XLIV Jornadas de Automática 2023

https://jautomatica.es

Detección de fallos en aerogeneradores flotantes mediante redes neuronales usando OpenFAST.

Galeote, I.[a,*], Aimara, G.A.[a], Esteban, S.[a], Santos, M.[b]

[a] *Dto. Arquitectura de Computadores y Automática, Facultad de Físicas, Universidad Complutense de Madrid, Plaza de Ciencias, 1, 28040 Madrid, España*
[b] *Instituto de Ingeniería del Conocimiento, Universidad Complutense de Madrid, 28040 Madrid, España*

To cite this article: Galeote, I., Aimara, G.A, Esteban, S., Santos, M., 2023. Fault detection in floating wind turbines by means of neural networks using FAST.
XLIV Jornadas de Automática. 144-149. https://doi.org/10.17979/spudc.9788497498609.144

Resumen

Uno de los principales problemas de la energía eólica es el de la continuidad en la producción, exacerbado en el caso de la tecnología flotante dada la complejidad añadida de las circunstancias ambientales. Dada la variabilidad intrínseca del viento, que conlleva la producción irregular de energía, es de especial importancia detectar y minimizar tanto la frecuencia como la gravedad de fallos o averías en las máquinas. En el presente trabajo se ha estudiado una turbina de referencia flotante en alta mar de 5 MW y se han simulado fallas de diversos elementos estructurales mediante el software OpenFAST del NREL, mediante una técnica de acople de simulaciones. Después, se ha entrenado una red neuronal empleando MATLAB con el objetivo de identificar aquellos sensores más adecuados para detectar estas anomalías, así como su respuesta característica que permita hacer un diagnóstico rápido y fiable del fallo.

Palabras clave: Detección y diagnóstico de fallos, Redes neuronales, Turbinas eólicas flotantes, Control basado en el conocimiento, Simulación de energías renovables.

Fault detection in floating wind turbines by means of neural networks using FAST.

Abstract

One of the main problems of wind energy is that of production continuity, exacerbated in the case of floating devides due to the added complexity of the environmental loads. Given the intrinsic variability of wind, which leads to irregularities in energy production, it is of particular importance to detect and minimize both the frequency and severity of machine failures or malfunctions. In this work, a 5 MW offshore floating reference turbine has been studied; and failures of various structural elements have been simulated using NREL's OpenFAST software, using simulation coupling techniques. Then, a neural network has been trained using MATLAB, with the aim of identifying the most suitable sensors to detect these anomalies, as well as their characteristic response that allows a fast and reliable diagnosis of the failure.

Keywords: Fault detection and diagnosis, Neural networks, Floating wind turbines, Knowledge-based control, Renewable Energy Simulation

1. Introducción

La energía eólica se encuentra firmemente en la lista de las energías renovables en creciente demanda, hallando en la tecnología flotante una clara avenida para la expansión tanto tecnológica como económica (Tomás-Rodríguez and Santos, 2019).

El acceso a vientos más fuertes y estables, junto con la flexibilidad de ubicación y las menores restricciones ambientales, que conjuntamente permiten la construcción de aerogeneradores de gran potencia y tamaño, convierten a la eólica flotante

*Autor para correspondencia: igaleote@ucm.es

en una opción ambiciosa, pero atractiva para el futuro de las renovables (García et al., 2023).

Por otro lado, operar y mantener las máquinas, tanto por su tamaño como por sus complejas condiciones ambientales de operación, se convierte en uno de los principales inconvenientes de esta tecnología.

En efecto, las turbinas pueden experimentar fallos que afectan doblemente a la producción, por un lado por el coste de reparación, y por otro lado la pérdida económica por parada de la máquina.

Además, el mayor tamaño de los aerogeneradores flotantes respecto a sus homólogos terrestres aumenta la frecuencia y gravedad de los fallos estructurales (Faulstich et al., 2011).

