**XVIII SIMPOSIO INTERNACIONAL DE INGENIERÍA ELÉCTRICA “SIE 2019”**

**Aplicación de las redes neuronales para detección de averías en grupos electrógenos**

***Application of neural networks to detect faults in generators***

**Alvis Cabrera Tejera1, Ernesto Estremera Toledo2, Yoel Dany Blanco Barrueta3**

1. Alvis Cabrera Tejera. Centro Nacional de Electromagnetismo Aplicado, Universidad de Oriente, Cuba. alvita@uo.edu.cu
2. Ernesto Estremera Toledo. Departamento de Control Automático, Universidad de Oriente, Cuba. estremera@uo.edu.cu
3. Yoel Dany Blanco Barrueta. Centro Nacional de Electromagnetismo Aplicado, Universidad de Oriente, Cuba. yoel.blanco@uo.edu.cu

**Resumen:** En este trabajo se presenta la implementación de una red neuronal multicapa con entrenamiento fuera de línea para detectar fallas en el Grupo Electrógeno Fuel Oil donde es usual la ocurrencia de fallos repetitivos por la gran cantidad de variables que intervienen en el proceso. El modelo que se utiliza es una red neuronal Perceptrón multicapas. Se implementa con tres entradas, ocho capas ocultas y tres salidas. La identificación de los pesos óptimos de la red neuronal se realiza fuera de línea a través del algoritmo de aprendizaje de la propagación de errores (Backpropagation) para redes multicapa y una vez que se obtienen estos pesos, la red neuronal se habilita para la detección de fallas en línea. Los valores provenientes de los sensores están en un rango de 0 a 5 V, que se llevan a un microcontrolador donde se programa la red neuronal. Pudiéndose lograr la detección oportuna de fallos y la mejora en eficiencia de la planta.

***Abstract:*** *This paper presents the implementation of a multilayer neural network with off-line training to find faults in the Fuel Oil Generating Group, where the occurrence of repetitive failures is common due to the large number of variables involved in the process. The model used is a red neuronal multilayer Perceptron. It is implemented with three inputs, eight hidden layers and three outputs. The identification of the optimal weights of the neural network is done outside the line through the learning algorithm of the propagation of errors (Backpropagation) for multilayer networks and once these weights are found, the neural network is enabled for the online fault detection. The values coming from the sensors are in a range of 0 to 5 V, which are taken to a microcontroller where the neural network is programmed. Being able to achieve the detection of the fall and the improvement in the efficiency of the plant.*

**Palabras Clave:** Redes neuronales; Grupo Electrógeno; Variables; Eficiencia

***Keywords:*** *Neural networks; Generating Group; Variables; Efficiency*

**1. Introducción**

Los modernos sistemas de control se vuelven cada vez más complejos y los algoritmos de control que se implementan son cada vez más sofisticados. Para los sistemas en los que la seguridad es crítica, las consecuencias de los fallos pueden ser extremadamente serias en términos de vidas humanas, impacto medioambiental y pérdidas económicas; por lo que existe una necesidad creciente en la supervisión en línea y en el diagnóstico de fallos con el objetivo de incrementar la fiabilidad. Para aquellos sistemas donde la seguridad no es crítica, las técnicas de diagnóstico de fallos en línea se pueden utilizar para mejorar la eficiencia, mantenibilidad, disponibilidad y fiabilidad de la planta. Los métodos modernos de diagnóstico de fallos pueden aportar información del estado del sistema que permita implementar un mejor plan de mantenimiento.

Se define fallo todo cambio en el comportamiento de alguno de los componentes del sistema (desviación no permitida de alguna de sus propiedades o parámetros de manera que éste ya no puede satisfacer la función para la cual ha sido diseñado) [1]. Los sistemas de control automático son susceptibles a los fallos pudiendo verse amplificados por el lazo de control llegando a provocar su mal funcionamiento. Además, los lazos de control pueden ocultar los fallos evitando ser observados hasta alcanzar un grado tal que produzcan una avería irreparable que obligue a detener el sistema o proceso. Las redes neuronales artificiales son un paradigma de aprendizaje y procesamiento automático. Se trata de un sistema de interconexión de neuronas en una red que colabora para producir un estímulo de salida. El objetivo es conseguir que las máquinas den respuestas similares a las que es capaz de dar el cerebro que se caracterizan por su generalización y su robustez [2]. Una red neuronal se utiliza para modelar un sistema dinámico no lineal de múltiples entradas y múltiples salidas. Después del entrenamiento, la red neuronal puede dar una estimación muy exacta de la salida del sistema. El entrenamiento de la red se llevó a cabo fuera de línea en MATLAB mediante el algoritmo de retrotropagación de errores y una vez que la red fue entrenada, se implementó en un microcontrolador PIC para que este determine si existe un fallo o no.

