Controlador neuronal incremental aplicado a un mezclador de flujos

Sergio L. Martínez[1], Enrique E. Tarifa[1,2] & Samuel Franco Dominguez[1]

(1) Facultad de Ingeniería, Universidad Nacional de Jujuy,
Gorriti 237, S. S. de Jujuy, Jujuy, Argentina
{smartinez, eetarifa, sfdominguez}@fi.unju.edu.ar
(2) Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET).
eetarifa@arnet.com.ar

Resumen. En este trabajo se diseña e implementa un controlador tipo MIMO basado en redes neuronales artificiales, aplicado a un modelo de mezclador de corrientes líquidas, configurado sobre el entorno de simulación gráfica de Matlab®. Se destaca una particular metodología de entrenamiento del controlador neuronal, basada en un punto de operación genérico asociado a un reducido entorno del espacio de datos definidos en forma incremental, permitiendo una importante reducción de datos de aprendizaje y esfuerzo computacional. El desempeño del controlador es comparado con otro controlador neuronal similar configurado bajo un proceso de entrenamiento estándar. Los resultados obtenidos permiten apreciar las ventajas del método de control incremental.

Palabras clave. Redes neuronales. Mezclador de flujos. Control. Simulación.

1 Introducción

Los dispositivos mezcladores de flujos son sistemas adicionales indispensables en muchos procesos industriales y son objeto de estudio y aplicaciones en diversas investigaciones [1], [2], [3]. Usualmente se complementan con sistemas de control que requieren correcciones permanentes, que según la complejidad de la dinámica del proceso podrían ser realizadas por un operario experto; o por un sistema de control automático.

En este trabajo se diseña e implementa un controlador tipo MIMO (*Multiple-input Multiple-output*) basado en redes neuronales, aplicado al modelo de un mezclador de corrientes líquidas, configurado sobre el entorno Simulink® de Matlab®. Se destaca una particular metodología de entrenamiento del controlador, basada en un punto de operación genérico que incursiona sobre un reducido entorno definido en forma incremental, permitiendo una importante minimización de datos requeridos para el aprendizaje de la red neuronal. El desempeño del controlador es comparado con otro controlador neuronal similar configurado bajo un proceso de entrenamiento estándar. Los resultados obtenidos permiten apreciar las ventajas del método de control incremental.

2 Proceso a controlar

Los mezcladores de flujos o de caudales (*flow mixers*), utilizados en muchos procesos industriales, suelen estar sometido a dinámicas exigentes, requiriendo la asistencia de sistemas de control automáticos. Un sistema de este tipo, cuyo esquema y modelo gráfico se muestran en la Fig. 1, es utilizado en este trabajo como sistema a controlar.

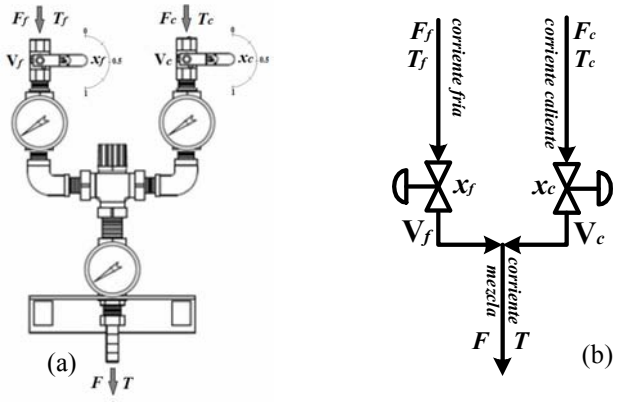

Fig. 1. Mezclador de caudales en línea. (a) esquema físico, (b) modelo.

Las entradas reciben las corrientes para la mezcla con caudales y temperaturas preestablecidos, y en la salida se obtiene una nueva corriente con propiedades específicas (caudal, presión, temperatura, composición).

