



**Editorial de la Universidad
Tecnológica Nacional**

FUZZY CONTROL

Ing. Juan Carlos Gómez

Grupo de Inteligencia Artificial y Robótica
Secretaría de Ciencia y Tecnología
Universidad Tecnológica Nacional – FRBA

Instituto Nacional de Tecnología Industrial – Electrónica

2008

FUZZY CONTROL

Ing. Juan Carlos Gómez

Grupo de Inteligencia Artificial y Robótica
Secretaría de Ciencia y Tecnología
Universidad Tecnológica Nacional – FRBA
Instituto Nacional de Tecnología Industrial – Electrónica
e-mail: juanca@inti.gob.ar

- **Introducción.**

Sea por aceptación, curiosidad o desconfianza, uno de los temas que está en la boca de las personas dedicadas al control es el de *fuzzy control*. De cualquier forma, a nadie escapa la importancia y la actualidad del mismo. Esto se observa en los artículos de las revistas especializadas cuando se lee sobre sus cualidades para desarrollar sistemas de control robustos, sencillos, económicos y de rápida implementación; cuando aparecen avisos publicitando cámaras de video *fuzzy*, lavarropas que emplean *fuzzy logic* para determinar las condiciones de lavado, sistemas ABS con *fuzzy logic* para automóviles, sistemas de aire acondicionado, subterráneos con controles *fuzzy*, etc, o que gran cantidad de compañías japonesas introducen controles *fuzzy* en sus nuevos productos, y que el furor está llegando a los EE.UU. donde se espera un volumen de mercado de 10.000 millones de dólares para 1998.

- **Sistemas basados en reglas.**

El desarrollo del *fuzzy control* tiene, su origen en los *sistemas basados en reglas* para la toma de decisiones y en la *fuzzy logic* para la evaluación de aquellas.

Los controladores *fuzzy*, al igual que otros, toman los valores de las variables de entrada, realizan algún procedimiento con ellas, deciden como modificar las variables de salida y lo realizan, afectando estas últimas a la planta. La diferencia esencial es que no procesan ecuaciones ni tablas, procesan reglas para decidir como cambiar las salidas. Esto lo hace una tecnología muy accesible, ya que las reglas están más cerca del sentido común que de las ecuaciones y los modelos.

Este método de procesamiento de reglas está vinculado con los sistemas expertos, uno de los frutos tangibles de la Inteligencia Artificial (IA). Estos sistemas nacieron como consecuencia natural del desarrollo de esta disciplina, ya que era necesario agregar a los programas ese *conocimiento y experiencia*, que el experto humano tiene para resolver un determinado problema. Estos programas de computadora, utilizan *conocimiento* y algún tipo de *mecanismo de inferencia* para resolver un tipo de problema en especial, con el objetivo de obtener resultados parecidos a los del experto humano, no sólo en cuanto a la conclusión final, sino también a la forma en que se llega a la misma.

Los problemas atacados con esta técnica son entre otros, los de diagnóstico médico, path planning, controladores de tráfico, acceso inteligente a bases de datos.

Este *conocimiento* consta de *hechos y heurística*, siendo los primeros la información pública y generalmente disponible, para la resolución de un problema en particular, mientras que la *heurística*, es más privativa del experto, es justamente ese conocimiento en particular que lo hace destacarse en su tarea. Cuando este *conocimiento* está expresado como reglas del tipo

SI antecedente ENTONCES consecuente o **SI condición ENTONCES acción**

entonces se dice que es un *sistema basado en reglas*.

En cuanto al *mecanismo de inferencia*, es la parte de control del programa, encargada de procesar las reglas contenidas en la *base de conocimiento*, tomando en cuenta los nuevos datos y el estado del sistema, y producir como resultado, un nuevo cambio en el estado del mismo para así comenzar un nuevo ciclo.

En general la evaluación de las reglas se realiza utilizando lógica aristotélica, la cual es muy “dura” y por lo tanto es a veces difícil representar el conocimiento del experto mediante este paradigma.

