

## Método de Agrupamiento en Línea para la Identificación de Modelos Borrosos Takagi-Sugeno

Boris Martínez\* Francisco Herrera\* Jesús Fernández\*  
Erick Marichal\*\*

\* *Dpto. de Automática y Sistemas Computacionales Universidad Central de Las Villas (UCLV); C. Camajuaní Km. 5.5; Santa Clara, Cuba.  
(e-mail: boris@uclv.edu.cu, herrera@uclv.edu.cu)*

\*\* *Facultad 2 Universidad de Ciencias Informáticas (UCI); C. San Antonio de Los Baños Km. 2.5; La Habana, Cuba. (e-mail: emarichal@uci.cu)*

Resumen: En este trabajo se presenta un método de obtención de modelos borrosos Takagi-Sugeno. Este método actualiza en línea tanto la estructura como los parámetros del modelo mediante la combinación de un nuevo algoritmo de agrupamiento en línea con técnicas de mínimos cuadrados. El algoritmo de agrupamiento propuesto se utiliza para la identificación de la estructura del modelo borroso, generando las clases de las cuales se obtienen los antecedentes de las reglas. La actualización de los parámetros del consecuente se logra mediante estimadores de mínimos cuadrados. Copyright © 2008 CEA.

Palabras Clave: identificación en línea, modelo borroso Takagi-Sugeno, agrupamiento en línea, adaptación del modelo.

### 1. INTRODUCCIÓN

Muchos procesos reales contienen complejas relaciones no lineales y variantes en el tiempo, difíciles de modelar. Esto ha llevado a la investigación y desarrollo de métodos y herramientas sofisticadas de construcción en línea de sistemas inteligentes. Tales sistemas deben ser rápidos, adaptables, no lineales y capaces de captar la dinámica del proceso. Además, las aplicaciones en línea exigen requisitos como eficiencia computacional y que los modelos sean compactos.

En los últimos años se ha prestado especial atención a las técnicas de manejo de datos para la generación de modelos flexibles, entre las que se encuentran los sistemas borrosos. Teóricamente se ha demostrado que, bajo ciertas condiciones, un sistema borroso se comporta como un aproximador universal (Wang, 1997). Dentro de los sistemas borrosos, el modelo Takagi-Sugeno (TS) (Takagi and Sugeno, 1985) se ha convertido en una herramienta ingenieril práctica y potente para el modelado de sistemas complejos, debido a que es capaz de describir sistemas altamente no lineales utilizando un pequeño número de reglas.

Gran parte de los métodos para la obtención de modelos borrosos auto-organizados utilizan métodos de agrupamiento para seccionar el espacio de datos de entrada-salida, combinados con algoritmos genéticos, mínimos cuadrados u optimización del tipo gradiente descendente. Sin embargo, no son muchos los que proporcionan un verdadero proceso de adaptación en línea. Para el aprendizaje en esta condición en los modelos TS, generalmente se utiliza un método de agrupamiento en línea para la determinación de la estructura del modelo. Recientemente, varios algoritmos de modelado borroso han sido reportados con estructura auto-adaptable en línea a partir de métodos de agrupamiento (Kasabov and Song, 2002; Angelov and Filev, 2004; Kukulj and Levi, 2004; Lughofer and Klement, 2005; Yu and Ferreyra, 2005; Angelov and Zhou, 2006).

El objetivo de este trabajo es presentar un algoritmo eficiente para la generación en línea de modelos borrosos del tipo Takagi-Sugeno. Este algoritmo combina un nuevo método de agrupamiento con métodos de mínimos cuadrados.

El trabajo se organiza de la siguiente forma. En la sección 2 se analiza el algoritmo de agrupamiento en línea propuesto. La sección 3 describe el método utilizado para el modelado borroso. La sección 4 presenta los resultados de la aplicación del método y, por último, en la sección 5 se presentan las conclusiones.

### 2. AGRUPAMIENTO EN LÍNEA

Las técnicas de agrupamiento en línea son útiles cuando se requiere modificar un modelo en tiempo real, o cuando existen restricciones temporales o de costo computacional. Los métodos de agrupamiento borroso (*fuzzy clustering*) generalmente conllevan a la optimización de un funcional no lineal y por tanto, genera costos computacionales restrictivos en algunas aplicaciones en tiempo real. Es por ello que para la obtención de modelos borrosos en línea, en los últimos años se aprecia un incremento en los análisis de métodos de agrupamiento basados en la lógica clásica (*hard clustering*) y en particular, los de una sola etapa o pasada (*one-pass*) (Kasabov and Song, 2002; Angelov and Filev, 2004; Martínez *et al.*, 2006; Bouchachia and Mittermeir, 2007).

Los métodos de una sola etapa o pasada (también se les conoce como de pasada simple) se caracterizan porque cada vez que se añade un dato, este es analizado por el método y luego es descartado. Por tanto, el dato es procesado solamente una vez, de ahí el nombre de algoritmos de una sola etapa, a diferencia de los algoritmos iterativos que procesan varias veces los datos al tratar de optimizar una función objetivo que evalúa la calidad de los grupos formados. Estos algoritmos de una sola etapa son populares en las aplicaciones de tiempo real debido a que

reducen las operaciones matemáticas y agilizan el proceso de agrupamiento. Como ejemplos de estos se encuentran el método de agrupamiento evolutivo *ECM* (Kasabov and Song, 2002) y el método de agrupamiento aglomerativo *AddC* (Guedalia *et al.*, 1999). Tomando como base a estos, se propone el método de agrupamiento evolutivo-aglomerativo, *MAEA*.

