

# FUZZYTOOL “HERRAMIENTA COMPUTACIONAL EDUCATIVA PARA EL APRENDIZAJE DE SISTEMAS DIFUSOS.”

## **Edwar Jacinto Gómez.**

Ingeniero en Control Electrónico e Instrumentación Universidad Distrital “Francisco José de Caldas”. Actualmente MSc (c). en la Universidad Distrital “Francisco José de Caldas”, docente investigador de la misma universidad en el proyecto curricular de Tecnología en Electrónica.  
edwarjg@hotmail.com

## **Horderlin V. Robles Vega**

Ingeniero en Control Electrónico e Instrumentación Universidad Distrital “Francisco José de Caldas”. Especialización en Bioingeniería de la misma. Actualmente MSc (c). en la Universidad Nacional de Entre Ríos en Paraná – Argentina. docente investigador de Tiempo completo de la Escuela Colombiana de Carreras Industriales en la coordinación de Ingeniería Biomédica.  
hordevega@eccci.edu.co

**Tipo:** Artículo de reflexión

**Fecha de Recepción:** Dic. 4 de 2010

**Fecha de Aceptación:** Marzo 15 de 2011

## FUZZYTOOL – AN EDUCATIONAL COMPUTATIONAL TOOL FOR TEACHING FUZZY SYSTEMS

### **Abstract**

This article presents the software package called FuzzyTool as an educational tool for the learning of fuzzy systems. The most important methods and operations employed in the design of fuzzy systems in the world are defined. FuzzyTool was implemented as a computational educational tool comparing it with each of the software packages available for the design, simulation and development of fuzzy systems and controllers (fuzzytech, Matlab, Unfuzzy, etc). The tool developed in this work provides an interface that guides users in the design of fuzzy systems by means of three fundamental steps, namely the edition of variables, the construction of a rule base (knowledge base) and finally, the rule inference.

**Keywords:** Fuzzy logic, educational software, design.

### **Resumen**

Este artículo presenta el software FuzzyTool desarrollado como una herramienta educativa para el aprendizaje de sistemas difusos. Se definen los métodos y operaciones difusas más importantes y utilizadas en el diseño de sistemas difusos en el mundo. FuzzyTool se implementó como una herramienta computacional educativa teniendo como referencia cada uno de los programas que se encuentran en la actualidad para el desarrollo, diseño y simulación de sistemas y controladores difusos (fuzzytech, Matlab, Unfuzzy, etc). La herramienta que se desarrolló le entrega al usuario una interfaz gráfica donde es guiado en el diseño de los sistemas difusos, por medio de tres pasos fundamentales como lo son: la edición de las variables, la construcción de la base de reglas o base de conocimiento y por ultimo, la inferencia de reglas.

**Palabras clave:** Fuzzy logic, software educativo, diseño

## 1. INTRODUCCIÓN

Los artículos electrónicos sin componentes difusos se están quedando gradualmente desfasados. Como una mordaza, que muestra la popularidad de la lógica difusa, cada vez es más frecuente un sello con "Fuzzy Logic" impreso sobre el producto. La lógica difusa es básicamente una lógica multivaluada que permite valores intermedios para poder definir evaluaciones convencionales como sí/no, verdadero/falso, negro/blanco, etc. Las nociones como "más bien caliente" o "poco frío" pueden formularse matemáticamente y ser procesados por computadoras [1]. Las matemáticas generadas por estas teorías son consistentes y la lógica difusa puede ser una generalización de la lógica clásica.

## 2. MARCO REFERENCIAL

La lógica difusa se inició en 1965 por Lotfi A. Zadeh, profesor de ciencia de computadoras en la Universidad de California en Berkeley quien publicó un trabajo acerca de los conjuntos difusos, el cual propone que los valores falso o verdadero operen sobre el rango de números reales. La estadística mide la probabilidad que un evento futuro ocurra, cuando la lógica difusa mide la ambigüedad de eventos que ya han ocurrido.

Cualquier sistema dinámico o estático que hace uso de la lógica difusa o de conjuntos difusos, se considera como un sistema difuso. Los sistemas difusos se pueden emplear en modelado y control (a nivel directo o supervisor) de sistemas, en reconocimiento de patrones, etc. Específicamente, en modelado, el sistema se puede describir mediante una relación difusa o un conjunto de reglas proposicionales difusas (cualitativas), que pueden tener distintas formas.

