4. SECCIÓN DEL CONTROLADOR DIFUSO

Este capítulo abarca la parte del proyecto que hace referencia a su “cerebro”. Nos referimos al control difuso y es llamado así, debido a la capacidad de tomar decisiones y en base a ellas regir al mecanismo. Comenzaremos haciendo una breve introducción a la lógica difusa y el control difuso. Posteriormente daremos una explicación de lo que es el desarrollo en cuanto a la estructura del controlador, ya que su implementación se desarrolla en otro capítulo más adelante en el libro. Esto es, sentaremos las bases del controlador y llevaremos a cabo su simulación. Ésta última es programada en MatLab©, se menciona el procedimiento y se muestran los resultados.

4.1 Control y lógica difusa

En nuestros días una alternativa muy útil en la realización de control sobre un sistema es el control difuso. Dicha alternativa permite, mediante el conocimiento experto de una o varias personas, generar una base de conocimientos que dará al sistema la capacidad de tomar decisiones sobre ciertas acciones que se presenten en su funcionamiento. Este apartado pretende hacer una muy breve introducción al control difuso. Primero se hace una pequeña mención de sus antecedentes, con referencia en [3]. Posteriormente se habla de su base, que es la lógica difusa. Finalmente se ve de manera muy general, la estructura de un controlador difuso.

4.1.1 Antecedentes del control difuso

Las bases de la lógica difusa fueron presentadas alrededor de 1965 por Lofti Zadeh, profesor de la Universidad de California en Berkley. Contraviniendo los conceptos de la lógica clásica, donde se marca únicamente un elemento como perteneciente o no a un conjunto, propone el concepto de pertenencias parciales a conjuntos que denominó difusos. En 1974, el británico Ebrahim Mandami, demuestra la aplicabilidad de la lógica difusa en el campo del control. Desarrolla el primer sistema de control difuso práctico, la regulación de un motor de vapor.

El profesor Zadeh habla de que la gente no maneja modelos matemáticos o información cuantitativa cuando ejecuta tareas del medio que lo rodea, realizando control altamente adaptable. Por ejemplo al caminar por la calle sin chocar contra los objetos y personas, o estacionar un automóvil o jugar a balancear un péndulo invertido (escoba). Si los

controladores convencionales pudieran aceptar entradas con ruido e imprecisas, podrían trabajar de una manera más eficiente y quizá implementarse más fácilmente.

4.1.2 Lógica Difusa

La denominada lógica difusa permite a los sistemas tratar con información que no es exacta; es decir, dicha información contiene un alto grado de imprecisión, contrario a la lógica tradicional que trabaja con información definida y precisa. Como ejemplo de información que maneja la lógica difusa tenemos: estatura media, temperatura alta, etc., que en términos difusos son realmente imprecisos. La teoría de conjuntos difusos parte de la similitud con los conjuntos clásicos en los cuales se tiene una función de pertenencia de 0 ó 1. En los conjuntos difusos se tiene la característica de que la función de pertenencia puede adquirir valores en el rango de 0 a 1. Es así como se introduce el concepto de conjunto difuso, el cual se encuentra asociado con un determinado valor lingüístico que está definido por una etiqueta, palabra o adjetivo (ver figura 4.1.2-1).

Así se define una función de pertenencia de una variable para cada conjunto difuso, en la que se indica el grado en que se encuentra incluida en el concepto representado por la etiqueta. De esta manera los conjuntos difusos pueden agrupar objetos por el valor de una cierta magnitud; como por ejemplo, las personas pueden ser agrupadas por su altura como puede verse en la figura citada en el párrafo anterior:

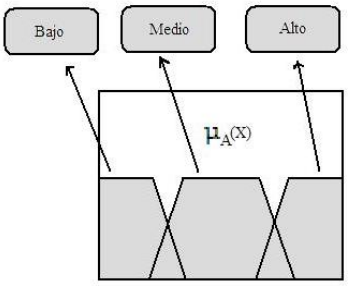

Figura 4.1.2-1
Conjuntos difusos

Además puede observarse claramente que la transición del valor del conjunto entre el cero y el uno es gradual y no cambia instantáneamente entre ellos como el caso de la lógica clásica. El concepto de función de membresía en un conjunto difuso se ilustra en el ejemplo a continuación:

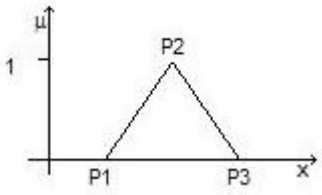

Figura 4.1.2-2
Función de membresía

Donde, según la figura 4.1.2-2, la función de membresía o pertenencia de un elemento x en el conjunto difuso triangular (A) está dado por:

$$\mu_A(x) \begin{cases} \dfrac{x - p_1}{p_2 - p_1} & p_1 < x < p_2 \\[2ex] \dfrac{p_3 - x}{p_3 - p_2} & p_2 < x < p_3 \\[2ex] 0 \text{ en otro lado} & \end{cases}$$

(1) Fórmulas para el cálculo de membresías en un cluster triangular

De esta manera, existen otras formas de conjuntos difusos, como son trapezoidales, tipo campana de Gauss, tipo S, etc. Para todos ellos, el concepto de función de membresía es el mismo; es decir, el tener un grado de pertenencia entre 0 y 1. Así, a grandes rasgos, este es el concepto básico de la lógica difusa, sus conjuntos difusos.

4.1.3 Control Difuso

La principal aplicación actual de la lógica difusa son los sistemas de control difuso, que utilizan las expresiones difusas para formular las reglas que controlarán dichos sistemas. Como la lógica difusa sugiere un cierto grado de pertenencia para un dato que se presente dentro de los conjuntos difusos, permite a un controlador difuso tomar diferentes grados de acción en un sistema. En los sistemas de control debe tomarse en cuenta el conocimiento experto de una o varias personas para la realización de la base de conocimientos sobre la cual se basará la toma de decisiones.

El control difuso puede aplicarse en innumerables sistemas, tanto sencillos, como brazos articulados y vehículos autónomos, en los cuales los modelos matemáticos son muy

complejos; así, empleando técnicas de razonamiento aproximado es posible controlar sistemas superiores cuando el entorno no se conoce de forma precisa. Dicha característica permite mayor flexibilidad que el control clásico en el que para la realización de un controlador se requiere de un alto grado de cálculo matemático. Así, al desarrollar un controlador difuso es posible prescindir de la rigidez matemática y transmitir el raciocinio humano hacía un sistema.

La estructura normalmente utilizada en un controlador difuso puede verse en la figura 4.1.3-1.

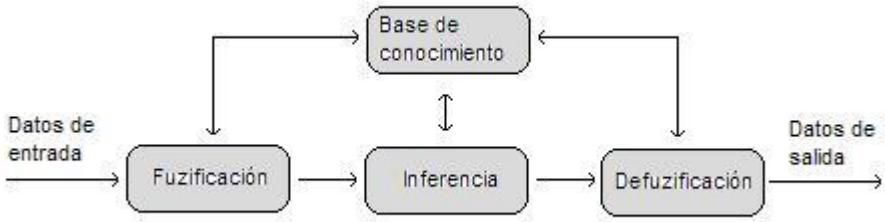

Figura 4.1.3-1
Estructura de un controlador difuso

En la que se puede ver un primer bloque llamado fuzificador en el que los datos de entrada son procesados para calcular el grado de membresía que tendrán dentro del controlador. Posteriormente se tiene el dispositivo de inferencia que junto con la base de conocimientos realizan la toma de decisiones que dictarán la forma en que actuará el sistema. El método de inferencia se basa en el grado de pertenencia de los datos de entrada en los conjuntos difusos de los espacios correspondientes, siempre que éstas se den, para tomar una decisión en el espacio de salida. El conjunto de reglas, que son la base de conocimiento, es del tipo antecedente – consecuente; es decir: *si… entonces…*, y existen distintos métodos de

llevarla a cabo (mínimo-máximo, máximo-producto, etc.). La última etapa que se tiene dentro del controlador es el defuzificador, que es quien realiza el procesado final con el fin de adecuar los valores difusos obtenidos de la inferencia en valores no difusos útiles para el proceso que se ha de controlar, como ejemplos tenemos el método del centro de área, de la máxima pertenencia, etc. De ésta manera obtenemos un panorama general de como se estructura un controlador difuso.