Para ello basta con tratar de describir con un conjunto de reglas, el conocimiento necesario para realizar una tarea sencilla y cotidiana como por ejemplo, la de estacionar un vehículo. Es en este punto donde comienza a cobrar importancia la *fuzzy logic*, ya que enseguida trataremos de esbozar reglas con calificativos no demasiado precisos, tales como, si el móvil está *muy cerca* del cordón y entonces , o si el *ángulo* es *pequeño* y entonces , o si entonces *girar un poco* a la izquierda, etc.

• Fuzzy logic.

A mediados de la década del 60 el Profesor Lotfi Zadeh, de la Universidad de California en Berkeley, introdujo su teoría de los conjuntos *fuzzy* , donde la idea de pertenencia de un elemento a un determinado conjunto no era ya *pertenece o no pertenece*, sino que *pertenece* en un cierto grado. Las funciones que vinculan al elemento con su grado de pertenencia, se llaman *membership functions*.

Basandose sobre esta idea, se construyó nuevamente toda la teoría de conjuntos, redefiniendo *inclusión, unión, operadores lógicos (AND, OR)*, otras operaciones y sus propiedades.

En lo referente a la lógica, aparecen nuevas alternativas respecto de la lógica tradicional, puesto que, al evaluar una proposición, ya no es *verdadera o falsa* sino que la misma tiene un cierto grado de veracidad, y que a su vez depende del grado de veracidad de sus premisas. De aquí que al utilizar *fuzzy logic* en la evaluación de reglas, todas las reglas cuyas premisas tengan algún grado de veracidad, influirán también en cierto grado a la solución del problema. Es decir toda regla que sepa algo opina.

• Controladores fuzzy.

Los controladores *fuzzy*, igual que otros controladores, toman el valor de las variables de entrada, procesan, y actúan sobre sus salidas a fin de controlar la planta.

En *fuzzy control* se utilizan sistemas basados en reglas, que emplean *fuzzy logic*, como elemento central.

Dado su estado, y los valores de las variables de entrada, estos sistemas evalúan la veracidad de cada regla, y así, toman decisión sobre los cambios a realizar en las variables de salida. Una vez actualizadas las mismas, estas producirán un cambio sobre la planta, luego se vuelven a obtener los valores de las variables de entrada, comenzando un nuevo ciclo.

Al utilizar este tipo de resolución de problemas, es importante observar que lo que se está procesando no son ya ecuaciones, números o tablas crudas sino, reglas, es decir se procesan cosas tales como:

“**SI** la temperatura es baja **ENTONCES** aumente el ciclo de actividad del calefactor”

Tal tipo de procesamiento está mucho mas cercano al razonamiento del experto humano, ya que permite realizar operaciones entre palabras como *mas bajo, subir un poquito* etc.

Este proceso cíclico puede ser dividido en tres grandes partes, *fuzzification, evaluación de reglas y defuzzification*.

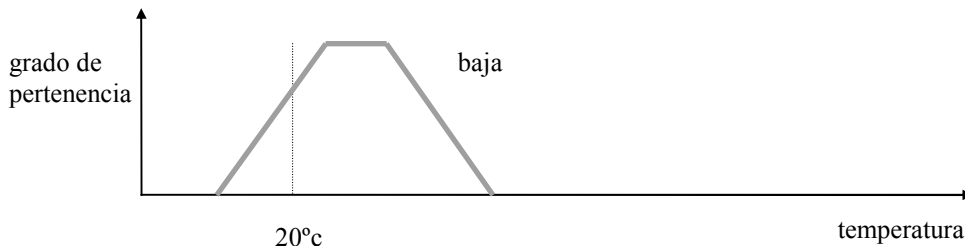
• **Fuzzification.**

Esta es la primera parte del proceso, donde luego del procedimiento convencional de adquirir los valores de las variables del sistema y calcular otras, se procede a calcular el grado de pertenencia de tales variables a todos los posibles conjuntos *fuzzy* (*fuzzy inputs*) que se le han asignado.

