SIMULACIÓN DE MODULADORES / DEMODULADORES, UTILIZANDO LA TEORÍA DE REDES NEURONALES

Adriana Cisneros.
Universidad Rafael Belloso Chacín. Venezuela

RESUMEN

En el presente trabajo se desarrollan modelos para la simulación de moduladores / demoduladores utilizando la teoría de redes neuronales. Para ello se generan los datos de entrada / salida a la red neuronal a través del modelo matemático que describe los diversos moduladores / demoduladores, y se busca una aproximación al comportamiento real de los mismos, mediante el desarrollo de redes del tipo Backpropagation. El algoritmo elegido para trabajar con estas redes es el algoritmo de Levenverg Marquardt. Para encontrar el modelo más óptimo de cada modulador, se realizó una comparación del error medio cuadrático obtenido de la combinación de diversas funciones de transferencia en la capa oculta y de salida, se fijó el número de capas e iteraciones a la red neuronal, y se eligió como moduladores / demoduladores aquellos modelos en los cuales el error medio cuadrático era mínimo. La herramienta utilizada para generar los datos y el desarrollo de los diversos modelos de esta investigación, fue el Toolbox Neural Network del Matlab, versión 6.0.

Palabras claves: moduladores / demoduladores, redes neuronales, algoritmo, error medio cuadrático.

ABSTRACT

The main purpose of this investigation is to develop models for the simulation of modulators/demodulators using the theory of neural network. To reach this goal input/output data from neural network is generated by means of mathematical model that describes different modulators/demodulators. In order to look for their approximate actual behavior through. the development of network from Backpropagation types. The selected algorithm to work with these kind of networks is the Levenverg Marquardt. To find the most optimal model from each modulator, a comparison was carried out of the half-squared error found from the combination of the different functions of transference in the hidden and outer shell. A number of shells and interactions to the neural network were fixed and modulators/ demodulators were elected from those models in which the half squared error was minimal. The tool used to generate data and the development of the various models was Toolbox Neural Network from Matlab, version 6.0.

Key words: modulators, demodulators, neural network, algorithm, half squared error.

Depósito Legal: PPX200002ZU2142 / ISSN:1856-4194. Volumen 3 Edición No 2 – Año 2005

INTRODUCCIÓN

Las señales audibles o de audiofrecuencia, no pueden irradiarse a través del espacio con costos accesibles, a su vez, si se quiere transmitir la información de varias personas en forma simultánea por distintas emisoras, las señales se mezclarían por ser todas de audio y resultaría imposible para un receptor separar una de otra. Estos inconvenientes se resuelven "modulando" a diferentes portadoras (cada una de distinta frecuencia) con cada información, de esta manera el receptor deberá reconocer a la portadora deseada y una vez logrado esto se extrae la información.

Esta investigación tiene como propósito crear modelos de moduladores/demoduladores a través de la simulación de modelos neuronales, evitando emplear complejos dispositivos para su manifestación física. A su vez tiene como propósito prestar apoyo para el estudio y enseñanza de la inteligencia artificial en el área de las comunicaciones, específicamente en el área de redes neuronales

PLANTEAMIENTO DEL PROBLEMA

Es evidente, que fueron numerosas y diversas las causas del impulso, creación y desarrollo de los diversos sistemas de comunicaciones, los cuales permitieran la transmisión de información a distancia sin demasiadas complicaciones.

En este sentido, podemos mencionar el caso particular de las señales audibles o de audiofrecuencia (baja frecuencia) comprendidas en el ancho de banda de 0 Hz a 25Khz, las cuales no pueden irradiarse a través del espacio con costos accesibles, debido a lo señalado por la teoría electromagnética la cual establece que la radiación eficiente de una señal se consigue cuando la antena radial tiene la longitud de cuando menos 1/10 de la longitud de onda de la señal a la cual se desea radiar.

Bajo las condiciones anteriormente mencionadas, la radiación directa de estas señales de audio en el espacio nos daría como inconveniente el uso de antenas de tamaño bastante considerable, por cuanto, las mismas no se encuentran en forma apropiada para la transmisión. (Herrera, 2001).

Con base en lo anterior si se logra trasladar la banda de audiofrecuencias a frecuencias lo suficientemente altas, la información es más fácil de propagar por el aire.

Depósito Legal: PPX200002ZU2142 / ISSN:1856-4194. Volumen 3 Edición No 2 – Año 2005

A través de la modulación, el espectro de determinado número de señales de información se puede trasladar a diferentes posiciones en el dominio de la frecuencia. Los espectros así trasladados se pueden entonces mezclar y transmitir por un canal único sin interferencia. Si bien la transmisión de información es simultánea y por un canal único, los espectros individuales sin traslaparse se pueden recuperar individualmente en el receptor. Esto se conoce como multiplexaje (MDF) y permite mayor aprovechamiento del ancho de banda del canal disponible y ahorros considerables en el canal de transmisión. (Herrera, 2001).

En otras palabras, la modulación no sólo resuelve el problema técnico de radiar la información, sino también, al usar radio frecuencias diferentes permite identificar a cada estación. Y por intermedio de esta frecuencia de radio, el circuito de sintonía del receptor logra captar la señal y procede a remodularla (proceso inverso al de modulación) para convertirla nuevamente en una señal de audio. (Herrera, 2001).

El trabajo que se presenta a continuación enfoca el desarrollo de modelos para simulación de moduladores / demoduladores, utilizando la teoría de redes neuronales. Dentro de ésta área, una idea obvia es la de simular directamente en una computadora el funcionamiento del cerebro utilizando neuronas artificiales y poder crear máquinas inteligentes. Y debido a la posibilidad de desarrollar un sistema neuronal y éste puede ser simulado a través de un programa computacional, para el desarrollo de los moduladores / demoduladores se utilizará el toolbox de redes neuronales correspondiente al software de Matlab versión 6.0.

METODOLOGÍA APLICADA

Para realizar el desarrollo de esta investigación se siguió una serie de fases comenzando por la simulación de los modelos matemáticos en el software de Matlab que permite obtener los datos que describen las entradas y salidas de los diferentes modelos.

Posteriormente, esta data se dividió en dos conjuntos: uno para la fase de entrenamiento y el segundo para la validación. En este momento se realiza a su vez el escalamiento de la data, si este es necesario. Se siguió con el desarrollo de los diversos modelos de de los moduladores / demoduladores utilizando las redes neuronales. Para ello se seleccionaron las entradas y salidas con las cuales se desarrolla la red, se define el tipo de red y se selecciona el algoritmo con el que se entrena.

Depósito Legal: PPX200002ZU2142 / ISSN:1856-4194. Volumen 3 Edición No 2 – Año 2005

Luego se establece el número de capas, cantidad de neuronas en la capa oculta, así como las diferentes funciones de activación de estas capas, que permitiendo generar los diversos pesos y bias iniciales.

La última fase consiste en realizar una validación del modelo a través de la comparación de los datos obtenidos con el entrenamiento de las redes neuronales y los datos seleccionados para validar el modelo.

Para el análisis de los datos en esta investigación, se establecieron criterios de comparación entre los diversos algoritmos de entrenamiento, tales como tipo de funciones de activación, número de iteraciones, pesos y bias aleatorios, y cálculo del mínimo error cuadrado, en cada caso.

RESULTADOS DE LA INVESTIGACIÓN

DESCRIPCIÓN DE LOS SISTEMAS MODULADORES

A continuación se presentan los modelos matemáticos que describen el comportamiento de los diversos moduladores/demoduladores, los cuales serán simulados en matlab para generar los datos que permitirán desarrollar estos modelos utilizando redes neuronales, en búsqueda de la aproximación al comportamiento real de los mismos.

1.) Modulación de amplitud modulada con supresión de portadora (AM-PS). El modelo que describe un modulador AM-PS está dado por:

$$\phi_{AM}(t) = f(t) A_C \cos w_c(t)$$

Donde A_C representa la amplitud de la señal portadora y *f(t)* es la señal mensaje, la cual se desea transmitir, y puede estar representada por una señal senoidal, tal como

$$A_m \cos w_m(t)$$

Los valores de los parámetros seleccionados para el sistema simulado en esta investigación son los siguientes:

$$A_C = 1 \text{ y } f_c = 890\text{Khz}$$

Representan la amplitud y frecuencia de la señal portadora.