### 2.1 Método propuesto

El método propuesto es un algoritmo incremental, basado en la distancia euclidiana, que utiliza dos parámetros: un valor de radio umbral, *Rthr*, y un valor de similitud umbral entre clases, *Sthr*.

Su funcionamiento básico es el siguiente. Cada vez que un nuevo dato está disponible, el algoritmo calcula las distancias del dato a los centros de clases existentes, asigna el dato a la clase más cercana si está lo suficientemente cerca del centro de esta clase, y recalcula la posición de dicho centro. En caso contrario, si el dato está muy alejado de todos los centros existentes, el algoritmo crea una nueva clase. Para estas comparaciones se toma como base el parámetro *Rthr*. Si durante la operación del algoritmo, dos centros de clases se acercan lo suficiente, estas clases son unidas. En este caso, la base de comparación es *Sthr*. Por consiguiente, si los parámetros umbrales son pequeños, se generarán muchas clases y a medida de que el valor de estos parámetros crezca, disminuirá el número de clases.

Cada clase  $C_i$  está caracterizada por un centro o prototipo  $Cc_i$  y un peso  $W_i$ , el cual representa el número de puntos de la clase. Este peso se toma en consideración para la actualización de los centros para potenciar la pertenencia a clases concretas, lo cual favorece la convexidad y la interpretabilidad de los modelos borrosos identificados (Díez *et al.*, 2004). Este peso, además, permite caracterizar las clases: clases concentradas que tienen un alto número de muestras y generalmente se relacionan con los estados estables del sistema, o clases dispersas que contienen una pequeña cantidad de puntos y que están más relacionadas con los regímenes transitorios o candidatos potenciales a ruidos y muestras de excepción (*outliers*).

Los pasos del método *MAEA* se describen a continuación:

1. Inicializar la primera clase con el primer dato. Para ello se hace coincidir su centro con dicho dato ( $Cc_1 = z_1$ ), se pone su contador en uno ( $W_1 = 1$ ) y se inicializa el número de clases ( $K = 1$ ).

*PARA* cada nuevo dato  $z_i$

2. Calcular las distancias entre el dato  $z_i$  y cada centro de clase ya creado  $Cc_j$ ,  $j = 1, \dots, K$ .
3. Retener la distancia  $d$  y el índice  $m$  de la clase que se encuentra más cercana al dato  $z_i$ .

$$d = \|z_i - Cc_m\| = \min_j \|z_i - Cc_j\|.$$

4. Si  $d > Rthr$ , se crea una nueva clase, para ello:

$$K = K + 1, Cc_K = z_i, W_K = 1.$$

Luego, se retorna al paso 2.

5. Si  $d \leq Rthr$ ,  $z_i$  es asignado a la clase  $C_m$ , la cual es actualizada como sigue:

$$Cc_m = Cc_m + \frac{z_i - Cc_m}{W_m + 1},$$

$$W_m = W_m + 1.$$

6. Recalcular las distancias entre  $Cc_m$  y el resto de los prototipos  $Cc_j$ ,  $j \neq m$ .
7. Retener la distancia  $D$  y el índice  $n$  de la clase que se encuentra más cercana al centro  $Cc_m$ .

$$D = \|Cc_m - Cc_n\| = \min_{j, j \neq m} \|Cc_m - Cc_j\|.$$

8. Si  $D \leq Sthr$ , unir las clases  $C_m$  y  $C_n$  de la siguiente forma:

$$Cc_m = \frac{Cc_m W_m + Cc_n W_n}{W_m + W_n},$$

$$W_m = W_m + W_n,$$

$$K = K - 1.$$

*final del PARA*

Una vez finalizado el flujo de datos, es útil añadir el siguiente paso de post-procesamiento:

9. Remover los grupos con un peso despreciable, para ello  $\forall j$ ,  $j = 1, \dots, K$ , si  $W_j < \varepsilon$ , eliminar  $C_j$  y hacer  $K = K - 1$ .

El método *MAEA* posee características comunes con los métodos *ECM* y *AddC*, tratando de sacar provecho a las principales potencialidades de estos. Sin embargo, del análisis de las características de cada uno, se tiene que *ECM* no fusiona clases, no actualiza una clase una vez que esta alcanza el radio umbral *Rthr* y no toma en consideración el peso de la clase  $W_j$  para la actualización de los prototipos. En el caso de *AddC*, este fusiona clases una vez alcanzado el número máximo de clases permitidas  $Kmax$  sin tomar en consideración que estas pueden no estar muy cercanas y no limita la dimensión espacial de las clases. Por otro lado, *MAEA* mejora estos aspectos, posibilitando abarcar en un método la adición y la fusión de clases, así como la actualización constante de las clases siguiendo la distribución de los datos tanto internamente en los agrupamiento como de manera global.

