

TUTORIAL DE LÓGICA FUZZY

Ing.MSc. Guillermo Tejada Muñoz

g270031@unmsm.edu.pe

Facultad de Ingeniería Electrónica de la Universidad Nacional Mayor de San Marcos de Lima -Perú

Resumen: El presente artículo pretende difundir los conocimientos básicos sobre la lógica *fuzzy*, denominada por algunos con los nombres de lógica borrosa, difusa o nebulosa. En este artículo se prefiere mantener su nombre original en inglés. También es el propósito del presente artículo difundir las aplicaciones que con esta tecnología se vienen realizando en distintas partes del mundo y las líneas de investigación que pueden ser exploradas.

Abstrac: This paper try to diffuse the basic knowledge on the fuzzy logic, denominated by some with the names of *lógica borrosa, difusa o nebulosa*. In this paper, it prefers to maintain its original name in English. It is also our purpose to diffuse the applications that it has been done, with this technology, in different places of the world, and the research lines that can be explored.

Palabras Claves: Lógica, Logic, Nebulosa, Difusa, Fuzzy, Fuzzyfication, Defuzzification, Incerteza, Aplicaciones.

I. DEFINICIÓN DE LÓGICA FUZZY

La lógica Booleana es conocida como la más precisa de todas las ciencias y disciplinas teóricas. La mayoría de las ciencias modernas y matemáticas se basan en sus principios. A pesar de las ventajas de su exactitud, la lógica Booleana tiene la desventaja de no poder reproducir los patrones del pensamiento humano.

Es así, como a mediados de los años sesenta, el profesor Lotfi Zadeh de la universidad de California en Berkeley, pretendiendo suplir esta deficiencia de la lógica tradicional, crea la que hoy se denomina como lógica *fuzzy*.

Como disciplina teórica matemática, la lógica *fuzzy* está diseñada para reaccionar a cambios continuos de la variable a ser controlada y se diferencia con la lógica Booleana por no estar restringida a dos únicos valores de 0 y 1. En su lugar permite valores parciales y multivalores de verdad. Se puede afirmar, tal como lo demostró Bart Kosko, que la lógica Booleana es un caso especial de la lógica *fuzzy*. Esta disciplina es especialmente ventajosa para problemas que no puedan ser fácilmente representadas por modelos matemáticos debido a que los datos están incompletos o por que el proceso es muy complejo.

El lenguaje del mundo real usado en control *fuzzy* permite a los programadores incorporar la lógica ambigua de los humanos dentro de la computadora. El uso de modelos lingüísticos en lugar de modelos matemáticos mejora grandemente la transparencia del sistema y facilita las potenciales modificaciones. La lógica *fuzzy* intenta controlar procesos, capturando el conocimiento que los especialistas poseen de su experiencia real, sin tener que modelar el sistema.

II. TIPOS DE INCERTEZA

Esta sección presenta los principios de la lógica *fuzzy* y es básica para entender el mecanismo por el cual trabaja el sistema de lógica *fuzzy*.

Muchas disciplinas matemáticas tratan con la descripción de incerteza, tales como la teoría de la probabilidad, la teoría de la información y la teoría del conjunto *fuzzy*. Es más conveniente clasificarlos por el tipo de incerteza que tratan. A continuación vamos a considerar sólo dos tipos de incerteza: Estocástica y Léxica.

2.1 Incerteza Estocástica

La incerteza estocástica trata con la incerteza hacia la ocurrencia de un cierto evento. El siguiente es un ejemplo de este caso:

La probabilidad de dar en el blanco es 0.8

El evento en sí mismo, *dar en el blanco*, está bien definido. La incerteza surge como consecuencia de que sí el blanco será o no alcanzado. Esta incerteza es cuantificado por un grado de probabilidad. En el caso que se analiza, la probabilidad es de 0.8. Combinación de oraciones similares podrían ser procesadas usando métodos estocásticos, tales como el cálculo de Bayes para probabilidad condicional. [VON ALTRROCK, 1995]

2.2 Incerteza Léxica o Imprecisión

Un tipo diferente de incerteza es aquel basado en el lenguaje de comunicación de los humanos, también denominado incerteza léxica o imprecisión. Este tipo de incerteza trata con la imprecisión que es propio en la mayoría de palabras humanas usadas para evaluar conceptos y derivar conclusiones.

Aunque la mayoría de conceptos no son definidos con precisión, los humanos pueden usarlos para evaluar situaciones muy complejas. Usando abstracción y pensando en analogías, una pocas oraciones describirán contextos complejos que sería muy duro de modelar con precisión matemática. Como ejemplo, consideremos la siguiente oración:

Probablemente daremos en el blanco.

A primera vista, es muy similar a la primera oración. Sin embargo, hay una significativa diferencia que se refiere a la forma de expresar la probabilidad. Mientras en la primera oración (ver 2.1), la probabilidad es expresada en un sentido matemático, en la segunda la probabilidad es más bien percibida en lugar de cuantificada matemáticamente.

