

Comparación de Dos Programas Basados en Redes Neuronales para Identificación de Procesos, Aplicados a una Columna de Destilación y al Actuador Hidráulico de una Grúa

Enrique Quintero Mármol Márquez*, Luis Ariel Doñate Alcázar*, Luis Gerardo Vela Valdés*, Omar E. Quintero Mármol Sánchez**

*Centro Nacional de Investigación y Desarrollo Tecnológico (CENIDET), Cuernavaca, Mor.,
eqm@cenidet.edu.mx, ariel.alcazar17ee@cenidet.edu.mx, velaluis@cenidet.edu.mx

**ITESM, campus Cuernavaca, A01421754@itesm.mx

Abstract: Se reporta la comparación de dos programas en la identificación de dos procesos, uno de ellos un programa neurodifuso (ANFIS – Adaptive Neuro-Fuzzy Inference System) y el otro de Redes Neuronales (NNSYSID20), el primero es un programa general que se adaptó en la medida de lo posible al área de control y el segundo es un programa especialmente creado para su uso en esta área. La comparación se realiza a través de ejemplos de referencia (benchmark) encontrados en la literatura, uno de ellos una columna de destilación y el segundo el actuador hidráulico de una grúa, hasta donde se conoce, en la literatura no existe este tipo de comparaciones y en medio de la abundancia de software dedicado tanto a redes neuronales como a sistemas difusos, quisimos señalar lo que nos ha parecido algo práctico, didáctico y sencillo de utilizar. En ambos equipos se trabajan ejemplos SISO, pero en la columna también se presenta el caso MISO con dos entradas y una salida. Con base en los resultados obtenidos, podemos asegurar que se logra la identificación de los sistemas con un error y un número de parámetros adecuado. Los programas ANFIS y NNSYSID20 son de software libre que se encuentra en internet, aunque es necesaria la plataforma de Matlab®.

Keywords: Identificación Procesos, Redes Neuronales, Lógica Difusa, ANFIS, Columnas de Destilación

1. INTRODUCCIÓN

Actualmente y debido al continuo incremento en la capacidad y rapidez de las computadoras, ha sido posible encontrar modelos de procesos que pueden incluso calcularse en línea, pudiendo actualizar en segundos una gran cantidad de parámetros con métodos iterativos como el Descenso más Rápido (Steepest Descent), en el cual el gradiente es obtenido por invocación sucesiva de la regla de la cadena, el procedimiento es conocido como Retropropagación del Error ó regla de aprendizaje por retropropagación (Rumelhart et al. 1986), debido a que se calcula en sentido opuesto al flujo de las señales. Los programas de redes neuronales que hace algunos años tardaban minutos ahora se ejecutan en segundos. La literatura sobre redes neuronales ha crecido acorde con la rapidez de cómputo, por ejemplo (Hunt, 1995; Henson, 1997; Patan, 2008; Poznyak, 2001; Mrugalski, 2014), en este artículo nos enfocaremos en los que a nuestro parecer son los que mejor nos han funcionado, y los cuales cuentan con software libre. Estos los reconoceremos como ANFIS (Adaptive Neuro-Fuzzy Inference System) un programa neurodifuso y el otro como NNSYSID20 un programa de Redes Neuronales, ambos pueden encontrarse en internet.

Para conseguir los datos entrada-salida de la columna de destilación fue necesario programar el modelo con las ecuaciones proporcionadas por la referencia original, para la grúa los datos se encontraron en internet.

El artículo está dividido en cinco secciones, en la sección 2 se describen en forma breve los métodos de cálculo, en la primera subsección las redes neurodifusas y en la segunda las redes neuronales. En la sección 3 se describen los procesos, por una parte la columna de destilación (Eskinat, 1991) y por la otra el actuador hidráulico de la grúa (Norgaard, 2000). En la sección 4 se presentan los resultados obtenidos en la identificación de ambos sistemas y se comparan por medio de los errores obtenidos en la salida del modelo con los datos originales, como índice de comparación se utiliza la raíz cuadrada del promedio de la suma de los errores al cuadrado (RMSE-Root Mean Square Error). Los datos en general se dividen en dos, la primera mitad son los datos de entrenamiento y la segunda los datos de validación, para la columna de destilación se cuenta con 100 datos y para el actuador hidráulico 1024.