Aunque el diseño del sistema difuso (modelo y/o controlador) se realice tomando en cuenta el conocimiento de un experto, o mediante algún mecanismo de optimiza-

ción para el ajuste de sus parámetros, finalmente éste constituye una aplicación (mapeo) no lineal de las entradas en las salidas, en su versión más básica. Esta aplicación es más compleja si el sistema está incluido dentro de una estructura dinámica.

Hasta hace poco tiempo, entre las múltiples críticas al control difuso se contaba la ausencia de métodos formales de prueba de estabilidad y la falta de índices que mostraran las ventajas y la conveniencia de utilizar controladores difusos en vez de controladores convencionales. Esto se debe a que en sus orígenes, el diseño del controlador difuso se basaba exclusivamente en heurística, o conocimiento basado en la experiencia. Sin embargo, actualmente, se están haciendo esfuerzos [2], [3], [4] para combinar algunas técnicas convencionales de control, como son el control PID, por realimentación del estado, predictivo, adaptativo y de estructura variable, con el diseño del controlador difuso, obteniendo resultados si no superiores, al menos iguales a ellos.

La teoría de conjuntos borrosos o difusos desarrollada por Lotfi Zadeh [5], parte de la teoría clásica de conjuntos y de la lógica booleana e introduce una función de pertenencia al conjunto, definida como un número real entre 0 y 1. Se introduce el concepto de valor lingüístico asociado a cada subconjunto borroso, y definido por una palabra o etiqueta lingüística A. Para cada valor lingüístico A se define una función  $\mu_A(x)$  que indica el grado de probabilidad de que la variable  $x$  esté incluida en el concepto representado por el valor lingüístico A.

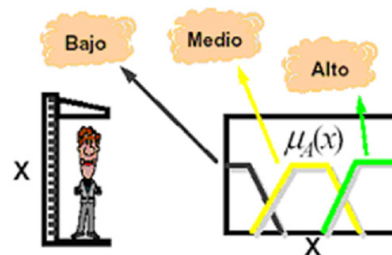


Fig. 1. Definición de conjuntos borrosos.

Para los valores  $x$  que puede tomar una magnitud física como la altura de una persona pueden definirse diversos conjuntos borrosos (Fig. 1). Las variables lingüísticas son variables cuyos valores son palabras o sentencias (no números) [6]. A menudo se quiere escribir el estado de un objeto o fenómeno, para ello se usa una variable cuyo valor hace la descripción. Unos ejemplos podrían ser temperatura (Fig. 2), limpieza, sabiduría, etc. Una variable lingüística admite que sus valores sean etiquetas lingüísticas, que son términos lingüísticos definidos como conjuntos difusos (sobre cierto dominio subyacente). Un ejemplo de lo anterior sería, Temperatura "Cálida", o "aproximadamente 25°C" (#25).

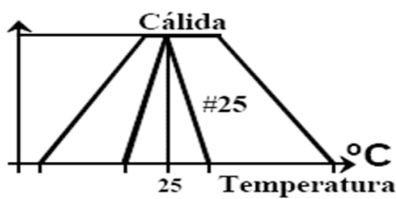


Fig. 2. Ejemplo gráfico de una variable lingüística.

Otro de los conceptos necesarios en el diseño de sistemas difusos es la inferencia difusa donde, se define la base de reglas y los métodos u operación de implicación, composición, combinación e inferencia que nos permiten obtener los resultados adecuados para nuestro sistema. Se llaman reglas difusas el conjunto de proposiciones IF-THEN que modelan el problema que se quiere resolver. Una regla difusa simple tiene la forma:

"si  $u$  es  $A$  entonces  $v$  es  $B$ "

Donde  $A$  y  $B$  son conjuntos difusos definidos en los rangos de " $u$ " y " $v$ " respectivamente. Una regla expresa un tipo de relación entre los conjuntos  $A$  y  $B$  cuya función característica sería  $\mu_{A \rightarrow B}(x, y)$  y representa lo que se conoce como implicación lógica. La elección apropiada de esta función carac-

terística esta sujeta a las reglas de la lógica proposicional. Luego se utilizan mecanismos de inferencia que usan los principios de lógica difusa explicados anteriormente para realizar un mapeo de los conjuntos de entrada a los conjuntos difusos de salida.

