

Diseño, simulación y comparación de controladores inteligentes aplicados a un proceso industrial

Sergio L. Martínez¹, Enrique E. Tarifa^{1,2} & Álvaro F. Nuñez¹

⁽¹⁾ *Facultad de Ingeniería, Universidad Nacional de Jujuy.*

smartinez@fi.unju.edu.ar, eetarifa@fi.unju.edu.ar; alfanunez@yahoo.com.ar

⁽²⁾ *Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET).*

eetarifa@arnet.com.ar

RESUMEN: En este trabajo se diseñan e implementan controladores inteligentes tipo MIMO, uno basado en lógica fuzzy y otro basado en redes neuronales. Se aplican al modelo de un mezclador de corrientes líquidas, configurado sobre el entorno de simulación gráfica de Matlab[®]. Se comprueba el desempeño de ambos controladores aplicando variaciones abruptas de las variables de referencia en forma simultánea. Se comparan los resultados de ambos controladores, rescatándose que aunque presentan un comportamiento operativo bastante similar, el controlador fuzzy se destaca por su facilidad de diseño y configuración.

Área: Ciencias Aplicadas. Disciplina: Computación. Subdisciplina: Inteligencia Artificial.

1 INTRODUCCIÓN

El control de equipos y procesos es una actividad indispensable en numerosos ámbitos, especialmente en la industria. Usualmente los sistemas de control convencionales requieren un conocimiento muy preciso del sistema sobre el que se aplican, ya que, cuando operan sobre un modelo necesitan de los datos que el sistema genera. La obtención de tal modelo no es siempre posible.

Los continuos avances en el campo de la Inteligencia Artificial, ha permitido generar diversos modelos de sistemas de control, en general de tipo adaptivo y basados en conocimiento, que rápidamente están reemplazando a los sistemas de control tradicionales. En la bibliografía pertinente (Chen et al, 2011; Chen, 2009; Labiod et al., 2005; Gu & Hu, 2002), se analizan algunos de estos tipos como casos típicos de control MIMO.

Los controladores basados en lógica fuzzy (FLC) han demostrado ser muy eficientes, asociados a un proceso de diseño relativamente fácil y accesible, aplicando conceptos basados en lenguaje natural. Utilizan heurísticas de control que no requieren específicamente de un modelo del proceso, pero la estrategia de control en la que se basan está sustentada por el conocimiento de un experto a partir del cual, se derivan las reglas de control (Yager, 2002). Los controladores neuronales son muy versátiles en el proceso de control, ya que las redes neuronales (RNA) se especializan en descubrir y asociar patrones de entrada-salida con relación lineal o no lineal, según sea su arquitectura, configuración de las neuronas y proceso de aprendizaje. De acuerdo a la

forma del agrupamiento y al tipo de interconexión se obtienen diferentes arquitecturas de redes, algunas de propósito general, y otras concebidas para aplicaciones específicas (He & Xu, 2009).

En este trabajo se diseñan e implementan controladores MIMO (*Multiple-input Multiple-output*, múltiple entrada múltiple salida), uno basado en lógica fuzzy y otro en redes neuronales, configurados con el formato convencional tipo integral.

Estos controladores se aplican a un proceso industrial muy utilizado, que es el mezclador de caudales en línea, con un modelo simplificado, que permite fácilmente generar e interpretar las acciones de control sobre pruebas simuladas.

2 DISEÑO DE CONTROLADORES

La ley general de control convencional define varios tipos de estructuras de control, destacándose las configuraciones de tipo Proporcional (P), Proporcional-Integral (PI), Proporcional-Derivativo (PD) y la más general Proporcional-Integral-Derivativo (PID). Los modelos de controladores no convencionales en general responden a estos tipos, dependiendo de su diseño y configuración.

El proceso de control se puede esquematizar en forma genérica, como se muestra en la Figura 1 siguiente.

El sistema a controlar –usualmente referenciado como planta–, genera a la salida las variables controladas (lo que el sistema produce), que son realimentadas y comparadas con sus respectivas variables de referencia o *set points* (lo que se

desea que produzca), obteniéndose los errores o desviaciones que el controlador debe ajustar.

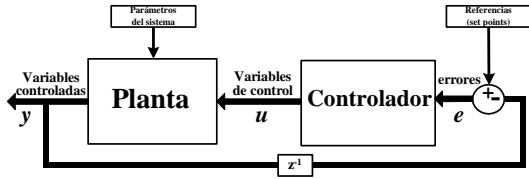


Figura 1. Esquema controlador-planta.