Finalmente en la sección 5 se presentan las conclusiones y se proporciona una página web con el programa de la columna de destilación, en las referencias se dan los datos para descargar el software mencionado (MathWorks) el cual incluye manuales de operación, solo es necesario registrarse.

2. MÉTODOS DE IDENTIFICACIÓN DE SISTEMAS

2.1 Redes Neurodifusas.

(Jang, 1997) propuso un sistema de inferencia neurodifusa adaptable (ANFIS), el objetivo es utilizar reglas difusas comprensibles para ajustar los modelos de los sistemas, éstas tratan de copiar el modo como se hablan las personas regularmente y permiten incorporar el conocimiento humano para inferir y tomar decisiones. El nombre neurodifuso se emplea porque ya construido el sistema difuso, cada parte se considera como una neurona, los parámetros se encuentran con los métodos de las redes neuronales, gradiente descendente y retropropagación.

Para entender la arquitectura del ANFIS, tenemos el ejemplo de la figura 1 (Jang, 1997), se tienen dos entradas x, y , una salida f , con un sistema Sugeno de primer orden con dos reglas, cada regla tiene dos antecedentes (A, B) que son difusos (no lineal) y un consecuente (f) que es lineal. Si en el antecedente se tuviera una campana generalizada,

$$bell(x; a, b, c) = \frac{1}{1 + \left| \frac{x - c}{a} \right|^{2b}}$$

los parámetros a determinar serían a, b, c de cada función de pertenencia del antecedente y p, q, r de cada regla del consecuente, por lo que en total tendríamos 18 parámetros

Regla 1: *If x is A₁ and y is B₁, then f₁ = p₁x + q₁y + r₁*

Regla 2: *If x is A₂ and y is B₂, then f₂ = p₂x + q₂y + r₂*

Y la arquitectura ANFIS quedaría,

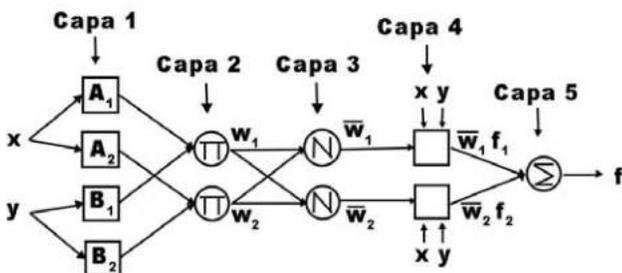
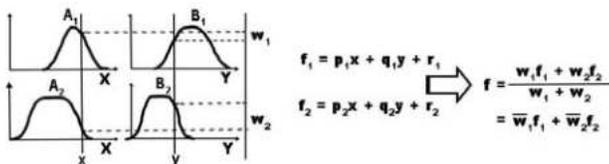


Fig. 1. Un modelo difuso Sugeno de primer orden con dos reglas y su arquitectura ANFIS equivalente (Jang, 1997)

La capa 2 etiquetada como Π , es el producto de las señales de entrada,

$$w_i = \mu_{A_i}(x)\mu_{B_i}(y), \quad i = 1,2$$

La capa 3 etiquetada como N , es la ponderación del peso de cada regla,

$$\bar{w}_i = \frac{w_i}{w_1 + w_2}, \quad i = 1,2$$

Una de las ventajas de este tipo de red, es que al momento de iterar en dos pasos para encontrar los parámetros, en el paso hacia adelante se dejan fijos los parámetros del antecedente y se obtienen los parámetros del consecuente con mínimos cuadrados porque son lineales, para el paso hacia atrás (retropropagación del error) se dejan fijos los parámetros del consecuente y se obtienen los parámetros del antecedente no lineales por el método del gradiente, de esta manera el problema se reduce significativamente.