Por ultimo, se utiliza el método de defuzificación que consiste en seleccionar un elemento representativo de un conjunto difuso y suprimir lo difuso para luego, elegir un valor concreto de salida que lo represente. Para esto, existen diversos criterios. El aporte para el proceso de defuzificación en un conjunto difuso (la salida de la operación de agregación del conjunto difuso) es un número crisp o real. Los conjuntos difusos ayudan a la evaluación de cada regla durante los pasos intermedios y la salida final para cada variable es generalmente un conjunto difuso. Entonces, dado el conjunto difuso que abarca un rango de valores de salida, y a sabiendas que se necesita devolver un número crisp, se aplica el método de defuzificación más adecuado (Fig. 3). Quizá el método de defuzificación más popular es el cálculo del centroide o Centro de Alturas; que regresa el centro de área bajo la curva.

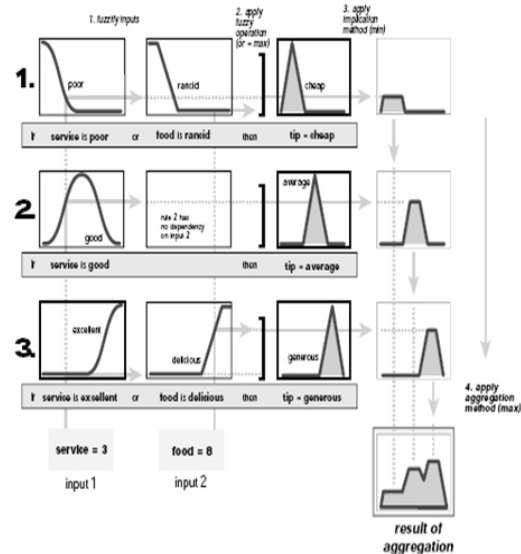


Fig. 3. Mecanismos de Inferencia.

Para el diseño de sistemas difuso en la ac-

tualidad, existen una gran variedad de aplicaciones (software), en las cuales, puedes diseñar, simular, implementar y analizar los distintos métodos y operaciones necesarias para los sistemas difusos vistos anteriormente. Por ejemplo, las aplicaciones más populares son el Fuzzytech a nivel industrial, el toolbox de Matlab, Xfuzzy, Unfuzzy, etc. Todas estas herramientas debido a su grado de complejidad requieren un estudio previo de la lógica difusa. Por lo anterior, se muestra una herramienta educativa para el aprendizaje de los sistemas difusos orientada a introducir al usuario a la lógica difusa.

### 3. DESARROLLO DEL SOFTWARE

#### 3.1 Descripción

El software educativo para sistemas difusos FuzzyTool muestra las más importantes operaciones y métodos utilizados en la lógica difusa para el diseño de sistemas de control (Fig. 4). Dicho software esta dividido en tres partes principales en las cuales se muestran paso a paso las opciones de diseño más populares.



Fig. 4. Ventana principal de FuzzyTool.

El primer paso es el diseño y edición de las variables de entrada y salida con sus respectivas funciones de pertenencia o conjuntos difusos, así se editan sus nombres,

rangos, funciones de pertenencia y demás características necesarias para el diseño de sistemas difusos. El segundo paso donde te guía el programa es el diseño de la base de reglas, en esta etapa es donde se escoge la base de conocimiento de nuestro sistema, en esta ventana además aparece la opción de cambiar las reglas y de cambiar los controles y operaciones utilizadas para dicho diseño. Por último, se tiene el paso número tres donde se razona acerca de todas las reglas antes diseñadas, en la etapa de inferencia de reglas se selecciona solamente una variable de entrada y un conjunto o función de pertenencia y se evalúa todo de acuerdo a los controles escogido (operación and, or, implicación, composición, combinación, método de defuzzificación y método de inferencia), para luego ser graficada (inferida) y así mirar la conclusión en termino de una de sus salidas. Además en esta etapa se utilizarán los métodos de defuzzificación para obtener el valor crisp o real de la conclusión.

