

# Modelado Vía Agrupamiento en Línea y SVM Difuso

Tovar Rodríguez Julio César\*, Ortiz Rodríguez Floriberto, Mariaca Gaspar Carlos Román y Velázquez Velázquez Juan Eduardo

ESIME Zacatenco IPN Av. IPN s/n  
 Lindavista, Gustavo A. Madero, C.P. 07738, México, Distrito Federal México  
 \*jctovar@ipn.mx Teléfono: (52)-55-57296000 ext 54603

## Resumen

En este artículo se presenta el agrupamiento en línea y la modificación del kernel de los Support Vector Machines (SVM's), por un kernel difuso para el modelado de una planta no lineal. La estructura de identificación se realiza por el agrupamiento en línea y los Support Vector Machines difusos (FSVM's). Cuando el proceso es lento, las reglas difusas se pueden obtener automáticamente. Se usan los parámetros de identificación de las redes neuronales difusas. El rango de aprendizaje variante en el tiempo asegura la estabilidad del modelado del error.

**Palabras clave:** Agrupamiento en línea, SVM's, kernel, función de membresía, reglas difusas.

## 1. Introducción

El propósito de FSVMs (fuzzy support vector machines Figura1) es tratar los datos entrenados con diferente importancia en el proceso de entrenamiento. La falta del término de la función de costo se minimiza, reformulando la construcción del problema de optimización, y la construcción del Lagrangiano así que las soluciones para el hiperplano óptimo en la primera forma pueden ser encontradas en la forma dual.

El SVM puede separar los datos en dos clases con un hiperplano de margen máximo [2]. Si el entrenamiento es separable para el hiperplano, la función es escogida como  $f(x)=(w \cdot x)+b$ . El margen es definido como la distancia mínima de una muestra para la superficie de la solución. El margen en turno que podemos medir para la longitud del vector  $w$ , de forma que los puntos cercanos al hiperplano satisface  $|(w \cdot x) + b| = 1$ .

En este trabajo usamos SVM para el cálculo de la función de estimación. Para encontrar los vectores de soporte en el  $j$  de grupo, se usan los datos de entrada /salida  $[y(k), x(k)]$ ,  $k \in [l_1^j, l_2^j]$  para aproximar una función no lineal. Considere la regresión en un conjunto de funciones no lineales

$$f(x) = w^T \varphi(x) + b \quad (1)$$

donde  $K(x, x_k) = \varphi(x)^T \varphi(x_k)$ .

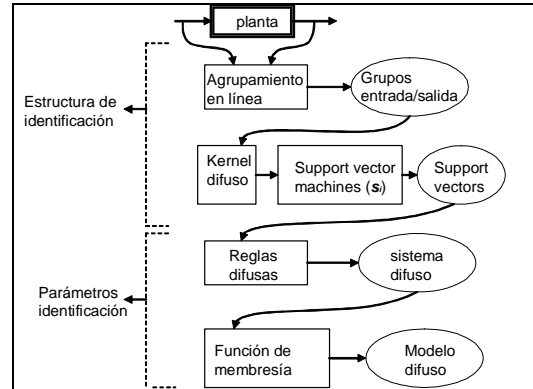


Figura 1. Agrupamiento en línea FSVM's y modelado difuso

## 2. Agrupamiento en línea

Se quiere el modelo del siguiente sistema no lineal suave de espacio de estados discretos en el tiempo

$$x(k+1) = f[x(k), u(k)], \quad y(k) = h[x(k)] \quad (2)$$

donde  $u(k) \in \mathcal{U}^m$  es el vector de entrada,  $x(k) \in \mathcal{X}^n$  es el vector de estado, y  $y(k) \in \mathcal{Y}^m$  es el vector de salida,  $f, y$

$h$  son funciones suaves no lineales. (1) se puede escribir como

$$\begin{aligned} y(k) &= h[x(k)] = F_1[x(k)], \\ y(k+1) &= h[f[x(k), u(k)]] = F_2[x(k), u(k)] \\ y(k+n-1) &= F_n[x(k), u(k), \dots, u(k+n-2)] \end{aligned} \quad (3)$$