Depósito Legal: PPX200002ZU2142 / ISSN:1856-4194. Volumen 3 Edición No 2 – Año 2005

A_m = 1 y f_m = 5hz

Representan la amplitud y frecuencia de la señal mensaje. Por lo tanto el sistema queda representado por la ecuación:

$$\phi_{AM}(t) = A_m Sen w_m(t) A_C \cos w_c(t) =$$

$$\phi_{AM}(t) = Sen(10\pi(t)).\cos(5592\times10^3\pi(t))$$

2.) Modulación de amplitud con portadora completa o simplemente modulación de amplitud (AM). El modelo que describe un modulador AM está dado por:

$$\phi_{AM}(t) = (A_C + f(t))\cos w_c(t)$$

Donde

$$A_C \cos w_c(t)$$

Representa la señal portadora y *f(t)* es la señal mensaje, la cual se desea transmitir y puede estar representada por una señal senoidal, tal como

$$A_m \cos w_m(t)$$

Los valores de los parámetros seleccionados para el sistema simulado en esta investigación son los siguientes:

A_C = 1 y f_c = 20Khz

Representan la amplitud y frecuencia de la señal portadora.

A_m = 1 y f_m = 1Khz

Representan la amplitud y frecuencia de la señal mensaje. Por lo tanto el sistema queda representado por la ecuación:

$$\phi_{AM}(t) = (1 + \cos(2\times10^3\pi(t)).\cos(40\pi\times10^3(t))$$

Desde el punto de vista físico la no linealidad de este sistema consiste en que la presencia de la onda portadora en la onda modulada ocasiona que se viole el principio de superposición que debe cumplir todo sistema lineal.

Depósito Legal: PPX200002ZU2142 / ISSN:1856-4194. Volumen 3 Edición No 2 – Año 2005

Los dos casos de modulación que se presentan a continuación: la modulación en fase y la modulación en frecuencia son casos especiales de la modulación angular:

3.) Modulación de frecuencia*:* El modelo que describe un modulador FM está dado por:

$$m_{FM}(t) = A\cos\left[w_c t + k_f \int f(t)dt\right]$$

Donde k_f constituye la desviación de la frecuencia de la portadora a partir de su valor inicial fijo w_c

Los valores de los parámetros seleccionados para el sistema simulado en esta investigación son los siguientes:

A = 1, $f(t) = A_m \cos w_m(t)$

En el cual,

A_m = 1 y f_m = 1Khz

Representan la amplitud y frecuencia de la señal moduladora.

A = 1 y f_c = 20Khz

Representan la amplitud y frecuencia de la señal portadora. El valor del parámetro k_f dependerá de de la frecuencia de la portadora y de la de muestreo.

4.) Modulación en fase: El modelo que describe un modulador PM está dado por:

$$m_{PM}(t) = A\cos\left[w_c t + k_P f(t)\right]$$

Donde kp constituye la sensibilidad de fase del modulador de fase.

Los valores de los parámetros seleccionados para el sistema simulado en esta investigación son los siguientes:

Depósito Legal: PPX200002ZU2142 / ISSN:1856-4194. Volumen 3 Edición No 2 – Año 2005

A = 1, $f(t) = A_m \cos w_m(t)$

En el cual

$A_m = 1$

y $f_m = 1\text{Khz}$

Representan la amplitud y frecuencia de la señal moduladora.

A = 1

$f_c = 20\text{Khz}$

Representan la amplitud y frecuencia de la señal portadora. El valor del parámetro k_f dependerá de de la frecuencia de la portadora y de la de muestreo.

Como una de las propiedades más resaltantes de la modulación angular es que ambos tipos son funciones no lineales.

Luego de obtener los diversos moduladores será necesario desarrollar los demoduladores correspondientes a cada tipo particular de modulación. El proceso de construcción de cada modulador se realizará luego de obtener la modulación por la simulación del modelo neuronal, para obtener la señal demodulada o mensaje original.

Es importante considerar que en el proceso de desmodulación se utilizan como entrada los valores (vectores) correspondientes a la señal simulada, obtenida en el proceso de modulación, conjuntamente con los datos correspondientes de la señal portadora. De manera tal que se pueda recuperar la señal de mensaje original.

RECOPILACIÓN DE LA DATA DE ENTRADA – SALIDA DE LOS SISTEMAS

A continuación se explica el procedimiento para obtener la data de entrada – salida de un modulador AM –PS:

La data a utilizar para el desarrollo del modulador AM-PS será tomada del archivo DATOAMPS, el cual contiene los valores generados por el programa

Depósito Legal: PPX200002ZU2142 / ISSN:1856-4194. Volumen 3 Edición No 2 – Año 2005

GDATOS_AM_PS, los cuales representan las variables de entrada: p (señal portadora), de tamaño 1000´1, m (señal mensaje), de tamaño 1000´1, y la salida Y (señal modulada), de tamaño 1000´1.

Para el resto de los casos se sigue un procedimiento similar, variando los nombres de los archivos, programas y tamaño de los vectores que forman las variables.

CONSTRUCCIÓN DE LOS CONJUNTOS DE ENTRENAMIENTO Y VALIDACIÓN DE LOS SISTEMAS MODULADORES

Para los diferentes moduladores, tenemos como variables de entrada m (mensaje) y p (portadora), y como variable de salida y (modulante).

En el caso de los demoduladores tenemos como variables de entrada y (modulante) y p (portadora) y como variable de salida m (mensaje). En cuanto al tamaño de cada uno de los conjuntos, no existe una forma establecida. Se recomienda que el 50% de la data total disponible, se utilice para el conjunto de entrenamiento (subíndices impares), el 50% restante para el conjunto de validación (subíndices pares).

SELECCIÓN DE LA ARQUITECTURA DE LA RED DE LOS SISTEMAS MODULADORES / DEMODULADORES

El tipo de red seleccionado para desarrollar los modelos de los moduladores/ demoduladores fue el Backpropagation, puesto que el mismo puede aproximar cualquier función si se escoge una adecuada configuración para la red y un número preciso de neuronas en la capa oculta, aún cuando no existe un procedimiento único para determinar la configuración exacta de la red, todo dependerá de la experiencia del desarrollador del modelo de red neuronal.

Luego de realizar varias pruebas, se observó que con 15 o menos neuronas en la capa oculta y no más de 125 iteraciones, el aprendizaje de la red, en cada caso era óptimo.

Es importante destacar que en algunos casos cuando se aumentaba el número de iteraciones disminuía el rendimiento de la red, es decir, el aprendizaje era mínimo, se mantenía o disminuía.

Por lo tanto se decidió realizar el entrenamiento con 15, 12 ó 10 neuronas en la capa oculta, y 100 ó 125 iteraciones, de acuerdo a cado caso, para el aprendizaje de la red.

Depósito Legal: PPX200002ZU2142 / ISSN:1856-4194. Volumen 3 Edición No 2 – Año 2005

Las capas de entrada y de salida se estructuraron de la siguiente manera:

PARA LOS MODULADORES:

En la capa de entrada: dos unidades, que corresponden a la señal moduladora, y la portadora. Capa de salida: una unidad, correspondiente a la señal modulada; excepto para la modulación PM, la cual contiene cuatro unidades, correspondiente a los sistemas dinámicos, que dependen de valores anteriores o retardos de la señal modulante.

PARA LOS DEMODULADORES:

Capa de entrada: dos unidades, que corresponden a la señal modulante, y la portadora. Capa de salida: una unidad, correspondiente a la señal demodulada; excepto para de la modulación PM, la cual contiene cuatro unidades, correspondiente a los sistemas dinámicos, que dependen de valores anteriores, de la señal mensaje.