III. LÓGICA FUZZY VS. TEORÍA DE PROBABILIDAD

Algunos expertos en teoría de probabilidades han negado la utilidad de la lógica *fuzzy*. Ellos afirman que todas las clases de incerteza pueden ser expresadas con la teoría de probabilidad. Sin embargo, se puede demostrar que esto no es así, luego de analizar la siguiente afirmación:

Los pacientes con hepatitis muestran en el 60 % de los casos fiebre alta, en el 45 % de los casos piel amarilla, y

en el 30% de todos los casos nauseas.

Si usted encuentra tal afirmación y desea implementarlo en un sistema, parecería muy fácil a primera vista. Si se tiene un paciente quien sufre de fiebre muy alta y nausea, pero su coloración es normal, se puede computar la probabilidad para una infección de hepatitis a través del cálculo de Bayes.

Aunque esto parece muy sencillo, el problema para la disciplina de la teoría de probabilidades se inicia cuando se debe definir que es una *fiebre alta*, ya que esta es una afirmación subjetiva o de incerteza léxica, debido a que no existe un umbral de decisión para considerar si una fiebre es o no alta.

La incerteza estocástica trata con la incerteza de si cierto evento ocurrirá y la teoría de probabilidad la puede modelar. Pero, la incerteza léxica trata con la incerteza de la definición del evento. En este caso, la teoría de probabilidad no puede ser usada para modelarla porque la combinación de categorías subjetivas en los procesos de decisiones humanas no sigue sus axiomas.

Para implementar la incerteza léxica o Imprecisión muy propia del lenguaje humano en soluciones de ingeniería, ha sido desarrollado la lógica *fuzzy* como un modelo matemático, de modo que permita la representación de decisiones humanas y evaluación de procesos en formas algorítmicas. [VON ALTRROCK, 1995]

IV. MODELANDO LA INCERTEZA LÉXICA O IMPRECISIÓN

En la teoría de conjuntos tradicional, se podría definir un conjunto con la denominación de *alto* para identificar a cada hombre como miembro o no del conjunto. La figura 1(a) es un ejemplo del conjunto de *alto*, donde una función definiría como *alto* a todos aquellos hombres con una altura superior a los 1.75 mts. , Clasificando a los demás, inclusive a los que están muy cerca de esta altura como hombres *no altos*.

En cambio, la figura 1(b) es un ejemplo del conjunto *fuzzy* de *alto*, donde a cada nivel le corresponde diferentes medidas de altura con diversos grados de pertenencia a ese conjunto, los limites entre los niveles no están claramente definidos, es decir, se diría que son *borrosos o difusos* o según su equivalencia en inglés, se diría que son *fuzzy*.

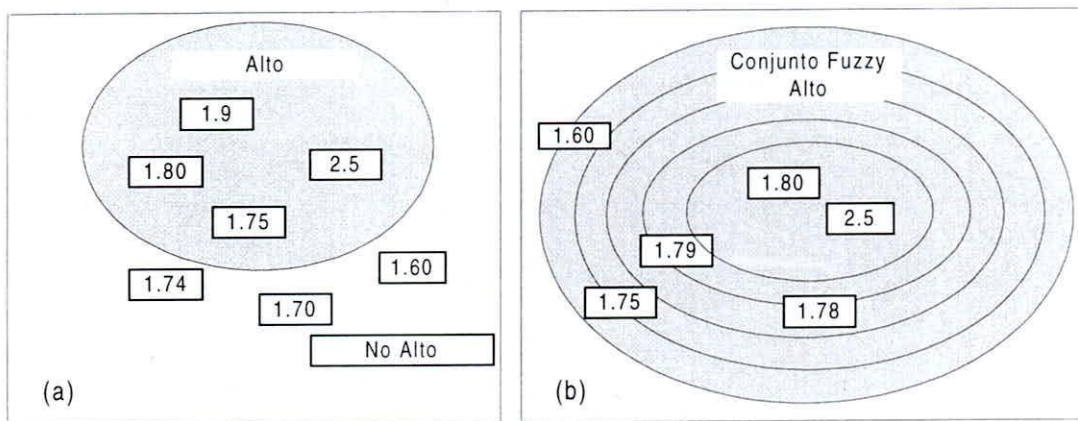


Figura 1. Conjunto Tradicional (a), Conjunto *Fuzzy* (b)

El grado de pertenencia al conjunto de *alto* (A) es representado por la función $\mu_A(x)$, donde x es la variable altura en el universo de X ($x \in X$). El rango de μ es cualquier valor entre 0 a 1, según represente ningún o un total grado de pertenencia al conjunto respectivamente. De esta manera, mientras una altura de 1.50 mts. no tendría ninguna pertenencia al conjunto, una altura de 1.80 mts. tendría una pertenencia completa. Alturas entre estos dos

valores extremos son miembros del conjunto en cierto grado. La tabla 1 representa diversos grados de pertinencia al conjunto de *alto*.