El algoritmo *MAEA* posee varias propiedades que lo hacen promisorio para el propósito de la identificación, tales como:

- La dimensión espacial de las clases está limitada y se define al prototipo de cada clase como un punto, características útiles para la generación de modelos borrosos que se empleará posteriormente.
- Es simple, rápido y de una pasada, lo que reduce la carga computacional.
- Es capaz de crear y fusionar clases.
- Es capaz de detectar clases con grandes diferencias en tamaño pues permite detectar tanto clases concentradas (con  $W_i$  altos) como clases dispersas (con  $W_i$  bajos).
- No requiere la determinación del número de clases, que generalmente es un procedimiento iterativo que disminuye la eficiencia computacional.

Sin embargo, este algoritmo propuesto no es universal en el sentido que:

- Solamente detecta clases hipersféricas.
- El resultado final depende del orden en que son presentados los datos. En simulaciones se ha encontrado que la influencia es mayor cuando existen pocas muestras y que disminuye cuando aumenta el número de datos. Esta característica no es tan influyente en el modelado en línea pues precisamente en este caso se persigue que el modelo se adapte continuamente a los cambios que ocurren en el proceso.

- Requiere la definición previa de  $Rthr$  y  $Sthr$ . La selección correcta de los parámetros del algoritmo necesita cierto conocimiento de los usuarios, lo cual es general para los métodos de agrupamiento incremental basados en la distancia. Otra solución podría ser la utilización de algoritmos genéticos u otra técnica de optimización, pero esto es aún un tema abierto de investigación.

Por último, es importante destacar que el algoritmo comienza sin información previa sobre las características de los datos y solamente con un dato de muestra. Esta es una característica interesante que hace al método potencialmente útil en control adaptativo, robótica, sistemas de diagnóstico; y como una herramienta para la adquisición de conocimiento a partir de datos (Angelov and Filev, 2004). Además, debido a que algunos métodos de agrupamiento borrosos, como el Gustafson-Kessel ( $GK$ ), pueden presentar problemas numéricos cuando solamente se dispone de un pequeño número de datos (Babuska, 1998), este algoritmo representa una alternativa promisoría en la obtención de modelos en estos casos.

## 2.2 Ejemplos de agrupamiento

Para evaluar el comportamiento de los métodos, en estos ejemplos se emplean varios índices de error cuantitativos. Tomando la distancia entre la muestra  $z_i$  y el centro de clase más cercano  $Cc_j$ , el índice  $J$  se define como:

$$J = \sum_{i=1}^N \min_j \|z_i - Cc_j\|, \quad (1)$$

donde  $N$  es la cantidad de datos.

Mientras el índice  $J$  proporciona una medida del funcionamiento promedio, el siguiente índice de distorsión local proporciona una buena medida de la calidad de representación de cada clase individual:

$$J_L = \sum_{i=1}^N \frac{1}{N_j} \left( \sum_{i, z_i \in C_j} \min_j \|z_i - Cc_j\| \right), \quad (2)$$

donde  $K$  es el número de clases generadas,  $C_j$  es la  $j$ -ésima clase y  $N_j$  es el número de puntos de  $C_j$ . La distorsión de cada punto es la distancia entre el punto y su prototipo más representativo, normalizada por el tamaño de su clase. Esto asegura que el efecto de cada clase en esta medida de comportamiento es relativamente igual, incluso las clases pequeñas influyen al resultado final (Guedalia *et al.*, 1999).

Por último, también se calcula la distancia máxima entre un punto que pertenece a una clase y el prototipo de la clase correspondiente ( $maxDist$ ).

**Primer ejemplo.** Se aplican los métodos  $ECM$ ,  $AddC$  y  $MAEA$  a una base de datos sencilla compuesta por 54 datos, con características similares a los datos de procesos no lineales, incluyendo datos erróneos como los producidos por los ruidos, errores de medición, etc. Los datos son bidimensionales en aras de simplificar el análisis gráfico. En este caso, las condiciones de diseño son: número máximo de clases  $Kmax = 3$  para  $AddC$ ,  $Rthr = 0,3$  para  $ECM$  y  $MAEA$ , y  $Sthr = 0,15$  para este último método. La Figura 1 muestra los resultados gráficos y los resultados numéricos aparecen en la Tabla 1.

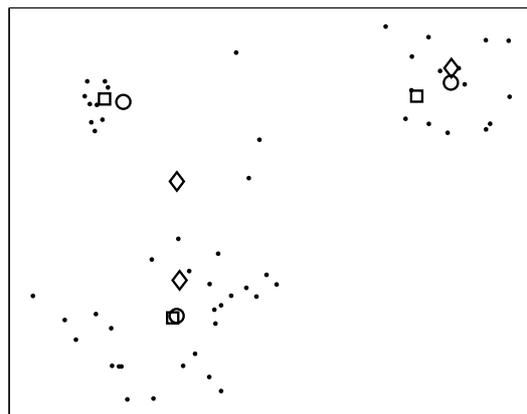


Figura 1. Resultados del agrupamiento del primer ejemplo: datos (puntos), centros ( $\diamond$ -ECM,  $\square$ -AddC,  $\circ$ -MAEA)

Tabla 1. Resultados del agrupamiento del primer ejemplo en 3 clases.