Por estos motivos, sólo el coste de mantenimiento puede llegar a alcanzar el 30 % del presupuesto de un proyecto de eólica flotante *offshore* (Dinwoodie et al., 2013).

Por otro lado, la literatura en torno al estudio de fallos estructurales en turbinas eólicas flotantes, como por ejemplo de las palas del rotor y de las *mooring lines*, es escasa. Los trabajos publicados en el área se centran principalmente en fallos en el control de *pitch*, la caja de engranajes, los rodamientos y el sistema eléctrico (Odgaard et al., 2009).

El empleo de redes neuronales para la detección de fallos en aerogeneradores ha sido explorado en (Cho et al., 2021) y (Yang et al., 2022), mientras que su aplicación en escenarios *offshore* es más escasa, destacando el trabajo de (Shaheen and Németh, 2023), que se centra en concreto en la detección y monitorización de fallos de la caja de engranajes.

Este trabajo presenta un estudio de tres potenciales fallos graves en un aerogenerador en alta mar: una ruptura de una de las palas, un desenganche del sistema de anclaje al fondo marino y de un fallo en el actuador de *pitch* de una de las palas, simulados con el software OpenFAST del NREL, que resulta ser la mejor opción por detrás del uso de datos reales, normalmente privados en tecnologías emergentes como la eólica offshore (Pandit et al., 2023).

Se analizan los principales sensores que podrían detectar rápidamente estas anomalías, así como las perturbaciones respecto al comportamiento normal que se observarían en los grados de libertad, concretamente en el *surge* y *yaw* de la plataforma, en las oscilaciones en la dirección *fore-aft* de la góndola, en la velocidad de giro del rotor y en el ángulo de *pitch* de cada una de las palas determinado por el control.

A continuación, se entrena una red neuronal tipo *perceptron* para identificar dichos fallos de forma rápida, y validamos esta evaluación con datos de OpenFAST.

El artículo se organiza en secciones que cubren el diseño de la simulación y del método de fallo, el entrenamiento de la red neuronal y una discusión de los resultados, presentados en ese orden.

2. Simulación

Actualmente no existe ningún *benchmark* que incluya escenarios de fallo de las palas del rotor o en las *mooring lines*, ni son directamente implementables en OpenFAST, ampliamente usado en la industria eólica como un software eficiente para computar las múltiples cargas no lineales presentes en los aerogeneradores.

Para solventar esto, la metodología novedosa empleada en este estudio consiste en realizar una simulación desde el estado estacionario, alterar manualmente las propiedades estructurales del aerogenerador para simular cada uno de los fallos, y continuar la simulación con la máquina defectuosa, empalmando las condiciones finales de la primera simulación con las condiciones iniciales de la segunda.

2.1. *Condiciones de simulación*

En el presente trabajo se ha estudiado una turbina estándar de la industria: la *5-MW Reference Wind Turbine for Offshore System Development* del NREL (Jonkman et al., 2009) con una base tipo *barge* de ITI Energy, tal y como se define en (Jonkman, 2007).

Implementada en OpenFAST, las condiciones ambientales escogidas han sido las de mar en calma y viento longitudinal uniforme de 18 m/s, en torno a los valores de producción nominal de potencia del aerogenerador (Figura 1).

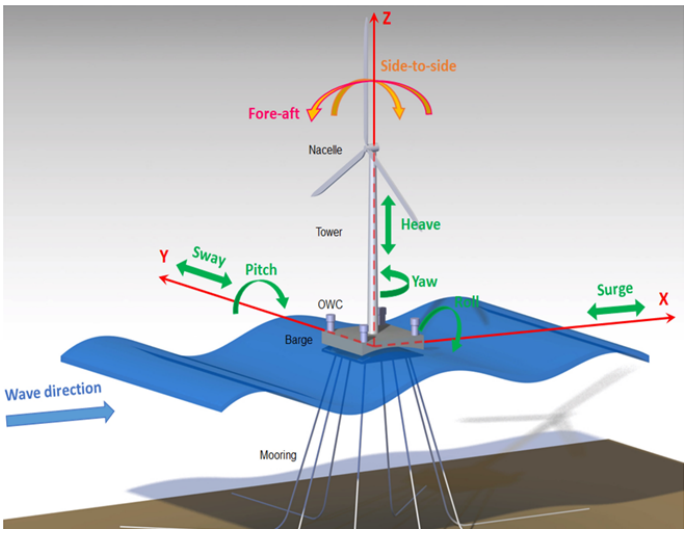

Figura 1: Diagrama de la turbina y barcaza empleada en el estudio.