ENTRENAMIENTO DE LA RED DE LOS SISTEMAS MODULADORES/ DEMODULADORES

Para el entrenamiento de las diferentes redes se utilizó el algoritmo de Levenberg-Marquard. Se fijo un criterio de comparación mediante la observación del mínimo error cuadrático obtenido por el número de iteraciones de 100 ó 125 de acuerdo al caso, y realizar 5 entrenamientos por cada combinación de las funciones de activación, seleccionando el entrenamiento óptimo a través del mínimo error cuadrático medio.

La red es creada mediante el comando newff para crear redes Backpropagation, como se muestra, para el caso de un modulador AM-PS:

```
net = newff(minmax(P1),[15,1],{'tansig','logsig'},'trainlm');
```

Los valores de iniciación de la matriz de pesos se generaron aleatoriamente y los valores de entrada a la red se agruparon en la matriz P1 correspondiente a las entradas me y pe.

Con los valores generados anteriormente se inicia el entrenamiento con el siguiente comando: net = train(net,P1,Ye).

Depósito Legal: PPX200002ZU2142 / ISSN:1856-4194. Volumen 3 Edición No 2 – Año 2005

SIMULACIÓN DE LOS MODULADORES / DEMODULADORES

A continuación se presentan los mejores resultados de cada tipo de modulador/demodulador, luego de realizar los cambios en las diferentes funciones de transferencia, entre la capa oculta y la capa de salida, es decir:

Tansig - purelin , Tansig – tansig, Tansig – logsig, Logsig – logsig, Logsig – purelin, y Logsisg – tansig.

SIMULACIÓN DEL MODULADOR AM-PS

- **Funciones de transferencia logsig y purelin. Algoritmo trainlm.**

El mejor de cinco entrenamientos, con los bias y pesos asociados al error mínimo resulto el entrenamiento n = 4.

A continuación se muestra la matriz formada por los errores de cinco entrenamientos: E = 1.0e-003 * [0.0781; 0.6108; 0.5565; 0.0051; 0.0061]; Y la selección del mínimo error cuadrático medio:

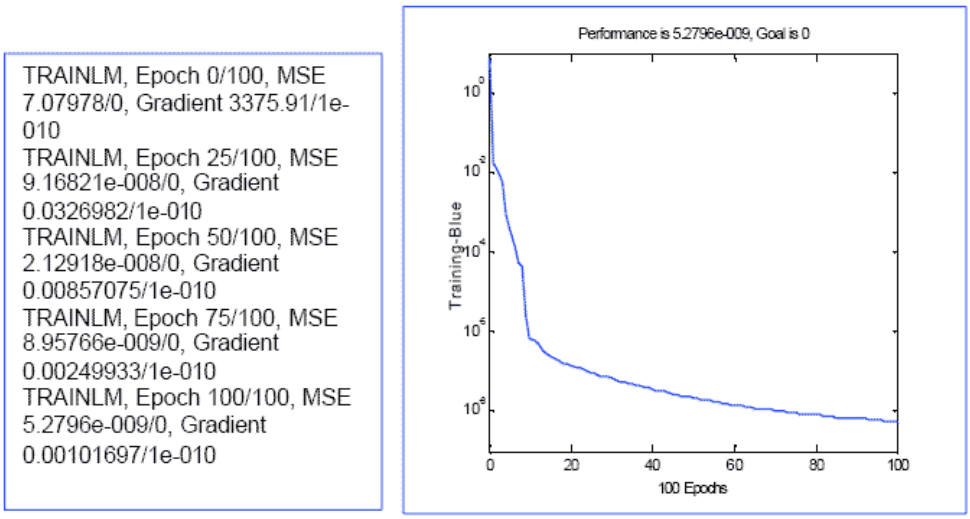


Figura 1. (a) Iteraciones. (b) Iteraciones Vs error en un modulador AM-PS.
Fuente: El autor

A continuación se muestran las gráficas de datos simulados por el entrenamiento asociado al error mínimo y datos de validación:

SIMULACIÓN DEL DEMODULADOR AM-PS

- **Funciones de transferencia tansig y purelin. Algoritmo trainlm.**

Depósito Legal: PPX200002ZU2142 / ISSN:1856-4194. Volumen 3 Edición No 2 – Año 2005

El mejor resultado de cinco entrenamientos, con los bias y pesos asociados al error mínimo resultó el entrenamiento n = 4.

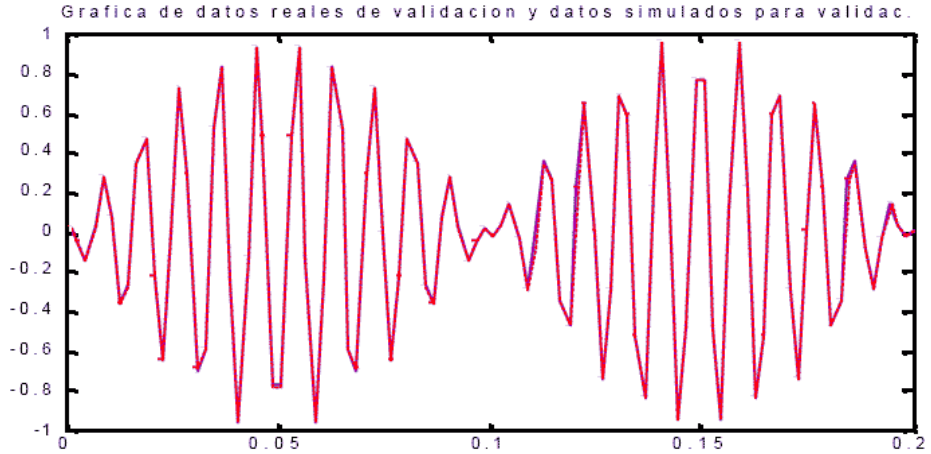


Figura 2. Simulación de datos reales (línea continua) Vs. datos de validación (línea punteada). Intervalo: [0, 0.2].
Fuente: el autor.

```
TRAINLM, Epoch 0/100, MSE 7.37779/0, Gradient 3459.4/1e-010
TRAINLM, Epoch 25/100, MSE 6.34484e-009/0, Gradient 0.0585264/1e-010
TRAINLM, Epoch 50/100, MSE 2.42178e-009/0, Gradient 0.0105783/1e-010
TRAINLM, Epoch 75/100, MSE 1.63673e-009/0, Gradient 0.00412596/1e-010
TRAINLM, Epoch 100/100, MSE 1.2608e-009/0, Gradient 0.00214533/1e-010
```

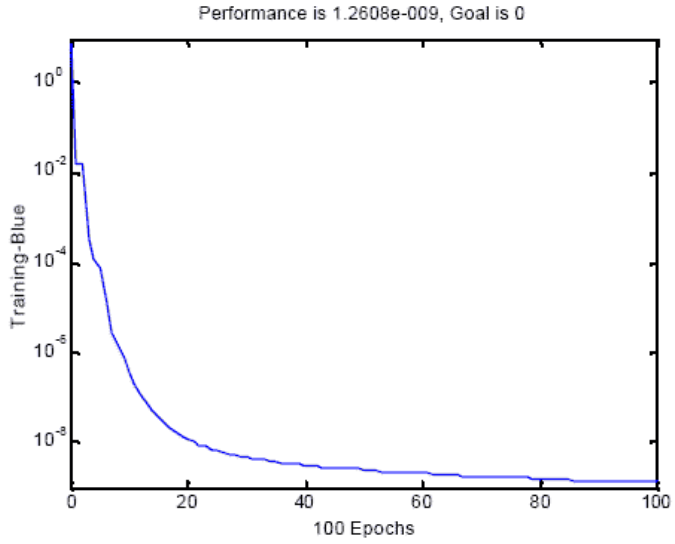


Figura 3. (a) Iteraciones. (b) Iteraciones Vs. error en un demodulador AM-PS. Fuente: el autor.

