

A Hybrid Approach for Fault Detection in Wind Turbines

Rubén Francisco Manrique, José Joaquín Parrado

Abstract—In this research, a hybrid system for fault detection and isolation (FDI) is proposed. The system incorporates the well-known benefits of residual generation approaches with a classification system for the automatic determination of thresholding. The architecture of the FDI system is divided in three levels:(i) the first one models the monitoring system signals (sensors) with a set of neural networks; (ii) the second makes the residual generation based on the difference of the actual output and the estimation of the neural network; ; (iii) and the third level assesses the set of residues using a classification system, at this stage the detection and fault isolation is done. The FDI system is evaluated in a simulated model of wind turbine with nine different types of issues. The results suggest a decrease in the number of false alarms, compared to a data-driven approach without loss in detection speed.

Keywords—*fault detection, data driven, neural networks.*

I. INTRODUCCIÓN

Uno de los mayores problemas que presentan los sistemas industriales es la identificación de la ocurrencia de una falla. En muchos casos su no identificación oportuna puede ser extremadamente costosa en términos de vidas humanas, impacto ambiental y pérdidas económicas. En los últimos años se evidencia una creciente demanda de sistemas industriales de mayor rendimiento, mejor seguridad y fiabilidad, que sean capaces de operar autónomamente y tomar la mejor decisión en presencia de una falla. Por esto los sistemas de detección y aislamiento de fallas (FDI Fault Detection and Isolation) se han convertido en un componente esencial en el desarrollo de sistemas inteligentes que estén en la capacidad de detectar, aislar (ubicar) y estimar autónomamente la severidad de una o varias fallas.

Los métodos de detección y aislamiento de fallas se pueden clasificar de manera amplia en dos tipos: los basados en modelos (model based -MB) o los libres de modelos controlados por los datos (data driven- DD). En términos generales, los basados en modelos (MB-FDI) usan el conocimiento a priori del sistema para desarrollar modelos, los cuales son usados como referencia para evaluar diferencias (residuos) entre el estado actual de operación, determinado por la mediciones del sistema de monitoreo, y el estado de operación esperado obtenido del modelo [1], [2]. La calidad de un sistema FDI con enfoque basado en modelos depende directamente de la precisión del modelo construido para describir la dinámica del proceso real

siendo por lo tanto, potencialmente sensibles a errores de modelado, variación paramétrica, ruido y perturbaciones [3].

Por otro lado, están los métodos controlados por los datos (DD-FDI) los cuales han ganado gran aceptación en los últimos años debido en parte al desarrollo de mejores técnicas de aprendizaje que permiten definir y detectar patrones [4]. Para estos, a diferencia de los anteriores no es necesario especificar un modelo matemático del sistema para los procesos de detección y aislamiento, en su lugar usan colecciones de datos provenientes del sistema de monitoreo para descubrir patrones que permitan caracterizar e identificar fallas en el sistema. El desarrollo de sistemas FDI DD aun es un campo que esta siendo explorado en la actualidad [5], [6] y debe superar limitaciones asociadas al alto número de falsas alarmas que pueden presentar. La presente investigación mezcla un proceso de generación de residuos mediante redes neuronales típico de sistemas MB-FDI con un sistema de clasificación para umbralización proveniente de los enfoques data-driven. En la sección II se presenta de forma global la arquitectura del sistema FDI propuesto, en la sección III se explica el modelo de turbina de viento y los escenarios de falla contemplados para la evaluación del sistema, en la sección IV se describe el proceso para la construcción de los modelos de regresión con redes neuronales y la generación de residuos, la construcción de los clasificadores se explica en la sección V, en la sección VI se evalúan los resultados obtenidos, finalmente se presentan las conclusiones en la sección VII.

II. ARQUITECTURA DEL SISTEMA HIBRIDO PROPUESTO

El esquema híbrido propuesto puede ser dividido en tres etapas (Fig. 1). La etapa 1 involucra el modelamiento de cada una de las señales del sistema de monitoreo (sensores) con una red neuronal, cuya función va ser la de predecir el valor de dicha señal en el instante de tiempo k a partir de un conjunto de señales de entrada (redundancia analítica). La etapa 2 consiste en la generación de residuos a partir de la comparación de la salida real y la estimación de la red neuronal y finalmente la etapa 3 evalúa el conjunto de residuos generados con un sistema de clasificación. En la etapa 3 se hace tanto la detección como es aislamiento de la falla.