#### 3.1.1 Formulario Principal

Es la primera ventana vista por el usuario a la ejecución del software (Fig. 4). En esta ventana aparecen las primeras opciones que tiene este software como lo son:

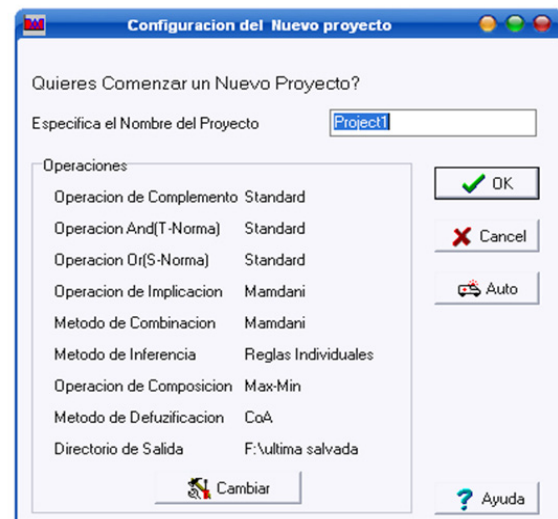


Fig. 5. Configuración de nuevo proyecto.

### Configuración de nuevo proyecto

En esta ventana puedes especificar en nombre del nuevo proyecto además, esta diseñado para que el usuario tenga la oportunidad de cambiar los controladores (operaciones y métodos a implementar) deseados para el sistema (Fig. 5).

Las opciones que aparecen en esta ventana son:

- Ok.
- Cancel.
- Auto.
- Cambiar.
- Ayuda.

### Configuración del sistema

Cada una de las operaciones, excepto la operación de complemento la cual no puede ser cambiada, pueden ser reemplazadas por el usuario presionando el botón cambiar que nos envía a la ventana de configuración del sistema (Fig. 6).

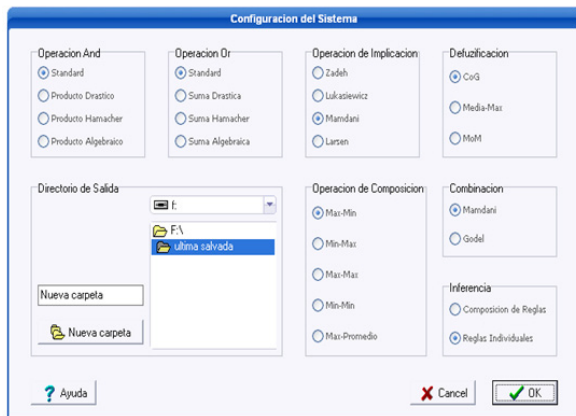


Fig. 6. Configuración del sistema.

En dicha ventana se tendrá la opción de escoger entre distintas operaciones y métodos como:

- Operación and o t-normas.
- Operación or o s-norma.
- Operación de implicación.

- Operación de composición
- Método de inferencia.
- Método de combinación.
- Método de defuzificación.

Los cuales determinan el comportamiento del sistema difuso.

### Configuración automática del sistema

Esta ventana es donde se escogen los mecanismo de inferencia mas usados o populares, cada uno de los mecanismos utilizan combinaciones de operaciones y métodos diferentes entre si (Fig. 7). Por lo tanto cada uno de los mecanismos de inferencia difiere en los resultados finales del sistema difuso diseñado.

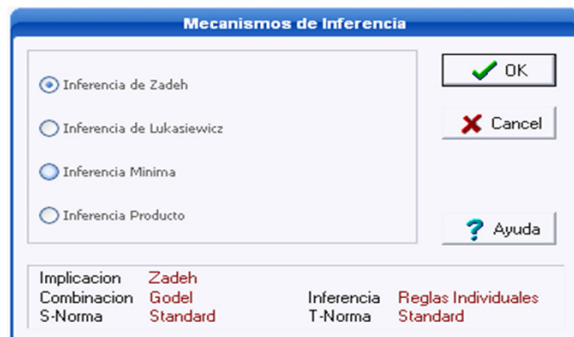


Fig. 7. Configuración automática del sistema.

La combinación de cada uno de los mecanismos de inferencia se muestra a continuación:

- Inferencia Producto.
- Inferencia Mínima.
- Inferencia de Zadeh.
- Inferencia de Lukasiewicz.

A continuación se dará una explicación mas detallada de cada una de las etapas del software para el diseño de los sistemas difusos.

### 3.2 Paso Nº 1: Diseño de variables de entrada y salida

El paso Nº 1 es la edición de las variables de entrada y salida, con sus respectivos con-

juntos difusos. En esta ventana (Fig. 8) se pueden apreciar cuadros de textos donde se podrían editar las variables de entrada y salida, así como también, sus respectivas funciones de pertenencia o conjuntos difusos, editando los parámetros de los mismos como tipo de conjunto, rango, etc.