Tabla 1 – Grados de Pertinencia al conjunto *fuzzy* de *alto*.

$\mu_A(1.30 \text{ mts.}) = 0$	$\mu_A(1.61 \text{ mts.}) = 0.1$	$\mu_A(1.78 \text{ mts.}) = 0.9$
$\mu_A(1.40 \text{ mts.}) = 0$	$\mu_A(1.64 \text{ mts.}) = 0.35$	$\mu_A(1.80 \text{ mts.}) = 1$
$\mu_A(1.50 \text{ mts.}) = 0$	$\mu_A(1.75 \text{ mts.}) = 0.65$	$\mu_A(1.81 \text{ mts.}) = 1$

El Grado de pertinencia puede también ser representado por una función continua (función de pertenencia) como se muestra en la figura 2(a). Se observa, por ejemplo, que una altura de 1.74 y una de 1.76 mts. son evaluadas con ligeramente diferentes valores de μ . Aquí el grado de pertinencia en un conjunto llega a ser el grado de verdad de la afirmación. Por ejemplo, para una altura de 1.75 mts., la expresión *el hombre es alto* sería 0.65 de verdadero.

Observar, que el conjunto *fuzzy* son una generalización de los conjuntos convencionales ya que aquí los casos de $\mu=0$ (ninguna pertenencia) y $\mu=1$ (total pertenencia) son sólo un caso especial del conjunto *fuzzy*. El uso de conjuntos *fuzzy* definidas por una función de pertenencia, en expresiones lógicas, es llamado de Lógica *Fuzzy*.

El bloque fundamental de cualquier sistema de lógica *fuzzy* es la llamada variable lingüística. Aquí, se combinan múltiples categorías subjetivas que describen el mismo contexto. Así, para el caso de la *altura*, de hombres existirán las categorías de: *bajo*, *mediano*, *alto* y *muy alto*. Estos son llamados términos lingüísticos y representan los posibles valores de una variable lingüística. La figura 2(b) muestra las funciones de pertenencia de todos los términos de la variable lingüística de *Altura* dentro del mismo gráfico.

Estas variables lingüísticas permiten el traslado de una medida de la altura dentro de una descripción de una variable lingüística. Por ejemplo, una altura de 1.75 mts. no sólo sería evaluado con un grado de pertinencia para *alto* sino también con un grado de pertinencia para *mediano*.

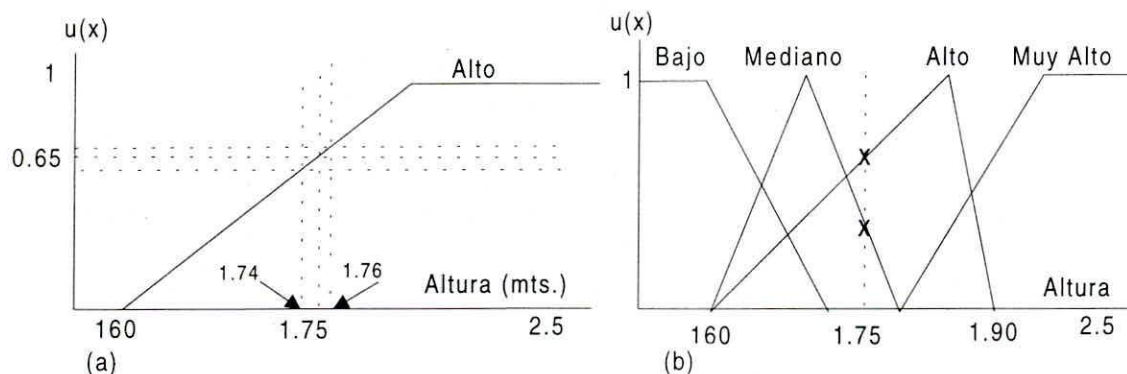


Figura 2. Variable *Altura* como una Función (a); Términos lingüísticos para la variable *Altura* (b)

V. SISTEMA FUZZY

5.1 Definiciones

Para entender un sistema *fuzzy* se necesita aclarar los términos siguientes: [MARSH STEVE et al., 1992], [BARRETO J.M, 1997]

Función de Pertenencia – Define a un conjunto *fuzzy* mapeando entrada abruptas de su dominio hasta su Grado de Pertenencia. En la Figura 2(b) tenemos las funciones de pertenencia: Bajo, Mediano, Alto, Muy Alto.

Grado de Pertenencia – grado al cual una entrada abrupta es compatible con una función de pertenencia, el cual puede tomar valores dentro de un valor predeterminado entre 0 y 1. En la figura 2(a) la entrada abrupta de 1.75 mts. tiene un grado de pertenencia con la función alto igual a 0.65.

Etiqueta – Nombre descriptivo usado para identificar una función de pertenencia. En la figura 2(b) tenemos los nombres: Bajo, Mediano, Alto, Muy Alto.