Es de notar que la diferencia principal con un esquema totalmente DD son las entradas al sistema clasificador, en vez de trabajar directamente con las señales del sistema de monitoreo se trabaja con un conjunto de residuos generados por la diferencia entre la salidas reales y estimaciones/predicciones de los modelos de regresión construidos con redes neuronales.

R. Manrique is with the School of Exact Sciences and Engineering, Universidad Sergio Arboleda (ruben.manrique@usa.edu.co).

J. Parrado is with the Department of Computer Science and Systems, Universidad San Martín (jose.parrado@ingenieria.sanmartin.edu.co) .

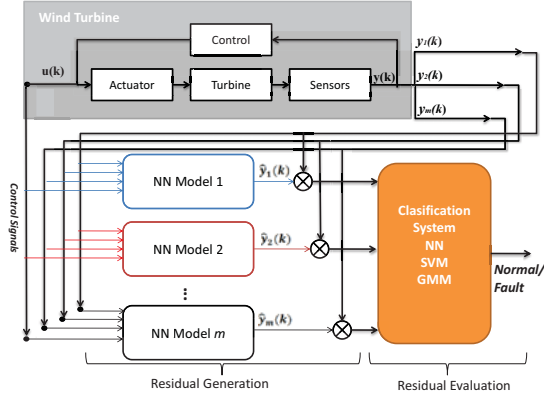


Fig. 1. Arquitectura híbrida utilizada.

III. CASO DE ESTUDIO: TURBINA DE VIENTO

Uno de los mayores inconvenientes en la industria eólica son los costos de mantenimiento necesarios para tener una turbina en funcionamiento. Estos costos en su mayoría son producto de fallas que no fueron detectadas con suficiente antelación como para tomar medidas correctivas que evitaran un daño permanente en la turbina. Según estadísticas de la World Wind Energy Association la mayoría de los mantenimientos realizados a turbinas de viento son para reemplazar partes dañadas que juegan un papel crítico en su funcionamiento [7], en Alemania la IWES Fraunhofer-Institut für Windenergie und Energiesystemtechnik establece que el 63,1% de las fallas que se presentaron en el 2009 tuvieron como consecuencia el apagado temporal (o en algunos casos permanente) de la turbina [8]. Se estima que los costos totales de mantenimiento a lo largo de la vida útil de una turbina llegan al menos a un 10-15% de los ingresos totales de la generación [7]. Las turbinas de viento trabajan bajo condiciones ambientales muy duras, la temperatura, la presión y la humedad están en constante cambio, por lo que los tipos y formas de fallas son muchas y variadas, adicionalmente el acceso a la maquinaria de una turbina eólica puede llegar a ser muy complicado y extremadamente costoso en especial para turbinas de alta potencia. El modelo de la turbina de viento trabajado fue propuesto en su versión inicial en [9] como inicio de una competencia que buscaba un sistema de detección y diagnóstico de fallas de alto rendimiento. El modelo está construido en Simulink (MATLAB).

A. Descripción general del sistema

En este modelo se considera una turbina de tres hojas de eje horizontal de velocidad variable con convertidor completo. Funcionalmente el viento hace girar las hojas de la turbina de viento, la conversión de energía cinética en energía mecánica puede ser controlada cambiando la aerodinámica de la turbina girando las hojas mediante el sistema de pitch o mediante el control de la velocidad de rotación. La energía mecánica se convierte en energía eléctrica mediante un generador totalmente acoplado a un convertidor. Entre el rotor y el generador una transmisión se utiliza para aumentar la velocidad de rotación final que llega al generador. El objetivo del sistema de control es seguir la potencia de referencia, o si no es posible

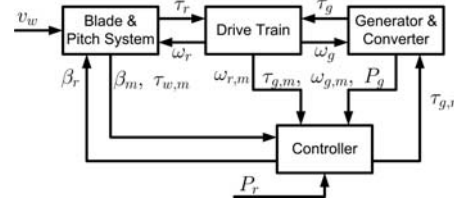


Fig. 2. Interconexión del modelo de turbina de viento

minimizar el error de referencia. En la Fig. 2 se puede ver en diagrama de bloques las relaciones entre los componentes del sistema: Hojas-sistema de Pitch, Transmisión, Generador y Controlador.