Método	ECM	AddC	MAEA
$J$	9.8656	8.2189	8.0177
$J_L$	0.5558	0.4327	0.4215
$maxDist$	0.3806	0.3697	0.3334

**Segundo ejemplo.** El conjunto de datos del proceso de combustión de gas en un horno (Box and Jenkins, 1970) es frecuentemente utilizado para validar algoritmos de identificación, control y aprendizaje adaptativo. El ejemplo consiste en 296 datos de entrada-salida, muestreados cada 9 segundos. En dicho proceso, la variable de entrada es el flujo de gas  $u(k)$  y la variable de salida es la concentración de dióxido de carbono  $y(k)$ . Se considera que el valor instantáneo de la salida en  $y(k)$  es influenciada por  $u(k-4)$  y  $y(k-1)$  (Kukolj and Levi, 2004). En este caso, el agrupamiento se realiza en el espacio de las entradas, de manera similar a (Kasabov and Song, 2002). Las condiciones de diseño en este ejemplo son:  $Kmax = 10$ ,  $Rthr = 0,12$  y  $Sthr = 0,06$ . Los resultados se muestran en la Tabla 2.

Tabla 2. Resultados del agrupamiento del conjunto de datos de Box-Jenkins en 10 clases.

Método	ECM	AddC	MAEA
$J$	27.368	25.593	25.043
$J_L$	0.9435	0.8851	0.8927
$maxDist$	0.2057	0.1873	0.1787

Nótese que en ambos ejemplos, el algoritmo propuesto presenta índices de error pequeños y las menores distancias máximas, indicando que con  $MAEA$  se obtienen clases más compactas y que este sigue la tendencia de los datos al estar los prototipos cercanos a la mayor concentración de datos dentro de cada clase.

## 3. MODELADO BORROSO TAKAGI-SUGENO

### 3.1 Sistema borroso Takagi-Sugeno (TS)

Aquí se utiliza el conocido sistema de inferencia Takagi-Sugeno (Takagi and Sugeno, 1985). Dicho sistema está compuesto por  $L$  reglas borrosas de la siguiente forma:

$$\begin{aligned} \mathfrak{R}_i : & \text{if } x_1 \text{ is } A_{i1} \text{ and } \dots \text{ and } x_r \text{ is } A_{ir} \\ & \text{then } y_i = a_{i0} + a_{i1}x_1 + \dots + a_{ir}x_r, \\ & i = 1, \dots, L \end{aligned} \quad (3)$$

donde  $x_j, j = 1, \dots, r$ , son las variables de entrada definidas en los universos de discurso  $X_j$  y  $A_{ij}$  son los conjuntos borrosos definidos por sus funciones de pertenencia  $\mu_{A_{ij}} : X_j \rightarrow [0, 1]$ . En la parte del consecuente,  $y_i$  es la salida de la regla  $i$  y  $a_{ij}$  son escalares.

Para un vector de entrada  $\mathbf{x} = [x_1, x_2, \dots, x_r]^T$ , cada función del consecuente puede expresarse como:

$$y_i = \mathbf{a}_i^T \mathbf{x}_e, \text{ donde } \mathbf{x}_e = [1, \mathbf{x}^T]^T. \quad (4)$$

El resultado de la inferencia, la salida del sistema  $y$ , es el promedio ponderado de la salida de cada regla  $y_i$ , como se indica a continuación:

$$y = \sum_{i=1}^L \tau_i y_i = \sum_{i=1}^L \tau_i \mathbf{a}_i^T \mathbf{x}_e, \quad (5)$$

donde

$$\tau_i = \frac{w_i}{\sum_{i=1}^L w_i}, \quad (6)$$

es el nivel de impacto normalizado de la regla  $i$ .

Nótese que el sistema borroso TS posibilita la aplicación de los métodos de mínimos cuadrados lineales ya que este solo requiere que el modelo sea lineal en los parámetros. Además, si la salida del sistema aparece como una de sus entradas, se obtiene el llamado sistema borroso TS dinámico.

Finalmente, todas las funciones de pertenencia utilizadas son del tipo *gaussianas*:

$$\mu(x_d; c_d, \sigma) = \exp\left(-\left(\frac{x_d - c_d}{2\sigma}\right)^2\right), \quad (7)$$

donde  $c_d$  es el valor del centro de la clase en la dimensión  $x_d$ , y  $\sigma$  es proporcional a la distancia desde el centro del agrupamiento hasta elemento más alejado entre los que pertenecen a dicha clase.

### 3.2 Algoritmo de aprendizaje

El algoritmo de aprendizaje en línea consta de dos pasos: la identificación de la estructura y la determinación de los parámetros. La identificación de la estructura agrupa los datos y encuentra las clases necesarias. Pero en la identificación en línea, siempre hay nuevos datos arribando y las clases deberán cambiar de acuerdo a esos nuevos datos. En este paso, aquí se utiliza el método de agrupamiento propuesto. Una nueva regla es generada solamente si hay información significativa presente en los datos. Una vez realizada la identificación de la estructura, los resultados del agrupamiento se utilizan para ajustar los centros y anchos de las funciones de pertenencia según (7).