Depósito Legal: PPX200002ZU2142 / ISSN:1856-4194. Volumen 3 Edición No 2 – Año 2005

A continuación se muestran las gráficas de datos simulados por el entrenamiento asociado al error mínimo y datos de validación:

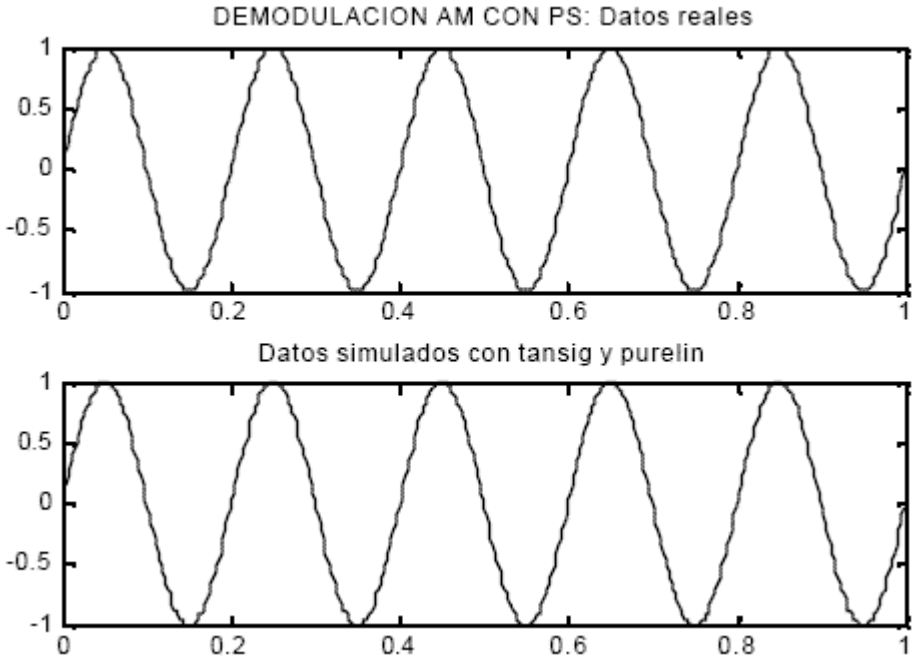


Figura 4. Simulación de datos reales (figura superior) Vs. datos de validación (figura inferior). Intervalo: [0,1]. (b).
Fuente: El autor

SIMULACIÓN DEL MODULADOR AM.

- **Funciones de transferencia tansig y logsig. Algoritmo trainlm.**

El mejor resultado de cinco entrenamientos, con los bias y pesos asociados al error mínimo resulto el entrenamiento n = 5

Depósito Legal: PPX200002ZU2142 / ISSN:1856-4194. Volumen 3 Edición No 2 – Año 2005

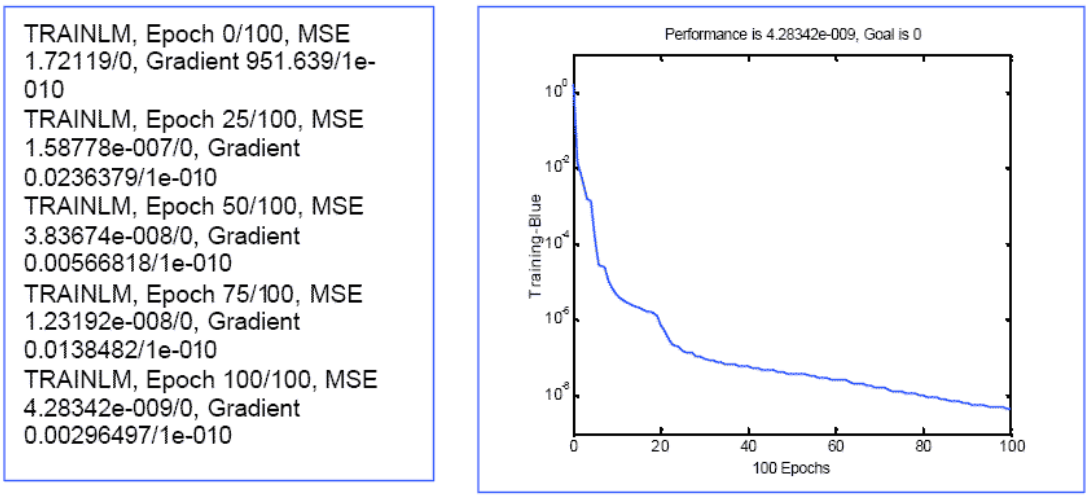


Figura 5. (a) Iteraciones. (b) Iteraciones Vs. error en un demodulador AMPS.
Fuente: El autor

A continuación se muestra la matriz que contiene los errores correspondientes a los cinco entrenamientos: E = [0.0001; 0.0016; 0.0001 0.0017; 0.0073]; minimo_e = 1.0285e-004; resultó en la iteración n = 1.

A continuación se muestran las gráficas de datos simulados por el entrenamiento asociado al error mínimo y datos de validación:

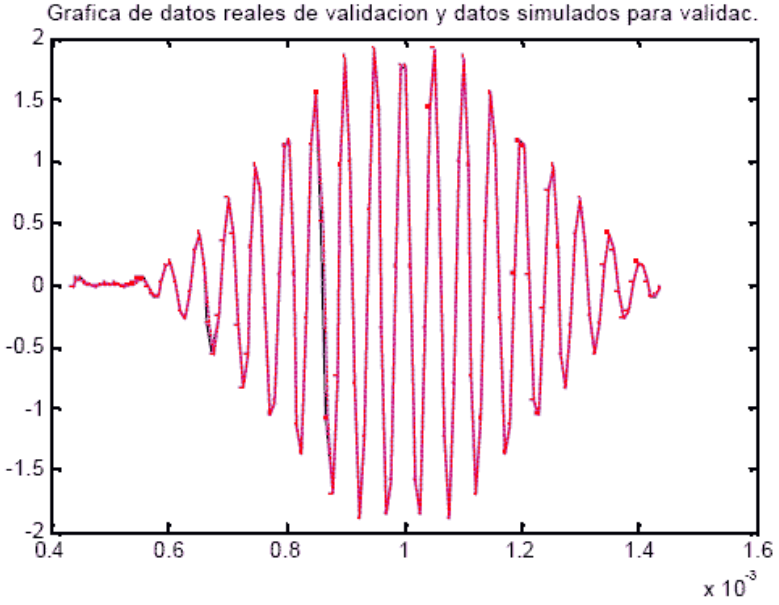


Figura 6. (a). Simulación de datos reales (línea continua) Vs. datos de validación (línea punteada). Intervalo: [0, 1.6×10-3].
Fuente: El autor

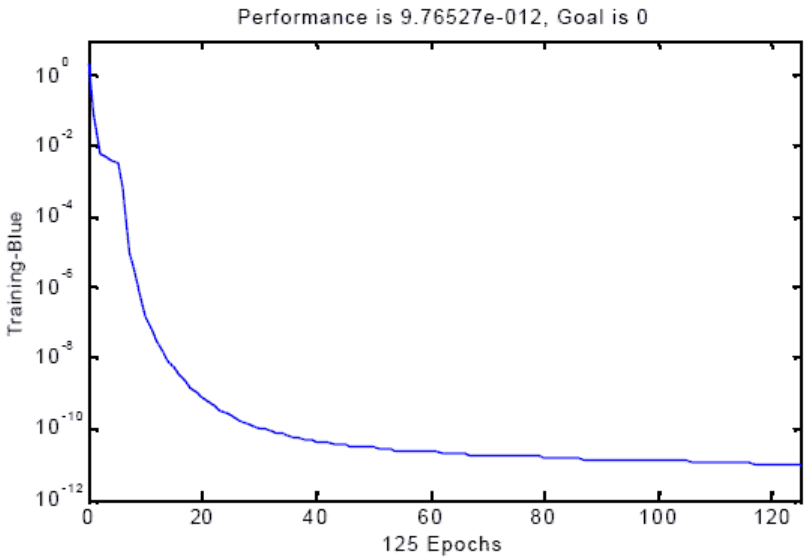
Depósito Legal: PPX200002ZU2142 / ISSN:1856-4194. Volumen 3 Edición No 2 – Año 2005

SIMULACIÓN DEL DEMODULADOR AM.