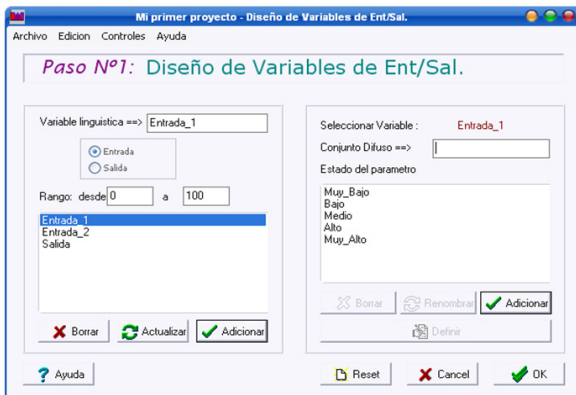


Fig. 8. Diseño de variables de entrada y salida.

La edición de las variables lingüísticas a utilizar se desarrolla en la parte izquierda de la ventana (Fig. 9). Aquí aparece la opción de seleccionar si la variable es una entrada o una salida, el rango o universo de discurso de la variable lingüística.

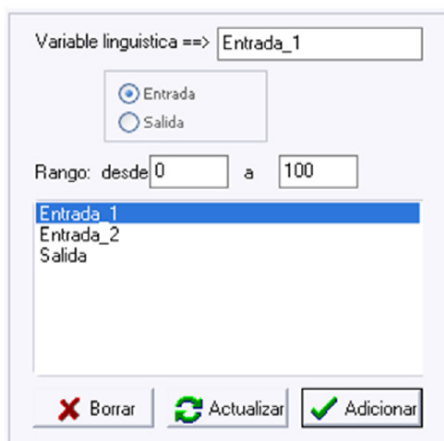


Fig. 9. Edición de variables.

En la parte de la derecha aparece la varia-

ble seleccionada, luego de seleccionarla se debe adicionar las funciones de pertenencia o conjuntos difusos que se van a utilizar en cada variables (Fig. 10).

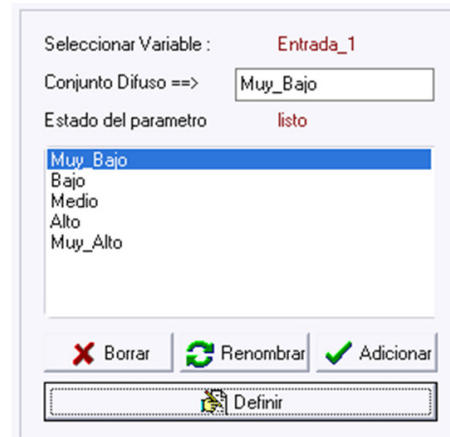


Fig. 10. Definición de Conjuntos difusos.

### 3.2.1 Formulario Principal

Luego se debe seleccionar el conjunto difuso a definir. La ventana donde se editan las funciones de pertenencia (Fig. 11), aquí vemos el nombre de la variable, el nombre del conjunto, el rango del mismo, las funciones de pertenencia o conjunto difusos que puedes escoger. Seis son las funciones de pertenencia o conjunto difuso que se pueden configurar en el software, las cuales son:

- Función Triangular.
- Función Trapezoidal.
- Función Gamma.
- Función L.
- Función Campana.
- Función Gaussiana.
- Función S.
- Función Z.

La funciones gamma y L se configurar con la función Trapezoidal mientras, que las funciones S y Z se configuran con la función Gaussiana.

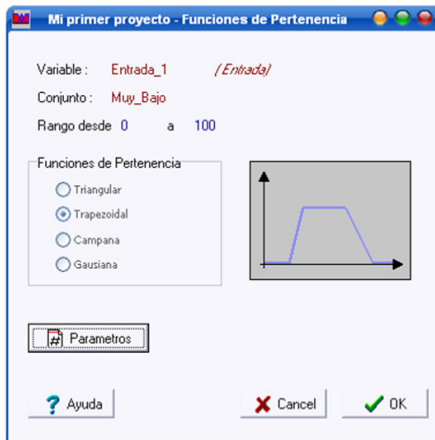


Fig. 11. Definición de funciones de pertenencia.