4.2 Desarrollo del controlador difuso

En esta sección se describe el procedimiento que se siguió para estructurar el controlador difuso, la implementación de él, se desarrolla en la sección 5.5.

4.2.1 Definición de variables

Comenzamos definiendo las variables de salida del sistema, sobre la base de ellas se estructuran las de entrada. Las salidas están dadas directamente por el planteamiento de las funciones del mecanismo. El sistema golpea la bola de billar a distinta altura (posición en el eje Y) y con distinta fuerza; esto nos da como variables de salida la **fuerza de golpeo** y la **altura**. Para definir las entradas, tiene que plantearse que tipo de variables se controlan en el juego, aplicando las variables de salida del mecanismo. Es decir; que puede hacer un jugador si sólo pudiera golpear la bola a lo largo de su eje perpendicular a la mesa y con distinta fuerza. Se definen entonces dos variables de entrada. La primera es **distancia**, por ella entendemos la distancia que debe viajar la bola que se golpea con el taco para lograr obtener la carambola. La segunda es **reacción**, por ella entendemos a lo que hará la bola que fue golpeada, al tener su primer choque con otra bola. La división de estas variables, se realiza en el siguiente apartado.

4.2.2 Clusterización de los espacios de entrada y salida

Se dividieron, tanto los espacios de entrada como los de salida, en cinco clusters triangulares. El número obedece a las etiquetas lingüísticas para las variables, estas etiquetas fueron asignadas de acuerdo a las posibilidades del juego de carambola y se definen más adelante. La forma de los clusters obedece a que fue a nuestro juicio la más indicada, ya que se va a realizar la implementación en un microcontrolador y los conjuntos triangulares presentan mayor facilidad en su representación, manejo y evaluación y son eficientes para realizar un controlador. Las variables finalmente tienen un rango de 0 a 252 con una partición equidistante; esto es conveniente, ya que se utilizó una resolución para los espacios en el microcontrolador de 8 bits. Es importante señalar que en el proceso se variaron estos rangos y divisiones, así como que el programa implementado en hardware esta diseñado para realizar estos cambios sin alterar su funcionamiento.

La partición de los espacios de entrada se llevó a cabo de la siguiente manera. Para el caso de la reacción, las etiquetas lingüísticas son los efectos posibles: “regresa mucho”, “regresa poco”, “detenerse”, “avanza poco” y “avanza mucho”, como se ve en la figura 4.2.2-1.

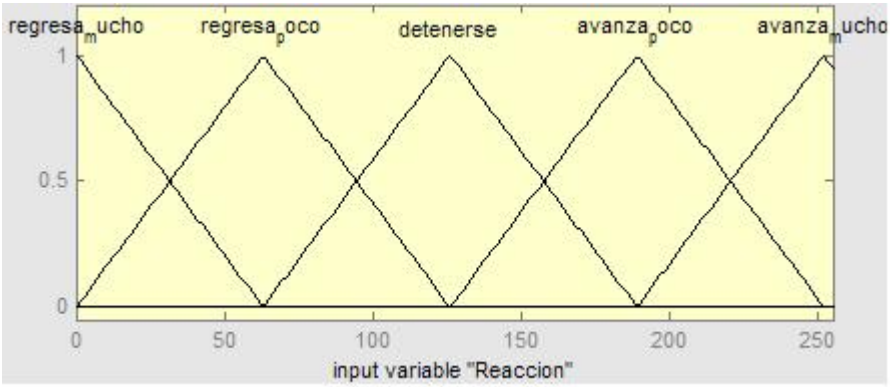

Figura 4.2.2-1
Clusterización de “Reacción”

Para una mejor ilustración del significado de estas etiquetas, se presenta la figura 4.2.2-2 y su explicación:

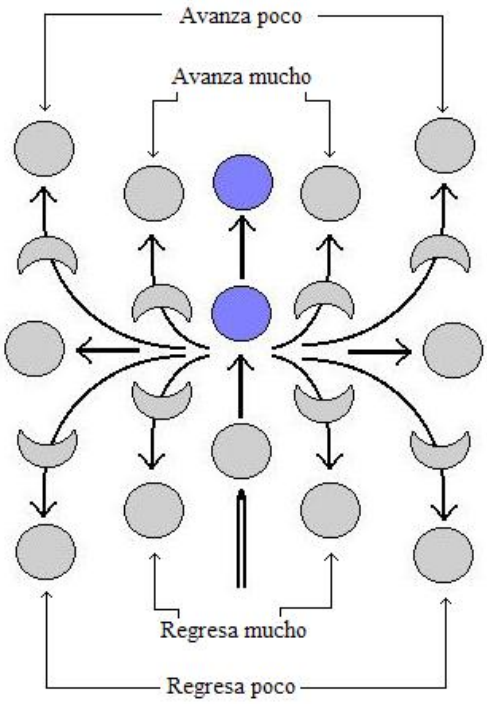

Figura 4.2.2-2
Significado de “Reacción”

La reacción de “detenerse”, significa que la bola tiradora al golpear la otra, detendrá su movimiento, como sabemos en el billar, este tiro es sólo posible en una colisión de frente; por ello, no nos referimos directamente al hacer alto, sino al no tener efecto. Para el resto de los efectos, podemos observar en la figura, que los significados de “regresa” y “avanza” son obvios, limitándonos a describir el “mucho” o “poco” del que hablamos. Concretamente, estos adjetivos hacen referencia a la cantidad de efecto que tiene la bola tiradora, no hay que confundir con la distancia, que es dada por la otra variable de entrada. Por ejemplo, al decir “avanza mucho”, significa que la bola, después de golpear a la otra, tenderá a avanzar inmediatamente después de cambiar su trayectoria producto de la colisión y si nos refiriéramos a “avanza poco”, la bola tardaría más en desviarse hacia delante de su trayectoria normal después de la colisión.

En el caso de distancia se divide en: “muy cerca”, “cerca”, “medio”, “lejos” y “muy lejos”. En la figura 4.2.2-3, podemos ver esta partición.