**2. Metodología**

Las redes neuronales artificiales imitan su funcionamiento a aquellas que se encuentran en el ámbito biológico. Son aptas para resolver problemas que no poseen un algoritmo claramente definido para transformar una entrada en una salida; aprenden, reconocen y aplican relaciones entre objetos. Las redes neuronales artificiales se basan en el circuito de procesamiento de entradas en el cual los pesos son sumados. En la implementación, las entradas a una neurona son pesadas multiplicando el valor de la entrada por un factor que es menor o igual a uno. El valor de los factores de peso es determinado por el algoritmo de aprendizaje (Figura1).



Figura 1. Arquitectura de la neurona

Las señales que llegan a las dendritas se representan como X1, X2,…, Xn (entradas). Las conexiones sinápticas se representan por unos pesos W1, W2,… Wn que ponderan (multiplican) a las entradas. Si el peso entre las neuronas j e i es positivo representa una sinapsis excitadora, si es negativo representa una sinapsis inhibidora y si es cero no hay conexión. La salida de la neurona se representa por y, la cual se obtiene mediante una función que se denomina función de salida, de transferencia o de activación dada en la ecuación 1:

$y\_{j}=f\left(Net\_{j}-θ\_{j}\right)=f\left(\sum\_{}^{}W\_{ji}X\_{i}-θ\right)$ Ecuación 1

La red neuronal artificial que se muestra en la Figura 2 queda entonces conformada por una red de 4 capas con 5 neuronas en cada una con conexiones completas, es decir todas las entradas se conectan a todas las neuronas dentro de una misma capa.



Figura 2. Esquema de la red neuronal multicapa.

Se utilizan las RNA debido a que estos procesos están caracterizados por la ocurrencia de los fallos repetitivos, presencia de condiciones cambiantes de operación, existencia de datos históricos sobre la ocurrencia de averías y causas que la provocan. Todo lo anterior ofrece las condiciones necesarias para el uso de una red neuronal que permita la detección o diagnóstico de fallos con una mayor eficiencia. Teniendo en cuenta la posibilidad de aprender de las redes neuronales, se deduce que estos problemas que precisan almacenar un gran número de datos requieren para su solución un extenso algoritmo y aprovechan mejor la estructura de una red neuronal, que aquellas que requieren de algoritmos cortos. Con el entrenamiento de gran cantidad de datos históricos de las variables, no habrá posibilidad de que la respuesta de la red sea errónea ya que considera todos los posibles valores, y los que detecte como fuera de rango los dará como fallo y notificará una acción al operario para resolverlo. Con una simple comparación de valores podría darse la posibilidad de algún valor que aunque no esté en el rango no sea proveedor de un fallo ya que la red esta entrenada con datos históricos que podrían incluirlo en tiempo real, y no haber introducido fallo a la planta. Las redes neuronales tienen entre sus características el aprendizaje. Esas reglas quedan almacenadas en las conexiones. Otra característica favorable de la misma es su generalización: Una vez entrenada, se le pueden presentar a la red datos distintos a los usados durante el aprendizaje. La respuesta obtenida dependerá del parecido de los datos con los ejemplos de entrenamiento. Tolerancia a fallos: Debido a que una RNA (red neuronal artificial) almacena la información de forma redundante, ésta puede seguir respondiendo de manera aceptable aun si se daña parcialmente. Flexibilidad: Puede manejar cambios no importantes en la información de entrada, como señales con ruido u otros cambios en la entrada. Tiempo real: La estructura de una RNA es paralela, por lo cual si esto es implementado con computadoras o en dispositivos electrónicos especiales, se pueden obtener respuestas en tiempo real. No linealidad: Esta característica permite modelar procesos intrínsecamente no lineales. Las neuronas son elementos de proceso generalmente no lineales.

**Capa de entrada**: Existen tres variables que están correlacionadas y que son de suma importancia a la hora de diagnosticar y aislar los fallos en el sistema, la selección de las variables se realizó a partir del criterio de expertos, así como la experiencia de los trabajadores en la planta. Para la implementación del modelo neuronal se tomaron como variables más correlacionadas las siguientes:

X1: Temperatura promedio de los gases a la salida de los cilindros.

X2: Presión de aire de descarga.

X3: Temperatura del agua al salir del motor.

Cada una de estas variables tienen rangos determinados, si se salen de dicho rango podrían proporcionar un fallo según sea la variable, pero en tiempo real esto no es tan así pues puede ocurrir que haya valores que no estén en el rango y que no generen fallos pues existe la correlación. Para procesar esos valores se acondiciona las señales provenientes de los sensores empleados, para luego ser convertidas a un código binario mediante un convertidor analógico digital (ADC Analog to Digital Converter por sus siglas en inglés) interno de 10 bits de resolución, el cual se encuentra dentro del PIC

Las lecturas obtenidas por el convertidor son agrupadas para posteriormente ingresarlas al sistema clasificador como vector de entrada a la red neuronal que se encuentra cargada en el mismo PIC.