2.2 Redes Neuronales

Las redes que utilizaremos serán las de (Norgaard, 2000), éstas consisten de una capa oculta en donde tendremos en cada neurona tangentes hiperbólicas como función de activación (f_i) y una capa de salida que será lineal (F_i). Cada neurona recibe las entradas φ_i multiplicadas por los parámetros más un bias (umbral), como se ve en la figura 2

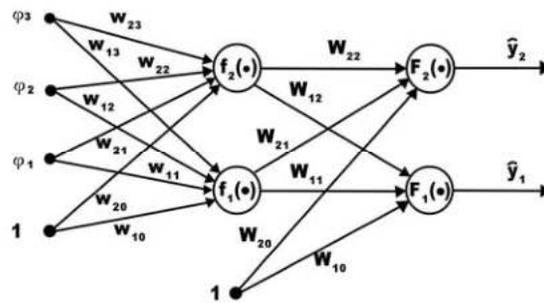


Fig. 2 Red neuronal totalmente conectada con tres entradas, dos unidades ocultas y dos salidas

Los sistemas pueden ser SISO, MISO, MIMO, aquí hay que hacer una diferencia, las NN (Redes Neuronales) pueden tener varias entradas de una sola variable y una salida, éstas serán SISO, en caso de que tuvieran varias variables de entrada serían MISO, esto está relacionado con los sistemas ARX, ARMAX, etc.

Otra confusión que en general también sucede es que uno espera que las entradas sean entradas físicas, en NN no es el caso, las entradas pueden ser las salidas retrasadas en el tiempo, de esta manera obtenemos redes dinámicas, un ejemplo en la figura 3.

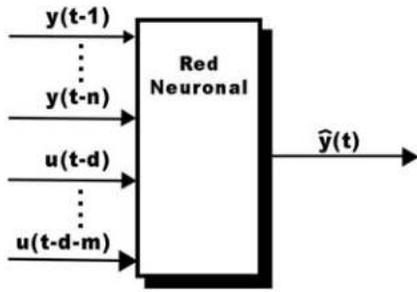


Fig. 3 Estructura de modelo NNARX

Se tienen las entradas físicas (u), pero también entran las salidas (y), son varias porque entran retrasadas en el tiempo, a éste tipo de configuración y en concordancia con sistemas lineales se le llamará NNARX (Neural Network AutoRegressive, eXternal Input), esta es una de las ventajas de estas redes pues utiliza nomenclatura ya conocida en el campo del Control.

3. PROCESOS DE PRUEBA

3.1 Columna de destilación

En la figura 4 se presenta la columna de destilación y en la tabla 1 las propiedades del diseño de ésta.

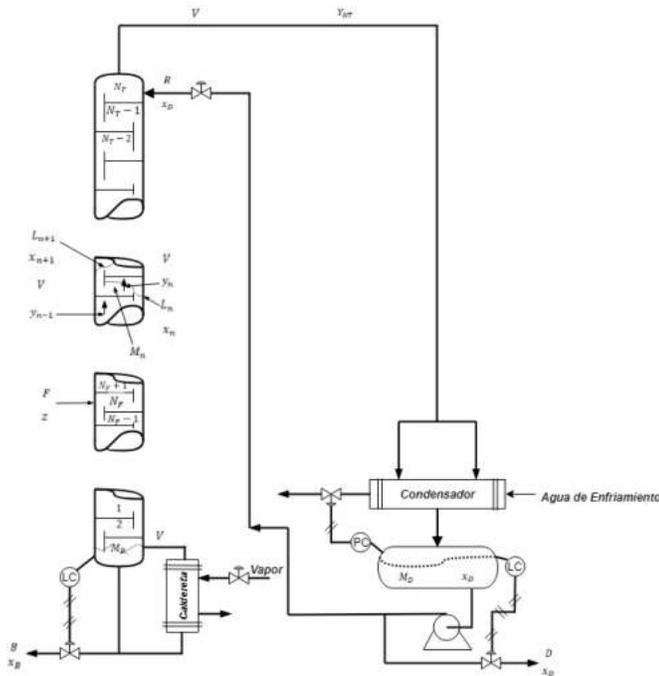


Fig. 4 Columna de destilación (Luyben, 1990)

Tabla 1. Propiedades usadas en las simulaciones de la columna de destilación (Eskinat, 1991)

