

DISEÑO DE SISTEMAS DE CONTROL BORROSO: MODELADO DE LA PLANTA

José M. Andújar, Antonio J. Barragán, Juan M. Córdoba, Iñaki Fernández de Viana

*Dpto. Ingeniería Electrónica, Sistemas Informáticos y Automática, Universidad de Huelva. E.P.S., Ctra. Huelva-La Rábida s/n, 21071. Palos de la Frontera (Huelva)
e-mail: {andujar, antonio.barragan, juanmanuel.cordoba, i.fviana}@diesia.uhu.es*

Resumen: Este trabajo presenta una metodología desarrollada por los autores que permite afrontar el modelado borroso en forma de ecuaciones de estado de un sistema de control multivariable no lineal. La formalización matemática del sistema de control borroso se aborda de forma totalmente general, sin restricciones ni en el vector de estado de la planta ni en el de control. Respecto de las técnicas de identificación de los parámetros del modelo de la planta, se estudia la ya clásica neuroborrosa basada en gradiente descendente y una más novedosa, denominada por los autores como híbrida, basada en la hibridación de la técnica anterior con la heurística inspirada en las colonias de hormigas. *Copyright © 2006 CEA-IFAC.*

Palabras Clave: Identificación, Colonia de hormigas, Modelado, Neuroborroso.

1 INTRODUCCIÓN

Cuando se aborda el diseño de un sistema de control se ha de investigar durante el proceso la estabilidad del sistema. Hasta tal punto los conceptos de control y estabilidad están ligados que no tiene sentido hablar de sistema de control si éste no es estable.

La estabilidad de los sistemas no lineales ha de ser investigada sin realizar aproximaciones lineales de los mismos. Para hacer esto existen varias técnicas basadas en el segundo método de Lyapunov, como por ejemplo el teorema de Krasovskii. Este teorema ofrece, en el caso de sistemas no lineales, una condición suficiente para que el estado de equilibrio sea asintóticamente estable. Para el caso de sistemas lineales la condición es necesaria y suficiente.

Los autores de este trabajo están interesados en la formalización de los sistemas de control borroso (Andújar y Barragán, 2003, 2004d, 2005b; Andújar y Bravo, 2005a; Andújar, *et al.*, 2004a, 2004c), de modo que las herramientas y metodologías de análisis y síntesis de sistemas de control aceptadas por la comunidad científica, sean de aplicación a este tipo de sistemas. En este sentido, la sistemática en desarrollo pretende, al final, sintetizar sistemas de

control borroso sin restricciones y que sean estables por diseño. Para ello, el diseño del controlador se aborda mediante dos técnicas bien diferentes, una basada en el teorema de Krasovskii y otra de aplicación directa del segundo método de Lyapunov, utilizando como herramienta la aritmética intervalar. Ambos procedimientos exigen los dos primeros pasos siguientes: (i) modelización de la planta en forma de ecuación de estado y (ii) identificación de los parámetros del modelo de la planta. En este trabajo se resuelven estos dos pasos.

La metodología que se propone requiere un arduo trabajo matemático para formalizar tanto el modelo borroso de la planta y del controlador, como el modelo borroso en lazo cerrado en forma de ecuaciones de estado. Sin embargo, la ventaja que se obtiene es que al final del proceso de modelado, se dispone de un modelo de estado parametrizado del sistema de control borroso en lazo cerrado. A partir de aquí, el proceso de identificación de los parámetros del controlador puede integrarse de manera formal en diferentes técnicas de diseño de sistemas de control no lineal.

El artículo está organizado del modo siguiente: en la sección 2 se aborda la forma de obtener la ecuación

de estado de un sistema de control borroso en lazo cerrado sin restricciones. En esta sección se propone un modelo en forma de ecuación de estado compacta, lo cual facilita su uso tanto en desarrollos teóricos como en algoritmos de programación. En la sección 3 se estudia la forma de identificar los parámetros correspondientes al modelo de la planta, lo cual se hace en base al modelo TSK mediante datos de entrada/salida. En esta sección se analiza la técnica que están desarrollando los autores y que denominan híbrida: basada en las técnicas neuroborrosas ya clásicas y en algoritmos de colonias de hormigas. Finalmente, en la sección 4 se aportan algunas conclusiones y líneas de investigación en curso.

2 ECUACIÓN DE ESTADO DE UN SISTEMA DE CONTROL BORROSO MULTIVARIABLE NO LINEAL

El primer paso para abordar el control de un sistema es conocer su dinámica. La propuesta que haremos aquí permite encontrar la ecuación de estado de un sistema de control borroso en el que la planta se modela en base a datos de entrada/salida. Esta forma de abordar el problema permite enfrentarse a sistemas sensiblemente no lineales y cuyo modelado por técnicas matemáticas usuales resulta muy complicado.