V_w es la velocidad del viento que es capturada a través de las hojas. El movimiento se propaga al rotor con velocidad angular w_r y torque T_r . A la salida de la transmisión está la velocidad de entrada al generador W_g (angular) y torque T_g . El sistema de control se vale de las señales medidas provenientes del sistema de monitoreo, las cuales son en últimas las disponibles en una turbina real. Estas señales son: $\beta_{m1}, \beta_{m2}, \beta_{m3}$, mediciones para cada uno de los tres pitch. W_{mr} medición para la velocidad del rotor. W_{mg} medición para la velocidad del generador. T_{mg} torque en el generador. P_{mg} potencia en el generador. Las salidas del sistema de control son las referencias de cada uno de los pitch $\beta_{r1}, \beta_{r2}, \beta_r$ así como el torque en el generador T_{rg} .

B. Escenarios de falla

En este modelo se consideran diferentes tipos de fallas enfocándose en las de mayor ocurrencia en las turbinas modernas. A diferentes intervalos de tiempo se inducen 9 diferentes tipos de fallas, las cuales pueden ser clasificadas dependiendo del origen de la misma:

Tabla 1. FALLAS PRESENTES EN EL SET DE DATOS

Nomenclatura	Descripción
Falla 1	Fixed value on Pitch 1 position sensor1
Falla 2	Scaling error on Pitch 2 position sensor2
Falla 3	Fixed value on Pitch 3 position sensor1
Falla 4	Fixed value on Rotor speed sensor1
Falla 5	Scaling error on Rotor speed sensor2 and Generator speed sensor2
Falla 6	Changed pitch system response pitch actuator2 high air content in oil
Falla 7	Changed pitch system response pitch actuator3 low pressure
Falla 8	Offset in Converter torque control
Falla 9	Changed Dynamics Drive train

- Fallas de los sensores: corresponden a fallas en los sensores del sistema de alabeo en cualquiera de las medidas de posición del pitch, velocidad del rotor, o velocidad de generador (Fallas 1- 5 Tabla 1).
- Fallas en el Actuador: los actuadores son los sistemas físicos sobre los cuales es posible aplicar una señal de control (con el objeto de mantener un estado deseado), en este caso los pitch y el sistema de torque (Fallas 6-8).
- Fallas del Sistema: corresponden a fallas en el eje principal de transmisión (Falla 9 Tabla 1).

Tabla 2. RESUMEN DE LAS CARACTERISTICAS DE LAS REDES NEURONALES ENTRENADAS

	$Wr - NN$	$Wg - NN$	$Tg - NN$	$Pg - NN$	$B_1 - NN$	$B_2 - NN$	$B_3 - NN$
Error de estimación	0.0973 rad/s	0.1054 rad/s	1.74 Nm	0.891 KNm/s	0.070 deg	0.091 deg	0.081 deg
Atributos de entrada	Tg_{ref}, Wr, Pg, Vm	Tg_{ref}, Wg, Pg, Vm	Tg_{ref}, Wg, Tg, Vm	Tg_{ref}, Pg, Vm	Tg_{ref}, B_1, B_1, Vm	Tg_{ref}, B_2, B_2, Vm	Tg_{ref}, B_3, B_3, Vm
Salida	$\hat{W}r$	$\hat{W}g$	$\hat{T}g$	$\hat{P}g$	\hat{B}_1	\hat{B}_2	\hat{B}_3
Neuronas en capa oculta	15	17	21	11	9	11	10
Normalización de los datos de entrada	0-10	0-10	-1-1	0-10	0-10	0-10	0-10

IV. CONSTRUCCIÓN DE LOS MODELOS DE REGRESIÓN CON REDES NEURONALES

Tal y como se muestra en la Fig. 3 un total de 7 modelos de regresión se construyeron. La estructura empleada en todas las redes neuronales se conoce en la literatura como EMRAN RBF, su selección se realizo debido a su excelente capacidad de generalización y su habilidad de reducir neuronas ocultas que no contribuyen significativamente a la salida de la red neuronal. Esta propiedad permite reducir el tamaño total de la red y su tiempo de ejecución lo cual es beneficioso para aplicaciones en línea donde existen limitaciones de potencia computacional.

