

# SISTEMA DE DETECCIÓN DE ALARMAS DE FALLOS EN EL TREN MECÁNICO DE UN AEROGENERADOR

David López de la Cruz  
[dlopez060@ikasle.ehu.es](mailto:dlopez060@ikasle.ehu.es)

Luis Pantoja Álvarez  
[luis.pantoja@ingetteam.com](mailto:luis.pantoja@ingetteam.com)

Eloy Irigoyen Gordo  
[eloy.irigoyen@ehu.es](mailto:eloy.irigoyen@ehu.es)

## Resumen

*Este trabajo presenta un sistema automático de detección de alarmas en el tren mecánico de un aerogenerador. Este sistema se basa en un gemelo digital, como un modelo físico simulado del conjunto de estudio (sistema de transmisión), para entrenar con datos simulados al sistema. A esto se une un modelo de detección entrenado basado en técnicas de clasificación, que, a partir de datos reales, devuelve las alarmas de fallos del dispositivo. De esta forma, se maximiza la cobertura ante fallos, minimizando falsos positivos y negativos del sistema. El trabajo abarca fallos en un rodamiento, con un procedimiento extrapolable al tren mecánico.*

**Palabras clave:** Gemelo digital, Análisis modal de fallos y efectos, Modelo de detección, Cobertura y Aerogenerador.

## 1. INTRODUCCIÓN

En el aerogenerador se busca maximizar la energía obtenida, aumentando la potencia extraíble del viento. El diámetro del rotor, la altura de la torre y el emplazamiento influyen en ello [1], pero aumentan la complejidad, coste y duración del mantenimiento. Los fallos en sus componentes son la causa principal de mantenimiento, siendo los elementos más críticos los del tren de potencia, en el interior de la góndola (W. Dong et al). Por otro lado, actualmente este sector usa un mantenimiento preventivo apoyado, en algunos casos, por un análisis manual asistido. Este análisis recae en un analista de vibraciones que, por experiencia, diagnostica fallos futuros. Este método requiere conocimiento y experiencia, añadiendo error humano y apreciación individual. A su vez, este mantenimiento, al evitar que llegue el límite de su vida útil, produce pocos datos históricos sobre fallos.

Este trabajo presenta un método basado en la detección automática de alarmas de fallo de los componentes del tren de potencia de un

aerogenerador. Se busca automatizar el proceso, con una detección temprana de fallos sin intervención manual, a partir de un gemelo digital (GD) que simule el comportamiento del componente, y un modelo que deduzca, una vez entrenado con casos simulados, las alarmas. Así, se optimiza el mantenimiento del aerogenerador, lo que, debido a sus características, supone menos viajes y menor coste, disminuyendo el tiempo de riesgo de los operarios. Además, el detallar los fallos influye en la mejora del producto, desarrollando mejores rediseños.

Este objetivo se divide en (figura 1): *preprocesado*, capacidad de filtrado e interpolación. Obtiene datos de sensores (acelerómetros) y devuelve las variables que definan sus fallos, de forma limpia y directa; *gemelo digital*, semejanza con el sistema real. Simula el comportamiento del rodamiento con un modelo físico, incluyendo fallos; *modelo de detección*, sistema inteligente desarrollado para asociar entradas y salidas. Trata datos preprocesados para estimar alarmas de fallo, en base a un entrenamiento con datos de simulación; y *postprocesado (histórico)*, organización de los resultados. Ordena, muestra alarmas y almacena datos.

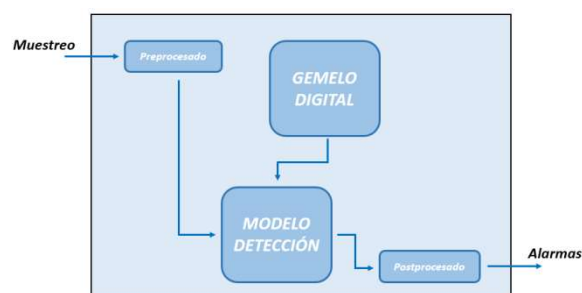


Figura 1: IDEF0 del trabajo

El sistema de la figura se diseña para detectar cuatro posibles estados del rodamiento: Funcionamiento normal, fallo en la carrera externa, en la carrera interna y en uno de los elementos rodantes. A su vez, el diseño maximiza la información de fuentes físicas (datos

reales del sensor o del GD), minimizando métodos estadísticos (ajustados a una muestra de valores y desconocidos en casos no modelizados), además de poseer un carácter flexible y adaptativo, por su capacidad de sintonización. Con todo ello, se busca el mejor modelo de simulación, ya que sus datos sirven al método de clasificación a relacionar vibraciones y alarmas de fallo. Por ello, se eligen las variables que mejor definan dicho fallo, a través de un análisis modal de fallos y efectos (FMEA):

Tabla 1: FMEA de un rodamiento

| Tipo de fallo | Modo de detección         | Detección fallo |      |
|---------------|---------------------------|-----------------|------|
|               |                           | Ideal           | Real |
| Carrera ext.  | $x_p, y_p, y_b$           | 100%            | 33%  |
| Carrera int.  | $x_s, y_s, y_b$           | 100%            | 33%  |
| Elem. rodante | $x_p, y_p, x_s, y_s, y_b$ | 100%            | 20%  |
|               |                           | 100%            | 29%  |

