# Prototipo de localizador de faltas, basado en redes neuronales, para su operación en tiempo real

A.J. Mazón, I. Zamora , V. Valverde, E. Torres, I. Albizu

Departamento de Ingeniería Eléctrica
E.T.S.I. de Bilbao (UPV/EHU)
Alda de Urquijo s/n – 48013 Bilbao (España)
Tel.:+34 946014062/63, fax:+34 946014200
e-mail: **iepmasaj@bi.ehu.es**, **iepzabei@bi.ehu.es**, **iepvasav@bi.ehu.es**, **ieptoige@bi.ehu.es**, **iepalfli@bi.ehu.es**

**Resumen.** En el campo de la ingeniería eléctrica, hay muchas áreas dónde se están aplicando las redes neurales artificiales (ANNs). Las redes neurales artificiales pueden se pueden simular, por medio de la ejecución de un software específico, en ordenadores secuenciales. No obstante, si se desea aprovechar su capacidad de cálculo masivo en paralelo, permitiendo su aplicación en tiempo real, es necesaria la implementación en un hardware específico. Un primer paso para resolver este problema consiste en llevar a cabo la implementación mediante "técnicas de prototipado rápido". En este sentido, la aplicación informática Matlab ofrece una herramienta, denominada xPC-Target, que permite simular los sistemas en tiempo real con objeto de llevar a cabo el desarrollo de prototipos. En esta ponencia se presenta el proceso de desarrollo del prototipo de una red neuronal del tipo perceptron multicapa (MLP), usando esta herramienta. El prototipo obtenido se ha verificado satisfactoriamente en el proceso de localización de faltas en líneas de transporte de energía eléctrica.

**Palabras llave**

Redes neuronales, prototipado rápido, simulación en tiempo real.

## 1. Introducción

Cuando se produce una falta permanente en una línea de transporte de energía eléctrica, es necesario realizar las tareas de reparación de la forma más rápida y económica posible. Para ello es necesario determinar la posición en la cual se ha producido la falta.

Los sistemas de localización se pueden clasificar en función del lugar donde se captan las medidas eléctricas en:

- Métodos que utilizan magnitudes medidas en un solo extremo de la línea
- Métodos que utilizan magnitudes medidas en los dos extremos de la línea

Los sistemas que utilizan valores medidos en ambos extremos de la línea dan resultados más exactos y más rápidos que aquellos que solo utilizan valores medidos en un extremo. Sin embargo, precisan que las medidas tomadas en los dos extremos de la línea estén sincronizadas, por lo que el proceso de toma de datos es más complejo. Dado que, además de precisión y rapidez, se requiere que los sistemas de localización de faltas sean económicos, a igualdad de condiciones, se prefieren sistemas implementados en un solo extremo, ya que permiten simplificar el proceso de localización de la falta.

El desarrollo de los sistemas informáticos ha permitido la modelización de los sistemas eléctricos de forma que se pueden representar condiciones cada vez más similares a las que se dan en la realidad. Sin embargo, son tantos los factores que influyen en los parámetros eléctricos durante una falta, que es muy difícil incluir todos ellos en el modelo: tipo de falta, posición de la falta, resistencia de falta, estados del sistema eléctrico en el momento en que se produce la falta, etc.

Las redes neuronales artificiales surgen como un método de resolución de problemas que no se pueden describir fácilmente mediante un enfoque algorítmico tradicional. Este método tiene su origen en la emulación de sistemas biológicos. Se trata de una forma de computación que es capaz de manejar las imprecisiones e incertidumbres de problemas relacionados con el mundo real. En la actualidad existen diferentes estudios sobre análisis de faltas aplicando redes neuronales, los cuales se pueden clasificar en tres grandes grupos: detección de faltas [1-4], clasificación de faltas [5-9] y localización de faltas [10-14].

Por otra parte, la mejora de los procesos de producción y la creciente competencia entre empresas provoca una necesidad cada vez mayor de nuevos productos con nuevas características y prestaciones. Esto hace que los departamentos de diseño deban encontrar nuevas soluciones a los problemas ya existentes y emplear nuevas técnicas de desarrollo para poder satisfacer dicha demanda.