Inicialmente, el sistema produce potencia a su valor nominal, con un control colectivo de *pitch* de velocidad variable, y sin control de torque.

Con el fin de obtener correctamente la identificación de los fallos del aerogenerador, se ha decidido trabajar en la tercera región de operación de un aerogenerador. Es por ello por lo que el control de torque no está presente en las simulaciones realizadas.

No obstante, sería interesante a modo de trabajo futuro, adecuar las señales para poder identificar fallos desde las primeras regiones de funcionamiento, ya que existen fallos propios de cada región de operativa.

La simulación completa se compone de 1000 segundos de simulación por cada señal, no obstante, se ha modificado el instante de tiempo en el cual se produce cada uno de los fallos implementados en el aerogenerador flotante. En concreto se han probado con instantes de fallos en 300, 500 y 700 segundos.

2.2. *Diseño de fallos*

Los tres fallos se han pasado por parámetros desde MATLAB a OpenFAST en el límite entre simulaciones, siguiendo los criterios:

- La ruptura de pala se recrea mediante una disminución de la masa de la pala a partir de cierta longitud a valores pequeños, pero no nulos para evitar problemas de convergencia.
- El desenganche de la línea de anclaje se recrea mediante una elongación de la *mooring line* de la que se quiere analizar su posible malfuncionamiento.
- El fallo en el actuador de *pitch* se recrea mediante la inserción de una señal de ruido blanco que se adhiere a la señal de referencia. Se implementa en Simulink directamente en la señal de *pitch*.

2.3. *Resultados de la simulación*

Una vez realizadas las simulaciones, se filtran las señales entregadas por el aerogenerador flotante para minimizar discordancias a la hora de acoplar simulaciones entre sí mediante un filtro de media móvil.

Por otro lado, se ha realizado una normalización de las señales respecto al valor pico de la señal, ya que es adecuado a la hora de entrenar la red neuronal para evitar problemas de *overfitting*. Es decir, cuando el sistema está sobre-ajustado por exceso de datos y empieza a aprender del ruido de las señales. Esto conlleva al sistema a comportarse adecuadamente con los datos de entrenamiento, mientras que no es capaz de generalizar a nuevos datos.

Se han analizado las respuestas en las siete señales de salida mencionadas en la sección anterior, pero por brevedad analizaremos tres que resultan interesantes: las revoluciones por minuto del rotor (Figura 2), el ángulo de *pitch* de una de las palas (Figura 3) y las oscilaciones en la dirección *fore-aft* de la góndola (Figura 4).

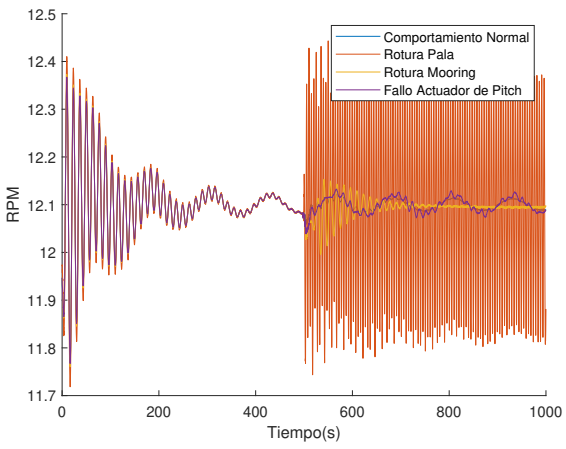

Figura 2: Comparación entre el comportamiento nominal de la velocidad angular del rotor frente a los diferentes fallos.

