation. <https://openfast.readthedocs.io/en/main/>, accessed: 2026-06-03.
- NREL, 2026b. OpenFAST reference 5-MW turbine repository. <https://github.com/OpenFAST/r-test/tree/main/glue-codes/openfast>, accessed: 2026-06-03.
- NREL, 2026c. ROSCO: Reference open source controller. <https://rosco.readthedocs.io/en/latest/>, accessed: 2026-02-07.
- Puleva, T., Rouzhekov, G., Slavov, T., Rakov, B., 2016. Hardware-in-the-loop (HIL) simulation of wind turbine power control. In: *Mediterranean Conference on Power Generation, Transmission, Distribution and Energy Conversion (MedPower 2016)*. Belgrade, pp. 1–8.
DOI: 10.1049/cp.2016.1053