Las funciones de los consecuentes son creadas y actualizadas por medio de estimadores de mínimos cuadrados de manera similar a (Kasabov and Song, 2002) y (Kukulj and Levi, 2004). Para ello se forman las matrices diagonales  $\mathbf{T}_i$  ( $i = 1, \dots, L$ ) donde el  $k$ -ésimo elemento de cada diagonal principal es formado utilizando los valores de los niveles de impacto normalizado obtenidos de (6). De aquí, una matriz de composición  $\mathbf{X}'$  es formada de la siguiente forma:

$$\mathbf{X}' = [(\mathbf{T}_1 \mathbf{X}_e), (\mathbf{T}_2 \mathbf{X}_e), \dots, (\mathbf{T}_K \mathbf{X}_e)] , \quad (8)$$

donde la matriz  $\mathbf{X}_e = [1, \mathbf{X}]$  está conformada por las filas  $\mathbf{x}_{e_k}^T = [1, \mathbf{x}_k^T]$ .

La siguiente expresión

$$\mathbf{a} = \left(\mathbf{X}'^T \mathbf{X}'\right)^{-1} \mathbf{X}'^T \mathbf{Y}, \quad (9)$$

es utilizada para obtener la matriz inicial de los parámetros del consecuente  $\mathbf{a} = [\mathbf{a}_1^T \mathbf{a}_2^T \dots \mathbf{a}_K^T]^T$ . Esta matriz es calculada con un conjunto de datos compuesto por  $m$  pares de datos. Nótese que (9) se puede reescribir como:

$$\begin{aligned} \mathbf{P} &= \left(\mathbf{X}'^T \mathbf{X}'\right)^{-1}, \\ \mathbf{a} &= \mathbf{P} \mathbf{X}'^T \mathbf{Y}. \end{aligned} \quad (10)$$

Para la actualización de la matriz de parámetros  $\mathbf{a}$ , en este trabajo se utiliza un estimador de mínimos cuadrados recursivos (RLS) con un factor de olvido. Sea  $\mathbf{x}'_k$  el  $k$ -ésimo vector fila de la matriz  $\mathbf{X}'$  y  $\mathbf{y}_k$  el  $k$ -ésimo elemento de  $\mathbf{Y}$ , entonces  $\mathbf{a}$  puede calcularse iterativamente mediante las ecuaciones:

$$\begin{aligned} \mathbf{P}_{k+1} &= \frac{1}{\lambda} \left( \mathbf{P}_k - \frac{\mathbf{P}_k \mathbf{x}'_{k+1} \mathbf{x}'_{k+1}^T \mathbf{P}_k}{\lambda + \mathbf{x}'_{k+1}^T \mathbf{P}_k \mathbf{x}'_{k+1}} \right), \\ \mathbf{a}_{k+1} &= \mathbf{a}_k + \mathbf{P}_{k+1} \mathbf{x}'_{k+1} \left( \mathbf{y}_{k+1} - \mathbf{x}'_{k+1}^T \mathbf{a}_k \right), \end{aligned} \quad (11)$$

donde  $\lambda$  es un factor de olvido constante, con valores típicos entre 0.8 y 1, utilizado para posibilitar un mejor seguimiento a sistemas variantes en el tiempo. Los valores iniciales,  $\mathbf{P}(0)$  y  $\mathbf{a}(0)$ , son calculados utilizando (10). Las ecuaciones en (11) tienen una interpretación intuitiva: los parámetros nuevos son iguales a los parámetros anteriores más un término de corrección basado en el nuevo vector de datos  $\mathbf{x}'_{k+1}$ .

El proceso de obtención del modelo borroso que aquí se sigue se basa en el enfoque de aprendizaje de DENFIS (Kasabov and Song, 2002), el cual se ejecuta en el modo en-línea, pero empleando el algoritmo de agrupamiento *EACM* en lugar del método *ECM*. Este proceso se resume en los siguientes pasos:

1. Inicializar el modelo borroso. Para ello:
  - (a) Tomar los primeros  $m$  datos.
  - (b) Obtener los centros de clase aplicando el método de agrupamiento MAEA.
  - (c) Obtener las reglas iniciales creando el antecedente de cada una con (7) y obteniendo los valores iniciales de  $\mathbf{P}$  y  $\mathbf{a}$  con (10).
2. Tomar el próximo dato.
3. Actualizar recursivamente los centros de clase utilizando el método MAEA.
4. Posible modificación de la base de reglas. Una nueva regla borrosa es creada si hay información significativa presente en la nueva clase creada. Para esto se utiliza la siguiente regla:

*if* el peso no es despreciable ( $W_i > \varepsilon$ ),  
*then* se crea una nueva regla borrosa.
5. Actualizar los antecedentes mediante (7)
6. Calcular recursivamente los parámetros de los consecuentes mediante (11)
7. Predecir la salida del próximo instante de tiempo mediante el modelo borroso TS

La ejecución del algoritmo continúa en el próximo instante de tiempo desde el paso 2.