Número de platos	=25	Volatilidad relativa	=2
Plato de alimentación	=12	Flujo destilado	=0.5
Composición de alimentación	=0.5	Flujo del fondo	=0.5
Composición de destilado	=0.995	Capacidad de los platos	=0.5
Composición del fondo	=0.005	Capacidad del	
Reflujo	=1.477	condensador y del fondo	=0.5
Evaporación	=1.977		

(Eskinat, 1991) utiliza Hammerstein models para representar el modelo de la planta no lineal, estos modelos están compuestos de un elemento no lineal estático en serie con una parte lineal dinámica, como puede verse en la figura 5.

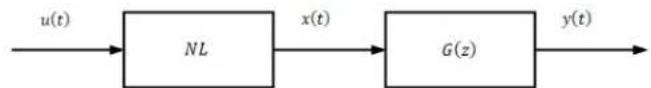


Fig. 5 Modelo Hammerstein

La columna se simuló con una entrada pseudorandom con una amplitud máxima de 10% del valor de estado estable, el reflujo a la columna (R) fue el que se cambió, se obtuvieron 100 datos con un tiempo de muestreo de 10 min. Como salida se tiene el flujo de destilado (D), más adelante en la sección de resultados se reproduce está entrada-salida junto con el obtenido con nuestro modelo de redes neuronales (NNSYSID20).

3.2 Actuador Hidráulico de una Grúa

La planta se presenta en la figura 6 (Norgaard, 2000), en la que el actuador se usa para controlar la posición del brazo de una grúa. Ésta tiene 4 actuadores, poste (boom), brazo (arm), extensión telescópica (telescopic extension), y rotación total de la grúa (rotation of the whole crane), el actuador considerado en este ejemplo es el que controla la articulación inferior del brazo

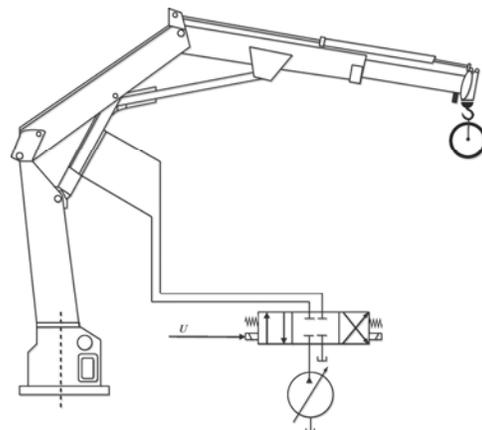


Fig. 6 Grúa hidráulica (Norgaard, 2000)

El conjunto de datos son la abertura de la válvula contra la presión del aceite, se tienen 1024 datos, 512 para entrenamiento y 512 para validación, en la sección de resultados se presentarán estas entrada-salida junto con nuestro modelo de ANFIS

4. RESULTADOS

4.1 Resultados columna de destilación.

Para las señales de entrada a columnas de destilación, (Eskinat, 1991) menciona que el uso de secuencias binarias pseudoaleatorias (PRBS) crea problemas en los algoritmos de estimación debido a que las señales de entrada y salida resultantes cambian rápidamente y el algoritmo usualmente converge a un mínimo local, por lo que emplea señales pseudorandom con amplitud aleatoria, como puede verse en las señales de entrada de las figuras abajo.

En la figura 7 se colocan los resultados obtenidos por (Eskinat, 1991), con un modelo Hammerstein de 3er orden en la no linealidad y de primer orden en la parte lineal, descrito por las siguientes ecuaciones,

$$y(t) + a_1y(t - 1) = b_1x(t - 1)$$

$$x(t) = \gamma_1u(t) + \gamma_2u^2(t) + \gamma_3u^3$$

$a_1 = -0.796$, $b_1 = 0.204$, $\gamma_1 = 1.01$, $\gamma_2 = -7.13$, $\gamma_3 = -7.43$, en total cinco parámetros, el error obtenido entre los datos reales y los del modelo es $RMSE = 0.08834$