De la tabla 1 se escogen como variables: los grados de libertad (GDL) de la carcasa/carrera externa [ $x_p, y_p$ ], los del eje/carrera interna [ $x_s, y_s$ ] y la vibración medida [ $y_b$ ] del sensor del rodamiento (a partir del modelo especificado en el punto 3.1.). De aquí se deduce la necesidad de variables inducidas para control de fallos (en el caso real, la cobertura disminuye hasta un 30%, suponiendo la misma relevancia para cada variable). Además, sabiendo que las frecuencias de defecto del rodamiento dependen de la velocidad del eje que apoya en éste, se toman como variables de estudio: *variables de contexto*,  $\omega_s$  (velocidad del eje); *variables medidas*,  $y_b$  (vibración en el acelerómetro); y *variables inducidas*:  $x_s, y_s, x_p$  e  $y_p$  (vibraciones en el eje/carrera interior y en la base/carrera exterior).

## 2. ESPECIFICACIONES DEL SISTEMA

La función del sistema presentado en este trabajo es la detección automática de alarmas de fallos en el tren mecánico de un aerogenerador, sin necesidad de intervención humana. Esta función engloba dos grandes bloques: *gemelo digital*, copia el comportamiento del elemento; y *modelo de detección*, sistema inteligente que detecta futuros fallos a través de datos actuales. Ambos bloques participan en distintas situaciones: *entrenamiento*, a partir de datos del GD se exporta un modelo de detección fiable; y *funcionamiento*, que engloba al gemelo digital, trabajando con variables inducidas simuladas, y al modelo exportado, capaz de detectar las alarmas automáticamente.

### 2.1. SUBSISTEMAS

El GD es una recreación del rodamiento real, para unas simulaciones virtuales exactas, para lo que se emplea

un modelado físico (no estadístico), verificado, a su vez, con señales de campo. Se usa la plataforma MATLAB© y herramientas de cálculo matemático de MATHWORKS© (*toolboxes*), trabajando con las variables del FMEA, a partir de datos de la máquina y del GD: medidas (sensores), inducidas (matemáticas) y de contexto (condiciones de trabajo). Este modelo simula fallos y observa las respuestas, realizando dos tipos de simulaciones: simulaciones de cada fallo para confirmar los valores y las variables, verificando el modelo físico; y combinaciones de fallos, datos para entrenar el modelo de detección.

El modelo de detección asocia datos de sensores a fallos, a partir de entrenamiento. Para simplificar el proceso, se usa como ejemplo una red neuronal artificial (RNA), pero, posteriormente, se busca la mejor técnica de clasificación. El entrenamiento introduce datos simulados del GD a la red, junto con sus resultados (alarmas) para que ésta sea capaz, posteriormente, de detectar esos fallos con datos similares. En una RNA, se ajusta el valor de los pesos de las conexiones entre neuronas de distintas capas (en otros casos se recogen relaciones, condiciones, etc.). Una vez entrenado, el modelo de detección es capaz de, a partir de datos de sensores, junto con variables inducidas del GD, definir alarmas de fallos.

### 2.2. SITUACIONES

En el entrenamiento (fig. 2) se provocan fallos en el gemelo digital. Estos datos se introducen en el modelo de detección, forzándolo a que, a partir de esas entradas, se obtengan como salidas los fallos asociados, de forma que la red “aprende”, creando un modelo de detección fiable. El GD recrea el funcionamiento del sistema a partir de datos de contexto. Esto, unido a una batería de fallos provocados, lleva a una lista de datos con la que entrenar un modelo de detección. En este boceto se usa una RNA como base, pero posteriormente se valora la mejor técnica de clasificación.

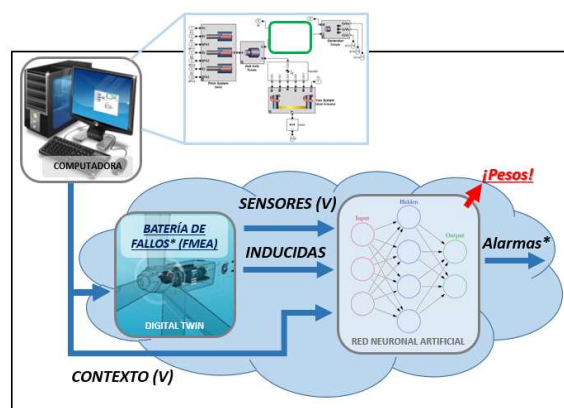


Figura 2: Resumen del entrenamiento

Es importante definir correctamente las variables de entrada (sensores e inducidas) a través del preprocesamiento, para que la asociación dato-alarma sea óptima. El *classification learner*, realiza pruebas para cada técnica de clasificación, devolviendo la mejor opción en relación a la cobertura de fallos.

Durante el funcionamiento (fig. 3), el GD solo se requiere para variables inducidas, que se llevan al modelo de detección junto con variables medidas y de contexto de la máquina real. El sistema de gestión de contenidos (CMS) indica las medidas de sensores, mientras que los resultados se visualizan en un interfaz humano-máquina (HMI). En esta situación se trabaja con un modelo ideal que puede no ajustarse a la máquina real, por lo que es necesario un estudio de si, el usar variables inducidas de un caso ideal, afecta a la detección de alarmas o no. En caso positivo, se introduciría un resintonizador del GD.