Considérese pues una planta multidimensional descrita por:

$$\begin{aligned}\dot{x}_1 &= f_1(x_1, x_2, \dots, x_n; u_1, u_2, \dots, u_m) \\ \dot{x}_2 &= f_2(x_1, x_2, \dots, x_n; u_1, u_2, \dots, u_m) \\ &\vdots \\ \dot{x}_n &= f_n(x_1, x_2, \dots, x_n; u_1, u_2, \dots, u_m)\end{aligned}\quad (1)$$

donde el controlador multivariable se define como:

$$\begin{aligned}u_1 &= \phi_1(x_1, x_2, \dots, x_n); \\ u_2 &= \phi_2(x_1, x_2, \dots, x_n); \\ &\vdots \\ u_m &= \phi_m(x_1, x_2, \dots, x_n)\end{aligned}\quad (2)$$

Las expresiones (1) y (2) representan un modelo totalmente general tanto para la planta como para el controlador, sin ningún tipo de restricciones, por lo que pueden ser no lineales. El sistema puede ser escrito de forma simplificada como en (3).

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}) \quad (3)$$

siendo $\dot{\mathbf{x}} = (\dot{x}_1, \dot{x}_2, \dots, \dot{x}_n)^T$, $\mathbf{x} = (x_1, x_2, \dots, x_n)^T$ el vector de estado y $\mathbf{u} = (u_1, u_2, \dots, u_m)^T$ el vector de control.

Un modelo borroso equivalente múltiple-entrada-múltiple-salida (*Multiple – Input – Multiple – Output*, MIMO) del proceso puede ser representado mediante el siguiente conjunto de reglas (Hung, *et al.*, 1995; Takagi y Sugeno, 1985):

$$\begin{aligned}R^{(l)}: & \text{SI } x_1 \text{ es } A_{1i}^l \text{ y } x_2 \text{ es } A_{2i}^l, \dots, \text{ y } x_n \text{ es } A_{ni}^l \text{ y} \\ & u_1 \text{ es } B_{1i}^l \text{ y } u_2 \text{ es } B_{2i}^l, \dots, \text{ y } u_m \text{ es } B_{mi}^l \\ & \text{ENTONCES } \dot{x}_i \text{ es } g_i^l(\mathbf{x}, \mathbf{u}; \boldsymbol{\theta}_i^l)\end{aligned}\quad (4)$$

Donde $l = 1 \dots M$ es el número de reglas del modelo borroso de la planta; A_{ki}^l el conjunto borroso definido en el universo de discurso de la variable de estado x_k , $k = 1 \dots n$, y la i -ésima ecuación diferencial del proceso. B_{ji}^l es el conjunto borroso definido en el universo de discurso de la j -ésima señal de control u_j , $j = 1 \dots m$, y la i -ésima ecuación diferencial del proceso.

Considerando en la expresión (4) que el consecuente es TSK con término afín (Johansson *et al.* 1999):

$$\begin{aligned}g_i^l(\mathbf{x}, \mathbf{u}; \boldsymbol{\theta}_i^l) &= a_{0i}^l + a_{1i}^l x_1 + \dots + a_{ni}^l x_n + \\ &+ b_{1i}^l u_1 + b_{2i}^l u_2 + \dots + b_{mi}^l u_m\end{aligned}\quad (5)$$

Donde a_{ki}^l y b_{ji}^l representan, respectivamente, el coeficiente constante para la variable de estado x_k y para la j -ésima señal de control de la ecuación diferencial i que representa el proceso. El vector $\boldsymbol{\theta}_i^l$ representa el conjunto de parámetros adaptables:

$$\boldsymbol{\theta}_i^l = (a_{0i}^l, a_{1i}^l, \dots, a_{ni}^l; b_{1i}^l, b_{2i}^l, \dots, b_{mi}^l)^T \quad (6)$$

Considerando un sistema borroso TSK con desborrosificador centro-promedio y método de inferencia producto, la expresión (3) puede ser escrita como (Wang, 1997):

$$\dot{x}_i = \frac{\sum_{l=1}^M w_i^l g_i^l(\mathbf{x}, \mathbf{u}; \boldsymbol{\theta}_i^l)}{\sum_{l=1}^M w_i^l} \quad (7)$$

Donde w_i^l es el grado de cumplimiento de la regla l :

$$w_i^l = \prod_{k=1}^n \mu_{F_k}^l(x_k, \boldsymbol{\theta}_i^l) \prod_{j=1}^m \mu_{F_j}^l(u_j, \boldsymbol{\theta}_i^l), \quad (8)$$

siendo $\mu_{F_k}^l(x_k, \boldsymbol{\theta}_i^l)$ y $\mu_{F_j}^l(u_j, \boldsymbol{\theta}_i^l)$ las funciones de pertenencia definidas en sus correspondientes universos de discurso X y U para el conjunto borroso $F \in (X \cup U)$ representado por un conjunto de pares ordenados $\{(x_k, \mu_{F_k}^l(x_k, \boldsymbol{\theta}_i^l)) : x_k \in X\}$ y $\{(u_j, \mu_{F_j}^l(u_j, \boldsymbol{\theta}_i^l)) : u_j \in U\}$.