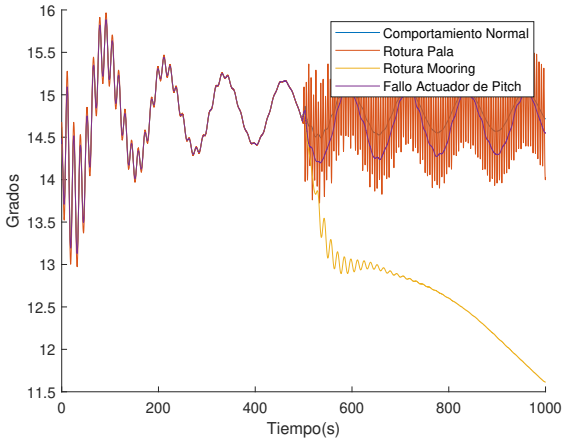

Figura 3: Comparación entre el comportamiento nominal del *pitch* de una de las dos palas intactas frente a los diferentes fallos.

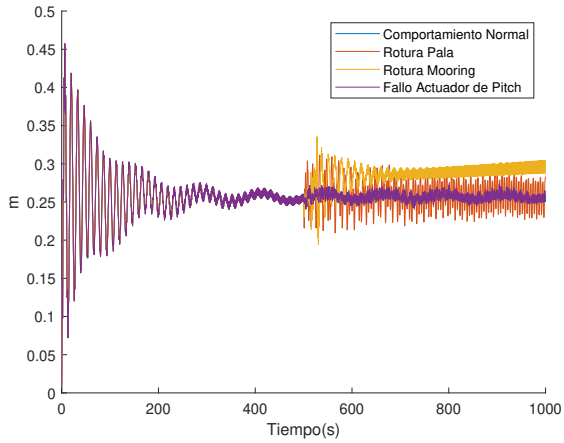

Figura 4: Comparación entre el comportamiento nominal de las oscilaciones de la góndola en la dirección *fore-aft* frente a los diferentes fallos.

Atendiendo a las señales de salida del aerogenerador, extraemos las siguientes conclusiones acerca de cada fallo:

- La rotura de pala lleva a un movimiento de precesión en el rotor, que se observa en una envolvente periódica en la señal de *pitch* que altera la velocidad de giro a una alta frecuencia. Fruto de esta descompensación, las oscilaciones de la góndola aumentan de forma simétrica a las nominales.
- La rotura del mooring supone alejarse de la fuente de viento, por lo que el actuador de *pitch* compensa adecuadamente disminuyendo el *pitch* para mantener las revoluciones del rotor. Por otro lado, las oscilaciones de la góndola aumentan progresivamente debido a la falta de fuerza restauradora que las amortigue desde la base.
- El fallo en el actuador de pitch lleva, como era de esperar, a la no estabilización del actuador de la pala intacta, al intentar compensar la señal defectuosa que envía la otra pala, generando el aumento de las oscilaciones *fore-aft*.

En cuanto a la velocidad del rotor, genera una componente extra de frecuencia rápida sobre la oscilación lenta de comportamiento nominal.

A continuación, se ha procedido a entrenar la red neuronal para la detección de estos fallos.

3. Red neuronal

Una red neuronal se puede definir como un conjunto de agentes conectados entre sí, y cuya finalidad es realizar un procesamiento de la información que llega a su entrada y enviarla por sus conexiones de salida (Figura 5). Existen reglas que gobiernan la dinámica del envío de información entre las unidades, también conocidos como pesos, los cuales permiten aportar aprendizaje o adaptación al sistema (Schlechtingen and Santos, 2011).