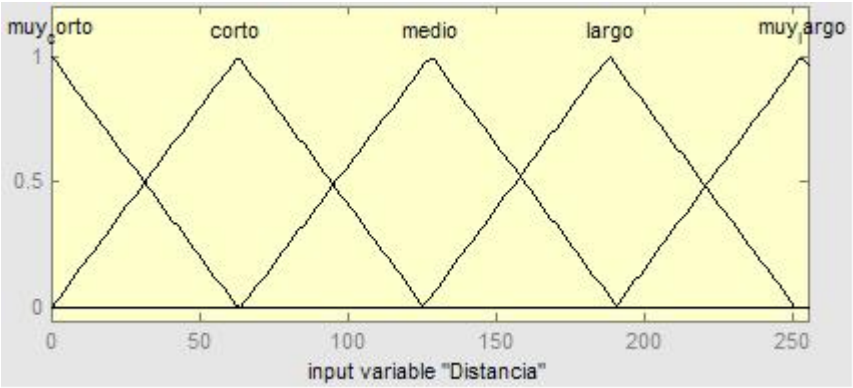

Figura 4.2.2-3
Clusterización de “Distancia”

La partición de los espacios de salida quedó como se describe a continuación. Para la variable de fuerza (figura 4.2.2-4): “muy despacio”, “despacio”, “medio”, “fuerte” y “muy fuerte”.

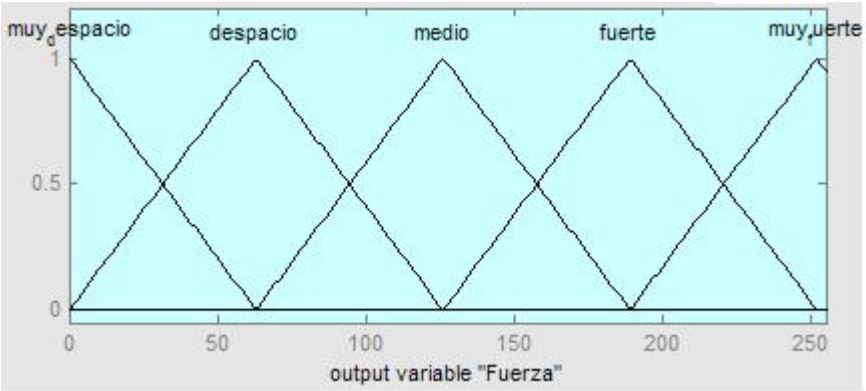

Figura 4.2.2-4
Clusterización de “Fuerza”

Para la variable de altura (figura 4.2.2-5): “muy abajo”, “poco abajo”, “centro”, “poco arriba” y “muy arriba”.

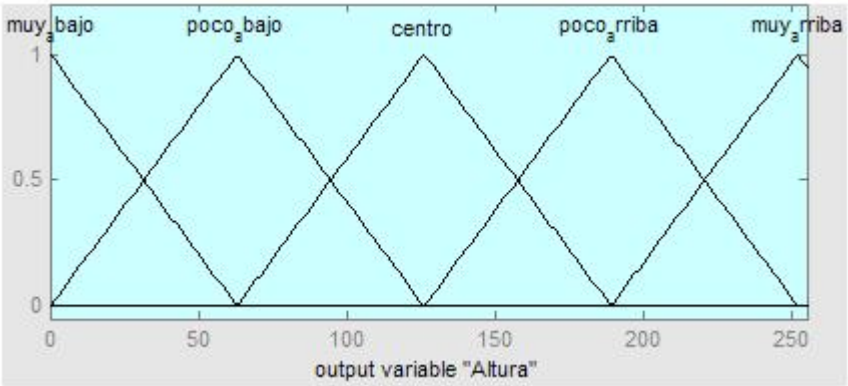

Figura 4.2.2-5
Clusterización de “Altura”

4.2.3 Fuzificación de las variables de entrada

Las entradas al controlador difuso son valores discretos en el rango mencionado (de 0 a 252). Con cada una de las variables de entrada, el valor recibido al controlador es

comparado con su respectivo espacio; así obtenemos la información de a que cluster pertenece. Posteriormente, se calcula la membresía del valor de cada entrada en cada uno de los clusters que tenga pertenencia. En la figura 4.2.3-1 observamos este concepto de manera gráfica, únicamente como ejemplo a lo anterior; la línea roja representa la variable y la parte sombreada, representa el grado de pertenencia a cada cluster.

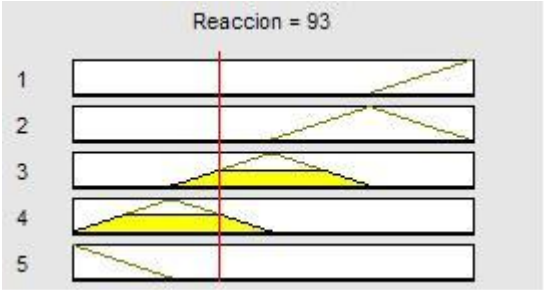

Figura 4.2.3-1
Valor de entrada para fuzificar

En el diagrama a bloques (figura 4.2.3-2) puede apreciarse el funcionamiento del fuzificador.

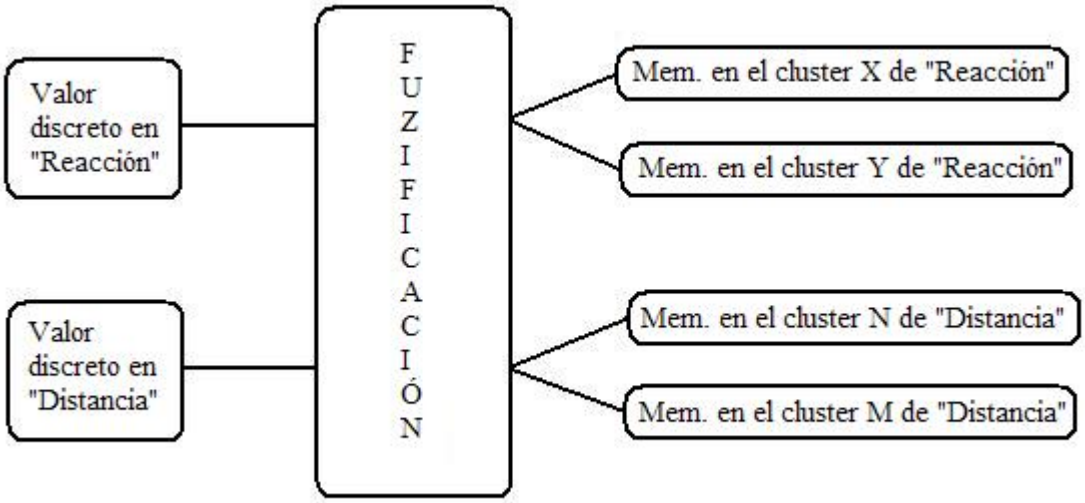

Figura 4.2.3-2
Funcionamiento del fuzificador

Recibido un valor de entrada para “Distancia” y uno para “Reacción”, devuelve el grado de pertenencia de cada uno a su respectivo espacio, en los clusters que abarquen al valor. Debido a la partición simétrica de los espacios, la variable sólo pueda pertenecer a uno o dos clusters, resultando en una o dos funciones de pertenencia por espacio únicamente.

4.2.4 Base de conocimiento

Esta sección describe la parte central del controlador difuso, nos referimos a la base de conocimientos, punto de partida para la generación del conjunto de reglas sobre las que se rige la inferencia. Se describen los términos usados y se muestra la base en forma de matriz de asociación difusa.

Para el desarrollo de la base de conocimientos, fue necesario hacer una investigación de campo. Ésta nos permitió confirmar los conocimientos empíricos para lograr tener un respaldo adecuado y que el controlador se basará en la mejor información posible. Se desarrollaron dos bases de conocimiento distintas, ya que al tener dos salidas es necesario un controlador por cada una, aún cuando dependan de las mismas entradas. Definimos:

En las entradas:

Distancia	Reacción
MC = Muy corto	AM = Avanza mucho
C = Corto	AP = Avanza poco
M = Medio	D = Detenerse
L = Largo	RP = Regresa poco
ML = Muy largo	RM = Regresa mucho

En las salidas:

Fuerza	Altura
MD = Muy despacio	MA = Muy abajo
D = Despacio	PA = Poco abajo
M = Medio	C = Centro
F = Fuerte	PAr = Poco arriba
MF = Muy fuerte	MAr = Muy arriba

Las matrices se muestran en las tablas 4.2.4-1 y 4.2.4-2.