Entradas Abruptas (*Crisp inputs*) – Diferentes valores discretos de la variable del sistema, en la figura 2(b) se puede ver por ejemplo: 1.60, 1.75, 1.80, 2.5.

Rango/Dominio – Intervalo sobre la cual una función de pertenencia es mapeada. En la figura 2(b), el dominio de la función de pertenencia Alto es de 1.60 a 1.9 mts. y su rango es de 0.3 mts.

Universo de Discurso – Rango de todos los valores posibles de la variable del sistema, en la figura 2(b) es de 0 a 2.5.

5.2 Etapas de un Sistema Fuzzy

Un sistema *fuzzy* consta de tres etapas:

- *Fuzzyfication*
- Reglas de Evaluación
- *Defuzzification*

5.2.1 Fuzzyfication

Esta etapa calcula el grado de pertenencia que podría tener una entrada abrupta a una o a varias funciones de pertenencia de una variable, el resultado se denomina entrada *fuzzy*. El procedimiento puede representarse como en la figura 3.

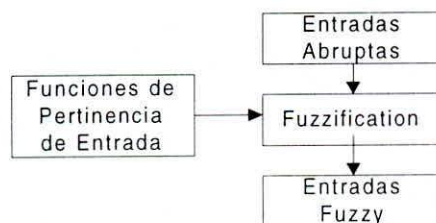


Figura 3 – Etapa de *Fuzzyfication*

Para transformar las entradas abruptas en entradas *fuzzy*, primero se debe determinar las funciones de pertenencia de la variable. Una vez realizado esto, la entrada abrupta es comparada con la función de pertenencia correspondiente produciendo valores de entrada *Fuzzy*.

Para ilustrar el proceso de *Fuzzyfication*, observemos lo que sucedería en un supuesto sistema de riego, en donde el control *fuzzy* tendría por finalidad la de controlar el tiempo de riego de la superficie de un terreno dependiendo de la temperatura del aire y de la humedad del suelo. Este sistema básicamente estaría constituido por dos variables de entrada (Humedad del suelo y Temperatura del aire) y una variable de salida (Tiempo de riego).

El primer paso consiste en asignar, dentro del Universo de Discurso, Etiquetas a cada variable. Así para la variable de entrada Temperatura del Aire, se podría asignar las siguientes Etiquetas:

CONGELADO, FRIO, NORMAL, TIBIO, CALIENTE

Mientras, para el caso de la variable de entrada Humedad del Suelo, las etiquetas podrían ser:

SECA, HÚMEDA, MOJADA

Luego definimos para estas etiquetas las funciones de pertenencia como se muestra en la figura 4. El eje Y (μ) de la figura 4 se refiere al grado de pertenencia para la cual los valores de entrada abrupta pertenecen a cada función de pertenencia. Así, por ejemplo, las entradas *fuzzy* para una temperatura de 33°C y 11% de acuerdo a la figura 4 serán:

- La temperatura es tibia con un grado de verdad de 0.2
- La temperatura es caliente con un grado de verdad de 0.46
- La humedad del suelo es seca con un grado de verdad de 0.25
- La humedad del suelo es húmeda con un grado de verdad de 0.75

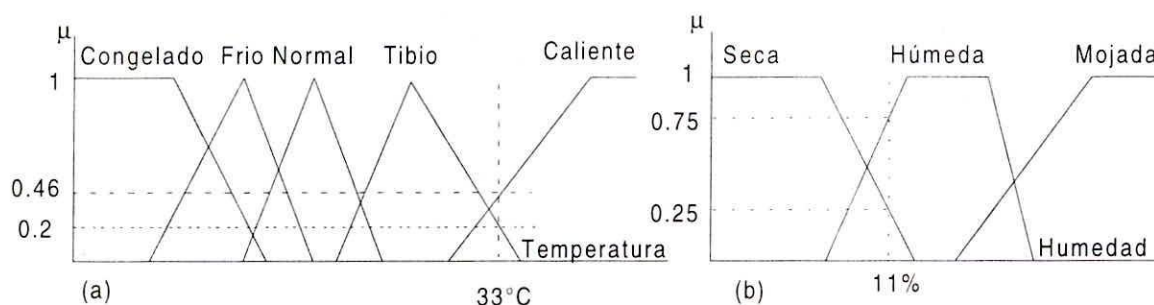


Figura 4. Funciones de Pertenencia para las variables: Temperatura del Aire (a) y Humedad del suelo (b)

5.1.2 Reglas de Evaluación

En la segunda etapa del procesamiento de la lógica *fuzzy*, llamada Regla de Evaluación o de Inferencia *fuzzy*, el controlador *fuzzy* usa reglas lingüísticas sobre los resultados que fueron generados en la etapa de *Fuzzyfication* (denominados entradas *fuzzy*). El procedimiento puede representarse como en la figura 5.