2.1 Modelo del mezclador en línea

El modelo para el mezclador de flujos en línea se puede representar con el siguiente conjunto de ecuaciones:

$$F = x_{\mathrm{f}} F_{\mathrm{f}} + x_{\mathrm{c}} F_{\mathrm{c}} . \tag{1}$$

$$T = \frac{x_{\mathrm{f}} F_{\mathrm{f}} T_{\mathrm{f}} + x_{\mathrm{c}} F_{\mathrm{c}} T_{\mathrm{c}}}{x_{\mathrm{f}} F_{\mathrm{f}} + x_{\mathrm{c}} F_{\mathrm{c}}} . \tag{2}$$

$$0 \le x_{\mathrm{f}} \le 1 \text{ y } 0 \le x_{\mathrm{c}} \le 1 \tag{3}$$

donde la corriente de entrada (fría), tiene un caudal máximo F_{f} y una temperatura T_{f}. Esta corriente es regulada por la apertura x_{f} de la válvula $\mathrm{V_f}$. La otra corriente de entrada (caliente), tiene un caudal máximo F_{c} y una temperatura T_{c}. Esta corriente es regulada por la apertura x_{c} de la válvula $\mathrm{V_c}$. 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.

2.2 Modelo inverso del mezclador en línea

Invirtiendo el modelo planteado, se obtiene un sistema de control ideal MIMO que proporciona las aperturas de las válvulas (x_f y x_c), para un caudal de referencia (*set-point*) F_{sp} y para una temperatura de referencia T_{sp} preestablecidos. El modelo matemático asociado a este sistema inverso está formado por las ecuaciones siguientes:

$$x_f = \frac{F_{sp}\left(T_c - T_{sp}\right)}{F_f\left(T_c - T_f\right)}. \tag{4}$$

$$x_c = \frac{F_{sp}\left(T_{sp} - T_f\right)}{F_c\left(T_c - T_f\right)}. \tag{5}$$

$$T_f \le T_{sp} \le T_c \text{ y } 0 \le F_{sp} \le F_{max} \tag{6}$$

donde F_{max} es el máximo valor que puede adoptar F_{sp} para un dado T_{sp}. El caudal F_{max} se obtiene a través de un problema de optimización, cuyo resultado se muestra en (7):

$$F_{max} = \begin{cases} \dfrac{F_f\left(T_c - T_f\right)}{T_c - T_{sp}}. & \text{si } \; T_f \le T_{sp} \le \dfrac{F_f T_f + F_c T_c}{F_f + F_c}. \\ \dfrac{F_c\left(T_c - T_f\right)}{T_{sp} - T_f}. & \text{en otro caso} \end{cases} \tag{7}$$

3 Sistemas de control

3.1 RNA como procesadores no lineales

Las redes neuronales artificiales (RNA), son modelos matemáticos que emulan –a nivel básico– la actividad del cerebro humano, dotados de la capacidad de aprender, "memorizar" y generalizar la información aprendida, con elevada tolerancia al ruido. Desde un punto de vista general, las 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 [4], [5]. La Fig. 2 presenta una arquitectura típica de red neuronal feedforward de tres capas, con M neuronas de entrada, L neuronas en la capa oculta y N neuronas de salida.

La primera capa de la RNA –de entrada– se configura con unidades ficticias que distribuyen las señales hacia la capa oculta. La salida de cada neurona de esta capa responde a la siguiente ecuación:

$$y_l = g\left(-w_{0l} + \sum_{m=1}^{M} w_{lm}\, x_m\right) \quad \text{con } l = 1,\ldots,L. \tag{8}$$

donde x_m es la m-ésima componente del patrón de entrada; w_{lm} es el peso de conexión entre la neurona l de la capa oculta y la neurona m de la capa de entrada; w_{0l} es el peso de ajuste (bias) de las neuronas de la capa oculta; $g(\cdot)$ es la función de transferencia de las neuronas ocultas y y_l es la salida de la l-esima neurona oculta. La salida de cada neurona de la última capa se modela con la ecuación (9):

$$z_n = h\left(-v_{0n} + \sum_{l=1}^{L} v_{nl}\ y_l \right) \quad \text{con } n = 1,...,N\,. \tag{9}$$

donde y_l es el valor de salida de las neuronas de la capa anterior; v_{nl} es el peso de conexión entre la neurona n de la capa de salida y la neurona l de la capa oculta; v_{0n} es el peso de ajuste (*bias*) de las neuronas de la capa de salida; $h(\cdot)$ es la función de transferencia de las neuronas de salida y z_n es la n-ésima componente del patrón de salida Z.

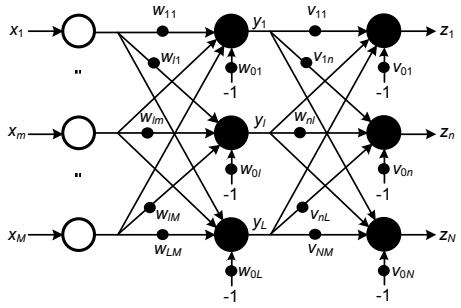

Fig. 2. Red feedforward tricapa genérica.