ALTURA		DISTANCIA ENTRE BOLAS				
		MC	C	M	L	ML
EFECTO	AM	MAr	MAr	MAr	MAr	MAr
	AP	PAr	PAr	PAr	PAr	PAr
	D	C	C	C	C	C
	RP	PA	PA	PA	PA	PA
	RM	MA	MA	MA	MA	MA

Tabla 4.2.4-1
Base de conocimientos de la altura

FUERZA		DISTANCIA ENTRE BOLAS				
		MC	C	M	L	ML
EFECTO	AM	D	D	M	F	MF
	AP	D	M	M	F	MF
	D	D	M	M	F	MF
	RP	F	F	MF	MF	MF
	RM	F	MF	MF	MF	MF

Tabla 4.2.4-2
Base de conocimientos de la fuerza

Las matrices son una forma sencilla de ver las reglas. De esta observación, podemos concluir claramente que en el caso correspondiente a la “Altura”, la variable “Distancia”, no tiene influencia en los consecuentes, ya que las diferentes consecuencias son regidas únicamente por la variable “Efecto”. De esta forma tenemos elaborada la base reglas que son aplicadas al controlador y permiten la decisión de los diferentes tipos de tiros.

4.2.5 Proceso de inferencia

El proceso de inferencia utilizado es el de mínimo – máximo, también conocido como Mandami; ya que por sus características es de los métodos más apropiados para realizar un controlador compilado. Este método se basa en lo siguiente:

- *Valores de entrada:* El resultado de la fuzificación arroja ciertos valores de membresía para distintos clusters en cada espacio de entrada.

Ejemplo: $\mu_{A1}(x) = 0.7$; Valor de membresía de la variable x en la entrada A, cluster 1

$\mu_{A2}(x) = 0.3$; Valor de membresía de la variable x en la entrada A, cluster 2

0 para los demás clusters del espacio de entrada A

$\mu_{B1}(y) = 0.4$; Valor de membresía de la variable y en la entrada B, cluster 1

$\mu_{B2}(y) = 0.6$; Valor de membresía de la variable y en la entrada B, cluster 2

0 para los demás clusters del espacio de entrada B

- *Aplicación de reglas:* Las pertenencias que existen se aplican sobre la base de reglas para saber en que cluster del espacio de salida se produce el consecuente de cierta combinación de antecedentes descritos en dichas reglas.

Ejemplo: R1: Si A1 y B2 entonces C3 ←

R2: Si A3 y B1 entonces C2

.

.

.

Rn: Si A2 y B2 entonces C5 ←

En general:

Regla#: Si Xn y Ym entonces Cz

Leyendo las reglas como: Si hay membresía en el cluster "n" del espacio "X" y en el cluster "m" del espacio "Y", entonces se da en el cluster "z" del espacio de salida "C".

En este ejemplo, se dan las reglas señaladas (R1 y Rn)

- *Membresías de salida:* El valor de membresía que se hereda al cluster de salida en cada regla cumplida, es el valor mínimo de las membresías de los clusters de los espacios de entrada que se involucran en la regla.

Ejemplo: $\mu_{C3}(z) = 0.6$; Valor de membresía de la variable z en la salida C, cluster 3

$\mu_{C5}(z) = 0.3$; Valor de membresía de la variable z en la salida C, cluster 5

- *Formación del polígono:* Después de aplicar todas las reglas que se cumplen, podemos tener varios valores de membresía para un mismo cluster del espacio de

salida, si varias reglas heredaron en él. Adicionalmente es necesario formar el polígono de salida, que refleja el valor de las membresías a lo largo de los clusters del espacio de salida. Para obtener el producto final del proceso de inferencia, se debe entonces hacer un barrido por cada cluster, tomando siempre el máximo valor de membresía que presente en cualquier regla. Entre clusters, se debe tomar también como el valor de membresía, la mayor de cada uno.

La figura 4.2.5-1 muestra un ejemplo de lo descrito en este último punto.

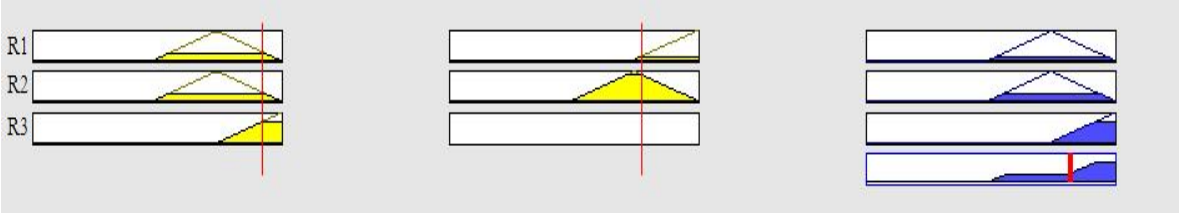

Figura 4.2.5-1
Formación del polígono de salida

En esta figura, podemos observar la formación del polígono de salida, así como también nos da una idea de los puntos anteriores. En ella, de cada renglón se toma la menor membresía de los espacios de entrada para heredar al de salida (mínimo). La última columna muestra la integración del polígono de salida, con los máximos de los mínimos. De esta manera queda conformado el método de inferencia del máximo - mínimo. A continuación se muestra gráficamente este método de inferencia (figura 4.2.5-2), con la finalidad de facilitar su comprensión.

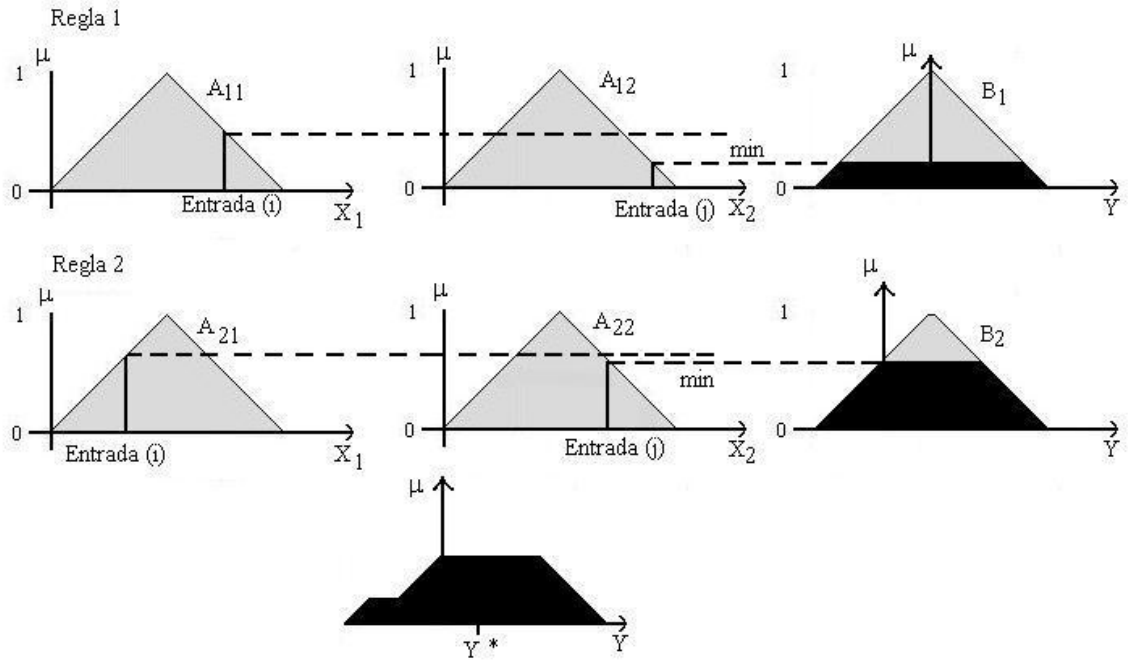

Figura 4.2.5-2
Método de inferencia máximo – mínimo (Mandami)