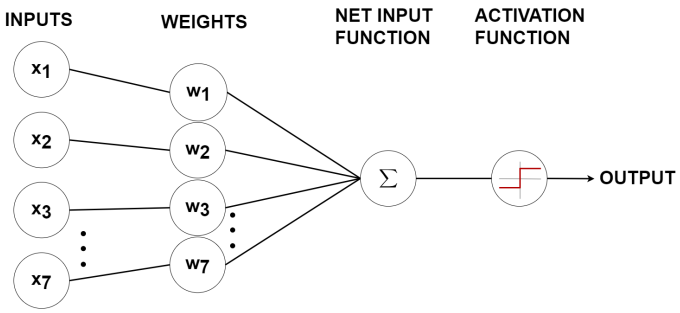

Figura 5: Esquema de un *perceptron* multicapa.

El propósito de una red neuronal abarca desde reconocimiento de patrones, procesamiento de señales, optimización, la robótica, control de procesos, predicción, monitorización e identificación.

Atendiendo a las prestaciones de las señales procesadas se diseña una red neuronal con quince neuronas en la capa oculta. Como entradas de la red, se han tomado las ya mencionadas siete salidas del aerogenerador flotante, en donde se observan con mayor claridad los efectos de los fallos producidos. La red es entrenada a partir de los datos obtenidos empleando el aprendizaje supervisado por el experto.

Por cada fallo se han incorporado tres ejemplos cambiando el tiempo de aparición del fallo. Al mismo tiempo, se han utilizado 160004 muestras en total para entrenar la red neuronal.

Por defecto, MATLAB emplea la función *dividerand*, que divide aleatoriamente las muestras en conjuntos de entrenamiento, validación y prueba en una proporción de 70/15/15. Es por ello por lo que se han llegado a utilizar en una primera instancia, 112003 muestras para el entrenamiento, 24001 para la validación y 24000 para la prueba.

Los elementos de la red utilizan un mecanismo de aprendizaje basado en el uso de funciones de criterio que permitan ponderar los valores de salida.

La función de activación (1) utilizada en la capa oculta de la red neuronal es la función tangente hiperbólica *tansig*. Esta función toma valores en el rango de -1 a 1 y puede modelar relaciones no lineales entre las entradas y las salidas.

La función de activación utilizada en la capa de salida es la función lineal *purelin*. Esta función no transforma la entrada y se utiliza para producir una salida continua que puede tomar cualquier valor real.

$$a = tansig(N) = \frac{2}{1 + e^{-2N}} - 1 \tag{1}$$

La función de salida, en función de la entrada x_i y el peso w_i, devuelve un 1 si la función de activación es positiva o 0 en caso contrario. El límite binario de salida es conocido como superficie de decisión.

De este modo, se aplica un aprendizaje a partir de ejemplo y corrección de error entregando las reglas necesarias a cada *perceptron*, y asignando una salida con valor discreto conocido en función del fallo provocado intencionalmente.

Se conoce como *perceptron* a cada una de las capas de neuronas de salida, las cuales están conectadas a todas y cada una de las entradas mediante un conjunto de pesos ajustables.

Las entradas se combinan linealmente utilizando los pesos y devolviendo el resultado a través de una función de activación, con el fin de producir una salida esperada en la red.

$$J(w) = \sum_{x \in X} |w.x| \tag{2}$$

La anterior ecuación 2 denominada función de criterio del *perceptron* corresponde a la suma de las distancias de la entradas mal clasificadas a la superficie de decisión. Se busca minimizar $J(w) = 0$, lo que supone una dirección del gradiente negativo. De forma iterativa, se realiza la Modificación de Pesos Sinápticos 3.

$$w_{t+1} = w_t + \rho - grad(J) \tag{3}$$

Donde ρ representa la velocidad de aprendizaje de la red. Como se puede observar en la siguiente Figura 6, la respuesta obtenida de la red neuronal en el entrenamiento da como resultado un correcto ajuste del modelo en la regresión lineal entre los datos de salida y los esperados.