El método de aprendizaje presentado asegura una base de reglas que evoluciona dinámicamente, actualizándose o modificándose mientras hereda la mayoría de las reglas.

#### 4. EJEMPLOS

El método de identificación expuesto es aplicado a dos ejemplos: la identificación de un sistema dinámico no lineal y la predicción de un parámetro en un proceso fermentativo. El primer ejemplo, bien conocido de la literatura al ser utilizado frecuentemente en la validación de algoritmos de identificación en el área de los sistemas borrosos, las redes neuronales y los sistemas híbridos, permitirá valorar las potencialidades del algoritmo. El segundo es un caso real de fermentación alcohólica donde el modelo borroso de estimación paramétrica se combina con un modelo macroscópico descrito por ecuaciones de balance.

##### 4.1 Sistema dinámico no lineal

Este ejemplo aborda la identificación de un sistema dinámico no lineal descrito por la siguiente ecuación no lineal (Narendra and Parthasarathy, 1990):

$$y(k+1) = \frac{y(k)}{1+y^2(k)} + u^3(k), \quad (12)$$

donde  $u(k)$  y  $y(k)$  son la entrada y la salida del sistema respectivamente. La entrada  $u(k)$  se define como:

$$u(k) = \begin{cases} \sin\left(\frac{2\pi k}{100}\right) & k \in [1, 99] \\ \sin\left(\frac{2\pi k}{100}\right) + \sin\left(\frac{0,5}{1+\rho}\right) \sin\left(\frac{2\pi k}{10}\right) & k \in [100, 250] \end{cases} \quad (13)$$

donde  $\rho \in [0, 1]$  es un número aleatorio uniformemente distribuido. Aquí se considera que la salida  $y(k+1)$  es influenciada por  $u(k)$  y  $y(k)$ , las cuales se toman como entradas del modelo borroso. Por tanto, el modelo en este ejemplo está dado por:

$$\mathfrak{R}_i : \text{if } u(k) \text{ is } A_{i1} \text{ and } y(k) \text{ is } A_{i2} \text{ then } y_i(k+1) = a_{i0} + a_{i1}u(k) + a_{i2}y(k). \quad (14)$$

Se toman los 10 primeros datos ( $m = 10$ ) para obtener el modelo borroso inicial de una regla y se selecciona un factor de olvido de 0.95. El modelo borroso evoluciona desde una regla hasta un total de cuatro reglas (figura 2). La figura 3 muestra el comportamiento de la salida inferida por el modelo borroso y la salida del sistema. En este ejemplo se obtiene un índice  $MSE$  igual a 0.0159. En las figuras 4 y 5 se muestran las adaptaciones de las funciones de pertenencia del antecedente y de los parámetros del consecuente, respectivamente, para las dos primeras reglas del modelo.

Nótese que el modelo es capaz de seguir los cambios en el sistema y que este adapta tanto la estructura de la base de reglas (partiendo de una regla en el modelo borroso inicial, el modelo final tiene cuatro reglas) como los parámetros durante la ejecución del algoritmo de aprendizaje.

##### 4.2 Proceso de producción de bioetanol

La bacteria *Zymomonas mobilis* se ha convertido en objeto creciente de estudio para la producción de bioetanol debido a

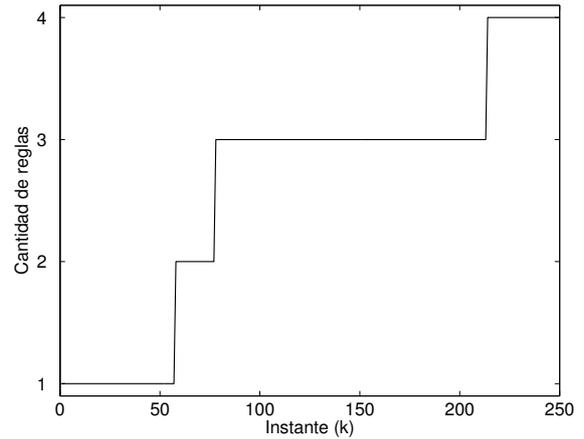


Figura 2. Evolución del número de reglas

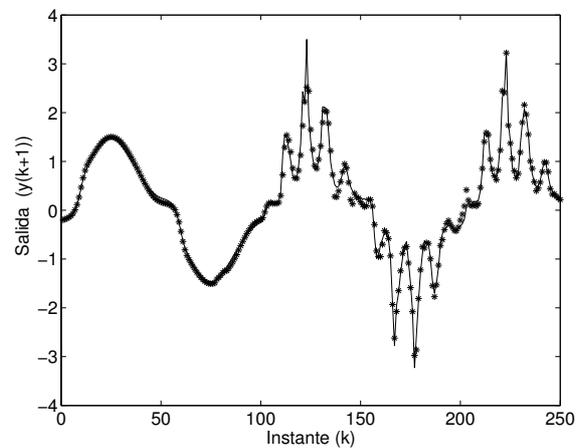


Figura 3. Estimación en línea del sistema no lineal; datos reales (-), datos estimados (\*)