4.2.6 Defuzificación

Como última parte de los procesos del controlador difuso, se lleva a cabo la defuzificación. En este proyecto, el método utilizado es el del centroide, su elección se debe a la misma razón que el método de inferencia elegido; es decir, practicidad para ser implementado. Es importante mencionar, que no por ello, se compromete de ninguna manera su eficiencia, ya que estos métodos son también muy confiables.

El proceso de defuzificacón por centro de área o centroide es simple, dado el polígono de salida por la el proceso de inferencia, simplemente se calcula el centro de gravedad de la figura, mediante la fórmula (2).

$$z^* = \frac{\int \mu_c(z)\, z\, dz}{\int \mu_c(z)\, dz}$$

(2) Fórmula para el cálculo del centroide

Basado en la figura 4.2.6-1, donde z^* es el centroide.

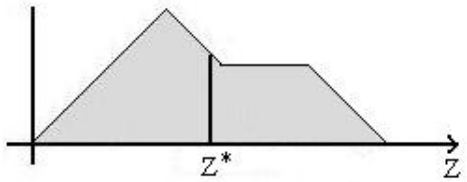

Figura 4.2.6-1
Método del centroide

O para el caso de sistemas discretos, se calcula el centroide discretizado con la fórmula (3).

$$z^* = \frac{\sum \mu_c(z)\, z}{\sum \mu_c(z)}$$

(3) Fórmula para el cálculo del centroide discretizado

Esta fórmula es la que se implementa en la programación del control difuso del proyecto.

Siguiendo la serie de procedimientos listados, se tiene la estructura del controlador difuso.

4.3 Simulación en MatLab©

En esta parte del proyecto se lleva a cabo una simulación del comportamiento del controlador difuso realizado. Esto con el fin de conocer si realiza las acciones que de él se esperan; es decir si tiene el comportamiento deseado. Existen varios programas que llevan a

cabo la simulación de control difuso; en este caso se utiliza el MatLab©. Este programa posee varias cajas de herramientas que abarcan diversas áreas de la computación técnica, lo que lo convierte en un auxiliar importante en este tipo de tareas. Una de estas herramientas es el "Fuzzy Logic Toolbox", éste permite realizar simulaciones del funcionamiento del controlador difuso.

La programación en esta herramienta es muy amigable, cuenta con una interfaz totalmente gráfica, que permite introducir las variables del controlador en una forma sencilla. A continuación se presenta una explicación de cómo se introducen los datos al simulador, acompañado de imágenes que muestran el procedimiento.

Se debe tomar en cuenta que se crearon dos controladores, uno para cada variable de salida ("Fuerza" y "Altura"), ambos se simulan pero el procedimiento es el mismo. El primer paso es definir el método del controlador entre el de Sugeno y Mandami, se eligió el de Mandami porque es el que se esta utilizando en la realización del proyecto.

En las figuras 4.3-1 y 4.3-2 se muestra la interfaz en la que se introducen las variables.

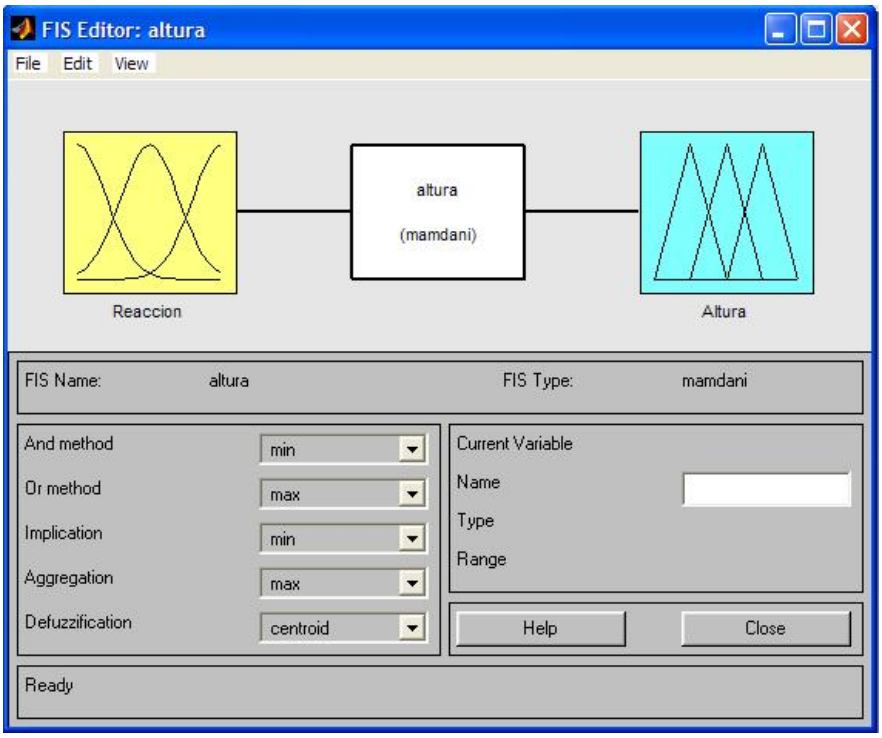

Figura 4.3-1
Declaración de la variable "Altura"

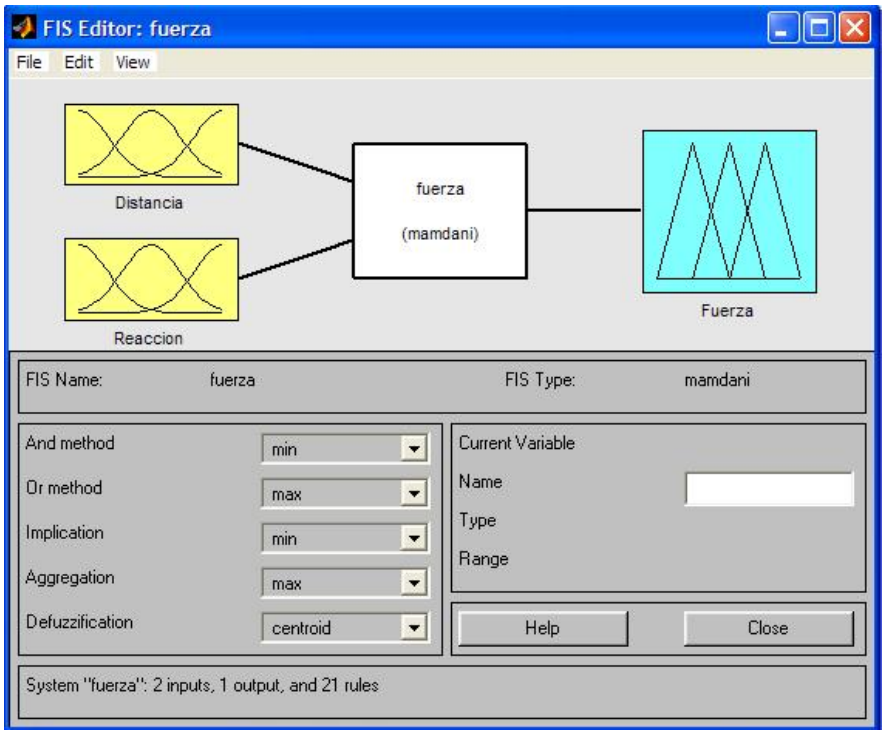

Figura 4.3-2
Declaración de la variable "Fuerza"

En esta misma interfaz se hace la definición de las variables tanto de entrada como de salida; y aunque se puede definir más de una salida se simularán por mayor orden y simplicidad de manera separada. Adicionalmente se define el tipo de disyunción, conjunción, implicación, agregación y defuzificación.