La ley de control general establece que la salida de control u se define como:

$$u(k) = u_0 + K_p e(k) + K_I \Sigma e(k) + K_D \Delta e(k) \quad (1)$$

que representa a una estructura tipo PID, donde k es el tiempo (discreto en este caso), u_0 es la componente en vacío, K_p , K_I y K_D son los parámetros del controlador y cada término define un tipo de configuración, tal que la configuración proporcional (P) es función del error $e(k)$ del sistema; la configuración integral (I) es función de la suma de los errores producidos $\Sigma e(k)$ y la configuración derivativa (D) responde a la variación entre el error actual y anterior $\Delta e(k)$ producido por el sistema. Según sea el diseño del controlador, la acción de control puede expresarse en forma absoluta (ec. (1)), o como una variación de la acción de control $\Delta u(k)$, tal que el valor total de control se calcula como:

$$u(k) = u(k-1) + \Delta u(k) \quad (2)$$

luego, con las ecuaciones (1) y (2) se puede deducir la variación de la acción de control que resulta:

$$\Delta u(k) = K_p (e(k) - e(k-1)) + K_I e(k) + K_D (e(k) - 2e(k-1) + e(k-2)) \quad (3)$$

$$\Delta u(k) = K_0 e(k) + K_1 e(k-1) + K_2 e(k-2) \quad (4)$$

$$\text{con } \begin{cases} K_0 = (K_p + K_I + K_D) \\ K_1 = -(K_p + 2K_D) \\ K_2 = K_D \end{cases}$$

A partir de las ecuaciones (1) ó (3) se pueden configurar los diferentes tipos estándares de controladores, como estructuras combinadas o separas, según se requiera. Por ejemplo, anulando los parámetros K_I y K_D , se obtiene un controlador tipo proporcional:

$$u(k) = K_p e(k)$$

$$\Delta u(k) = K_p (e(k) - e(k-1)) \quad (5)$$

$$\text{con } u_0 = 0 \text{ y } K_I = K_D = 0$$

La eliminación de los parámetros K_p y K_D el controlador resultante es de tipo integral, cuyas ecuaciones representativas son:

$$u(k) = K_I \Sigma e(k)$$

$$\Delta u(k) = K_I e(k) \quad (6)$$

$$\text{con } u_0 = 0 \text{ y } K_p = K_D = 0$$

Anulando el parámetro K_D , se obtiene un controlador tipo proporcional-integral:

$$u(k) = K_p e(k) + K_I \Sigma e(k)$$

$$\Delta u(k) = K_p (e(k) - e(k-1)) + K_I e(k) \quad (7)$$

$$\text{con } u_0 = 0 \text{ y } K_D = 0$$

De manera similar, la anulación del parámetro K_I genera la estructura del controlador tipo proporcional-derivativo:

$$u(k) = K_p e(k) + K_D \Delta e(k)$$

$$\Delta u(k) = K_p (e(k) - e(k-1)) + K_D (e(k) - 2e(k-1) + e(k-2)) \quad (8)$$

$$\text{con } u_0 = 0 \text{ y } K_I = 0$$

La selección de las diferentes combinaciones, dependerá del tipo de control que se requiera, haciendo uso de las ventajas de cada una. En base a estos conceptos, se diseñan e implementan controladores fuzzy basados en las configuraciones descriptas, que luego son aplicadas a un proceso industrial como se detalla a continuación.

3 PROCESO A CONTROLAR

Los dispositivos conocidos como mezcladores de flujos o de caudales (*flow mixers*), mezclan corrientes de fluidos para producir una nueva corriente con propiedades específicas (caudal, presión, temperatura, composición, etc.). Son muy utilizados como complemento de diversos procesos industriales (Shao *et al.*, 2010; Montante *et al.*, 2006; Baksaran *et al.*, 2003; Balannec *et al.*, 2002; Oshinowo *et al.*, 2001), y suele estar sometido a dinámicas exigentes, siendo necesaria la asistencia de sistemas de control automáticos.