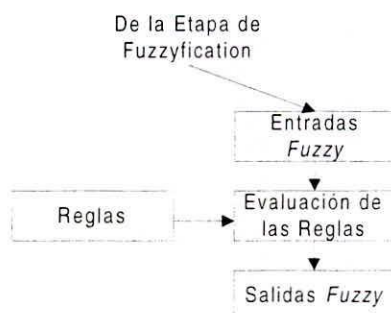


Figura 5. Etapa de Reglas de Evaluación

Las reglas *fuzzy* son sentencias Si-Entonces (*If-Then*) que describen la acción a ser tomada en respuesta a varias entradas *fuzzy*. Las reglas están confinadas a un predefinido conjunto de términos lingüísticos y cuya sintaxis es la siguiente:

IF Antecedente 1 *AND* Antecedente 2...*Then* Consecuente 1 *AND* Consecuente 2...

donde, *AND* es uno de los operadores de la lógica *fuzzy*. El Antecedente tiene el siguiente formato:

Variable de Entrada = Etiqueta [la temperatura (variable) es caliente (etiqueta)]

El consecuente está en la forma de:

Variable de Salida = Etiqueta [la duración de regadío (variable) es prolongada (etiqueta)]

Las reglas siguen el sentido común del comportamiento del sistema y son escritos en términos de las etiquetas de las funciones de pertenencia.

Para el sistema del ejemplo, de dos entradas y una salida, las reglas pueden ser representadas por la matriz de la figura 6, en donde se observa las etiquetas de las variables de entrada Temperatura y Humedad y las etiquetas de la variable de salida (Duración de Regadío) las cuales son: corto, medio y prolongado.

A N T E C E D E N T E		ANTECEDENTE 1				
		TEMPERATURA				
H U M E D A D		Congelado	Frio	Normal	Tibio	Caliente
	Mojada	Corto	Corto	Corto	Corto	Corto
	Húmeda	Corto	Medio	Medio	Medio	Medio
	Seca	Prolongado	Prolongado	Prolongado	Prolongado	Prolongado
2						

Figura 6. Reglas posibles para el Sistema de Regadío.

De acuerdo a esta matriz, para el sistema en total existirán 15 reglas posibles, pero para nuestro ejemplo, sólo serán disparadas las reglas cuyos antecedentes fueron calculadas en la etapa de *fuzzification* (fig.4) y que involucra a 4 etiquetas, dos de la variable temperatura: tibio, caliente y dos de la variable humedad del suelo: seca y húmeda. Entonces con estas etiquetas, como se puede deducir de la figura 6, sólo serán posibles las siguientes reglas:

REGLA 1	<i>IF</i> la temperatura es caliente <i>AND</i> la humedad es seca, <i>THEN</i> la duración es prolongada
REGLA 2	<i>IF</i> la temperatura es tibia <i>AND</i> la humedad es húmeda, <i>THEN</i> la duración es media.
REGLA 3	<i>IF</i> la temperatura es tibia <i>AND</i> la humedad es seca, <i>THEN</i> la duración es prolongada.
REGLA 4	<i>IF</i> la temperatura es caliente <i>AND</i> la humedad es húmeda, <i>THEN</i> la duración es media.

Los antecedentes de estas reglas poseen un grado de pertenencia o relevancia, que fueron calculados en la etapa de *Fuzzyfication*. Debido a que los antecedentes son conectados por el operador *AND* la regla de fuerza asume como resultado el valor más pequeño de los antecedentes.

El procedimiento de cálculo es mostrado a continuación:

REGLA 1	IF la temperatura es caliente (0.46) AND la humedad es seca (0.25), THEN la duración es prolongada	0.25
REGLA 2	IF la temperatura es tibia (0.2) AND la humedad es húmeda (0.75), THEN la duración es media.	0.20
REGLA 3	IF la temperatura es tibia (0.2) AND la humedad es seca (0.25), THEN la duración es prolongada.	0.20
REGLA 4	IF la temperatura es caliente (0.46) AND la humedad es húmeda (0.75), THEN la duración es media.	0.46

Hasta aquí, se ha evaluado cada una de las cuatro reglas, el próximo paso es determinar la salida *fuzzy* comparando las fuerzas de todas las reglas que especifican el mismo consecuente es decir la misma acción de salida. Así, se puede observar que las reglas 2 y 4 ordenan la misma acción de salida, es decir una *duración de regadío media*, mientras que las reglas 1 y 3 ordenan una *duración de regadío prolongada*, en este caso para las reglas 2 y 4 lo mismo que para las reglas 1 y 3, la salida *fuzzy* es determinada por la regla de máxima fuerza. En términos simples, si dos o más reglas intentan afectar la misma salida, la regla que es más verdadera (de mayor fuerza) dominará. Habrá una salida *fuzzy* para cada etiqueta o función de pertenencia. Así en este caso tendremos:

- La Duración de Regadío es prolongada con un grado de verdad de 0.25
- La Duración de Regadío es media con un grado de verdad de 0.46

El método de regla de evaluación usada aquí es llamado inferencia *MIN-MAX*, ya que toma el mínimo de los antecedentes para determinar la fuerza de cada regla y toma la regla más fuerte para cada consecuente, los cuales determinan las salidas *fuzzy*. [MARSH STEVE et al., 1992]

5.1.3 Defuzzification

En la *Defuzzification*, el procedimiento de cálculo se representa como en la figura 7, en donde el objetivo final consiste en encontrar las salidas abruptas, para esto cada salida *fuzzy*, las que fueron encontradas en la etapa de reglas de evaluación, modificaran a su respectiva *función de pertenencia de salida*. Las etiquetas para estas funciones de salida hacen referencia a la duración de regadío del terreno, es decir serán de: corto, medio y prolongado.



Figura 7. Etapa de *Defuzzification*

Los métodos más comunes en la *Defuzzification* son:

Método de la Media Ponderada, se trata de un sencillo cálculo del promedio entre los valores de salida que se obtendrían para cada uno de los conjuntos *fuzzy* multiplicados (ponderados) por el peso de la correspondiente regla o grado de pertenencia al subconjunto.

- Método de Centro de Masa (Centro de Gravedad o Centroide), en el cual es determinado el centro de gravedad del conjunto de salida, resultante de la unión de las contribuciones de todas las reglas válidas. El valor de salida es la abscisa del punto del centro de gravedad.
- Singleton, método derivado del Centro de Masa, en el cual se obtiene las abscisas de los puntos del centro de gravedad para cada regla válida. El valor de salida es obtenido por la media ponderada, con relación a los grados de pertenencia de estos valores. [HILERA; MARTINEZ, 1995] [KLIR; YUAN, 1995]

A continuación, aplicaremos el método de Centroide o de Centro de Gravedad (COG). En este método, cada función de pertenencia de salida son cortadas (corte Lambda) en el nivel indicado por su respectiva salida *fuzzy*. Las funciones de pertenencia cortadas resultantes son luego combinadas para calcular su centro de gravedad. Las salidas *fuzzy* que fueron calculadas en la etapa anterior, se muestran en la tabla 2, estos valores son usados como niveles de corte de cada función de pertenencia de salida como se muestra en la figura 8.

Tabla 2 – Salidas Fuzzy

Corto	Medio	Prolongado
0	0.46	0.25

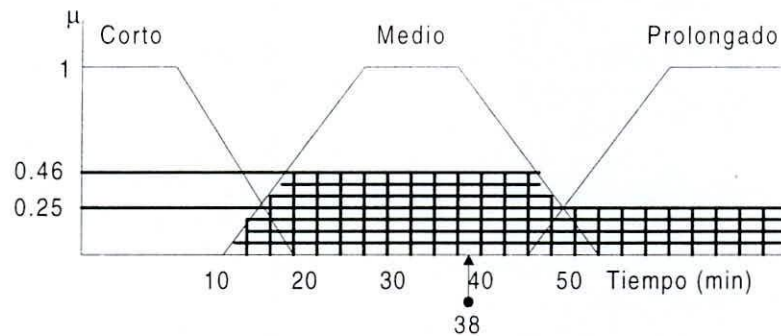


Figura 8. Funciones de Pertenencia de salida cortadas por las salidas fuzzy

La próxima etapa es encontrar el *punto de balance*, del centro de gravedad, del área sombreada. Este valor representa la *defuzzification*. La fórmula empleada es [KLIR; YUAN, 1995]:

$$\text{Salida Abrupta} = \frac{\sum_{i=1}^n y_i \mu(y_i)}{\sum_{i=1}^n \mu(y_i)}$$

Finalmente, de la fórmula se obtendrá una salida abrupta de 38 minutos, que equivale al tiempo de duración de regadío para una humedad de 11% y una temperatura de 33°C.

VI. REDES NEURONALES Y LÓGICA FUZZY

Debido a su constitución y a sus fundamentos, las redes neuronales artificiales presentan un gran número de características semejantes a las del cerebro. Por ejemplo, son capaces de aprender de la experiencia (datos de muestra), de generalizar a partir de casos anteriores a nuevos casos, de abstraer características esenciales a partir de entradas que representen información irrelevante. [HILERA; MARTINEZ, 1995]

Sin embargo, las redes neuronales tienen ciertas desventajas, entre las cuales se pueden citar [VON ALTRROCK, 1995]:

- permanecer como una *caja negra*, de la cual no se puede detectar ni modificar la causa de un determinado comportamiento;
- requerir de esfuerzos computacionales prohibitivos para la mayoría de aplicaciones comerciales;
- requerir de gran experiencia para la selección de un apropiado modelo de red y el establecimiento del algoritmo de aprendizaje (técnicas todavía algo oscuras).