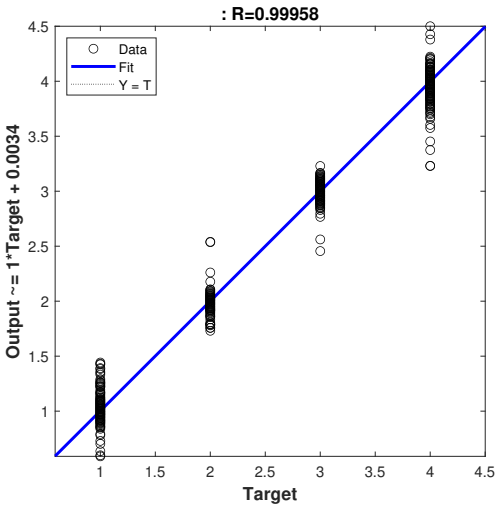

Figura 6: Regresión lineal de los *targets* relativos a las salidas. Cuanto más próxima a 1 es la pendiente, mejor es relación entre los conjuntos de datos.

La red empleada en el presente trabajo corresponde a una red *feedforward* de ajuste de funciones multicapa. Ambas capas tienen conexiones de sesgo. El número de capas, elementos de peso y funciones elegidas están diseñados para lograr un buen ajuste a los datos. Algunos de los parámetros más importantes de la red son los siguientes:

- La función de adaptación, permite a la red neuronal aprender mientras está en uso continuo, ajustando sus pesos y sesgos en tiempo real a medida que recibe nuevos datos en cada entrenamiento.
- La función de derivación se emplea para calcular el gradiente y la actualización de los pesos.
- La función de división segmenta aleatoriamente los datos de entrada en conjuntos de entrenamiento, de validación y de prueba.
- Los parámetros de relación de entrenamiento, de valor y de testeo determinan la proporción de datos asignados a cada conjunto.
- La función de rendimiento (mse) calcula el error cuadrático medio entre las salidas deseadas y las salidas reales de la red. Dicha función utiliza el algoritmo Levenberg-Marquardt 4 para entrenar la red. Este algoritmo es un método iterativo que busca minimizar el error cuadrático.

$$x_{k+1} = x_k - [J^T J + \mu I]^{-1} J^T e \tag{4}$$

  Donde J es la matriz jacobiana construida a partir de las primeras derivadas de los errores de la red relacionados con los pesos y las restricciones. Por otro lado, e representa el vector de errores de la red.

  El valor del escalar μ va disminuyendo a medida que mejora el resultado con cada iteración del algoritmo. Esto es debido a que la anterior ecuación se asemeja al método de Newton utilizando la matriz hessiana aproximada cuando μ tiene un valor cercano a cero, siendo más preciso y rápido con un margen de error mínimo.

4. Resultados de la red neuronal

A continuación, se ha comprobado el funcionamiento de la red neuronal para cada uno de los ejemplos entregados por la simulación. Se debe recordar la motivación principal para el desarrollo y diseño de la red neuronal, y es la identificación en tiempo real del estado de un aerogenerador flotante.

En la Figura 7 se representan las salidas de la red neuronal para cada caso conocido de fallo y para un comportamiento normal.

Para una mejor observabilidad, se ha decidido establecer estados discretos en función del tipo de fallo a detectar y ajustar el instante de tiempo en el que se producen los fallos a los 500 segundos de simulación.

Para un comportamiento sin fallos se ha aplicado un estado de aerogenerador igual a 1, mientras que para los fallos de rotura de pala, rotura de *mooring* y fallo en el actuador de *pitch* corresponden a los niveles 2, 3 y 4, respectivamente.

Se observa una correcta identificación de cada uno de los fallos, partiendo de un estado de funcionamiento normal que se mantiene en cada uno de los casos hasta alcanzar el instante de tiempo donde se han provocado los fallos en el aerogenerador.

La red es capaz de devolver una señal muy próxima al deseado, consiguiendo una identificación en tiempo real del estado del aerogenerador bastante eficaz.