El mejor resultado de cinco entrenamientos, con los bias y pesos asociados al error mínimo resulto el entrenamiento n = 1.

```
TRAINLM, Epoch 0/100, MSE 7.37779/0, Gradient 3459.4/1e-010
TRAINLM, Epoch 25/100, MSE 6.34484e-009/0, Gradient 0.0585264/1e-010
TRAINLM, Epoch 50/100, MSE 2.42178e-009/0, Gradient 0.0105783/1e-010
TRAINLM, Epoch 75/100, MSE 1.63673e-009/0, Gradient 0.00412596/1e-010
TRAINLM, Epoch 100/100, MSE 1.2608e-009/0, Gradient 0.00214533/1e-010
```

(a)



(b)

Figura 7. (a) Iteraciones. (b) Iteraciones Vs. error en un demodulador AM
Fuente: El autor

A continuación se muestra la matriz que contiene los errores correspondientes a los cinco entrenamientos:

E = 1.0e-004 * [0.0950 0.0005; 0.1521; 0.2184; 0.3231];

Y la selección del mínimo error cuadrático medio: minimo_ error = 4.7656e-008, n = 2.

A continuación se muestra la gráfica de datos simulados por el entrenamiento asociado al error mínimo y datos de validación:

Depósito Legal: PPX200002ZU2142 / ISSN:1856-4194. Volumen 3 Edición No 2 – Año 2005

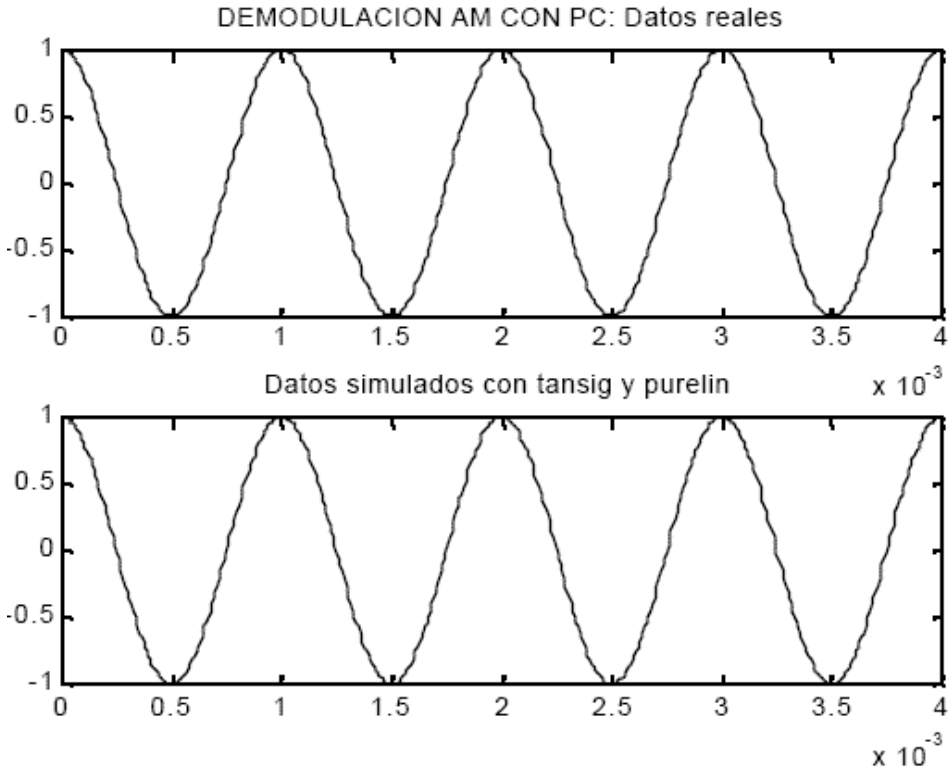


Figura 8. Simulación de datos reales (figura superior) vs. datos de validación (figura inferior). Intervalo: [0, 4×10e-3].
Fuente: el autor.

SIMULACIÓN DEL MODULADOR FM

- **Funciones de transferencia tansig y purelin**

El mejor resultado de cinco entrenamientos, con los bias y pesos asociados al error mínimo, resulto el entrenamiento n = 1

```
TRAINLM, Epoch 0/100, MSE 5.79598/0, Gradient 22983.8/1e-010
TRAINLM, Epoch 25/100, MSE 7.64547e-005/0, Gradient 13.0351/1e-010
TRAINLM, Epoch 50/100, MSE 2.17566e-005/0, Gradient 1.49066/1e-010
TRAINLM, Epoch 75/100, MSE 1.90831e-005/0, Gradient 0.779421/1e-010
TRAINLM, Epoch 100/100, MSE 7.56534e-006/0, Gradient 0.170321/1e-010
```

(a)

Depósito Legal: PPX200002ZU2142 / ISSN:1856-4194. Volumen 3 Edición No 2 – Año 2005

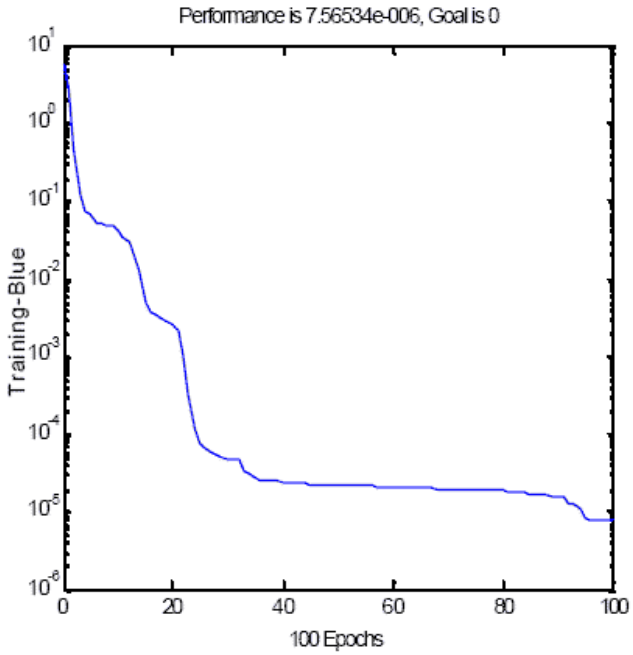


Figura 9. (a) Iteraciones. (b) Iteraciones Vs. error en un modulador FM.
Fuente: el autor.

A continuación se muestra la matriz que contiene los errores correspondientes a los cinco entrenamientos: E = [0.0000; 0.0123; 0.0000 0.0000; 0.0006]; minimo _ error = 3.0267e-005, resultó en la iteración n =1.

A continuación se muestran las gráficas de datos simulados por el entrenamiento asociado al error mínimo y datos de validación:

Depósito Legal: PPX200002ZU2142 / ISSN:1856-4194. Volumen 3 Edición No 2 – Año 2005

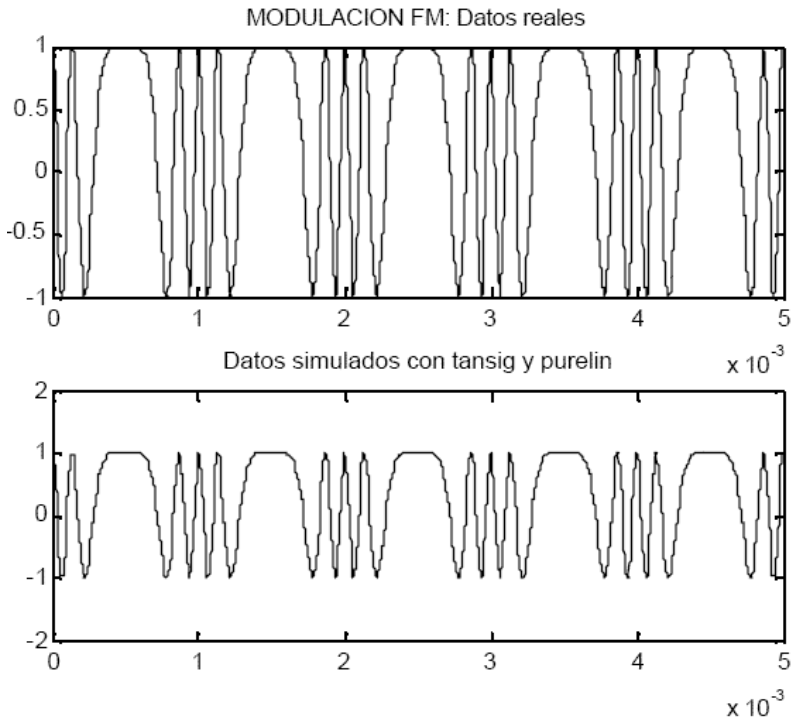


Figura 10. Simulación de datos reales Vs. datos de validación.
Intervalo: [0, 4×10e-3].
Fuente: el autor.