Las técnicas del prototipado rápido forman parte de las nuevas técnicas de desarrollo de producto. Se basan en introducir metodologías de diseño que permitan realizar un modelado del problema con un nivel de abstracción superior permitiendo reducir el ciclo de desarrollo del nuevo producto. El mercado ofrece al usuario numerosas

herramientas para realizar estos procesos de prototipado rápido, entre las que destaca la ofrecida por Mathworks Inc denominada xPC Target [15]

Tal como se muestra en la presente ponencia, la combinación de las redes neuronales artificiales con la herramienta xPC Target genera resultados altamente satisfactorios en la resolución de problemas como la clasificación y localización de faltas en sistemas eléctricos de transporte de energía eléctrica,

## 2. Prototipado de redes neuronales

### *A. Prototipado rápido de redes neuronales*

Las técnicas de prototipado rápido constituyen un paso intermedio en la implementación de redes neuronales. Se encuentran a medio camino entre la simulación software, mas económica y frecuente, y la implementación hardware, que permite obtener las mejores prestaciones de la capacidad de cálculo paralelo de las redes neuronales. El desarrollo de herramientas como el xPC Target de Matlab permiten agilizar las tareas iniciales del proceso de diseño de cualquier prototipo, con el consiguiente ahorro de tiempo y dinero.

El xPC Target es una aplicación cliente/servidor (Target/Host) que permite conectar los modelos creados en Simulink con sistemas físicos, y simularlos en tiempo real. xPC Target permite añadir bloques input/output a los modelos de Simulink, generar código con Real-Time Workshop y descargar dicho código a un segundo PC que opera el núcleo en tiempo real del xPC Target.

El xPC Target permite realizar un prototipado rápido con el que probar los diseños realizados en el hardware destino, monitorizar en tiempo real los resultados obtenidos y cambiar, en función de estos, los distintos parámetros y variables de diseño. En definitiva posibilita ejecutar modelos en tiempo real en un hardware PC estándar.

La aplicación se desarrolla en la máquina Host, la cual es un PC de escritorio en el que se encuentran instaladas las siguientes herramientas de Mathworks: MatLab, Simulink, Real-Time Workshop y xPC Target. Además es necesario disponer de un compilador de C. En este PC se crean los modelos usando bloques de Simulink y se genera el código correspondiente. Posteriormente, este código se ejecuta en tiempo real en un segundo PC (Target). Tanto el Host como el Target no requieren ningún hardware específico.

El Target PC arranca usando un disquete creado previamente por el xPC Target toolbox que contiene el sistema operativo para trabajo en tiempo real. Una vez cargado el sistema operativo, desde el Host PC se descarga el código generado a partir del modelo de Simulink, ya sea vía red o a través del puerto serie.

De entre las ventajas que proporciona la utilización de la herramienta xPC Target caben destacar las siguientes [16]:

- Se facilita la tarea de programación de modelos complejos con la utilización de la herramienta Simulink
- xPC Target funciona como un sistema operativo en tiempo real, lo que hace que el modelo se comporte como un proceso real.
- Para el funcionamiento en tiempo real del modelo no es necesaria una programación de bajo nivel de las tareas, ya que la generación del código se realiza automáticamente.
- El sistema no requiere de un hardware específico, basta con un PC de escritorio con una tarjeta de adquisición de datos o una tarjeta ethernet.

### *B. Prototipado de un MLP*

El perceptrón multicapa es el modelo neuronal más popular y empleado en la práctica. Es una red unidireccional que consta de una capa de entrada, una capa de salida y un número indeterminado de capas ocultas. Las neuronas de cada capa están conectadas a todas las neuronas de la capa anterior, tal como se muestra en la figura 1

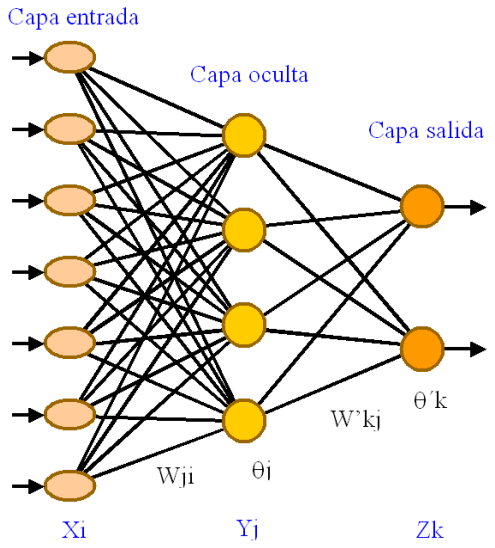


Fig. 1. Estructura de la red MLP

Matemáticamente, la operación de una red MLP con una capa oculta y neuronas de salida lineales se expresa como se indica en la expresión (1).