La EMRAN RBF consiste en un grupo de señales de entrada, una capa oculta y una capa de salida. Empieza inicialmente con cero neuronas ocultas y se van añadiendo si y solo si los siguientes tres criterios se cumplen:

$$e_k = |y_k - \hat{y}_k| > E1 \quad (1)$$

$$e_{RMS_k} = \sqrt{\sum_{j=k-(n-1)}^k \frac{e_j^2}{n}} > E2 \quad (2)$$

$$d_k = \|\mathbf{x}_k - \mu_{rk}\| > E3 \quad (3)$$

Donde y_k y \hat{y}_k son la salida real del sensor y la estimación de la red neuronal en el tiempo de muestra k , y μ_{rk} es el centro (media) de la neurona oculta más cercana al vector de entrada actual x_k . $E1$ asegura que el error de estimación sea menor que un umbral, $E2$ chequea si la RMS (root mean square) de los pasados n errores de estimación son suficientemente pequeños y $E3$ verifica si la distancia mínima entre el vector actual de entrada y los centros de las neuronas ocultas es significativamente pequeña. $E3$ es generalmente expresada como $E3 = \max\{\epsilon_{max}\gamma_{df}^i, \epsilon_{min}\}$, donde ϵ_{max} es el valor máximo inicial que decae exponencialmente según γ_{df}^i hasta un valor mínimo ϵ_{min} [10].

Si al menos uno de estos tres criterios no se cumple entonces el algoritmo de entrenamiento actualiza los parámetros (peso, centro y ancho) de la neurona 'mas' activa (o también llamada la 'ganadora'). Esto reduce el número de parámetros a ser actualizados lo que agiliza el proceso de entrenamiento. Una descripción detallada de su implementación y evaluación puede ser encontrada en [10].

A. Selección de entradas/salidas de las redes neuronales

En principio se construyeron redes que tenían como entradas todas las posibles señales del sistema de monitoreo, como resultado se obtuvieron procesos de entrenamiento lentos, redes de grandes tamaños y modelos resultantes con poca capacidad de generalización. Fue necesario realizar pequeñas pruebas para eliminar aquellas señales de entrada que tienen una contribución menor en la estimación. Esto se logro calculando la diferencia entre error de estimación RMS antes y después que un atributo es incluido en el set de entrenamiento. Si la diferencia es pequeña y se mantiene en diferentes modelos entrenados, entonces el atributo puede ser removido de forma segura sin que esto afecte de forma significativa la estimación. El resultado de las 7 redes se describe por las siguientes relaciones de entrada/salida, donde k es el tiempo de muestreo:

$$Wr_k = NN(Tg_{ref_{k-1}}, Wr_{k-1}, Pg_{k-1}, Vm_{k-1}) \quad (4)$$

$$Wg_k = NN(Tg_{ref_{k-1}}, Wg_{k-1}, Pg_{k-1}, Vm_{k-1}) \quad (5)$$

$$Tg_k = NN(Tg_{ref_{k-1}}, Wg_{k-1}, Tg_{k-1}, Vm_{k-1}) \quad (6)$$

$$Pg_k = NN(Tg_{ref_{k-1}}, Pg_{k-1}, Vm_{k-1}) \quad (7)$$

$$B1_k = NN(Tg_{ref_{k-1}}, B1_{ref_{k-1}}, B1_{k-1}, Vm_{k-1}) \quad (8)$$

$$B2_k = NN(Tg_{ref_{k-1}}, B2_{ref_{k-1}}, B2_{k-1}, Vm_{k-1}) \quad (9)$$

$$B3_k = NN(Tg_{ref_{k-1}}, B3_{ref_{k-1}}, B3_{k-1}, Vm_{k-1}) \quad (10)$$

Un resumen de las estructuras de las redes construidas se muestra en la Tabla 2. El error de estimación se obtuvo al evaluar el desempeño individual de cada red aplicado a un set de pruebas de 5.000 registros.

B. Generación de Residuos.

Un residuo se forma por la diferencia entre la salida real del sensor y la estimación, sin embargo este enfoque hace al sistema susceptible a falsas alarmas. En este trabajo se uso una forma filtrada utilizada en [11], [12]:

$$r_k = \bar{w} \sum_{j=k-(\Omega-1)}^k \frac{(y_j - \hat{y}_j)^2}{\Omega} \quad (11)$$

En esta fórmula para el cálculo del valor final del residuo r_k se emplea un filtro de medias móviles para los Ω residuos pasados. \bar{w} cumple con la función de escalar la salida del valor del residuo final (en caso de que sea necesario).