Denotando  $y(k)=[y(k), y(k+1), \dots, y(k+n-1)]^T$ ,  $U(k)=[u(k), u(k+1), \dots, u(k+n-2)]^T$ ,  $y(k)=F[x(k), U(k)]$ ,  $F=[F_1 \dots F_n]^T$ . Desde (2) es un sistema no lineal suave, (3) se puede expresar como  $x(k+1)=g[Y(k+1), U(k+1)]$ . Esto conduce a un modelo multivariable NARMA

$$y(k) = h[x(k)] = \Psi[X(k)] \quad (4)$$

donde

$$X(k)=[y(k-1), y(k-2), \dots, u(k-d), u(k-d-1), \dots]^T \quad (5)$$

$\Psi(\cdot)$  es una función no lineal desconocida representando las dinámicas de la planta,  $u(k)$  y  $y(k)$  son entradas y salidas escalares medibles,  $d$  es un retardo en el tiempo.

El objetivo de la identificación de estructuras es la de particionar los datos de entrada y salida  $[y(k), x(k)]$  del sistema no lineal y extraer reglas difusas. Se usa el siguiente ejemplo para explicar la importancia del agrupamiento en línea en el mismo tiempo indexado. Se considera la función no lineal como

$$y(k) = f[x(k)] \quad (6)$$

Por la norma del método de agrupamiento en línea propuesto en [1] la entrada y la salida se pueden particionar en 4 grupos, como se puede ver en la Figura 2. Estos grupos se pueden formar dentro de 4 reglas como "IF  $x(k)$  is  $A_j$  THEN  $y(k)$  is  $B_j$ ",  $j=1..4$ . Obviamente, para la 3rd regla: "IF  $x(k)$  is  $A_3$  THEN  $y(k)$  is  $B_3$ ", esta no satisface la relación (6), porque la precondition  $x(k)$  y la consecuencia  $y(k)$  no ocurre al mismo tiempo.

La idea básica del agrupamiento en línea es que los espacios particionados entrada y salida son acarreados fuera en el mismo tiempo indexado. Si la distancia desde el punto al centro es menor que la longitud requerida, el punto está dentro de ese grupo. Cuando un nuevo dato llega, el centro y el grupo deberían ser cambiados de acuerdo al nuevo dato. Se da el siguiente algoritmo.

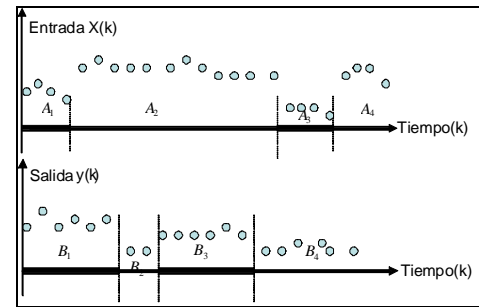


Figura 2. Particionamiento del espacio de entrada/salida

La distancia Euclidiana al tiempo  $k$  se define como

$$\begin{aligned} d_{k,x} &= \left( \sum_{i=1}^n \left[ \frac{x_i(k) - \bar{x}_i^{-j}}{x_{i,\max} - x_{i,\min}} \right]^2 \right)^{1/2} \\ d_{k,y} &= \left| \frac{y(k) - \bar{y}^{-j}}{y_{\max} - y_{\min}} \right| \\ d_k &= \alpha d_{k,x} + \beta d_{k,y} \end{aligned} \quad (7)$$

donde  $x_{i,\max} = \max\{x_i(k)\}$ ,  $x_{i,\min} = \min\{x_i(k)\}$ ,  $y_{\max} = \max\{y(k)\}$ ,  $y_{\min} = \min\{y(k)\}$ ,  $\bar{x}_i(k)$  y  $\bar{y}(k)$  son los centros de  $x_i$  y  $y$  al tiempo  $k$ ,  $\alpha$  y  $\beta$  son factores positivos, normalmente se pueden escoger  $\alpha = \beta = (1/2)$ . Para el Grupo  $j$ , los centros son actualizados como sigue