Si el consecuente de la regla tiene la forma definida en (5), entonces se puede escribir (7) como en (9) o (10):

$$\dot{x}_i = \frac{1}{\sum_{l=1}^M w_l^i} \left[\sum_{l=1}^M w_l^i a_{0i}^l + \dots + \sum_{l=1}^M w_l^i a_{ni}^l x_n + \sum_{l=1}^M w_l^i b_{1i}^l u_1 + \dots + \sum_{l=1}^M w_l^i b_{mi}^l u_m \right] \quad (9)$$

$$\begin{aligned} \dot{x}_i &= a_{0i}^i + a_{1i}^i x_1 + a_{2i}^i x_2 + \dots + a_{ni}^i x_n + \\ &+ b_{1i}^i u_1 + b_{2i}^i u_2 + \dots + b_{mi}^i u_m = \\ &= a_{0i}^i + \sum_{k=1}^n a_{ki}^i x_k + \sum_{j=1}^m b_{ji}^i u_j \end{aligned} \quad (10)$$

Donde a_{0i}^i , a_{ki}^i y b_{ji}^i son coeficientes variables (Wong, *et al.*, 1997) definidos por las expresiones siguientes:

$$a_{0i}^i = \frac{\sum_{l=1}^M w_l^i a_{0i}^l}{\sum_{l=1}^M w_l^i}, \quad a_{ki}^i = \frac{\sum_{l=1}^M w_l^i a_{ki}^l}{\sum_{l=1}^M w_l^i}; \quad k = 1, \dots, n \quad (11)$$

$$b_{ji}^i = \frac{\sum_{l=1}^M w_l^i b_{ji}^l}{\sum_{l=1}^M w_l^i}; \quad j = 1, \dots, m \quad (12)$$

Con objeto de simplificar la ecuación (10) proponemos extender el vector de estados en una coordenada, de manera que éste adopte la forma:

$$\tilde{\mathbf{x}} = \begin{pmatrix} 1 \\ \mathbf{x} \end{pmatrix} = (x_0, x_1, \dots, x_n)^T; \quad x_0 = 1 \quad (13)$$

Al sustituir (13) en (10), se obtiene la siguiente expresión más compacta:

$$\dot{x}_i = \sum_{k=0}^n a_{ki}^i \tilde{x}_k + \sum_{j=1}^m b_{ji}^i u_j \quad (14)$$

Obtenida la ecuación de estado de la planta que representa su modelo borroso equivalente, se procederá de forma similar con el controlador. Para ello se considera que su diseño es completamente independiente del proceso de identificación del modelo de la planta, de forma que la partición del universo de discurso de los estados para la planta y controlador no tienen porqué coincidir. Por lo mismo, sus modelos borrosos no tienen porqué tener el mismo número de reglas. En consecuencia, el controlador borroso puede ser representado por el siguiente conjunto de reglas:

$$\begin{aligned} R^{(r)}: & \text{SI } x_1 \text{ es } C_{1j}^r \text{ y } x_2 \text{ es } C_{2j}^r, \dots, \text{ y } x_n \text{ es } C_{nj}^r \\ & \text{ENTONCES } u_j^r = c_{0j}^r + c_{1j}^r x_1 + \dots + c_{nj}^r x_n \end{aligned} \quad (15)$$

Donde C_{kj}^r representa el conjunto borroso de las $k = 1 \dots n$ variables de estado, con $j = 1 \dots m$ acciones de control y $r = 1 \dots N$ reglas.

Procediendo de forma similar a la mostrada en (7), (8), (9) y (10), las señales de control pueden expresarse por:

$$u_j = \frac{1}{\sum_{r=1}^N \omega_j^r} \left[\sum_{r=1}^N \omega_j^r c_{0j}^r + \dots + \sum_{r=1}^N \omega_j^r c_{nj}^r x_n \right] \quad (16)$$

o de forma más simplificada:

$$u_j = c_{0j}^i + c_{1j}^i x_1 + c_{2j}^i x_2 + \dots + c_{nj}^i x_n = c_{0j}^i + \sum_{k=1}^n c_{kj}^i x_k \quad (17)$$

donde los coeficientes variables vienen determinados por:

$$c_{0j}^i = \frac{\sum_{r=1}^N \omega_j^r c_{0j}^r}{\sum_{r=1}^N \omega_j^r}; \quad c_{kj}^i = \frac{\sum_{r=1}^N \omega_j^r c_{kj}^r}{\sum_{r=1}^N \omega_j^r}; \quad k = 1, \dots, n \quad (18)$$