Al mismo tiempo, se puede apreciar un cierto ruido en cada una de la salidas debido a la necesidad de aplicar un entrenamiento más exhaustivo donde se incorporen nuevos ejemplos de señales de entrada y objetivos que ayuden a generar un diseño de red neuronal mucho más robusto.

Del mismo modo, se puede apreciar una alteración en la dinámica del comportamiento normal del aerogenerador flotante en la intersección de las dos simulaciones. Esto es debido a que el software OpenFast no incluye de forma predeterminada la posibilidad de modificar los parámetros del aerogenerador en medio de la simulación, tal y como se ha explicado anteriormente.

5. Conclusiones y trabajos futuros

En el presente trabajo se ha puesto de manifiesto la utilidad de emplear redes neuronales para la detección de fallos en aerogeneradores, pues no es necesario especificar el valor de cada entrada para lograr una identificación satisfactoria del fallo.

También se nota la falta de referencias y estudios de averías y fallos, especialmente estructurales, de aerogeneradores flotantes. Esto se puede deber a la juventud de la tecnología, y a la falta de datos públicos de emplazamientos reales (Pandit et al., 2023). En consecuencia, los *softwares* de simulación de aerogeneradores no poseen la habilidad para recrear fielmente estas situaciones de fallo, por lo que es necesario hacer un compromiso con las herramientas disponibles.

Los próximos pasos de este trabajo consisten en un entrenamiento más fino que permita una mejor respuesta de la red neuronal. Del mismo modo, cabría la posibilidad de incorporar el espectro en frecuencia de las señales para mejorar el entrenamiento de la red.

Por otro lado, aumentar el número de casos de entrenamiento en cuanto a gravedad del fallo y momento temporal en el que ocurre mejoraría los resultados.

Del mismo modo, sería interesante incorporar a la detección de fallos casos en los que estén presentes las demás regiones de funcionamiento del aerogenerador, así como el control de torque. También cabe la posibilidad de comprobar la eficacia de otros tipos de redes neuronales basadas en la memoria en corto y largo plazo como LSTM, incluyendo etapas de retroalimentación en el entrenamiento.

Se puede completar el entrenamiento con un mayor número de muestras de variantes en los fallos del aerogenerador, como por ejemplo, probando distintas alturas a la hora de diseñar la ruptura de una de las palas.

Otra avenida de expansión de este estudio es el ampliar el número y la naturaleza de fallos a estudiar, incluyendo averías de origen eléctrico, frecuentes en los homólogos terrestres.