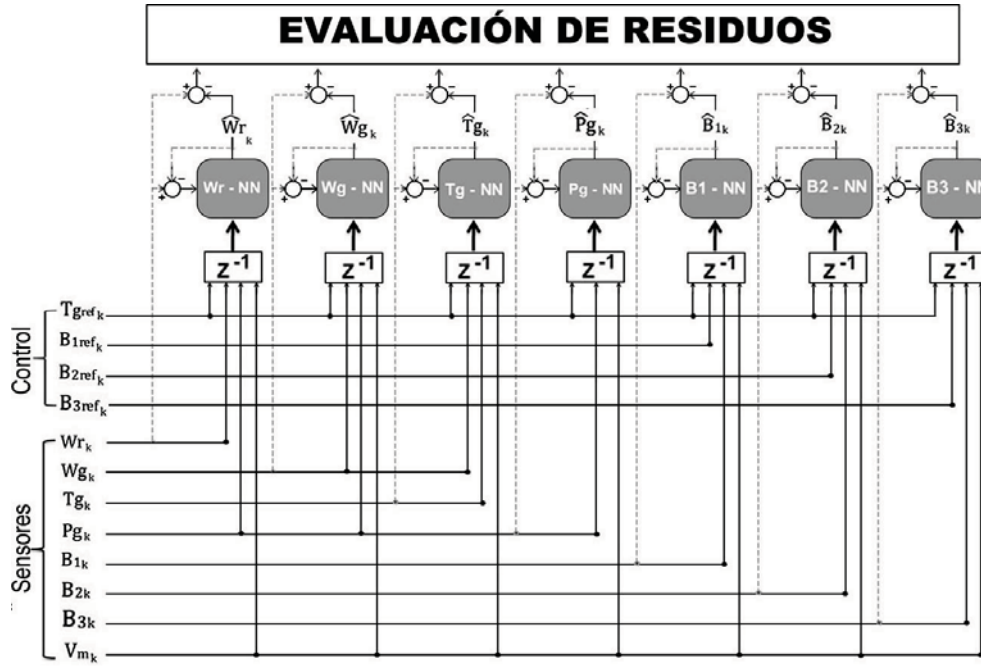


Fig. 3. Redes neuronales para la generación de residuos.

Los valores de los Ω de cada modelo se obtuvieron experimentalmente manteniendo un balance entre la sensibilidad a fallas y el número de falsas alarmas. Por su parte se usó \bar{w} para mantener los valores de todos los residuos en un rango común para facilitar su evaluación con los clasificadores. Los valores encontrados de estas variables se resumen en la Tabla 3.

Tabla 3. VALORES DE Ω Y \bar{w} UTILIZADOS PARA LA GENERACIÓN DE RESIDUOS

	$Wr - ri, Wg - ri$	$Tg - ri$	$Pg - ri$	$B1 - ri, B2 - ri, B3 - ri$
\bar{w}	100	0.1	10^{-4}	1
Ω	13	15	12	10

En la Figura 4 se ejemplifica la generación de residuos para el caso de la Falla 2 de valor escalado en el sensor del pitch 2 introducida a los 50 tiempos muestrales de la simulación. En un ambiente ideal se obtendrían residuos diferentes de cero exclusivamente con la ocurrencia de una falla, sin embargo como ya se ha señalado el ruido y el error de estimación propio de los modelos pueden generar perturbaciones en los residuos. Determinar el umbral para el cual un residuo puede ser con certeza categorizado como falla puede ser una tarea ardua y requiere de amplia experimentación sobre el sistema. Como alternativa, en este trabajo de investigación se construirán clasificadores para la evaluación de los residuos, los algoritmos de SVM (Support vector machines), NN (Neural Networks), KNN (K-nearest neighbors) y GMM (Gaussian mixture model) serán empleados y se seleccionará aquel con mejor desempeño. La selección y optimización de los parámetros de cada algoritmo se realizó de forma experimental.

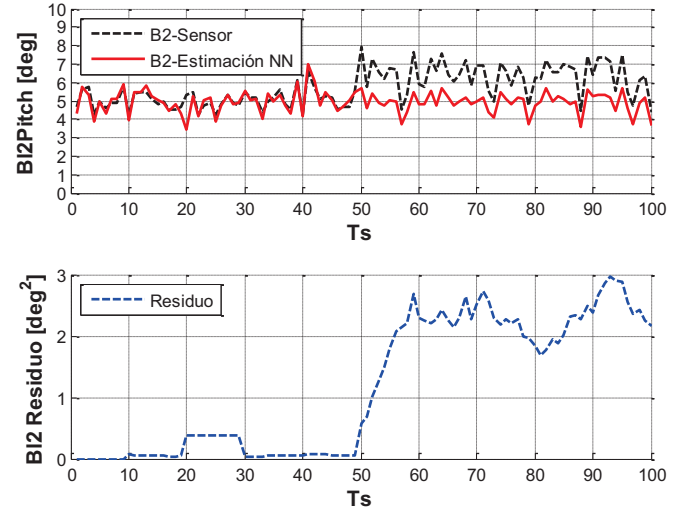


Fig. 4. Comportamiento de los residuos $B_2 - ri$ (gráfica inferior) al ser introducida una falla de valor escalado a los 50 Ts de ejecución en la señal B_2 (gráfica superior)