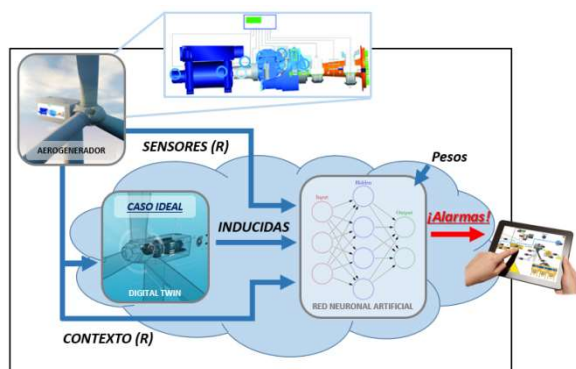


Figura 3: Resumen del funcionamiento

Los datos que entran al modelo se preprocesan, añadiendo filtrado de ruido e interpolación, por ser datos reales de sensores. Es necesario un estudio para comprobar cómo afectan las variables inducidas ideales en la resolución de alarmas. Puede ser necesaria una resintonización del GD a partir de variables medidas, tal que sus variables inducidas se ajusten más al caso realista. En este caso se entrena un segundo modelo de detección únicamente a partir de los datos de sensores. De esta forma, durante su funcionamiento, sus resultados (fallos a implementar en el GD) hacen trabajar con un modelo físico más realista. Un postprocesado final simplifica la visualización de los fallos.

### 3. DISEÑO DE ARQUITECTURA

#### 3.1. GEMELO DIGITAL

El éxito del trabajo radica en la exactitud con la que el GD simula el comportamiento del rodamiento. El “modelo dinámico de rodamientos con elementos rodantes” [4] deduce un modelo de vibraciones en el rodamiento, implementando averías localizadas. Este

trabajo toma como base el modelo de dos GDL de Fukata, incorporándole la no linealidad de rodamientos de elementos rodantes (contacto Hertziano), rigidez variable y holgura entre elementos rodantes y carreras del rodamiento (ver figura 4).

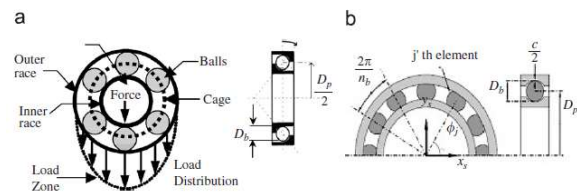


Figura 4: (a) Distribución carga, (b) Modelo 2 GDL

Esto, incluyendo fallos (carrera externa, carrera interna y elemento rodante) completa el modelo carcasa-rodamiento (fig. 5), con parámetros como:  $\delta_j$ , deformación general de contacto del elemento rodante  $j$ ;  $\gamma_j$ , estado de contacto de los elementos rodantes;  $\phi_j$ , posición angular del elemento rodante; y  $f_x, f_y$ , fuerzas de contacto en dirección  $x$  e  $y$ .

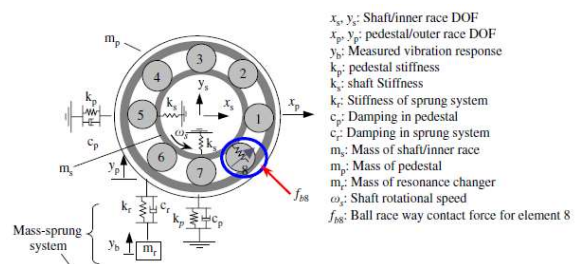


Figura 5: Modelo rodamiento-carcasa de 5 GDL

Estos parámetros, junto con las propiedades del rodamiento (diámetros, masas, etc.) conforman un sistema de ecuaciones diferenciales (ecuaciones 1-5), que describe el movimiento del sistema rotor-rodamiento, con cinco GDL. Al parámetro  $\delta_j$  le afecta directamente la variable  $\beta_j$ , interruptor de fallo para la pérdida de contacto en una posición angular. Esta, unida a características de la hendidura (profundidad, distancia angular, etc.) definen la región de fallo, existiendo, para cada avería, fórmulas de cálculo de  $\beta_j$  (fig. 6). Resolviendo el sistema de ecuaciones se obtienen las vibraciones (posición y velocidad).

$$m_s \ddot{x}_s + k_s x_s + f_x = 0 \tag{1}$$

$$m_s \ddot{y}_s + k_s y_s + f_y = 0 \tag{2}$$

$$m_p \ddot{x}_p + c_p \dot{x}_p + k_p x_p - f_x = 0 \tag{3}$$

$$m_p \ddot{y}_p + (c_p + c_r) \dot{y}_p + (k_p + k_r) y_p - k_r y_b - c_r \dot{y}_b - f_y = 0 \tag{4}$$

$$m_r y_b + k_r (y_b - y_p) + c_r (\dot{y}_b - \dot{y}_p) = 0 \tag{5}$$

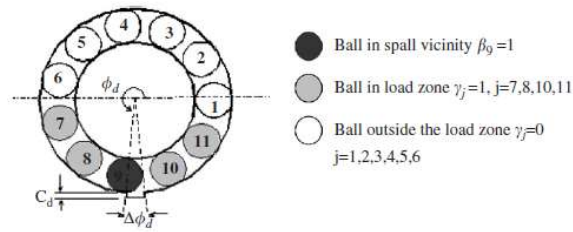


Figura 6: Hendidura en la carrera externa

### 3.2. PREPROCESAMIENTO

Los resultados del GD no son óptimos para definir el fallo, por lo que se procesan antes de introducirlo como entrenamiento al modelo de detección. El primer paso es trabajar sólo con la parte estacionaria, a partir de un filtro que elimine el transitorio. A su vez, se limita el número de puntos a una potencia de dos, simplificando el trabajo. Posteriormente, se usan un filtro paso alto y uno paso bajo (fig. 7): el primero disminuye las frecuencias por debajo de un nivel mientras que el segundo mitiga las más altas.