$$\begin{aligned} \bar{x}_i^{-j} &= \frac{1}{l_2^j - l_1^j + 1} \sum_{l=l_1^j}^{l_2^j} x_i(l) \\ \bar{y}^{-j} &= \frac{1}{l_2^j - l_1^j + 1} \sum_{l=l_1^j}^{l_2^j} y_i(l) \end{aligned} \quad (8)$$

donde  $l_1^j$  es el primer número del Grupo  $j$ ,  $l_2^j$  es el último número del Grupo  $j$ . La longitud del Grupo  $j$  es  $m^j = l_2^j - l_1^j + 1$ . El intervalo de tiempo del Grupo  $j$  es  $[l_1^j, l_2^j]$ . El proceso de la estructura de identificación se puede formar con los siguientes pasos

1. Para el primer dato  $G_x$ ,  $k=1$ .  $y(1)$ ,  $x_i(1)$  son los centros del primer grupo,  $\bar{x}_i^1 = x(1)$ ,  $\bar{y}^1 = y(1)$ ,  $l_1^1 = l_2^1 = 1$ .
2. Si un nuevo dato  $[y(k), x_i(k)]$  llega,  $l_2^j = l_2^j + 1$ , se usa (7) y (8) para calcular  $d_k$ . Si ningún dato nuevo llega, ir a 5.

3. Si  $d_k \leq L$  entonces  $[y(k), x_i(k)]$  está todavía en el grupo  $G_j$ , ir a 2
4. Si  $d_k > L$  entonces  $[y(k), x_i(k)]$  está en un nuevo grupo  $j=j+1$ , el centro de  $G_j$  es  $\bar{x}_i^j = x(k)$ ,  $\bar{y}^j = y(k)$ ,  $l_i^j = l_i^j = k$ , ir a 2
5. Checar la distancias entre todos los centros  $\bar{x}^j$ ,  $\bar{y}^j$ , Si  $\sum_{i=1}^n \left[ \bar{x}_i^p - \bar{x}_i^q \right]^2 + \left[ \bar{y}^p - \bar{y}^q \right]^2 \leq L^2$  los dos grupos  $G_p$  y  $G_q$  están combinados dentro de un grupo.

### 3. Kernel difuso

Existen muchas maneras de escoger el kernel  $K(x, x_k)$ , solo se requiere que  $K(x, x_k)$  satisfaga la condición de Mercer [2]. Por ejemplo el kernel lineal  $K(x, x_k) = x_k^T x$ , el kernel MPL  $K(x, x_k) = \tanh(k_1 x_k^T x + k_2)$ , kernel RBF  $K(x, x_k) = \exp(-\|x - x_k\|^2 / \sigma^2)$ . En este trabajo, se usa el kernel difuso  $K(\hat{x}, \hat{z})$  el cuál se define como

$$K(x, x_k) = \begin{cases} \prod_{i=1}^M u_i(x_k) u_i(x) & x_k \text{ y } x \text{ ambos estan en } j\text{th grupo} \\ 0 & \text{de otra manera} \end{cases}$$

Donde  $\hat{x} = [x_1, x_2, x_3, \dots, x_M] \in \mathbb{R}^M$  y  $\hat{z} = [z_1, z_2, z_3, \dots, z^M] \in \mathbb{R}^M$  son cualesquiera dos muestras entrenadas.  $u_i(x_k)$  es la función de membresía del  $j$ th agrupamiento.