3.2 Controlador neuronal incremental

Se pueden adoptar dos enfoques generales para la implementación de controladores neuronales. En un primer enfoque, la RNA tiene como entrada tanto las variables controladas como los correspondientes valores de referencia, mientras que las salidas son las variables manipuladas (F, T, F_{sp} y T_{sp} en este caso); mientras que las salidas son las variables de control (x_f y x_c). Si se tiene algún conocimiento previo del sistema –tal como el modelo del proceso, secuencias históricas o muestras de ejemplo con sus correspondientes valores de salida–, se puede utilizar para el entrenamiento de la red. En este caso, la RNA aprenderá a asociar cada patrón de entrada con la salida correspondiente, sobre la totalidad del espacio de datos del sistema, dentro de un nivel de error predeterminado.

Aunque este es un enfoque clásico y ampliamente utilizado, tiene algunos inconvenientes que pueden considerarse críticos. El primer inconveniente surge por la gran cantidad de información que se requiere para cubrir adecuadamente el espacio de datos del sistema, que en muchas ocasiones no están disponibles. Otro inconveniente destacable es la compleja arquitectura que puede alcanzar la RNA, eventualmente con una importante cantidad de unidades neuronales internas, demandando un elevado esfuerzo computacional, que en ciertos casos obliga a aplicar técnicas adicionales para reducir el flujo de información a costa de un incremento en el error de aprendizaje [6].

En un segundo enfoque para implementar una RNA como controlador, se puede considerar la operación de la red bajo un esquema incremental. Para ello, se configura la RNA 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:

$$e_F = F_{sp0} - F \qquad e_T = T_{sp0} - T\,. \tag{10}$$

$$\Delta x_f\left(t\right) = x_f\left(t\right) - x_f\left(t - \Delta t\right) \tag{11}$$

$$\Delta x_c\left(t\right) = x_c\left(t\right) - x_c\left(t - \Delta t\right) \tag{12}$$

donde t es el tiempo de simulación, y Δt es el paso de simulación. 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 descripta 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 este nuevo 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 Fig. 3 se muestra un espacio de datos bidimensional genérico con sus límites, un punto de operación base definido y el entorno asociado con los respectivos incrementos de las variables de control.

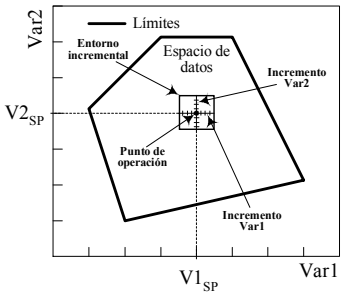

Fig. 3. Punto de operación y entorno incremental para un proceso genérico.

La adecuada implementación de la estrategia de control propuesta requiere algunas consideraciones adicionales. Por un lado está la selección del punto base, cuya ubicación no debe sobrepasar los límites del espacio de datos, y debería ser el punto

establecido por las condiciones nominales de diseño del sistema. Por otro lado, está la determinación de la extensión del entorno alrededor del punto base, que debe ser suficientemente extensa como para capturar la naturaleza de la superficie de control.

Un tercer criterio a considerar es el intervalo de subdivisión del entorno de operación: incrementos muy grandes no permiten capturar las variaciones del entorno de la superficie de control, generando errores considerables en otras ubicaciones del punto base; incrementos muy pequeños producirían gran cantidad de datos innecesarios por estar más allá de la sensibilidad de los componentes del sistema.

4 Desarrollo experimental

El modelo experimental del mezclador de flujos (Fig. 1) se ha implementado inicialmente para operación manual sobre el entorno Simulink® de Matlab® (Fig. 4) y se ha instanciado con los siguientes parámetros:

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

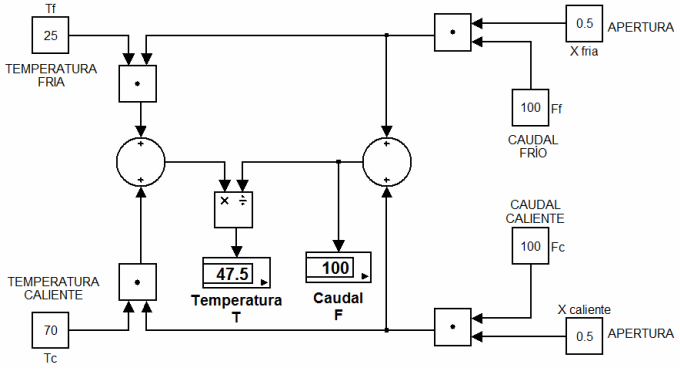

Fig. 4. Modelo Simulink® del mezclador de flujos en línea.