$$z_k = \sum_j w'_{kj} \cdot y_j - \theta'_i = \sum_j w'_{kj} \cdot f\left(\sum_i w_{ji} \cdot x_i - \theta_j\right) - \theta'_i \quad (1)$$

La figura 2 muestra el diagrama de bloques que se debe seguir en el proceso de prototipado de la red MLP.

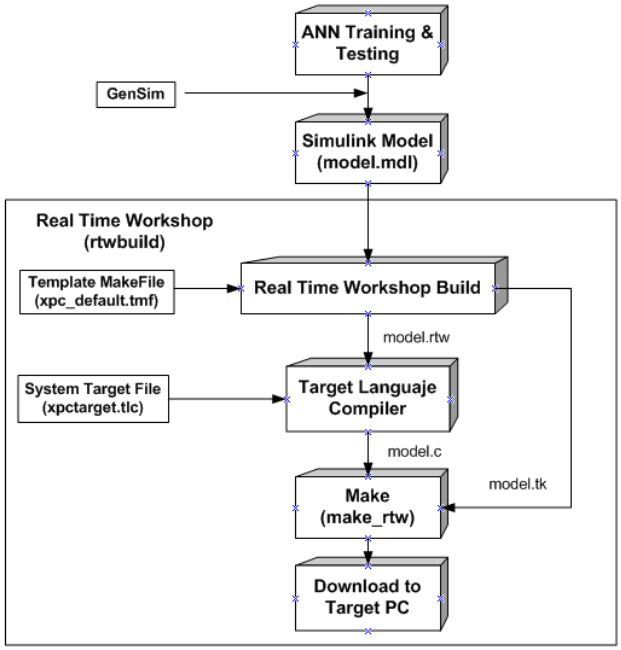


Fig. 2. Process of MLP prototyping

Para comenzar el proceso de prototipado de la red MLP con el xPC Target, es necesario realizar en primer lugar el modelado de esa red neuronal en Simulink y generar el código que ha de ser descargado en el Target PC. Así, una vez creada y entrenada la red neuronal artificial, MatLab dispone de la función "gensim" para incorporarla a los modelos de Simulink. Esta función "gensim" genera automáticamente los bloques de simulación de la red neuronal y permite visualizarlos de forma gráfica.

Tras incorporar dichos bloques al modelo de Simulink, se puede iniciar el proceso de compilado y transferencia a la máquina definida como Target. Para ello, MatLab hace uso de la herramienta Real-Time Workshop (RTW) que genera el código necesario, para su ejecución posterior en el Host PC, a partir de modelos Simulink creados anteriormente con la función "gensim".

Para que RTW genere el código de forma correcta, antes del comienzo del proceso de compilado y descarga del modelo, se deben especificar una serie de parámetros:

- Se debe definir el "System Target File", cuya misión es la de indicar cómo se genera el código para la aplicación xPC Target. El archivo escogido se denomina "xpctarget.tlc" y es el punto de entrada para que el Target Language Compiler cree el ejecutable.

- Es necesario seleccionar además un "Template Makefile". Para crear el ejecutable, el generador de código utiliza plantillas de archivo makefile. RTW genera un makefile con el nombre del modelo que se quiere compilar a partir de la plantilla seleccionada. Este archivo makefile sirve de base para crear el ejecutable. La plantilla seleccionada se denomina "xpc_default.tmf". El proceso es gobernado por una función escrita en código ".m" de alto nivel denominada "make_rtw".

Una vez definidos estos parámetros, se inicia la generación del código y su posterior transferencia al Target PC empleando la función "rtwbuild". El código creado se envía al Target PC utilizando bien el protocolo de transferencia de datos TCP/IP o el RS232.