El segundo paso es definir tanto los espacios de entrada como los de salida. Esto se realiza en una nueva interfaz (figura 4.3-3 y 4.3-4) en los que se define el rango de los espacios, el número y el tipo de clusters en los que éstos se dividen. Se definen también las etiquetas de cada cluster, así como los parámetros de cada uno (puntos numéricos que marcan los límites de los clusters).

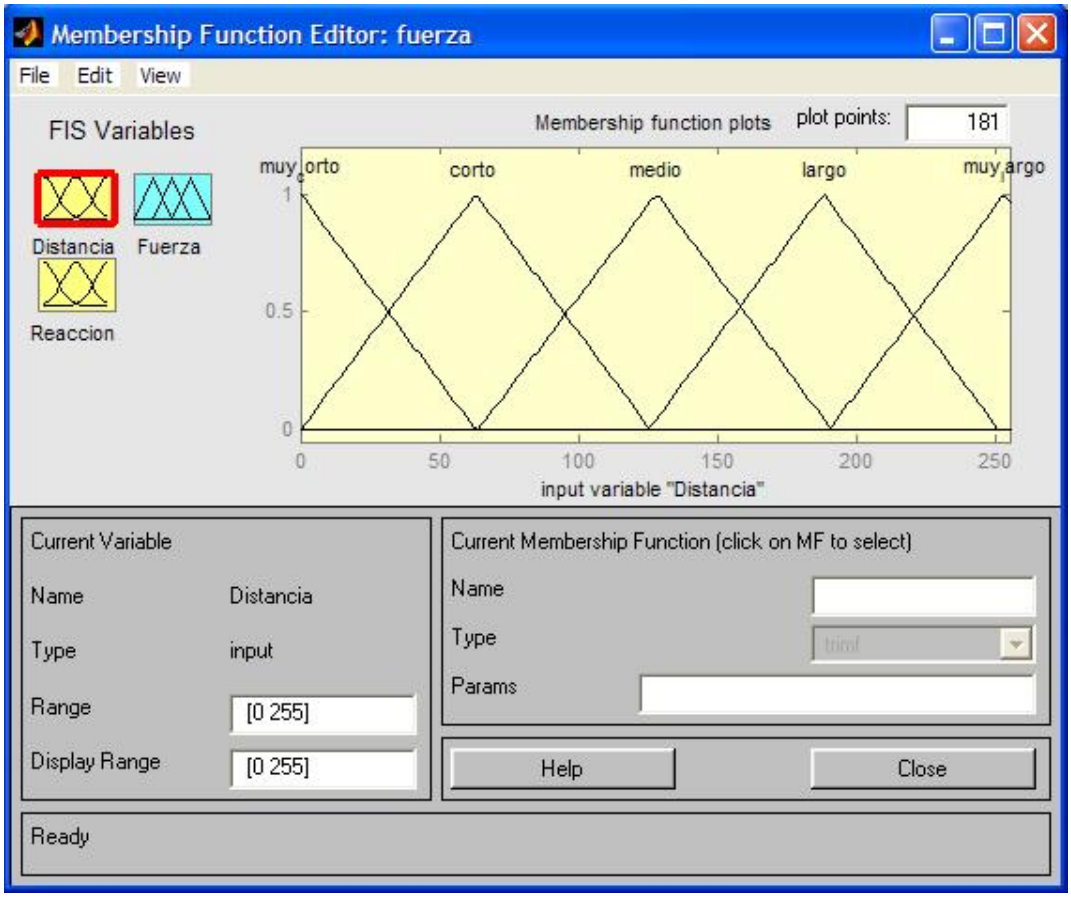

Figura 4.3-3
Variable de entrada “Distancia”

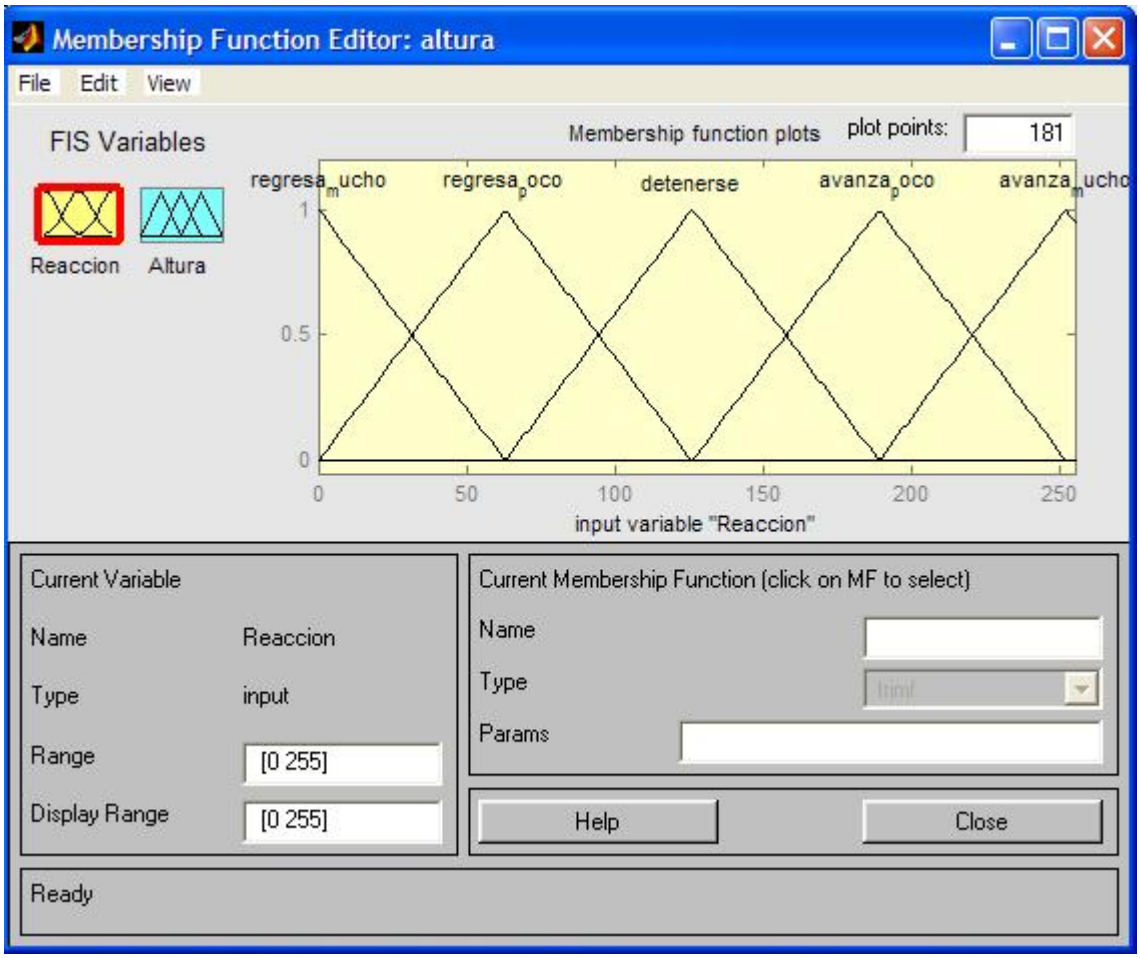

Figura 4.3-4
Variable de entrada "Reacción"

El siguiente paso es editar las reglas que dan forma al controlador; una vez más en una nueva interfaz (figura 4.3-5 y 4.3-6). En ella se presentan las variables definidas en la etapa anterior, así que basta con seleccionarlas junto a las condiciones que las relacionan, apegadas a la base de conocimiento del ambiente gráfico y una vez hecho esto las reglas se van agregando de manera sencilla.