SIMULACIÓN DEL DEMODULADOR FM

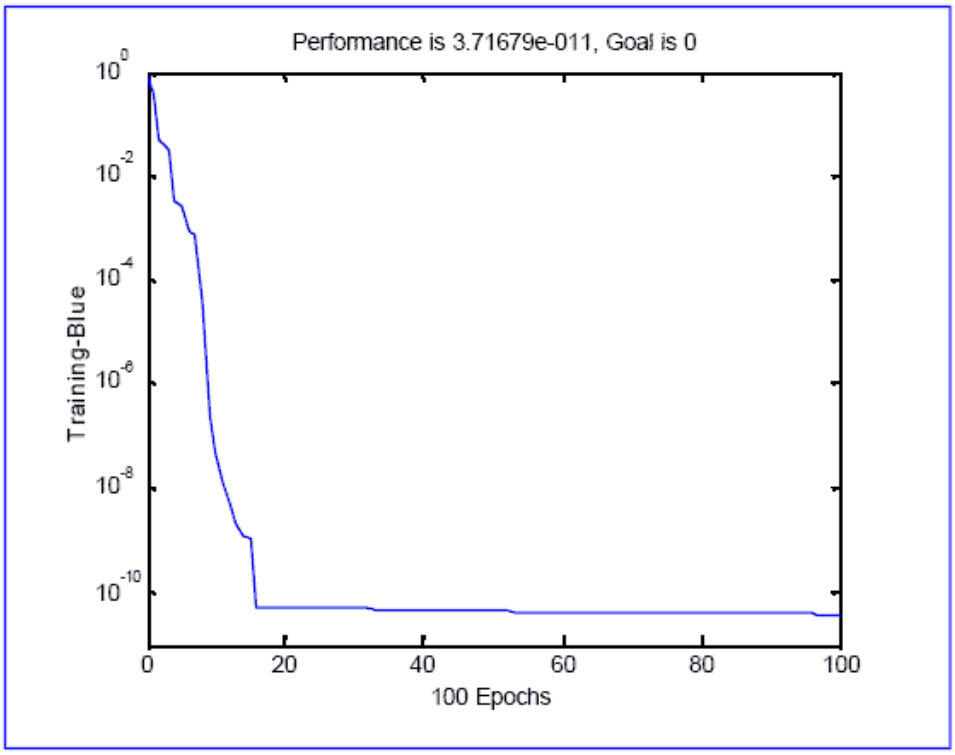
- **Funciones de transferencia tansig y purelin. Algoritmo trainlm.**

El mejor resultado de cinco entrenamientos, con los bias y pesos asociados al error mínimo resulto el entrenamiento n = 5.

```
TRAINLM, Epoch 0/100, MSE 1.00599/0, Gradient 4534.58/1e-010
TRAINLM, Epoch 25/100, MSE 4.6889e-007/0, Gradient 0.713187/1e-010
TRAINLM, Epoch 50/100, MSE 1.77674e-007/0, Gradient 0.0593627/1e-010
TRAINLM, Epoch 75/100, MSE 1.22661e-007/0, Gradient 0.0194549/1e-010
TRAINLM, Epoch 100/100, MSE 9.88097e-008/0, Gradient 0.00955367/1e-010
```

(a)

Depósito Legal: PPX200002ZU2142 / ISSN:1856-4194. Volumen 3 Edición No 2 – Año 2005



(b)

Figura 11. (a) Iteraciones. (b) Iteraciones Vs. error en un demodulador

A continuación se muestra la matriz que contiene los errores correspondientes a los cinco entrenamientos:

E = 1.0e-006 * [0.1004; 0.0101; 0.0023; 0.1445; 0.0001]
minimo _error = 1.4867e-010, resultó en la iteración n = 5.

A continuación se muestran las gráficas de datos simulados por el entrenamiento asociado al error mínimo y datos de validación:

Depósito Legal: PPX200002ZU2142 / ISSN:1856-4194. Volumen 3 Edición No 2 – Año 2005

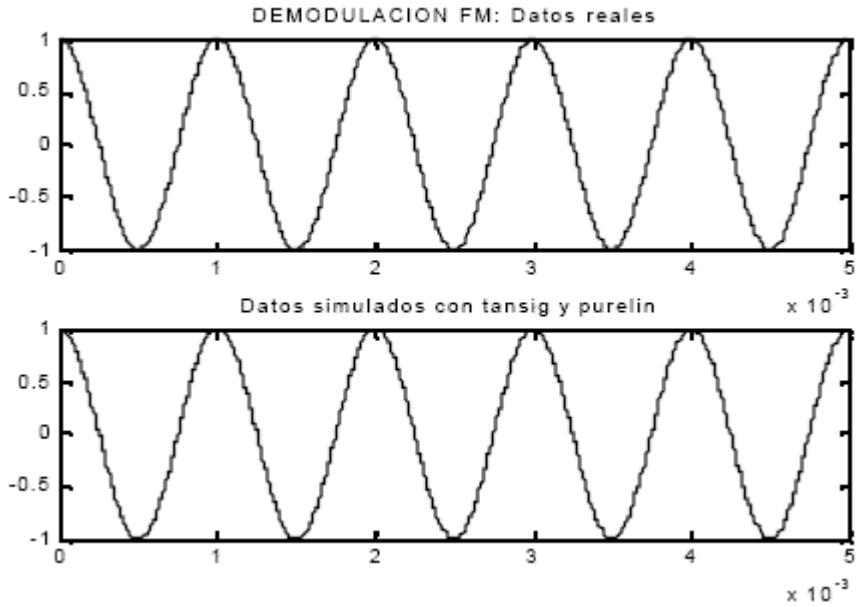


Figura 12. Simulación de datos reales Vs. datos de validación. Intervalo: [0, 5×10e-3]. Fuente: el autor.

SIMULACIÓN DEL MODULADOR PM.

- **Funciones de transferencia tansig y purelin.**

El mejor resultado de cinco entrenamientos, con los bias y pesos asociados al error mínimo resulto el entrenamiento n = 1.

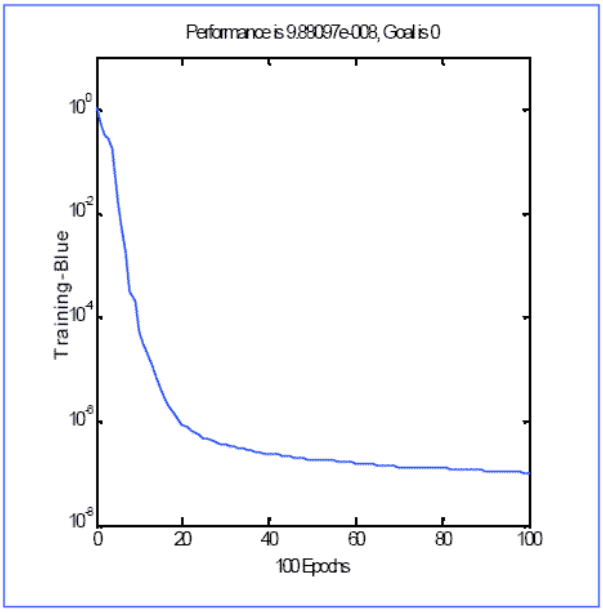
A continuación se muestra la matriz que contiene los errores correspondientes a los cinco entrenamientos: E = 1.0e-006 * [0.3952; 0.8513; 0.9556; 0.8770; 0.5237]; Y la selección del mínimo error cuadrático medio: mínimo_ error = 3.9524e-007, n = 1.