y el grado de activación de la regla r del controlador por:

$$\omega_j^r = \prod_{k=1}^n \mu_{F_k}^r(x_k, \varphi_j^r). \quad (19)$$

Siendo φ_j^r el vector de parámetros adaptables del controlador:

$$\varphi_j^r = (c_{0j}^r, c_{1j}^r, \dots, c_{nj}^r)^T \quad (20)$$

Sustituyendo la expresión (17) en (10), se obtiene la ecuación de estado del sistema no lineal en lazo cerrado:

$$\begin{aligned} \dot{x}_i &= a_{0i}^i + \sum_{j=1}^m b_{ji}^i c_{0j}^i + \sum_{k=1}^n \left(a_{ki}^i + \sum_{j=1}^m b_{ji}^i c_{kj}^i \right) x_k \\ & i, k = 1, \dots, n \text{ y } j = 1, \dots, m \end{aligned} \quad (21)$$

Aplicando la extensión del vector de estados a (21), se obtiene la siguiente expresión más compacta:

$$\begin{aligned} \dot{x}_i &= \sum_{k=0}^n \left(a_{ki}^i + \sum_{j=1}^m b_{ji}^i c_{kj}^i \right) \tilde{x}_k \\ & k = 0, 1, \dots, n, i = 1, \dots, n \text{ y } j = 1, \dots, m \end{aligned} \quad (22)$$

La expresión (22) es un modelo en forma de ecuaciones de estado de un sistema de control borroso multivariable no lineal.

3 IDENTIFICACIÓN DE LA PLANTA

El problema de la obtención del modelo borroso de una planta consiste, fundamentalmente, en el ajuste de los parámetros característicos del mismo, de modo que represente, de la forma más fiel posible, la dinámica de ésta. En la bibliografía se han propuesto diferentes metodologías, aunque las más ampliamente utilizadas han sido las basadas en el gradiente

descendente (Jang y Sun, 1995) y, en menor medida y más recientemente, en algoritmos evolutivos (Hoffmann, 1998). Independiente de las metodologías propuestas, el modelo obtenido suele depender de la estimación inicial de sus parámetros. Para esta estimación inicial, se han propuesto en la literatura varias técnicas, como la de prueba y error (Jang, et al., 1997), *clustering* (Chiu, 1994), o la simple asignación aleatoria (Yen y Langari, 1999).

Los autores han estado trabajando con la metodología neuroborrosa basada en gradiente descendente (Andújar y Bravo, 2005a), y más recientemente con la hibridación de algoritmos OCH (Optimización basada en Colonias de Hormigas) y de gradiente descendente (Andújar, et al., 2004b).

3.1 Metodología neuroborrosa

El primer algoritmo que proponemos para identificar la planta es neuroborroso basado en gradiente: una forma de aprovechar la habilidad de las redes neuronales artificiales para construir modelos borrosos. En particular, aplicaremos a la estimación de los parámetros del modelo de la planta, uno de los más importantes algoritmos de aprendizaje de las redes neuronales: *BackproPagation* (BP).

En una red neuronal (Yen y Langari, 1999), BP realiza dos pasadas a través de las diferentes capas de la red: una hacia delante y otra hacia atrás. En la primera, se aplica un vector de entrada a los nodos de entrada de la red, propagándose su efecto capa a capa a través de la misma. Después de la última capa se obtiene, como consecuencia del vector de entrada introducido, un vector respuesta. Durante la primera pasada (hacia delante) a través de la red, los pesos sinápticos permanecen fijos. Por el contrario, durante la segunda pasada (hacia atrás), todos los pesos sinápticos se ajustan en función de una regla de corrección del error (una generalización de la regla delta del algoritmo de aprendizaje de Widrow y Hoff, 1960). Por señal de error se entiende la diferencia entre la respuesta de salida de la red y la deseada. La señal de error resultante se propaga entonces hacia atrás, en sentido contrario a las conexiones sinápticas. Este método de ajuste de pesos se repite hasta que ya no se producen cambios en los mismos, lo cual se conoce como convergencia del algoritmo de aprendizaje. En este punto, el proceso de aprendizaje ha finalizado y la respuesta real de la red es lo más parecida posible a la deseada.

En el caso de un modelo borroso, la aplicación del algoritmo anterior funciona en la pasada hacia delante de forma análoga a la red neuronal, sólo que ahora, en vez de permanecer fijos los pesos sinápticos, se mantienen sin cambio los parámetros del antecedente y el consecuente de las reglas. En la pasada hacia atrás, la señal de error resultante de la diferencia entre la respuesta real del modelo y la deseada, se propaga para ajustar los parámetros θ según la regla de corrección del error. Para obtener el

modelo borroso se define la siguiente función de coste (error cuadrático medio):

$$J(k) = \frac{1}{2} (y(k) - \hat{y}(k))^2 \quad (23)$$

Donde $y(k)$ es la salida real en la k -ésima iteración e $\hat{y}(k)$ la salida estimada.