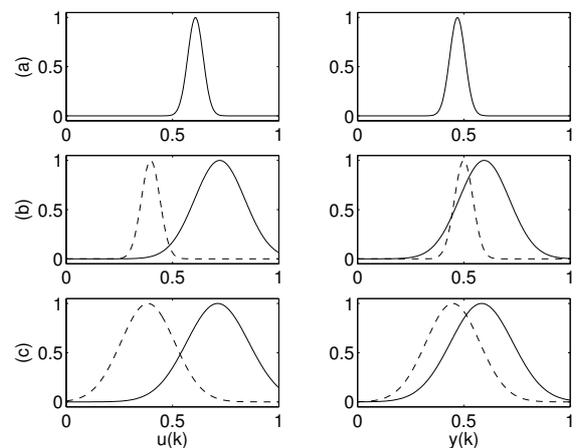


Figura 4. Funciones de pertenencia de las dos primeras reglas; (a) en  $k = 10$  al obtenerse el modelo borroso inicial con una regla, (b) en  $k = 60$  al crearse la segunda regla, (c) en  $k = 250$  al finalizar el aprendizaje

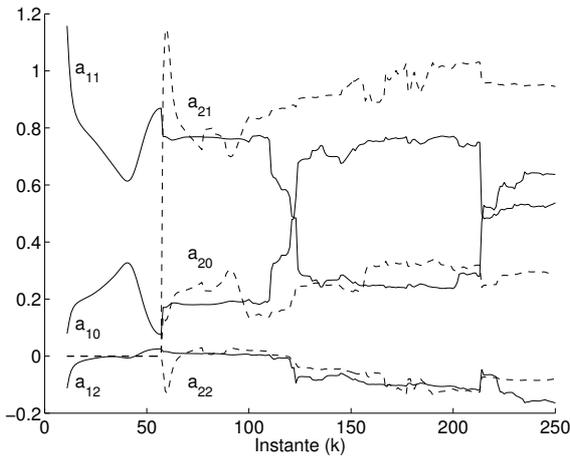


Figura 5. Evolución de los parámetros del consecuente de las dos primeras reglas; (—) primera regla, (---) segunda regla

que presenta altos rendimientos, pero junto a esto, tiene comportamientos altamente no lineales y oscilatorios (Echeverry *et al.*, 2004). Desde el punto de vista del control esto representa una gran dificultad, por lo que obtener un modelo preciso y adecuado es de gran importancia para maximizar el rendimiento y potenciar su utilización.

En los modelos cinéticos en los cuales las variables de estado son las concentraciones de biomasa, de sustrato y de etanol, el modelado de la velocidad de crecimiento de la biomasa ( $\mu$ ) es uno de los pasos más importantes, pero aún no existe una solución aceptada por todos.

A continuación se obtiene un modelo borroso TS para la estimación de  $\mu$  a partir de datos, el cual describe su dependencia de las variables de estado y del valor del parámetro en el instante de tiempo anterior, con lo cual se obtiene un modelo dinámico. Dicho modelo borroso se incorpora después en el siguiente modelo matemático (Garro, 1993):

$$\frac{dX}{dt} = \mu X, \tag{15}$$

$$\frac{dS}{dt} = - \left( \frac{\mu}{Y_{XS}} + \eta \right) X, \tag{16}$$

$$\frac{dP}{dt} = (\alpha\mu + \beta) X, \tag{17}$$

donde  $X, S, P$  son las concentraciones de biomasa, sustrato y del producto etanol (g/l) respectivamente,  $\mu$  es la velocidad específica de crecimiento (1/h),  $Y_{XS}$  es la constante de rendimiento celular para un sustrato determinado,  $\eta$  es la constante de mantenimiento (1/h),  $\alpha$  expresa la porción del etanol producido asociado con el crecimiento de las células,  $\beta$  expresa la porción del etanol producido para el mantenimiento de las células (1/h).

En este ejemplo se obtiene un modelo de cuatro reglas. Los resultados de la estimación del parámetro se muestran en la figura 6 y en la figura 7 se grafican las estimaciones de las variables de estado, en que para el caso del producto etanol, se obtienen un índice de error  $MSE = 3,8 * 10^{-3}$ . Las condiciones iniciales son:  $X_0 = 0,03$  g/l,  $S_0 = 20,0$  g/l y  $P_0 = 0,0$  g/l.

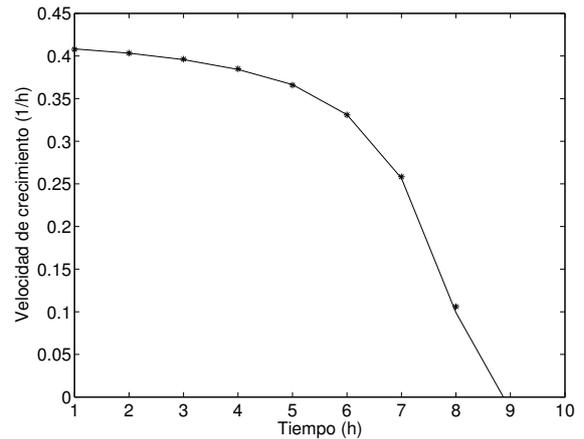


Figura 6. Estimación en línea del sistema no lineal; datos reales (\*), datos estimados (—)