V. EVALUACIÓN DE RESIDUOS: CONSTRUCCIÓN DE CLASIFICADORES

El sistema de clasificación propuesto tiene como entrada todos los residuos generados en la etapa previa y debe identificar 10 diferentes clases (9 de Falla y una de comportamiento Normal). Los posibles escenarios de clasificación son: i) Se está en presencia de un comportamiento normal, todos los residuos deben estar en niveles muy próximos a cero. ii) Hay algún tipo de falla, los valores de uno o más de los atributos de entrada presentan valores mayores 0. Para la construcción

del set de entrenamiento se generaron secuencias de residuos para cada uno de los 9 tipos de Fallas. En un número igual se generaron residuos de comportamiento normal con el objeto de que el clasificador aprenda a diferenciar el ruido natural de las señales evaluadas. Un set total de 100.000 registros de residuos (80.000 de Fallas 1-9 y 20.000 de comportamiento normal) se empleo para el entrenamiento de los clasificadores.

Para su posterior evaluación se construyeron 50 set de pruebas, cada uno correspondiente a una simulación de 1000 segundos (100.000 registros), los resultados promedios obtenidos por los algoritmos evaluados en términos de precisión y sensibilidad se muestran en la Tabla 4.

$$Pre = \frac{\text{Registros correctamente clasificados}}{\text{Numero de registros}} \times 100\% \quad (12)$$

La sensibilidad se puede definir para cada clase c_i como:

$$Sens_{c_i} = \frac{\text{Registros de la clase } c_i \text{ correctamente clasificados}}{\text{Numero de registros de la clase } c_i} \times 100\% \quad (13)$$

En precisión global las GMM superan a las máquinas de vectores de soporte y redes neuronales. La sensibilidad para las fallas en sensores supera el 90% en todos los algoritmos, siendo las SVM las de mejor desempeño. No lograr un porcentaje del 100% de sensibilidad se debe en principio a el tiempo requerido para el levantamiento del residuo y su posterior clasificación, tal y como se ejemplifica en la Fig. 5 con la ocurrencia de la Falla 2 (introducida a los 50 Ts). En esta figura se ven todos los residuos de entrada al clasificador, sin embargo dada la naturaleza de la falla, el residuo $B_2 - ri$ es el único que se incrementa. La detección se realiza cuando este residuo supera el umbral imaginario a los 53 TS, es decir 3 Ts después. Estos tiempos de levantamiento van en detrimento de la sensibilidad de cada clase y en últimas van a determinar el tiempo que tarda el sistema FDI en detectar una falla.

Se habla de umbral imaginario y no real porque los sistemas de clasificación empleados definen su salida en función de la relación (lineal o no lineal) de múltiples atributos de entrada y su predicción puede variar según las condiciones de cada uno de ellos. Para el caso expuesto en la Fig. 5 un clasificador de tipo SVM determino que el vector de entrada $[B_1 - ri = 0.049, B_2 - ri = 1.282, B_3 - ri = 0.02, Wr - ri = 0.1016, Wg - ri = 0.1029, Tg - ri = 0.72, Pg - ri = 0.19]$ presente a los 53 Ts pertenece a la clase Falla 2. Esta clasificación puede variar si por ejemplo se cambian el valor de Pg-ri a 1.16, en este caso la SVM determina que dicho vector pertenece a una Falla 6 que corresponde a una falla hidráulica en el pitch 2.

Finalmente, dado que presenta la mayor precisión y una alta sensibilidad a cada una de las Fallas, se selecciona las GMM para la construcción final del sistema FDI híbrido. A continuación se entrará a realizar pruebas experimentales en término de las métricas definidas en el benchmark.

VI. PRUEBAS EXPERIMENTALES

Se utilizaron 50 set de datos de 4400 segundos de simulación (440.000 registros) contruidos para la evaluación del

sistema FDI propuesto en [5]. Cada set de datos contiene diferentes secuencias de Fallas. Las métricas de evaluación empleadas fueron:

- Tiempo de detección (promedio -TDP-, máximo -TDMA- y mínimo -TDMI-): Tiempo que le toma al sistema FDI detectar la ocurrencia de una falla.
- Fallas no detectadas (FND): Número de fallas que no fueron detectadas.
- Falsas alarmas (FA): Número promedio de falsas alarmas.
- Tiempo de recuperación de una falsa alarma (TR-FA): Tiempo promedio que tarda el sistema FDI en recuperarse posterior a la ocurrencia de una falsa alarma.