La aplicación del método del gradiente descendente para ajustar los parámetros θ (Freman y Skapura, 1991) proporciona la expresión siguiente:

$$\theta(k+1) = \theta(k) - \eta \left. \frac{\partial J(k)}{\partial \theta(k)} \right|_{\theta=\theta(k)} \quad (24)$$

Usando funciones de pertenencia Gaussianas en el antecedente de las reglas y consecuente TSK con término afín en el consecuente, se obtienen mediante la ecuación (24) las siguientes expresiones de adaptación de parámetros:

Centros de las funciones de pertenencia Gaussianas:

$$\gamma_i^l(k+1) = \gamma_i^l(k) - 2\eta e(k) \frac{\prod_{i=1}^n \mu_{F_i}^l(x_i(k), \theta_i^l(k)) (x_i(k) - \gamma_i^l(k))}{\sum_{l=1}^M \left[\prod_{i=1}^n \mu_{F_i}^l(x_i(k), \theta_i^l(k)) \right] (\beta_i^l(k))^2} \quad (25)$$

Anchos de las funciones de pertenencia Gaussianas:

$$\beta_i^l(k+1) = \beta_i^l(k) - 2\eta e(k) \frac{\prod_{i=1}^n \mu_{F_i}^l(x_i(k), \theta_i^l(k)) (x_i(k) - \gamma_i^l(k))^2}{\sum_{l=1}^M \left[\prod_{i=1}^n \mu_{F_i}^l(x_i(k), \theta_i^l(k)) \right] (\beta_i^l(k))^3} \quad (26)$$

Donde η el factor de aprendizaje,

$$e(k) = y(k) - \hat{y}(k) \quad (27)$$

la señal de error, y

$$\mu_{F_i}^l(x_i(k), \theta_i^l(k)) = \exp \left[-\frac{(x_i(k) - \gamma_i^l(k))^2}{(\beta_i^l(k))^2} \right] \quad (28)$$

El término afín del consecuente se ajusta mediante:

$$a_0^l(k+1) = a_0^l(k) - \eta e(k) \frac{\prod_{i=1}^n \mu_{F_i}^l(x_i(k), \theta_i^l(k))}{\sum_{l=1}^M \left[\prod_{i=1}^n \mu_{F_i}^l(x_i(k), \theta_i^l(k)) \right]} \quad (29)$$

y el resto de términos del consecuente TSK con:

$$a_i'(k+1) = a_i'(k) - \eta e(k) \frac{\prod_{i=1}^n \mu_{F_i}'(x_i(k), \theta_i'(k)) x_i(k)}{\sum_{l=1}^M \left[\prod_{i=1}^n \mu_{F_i}'(x_i(k), \theta_i'(k)) \right]} \quad (30)$$

Para mejorar la eficiencia del proceso de identificación, el factor de aprendizaje puede hacerse adaptativo (Moscinsky y Ogonowski, 1995):

$$\eta(k+1) = \begin{cases} inc_ \eta \times \eta(k) & J(k) \leq J(k-1) \\ dec_ \eta \times \eta(k) & J(k) \geq r \times J(k-1) \\ \eta(k) & J(k-1) \leq J(k) < r \times J(k-1) \end{cases} \quad (31)$$

Donde r e $inc_ \eta$ y $dec_ \eta$ son números ajustables, definidos respectivamente como *margen de tolerancia* y *momentos*.

El algoritmo 1 implementa la ley de adaptación de parámetros descrita. En él, “m” es el número de reglas, “n” el número de entradas, “ γ ” los centros de las funciones de pertenencia Gaussianas, “ β ” sus anchos, “ e ” el error, “MSE” la función de coste (Error cuadrático medio, *Medium Square Error*), “ η ” el factor de aprendizaje, “x” las entradas, “y” la salida, “y’” el consecuente, “r” el margen de tolerancia, y “ ε ” el límite de error.

```

 $\gamma, \beta, y', e = bp(\eta, m, n, \gamma, \beta, x, y, r, \varepsilon)$ 
for each epoch
  for each training point
    y'(k) = Compute_Fx
    e(k) = y(k) - y'(k)
     $\gamma$  = Update_Centers
     $\beta$  = Update_Widths
    y' = Update_Consequents
  endfor
  error(epoch) = MSE(e)
  if (error(epoch) <  $\varepsilon$ ) Or (Stable(error(epoch)))
    return ( $\gamma, \beta, y', error$ )
  endif
  if error(epoch) < error(epoch-1)
    Inc( $\eta, r$ )
  endif
  if error(epoch) > error(epoch-1)
    Dec( $\eta, r$ )
  endif
endfor
return ( $\gamma, \beta, y', error$ )
end
    
```

Algoritmo 1

La adaptación del factor de aprendizaje se realiza en las funciones “Inc(η, r)” y “Dec(η, r)” de acuerdo con la expresión (31). La función “Fx” implementa el consecuente TSK con término afín u otro.