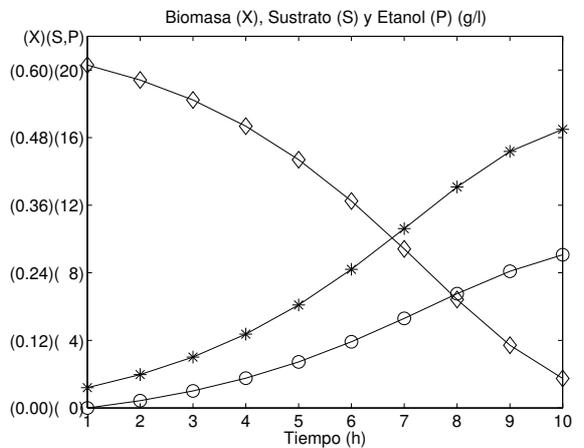


Figura 7. Estimaciones de la biomasa (\*), sustrato (◊) y etanol (◊); datos reales (símbolos), datos estimados (líneas). Nótese que la escala para la biomasa es 0-0.6 g/l y la del sustrato y el etanol es 0-20 g/l

### 5. CONCLUSIONES

El trabajo ha presentado una propuesta de método de agrupamiento incremental y su aplicación en un método de generación en línea de modelos borrosos Takagi-Sugeno. La técnica de agrupamiento en línea propuesta es analizada y comparada con las técnicas que le sirven de base, obteniéndose mejoras en el proceso de agrupamiento y en el resultado final.

El método de identificación borrosa utilizado actualiza en línea tanto la estructura como los parámetros del modelo. El método se caracteriza por tener una gran exactitud de modelado y por un proceso de aprendizaje rápido. La sencillez y la exactitud del método propuesto lo hacen conveniente para la obtención de modelos de identificación adaptables en línea y para la utilización en aplicaciones de tiempo real.

### AGRADECIMIENTOS

Este trabajo ha sido parcialmente financiado por el proyecto MES 6.111 del Ministerio de Educación Superior de Cuba

## REFERENCIAS

- Angelov, P. and D. Filev (2004). An approach to online identification of Takagi-Sugeno fuzzy models. *IEEE Trans. on Syst. Man, and Cyb.-Part B* **34**(1), 484–498.
- Angelov, P. and X.-W. Zhou (2006). Evolving fuzzy systems from data streams in real-time. In: *EFS'06*. IEEE Press. Ambleside, UK.
- Babuska, R. (1998). *Fuzzy Modeling for Control*. Kluwer Acad. Pub.. Boston, USA.
- Bouchachia, A. and R. Mittermeir (2007). Towards incremental fuzzy classifiers. *Soft Comput.* **11**(2), 193–207.
- Box, G.E.P. and G.M. Jenkins (1970). *Time series analysis, forecasting and control*. Holden Day. San Francisco, USA.
- Díez, J.L., J.L. Navarro and A. Sala (2004). Algoritmos de agrupamiento en la identificación de modelos borrosos. *RIAI* **1**(2), 32–41.
- Echeverry, N., O. Quintero, M. Ramírez and H. Álvarez (2004). Control de un biorreactor para fermentación alcohólica en continuo. In: *CLCA'04*. La Habana, Cuba.
- Garro, O.A. (1993). Fermentación alcohólica con *Zymomonas* sp. Estudio de modelos matemáticos y su verificación mediante ensayos de fermentación. Tesis doctoral. Universidad Nacional de Tucumán, Argentina.
- Guedalia, I., M. London and M. Werman (1999). An on-line agglomerative clustering method for non-stationary data. *Neural Comput.* **11**(2), 521–540.
- Kasabov, N. and Q. Song (2002). DENFIS: Dynamic evolving neural-fuzzy inference system and its application for time-series prediction. *IEEE Trans. on Fuzzy Syst.* **10**(2), 144–154.
- Kukulj, D. and E. Levi (2004). Identification of complex systems based on neural and Takagi-Sugeno fuzzy model. *IEEE Trans. on Syst. Man, and Cyb.-Part B* **34**(1), 272–282.
- Lughofer, E. and E.P. Klement (2005). FLEXFIS: A variant for incremental learning of Takagi-Sugeno fuzzy systems. In: *FUZZ-IEEE 2005*. Reno, USA. pp. 915–920.
- Martínez, B.L., F. Herrera and J.A. Fernández (2006). Métodos de agrupamiento clásico para el modelado difuso en línea. In: *FIE'06*. Santiago de Cuba, Cuba.
- Narendra, K. S. and K. Parthasarathy (1990). Identification and control of dynamical systems using neural networks. *IEEE Trans. on Neural Netw.* **1**(1), 4–27.
- Takagi, T. and M. Sugeno (1985). Fuzzy identification of systems and its applications to modelling and control. *IEEE Trans. on Syst. Man, and Cyb.* **15**(1), 116–132.
- Wang, L.X. (1997). *Adaptive fuzzy systems and control*. 2nd ed.. Prentice Hall. Englewood Cliffs, USA.
- Yu, W. and A. Ferreyra (2005). On-line clustering for nonlinear system identification using fuzzy neural networks. In: *FUZZ-IEEE 2005*. Reno, USA. pp. 678–683.