4.1 Generación de datos

Para determinar el punto base de operación se utilizó el modelo directo del mezclador, definido por las ecuaciones (1) a (3), donde las variables de entrada son las aperturas x_f y x_c, y las variables de salida son el caudal F y temperatura T de la mezcla. Evaluando este modelo para el caso en que ambas válvulas están abiertas a la mitad ($x_f = x_c = 0.5$), se obtiene a la salida un caudal $F = 100$ l/min y una temperatura $T = 47.5$ ºC. Este punto fue tomado como base para el proceso de entrenamiento de la RNA.

A continuación se estableció el entorno de operación en ±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:

$$F_{\mathrm{sp}i} = F_{\mathrm{sp}0} + (i-11)\Delta F\, F_{\mathrm{sp}0} \quad \text{con} \quad i = 1,\ldots,21$$
$$F_{\mathrm{sp}}^{\min} = 90 \ \text{l/min} \qquad F_{\mathrm{sp}}^{\max} = 110 \ \text{l/min} \tag{13}$$

$$T_{\mathrm{sp}j} = T_{\mathrm{sp}0} + (j-11)\Delta T\, T_{\mathrm{sp}0} \quad \text{con} \quad j = 1,\ldots,21$$
$$T_{\mathrm{sp}}^{\min} = 42.75\,^{\circ}\mathrm{C} \qquad T_{\mathrm{sp}}^{\max} = 52.25\,^{\circ}\mathrm{C} \tag{14}$$

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

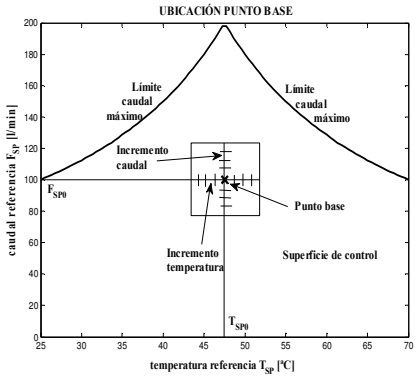

Fig. 5. Posición de punto base y entorno incremental del mezclador de flujos.

Definido el entorno de operación a estudiar, se utilizó el modelo inverso, ecs. (4) a (6), 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. 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:

$$e_{\mathrm{F}i} = F_{\mathrm{sp}0} - F_{\mathrm{sp}i} = -(i-11)\Delta F\, F_{\mathrm{sp}0} \tag{15}$$

$$e_{\mathrm{T}j} = T_{\mathrm{sp}0} - T_{\mathrm{sp}j} = -(j-11)\Delta T\, T_{\mathrm{sp}0}\,. \tag{16}$$

$$\Delta x_{\mathrm{f}ij} = x_{\mathrm{f}ij} - 0.5. \qquad \Delta x_{\mathrm{c}ij} = x_{\mathrm{c}ij} - 0.5. \tag{17}$$

4.2 Diseño del controlador neuronal incremental

A partir de una arquitectura neuronal feedforward, los parámetros más significativos a considerar son el número de capas y las cantidades de neuronas por capa. La dimen-

sionalidad 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 [7], 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, aunque 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 $N^{(o)} = \sqrt{N^{(e)}.N^{(s)}}$ [8]. 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 [9]. Otra regla práctica, utilizada por Goethals *et al*. [10], sugiere que el número de neuronas ocultas ($N^{(o)}$) se relaciona con el número de neuronas de entrada ($N^{(e)}$) de la forma $N^{(o)} = 2 \times N^{(e)} + 1$.

Bajo las consideraciones 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:

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

4.3 Configuración y operación del sistema

El modelo experimental del sistema completo (controlador–planta), fue configurado en el entorno de simulación gráfica de Matlab® (Fig. 6). En este modelo, 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 ($errF = F_{sp} - F$) y el error de temperatura ($errT = T_{sp} - T$), 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 ($x_f(k) = x_f(k-1) + \Delta x_f(k) \parallel x_c(k) = x_c(k-1) + \Delta x_c(k)$) para ser realimentadas al mezclador de flujos, cerrando el lazo de control.