Los resultados obtenidos (promedios) se muestran en la Tabla 5.

VII. CONCLUSIONES

De la comparación de estos resultados con los obtenidos por las investigaciones [5], [6] se pueden concluir que hay una reducción notoria del número de falsas alarmas presentadas y se pueden reducir aun más con incrementos en el parámetro Ω a costa de tiempo de detección. Con un valor de $\Omega=100$ en todos los modelos de redes, las falsas alarmas se reducen a 0 con tiempos de detección entre 25 Ts y 45 Ts mayores a los presentados en investigaciones similares. Esto constituye una ventaja ya que fácilmente se puede variar este parámetro y adecuar la respuesta del sistema a las necesidades del problema. En general el modelo aquí propuesto mejora los tiempos de detección de todas las fallas, a pesar de esto los problemas de oscilación entre zonas de operación de la turbina reportados por [5] siguen afectando la velocidad de detección de las Fallas 6 y 7. Con la inclusión de las señales de control se mejora de forma notoria la detección de la Falla 8, de hecho dado que el offset se presenta directamente en el lazo de control es todo un reto detectar esta falla a partir de las señales del sistema de monitoreo. Un punto en contra del modelo híbrido propuesto es su lenta implementación, la construcción de los modelos de regresión con redes neuronales toma un tiempo considerable. El costo computacional en general es superior a esquemas netamente DD o MB ya que se tienen que emplear en paralelo 7 modelos de regresión, generadores de residuos y un clasificador.

Tabla 5. RESULTADOS DE LAS PRUEBAS EXPERIMENTALES APLICADAS AL SISTEMA FDI HÍBRIDO

Falla No.	TDP	TDMA	TDMI	FA	TR-FA	FND
1	3.7 Ts	5 Ts	2 Ts	0	-	0
2	3.5 Ts	5 Ts	2 Ts	0	-	0
3	3.5 Ts	5 Ts	2 Ts	0	-	0
4	3.5 Ts	5 Ts	2 Ts	1	2 Ts	0
5	5.2 Ts	6 Ts	2 Ts	1	3 Ts	0
6	5051 Ts	5053 Ts	5050 Ts	1	4 Ts	0
7	1578 Ts	1580 Ts	1575 Ts	2	3 Ts	0
8	5.5 Ts	7 Ts	3 Ts	0	-	0
9	983 Ts	1043 Ts	530 Ts	2	7 Ts	0

REFERENCIAS

- [1] A. S. Willsky, "A survey of design methods for failure detection in dynamic systems," *Automatica*, vol. 12, no. 6, pp. 601-611, 1975.

Tabla 4. EVALUACIÓN DE LOS CLASIFICADORES CONSTRUIDOS PARA LA EVALUACIÓN DE RESIDUOS

	Sensibilidad (%)										Presición
	F1	F2	F3	F4	F5	F6	F7	F8	F9	Normal	Global
SVM	98,4	98,2	96,06	98,7	95,3	75,47	82,5	98,5	87,6	99,8	93,8
NN	99,2	97,2	98,9	97,5	93,2	70,22	84,1	98,7	88,3	99,7	93,48
KNN	92,5	93,5	91,2	89,7	67,4	17,2	43,5	77,8	77,5	94,8	76,76
GMM	99,1	98,3	93,21	98,3	94,5	77,2	90,3	98,2	90,2	99,8	94,57