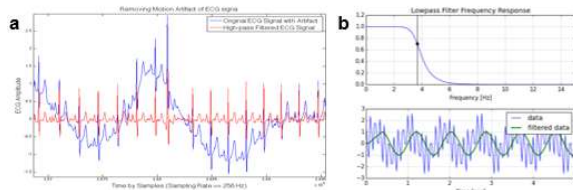


Figura 7: Filtro (a) paso alto y (b) paso bajo

Los siguientes puntos se toman del capítulo 4 del artículo de Scheffer y Girdhar. Por un lado, la envolvente crea una función que “envuelve” la señal en el tiempo, siendo tangente a ella (ver figura 8).

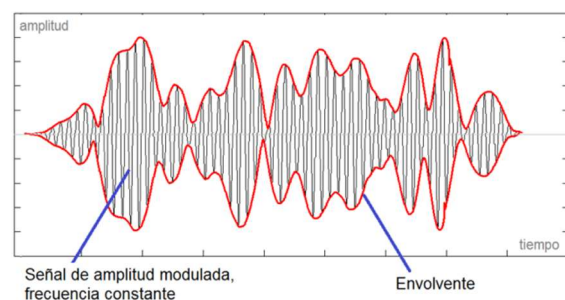


Figura 8: Envolvente a una señal en el tiempo

A continuación se “eventana” la señal envolvente (fig. 9), para acotarla y minimizar el efecto negativo de una repetición periódica, en frecuencias. De esta forma, se evitan discontinuidades al principio y al final de los bloques, a pesar de disminuir el valor de la señal (a compensar a través de una ganancia).

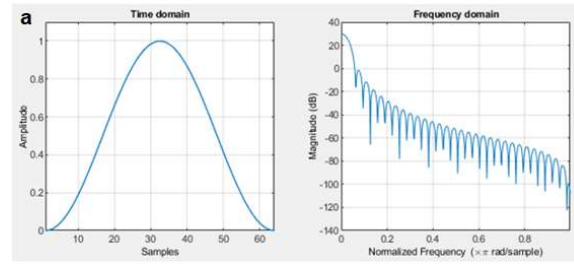


Figura 9: Eventanado 'hanning'

En contraposición al tiempo, en el dominio de la frecuencia las amplitudes se muestran como series de senos y cosenos. Para ello, se usa la Transformada Rápida de Fourier (FFT) compensada con una ganancia proporcional a la amplitud de la señal sin eventanar. Para definir las vibraciones en frecuencia se opta por usar los picos de la señal de forma que, como datos de entrada al modelo en cada fallo, se introducen conjuntos picos/frecuencias. Esta información, para usarla como entrenamiento, se ordena en forma de tabla en un archivo externo.

### 3.3. MODELO DE DETECCIÓN

Este modelo tiene un proceso de entrenamiento para que, una vez completado, se exporte y se use directamente sobre datos reales. Los datos simulados del GD se introducen en la aplicación de MATLAB®: *classification learner* (fig. 10), la cual realiza un aprendizaje a partir de datos de entrada y respuestas conocidas, usándolo para entrenar un modelo que genere predicciones sobre nuevos datos. En este caso, los datos de entrada son las salidas del preprocesamiento, siendo las etiquetas a “predecir” los fallos provocados.

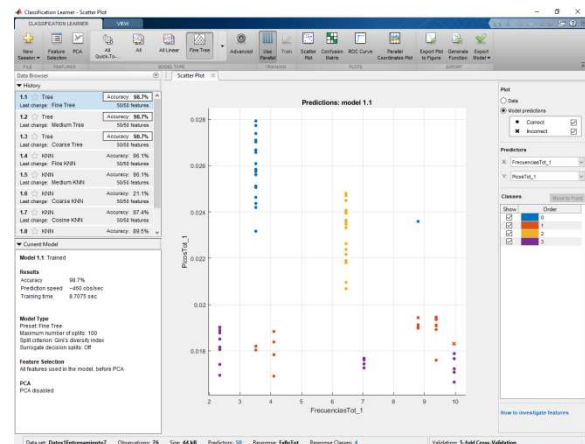


Figura 10: Interfaz de *classification learner*

La aplicación calcula el rendimiento (cobertura de validación) de los métodos, además de definir características de velocidad de predicción, uso de memoria e interpretabilidad. Existen distintas

técnicas: árboles de decisión, análisis discriminantes, regresión logística, máquinas de vector de soporte, clasificadores de vecinos más cercanos y clasificadores de conjunto. El modelo se exporta al área de trabajo y, introduciendo vibraciones del rodamiento (preprocesadas), devuelve alarmas de fallo. Aunque el trabajo es manual a través del interfaz de la aplicación, se desarrolla un sistema automático, con un proceso continuo e independiente del usuario.