En Andújar y Bravo (2005a) se muestra un caso práctico de modelado (una planta de tratamiento de aguas residuales de una fábrica de celulosas) usando la técnica descrita.

3.2 Metodología híbrida: neuroborrosa mediante colonia de hormigas

Recientemente, las *Heurísticas basadas en la naturaleza o Algoritmos bioinspirados* (Bentley, 2001; Bonabeon, *et al.*, 1999) han sido uno de los más prometedores campos de investigación en el desarrollo de metaheurísticas. Dentro de estas metaheurísticas (Blum y Roli, 2001) se encuentran los Algoritmos Genéticos, Enfriamiento simulado y Algoritmos de optimización basados en Colonias de Hormigas (OCH), entre otros.

La metaheurística OCH (Dorigo y Di Caro, 1999) se basa en simular la forma en que las hormigas llevan la comida hasta el hormiguero recorriendo un camino de longitud mínima. En el proceso de búsqueda de comida, las hormigas van depositando una sustancia denominada *feromona*, la cual son capaces de oler y dirigir su búsqueda y, por tanto, la de la colonia, en función de las deposiciones de feromona. Cuando una hormiga llega a una intersección, decide el camino a seguir de un modo probabilístico en función de la feromona depositada en cada ramificación. Este movimiento continuado provoca que los caminos más cortos sean más frecuentados, por lo que reciben mayor cantidad de feromona, mientras que los más largos serán progresivamente abandonados y la feromona asociada acabará por evaporarse. De este modo, se acaba formando un camino de longitud mínima entre el hormiguero y la comida. Dependiendo del problema, esta longitud vendrá definida por una función de coste a minimizar.

Para poder aplicar este tipo de algoritmos, es necesario que una solución a un problema se pueda representar como un camino en un grafo. Este grafo lo representaremos como $G = (C, L)$, donde C es un conjunto de nodos y L es un conjunto de arcos que conectan los nodos de C .

Para poder aplicar los algoritmos OCH al modelado borroso TSK se ha diseñado un grafo G donde el conjunto de nodos C se ha organizado por capas. Esto se muestra en la Figura 1.

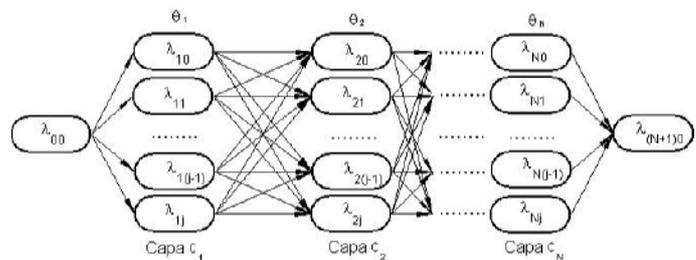


Figura 1. Grafo para el ajuste de los parámetros del modelo TSK.

Cada capa de la Figura 1 se corresponde con un parámetro del modelo TSK a ajustar. En cada capa, los nodos representan valores que puede tomar el parámetro que se corresponde con dicha capa. Cada nodo de una capa está unido a todos los nodos de la siguiente mediante arcos. Así, el ajuste de los

parámetros del modelo TSK puede ser interpretado como el recorrido de una hormiga por el grafo G guiada por la feromona, donde cada nodo recorrido indica la elección de un valor para cada uno de los parámetros del modelo TSK.

El concepto del procedimiento de modelización desarrollado está basado en la aplicación, en una primera fase, del algoritmo de OCH para generar prometedoras soluciones iniciales, para a continuación aplicar una búsqueda local basada en el conocido algoritmo de gradiente descendente. El algoritmo 2 implementa la estructura de modelización híbrida descrita

El primer paso comprende la inicialización de los parámetros considerados por el algoritmo. Entre otros, se tendrán que especificar los valores de la cantidad inicial de feromona asociada a cada transición (valor positivo pequeño que debe ser igual para todas ellas para que, en un primer momento, todas presenten la misma “deseabilidad”), el número de hormigas considerado, m , y los pesos que ponderan la importancia de la información heurística y memorística en la regla probabilística de transición. En el caso del problema del ajuste de los parámetros del modelo TSK no se cuenta con información heurística.

El procedimiento principal de la Metaheurística_ACO gestiona, mediante el constructor `Planificacion_de_actividades`, la planificación de las dos componentes: (i) la generación y activación de las hormigas artificiales, (ii) la evaporación de feromona.