A continuación se muestran las gráficas de datos simulados por el entrenamiento asociado al error mínimo y datos de validación:

Depósito Legal: PPX200002ZU2142 / ISSN:1856-4194. Volumen 3 Edición No 2 – Año 2005

```
TRAINLM, Epoch 0/100, MSE 1.00599/0, Gradient 4534.58/1e-010
TRAINLM, Epoch 25/100, MSE 4.6889e-007/0, Gradient 0.713187/1e-010
TRAINLM, Epoch 50/100, MSE 1.77674e-007/0, Gradient 0.0593627/1e-010
TRAINLM, Epoch 75/100, MSE 1.22661e-007/0, Gradient 0.0194549/1e-010
TRAINLM, Epoch 100/100, MSE 9.88097e-008/0, Gradient 0.00955367/1e-010
```

(a)



(b)

Figura 13. (a) Iteraciones. (b) Iteraciones Vs. error en un modulador PM.
Fuente: El autor.

A continuación se muestra la matriz que contiene los errores correspondientes a los cinco entrenamientos:

E = 1.0e-005 *[0.0115; 0.0023; 0.1054; 0.0055; 0.0616];
minimo_ error = 2.3303e-008, n = 2.

A continuación se muestran las gráficas de datos simulados por el entrenamiento asociado al error mínimo y datos de validación:

Depósito Legal: PPX200002ZU2142 / ISSN:1856-4194. Volumen 3 Edición No 2 – Año 2005

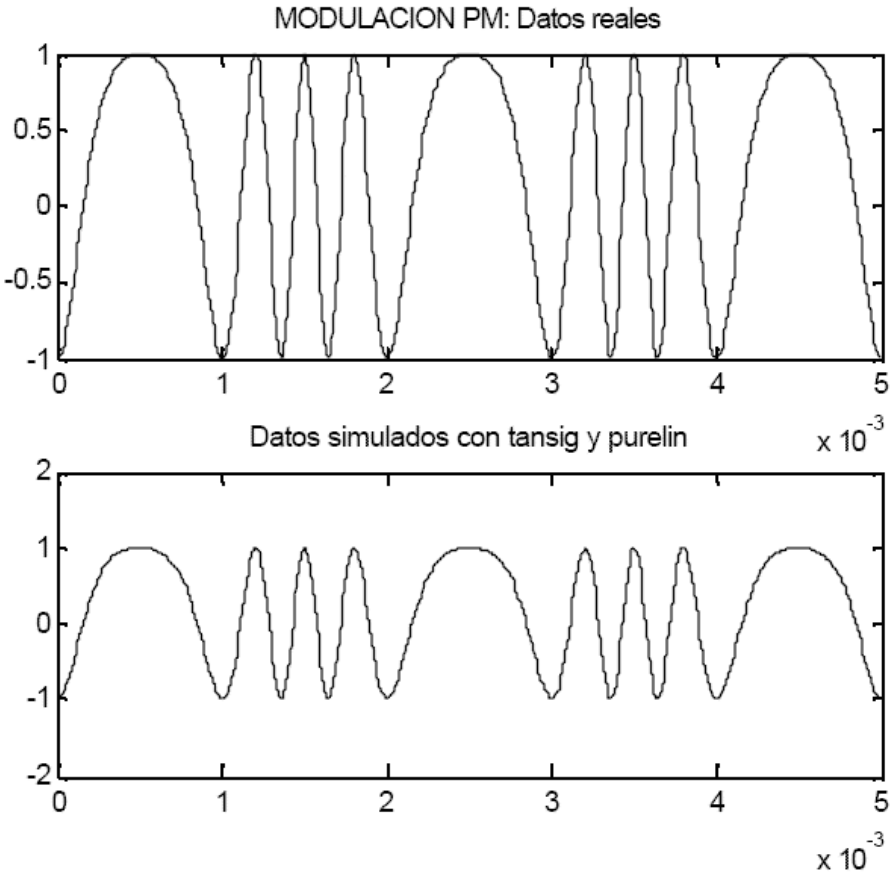


Figura 14. Simulación de datos reales Vs. datos de validación. Intervalo: [0, 5×10e-3]. Fuente: el autor.

Simulación del demodulador PM

- **Funciones de transferencia tansig y purelin**

El mejor resultado de cinco entrenamientos, con los bias y pesos asociados al error mínimo resulto el entrenamiento n = 2.

```
TRAINLM, Epoch 0/100, MSE 5.36437/0, Gradient 12147.3/1e-010
TRAINLM, Epoch 25/100, MSE 1.936e-008/0, Gradient 0.252399/1e-
010
TRAINLM, Epoch 50/100, MSE 1.01506e-008/0, Gradient 0.049898/1e-
010
TRAINLM, Epoch 75/100, MSE 7.30493e-009/0, Gradient
0.0209196/1e-010
TRAINLM, Epoch 100/100, MSE 5.82413e-009/0, Gradient
0.0114921/1e-010
```

Depósito Legal: PPX200002ZU2142 / ISSN:1856-4194. Volumen 3 Edición No 2 – Año 2005

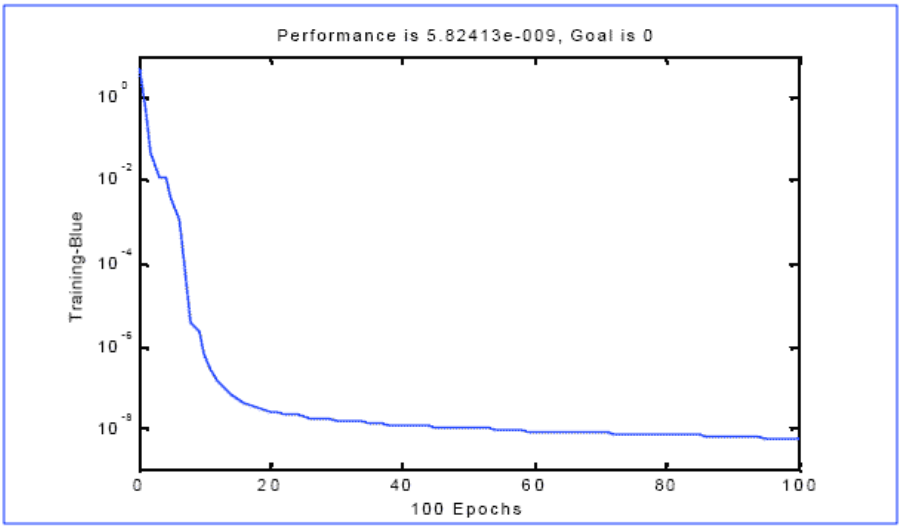


Figura 15. (a) Iteraciones. (b) Iteraciones Vs. error en un modulador PM.
Fuente: El autor.

