

Modelado Lingüístico con 2-tuplas en el Diseño de Controladores Borrosos

José I. Peláez¹, Jesús M. Doña¹, David L. La Red²

¹ Dpto. de Lenguajes y Ciencias de la Computación
Universidad de Málaga. Málaga. España

² Dpto. Informática
Universidad Nacional del Nordeste. Corrientes. Argentina
{jignacio, [jmdona](mailto:jmdona@lcc.uma.es)}@lcc.uma.es
lmdavid@exa.unne.edu.ar

Resumen. Un factor crítico en el diseño de controladores borrosos es el tiempo de respuesta, el cual se halla directamente relacionado con el proceso de desfuzzificación. Este proceso consiste básicamente en determinar qué regla se aplicará al controlador. El modelo de representación lingüística de 2-tuplas es un modelo que realiza la representación lingüística mediante dos valores. El propósito de este trabajo es mostrar cómo dicha representación facilita el diseño de los controladores al simplificar el proceso de desfuzzificación.

Palabras clave: controladores borrosos, desfuzzificación, 2-tuplas.

Resumen Ampliado

Un controlador borroso está formado por cuatro módulos. El primero es el módulo fuzzyficador, el segundo es la base de conocimiento, el tercero el motor de inferencia, y finalmente, el módulo desfuzzyficador. En la figura 1 se muestra el esquema de un controlador borroso.

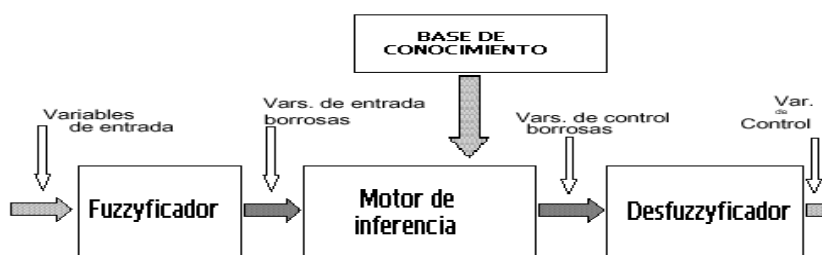


Fig. 1. Esquema de un Controlador Borroso.

El módulo fuzzyficador es el encargado de convertir los valores de naturaleza discreta en variables borrosas.

El segundo módulo es la denominada base de conocimiento, que está formada por un conjunto de reglas que describen el comportamiento del sistema. Las reglas tienen la siguiente estructura:

“SI V1 ES valor1 AND V2 ES valor2... ENTONCES Vsalida ES ValorSalida”

El tercer módulo es el denominado motor de inferencia, el cual realiza una exploración de la base de conocimiento activando diferentes reglas en función de la entrada. Como salida el motor de inferencia proporciona una función que es resultado de la composición de las funciones que cada regla genera.

Y finalmente, el módulo desfuzzyficador, cuya misión es la de convertir una variable de tipo borroso o difusa a una variable discreta para que pueda ser aplicada. Para realizar este proceso existen varios métodos. A continuación mostramos los más comunes:

- Método COA. Este método determina el área de la función a realizar. Para ello utiliza la siguiente expresión:

$$Z^* = \frac{\sum_{j=1}^q z_j \mu_c(z_j)}{\sum_{j=1}^q \mu_c(z_j)} \quad (1)$$

donde q es el rango de la función de salida; z_j la suma de las salidas de control en j ; y $\mu_c(z_j)$ la función de pertenencia en c .

- Método MON. Este método realiza una media de los valores máximos. Para ello utiliza la siguiente expresión:

$$Z^* = \sum_{j=1}^l \frac{z_j}{l} \quad (2)$$

donde l es el número de valores que alcanzan el máximo; y z_j el punto en el que se alcanza el máximo.

El diseño de un controlador borroso supone llegar a un compromiso entre diferentes factores como son la velocidad, precisión y flexibilidad [1]. La velocidad de respuesta está limitada por otros factores como el grado de precisión deseado o la flexibilidad del diseño.

En este trabajo se presenta una forma de aumentar tanto la velocidad de respuesta como la precisión utilizando el modelado lingüístico basado en 2-tuplas.

Muchas veces es necesario realizar operaciones con etiquetas lingüísticas, agregación, comparación, etc., por lo que es necesario definir técnicas computacionales que nos permitan realizar dichas operaciones. Existen distintos modelos con los que poder realizar una representación lingüística y operar con variables lingüísticas: (i) el modelo basado en el Principio de Extensión, (ii) el modelo simbólico, y (iii) el modelo basado en 2-tuplas.

A continuación se realiza un breve repaso de los tres modelos:

1. *Modelo lingüístico basados en el Principio de Extensión.* Este modelo está basado en el Principio de Extensión de Zadeh, que puede aumentar la imprecisión de los resultados porque los resultados obtenidos por los operadores borrosos no coincidan con ningún término del conjunto inicial, por lo que será necesario un proceso de aproximación lingüístico [2], [3], para expresar los resultados en el dominio de expresión original.

2. *Modelo simbólico.* Este modelo no usa las funciones de pertenencia asociadas a las etiquetas para realizar el cómputo, sino que usa el orden que ocupa cada etiqueta lingüística en su conjunto de términos. Por lo general se utiliza la estructura ordenada del conjunto de términos lingüísticos, $P = \{p_0, \dots, p_g\}$ donde $p_i < p_j$ si $i < j$, para realizar los cómputos [4].
3. *Modelo de representación borroso lingüístico 2-tuplas.* Este modelo se basa en el concepto de Traslación Simbólica [5], [6]. Utiliza como base para la representación de la información lingüística un par de valores o una 2-tupla (p, α) , donde p es un término lingüístico y α es un valor numérico que representa el valor de la traslación simbólica [7], [8], [9].