Un modelo simplificado del sistema mezclador de caudales se muestra en la figura siguiente. La corriente de entrada, que se identifica como fría, tiene un caudal máximo F_f y una temperatura T_f . Esta corriente es regulada por la apertura x_f de la

válvula V_f . La otra corriente de entrada, que se identifica como caliente, tiene un caudal máximo F_c y una temperatura T_c . Esta corriente es regulada por la apertura x_c de la válvula V_c . Las aperturas x_f y x_c incursionan en el intervalo $[0, 1]$, donde 0 corresponde a la válvula completamente cerrada y 1 a la válvula completamente abierta, de modo que permite pasar la totalidad del caudal asignado. La corriente mezcla presenta a la salida un caudal F y una temperatura T .

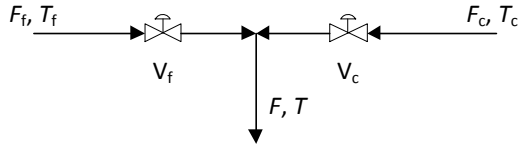


Figura 2. Esquema de mezclador de caudales en línea.

El sistema descrito requiere un controlador del tipo MIMO. Las variables controladas son F y T , mientras que las variables manipuladas son x_f y x_c .

3.1 Modelo del mezclador en línea

El modelo estático del mezclador en línea mostrado en la Figura 2, se puede representar con el siguiente conjunto de ecuaciones:

$$F = x_f F_f + x_c F_c \quad (9)$$

$$T = \frac{x_f F_f T_f + x_c F_c T_c}{x_f F_f + x_c F_c} \quad (10)$$

$$0 \leq x_f \leq 1 \text{ y } 0 \leq x_c \leq 1 \quad (11)$$

Se asume que los sensores de las variables controladas están cerca del punto de mezcla, por lo cual el retardo de las mediciones es despreciable.

El modelo experimental se ha implementado sobre el entorno de simulación gráfica de Matlab® y se ha instanciado con los siguientes parámetros, que serán aplicados a las pruebas experimentales:

- Corriente fría: caudal de entrada $F_f = 100$ l/min, temperatura $T_f = 25$ °C.
- Corriente caliente: caudal de entrada $F_c = 100$ l/min, temperatura $T_c = 70$ °C.

4 CONTROLADOR FUZZY

Para este caso, se diseña un controlador fuzzy tipo integral (I) basado en las ecuaciones (6). Las variables de entrada-salida se particionan en siete conjuntos fuzzy (NG-negativo grande, NM-negativo medio, NP-negativo pequeño, Z-cero, PP-positivo pe-

queño, PM-positivo medio y PG-positivo grande, generándose la siguiente base de reglas:

Tabla 1. Base de reglas para FLC tipo I.

Δx_f Δx_c	e^T						
	NG	NM	MP	Z	PP	PM	PP
NG				NG			
NM				NM			
NP				NP			
Z	PG	PM	PP	Z	NP	NM	NG
PP	NG	NM	NP	Z	PP	PM	PG
PM				PP			
PG				PM			
				PG			

El modelo completo controlador-planta, fue configurado en el entorno de simulación gráfica de Matlab® como se muestra en la Figura 3 siguiente.

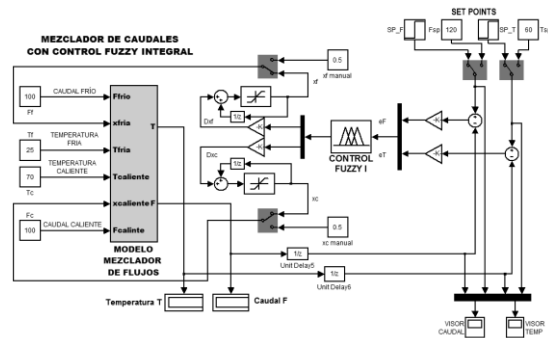


Figura 3. Modelo control-planta con controlador fuzzy tipo integral.

En este esquema, el sistema a controlar se incorpora como un bloque que recibe los parámetros de caudales de entrada (F_f y F_c), de temperatura (T_f y T_c) de tales caudales y las variables de control (x_f y x_c), produciendo las variables de salida –caudal (F) y temperatura (T)–.

La respuesta del controlador se comprueba aplicando variaciones abruptas (tipo escalón) a las variables de referencia caudal (F_{sp}) y temperatura (T_{sp}), considerando que representan las condiciones más exigentes para su desempeño.

Para los requerimientos de las variables de referencia, el sistema genera una muy buena respuesta. Las aproximaciones iniciales en $t_k = 0$ se deben a las condiciones iniciales definidas en el modelo de simulación. Se observa una ligera desviación de la

temperatura en $t_k = 40$, en el instante donde el caudal cambia el valor de referencia y un cambio de offset para el caudal en $t_k = 60$ donde la temperatura cambia de referencia.