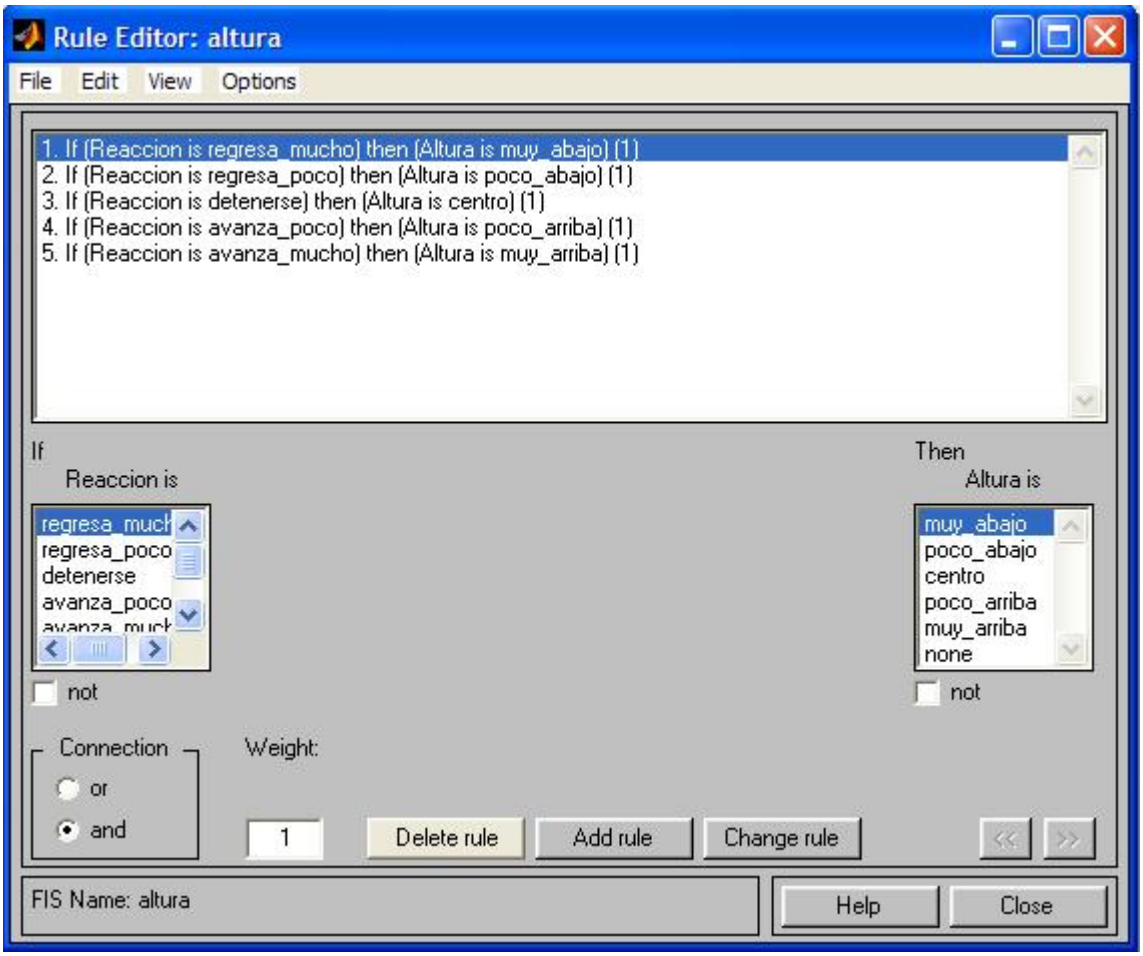

Figura 4.3-5
Editor de reglas para “Altura”

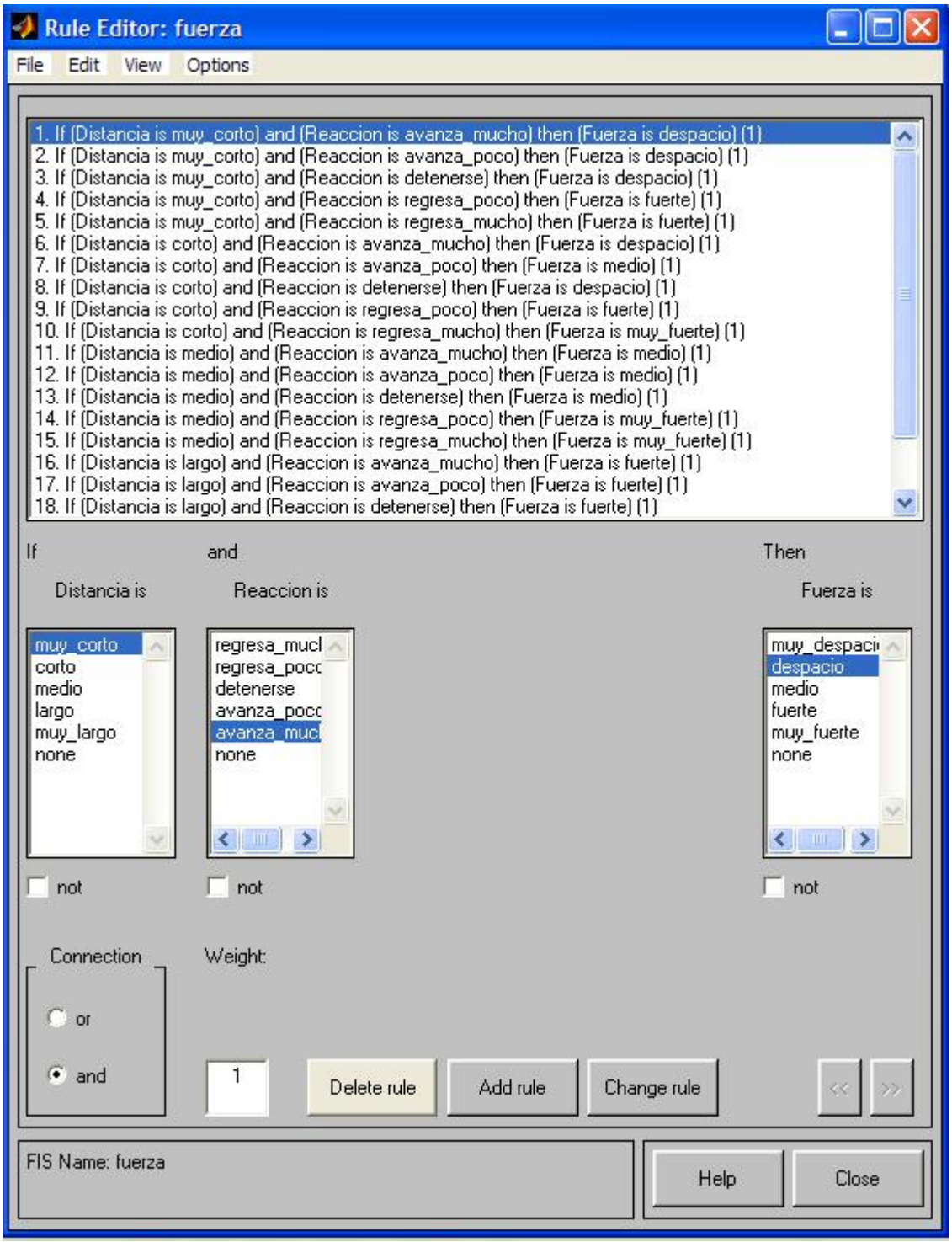

Figura 4.3-6
Editor de reglas para “Fuerza”