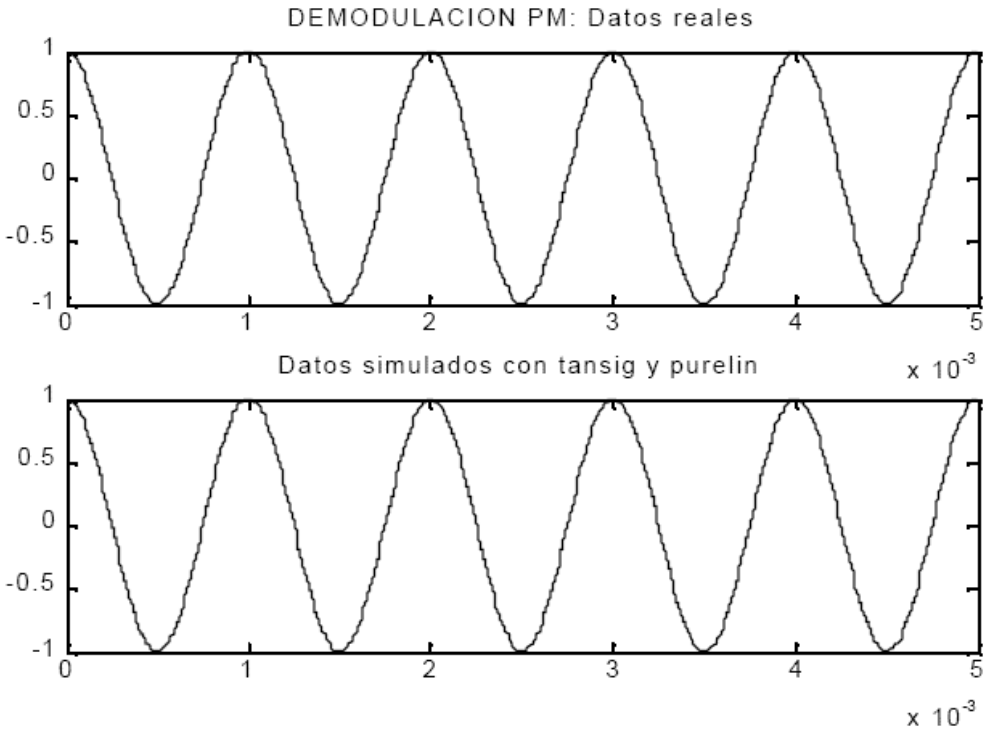


Figura 16. Simulación de datos reales Vs. datos de validación.
Intervalo: [0, 5×10e-3].
Fuente: el autor.

Depósito Legal: PPX200002ZU2142 / ISSN:1856-4194. Volumen 3 Edición No 2 – Año 2005

CONCLUSIONES

Dentro del área de inteligencia artificial se ha comprobado la facilidad de las redes neuronales de resolver una gran cantidad de problemas de ingeniería, tanto para sistemas en los cuales se les conoce su representación matemática, como aquellos en los cuales se les conoce sólo ciertos datos de su funcionamiento.

Lo anterior, se debe a la capacidad de las redes neuronales de emular inteligencia artificial a través de la construcción de sencillos modelos biológicos del cerebro, los cuales han resultado altamente seguros y flexibles.

En el desarrollo de los moduladores / demoduladores se llegaron a las siguientes conclusiones:

- Para la simulación de los moduladores se tomó como entradas, la señal moduladora y la portadora, y como salida la señal modulante. En el caso de los demoduladores se escogió como entradas la señal modulante y adicionalmente la señal portadora, por cuanto esta última es conocida tanto para la transmisión como en la recepción, permitiendo de esta manera obtener la señal mensaje.

- Se generaron los datos de entrada salida ejecutando el programa GDATO en el software de Matlab, los cuales fueron diseñados basados en el modelo matemático correspondiente, de acuerdo al tipo de modulación / demodulación empleado.

- Luego de identificar el número de entradas / salidas del modelo a simular, el número de entradas del sistema proporciona el número de neuronas de la capa de entrada y el número de salidas proporciona el número de neuronas de la capa de salida, quedando solo como ajuste por ensayo y error, el número de neuronas en la capa oculta, a fin de lograr una arquitectura adecuada, la cual permita obtener un rendimiento óptimo de la red. Por otra parte, si la salida del sistema es conocida (como en el caso de los moduladores / demoduladores) el problema se restringe a escoger una red con aprendizaje supervisado; es por ello que el tipo de red escogido para esta investigación fue el backpropagation.

- Después de realizar pruebas con diversos algoritmos, se comprobó que el algoritmo correspondiente al método de Levenberg Marquardt garantizaba una alta velocidad de aprendizaje y una excelente generalización de los patrones de entrenamiento que se habían presentado inicialmente, y

Depósito Legal: PPX200002ZU2142 / ISSN:1856-4194. Volumen 3 Edición No 2 – Año 2005

es por ello que se escogió para realizar el entrenamiento de los moduladores / demoduladores.

- Luego de seleccionar el número de capas y el algoritmo a utilizar, se procedió a realizar varios entrenamientos variando el número de iteraciones, resultando como entrenamiento óptimo, 100 iteraciones para todos los tipos de modulación y demodulación, con excepción de la demodulación AM, la cual fue trabajada con 125 iteraciones para alcanzar un rendimiento óptimo.

En cuanto al criterio establecido para realizar pruebas con diferentes funciones de transferencia a la red neuronal tipo backpropagation, se pueden establecer las siguientes conclusiones:

- El modulador AM – PS, el modulador AM y el demodulador PM tuvieron un excelente rendimiento con la función de activación: logsin – purelin, tomándose éstas redes como la más próxima al comportamiento real de dichos sistemas.

- El demodulador AM –PS, el demodulador AM, el modulador FM, el demodulador FM, y demodulador PM tuvieron un excelente rendimiento con la función de activación: tangsin – purelin, tomándose éstas redes como la más próxima al comportamiento real de dichos sistemas.

- Para el desarrollo de todos los demoduladores y del modulador PM específicamente, se aplicó la fórmula de identificación de un sistema dinámico: ((1 (2 ( ) (1) ( ) (1))),), , , , - - - - = n n n n n n n Z f z z x x y y , lo que significa que para identificar correctamente los modelos era necesario valores anteriores de la señal.

- Para probar la validez de los distintos modelos, se utilizó el criterio del error cuadrático medio fijado en 1?10-4 , el fue calculado considerando los datos reales y los generados por la red, para las diferentes combinaciones de las funciones de activación, seleccionando como modelo óptimo el correspondiente al error cuadrático medio mínimo de los diversos entrenamientos.

Como última conclusión puede afirmarse que el diseño de los moduladores / demoduladores permitió corroborar la versatilidad de las redes backpropagation y del algoritmo seleccionado, puesto que se obtuvo una excelente aproximación a los modelos que se deseaban simular.

Depósito Legal: PPX200002ZU2142 / ISSN:1856-4194. Volumen 3 Edición No 2 – Año 2005

BIBLIOGRAFÍA

Aguilar, H. (2002). **Fundamentos de los sistemas modernos de comunicaciones**. Editorial Alfaomega.

Burrus, S. (1998). **Tratamiento de la señal**. Editorial Prentice Hall.

Carlson, Bruce (1980). **Sistemas de Comunicación**. Editorial McGraw Hill.

Couch, L. (1997). **Sistemas de comunicación digitales y analógicos**. Editorial Pearson Educación.

Etter, D (1997). **Solución de problemas de ingeniería con Matlab**. Editorial Prentice Hall.

Fausett, L. (1994). **Fundamentals of Neural Networks: Architectures, Algorithms and Applicatio**ns. Prentice Hall.

Haykin, S. et al (2001). **Señales y sistemas**. Editorial Limusa.

Herrera, E. (2001) **Comunicaciones I**. Editorial Limusa.

IEEE. (1990). Transactions on Neural Networks, volumen I, March, pp.4-27. Neural Network Toolbox**. For use with Matlab**. Versión 3.0.

Shannon, R. (1988). **Simulación de sistemas: Diseño, Desarrollo e implantación**. Editorial Trillas.

Soliman et al. (1999). **Señales y sistemas**. Editorial Prentice Hall.

Stremler, F.G. (1993). **Introducción a los sistemas de comunicación**. Editorial Pearson Educación.

Sulbarán C. (2000). **Implementación de estrategias de control basadas en modelos no lineales, utilizando redes neurales**. Trabajo de Ascenso. Universidad del Zulia.

The Math Works, Inc. (1998). **Matlab edición de estudiante**. Editorial Prentice Hall.

Tomasi, W (1996). **Sistemas de comunicaciones electrónicas**. Editorial Pearson Educación.

Depósito Legal: PPX200002ZU2142 / ISSN:1856-4194. Volumen 3 Edición No 2 – Año 2005

http://www.iiia.csic.es/~mario/Default.htm

http://www.electronica.com.mx/neural/

http://ohm.utp.edu.co/neuronales

Depósito Legal: PPX200002ZU2142 / ISSN:1856-4194. Volumen 3 Edición No 2 – Año 2005