### 3.4. POSTPROCESADO

Se dan resultados sencillos y comprensibles al usuario, con un mensaje por pantalla de la avería detectada.

## 4. ANÁLISIS DE RESULTADOS

### 4.1. PRUEBAS DE RENDIMIENTO

Existen parámetros en la resolución del sistema de ecuaciones a evaluar. Las gráficas presentan las vibraciones en  $y_b$ , de un fallo externo.

Las condiciones iniciales (CI, valor inicial para cada incógnita) en sistemas estables no afectan al resultado final (fig. 11). Sin embargo, unos valores discordes provocan un transitorio más grande en el tiempo y ruido frecuencial en el FFT (a). Para evitarlo, se busca ajustarlas a valores cercanos a su valor final, o, por contra, a un valor nulo. Dada la configuración del sistema, condiciones iniciales nulas devuelven vibraciones nulas (c). Por tanto, dado que las vibraciones en posición son de órdenes tan bajos, la idea óptima es asignarles valores cercanos a cero, pero no nulos, que se asemejen a sus valores en estado estacionario (b), a partir de procesos de prueba y error.

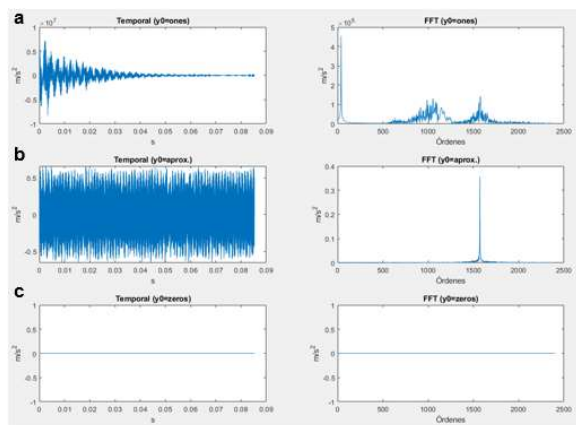


Figura 11: CI (a) altas, (b) aproximadas y (c) nulas

El error relativo ( $e_{rel}$ ) mide la tolerancia a la magnitud. En cada paso, el ODE estima el error local en el componente  $i$  de la solución. Como el sistema resuelve posiciones y velocidades de vibraciones, dadas las diferencias de órdenes, se usa solo el error relativo

(fig. 12). A medida que el error es más estricto las respuestas son más detalladas y menos bruscas (órdenes de magnitud). Sin embargo, a partir de un punto las respuestas no cambian notablemente. Dado que cuanto menor es el error mayor es el tiempo de cómputo, se valora el resultado según un baremo calidad-coste, en relación a la mejora en la salida respecto al tiempo de ejecución.

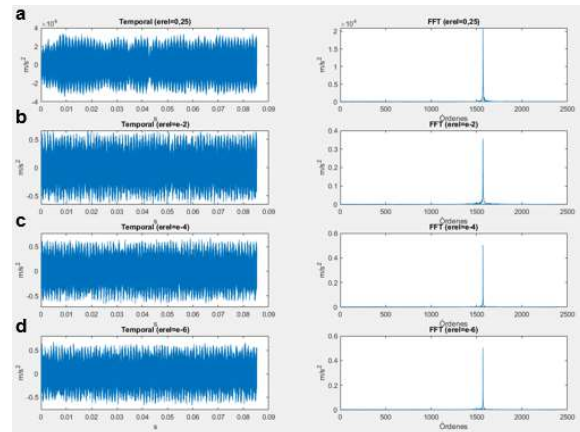


Figura 12:  $e_{rel}$  (a) 0.25, (b)  $10^{-2}$ , (c)  $10^{-4}$  y (d)  $10^{-6}$

Los ODE Solver resuelven ecuaciones diferenciales ordinarias de primer orden, tal que, en cada paso, el solucionador aplica un algoritmo a los resultados de los pasos anteriores, siendo, en el primero paso, las condiciones iniciales (ver figura 13). El resultado final es un vector de pasos de tiempo  $t = [t_0, t_1, t_2, \dots, t_f]$  y la solución  $y = [y_0, y_1, y_2, \dots, y_f]$ . Existen muchos solucionadores en función del tipo de problema y la precisión, diferenciándose, en el primer caso, la rigidez, como función de su dificultad de evaluación (diferencia en el escalamiento). Según su rigidez, se visualiza una diferencia clara entre los tres primeros casos y el último, por lo que se supone el problema no rígido. Por otro lado, centrados en los solucionadores no rígidos, en la FFT se observa más ruido en el b y c.