Mientras que en la aplicación a problemas no distribuidos se suele considerar una planificación estrictamente secuencial, en el caso de problemas distribuidos puede ser fácil y eficientemente explotado algún tipo de paralelismo. En el caso del ajuste del modelo TSK se puede mejorar esta eficiencia paralelizando el algoritmo. En nuestro caso, la feromona que se deposita en los arcos depende de la función de coste. La función de coste que hemos utilizado es el error cuadrático medio entre la salida real del modelo y la deseada. Una vez calculada, esta feromona determinará la probabilidad de transición en los arcos.

El procedimiento `actualizar_memoria_de_la_hormiga` supone situarla en un estado inicial e incluir la componente en la memoria de la hormiga L . En nuestro caso suponemos que existe un nodo inicial único, que no se corresponde con ningún valor válido para el modelo TSK a ajustar, pero que facilita el recorrido de la hormiga.

Por último, comentaremos también que los procedimientos `calcular_las_probabilidades_de_transición` y `aplicar_regla_probabilística_de_transición`, consideran el estado actual de la hormiga, los valores actuales de la tabla local de encaminamiento existentes en dicho nodo, y las restricciones del problema Ω para

establecer el proceso de transición probabilística a otro estado factible.

```

1 Procedimiento Metaheurística_ACO
2   Inicializacion_de_parametros
3   Mientras (no se de el criterio de parada)
4     Planificacion_de_actividades
5     generacion_y_activacion_de_hormigas
6     evaporacion_de_feromona
7     Fin Planificación_de_actividades
8   Fin Mientras
10  Aplicar_gradiente_descendente_a_mejor_hormiga
11 Fin Procedimiento

```

```

1 Procedimiento generacion_y_activacion_de_hormigas
2   Repetir en Paralelo para k=1 hasta Numero_de_Hormigas
3     nueva_hormiga(k)
4   Fin Procedimiento

```

```

1 Procedimiento nueva_hormiga(id_hormiga)
2   Inicializar_hormiga(id_hormiga)
3   L = actualizar_memoria_de_la_hormiga
4   Mientras (estado_actual ≠ estado_objetivo)
5     B = leer_tabla_local_de_encaminamiento
6     P = calcular_las_probabilidades_de_transicion(A,L,Ω)
7     estado_siguiete=regla_probabilística_de_transicion(P,Ω)
8     moverse_al_estado_siguiete(estado_siguiete)
9   Fin Mientras
8   Evaluar_hormiga
9   Para cada arco recorrido hacer
10    depositar_feromona_en_el_arco_recorrido
11    actualizar_tabla_local_de_encaminamiento
12  Fin Para
13  liberar_recursos_de_la_hormiga(id_hormiga)
14 Fin Procedimiento

```

Algoritmo 2

En Andújar, *et al.* (2004b) se muestran varios casos prácticos que demuestran que los resultados obtenidos mediante esta metodología mejoran los resultantes con el método tradicional del gradiente descendente.

4 CONCLUSIONES

Es de sobra conocido que cuanto mejor sea el modelo obtenido de la planta, más factible será su control, ya que su dinámica puede ser prevista y, por tanto, controlados sus estados.

En este trabajo se ha presentado un modelo matemático equivalente de un sistema de control borroso multivariable no lineal, el cual, en base a datos de entrada/salida, permite obtener la ecuación de estado de la planta. Esta ecuación constituye el paso previo para controlar un sistema mediante los métodos que le sean de aplicación. A continuación, una vez diseñado el controlador con los desempeños que se establezcan, la ecuación (22) sintetiza de forma simple y concisa la ecuación del sistema de control en lazo cerrado.

La segunda parte del artículo ha estado dedicada al proceso de identificación de la planta, para el que se han presentado dos metodologías: (i) la neuroborrosa ya clásica, o ajuste de parámetros basado en el algoritmo BP según la regla de corrección del error basada en el gradiente descendente; y (ii) una nueva propuesta que realiza una primera selección, utilizando algoritmos basados en colonias de

hormigas, para a continuación realizar la búsqueda, ya local, mediante el procedimiento clásico.

La gran ventaja que ofrece la utilización de colonias de hormigas, es que permitimos tener varias posibles soluciones activas en diferentes puntos del espacio de búsqueda. Esta simultaneidad de posibles soluciones nos permite explorar varios caminos hacia el mínimo global de la función de coste. La sola utilización de un único camino de búsqueda basado en el gradiente por ejemplo, puede llevar la búsqueda a un mínimo local.

Queremos resaltar que tanto el algoritmo que implementa el método del gradiente descendente como el híbrido: colonia de hormigas + gradiente descendente, proporcionan, junto a la metodología teórica desarrollada, la ecuación de estado de la planta de forma directa, lo cual supone un ahorro importante en el esfuerzo realizado por el Ingeniero de Control.