Luego de escoger la función de pertenencia o conjunto difuso se debe definir los valores del conjunto difuso (Fig. 12). Aquí aparecen en la parte superior los datos de la variable, su forma grafica y al lado derecho las campos para editar el valor de sus parámetros.

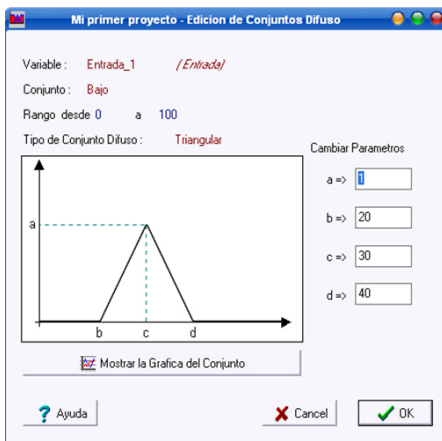


Fig. 12. Edición de conjuntos difusos.

Luego de introducir los parámetros adecuados se puede observar la gráfica del conjunto difuso (Fig. 13). Por ejemplo, para la función triangular la grafica que muestra el programa es:

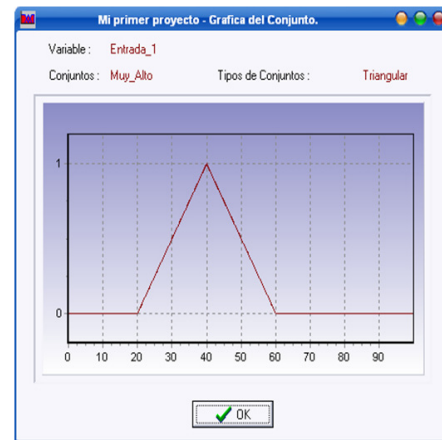


Fig. 13. Gráfica del conjunto.

En esta ventana se muestra el nombre de la variable, el conjunto de la variable, el tipo de conjunto usado y la gráfica del conjunto de acuerdo a los valores escogidos.

### 3.3 Paso N° 2: Base de Reglas

La ventana de edición de base de reglas es el paso N° 2 del diseño en nuestro programa (Fig. 14). En la parte superior nuevamente aparecen los menús de archivo, controles y ayuda. En esta ventana aparecen al lado derecho los botones necesarios para agregar, borrar, y cambiar las reglas, editar las variables anteriormente definidas, resetear los campos de la ventana y por ultimo los botones de aprobación y desaprobación de la base de reglas y salir del programa.

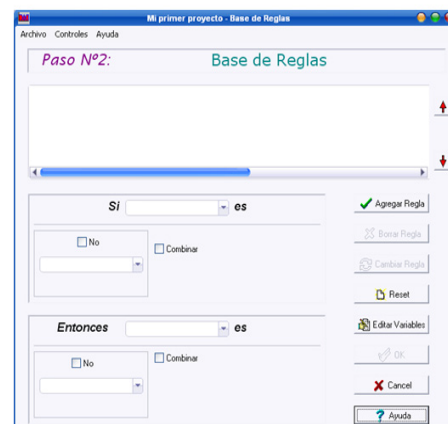


Fig. 14. Edición de base de reglas.

En FuzzyTool se tiene la posibilidad de escoger entre reglas simples definidas como:

- *Si / IF <variable de entrada> es <conjunto difuso> entonces / THEN <variable de salida> es <conjunto difuso>.*

Un ejemplo de este tipo de reglas es:

1. IF Entrada\_1 es Muy\_Bajo THEN Salida es Muy\_Lento

y reglas compuestas definidas como:

- *Si / IF <Variable de entrada> es <conjunto difuso> and / or <conjunto difuso> entonces / THEN <variables de salida> es <conjunto difuso> and / or <conjunto difuso>.*

Un ejemplo de este tipo de reglas es:

3. IF Entrada\_1 es Medio And Alto THEN Salida es Rápido Muy\_Rápido

Para el FuzzyTool el máximo de reglas permitido por proyecto es de 14, esto es debido a que para base de reglas extensas se considera que el usuario debe tener un conocimiento avanzado en lógica difusa.

### 3.4 Paso Nº 3: Inferencia de reglas

Este es el último paso en el diseño, en el cual, se infiere (razona) la base de reglas y se analizan los resultados obtenidos. Aquí aparecen dos partes; en una se evidencia la escogencia del hecho que queremos analizar y otra donde se carga una gráfica del conjunto de salida de dicha regla (Fig. 15).