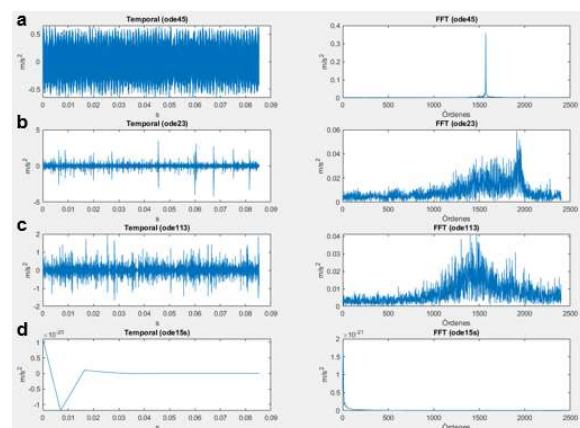


Figura 13: (a) ode45, (b) ode23, (c) ode113 y (d) ode15s

Dada una frecuencia de muestreo fija (48kHz para visualizar los fallos), el número de puntos ( $n_{tot}$ ) influye en el tiempo total simulado. Aumentarlo aumenta la resolución frecuencial, mejorando la precisión en los resultados (ver figura 14). No obstante, la calidad tiene un coste: cuantos más puntos a simular, mayor tiempo de ejecución. Se valora la opción óptima en relación calidad-tiempo, usando gráficas de la envolvente de  $y_b$  en tiempo y frecuencia.

Las frecuencias de fallos, a partir de las fórmulas del “Curso de Análisis de Vibraciones” [3] son: **BPFO** (defecto de carrera externa): 48,9 Hz, **BPFI** (defecto de carrera interna): 71,1 Hz y **BSF** (velocidad de giro de bolas): 26,5 Hz. Con estos datos (junto con la frecuencia de giro: 10 Hz) se busca una solución que los englobe de forma óptima, por ejemplo, de 3 Hz. Conocida la resolución, el número de puntos óptimo, teniendo en cuenta restricciones de potencia de dos y transitorio (10%), es de 18210, con la mínima pérdida de información (ecuaciones 6-7).

$$18210 * 0,9 = 16389 \rightarrow 2^{14} = 16384 \quad (6)$$

$$48 \text{ kHz} / 16384 = 2,93 \text{ Hz} \quad (7)$$

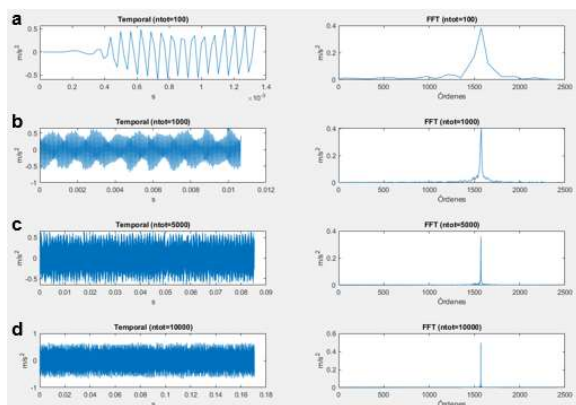


Figura 14:  $n_{tot}$  (a) 100, (b) 1000, (c) 5000 y (d) 10000

## 4.2. RESULTADOS CONJUNTOS

Se definen las cinco variables de vibraciones: los GDL del eje/carrera interna, los de la carcasa/carrera externa (inducidas) y la respuesta a la vibración en el sensor (medida). Destacar que la vibración  $y_s$  e  $y_b$  son proporcionales, ambas simulando la vibración en un mismo objeto, aunque la  $y_b$  a través de un sensor unido por un sistema masa-resorte (volver a la figura 5). Cada caso de fallo se trata con las vibraciones de la variable que más lo caracterice, en el espectro temporal y frecuencial, sin procesar (fig. 15).

**FALLO EXTERNO:** Las bolas internas del rodamiento giran a una velocidad  $\omega_c$  de 4,075 Hz. Sabiendo el número de bolas del rodamiento ( $n_b = 12$ ), se calcula el tiempo entre dos golpes consecutivos. Además, para

demostrar este valor, la inversa de ese tiempo es igual al **BPFO** (ver ecuación 8).

$$1/\omega_s * 1/n_b = 0,0204 \text{ s} \quad (8)$$

**FALLO INTERNO:** En el fallo externo uno de los elementos de choque tenía velocidad de giro nula (carrera externa). Sin embargo, ahora se diferencian dos elementos a distintas velocidades: las bolas, a  $\omega_c$ , y la carrera interna, a  $\omega_s = 10$  Hz (velocidad de giro del eje). A través de la velocidad relativa, se obtienen los resultados, donde se diferencian esos “golpes” de vibraciones. Por otro lado, haciendo la inversa del valor temporal se llega al **BPFI** (ecuación 9).

$$1/(\omega_s - \omega_c) * 1/n_b = 0,0141 \text{ s} \quad (9)$$

**FALLO BOLA:** Para el fallo de la bola, se requiere de la velocidad de giro de las bolas dentro del rodamiento  $\omega_{spin} = 26,1$  Hz. Sabiendo que, por cada giro de la bola, se producen dos golpes (en la carrera interna y en la externa), la frecuencia es de 52,2 Hz, igual que el doble de la frecuencia de giro de la bola (ecuación 10).