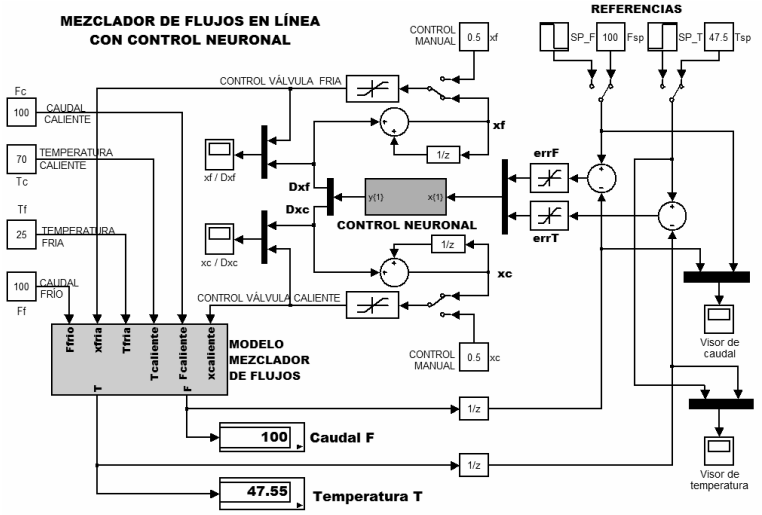

Fig. 6. Modelo experimental del mezclador de flujos con control neuronal.

4.4 Prueba del sistema

Para comprobar la operación del sistema, se aplicaron diferentes condiciones en los parámetros de referencia (*setpoints*). En el primer caso, los parámetros de referencias requirieron variaciones abruptas para el caudal y la temperatura como muestra la Fig. 7.

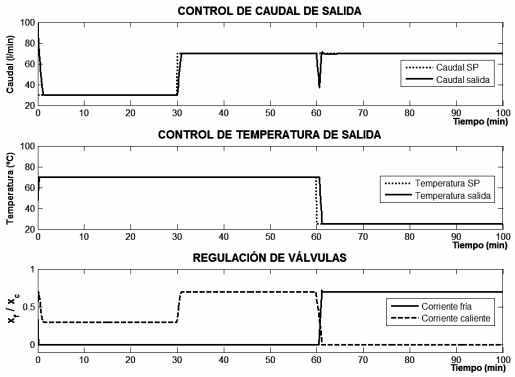

Fig. 7. Control de caudal y temperatura de salida para referencia tipo escalón.

Se observa que el sistema ejecutó una muy buena acción de control para responder a las variaciones abruptas de los valores de referencia de caudal y temperatura.

En el segundo caso, el parámetro de referencia caudal (F_{sp}) exigió una variación sinusoidal suave mientras que la temperatura (T_{sp}) debió mantenerse en un valor constante (Fig. 8). En este caso que la variación exigida por la referencia del caudal (F_{sp}) es correctamente seguida por la planta, mientras la temperatura se mantiene constante en $T = 47.5$ ºC como requiere la referencia T_{sp}.

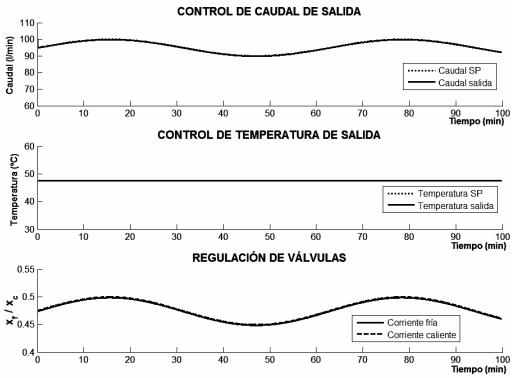

Fig. 8. Control de caudal y temperatura de salida para referencia tipo sinusoidal.

En el tercer caso, se provocó una perturbación sinusoidal del 2% en el parámetro caudal de corriente caliente (F_c), al mismo tiempo que se aplicó una variación lineal tipo rampa a la temperatura de referencia (T_{sp}), que se inició en 25 ºC, obteniéndose los resultados mostrados en la Fig. 9.