Sea el conjunto entrenado  $S = \{(x_1, y_1), (x_2, y_2), \dots, (x_v, y_v)\}$  con variables interpretativas  $x_i$  y las clases etiquetadas correspondientes  $y_i$ , para todo  $i=1, 2, \dots, v$  donde  $v$  es el número total de muestras entrenadas. Se asume que las muestras entrenadas están particionadas dentro del agrupamientos. Se puede ejecutar la siguiente permutación de muestras entrenadas:

$$\begin{aligned} \text{grupo 1} &= \{(x_{k_1}^1, y_{k_1}^1), \dots, (x_{k_1}^1, y_{k_1}^1)\} \\ \text{grupo 2} &= \{(x_{k_2}^2, y_{k_2}^2), \dots, (x_{k_2}^2, y_{k_2}^2)\} \\ &\vdots \\ \text{grupo } l &= \{(x_{k_l}^l, y_{k_l}^l), \dots, (x_{k_l}^l, y_{k_l}^l)\} \end{aligned} \quad (9)$$

donde  $k_g$ ,  $g = 1, 2, \dots, l$  es el número de puntos de calidad para el  $g$ th agrupamiento, así que se tiene  $\sum_{g=1}^l k_g = v$ . Entonces el kernel difuso puede ser calculado usando el conjunto entrenado en (9), y la

obtención matricial kernel  $K$  se puede reescribir de la siguiente forma:

$$K = \begin{bmatrix} K_1 & 0 & \dots & 0 \\ 0 & K_2 & \dots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & K_l \end{bmatrix} \in \mathbb{R}^{v \times v} \quad (10)$$

Como escoger la función de membresía  $u_j(x_i)$  es otro problema. La función Gaussiana y la función triángulo son las funciones más populares para la función de membresía de sistemas difusos. Cuando  $u_j(x_i)$  es una función Gaussiana, la función kernel es

$$K(x_k, x_j) = \varphi(x_k) \varphi(x_j) = \exp\left(-\frac{\|x_k - x_j\|^2}{2\sigma^2}\right) \quad (11)$$

Se supone que el conjunto difuso entrenado es obtenido ya, el siguiente paso es formular el FSVM. Se comienza con la construcción de una función de costo, el FSVM también quiere maximizar el margen de separación y minimizar el error de clasificación tal que una buena capacidad de generalización puede ser ejecutada.

La construcción del problema de optimización del FSVM se formula como

$$\begin{aligned} \min \Phi(w, \xi, \mu) &= \frac{1}{2} w^T w + C \sum_{i=1}^n \mu_i^m \xi_i \\ \text{sujeto } & y_i (w^T x_i + b) \geq 1 - \xi_i, \quad i = 1, 2, \dots, n \\ & \xi_i \geq 0, \quad i = 1, 2, \dots, n \end{aligned} \quad (12)$$

donde  $m$  influye defusificación de la falta del término defusificado en la función de costo. Ahora la función de Lagrange es

$$\begin{aligned} Q(w, b, \xi, \alpha, \beta, \mu) &= \frac{1}{2} w^T w + C \sum_{i=1}^n \mu_i^m \xi_i \\ &- \sum_{i=1}^n \alpha_i [y_i (w^T x_i + b) - 1 + \xi_i] - \sum_{i=1}^n \beta_i \xi_i \end{aligned} \quad (13)$$

donde  $\alpha_i$  y  $\beta_i$  son multiplicadores de Lagrange no negativos. Diferenciando  $Q$  respecto a  $w$ ,  $\beta$  y  $\xi_i$ , e igualando los resultados a cero se obtienen las tres condiciones de optimalidad siguientes

$$\begin{aligned}\frac{\partial Q(w, b, \xi, \alpha, \beta, \mu)}{\partial w} &= w - \sum_{i=1}^n \alpha_i y_i x_i = 0 \\ \frac{\partial Q(w, b, \xi, \alpha, \beta, \mu)}{\partial b} &= \sum_{i=1}^n \alpha_i y_i = 0 \\ \frac{\partial Q(w, b, \xi, \alpha, \beta, \mu)}{\partial \xi} &= C \mu_i^m - \alpha_i - \beta_i = 0\end{aligned}\quad (14)$$