El controlador fuzzy no genera sobrepicos y las aproximaciones en los niveles altos y bajos son buenas, destacándose que el offset final para el caudal es del 97 % por defecto y del 95 % por exceso para la temperatura. Las curvas de prueba se presentan en la gráfica siguiente.

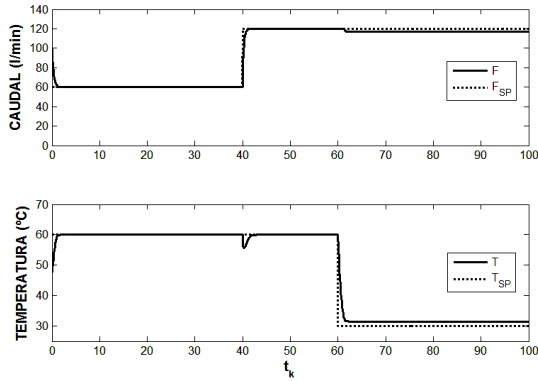


Figura 4. Respuesta del controlador fuzzy tipo I para referencias tipo escalón.

5 CONTROLADOR NEURONAL

En el caso de controladores neuronales, el tipo de red utilizado más frecuentemente es la feedforward multicapa con aprendizaje supervisado (Chen *et al.*, 2011; Chen & Huang, 2004), donde se explota una de sus principales características: la asociación de patrones de entrada-salida prácticamente sobre cualquier función, con la precisión que se desee (Lin, 1996).

Para diseñar un controlador neuronal similar al anterior (tipo I), se configura una RNA feedforward para que las entradas sean los errores de las variables controladas (e_F y e_T), y las salidas sean las correcciones que deben realizarse sobre las variables manipuladas (Δx_f y Δx_c). Estas nuevas variables se definen como:

$$e_F = F_{sp0} - F \quad (12)$$

$$e_T = T_{sp0} - T \quad (13)$$

$$\Delta x_f(k) = x_f(k) - x_f(k-1) \quad (14)$$

$$\Delta x_c(k) = x_c(k) - x_c(k-1) \quad (15)$$

donde k es el tiempo de simulación discreto. Con este cambio de variables, el punto de operación deseado es aquel que anula los errores. Cualquier

desviación de este punto de operación, requiere que el controlador realice correcciones sobre las variables controladas. Es decir, cuando ambas entradas de la RNA sean nulas, sus salidas también lo serán y no es necesaria ninguna acción de control. En cambio cuando una o ambas entradas dejen de ser nulas, las salidas de la RNA deberán proveer las correcciones necesarias sobre las variables de control.

La conducta descrita es independiente del punto base que se tome. Más aún, si la superficie de control es lineal, las magnitudes de las correcciones serán independientes de la posición del punto base. En cambio, si la superficie de control no es lineal, las correcciones dependerán del punto base. En este esquema de control, no es necesario entrenar a la RNA en toda la región de control. Concretamente, bajo este enfoque se establece un punto de operación genérico que representa la situación de referencia general de las variables de control y se definen las condiciones de operación sobre un entorno, mediante valores incrementales a partir del punto de operación establecido. En la FIGURA 5 se muestran los límites de la región de control para el sistema mezclador y el punto de operación definido como punto base utilizado en el entrenamiento de la RNA.

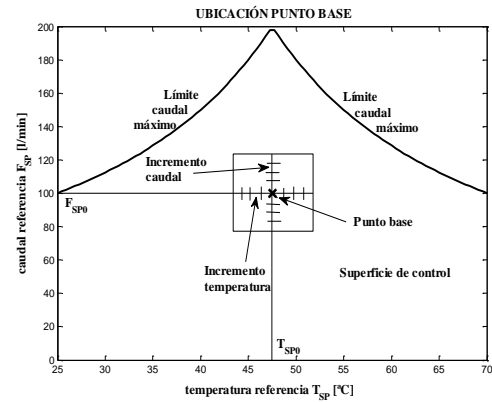


Figura 5. Posición de punto base y entorno incremental del mezclador de flujos.

5.1 Generación de datos

El punto base de operación de la RNA fue definido en la mitad del recorrido de las variables caudal (F) y temperatura (T) que se corresponde con la mitad de las aperturas de las válvulas x_f y x_c del mezclador de caudales.