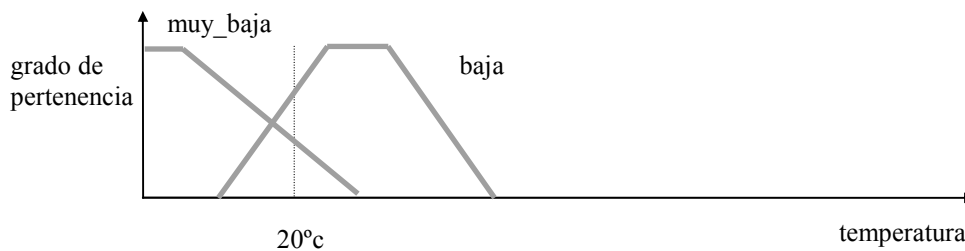
Tomando como ejemplo otra vez la variable temperatura, con un valor de 20°C, se cuantifica su grado de pertenencia a los conjuntos representados con las siguientes etiquetas lingüísticas, *muy_baja*, *baja*, *templada*, *alta* y *muy_alta*. Para ello se debe haber definido, para cada una de las etiquetas, una *membership function*, que define qué valores de la variable temperatura les pertenecen y con qué grado.

Son las *membership functions* las que permiten cuantificar los conceptos que representan las etiquetas.

En el gráfico se puede observar la *membership function* para la *fuzzy input baja*, y el valor de pertenencia para 20°C.



Si ahora se realiza el mismo procedimiento para *muy_baja*, se verá que también, para 20°C, es parcialmente *muy_baja*.



Al finalizar este proceso, queda asignado a cada *fuzzy input*, el grado de pertenencia del valor actual de la variable que le corresponde.

• **Evaluación de reglas.**

Una vez realizada la *fuzzification*, se está en condiciones de evaluar los antecedentes de las reglas, obteniendo el grado de verdad o “peso”, para cada una de ellas. Si se tiene una regla como la siguiente:

SI la temperatura es *baja* **ENTONCES** aumente el ciclo de actividad del calefactor.

El peso de la regla estará dado por la veracidad de su antecedentes.

Se asigna directamente como peso, el grado de pertenencia del valor leído de temperatura a la etiqueta lingüística *baja*.

En el caso de antecedentes de reglas con conectivos lógicos **Y**, como por ejemplo:

SI *la temperatura es baja* **Y** *el cambio de temperatura es ligeramente creciente*
ENTONCES *aumente el ciclo de actividad del calefactor.*

La regla será tan verdadera como lo sea el menos verdadero de sus antecedentes. Es decir, se le asigna a la regla como peso, el menor de los grados de pertenencia de las variables de los antecedentes a las respectivas etiquetas lingüísticas.

Este proceso se realiza para cada una de las reglas del sistema, quedando luego de este ciclo cada regla con su peso correspondiente.

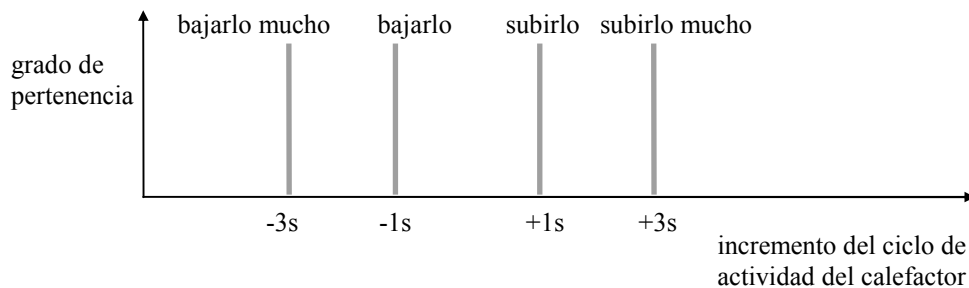
Ahora se trabaja con los consecuentes de las mismas.