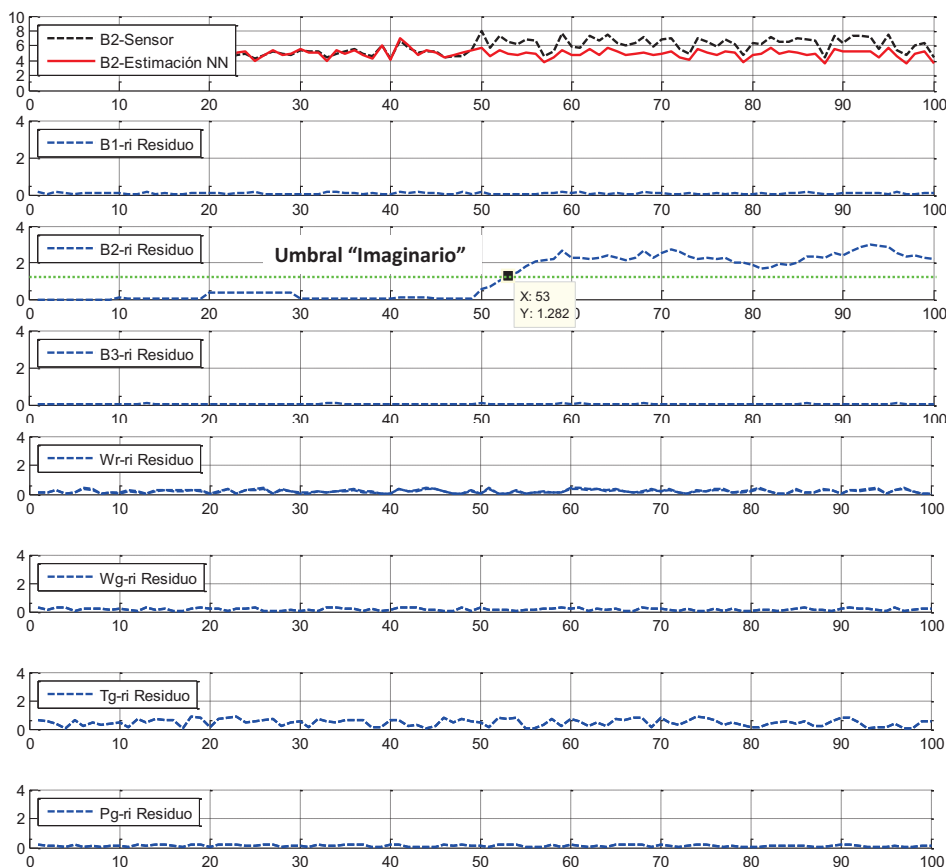


Fig. 5. Ejemplo de conjunto de residuos de entrada al clasificador.

[2] J. Chen and R. J. Patton, *Robust model-based fault diagnosis for dynamic systems*, K.-Y. Cai and C. Beijing University Of Aeronautics Beijing, Eds. Kluwer Academic Publishers, 1999, vol. 11, no. 14.

[3] J. Korbicz, C. Cempel, T. U. of Zielona Gora, and L. S. S. in Zielona Gora, *Analytical and Knowledge-based Redundancy in Fault Detection and Diagnosis*. Technical University Press, 1993.

[4] P. M. Frank, "Analytical and qualitative model-based fault diagnosis - a survey and somenew results," *European Journal of Control*, vol. 2, no. 1, pp. 6-28, 1996.

[5] R. F. Manrique, "Detección y diagnostico de fallas en turbinas de viento usando técnicas de minería de datos," Master's thesis, Universidad Nacional de Colombia, October 2012, máster en Sistemas y Computación. [Online]. Available: <http://www.bdigital.unal.edu.co/10309/>

[6] R. Manrique, F. Giraldo, and J. Esmeral, "Fault detection and diagnosis for wind turbines using data-driven approach," in *Computing Congress (CCC), 2012 7th Colombian*, Oct 2012, pp. 1-6.

[7] C. A. Walford, "Wind turbine reliability: Understanding and minimizing windturbine operation and maintenance costs," *Energy*, no. March, pp. SAND2006-1100, 2006. [Online]. Available: <http://prod.sandia.gov/techlib/access-control.cgi/2006/061100.pdf>

[8] J. Ribrant, "Reliability performance and maintenance a survey of failures in wind power systems?" Master's thesis, KTH School of Electrical Engineering, 2006.

[9] P. F. Odgaard, J. Stoustrup, and M. Kinnaert, "Fault tolerant control of wind turbines - a benchmark model," in *Proceedings of Fault Detection, Supervision and Safety of Technical Processes*, Jun. 2009. [Online]. Available: <http://www.ifac-papersonline.net/Detailed/40082.html>

[10] Y. Li, N. Sundararajan, and P. Saratchandran, "Analysis of minimal radial basis function network algorithm for real-time identification of nonlinear dynamic systems," *Control Theory and Applications, IEE Proceedings -*, vol. 147, no. 4, pp. 476 -484, jul 2000.

[11] I. Samy, I. Postlethwaite, and D.-W. Gu, "Survey and application of sensor fault detection and isolation schemes," *Control Engineering Practice*, vol. 19, no. 7, pp. 658 - 674, 2011.

[12] R. Isermann, "Supervision, fault-detection and fault-diagnosis methods: An introduction," *Control Engineering Practice*, vol. 5, no. 5, pp. 639 - 652, 1997. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0967066197000464>