Cada grupo tiene un modelo difuso. Usando la idea del modelo Takagi-Sugeno [8], se pueden combinar los modelos locales en cada grupo dentro de un modelo global. Las reglas difusas tienen la siguiente forma

$$\begin{aligned}R^j : IF \ x_{i_1} \leq x_1 \leq x_{i_2} \text{ and } \dots \ x_{i_n} \leq x_n \leq x_{i'_2} \\ THEN \ y(k) = f_i [X(k)]\end{aligned}$$

donde  $j=1 \dots p$ ,  $p$  es el número del grupo del agrupamiento en línea. Las funciones de membresía para  $x_i$ . El modelo difuso final es

$$\hat{y} = \frac{\sum_{i=1}^p f_i [X(k)] \left[ \prod_{j=1}^n (\mu_{A_j^i}) \right]}{\sum_{i=1}^p \left[ \prod_{j=1}^n (\mu_{A_j^i}) \right]} \quad (15)$$

El siguiente teorema ofrece un algoritmo de gradiente descendente estable para el modelado neuro difuso.

**Teorema 1** Si se usa el sistema difuso para identificar la planta no lineal (1) en el grupo  $j$ , el siguiente algoritmo de gradiente descendente con un rango de aprendizaje variante en el tiempo puede identificar el error  $e(k)$  acotado

$$W(k+1) = W(k) - \eta_k e(k) \Phi^T [X(k)] \quad (16)$$

donde  $\eta_k = \eta / (1 + \|\Phi [X(k)]\|^2)$ ,  $0 < \eta \leq 1$ . El error de identificación normalizado

$$e_N(k) = \frac{e(k)}{1 + \max_k \left( \|\Phi [X(k)]\|^2 \right)} \quad (17)$$

satisface la siguiente realización promedio

$$\limsup_{T \rightarrow \infty} \frac{1}{T} \sum_{k=1}^T \|e_N(k)\|^2 \leq \bar{\mu} \quad (18)$$

donde  $\bar{\mu} = \max_k \{ \|\mu\|^2 \}$ .

**Observación 1** En general, un sistema difuso no puede igualar cualquier sistema no lineal exactamente. Los parámetros del sistema difuso no convergerán a sus valores óptimos. La idea de identificación en línea es forzar la salida del sistema difuso a seguir la salida de la planta. Aunque los parámetros no pueden converger a sus valores óptimos, (18) muestra que la normalización del error de identificación convergerá a la bola de radio  $\bar{\mu}$ . Si el sistema difuso puede igualar la planta no lineal (4) exactamente ( $\mu(k)=0$ ), i.e. se puede encontrar la mejor función de membresía  $\mu_{A_j}$  y  $W^*$  tal que el sistema no lineal se pueda escribir como  $y(k) = W^* \Phi[\mu_{A_j}]$

Puesto que  $\|e(k)\| > 0$ , la misma ley de aprendizaje (16) hará asintóticamente estable el error de identificación  $\|e(k)\| > 0$ .

**Observación 2** Normalizando el rango de aprendizaje en (16) que es variante en el tiempo en orden para asegurar la estabilidad del error de identificación. Estos rangos de aprendizaje son más fácil de decidir que en [8] (por ejemplo seleccionando  $\eta=1$ ), sin necesidad de cualquier información a priori. Los rangos de aprendizaje variantes en el tiempo pueden ser encontrados en un mismo esquema estándar adaptivo [3]. Pero estos necesitan modificaciones robustas para garantizar estabilidad de la identificación. El algoritmo esta derivado del análisis de estabilidad (o función ISS-Lyapunov), el algoritmo de [5] fue obtenido de la minimización de la función de costo. El enfoque a la cota del error de identificación, [5] enfocado al análisis de convergencia. Es interesante ver que dos métodos diferentes pueden obtener resultados similares.