Así como para las variables de entrada, a cada variable de salida le corresponde un grupo de *membership functions*, representadas con etiquetas lingüísticas. Cada una de ellas es una *fuzzy output*. (*subirlo mucho, subirlo, bajarlo, bajarlo mucho*).

A cada una de las *fuzzy outputs* se le asigna como valor, o *grado de aplicabilidad*, el máximo valor entre todas las reglas que la mencionan, de manera que al finalizar toda la evaluación, queda cada *fuzzy output* con su valor.

- **Defuzzification.**

Para el ejemplo se tomará la forma más simple de *membership function*, esta es la que considera a un solo elemento del universo de discurso perteneciente totalmente al conjunto. Este tipo de funciones se denominan *singleton*.



A cada *fuzzy output*, se le asignó el máximo valor de verdad o peso de las reglas que la mencionan como consecuente.

Ahora quedan varias *fuzzy outputs* para cada variable de salida, cada una con su valor de verdad, o *grado de aplicabilidad*, pero ¿cuál es el nuevo valor de la variable de salida?. Una forma simple y efectiva de determinarlo es realizando un promedio ponderado entre todas las *fuzzy outputs* que le corresponden a una variable de salida.

Para el caso en que las *membership functions* no sean *singletons*, se procede en forma similar, pero considerando las superficies de las *fuzzy outputs*, afectadas por su *valor de aplicabilidad*, y tomando el centro de gravedad de las mismas como el nuevo valor de la variable de salida en consideración.

Una vez actualizada la variable de salida, se está en condiciones de comenzar un nuevo ciclo.

Conclusiones.

Se ha realizado una breve descripción de los *fuzzy controllers*, sistemas de control cuyo funcionamiento está basado en la evaluación de reglas utilizando *fuzzy logic*.

Se pudo apreciar la sencillez en la implementación del sistema, ya que al describir el comportamiento del controlador mediante reglas, no es necesario realizar una modelización exacta, sino que basta con entender conceptualmente su comportamiento, contar con un modelo muy simplificado o con un operador experimentado que pueda describir cómo realiza la tarea de control manualmente. Esta característica lo hace particularmente apropiado para sistemas muy alineales, cuya modelización por los métodos tradicionales suele ser compleja, acelerando la puesta en el mercado del producto (time to market).

Se comprobó la robustez de los *fuzzy controllers*, ya que es posible describir reglas para todo el rango de variación de las variables de entrada, obteniéndose controles que no necesitan un posicionamiento manual en el arranque, proceso utilizado cuando los parámetros del sistema son muy diferentes al comienzo, respecto de los que tiene en el punto de funcionamiento.

Finalmente se observa, que mediante el método de evaluación de reglas, resultan acciones de control suaves, una de las principales características de los *fuzzy controllers*.

• Referencias.

Creating Fuzzy Micros. James M. Sibigroth. Embedded Systems Programming, Dec. 91.

Fuzzy Logic for Embedded Microcontrollers. J. Sibigroth. Circuit Cellar INK. March 95.

Fuzzy Logic Finally Gains Acceptance in the U.S. Richard Nass. Electronic Design. June 92.

Fuzzy Logic. Lotfi A. Zadeh. IEEE Computer. April 88.

Software that can Dethrone Computer Tyranny. Larry Armstrong. Business Week. April 92.

Generating Fuzzy Rules by Learning from Examples. Li-Xin Wang. IEEE Transactions on Systems, Man and Cybernetics. Nov 92.

Adaptive Fuzzy Systems. Earl Cox. IEEE Spectrum. Feb 93.

Fuzzy Control and introduction to Home Appliances. Techno Japan. September 91.

Fuzzy Set Theory and Its Applications. H.J.Zimmermann. Kluwer-Nijhoff Publishing.

Fuzzy Mathematical Techniques with Applications. Abraham Kandel. Florida State University. Addison-Wesley Publishing Company.

Artificial Intelligence, Expert Systems. William B. Gevarter. Noyes Publications.