Si bien esta técnica de modelado que estamos desarrollando está aún en fase preliminar, los resultados obtenidos son francamente esperanzadores y esperamos que en próximos trabajos, con algoritmos ya más depurados, los resultados sean aún mejores.

Este trabajo es un paso más dentro de un objetivo mucho más amplio que constituye una de las líneas de investigación de sus autores: la síntesis de sistemas de control borroso estables por diseño.

REFERENCIAS

- Andújar, J. M. y A. J. Barragán (2003). Matriz Jacobiana de un Sistema de Control Borroso en Lazo Cerrado. Aplicación a la Síntesis de Sistemas de Control Borroso. *Proc. XXIV Jornadas de Automática*. León, España.
- Andújar, J. M., J. M. Bravo and A. Peregrín (2004a). Stability analysis and synthesis of multivariable fuzzy systems using interval arithmetic. *Fuzzy Sets and Systems*, **148(3)**
- Andújar, J. M., J. M. Córdoba y I. Fernández (2004b). Una técnica híbrida de modelización neuroborrosa mediante colonia de hormigas. *Proc. XXV Jornadas de Automática*. Ciudad Real, España. <http://www.isa-cr.uclm.es/xxvjornadas/web/contenido/listatrabajos.html>
- Andújar, J. M., A. J. Barragán, M. E. Gegúndez y F. J. de Toro (2004c). Aplicación de la Matriz Jacobiana de un sistema de control borroso a la obtención de sus estados de equilibrio. *Proc. XXV Jornadas de Automática*. Ciudad Real, España.
- Andújar, J. M. y A. J. Barragán (2004d). Algoritmo para el Cálculo de la Matriz Jacobiana de un Sistema de Control Borroso en Lazo Cerrado. *Proc. ESTYLF'04*. Jaén, España.
- Andújar, J. M. and J. M. Bravo (2005a). Multivariable fuzzy control applied to the physical-chemical treatment facility of a cellulose factory. *Fuzzy Sets and Systems*, **150(3)**.
- Andújar, J. M. and A. J. Barragán (2005b). A methodology to design stable nonlinear fuzzy control systems. *Fuzzy Sets and Systems*, **154(2)**.
- Bentley, P. J. (2001). *Digital Biology*. Headline.
- Blum, C. and A. Roli (2001). Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison. *Technical report 2001-13 IRIDIA*. <http://iridia.ulb.ac.be/~mdorigo/ACO/ACO.html>
- Bonabeon, E., M. Dorigo and G. Theraulaz (1999). *Swarm Intelligence From Natural to Artificial Systems*. Oxford University Press.
- Chiu, S. L. (1994). Fuzzy model identification based on cluster estimation. *Journal of Intelligent and Fuzzy Systems*, **2(3)**.
- Dorigo, M. and G. Di Caro (1999). *The Ant Colony Optimization Meta-Heuristic*. D. Corne, M. Dorigo and F. Glover (eds.), *New Ideas in Optimization*, 11-32, McGraw-Hill.
- Freeman, J. and D. Skapura (1991). *Neural Networks. Algorithms, Applications and Programming Techniques*. Addison-Wesley.
- Hoffmann, F. (1998). Incremental tuning of fuzzy controllers by means of an evolution strategy. *GP'98 Conf.*, Madison, Wisconsin.
- Hung, T., M. Sugeno, R. Tong and R. R. Yager (1995). *Theoretical aspects of fuzzy control*, John Wiley and Sons Inc.
- Jang, J.-S. R. and C.-T. Sun. (1995). Neuro-fuzzy modeling and control. *Proc. of the IEEE* **83(3)**, 378-406.
- Jang, J.-S. R., C.-T. Sun and E. Mizutani (1997). *Neurofuzzy and soft computing: a computational approach to learning and machine intelligence*. Prentice-Hall.
- Johansson, M., Rantzer, A and Arzen, K. E.. Piecewise quadratic stability for fuzzy systems. *IEEE Transactions on Fuzzy Systems* **7(6)**, 713-722.
- Moscinsky, J. and Z. Ogonowski (1995). *Advanced Control with MATLAB and Simulink*. Prentice-Hall.
- Takagi T. and M. Sugeno (1985). Fuzzy identification of systems and its applications to modeling and control. *IEEE Transaction on Systems, Man and Cybernetics* **15**, 116-132.
- Wang L. X. (1997). *A Course in Fuzzy Systems and Control*. Prentice-Hall, New Jersey, USA.
- Widrow, B. and M. E. Hoff (1960). Adaptive switching circuits. *IRE WESCON Conventional Record*, 96-104.
- Wong L., F. Leung and P. Tam (1997). Stability design of TS model based fuzzy systems. *Proc. Six IEEE international conference on fuzzy systems*, Barcelona, Spain, 83-86.
- Yen, J. and R. Langari (1999). *Fuzzy Logic. Intelligence, control and information*. Prentice-Hall.