A continuación se estableció el entorno de operación en $\pm 10\%$ para ambas variables y la variación incremental ΔF y ΔT en el 1%, considerando que es ésta la mínima resolución del sistema. De esta manera, cada punto del entorno de operación queda definido de la siguiente forma:

$$F_{sp_i} = F_{sp0} + (i-1)\Delta F F_{sp0} \quad \text{con } i=1, \dots, 21$$

$$F_{sp}^{\min} = 90 \text{ l/min} \quad F_{sp}^{\max} = 110 \text{ l/min} \quad (16)$$

$$T_{sp_j} = T_{sp0} + (j-1)\Delta T T_{sp0} \quad \text{con } j=1, \dots, 21$$

$$T_{sp}^{\min} = 42.75 \text{ }^\circ\text{C} \quad T_{sp}^{\max} = 52.25 \text{ }^\circ\text{C} \quad (17)$$

El entorno de operación queda particionado en 21 intervalos, obteniéndose un total de 441 puntos de operación incrementales alrededor del punto base, como se esquematizó en la Figura 5.

Definido el entorno de operación, se utilizó el modelo inverso del sistema mezclador de flujos para obtener las acciones de control requeridas para cada uno de los puntos incluidos en el entorno del punto base. En este modelo inverso, las variables de entrada son el caudal de referencia (F_{sp}) y la temperatura de referencia (T_{sp}); mientras que las variables de salida son las aperturas x_f y x_c que permitirán alcanzar el estado deseado.

Finalmente, conocidas las acciones de control necesarias para llegar a cada punto del entorno de operación, se las expresa como cambios requeridos en función de los errores medidos:

$$e_{F_i} = F_{sp0} - F_{sp_i} = -(i-1)\Delta F F_{sp0} \quad (18)$$

$$e_{T_j} = T_{sp0} - T_{sp_j} = -(j-1)\Delta T T_{sp0} \quad (19)$$

$$\Delta x_{f_{ij}} = x_{f_{ij}} - 0.5 \quad (20)$$

$$\Delta x_{c_{ij}} = x_{c_{ij}} - 0.5 \quad (21)$$

5.2 Diseño del controlador neuronal

Para configurar la RNA que actúa como controlador, los parámetros más significativos a considerar son el número de capas y las cantidades de unidades de proceso incluidas. La dimensionalidad de las capas de entrada y salida queda definida por las características del problema, siendo cada una bidimensional, en este caso. De acuerdo con la interpretación del teorema de aproximación de Cybenko (Poznyak *et al.*, 2001), una capa oculta con una cantidad finita de unidades con funciones monótonas crecientes, es suficiente para cualquier mapeo no lineal de entrada-salida con un nivel de error suficientemente bajo.

Luego, el parámetro de mayor criticidad es la cantidad de neuronas ocultas encargadas del procesamiento interno; una cantidad insuficiente de neuronas ocultas puede impedir alcanzar el nivel de error deseado, mientras que una cantidad excesiva puede disminuir la capacidad de generalización de la red. En muchos casos, la determinación de este parámetro se realiza en forma experimental, método que

puede insumir un gran esfuerzo computacional, sin garantías de resultados aceptables.

Existen algunas heurísticas dedicadas a la determinación de la cantidad de neuronas ocultas, que aunque no son matemáticamente rigurosas, pueden producir buenas aproximaciones. Así por ejemplo, se puede citar a la regla de la pirámide geométrica, donde el número de neuronas ocultas ($N^{(o)}$) se obtiene como una progresión geométrica entre el número de neuronas de entrada ($N^{(e)}$) y el número de neuronas de salida ($N^{(s)}$), de la forma $N^{(o)} = \sqrt{N^{(e)} \cdot N^{(s)}}$ (Blum, 1992). La regla de las capas entrada-oculta establece que el número de neuronas ocultas no debe superar dos o tres veces la cantidad de neuronas de entrada (Berry & Linnoff, 2011). Otra regla práctica, utilizada por Goethals *et al.* (2007), sugiere que el número de neuronas ocultas ($N^{(o)}$) se relaciona con el número de neuronas de entrada ($N^{(e)}$) de la forma $N^{(o)} = 2 \times N^{(e)} + 1$.