## 3. Prototipo de localizador de faltas

### *A. Estructura de red de clasificación*

El localizador de faltas prototipado realiza una clasificación previa de la falta mediante el análisis independiente de cada una de las fases de la línea en falta, de forma que se puede obtener si la fase analizada está en falta o no lo está, a partir de los módulos de la componente fundamental de la tensión y de la intensidad. Concretamente, se parte de los valores de tensión e intensidad de fase, durante la falta, medidos en un extremo de la línea, expresados en magnitudes relativas a la situación de prefalta, es decir ($V/V^{pf}$) e ($I/I^{pf}$). Así, en el caso de líneas simples de dos extremos, la red de clasificación tiene la estructura que se muestra en la Figura 3.

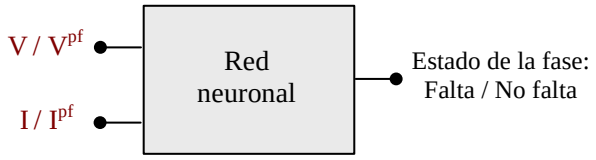


Fig. 3. Estructura de red para clasificación

La estructura de red neuronal seleccionada es un Perceptrón Multicapa que utiliza un algoritmo de entrenamiento de retropropagación de errores (backpropagacion o BP) con optimización de Levenber-Marquardt. Este algoritmo reduce el tiempo de entrenamiento, presentando un error prácticamente nulo y permitiendo el entrenamiento con una muestra de datos aleatorios sin requerir preparación previa de los mismos.

En cuanto a la topología de red más adecuada, es posible considerar diversas alternativas sin que se aprecien diferencias significativas en sus resultados, siempre que sean redes MLP con la capa de salida lineal. Es conveniente que el número de neuronas no sea excesivo para no incrementar innecesariamente el tiempo de entrenamiento, por lo que no se recomienda utilizar redes con más de seis neuronas en las capas ocultas.

### *B. Estructura de red de localización*

Para resolver el problema de la localización de faltas, el prototipo de localizador de faltas desarrollado emplea los módulos de las tensiones e intensidades de la línea en falta, expresados en valores relativos a la situación de prefalta, es decir ($V/V^{pf}$) e ($I/I^{pf}$). Así, la red neuronal

encargada de la localización de la falta tiene, para el caso de una falta monofásica, la estructura que se muestra en la Figura 4, dando como resultado la posición y la resistencia de falta.

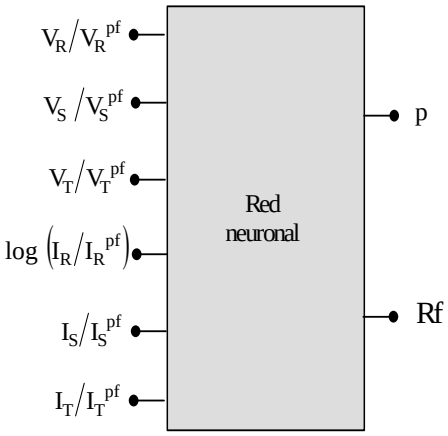


Fig. 4. Estructura de la red para la determinación de la distancia y la resistencia de falta

Por la estructura y características del problema a resolver, la estructura de red seleccionada es, de nuevo, el perceptrón multicapa con algoritmo de entrenamiento BP con optimización de Levenber-Marquardt.

En general, con 8 ó 9 neuronas en la primera capa y entre 4 y 6 neuronas en la segunda, los resultados de las estructuras MLP en la determinación de la posición y resistencia de la falta son muy satisfactorios.

### *C. Creación del prototipo*

Empleando la metodología descrita en el apartado 2 se ha realizado el prototipado de las estructuras de redes empleadas en los procesos de clasificación y localización descritos.

Para ello se ha desarrollado un sistema emulador de faltas basado en dos módulos: El primer módulo se ejecuta en el Host PC y se encarga de obtener las medidas de tensión e intensidad efectuadas en la línea que se quiere monitorizar. El segundo módulo se ejecuta en el Target PC, y se encarga de procesar los datos de tensión e intensidad enviados por el Host PC, y de proporcionar la clasificación y la localización de la falta. A modo de ejemplo, la figura 5 muestra el módulo que se ejecuta en el Target PC. Como se puede apreciar, el modulo contiene los bloques encargados de implementar la clasificación de la falta (una red neuronal para analizar cada fase). Así mismo, el modulo implementa la localización de la falta mediante tres redes neuronales, una para cada tipo de falta (monofásica, bifásica o trifásica).