Los pasos anteriores son los únicos necesarios para configurar la simulación. Lo único que se necesita ahora es pedirle al programa que muestre las reglas, lo cual hace en una nueva ventana que se muestra en la siguiente serie de figuras. En ellas se aprecia la naturaleza gráfica como es mostrado el comportamiento del controlador en el simulador.

El programa despliega el conjunto de reglas. Existen dos formas de ingresar los valores de entrada; una en el cuadro de diálogo correspondiente o simplemente dando clic con el ratón sobre las representaciones gráficas de las entradas en un punto deseado. Las salidas se muestran en la columna de la extrema derecha, donde se aprecian claramente las reglas cumplidas y en la parte inferior de esta columna el polígono resultante de la inferencia. En las figuras 4.3-7 y 4.3-8 se muestra el resultado de la defuzificación tanto de forma gráfica con una línea roja como numérica en la parte superior de la columna.

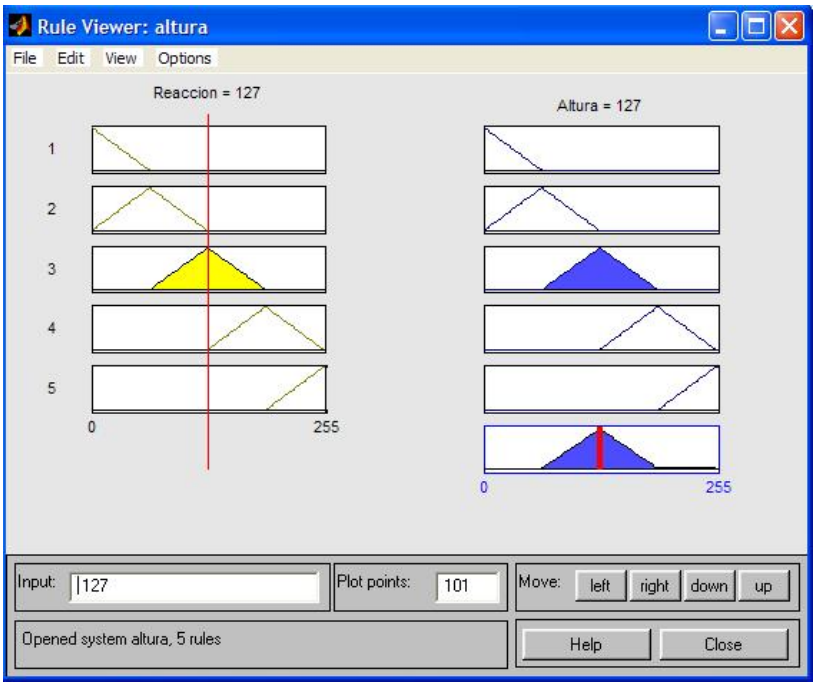

Figura 4.3-7
Salida de la variable "Altura"

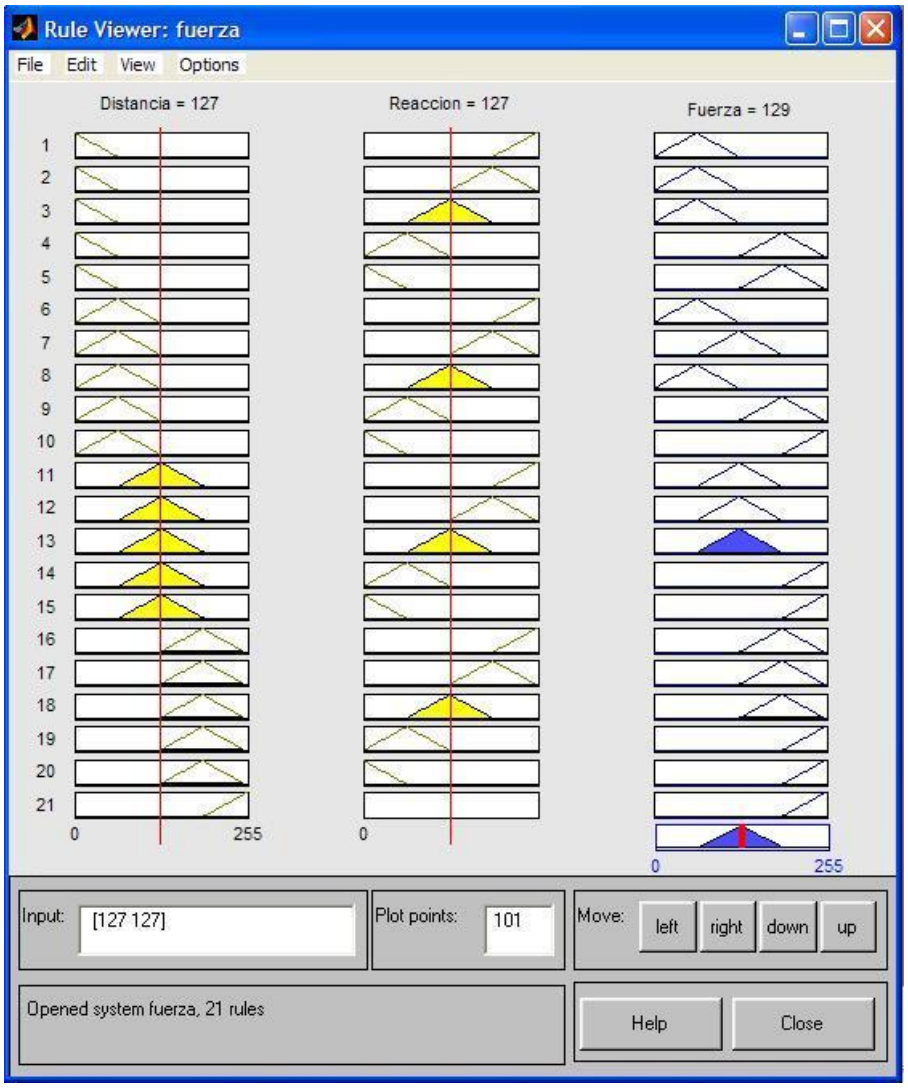

Figura 4.3-8
Salida de la variable "Fuerza"

Como resultado al procedimiento descrito en este capítulo, previa recopilación del conocimiento experto, se tiene conformado el controlador difuso. Realizada su simulación, podemos tener la seguridad de que funciona de acuerdo a lo esperado. Esto nos permite realizar la programación del controlador difuso en el microcontrolador (sección 5.5) sobre una base confiable.