Sea $S = \{s_0, \dots, s_g\}$ un conjunto de términos lingüísticos, y $\beta \in [0, g]$ un valor en el intervalo de granularidad de S . La representación de información lingüística basada en 2-tuplas será un par de valores (r_i, α) donde $r_i \in S$ y $\alpha_i \in [-0.5, 0.5]$. Una vez pasada la información a 2-tuplas tendremos la información representada en una tupla (r_i, α) siendo i el índice de la etiqueta más cercana al valor β que representa la información de todos los conjuntos difusos sin pérdida de información, y α_i la traslación simbólica respecto de dicha etiqueta.

Para realizar el modelado de un controlador borroso utilizando la representación lingüística de dos tuplas vamos a mostrar cómo afectaría dicha representación a los diferentes módulos que componen el controlador.

1. *Módulo Fuzzyficador.* En este módulo las entradas del sistema que son de tipo discreto se convertirán en variables borrosas. Con la representación del modelo de 2-tuplas se consigue, sin pérdida de información, que cada variable sólo pertenezca a una única etiqueta lingüística con su correspondiente desplazamiento relativo, mientras que en los modelos tradicionales cada variable pertenecía a varios conjuntos borrosos.
2. *Búsqueda en la base de conocimientos.* En este módulo el motor de inferencia produce la activación de una única regla, ya que cada variable está representada mediante el modelo de 2-tupla con lo que pertenece a un único conjunto difuso. El valor de salida será una tupla (r_i, α) , siendo r_i el conjunto borroso al que pertenece la salida y α_i la media ponderada de los desplazamientos de las variables.
3. *Módulo Defuzzyficador.* En este módulo la salida borrosa se convertirá a una salida discreta. Con el modelo de 2-tuplas esta operación es directa, ya que lo único que se tiene que calcular es el valor discreto correspondiente al desplazamiento relativo de la etiqueta.

Para mostrar la utilización del modelo de 2-tuplas se ha implementado un controlador borroso para el problema del péndulo invertido. Además se han implementado varios controladores borrosos con diferente número de reglas, las cuales han variado desde 49 hasta 512 [10].

Se ha mostrado cómo se puede mejorar el tiempo de respuesta de los controladores borrosos utilizando la representación lingüística de 2-tuplas. Esta mejora en la

velocidad se ha producido por una disminución del coste computacional en el motor de inferencia y el módulo de desfuzzyficación.

La mejora en el motor de inferencia está motivada porque el controlador realiza un único acceso a la base de conocimiento cuando se utiliza la representación de 2-tuplas, mientras que con la representación tradicional son necesarios varios accesos.

La mejora en el módulo desfuzzyficador está motivada porque dicho proceso con la representación de 2-tuplas es directo ya que solamente se activa una regla, mientras que con la representación tradicional dicho proceso activa varias reglas, las cuales tiene que componer posteriormente para determinar la respuesta del sistema.

Finalmente, se ha realizado una comparación de los métodos con diferente número de reglas. Estos resultados han mostrado en primer lugar una disminución del tiempo de respuesta, y en segundo lugar, una estabilización más suave cuando se utiliza este nuevo tipo de representación.

Referencias

1. Sanz, A.: Lógica borrosa: Incidencias en las aplicaciones industriales. Sesión Invitada al Seminario Anual de Automática y Electrónica Industrial, pp. 36-42, Zaragoza, España (1996).
2. Bonissone, P. P., Decker, K. S.: Selecting uncertainty calculi and granularity: An experiment in Trading-off precision and complexity, in: L. H. Kanal and J. F. Lemmer. Eds. *Uncertainty in Artificial Intelligence*, pp. 217-247, North-Holland (1986).
3. Degani R., Bortolan, G.: The problem of linguistic approximation in clinical decision making, *Int. Journal of Approximate Reasoning* 2, pp. 143-162 (1988).
4. Delgado, M., Verdegay, J. L., Vila, A.: On aggregation operations of linguistic labels, *Int. Journal of Intelligent Systems* 8, pp. 351-370, (1993).
5. Herrera, F., Herrera-Viedma, E., Martínez, L. A.: Fusion approach for managing multi-granularity linguistic term sets in decision making, *Fuzzy Sets and Systems*, Vol. 114, pp. 43-58 (2000).
6. Herrera, F., Martínez, L.: A 2-tuple fuzzy linguistic representation on model for computing with words. *IEEE Transactions on fuzzy systems* 8:6, pp. 746-752 (2000).
7. Herrera, F., Martínez, L.: A Fusion Method for Multi-Granularity the 2-tuple Fuzzy Linguistic Representation Model, Technical Report #DECSAI-99107 (1999).
8. Tang, Y., Zheng, J.: Linguistic modelling based on semantic similarity relation among linguistic labels. *Fuzzy Sets and Systems*, 157(12): 1662-1673 (2006).
9. Herrera, F., Martínez, L.: The 2-tuple linguistic computational model. Advantages of its linguistic description, accuracy and consistency. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*. 9(Suppl.): 33-49 (2001).
10. Sivanandam, S.N., Sumathi, S., Deepa S.N.: *Introduction to Fuzzy Logic using MATLAB*. Ed. Springer, Berlin, New York (2007).