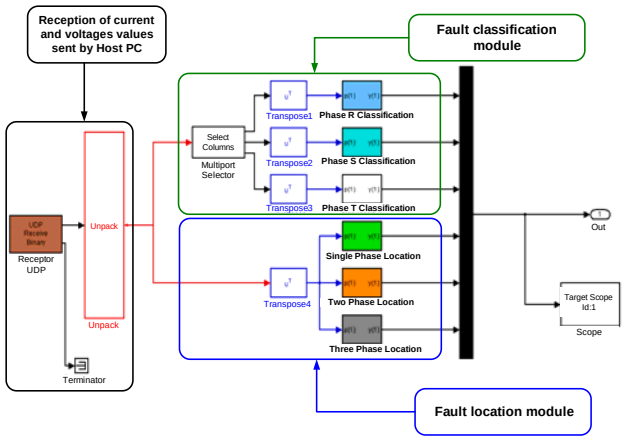


Fig. 5. Modelo ejecutado en el Target PC

## 4. Resultados

En este apartado se presenta un ejemplo completo de clasificación y localización de faltas en la línea 'La Lomba-Herrera' para cada tipo de falta utilizando el prototipo desarrollado. Esta línea aérea de transporte de energía eléctrica tiene una tensión nominal de 380 kV y una longitud de 189,30 km. Las faltas analizadas son las que se muestran en la Tabla 1.

TABLA 1. Faltas analizadas en la línea "La lomba-Herrera"

| Línea LA LOMBA - HERRERA | | |
|---|---|---|
| Tipo de Falta | Distancia de Falta (%) | Resistencia de Falta (Ω) |
| RG | 18,5 | 27,3 |
| RS | 32,7 | 16,2 |
| RSG | 87,3 | 22,5 |
| RST | 50,2 | 7 |

Para cada una de las faltas descritas en de la Tabla 1, se obtienen los valores de las magnitudes de tensión e intensidad de falta, que se muestran en la Tabla 2, medidos en el extremo "La Lomba",. Estos valores, son suministrados como entradas al localizador de faltas para que éste proporcione el valor de la distancia y resistencia de falta.

TABLA 2. Valores de las tensiones e intensidades

| **Línea LA LOMBA - HERRERA** | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Tipo de Falta** | **Dist (%)** | **Resis (Ω)** | **$V_R$ (kV)** | **$V_S$ (kV)** | **$V_T$ (kV)** | **$I_R$ (A)** | **$I_S$ (A)** | **$I_T$ (A)** |
| **RG** | 18,5 | 27,3 | 212,3 | 243,2 | 238,7 | 5383,8 | 593,4 | 505,3 |
| **RS** | 32,7 | 16,2 | 205,2 | 175,8 | 235,5 | 7132,8 | 6591,3 | 566,2 |
| **RSG** | 87,3 | 22,5 | 199,1 | 186,0 | 234,3 | 3085,9 | 2778 | 566,1 |
| **RST** | 50,2 | 7 | 169,2 | 169,2 | 169,2 | 6059,4 | 6059,4 | 6059,4 |

A partir de los valores de tensión e intensidad correspondientes a la falta monofásica, el localizador realiza correctamente la clasificación de la misma. Es decir, la red neuronal encargada de la clasificación comprueba, de forma independiente, si cada una de las fases está en falta (1) ó no (0). El resultado de este análisis proporciona el tipo de falta indicado en la tabla 3.

TABLA 3. Resultado de la clasificación. Falta monofásica

| Fase | V (kV) | I (A) | Resultado | Estado | Tipo de Falta |
|---|---|---|---|---|---|
| R | 212,31 | 5383,8 | 1 | FALTA | RG |
| S | 243,21 | 593,4 | 0 | NO FALTA | |
| T | 238,71 | 505,3 | 0 | NO FALTA | |

Una vez clasificada la falta, los valores de tensión e intensidad se emplean como entradas de la red neuronal encargada de la localización de la falta. La tabla 4 muestra los resultados de distancia y resistencia de falta proporcionados por el localizador prototipado, así como la comparación con los valores definidos en la tabla 1.

TABLA 4. Resultado de la localización. Falta monobásica

| | Distancia de falta | | Resistencia de falta |
|---|---|---|---|
| | % | Km. | Ω |
| **Valores esperados** | 18,50 | 35,020 | 27,30 |
| **Valores obtenidos** | 18,47 | 34,963 | 27,39 |
| **Error** | - 0,03 | - 0,057 | + 0,09 |

En el caso de la falta bifásica, la red neuronal encargada de la clasificación es capaz de determinar que existen dos fases en falta, tal como se indica en la tabla 5.