De todas estas desventajas probablemente la de mayor limitación, por permanecer como una caja negra, es la de no poder verificar y optimizar la red a través de un camino fácil y directo.

Por otro lado, en lógica *fuzzy*, la principal de sus ventajas es que describe al sistema con simples inferencias *if-then*, convirtiéndose de esta manera en un sistema muy flexible para cualquier modificación y verificación directa. Sin embargo, su principal desventaja es la de no poder aprender automáticamente de datos de muestra, complicándose en sistemas que requieren un gran conjunto de reglas, lo cual demandará un gran esfuerzo.

Tanto las redes neuronales y la lógica *fuzzy* son técnicas de diseños potentes que tienen fortalezas y debilidades. Por lo tanto, una combinación inteligente de ambas tecnologías trae como consecuencia altos beneficios.

Actualmente, existe una gran cantidad de propuestas de combinación de la capacidad de aprendizaje de las redes neuronales y del procesamiento de información de la lógica *fuzzy*. Esta combinación suele referirse a aspectos como [HILERA; MARTINEZ, 1995]:

- la utilización de neuronas *fuzzy*, cuyo funcionamiento se describe en función de operadores *fuzzy* (Producto y Suma), en lugar de aritméticos;
- la adaptación de algoritmos *fuzzy* de aprendizaje de redes conocidas, como Perceptron (con o sin capas ocultas);
- la simulación con redes neuronales de sistemas expertos dinámicos que integran las *opiniones fuzzy* de varios especialistas;
- el almacenamiento y evaluación con redes neuronales de las reglas utilizadas en sistemas de control *fuzzy*;
- la generación y adaptación de las Funciones de pertenencia a conjuntos *fuzzy* mediante redes neuronales, etc.

6.1 Tecnología NeuroFuzzy para Generación de Funciones de Pertenencia y Reglas de Inferencia.

Muchos caminos alternativos de integración de redes neuronales y lógica *fuzzy* han sido propuestos en la literatura científica. Pero sólo muy pocos han sido exitosamente logrados para aplicaciones industriales.

Para ayudar a generar y optimizar (sobre la base de información de algunos datos de entrenamiento) las Funciones de pertenencia y las Reglas de Inferencia, una de las aproximaciones más comunes es la llamada *fuzzy Associative Memories* (FAMs). Una FAM es una regla de lógica *fuzzy* con un peso asociado. Un algoritmo mapea la FAM a una Neurona que pertenece a una red neuronal basado en un algoritmo *Back Propagation* (este algoritmo ha llegado a ser el estándar para la mayoría de redes neuronales por su gran desempeño).

Hoy en día existen herramientas de software *Neurofuzzy* que trabajan como un asistente *inteligente* con su diseño sin que el usuario tenga que preocuparse de los detalles de los algoritmos en las cuales están basados.

Una de las ventajas de los software *Neurofuzzy* frente a una solución sólo con redes neuronales es que el código generado es mucho más eficiente. La computación de un sistema de lógica *fuzzy* entrenado con *Neurofuzzy* sobre un microcontrolador o una PC puede requerir como 0.1 ms. y 1 KB de memoria. Esto es varios ordenes de magnitud más rápido y compacto que una solución con redes neuronales para la misma aplicación. Por este motivo, una solución *Neurofuzzy* permitirá más aplicaciones en tiempo real.

VII. AVANCES EN LÓGICA FUZZY

La teoría de Lógica *fuzzy* fue inventada en los Estados Unidos, perfeccionada en Europa, y aplicada comercialmente por los Japoneses. La primera publicación en lógica *fuzzy* se hizo en U.S.A. por Lotfi Zadeh, Profesor en Teoría de Sistemas de la Universidad de California, Berkeley en 1965.

Pero no es que hasta fines de los 70 y principios de los 80 que se comenzó a aplicar, debido a que se habían superado las limitaciones de Hardware de la década de los sesenta que no permitían gran cantidad de procesamiento de datos. La primera aplicación industrial se realizó en Queen Mary College de Londres, Inglaterra, en 1970; donde Ebrahim Mamdani controló un generador de vapor que no se lograba por métodos convencionales.

Hace sólo siete años (1993), las principales corporaciones europeas para no perder esta tecnología en manos de los Japoneses empezaron un esfuerzo para promover la lógica *fuzzy* y sus aplicaciones. Desde entonces, más de 200 productos se han lanzado al mercado, la mayoría convirtiendo a los aparatos electrodomésticos en inteligentes permitiéndoles el ahorro de consumo de energía. Además, de existir una innumerable cantidad de aplicaciones en el área de automatización industrial.