$$1/\omega_{spin} * 1/2 = 0,0190 \text{ s} \quad (10)$$

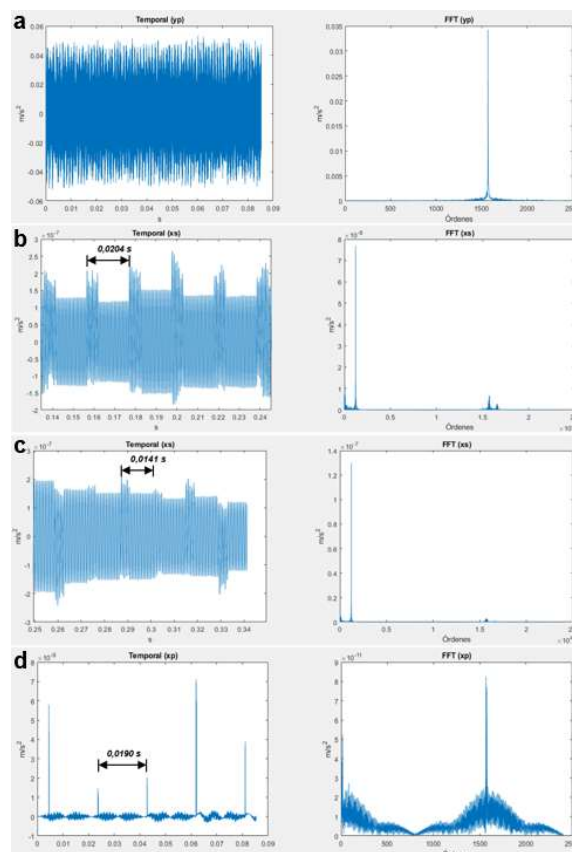


Figura 15: (a)  $y_s$  sin fallo (b)  $x_s$  fallo externo, (c)  $x_s$  fallo interno y (d)  $x_p$  fallo de bola

**CONJUNTO PROCESADO:** Gracias a la resolución, las gráficas de las señales procesadas presentan, de forma correcta, el espectro de frecuencias en cada fallo (fig. 16). Sin embargo, las frecuencias de fallo no se visualizan de una forma óptima, por factores como: la exactitud del modelo, el ruido que introduce al sistema o la necesidad de una resolución mayor.

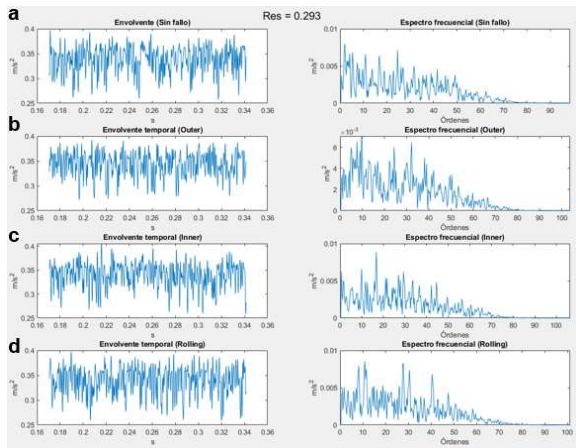


Fig. 16:  $y_b$  (a) Sin fallo, (b) fallo externo, (c) fallo interno y (d) fallo de bola

Introducida la tabla de datos de ensayos, se procesan con los métodos para clasificar, para ver cada porcentaje de acierto. Este proceso se hace a través de la técnica de validación escogida, a partir de un porcentaje de los datos de entrada, tal que el sistema “prueba” el modelo con esos datos para confirmar su funcionamiento. Observando la precisión de cada uno, se elige el árbol de decisión como técnica de clasificación óptima, con un 98,7% de cobertura (ver tabla 2).

Tabla 2: Técnicas de clasificación y cobertura

| Model | Method      | Accuracy        | Features       |
|-------|-------------|-----------------|----------------|
| 3     | Tree        | Accuracy: 98.7% | 50/50 features |
| 4     | Naive Bayes | Failed          | 50/50 features |
| 5     | SVM         | Accuracy: 21.1% | 50/50 features |
| 6     | KNN         | Accuracy: 21.1% | 50/50 features |
| 7     | Ensemble    | Accuracy: 21.1% | 50/50 features |
| 8     | Ensemble    | Accuracy: 84.2% | 50/50 features |
| 9     | Ensemble    | Accuracy: 21.1% | 50/50 features |

A continuación, se extraen los parámetros que caracterizan el modelo. En el gráfico de dispersión (fig. 17) se trazan los datos de los predictores del sistema, donde los puntos se diferencian en datos que el sistema, en validación, acierta o falla, dando la cobertura final. Por otro lado, la matriz de confusión (fig. 18) muestra el número de fallos que el modelo entrenado comete en validación, graficando los fallos

que debería dar y las que realmente da. Se observa cómo los cuadrados que salen de la diagonal principal suponen errores a la hora de predecir el fallo que se detecta en el sistema.

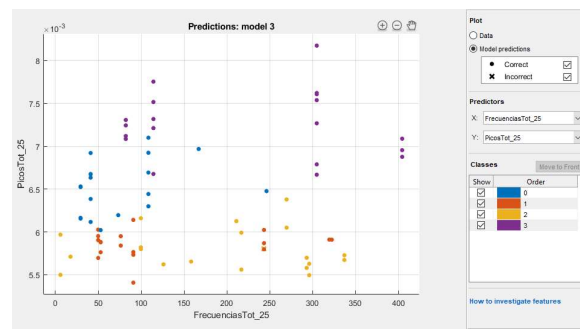


Figura 17: Gráfico de dispersión

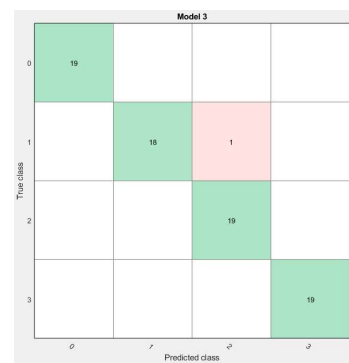


Figura 18: Matriz de confusión

Por último, el modelo proporciona unas características intrínsecas del proceso: *observaciones*, número de ensayos que se usan como datos de entrada (19 ensayos por cada situación de fallo, en total 76 casos); *tamaño*, espacio que ocupan los datos de entrada (44 kB); *predictores*, número de variables que definen el modelo (cinco parejas picos-frecuencias en cada fallo, en total 50 para cada ensayo); *clases de respuesta*, tipos de salidas (número de fallos: nulo, externo, interno o de bola); y *validación*, tipo y clase (5-fold *cross-validation*). Como otras características, se obtienen la velocidad de predicción, el tiempo de entrenamiento y el tipo de modelo, entre otros.