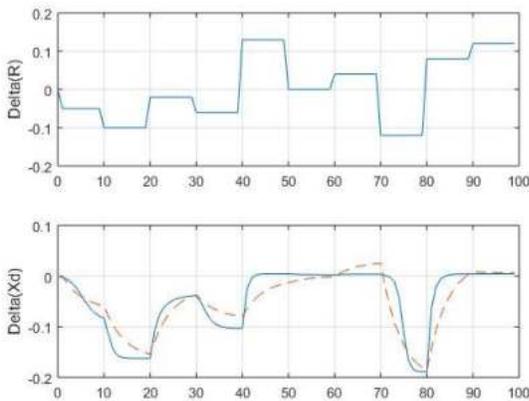


Fig. 7 Respuestas simuladas del modelo de 3er orden estimado por Eskinat vs. datos

A continuación se presentan los resultados obtenidos por una red neuronal (NNARX111) que es una red neuronal con dos entradas (reflujo y destilado), y un retraso de tiempo entre la salida y la entrada, la capa oculta es con 1 neurona tanh y la salida lineal, por lo que el número de parámetros es 5, igual que (Eskinat, 1991).

En la figura 8 se muestra la salida estimada vs los datos, así como la gráfica del error, el RMSE es 0.01371,

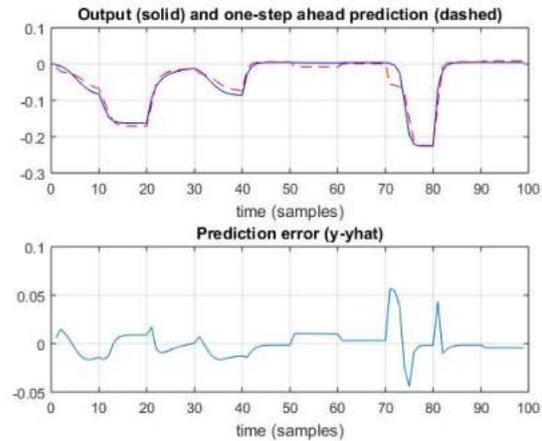


Fig. 8 Respuestas simuladas del modelo estimado vs. datos

Para el caso MISO se anexan los resultados obtenidos por una red neuronal NNARX, con entradas $NN = [1 \ 1 \ 1] [1 \ 1]$, esto es tres entradas (reflujo, vapor y destilado) y un retraso de tiempo entre la salida y cada variable de entrada

En la figura 9 se presentan los datos de entrada, se colocan como en la referencia original utilizando variables de desviación (variable medida – variable de estado estable), los cambios considerados son en las variables de reflujo y vapor.

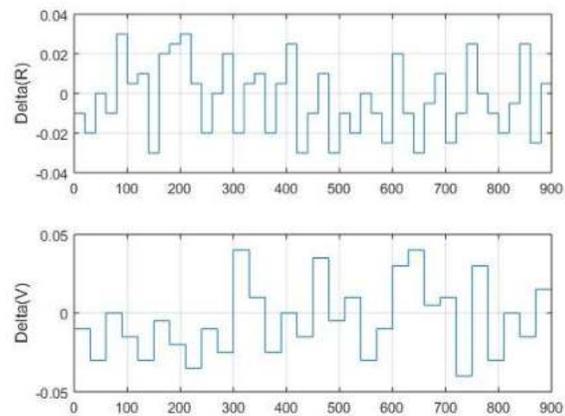


Fig. 9 Datos de entrada para el modelo MISO

La red tiene 4 tanh en la capa oculta y la salida es lineal, por lo que el número de parámetros es 21, el $RMSE = 2.651 \times 10^{-3}$. Eskinat (1991) utiliza 34 parámetros y no se dispone de información del error, pero el resultado obtenido con esta red se muestra cualitativamente mejor, ver figura 10.