Desde las primeras aplicaciones europeas, las compañías japonesas han empezado a usar lógica *fuzzy*. Algunas de las primeras aplicaciones fueron hechas para una planta de tratamiento de agua realizada por la empresa Fuji Electric en 1983 y el sistema de control para Tren Subterráneo realizada por Hitachi en 1987 (en donde el comportamiento del controlador *fuzzy* reportado fue superior al de un control PID en cuanto a la conducción, precisión de la parada, ahorro de energía, tiempo de funcionamiento y robustez). Omron Tateishi Electric Co. Ltd construyó una computadora *fuzzy* capaz de procesar información *fuzzy* a una velocidad de 10 mega inferencias lógicas *fuzzy* por segundo [HABER, 1992]. Las cámaras fotográficas y las de vídeo usan lógica *Fuzzy* y Mitsubishi anunció el primer automóvil del mundo que usa un control basado en la lógica *fuzzy*. En automatización de la fábrica, Omron S.A. tiene más de 350 patentes.

En los Estados Unidos, en marzo de 1989 el centro de Microelectrónica de Carolina del Norte de los estados Unidos fabricó un chip de lógica borrosa, diseñado por Watanabe que alcanzó una capacidad de 580000 inferencias lógicas por segundo. Actualmente en USA se está buscando un área de aplicación en donde competir. Saben que los japoneses son líderes indiscutibles en aplicaciones sobre Camcorders, equipos de Alta Fidelidad, etc. y los Europeos, líderes en aplicaciones sobre automatización industrial. Por ese motivo, piensan competir fuertemente con los japoneses en aplicaciones sobre el área de ingeniería automotriz, además de explotar la aplicación de la combinación de Redes Neuronales y Lógica *fuzzy* para el desarrollo de su tecnología.

VIII. LINEAS DE INVESTIGACIÓN

Entre las áreas temáticas de las cuales se priorizan las investigaciones acerca del control en lógica *fuzzy* pueden señalarse las siguientes [HABER, 1992]:

- Eficiencia y sistematicidad en el proceso de adquisición de los conocimientos de la persona experta hacia la base de conocimientos del control de lógica *fuzzy*.
- Modelado *fuzzy* de procesos y uso de esos modelos en el diseño de un controlador de lógica *fuzzy*.

- Procedimientos sistemáticos para el diseño de lazos de control borroso y la correspondiente teoría de sistemas dinámicos *fuzzy* (estabilidad, controlabilidad, etc.).
- Diseño de sistemas capaces de aprender. Eventual combinación lógica *fuzzy*-redes neuronales.
- Hardware para el control *fuzzy* tal que se provean soluciones para problemas aún existentes, como es el caso de la velocidad en control de robots.

IX. CONCLUSIONES

El beneficio de la lógica *fuzzy* es el de permitir describir el comportamiento de un sistema con simples reglas o sentencias de programa *if-then* (si-entonces) que reflejan la experiencia del operador humano sobre el sistema. Sin embargo, un sistema *fuzzy* no aprende como en el caso de las redes neuronales, es decir, solo responderá a aquellas entradas para cuales las reglas han sido programadas.

Gracias a la fusión de lógica *fuzzy* con las redes neuronales es posible la generación y optimización automática de las funciones de pertenencia y las reglas de inferencia. La solución de problemas de control con software Neurofuzzy debido a su ventaja comparativa de tiempo de procesamiento frente a las redes neuronales las hace óptimas para aplicaciones complejas en tiempo real.

Desde la aparición de la lógica *fuzzy*, hace más de 30 años, se han escrito y diseñado cientos de aplicaciones, el liderazgo como es comprensible lo poseen los países industrializados. Sin embargo, los softwares de desarrollo y simulación están al alcance de países como el nuestro, lo que nos permitiría aplicar esta tecnología a nuestros propios problemas. Por esta razón, es necesario, formar grupos de discusión y de investigación en las universidades de tal manera de permitir intercambiar conocimientos y experiencias sobre este tema.

X. REFERENCIAS BIBLIOGRAFICAS

- Barreto Jorge Muniz; *Inteligencia Artificial No Lineal del Siglo XXI*; ppv Edições Florianópolis, 1997.
- Haber Haber Rodolfo, *Control Borroso*, Departamento de Control automático – Facultad de Ingeniería Eléctrica – Instituto Politécnico “Julio Antonio mella” – Santiago de Cuba, 1992.
- Hilera González Ramon José, Martínez Hernando Víctor José; *Redes Neuronales Artificiales Fundamentos, Modelos y Aplicaciones*; Addison-Wesley Iberoamericana, 1995.
- Klir George J. and Yuan Bo; *Fuzzy Sets and Fuzzy Logic theory and Applications*; Prentice Hall, 1995.
- Marsh Steve, Wei Huang Yee, Sibigtroth Jim, Paloian Dave, Mazuelos Duberty, Weiss Don, Spielman Jason, Dumas John, Leung Michael, Tomazin Tom and Osborn Steve; *Fuzzy Logic Education program*, Center for Emerging Computer Technologies, Motorola, Inc, 1992.
- Von Altrock Constantin; *Fuzzy Logic and NeuroFuzzy Applications Explained*; Prentice Hall, 1995.