Con la extracción del modelo se define un proceso automático de generalización, de forma que se crucen unas pruebas de simulación con el modelo de detección, para comprobar la cobertura. El entrenamiento se realiza a través de ensayos de fallos, añadiéndoles ruido al espectro frecuencial de la envolvente. En el funcionamiento se prueba el modelo introduciendo los datos de esos cuatro ensayos limpios. El resultado de predicción del modelo obtiene un 100% de aciertos.

## 5. CONCLUSIONES

En este trabajo se presenta un sistema automático de detección de alarmas en el rodamiento del tren mecánico de un aerogenerador que, junto a un gemelo digital (modelo físico virtual del componente) y empleando un modelo de detección entrenado con datos simulados, consigue devolver las alarmas de fallos en el dispositivo. El sistema simula el comportamiento del rodamiento ante situaciones de fallo, para introducir esos resultados como entrada al modelo de detección. Este, basándose en una técnica de clasificación, entrena para ser capaz de, posteriormente, asociar vibraciones de sensores reales con casos de fallo.

El motivo principal de este trabajo es evolucionar el mantenimiento de aerogeneradores a un método predictivo, evitando que se pierda vida útil de componentes, de tal forma que se maximice la cobertura del sistema ante fallos, evitando falsos positivos/negativos. Esta situación, valiéndose de modelos físicos y no estadísticos, induce a una mejor valoración de los datos del rodamiento, obteniendo más de un 90% de cobertura, con una lista muy reducida de ensayos. Al usar ecuaciones físicas en vez de métodos estadísticos, se aumenta la fiabilidad, disminuyendo la aleatoriedad en el sistema.

Por otro lado, dejando a un lado la óptima cobertura, lo más relevante es el “precio a pagar” por la calidad de los resultados. Tanto el tiempo de simulación como la resolución frecuencial dependen directamente del tiempo. Además, también influye en la calidad del entrenamiento el número de pruebas que se introduzcan como datos de entrada, de cara a abarcar una mayor variedad de opciones. Además, lograr las mejores respuestas en el modelo de detección depende de la exactitud con la que el GD simule el comportamiento real del elemento. Cuanto más se ajuste la formulación matemática al movimiento del rodamiento, mejores resultados se obtienen. Por otro lado, el preprocesamiento de las vibraciones es, sin duda, el aspecto más completo, dada su gran cantidad de variantes y métodos.

### Agradecimientos

El autor quiere agradecer a la empresa Ingeteam Power Technology S.A., en especial al departamento de Automation Devices: Luis Pantoja, Nuria Bilbao y demás, a Bob Randall, de la UNSW, y a la escuela de ingeniería de Bilbao, a profesores del departamento de Sistemas y Automática y de Mecánica como Mikel Larrea y Josu Aguirrebeitia, y, especialmente, al tutor de mi Proyecto Fin de Máster, Eloy Irigoyen.

## English summary

### DETECTION SYSTEM OF FAILURE ALARMS IN THE MECHANICAL TRAIN OF A WIND TURBINE

#### Abstract

*This project develops an automatic system for detecting alarms in the mechanical train of a wind turbine. This system is based on a digital twin, that is, a simulated physical model of the study set (system), to train with simulated data to the system. To this is added a trained detection model based on classification techniques, which, based on real data, returns the possible failure alarms of the study element. In this way, coverage is maximized against failures, minimizing false positives and negatives of the system. The project covers faults in a bearing, with a procedure that can be extrapolated to the entire mechanical train.*

**Keywords:** Digital twin, Failure mode and effects analysis, Detection model, Coverage and Wind turbine.

#### Referencias

- [1] Antonio J. Barbero García, “Energía eólica”, Tema 6, curso de Física Ambiental (UCLM).
- [2] Dong, W., Xing, Y and Moan, T., “Time Domain Modeling and Analysis of Dynamic Gear Contact Force in a Wind Turbine Gearbox with Respect to Fatigue Assessment”, *Energies*, 2012.
- [3] Ingeteam, “Curso Análisis de Vibraciones”, 2018.
- [4] Sawalhi, N. and Randall, R.B., “Simulating gear and bearing interactions in the presence of faults”, *Mechanical System and Signal Processing* 22 (1924-1951), 2008.
- [5] Scheffer, C. and Girdhar, P., “Machinery Vibration Analysis & Predictive Maintenance”, 2004.



© 2019 by the authors. Submitted for possible open access publication under the terms and conditions of the Creative Commons Attribution CC BY-NC-SA 4.0 license (<https://creativecommons.org/licenses/by-nc-sa/4.0/deed.es>).