TABLA 5. Resultado de la clasificación. Falta bifásica

| Fase | V (kV) | I (A) | Resultado | Estado | Tipo de Falta |
|---|---|---|---|---|---|
| R | 205,21 | 7132,8 | 1 | FALTA | RS / RSG |
| S | 175,84 | 6591,3 | 1 | FALTA | |
| T | 235,53 | 566,2 | 0 | NO FALTA | |

Para poder discernir si la falta es bifásica o bifásica a tierra, el localizador calcula la componente homopolar de las corrientes de falta. En caso de que sea cero, la falta es bifásica, mientras que, en caso contrario, la falta es bifásica a tierra.

En la tabla 6 se muestra el resultado de la localización de la falta bifásica clasificada

TABLA 6. Resultado de la localización. Falta bifásica

| | Distancia de falta | | Resistencia de falta |
|---|---|---|---|
| | % | Km. | Ω |
| **Valores esperados** | 32,70 | 61,901 | 16,2 |
| **Valores obtenidos** | 32,69 | 61,882 | 16,19 |
| **Error** | - 0,01 | - 0,019 | - 0,01 |

Finalmente, las tablas 7 a 10 muestran los resultados de la clasificación, la localización de la distancia y la determinación de la resistencia de falta en el caso de las faltas bifásica a tierra y trifásica indicadas en la tabla 1.

TABLA 7. Resultado de la clasificación. Falta bifásica a tierra

| Fase | V (kV) | I (A) | Resultado | Estado | Tipo de Falta |
|---|---|---|---|---|---|
| R | 199,17 | 3085,9 | 1 | FALTA | RS / RSG |
| S | 186,09 | 2778 | 1 | FALTA | |
| T | 234,35 | 566,1 | 0 | NO FALTA | |

TABLA 8. Resultado de la localización. Falta bifásica a tierra

| | Distancia de falta | | Resistencia de falta |
|---|---|---|---|
| | % | Km. | Ω |
| **Valores esperados** | 87,30 | 165,259 | 22,5 |
| **Valores obtenidos** | 87,29 | 165,240 | 22,51 |
| **Error** | - 0,01 | - 0,019 | + 0,01 |

TABLA 9. Resultado de la clasificación. Falta trifásica

| Fase | V (kV) | I (A) | Resultado | Estado | Tipo de Falta |
|---|---|---|---|---|---|
| R | 166,24 | 6059,4 | 1 | FALTA | RST |
| S | 166,24 | 6059,4 | 1 | FALTA | |
| T | 166,24 | 6059,4 | 1 | FALTA | |

TABLA 10. Resultado de la localización. Falta trifásica

| | Distancia de falta | | Resistencia de falta |
|---|---|---|---|
| | % | Km. | Ω |
| **Valores esperados** | 50,20 | 98,436 | 7 |
| **Valores obtenidos** | 52,00 | 95.028 | 7,08 |
| **Error** | - 1,8 | -3.407 | + 0,08 |

Resultados similares a los mostrados en este ejemplo han sido obtenidos para otras líneas del sistema eléctrico

español. Por lo tanto, se puede considerar que las redes neuronales prototipadas operan correctamente, lo que permite que se puedan implementar en un hardware específico que se ajuste a las directrices marcadas por el prototipo establecido.

## 4. Conclusiones

Las redes neuronales artificiales se aplican en diversos campos de la ingeniería eléctrica con excelentes resultados. No obstante, como paso previo a la realización de una implementación hardware, es necesario estudiar su comportamiento real mediante el desarrollo de un prototipo, que se puede llevar a cabo mediante las técnicas de prototipado rápido. La presente ponencia muestra el proceso de prototipado rápido de un localizador de faltas, basado en redes neuronales, empleando el entorno MatLab/xPC Target.

El xPC Target es una aplicación comercial que permite llevar a cabo el proceso de prototipado a partir de una configuración cliente-servidor, con la que se puede monitorizar los resultados en tiempo real y establecer las características fundamentales de la red a implementar.