Fig. 15. Inferencia de reglas.

FuzzyTool solo muestra la gráfica de la variable inferida de reglas individuales. En la parte inferior de la ventana de muestra el desarrollo y la gráfica del conjunto de salida. El método de inferencia solo se activa cuando escogemos las variables y conjuntos, luego de obtener el conjunto de salida se activa el método de defuzificación. Luego de obtener el conjunto de salida, se puede verificar el valor real (crisp) de la salida. FuzzyTool tienen tres métodos de defuzificación:

- Centro de Altura (COA).
- Media de Máximos.
- Máximo de los Máximos (LoM).

Por último, vale la pena mencionar que no se debe tomar a FuzzyTool como una herramienta de diseño de controladores difusos industriales sino, como una herramienta educativa introductoria a los sistemas difusos.

### 3.5 Requerimientos del Sistema

Los requerimientos mínimos que debe tener en equipo donde se desee correr FuzzyTool son los siguientes:

- SO Windows 98/Me/2000/XP.
- 2MBytes libres en disco duro.
- 32MBytes en RAM



#### 4. RESULTADOS

La ventana de diseño de variables lingüística le permite al usuario configurar los conjuntos difusos o funciones de pertenencia mas usados en el diseño de sistemas difusos.

Permite al usuario que no tenga un conocimiento avanzado en bases difusas, crear una base de reglas que muestre en funcionamiento del proceso a controlar.

El uso de la simulación de los conjuntos de salida de los hechos deseados permite, analizar las ventajas y desventajas de las distintas operaciones y métodos aplicados a los sistemas difusos.

#### 5. CONCLUSIONES

El desarrollo de FuzzyTool ha dejado como resultado una herramienta computacional de aprendizaje de sistemas difusos,

Cada uno de los pasos diseñados en el software orienta al usuario en la adquisición de los fundamentos necesarios para desarrollar sistemas en lógica difusa. Así, para el paso numero uno, donde el usuario edita las variables lingüísticas de entrada y sa-

lida, empieza a adquirir conocimiento en lógica difusa, tal como: que es una variable lingüística, como se utiliza, que es una función de pertenencia, como se diseña, cuales son las mas utilizadas, ventajas y desventajas de las mismas, etc. Todo este conocimiento lo adquiere el estudiante con la práctica en el software FuzzyTool.

Consecutivamente en el paso numero dos, se muestra al usuario el diseño de la base de reglas o base de conocimiento la cual, determina el razonamiento de los procesos desarrollados en el programa. Por ultimo, y luego de que el usuario escoge la base de reglas adecuada se infiere la base de reglas obteniendo los resultados gráficamente, lo cual, le permite al usuario analizar cada una de las opciones entregadas por el programa.

Aunque existen en el mercado herramientas mas potentes que FuzzyTool, este trabajo básicamente quiere demostrar la capacidad cognoscitiva de los estudiantes egresados de la universidad, capaces de adquirir conocimiento extendiendo sus horizontes de aprendizaje como una concepción general de profesionales íntegros, sin limitación en el trabajo y con la convicción de auto-aprendizaje.

#### Referencias Bibliográficas

- [1] L. Zadeh, "Fuzzy Logic". IEEE Computer, 21(4), pp. 83-92, 1988.
- [2] R. Palm, D. Driankov, H. Hellendorn, Model based fuzzy control: fuzzy gain schedulers and sliding mode fuzzy controllers. Springer-Verlag. Berlin, 1997.
- [3] H. Chung, C. Chen, J. Lin, Type fuzzy controller with self-tuning scaling factors. Fuzzy Sets and Systems 93, pp. 23-28, 1998.
- [4] B. Graham, B.P. and R.B. Newell, Fuzzy adaptive control of a first order process. Fuzzy Sets and Systems 31, pp. 47-65, 1997.
- [5] L. Zadeh, Fuzzy sets. Information and Control 8, pp. 338-353, 1965.
- [6] L.A. Zadeh, "The Concept of Linguistic Variable and its Application to Approximate Reasoning". Information Sciences, 8, pp. 199-249, 1975 (part I), 8, pp. 301-357, 1975 (part II), 9, pp. 43-80, 1976 (part III).