**Capa de salida**: Para cada una de las variables ocurrirá un fallo si esta es modificada fuera del rango determinado. Es por eso que se nombrarán los siguientes fallos:

F1: fallo en la temperatura de los gases a la salida de los cilindros.

F2: fallo en la presión de aire de descarga.

F3: fallo en la temperatura de salida del motor.

1- Temperatura de los gases a la salida de los cilindros rango de trabajo de 400 a 500ºC. Rango de temperatura que mide el instrumento 0-600ºC. En el Motor de Combustión Interna (MCI), la principal avería tecnológica lo constituyen las válvulas de escape quemadas, por lo que el operador realiza el chequeo de funcionamiento de los cilindros frecuentemente partiendo de los datos de temperatura. Este fallo se localiza en la salida de los cilindros siendo de gran importancia tenerlo en el rango determinado puesto que el instrumento puede medir menos o mucho más que este rango, sin embargo no estar funcionando correctamente.

2- Presión del aire de carga: rango de trabajo 2.0 - 3.2 bar. Rango que mide el instrumento 0-6 bar. Si la presión medida está por debajo de los 2.5 bar. Se debe verificar primeramente si la temperatura de los gases de escape aumentó. Si la presión en los cilindros disminuyó y también si la presión del FO (fuel oil) aumentó. En caso de que ocurriese un fallo esto provocaría: Un consumo excesivo de combustible por kWh generado provocando baja eficiencia y desperdicio del preciado combustible además de provocar un calentamiento del motor generador.

3- Temperatura del agua a la salida de los motores: rango de trabajo 75 - 85ºC. Rango que mide el instrumento 0-200 ºC. Si la temperatura medida baja de los 75 ºC se debe verificar: Si el motor está trabajando por debajo de la temperatura de trabajo. Si la temperatura del LO (Aceite lubricante) es baja. Si la presión del FO es baja. Si la presión del LO es alta. Si la temperatura de los gases de escape es baja. Si la temperatura medida sube de los 85 ºC se debe verificar: Si el motor está trabajando por encima de la temperatura de trabajo. Si la presión del FO es alta. Si la presión del LO es baja. Si la temperatura del LO es alta. Si la temperatura de los gases de escape es alta. Si el nivel en el tanque de agua de enfriamiento es bajo.

**Entrenamiento de la Red Neuronal**: La Backpropagation es una de las redes más usadas, incluso en la actualidad. El algoritmo de aprendizaje de la red Backpropagation es supervisado, el que utiliza un ciclo propagación – adaptación de dos fases. Una vez que se ha aplicado un patrón a la entrada de la red como estímulo, este se propaga desde la primera capa a través de las capas superiores de la red, hasta generar una salida. La señal de salida se compara con la salida deseada y se calcula una señal de error para cada una de las salidas. Las salidas de error se propagan hacia atrás, partiendo de la capa de salida, hacia todas las neuronas de la capa oculta que contribuyen directamente a la salida; las neuronas de la capa oculta solo reciben una fracción de la señal total del error, basándose aproximadamente en la contribución relativa que haya aportado cada neurona a la salida original. Este proceso se repite, capa por capa, hasta que todas las neuronas de la red hayan recibido una señal de error que describa su contribución relativa al error total. Basándose en la señal de error percibida, se actualizan los pesos de conexión de cada neurona, para hacer que la red converja hacia un estado que permita clasificar correctamente todos los patrones de entrenamiento [3].

 Este proceso tiene una característica muy importante, y es que a medida que se entrena la red, las neuronas de las capas intermedias se organizan a sí mismas de tal modo que las distintas neuronas aprenden a reconocer distintas características del espacio total de entrada. Después del entrenamiento, cuando se les presente un patrón arbitrario de entrada que contenga ruido o que esté incompleto, las neuronas de la capa oculta de la red responderán con una salida activa si la nueva entrada contiene un patrón que se asemeje a aquella característica que las neuronas individuales hayan aprendido a reconocer durante su entrenamiento. Para realizar el proceso de aprendizaje se debe inicialmente tener definida la topología de la red: número de neuronas en la capa de entrada el cual depende del número de componentes del vector de entrada, cantidad de capas ocultas y número de neuronas en cada una de ellas, número de neuronas en la capa de salida el cual depende del número de componentes del vector de salida o patrones objetivos y funciones de transferencia requeridas en cada capa, con base en la topología escogida se asignan valores iniciales a cada uno de los parámetros que conforman la red. El término backpropagation se refiere al método para calcular el gradiente de error en una red feed forward [4]. La red neuronal Backpropagation presenta una gran variedad de opciones de configuración, dependiendo de la necesidad de aprendizaje y de la aplicación que se esté desarrollando. Para el aprendizaje se emplea la herramienta Matlab, la cual cuenta con un toolbox de Redes Neuronales: newff: crea una red tipo Perceptrón con algoritmo de entrenamiento Backpropagation, para configurar esta red se requiere que sean especificados algunos parámetros.