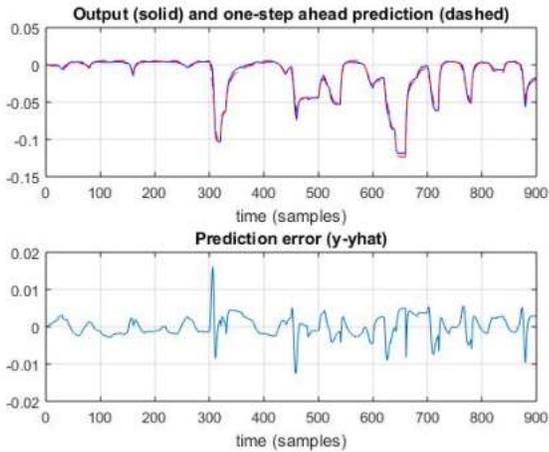


Fig. 10 Respuesta del sistema MISO y errores de predicción

4.2 Resultados actuador hidráulico de una grúa

El actuador de la grúa se controla cambiando la abertura de la válvula a través de la cual fluye aceite. Los valores medidos de la abertura de la válvula (entrada) y la presión de aceite (salida) se muestran en la figura 11.

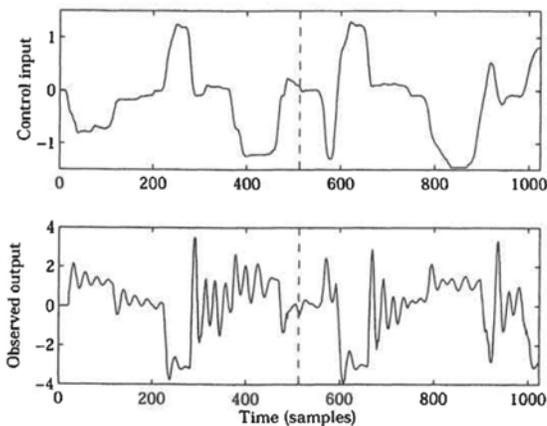


Fig. 11 Datos entrada/salida divididos para entrenamiento y validación del actuador de la grúa

En esta sección a diferencia de la anterior usaremos el programa ANFIS porque la referencia original (Norgaard, 2000) utiliza NNSYSID20.

El ANFIS empleado contiene 5 entradas $[u(t-1), u(t-2), y(t-1), y(t-2), y(t-3)]$ y una salida $[y(t)]$, como puede verse se emplean señales atrasadas tanto de la entrada física como de la salida (como en la figura 3), para cada entrada se emplean dos funciones de pertenencia campana generalizada, y como consecuente Sugeno de primer orden, por lo que se obtienen 32 reglas difusas, la información que produce el programa es la siguiente,

ANFIS info:
 Number of nodes: 92
 Number of linear parameters: 192
 Number of nonlinear parameters: 30
 Total number of parameters: 222
 Number of training data pairs: 512
 Number of checking data pairs: 512
 Number of fuzzy rules: 32

Los resultados obtenidos con los datos de validación tienen un RMSE de 0.2218, en la figura 12 se presentan éstos y los errores de predicción, cabe mencionar que no se ha intentado optimizar éstos.

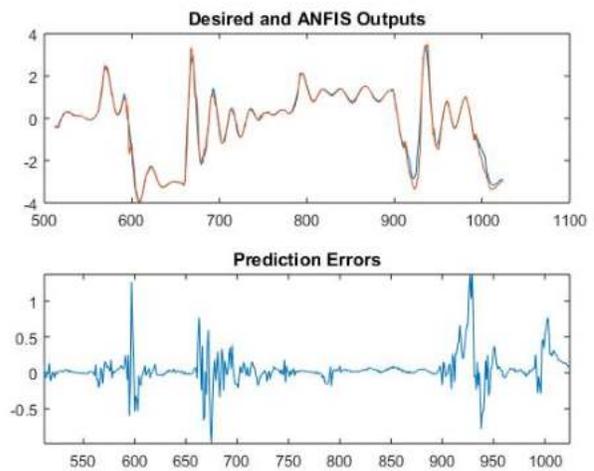


Fig. 12 Respuesta del sistema y errores de predicción

Norgaard (2000), reporta que utilizó una red NNARX(3,2,1) con 8 neuronas ocultas y obtuvo un RMSE de 0.38, con una red completa, esto es sin reducirla (“pruning”), el total de parámetros es de 57.