## Agradecimientos

El trabajo presentado en esta ponencia ha sido desarrollado por el equipo investigador que lleva a cabo los proyectos UPV 00142-345-15300/2003, financiado por la UPV/EHU, y DPI2003-02616 financiado por el MCYT.

## Referencias

[1] A.K. Pradhan. P.K. Dash. G. Panda. *A fast and accurate distance relaying scheme using an efficient radial basis function neural network.* Electric Power System Research. Vol. 60. 2001. pp. 1-8.

[2] S. Vasilic. M. Kezunovic. *New Design of a Neural Network Algorithm for Detecting and Classifying the Transmission Line Faults*. IEEE PES Transmission & Distribution Conference. Atlanta. Oct. 2001

[3] F. Zahra. B. Jeyasurya. J.E. Quaicoe. *High-speed transmission line relaying using artificial neural networks.* Electric Power Systems Research. 53. 2000. p.173-179

[4] S.A. Khaparde. N. Warke. S.H. Agarwal. *An adaptive approach in distance protection using an artificial neural network*. Electric Power Systems Research. 37. 1996. p.39-44

[5] Y.H. Song. Q.X. Xuan. A.T. Johns. *Comparison Studies of Five Neural Network Based Fault Classifiers for Complex Transmission Lines*. Electric Power Systems Research. 43.1997. p.125-132

[6] R.K. Aggarwal. Q.Y. Xuan. R.W. Dunn. *A novel fault classification technique for double circuit lines based on a combined unsupervised/supervised neural network*. IEEE Transactions on Power Delivery. Vol. 14. Nº 4. October 1999. p.1250-1256

[7] W.M. Lin. C.H. Yang. J.H. Lin. M.T. Tsay. *A fault classification method by RBF neural network with OLS learning procedure* . IEEE Transactions on Power Delivery. Vol. 16. Nº 4. Oct 2001. p.473-477

[8] M. Kezunovic. S. Vasilic. F. Gul-Bagriyanik. *Advanced approaches for detecting and diagnosing transients and faults.* Med Power 2002. Athens. Greece. November 2002.

[9] S. Vasilic. M. Kezunovic. *An Improved Neural Network Algorithm for Classifying the Transmission Line Faults*. IEEE PES Winter Meeting. January 2002.

[10] R.K. Aggarwal. M. Joorabian. Y.H. Song. *Fuzzy neural network approach to accurate transmission line fault location.* Engineering Intelligent Systems. Vol. 5. Nº 4. December 1997. p.251-258

[11] Z. Chen. J.C. Maun. *Artificial neural network approach to single-ended fault locator for transmission lines.* IEEE Transactions on Power Systems. Vol. 15. Nº 1. February 2000. p.370-375

[12] A.J. Mazón. I. Zamora. J.F. Miñambres. M.A. Zorrozúa. J.J. Barandiarán. K. Sagastebeitia. *A new approach to fault location in two-terminal transmission lines using artificial neural networks.* Electric Power System Research. Vol. 56. 2000. p. 261-266.

[13] A.J. Mazon. I. Zamora . J. Gracia. J. Bilbao. J.R. Saenz. *Falneur: Artificial neural network based software to fault location in electrical transmission lines.* IASTED International Conference on Applied Informatics (AI2001). February 2001. Innsbruck. Austria.

[14] D.V. Coury. M. Oleskovicz. R.K. Aggarwal. *An ANN routine for fault detection. classification and location in transmission lines.* Electric Power Components and Systems. November 2002. p.1137-1149

[15] The Mathworks Inc. xPC Target, User´s Guide, Version 2.0.1. September 2003.

[16] [2] D. Burns and T.G. Sugar. *Rapid Embedded Programming in the Mathworks Environment.* Journal of Computing and Information Science in Engineering. Vol 2, pp 237-241 September 2002

[17] A.J. Mazon. I. Zamora. J.Gracia. K.J. Sagastebeitia. J.R. Saenz. *Selecting ANN structures to find transmission faults.* IEEE Computer Applications in Power. Vol 14. nº3. Julio 2001. pag 44-48

[18] M. Kezunovic. et al. *Intelligent Systems in Protection Engineering*. Working Group C-4. IEEE Power Engineering Society Power System Relaying Committee Systems Protection Subcommittee. Draft February 1999.

[19] H. Demuth. *Neural Network ToolBox For Use with MatLab*. 1998.