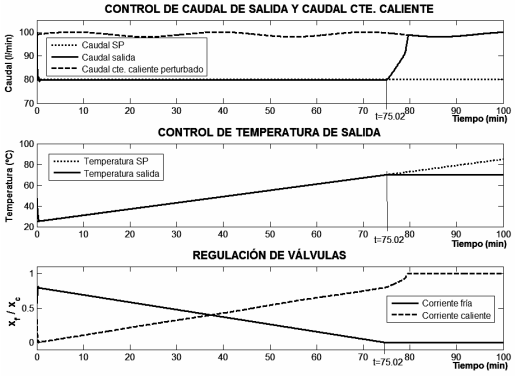

Fig. 9. Control de caudal y temperatura de salida para perturbación en caudal.

En este caso la perturbación fue adecuadamente absorbida por el controlador, hasta el instante $t = 75.02$ min, donde el controlador se satura (la válvula de la corriente fría se cierra totalmente, $x_f = 0$) y la temperatura de salida se estabiliza en $T = 70$ ºC.

El desempeño del controlador neuronal incremental (CN incremental) desarrollado se ha comparado con un controlador neuronal equivalente (CN estándar), con entrenamiento clásico, para la misma planta y con idénticos parámetros [6]. La comparación de los parámetros más representativos se observan en la Tabla 1.

Los datos de esta tabla, evidencian una indiscutible ventaja del controlador neuronal incremental en varios aspectos, tales como una arquitectura más simple, menor cantidad de variables de entrada, gran reducción en la cantidad de datos –sin necesidad de preprocesamiento previo–, menor tiempo de entrenamiento y mejora en el error general de aprendizaje.

Tabla 1. Parámetros de comparación entre controladores equivalentes.

Parámetro	CN incremental	CN estándar
Variables de entrada	2	4
Variables de salida	2	2
Cantidad de capas	3	3
Neuronas ocultas	5	10
Patrones de entrenamiento	2×441	4×5034
ECM de entrenamiento	2.11×10^{-10}	4.83×10^{-9}
Tiempo de entrenamiento	36 seg	1 min 47 seg

5 Conclusiones

Se ha diseñado e implementado en un sistema de simulación gráfica, un controlador neuronal MIMO configurado para trabajar por incrementos a partir de un entorno genérico predefinido, aplicado a un mezclador en línea de corrientes líquidas.

La capacidad del conjunto controlador–planta se ha comprobado experimentalmente bajo diversas condiciones, tales como variaciones abruptas y oscilantes de los parámetros de referencia y perturbaciones de los parámetros de la planta, mostrando un muy buen desempeño del sistema de control neuronal propuesto.

El controlador neuronal incremental se ha comparado con un controlador neuronal estándar equivalente aplicado a la misma planta, poniéndose en evidencia las ventajas en diseño, entrenamiento y operación del modelo incremental.

6 Referencias

[1] Palencia Díaz A. Estudio de Diferentes Estrategias de Control para un Tanque de Mezclado: PID, Control de Matriz Dinámica y Lógica Difusa. Prospect, 8(1), pp. 43-51, (2010)

[2] Mohamed Sultan M., Sha A.S., David C.O.: Controllers optimization for a fluid mixing system using metamodeling approach. In: International Journal of Simulation Modelling, 8(1), pp. 48–59, (2009)

[3] Yan Deng S.: Nonlinear & Linear MIMO Control of an Industrial Mixing Process. Master Thesis, McGill University, Montreal, Canada, (2002)

[4] Galushkin A.I.: Neural Networks Theory. Springer, New York (2007)

[5] He X., Xu S.: Process Neural Networks - Theory and Applications. Springer-Verlag, Berlín (2009)

[6] Martínez S.L., Tarifa E.E., Sánchez Rivero V.D.: Control neuronal tipo MIMO aplicado a un mezclador de corrientes líquidas. En: Investigaciones en Facultades de Ingeniería del NOA, T2, pp. 865-874. Catamarca, Argentina (2011)

[7] Poznyak A.S., Sanchez E.N., Yu W.: Differential Neural Networks for Robust Nonlinear Control. World Scientific Publishing, Singapore (2001)

[8] Blum A.: Neural Networks in C++: An Object-Oriented Framework for Building Connectionist Systems. John Wiley & Sons Inc, New York (1992)

[9] Berry M. J. A., Linoff G. S.: Data Mining Techniques: For Marketing, Sales, and Customer Relationship Management. John Wiley & Sons, New York (2011)

[10] Goethals P.L., Dedecker A.P., Gabriels W., Lek S., De Pauw N.: Applications of artificial neural networks predicting macroinvertebrates in freshwaters. Springer - Aquatic Ecology, V. 41, pp. 491-508 (2007)