Ahora el caso de las funciones de membresía entrenadas de consecuencia y premisa. Las condiciones iniciales son  $c_{ji}(1) = x_i^*$ ,  $w_j(1) = y_i^*$ ,  $\sigma_{ji}(1)$  es aleatorio en (0; 1). Puesto que las funciones de membresía son funciones Gaussianas, la salida del sistema difuso se puede expresar como

$$\hat{y} = \frac{\sum_{i=1}^{sv_j} w_i \left[ \prod_{j=1}^n \exp \left( -\frac{(x_j - c_{ji})^2}{\sigma_{ji}^2} \right) \right]}{\sum_{i=1}^{sv_j} \left[ \prod_{j=1}^n \exp \left( -\frac{(x_j - c_{ji})^2}{\sigma_{ji}^2} \right) \right]} \quad (19)$$

**Teorema 2** Para el modelo neuro difuso tipo TSK, se selecciona  $A_{ji}$  como funciones Gaussianas. La salida del sistema lógico difuso se puede expresar como

$$\hat{y} = \frac{\sum_{i=1}^{sv_j} \left( \sum_{k=0}^n p_k^i x_k \right) \left[ \prod_{j=1}^n \exp \left( -\frac{(x_j - c_{ji}^*)^2}{\sigma_{ji}^{*2}} \right) \right]}{\sum_{i=1}^{sv_j} \left[ \prod_{j=1}^n \exp \left( -\frac{(x_j - c_{ji}^*)^2}{\sigma_{ji}^{*2}} \right) \right]} \quad (20)$$

donde  $c_{ji}^*$  y  $\sigma_{ji}^{*2}$  son parámetros desconocidos los cuales minimizan la dinámica no modelada  $\mu$ , y  $x_0 = 1$ .

#### 4. Simulaciones

Se considera la función no lineal la cual tiene la forma [7]

$$f(x_1, x_2) = 0.52 + 0.1x_1 + 0.28x_2 - 0.6x_1x_2$$

Los datos de entrenamiento son seleccionados como  $x_1(k) = -1 + ((2k)/T)$ ,  $x_2(k) = 1 - ((2k)/T)$ ,  $k = 1, 2, \dots, T$ . Se escoge  $\alpha = 0.4$ ,  $\beta = 0.6$ . Se observa que los cambios máximos en la entrada y la salida son aproximadamente 1 y 2, así  $\alpha \|x_{max} - x_{min}\| + \|y_{max} - y_{min}\| \beta = 16$ ,  $L$  se podría escoger como  $L < 1.2$ , en esta aplicación se selecciona  $L = 0.5$ . El particionamiento entrada-salida se muestra en la Figura (3). Se representa el centro como "o" de cada grupo, y "+" es la frontera entre los grupos. Se puede ver que hay 7 grupos. Por ejemplo, los intervalos de tiempo del 4th grupo son  $l_1^2 = 94$ ,  $l_2^2 = 238$ .

Las funciones de membresía entrenadas para cada grupo, son 7 funciones de membresía para  $x_1$  se muestran en la Figura (4). Entonces los parámetros fijados de las funciones de membresía, se usan otros 300 datos para la prueba del modelo. El dato probado es  $x_1(k) = 1 - ((2k)/T)$ ,  $x_2(k) = -1 + ((2k)/T)$ ,  $k = 1, 2, \dots, T$ . El resultado final del modelado después de los modelos locales ( $p=7$ , Figura 5) se muestra en la Figura (6).

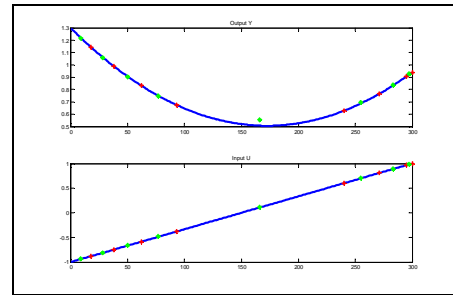


Figura 3. Agrupamiento en línea.

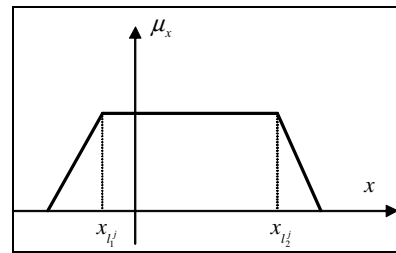


Figura 4. Función de membresía de  $x_1$ .