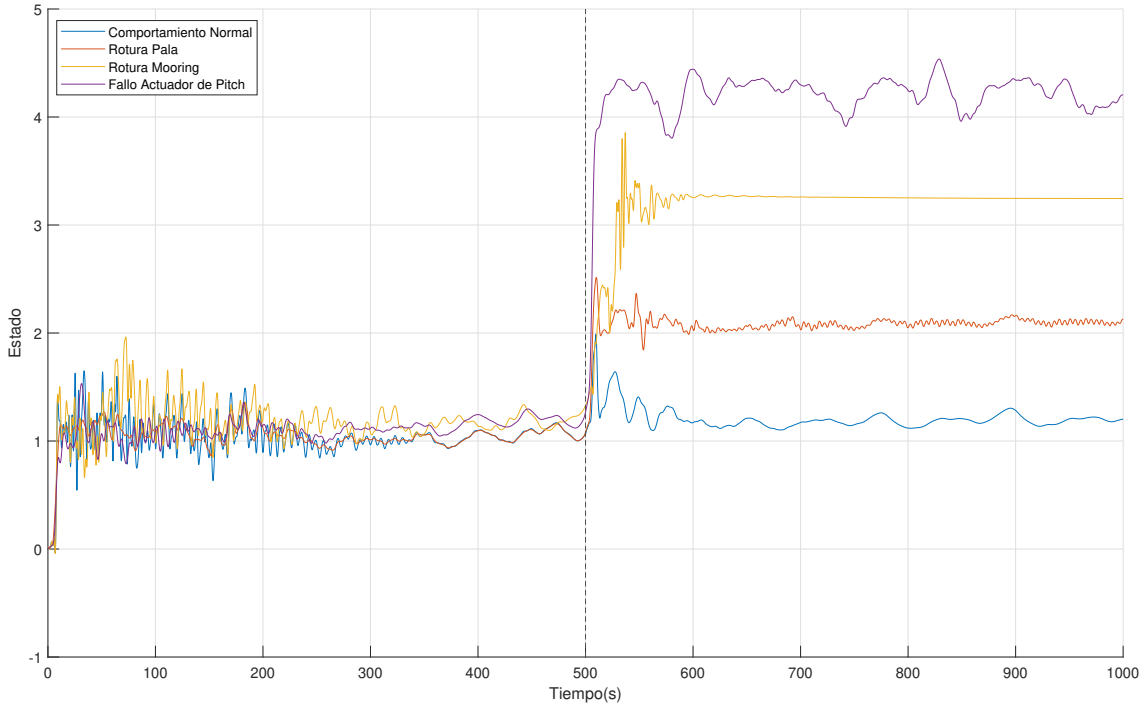

Figura 7: Respuesta de salida de la red neuronal para cada uno de los casos. La línea punteada indica el momento en el que se producen los fallos.

Agradecimientos

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

Referencias

Cho, S., Choi, M., Gao, Z., Moan, T., 2021. Fault detection and diagnosis of a blade pitch system in a floating wind turbine based on kalman filters and artificial neural networks. Renewable Energy 169, 1–13.

Dinwoodie, I., McMillan, D., Revie, M., Lazakis, I., Dalgic, Y., 2013. Development of a combined operational and strategic decision support model for offshore wind. Energy Procedia 35, 157–166.

Faulstich, S., Hahn, B., Tavner, P. J., 2011. Wind turbine downtime and its importance for offshore deployment. Wind Energy 14 (3), 327–337.

García, I. G., San Román, S. E., Peñas, M. S., 2023. Identificación preliminar de la dinámica de la barcaza de una turbina flotante. In: WWME 2022 IV. Jardunaldia-Berrikuntza eta irakaskuntza energia berriztagarrien aurrerapenetan. Servicio Editorial= Argitalpen Zerbitzua, pp. 25–30.

Jonkman, J., Butterfield, S., Musial, W., Scott, G., 2009. Definition of a 5-mw reference wind turbine for offshore system development. Tech. rep., National Renewable Energy Lab.(NREL), Golden, CO (United States).

Jonkman, J. M., 2007. Dynamics modeling and loads analysis of an offshore floating wind turbine. University of Colorado at Boulder.

Odgaard, P. F., Stoustrup, J., Kinnaert, M., 2009. Fault tolerant control of wind turbines–a benchmark model. IFAC Proceedings Volumes 42 (8), 155–160.

Pandit, R., Astolfi, D., Hong, J., Infield, D., Santos, M., 2023. Scada data for wind turbine data-driven condition/performance monitoring: A review on state-of-art, challenges and future trends. Wind Engineering 47 (2).

Schlechtingen, M., Santos, I. F., 2011. Comparative analysis of neural network and regression based condition monitoring approaches for wind turbine fault detection. Mechanical systems and signal processing 25 (5), 1849–1875.

Shaheen, B. W., Németh, I., 2023. Performance monitoring of wind turbines gearbox utilising artificial neural networks—steps toward successful implementation of predictive maintenance strategy. Processes 11 (1), 269.

Tomás-Rodríguez, M., Santos, M., 2019. Modelling and control of floating offshore wind turbines. Revista Iberoamericana de Automática e Informática Industrial 16 (4).

Yang, B., Cai, A., Lin, W., 2022. Analysis of early fault vibration detection and analysis of offshore wind power transmission based on deep neural network. Connection Science 34 (1), 1005–1017.