 El algoritmo utilizado para el entrenamiento de la red neuronal en cuestión es Trainlm. Este algoritmo actualiza los pesos y las ganancias de acuerdo a la optimización de Levenberg-Marquardt. Es el algoritmo más rápido para redes Backpropagation, tiene como desventaja que requiere de un conjunto de entrenamiento lo más estándar posible, pues de otra forma sólo aproximará correctamente valores que se encuentren dentro de los patrones de aprendizaje [5].

 Es necesario conocer además la topología de la red que se quiere crear, lo cual consiste en definir la cantidad de capas y la cantidad de neuronas de entrada, de salida.

En cuanto a la capa oculta este número de neuronas es aproximadamente 2n, donde n es el número de neuronas de la capa de entrada. Una vez realizado el entrenamiento se toman los valores de los pesos y de los umbrales para luego programar el PIC.

Implementación en un microcontrolador: La estructura de la red neuronal y los valores que resultaron del entrenamiento se ingresaron a un microcontrolador PIC18F452. Este microcontrolador se eligió principalmente porque tiene la capacidad de manejar datos de tipo flotante y un gran espacio para datos. Entre las características principales del microcontrolador se encuentran:

• Convertidores A/D (10 bits de resolución).

• Canales con modulación por ancho de pulso (PWM).

• Temporizadores de 8 y 16 bits (hasta 3 temporizadores por dispositivo).

• Instrucciones a 16 bit.

 El microcontrolador fue programado en lenguaje C utilizando el IDE MPLAB de Microchip. Una vez declaradas las variables a utilizar, el programa se ejecuta como lo muestra el siguiente pseudocódigo.

1. Se cargan las matrices de pesos y umbrales obtenidos del entrenamiento de la red en el Matlab.

2. Se configura e inicializa los puertos a utilizar.

3. Se desactivan las interrupciones y se configura al ADC.

4. Se convierten las tres entradas de voltaje al conversor y se llevan a variables ingenieriles (temperatura, presión).

5. Se introducen los valores a la red, la cual está programada.

6. Se calcula el error entre la salida de la red y el valor entrado.

7. Se determina si ese error está fuera de los límites para activar una alarma de fallo.

**3. Resultados y discusión**

Con la realización de este trabajo se generó un análisis minucioso de las diferentes variables con mayor ocurrencia de fallos en este tipo de plantas y las variables correlacionadas con ellas, con la ayuda de expertos y operarios de la planta. Se generó una red neuronal con gran cantidad de datos históricos reales de dichas variables lo que permitió un entrenamiento confiable. Y finalmente se le introdujo esta red neuronal al PIC y realizar las tareas de control pertinentes. Con la ayuda de las RNA se logrará un control eficiente de la planta y por su puesto el ahorro de combustible.

**4. Conclusiones**

En este trabajo se presenta el desarrollo de un sistema basado una red neuronal multicapa con entrenamiento fuera de línea para para detección de fallos. Se concluye que las Redes Neuronales Artificiales representan una técnica adecuada para la resolución del problema de diagnóstico y detección de fallos. Se hace énfasis en que el uso de redes neuronales puede extenderse a otros campos de aplicación por lo que el sistema desarrollado puede modificarse o complementarse para trabajos futuros ya sea sobre la misma problemática o tomar el procesamiento de redes neuronales como método de solución a otros problemas e implementarlo para alguna otra aplicación.

**5. Referencias bibliográficas**

[1] Arafet, P. y otros, “Monografía sobre identificación de sistemas”. Universidad de Oriente, Santiago de Cuba, Cuba, 2004.

[2] Bello, R. “Curso Introductorio a las Redes Neuronales Artificiales”. Universidad Central de Las Villas, Cuba, 2003.

[3] Redes Neuronales: Conceptos Básicos y Aplicaciones.

http://www.dynamics.unam.edu/DinamicaNoLineal/Articlos/MineriaDatos

[4] Llosas, Yolanda. “Control Inteligente”. Monografía curso de posgrado. Universidad de Oriente. Cuba, 2004.

[5] Blázquez L. Felipe; de Miguel Luis J. “Diagnóstico automático de fallos para sistemas dinámicos no lineales”. Universidad de Valladolid.

[6] Galhardo R. Fernández, Diego Rodrigo Cabral Silva, “Faults Detection and Isolation Based On Neural Networks Applied to a Levels Control System”, 2007.

[7] Vilaragut, M. “Curso de posgrado: Redes Neuronales Artificiales” 2005.