Considerando las heurísticas anteriores, se definió una RNA feedforward con arquitectura 2+5+2 para actuar como un controlador neuronal incremental tipo MIMO (2 entradas y 2 salidas), con las siguientes características:

- 2 neuronas en la capa de entrada.
- 5 neuronas en la capa oculta. Función de transferencia tangente sigmoide.
- 2 neuronas en la capa de salida. Función de transferencia lineal.

Con los parámetros anteriores definidos, la RNA fue entrenada con el algoritmo *backpropagation* obteniéndose los siguientes resultados:

- Entrenamiento con algoritmo *backpropagation*, variante LM (Demuth *et al.*, 2010).
- Cantidad de iteraciones: 500.
- Error cuadrático medio (ECM) de entrenamiento: 2.11×10^{-10} .

5.3 Configuración y operación del sistema

El modelo completo controlador-planta, fue configurado en el entorno de simulación gráfica de Matlab[®] como se muestra en la Figura 6 siguiente. En este esquema, el sistema a controlar se incorpora como un bloque que recibe los parámetros de caudales de entrada (F_f y F_c), de temperatura (T_f y T_c) de tales caudales y las variables de control (x_f y x_c), produciendo las variables de salida –caudal (F) y temperatura (T)–.

El controlador neuronal, recibe a la entrada el error de caudal ($eF = F_{sp} - F$) y el error de temperatura ($eT = T_{sp} - T$), ambos modulados por limitadores que mantienen a los valores incrementales dentro del entorno definido, mejorando la estabilidad del sistema; y genera las variaciones de apertura de la válvula fría (Δx_f) y de la válvula caliente (Δx_c). Estas variaciones se componen con las

aperturas del estado anterior ($x_f(k) = x_f(k-1) + \Delta x_f(k)$ || $x_c(k) = x_c(k-1) + \Delta x_c(k)$) para ser realimentadas al mezclador de flujos, cerrando el lazo de control.

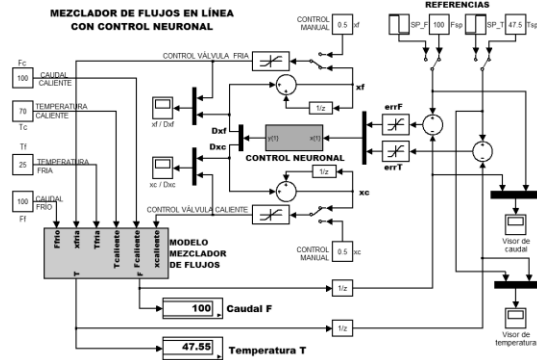


Figura 6. Modelo control-planta con controlador control neuronal tipo I.

5.4 Prueba del sistema

Para comprobar la operación del sistema bajo las mismas condiciones del controlador fuzzy, se aplicaron variaciones abruptas para el caudal y la temperatura como muestra la Figura 7.

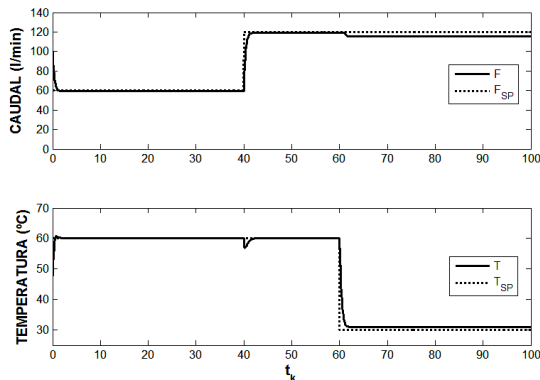


Figura 7. Respuesta del controlador neuronal tipo I.

Para los requerimientos de las variables de referencia, el sistema genera una muy buena respuesta, similar al controlador fuzzy. Las aproximaciones iniciales en $t_k = 0$ se deben a las condiciones iniciales definidas en el modelo de simulación. Se observa, como en el caso del controlador anterior, una ligera desviación de la temperatura en $t_k = 40$, en el instante donde el caudal cambia el valor de referencia y un cambio de offset para el caudal en $t_k = 60$ donde la temperatura cambia de referencia. El controlador neuronal tampoco genera sobrepicos y las aproximaciones en los niveles altos y bajos son buenas, destacándose que el offset final

para el caudal es del 96 % por defecto y del 96 % por exceso para la temperatura.