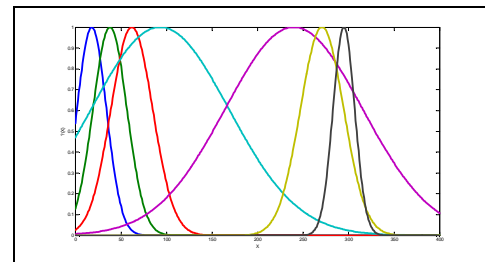


Figura 5. Funciones de Membresía para 7 reglas usando kernel difuso.

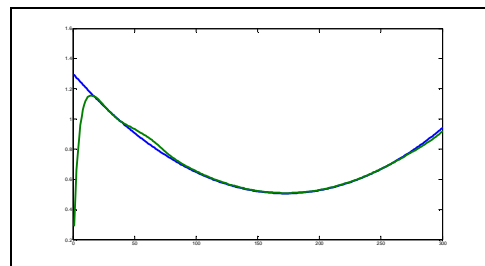


Figura 6. Resultado final.

#### 5. Agradecimiento

Se agradece el apoyo dado por el proyecto SIP con No. 20113187 cuyo título es: "Diseño de algoritmos y sistemas de control inteligente, aplicado a plantas no lineales inmersos en entornos dinámicos", así como a COFAA del Instituto Politécnico Nacional, ESIME Zacatenco.

## 6. Conclusión

En este trabajo se propone un eficiente enfoque para modelado de sistemas no lineales usando FSVM junto con la extracción de reglas. Algunas técnicas son utilizadas para este nuevo enfoque. Primero se propone un agrupamiento en línea la cual divide el espacio de datos de entrada/salida en grupos en el mismo intervalo de tiempo. Después se utiliza el enfoque de FSVM para obtener los support vectors en cada grupo. Con estos support vectors se construyen las reglas difusas para obtener el correspondiente sistema difuso. Posteriormente la estructura de identificación, en el intervalo de tiempo variante es aplicada para obtener los parámetros de identificación. Las contribuciones de este trabajo son: 1) el método de agrupamiento en línea y el enfoque de FSVM son usados para la extracción de las reglas, 2) la cota superior del error de modelado y la estabilidad son probados para el modelado difuso.

## Referencias

- [1] Angelov P., *An approach for fuzzy rule-base adaptation using online clustering*, International Journal of Approximate Reasoning, Vol.35, No.3,275-289, 2004.
- [2] Cristianini N. and Shawe-Taylor J., *An Introduction to Support Vector Machines* Cambridge Univ. Press, 2000.
- [3] Ioannou P.A. and Sun J., *Robust Adaptive Control*. Upper Saddle River, NJ: Prentice-Hall, 1996.
- [4] Juang C.F., *Combination of on-line clustering and Q-value based GA for reinforcement fuzzy system desing*, IEEE Trans. Fuzzy Syst., Vol. 13, No.3, 289-302, 2005.
- [5] Mandic D.P., Hanna A.I. and Razaz M., *A normalized gradient adaptive step size*, IEEE Signal Processing Lett., vol 8, pp. 295-297, Feb 2001.
- [6] Takagi T. and Sugeno M., *Fuzzy identification of systems and its applications to medeling and control*, IEEE Trans. Syst., Man. and Cybern., vol. 1, pp. 116-132, Jan. 1985.
- [7] Wang L.X., *Adaptive Fuzzy Systems and Control*, Englewood Cliffs NJ: Prentice-Hall, 1994.
- [8] Wang C.H., Liu H.L. and Lin C.T., *Dynamic optimal learning rates of a certain class of fuzzy neural networks and its applications with genetic algorithm*, IEEE Trans. Syst., Man, Cybern. B, Vol.31, 467-475, 2001.