Resumen de Resultados,

Equipo	Método	RMSE	No. parámetros
Columna destilación	SISO Eskinat	8.834×10^{-2}	5
	SISO NNSYSID20	1.371×10^{-2}	5
	MISO Eskinat	---	34
	MISO NNSYSID20	2.651×10^{-3}	21
Actuador Grúa	ANFIS	0.2218	222
	NNSYSID20	0.38	57

En el actuador de la Grúa se comparan los dos métodos propuestos ANFIS vs NNSYSID20 y como puede verse en la tabla, al utilizar redes neuronales generalmente se tiene la disyuntiva entre precisión y número de parámetros, la selección dependerá de las prioridades del problema en cuestión.

5. CONCLUSIONES

Como se había mencionado en el abstract el ANFIS es un programa general que se adaptó en la medida de lo posible al área de control dando buenos resultados, pero el NNSYSID20 es un programa especialmente creado para su uso en control y en el diseño de controladores. Algunas de las características adicionales que pueden obtenerse de éste último y que por falta de espacio no se incluyeron aquí, están por ejemplo la obtención de las gráficas de Auto-correlación del error, la Cross-correlación de las entradas con el error, la linealización a cada instante de muestreo de los parámetros de la red, la posibilidad de reducir el número de parámetros por medio de poda (pruning), la obtención del histograma de errores de predicción, e incluso puede dibujarse la red para tener una idea de su configuración.

Creemos que el ANFIS puede mejorarse a través de funciones que complementen su alcance para aplicarse al área del control, este será uno de los caminos que seguiremos en el futuro, ya que es una ventaja el poder expresar mediante reglas el conocimiento del proceso y obtener la solución mediante los métodos de NN.

En trabajos posteriores se verá cómo utilizar los modelos identificados para su uso en el diseño de controladores. El programa para simular la columna de destilación se encuentra en la página web <http://equinterom.xyz/> en la sección de Control de Procesos

6. AGRADECIMIENTOS

El Dr. Enrique Quintero menciona que algunos de estos resultados se han encontrado después de varios años de impartir el curso de Control e Inteligencia Computacional en el CENIDET, y agradece a las generaciones entusiastas de alumnos que han apoyado a través de sus comentarios, tareas y proyectos.

REFERENCIAS

- MathWorks, File Exchange, <https://la.mathworks.com/matlabcentral/fileexchange/87-nnsysid>
<https://la.mathworks.com/matlabcentral/fileexchange/86-nnctrl>
<https://la.mathworks.com/matlabcentral/fileexchange/2173-neuro-fuzzy-and-soft-computing>
- Mrugalski, M. Advanced Neural Network-Based Computational Schemes for Robust Fault Diagnosis. Springer International Publishing, 2014
- Norgaard, M., O. Ravn, N.K. Poulsen, L.K. Hansen. *Neural Networks for Modelling and Control of Dynamic Systems*. Springer-Verlag 2000
- Patan, K. Artificial neural networks for the modelling and fault diagnosis of technical processes. Springer-Verlag, 2008
- Poznyak, A.S., E.N. Sanchez, W. Yu. Differential Neural Networks for Robust Nonlinear Control. World Scientific Publishing Co., 2001
- Rumelhart, D.E., G.E. Hinton, R.J. Williams. Learning internal representations by error propagation. In D.E. Rumelhart and J.L. McClelland, editors, *Parallel distributed processing: explorations in the microstructure of cognition*, volumen 1, chapter 8, pages 318-362. MIT Press, 1986
- Eskinat, E., S.H. Johnson, W.L. Luyben. "Use of Hammerstein models in identification of nonlinear systems". *AIChE Journal*, vol. 37, no.2, 1991
- Henson, M.A., D.E. Seborg, editors. *Nonlinear Process Control*. Prentice Hall, 1997
- Hunt, K, G. Irwin, K. Warwick. *Neural Network Engineering in Dynamic Control Systems*. Springer-Verlag, 1995
- Jang, J., C. Sun, E. Mizutani. *Neuro-Fuzzy and Soft Computing*. Prentice Hall 1997
- Luyben, W.L. *Process Modeling, Simulation and Control for Chemical Engineers*. McGraw-Hill, 1990 2nd ed.