6 CONCLUSIONES

Se han diseñado dos tipos de controladores inteligentes, que responden a la configuración convencional de tipo integral. El primer controlador se desarrolló con lógica fuzzy con particiones de las variables en siete conjuntos fuzzy y trece reglas de control; el segundo controlador se desarrolló con redes neuronales, con cinco neuronas ocultas y un entrenamiento basado en un punto de operación fijo con particiones incrementales en un entorno de $\pm 10\%$ alrededor de tal punto.

Ambos controladores fueron aplicados a un modelo simplificado de un mezclador de caudales en línea, también diseñado y configurado sobre el entorno de simulación gráfica de Matlab®.

Las pruebas de los conjuntos controlador-planta se realizaron con variaciones abruptas (tipo escalón), aplicados simultáneamente a las variables de referencia caudal (F_{SP}) y temperatura (T_{SP}). Los resultados mostraron comportamientos similares con muy buen desempeño para ambos controladores.

Se puede destacar que aunque ambos controladores responden a configuraciones del mismo tipo –integral en este caso–, el proceso de diseño del controlador fuzzy es mucho más simple, no requiere la generación de datos previos, ni un proceso de entrenamiento como en el caso neuronal; para la generación de las reglas de control, es necesario un conocimiento superficial del modelo a controlar.

7 REFERENCIAS

- Baksaran K., L.M. Palmowsky & B.M. Watson. Wastewater reuse and treatment options for the dairy industry. *Water Science and Technology Water Supply*, Vol. 3(3), pp. 85-91. 2003.
- Balanec B., G. Gésan-Guizoui, B. Chaufer, M. Rabiller-Baudry & G. Daufin. Treatment of dairy process waters by membrane operations for water reuse and milk constituents concentration. *Desalination*, V. 147, pp. 89-94. 2002.
- Berry M. J. A. y G. S. Linoff. *Data Mining Techniques: For Marketing, Sales, and Customer Relationship Management*. John Wiley & Sons, ISBN 978-0-470-65093-6. New York, 2011.
- Blum A. *Neural Networks in C++: An Object-Oriented Framework for Building Connectionist Systems*. Ed. John Wiley & Sons Inc, ISBN-13: 978-0471538479. New York, 1992.
- Chen C.S., Dynamic structure adaptive neural fuzzy control for MIMO uncertain nonlinear systems, *Information Sciences*, Volume 179, Issue 15, 2676–2688, 2009.
- Chen J. & T.C. Huang. Applying neural networks

- to on-line updated PID controllers for nonlinear process control. *Journal of Process Control*, V.14, pp. 211–230, 2004.
- Chen J., Z. F. He, X. Qi, A new control method for MIMO first order time delay non-square systems, *Journal of Process Control*, Volume 21, Issue 4, 538–546, 2011.
- Goethals P.L., A.P. Dedecker, W. Gabriels, S. Lek & N. De Pauw, Applications of artificial neural networks predicting macroinvertebrates in freshwaters. *Springer - Aquatic Ecology*, V. 41, pp. 491-508, 2007.
- Gu D. & H. Hu, Neural Predictive Control for a Car-like Mobile Robot. *International Journal of Robotics and Autonomous Systems*, V. 39(2-3), 2002.
- He X. & S. Xu, *Process Neural Networks - Theory and Applications*, Springer-Verlag, Berlín, 2009.
- Labiod S., M.S. Boucherit & T.M. Guerra, Adaptive fuzzy control of a class of MIMO nonlinear systems, *Fuzzy Sets and Systems*, Volume 151, Issue 1, 59–77, 2005.
- Lin C.T. & C.S. George Lee. *Neural Fuzzy Systems. A Neuro-Fuzzy Synergism to Intelligent Systems*. Ed. Prentice Hall PTR, ISBN N° 0-13-235169-2. New Jersey, 1996.
- Oshinowo L.M., A. Bakker & E.M. Marshall. Virtual Efficiency - Simulating Mixing Impellers Using Computational Fluid Dynamics. *Flow Control*, V. VII(7), pp. 28-33. 2001.
- Poznyak A.S., E.N. Sanchez & W. Yu. *Differential Neural Networks for Robust Nonlinear Control*. Ed. World Scientific Publishing, ISBN 981-02-4624-2. Singapore, 2001.
- Shao X., L. Lynd, A. Bakker, R. LaRoche & C. Wyman. Reactor scale up for biological conversion of cellulosic biomass to ethanol. *Bioprocess and Biosystems Engineering*, V. 33(4), pp 485-493, 2010.
- Yager R. R. *Essential of Fuzzy Modelling and Control*. Co-CBS. USA, 2002.