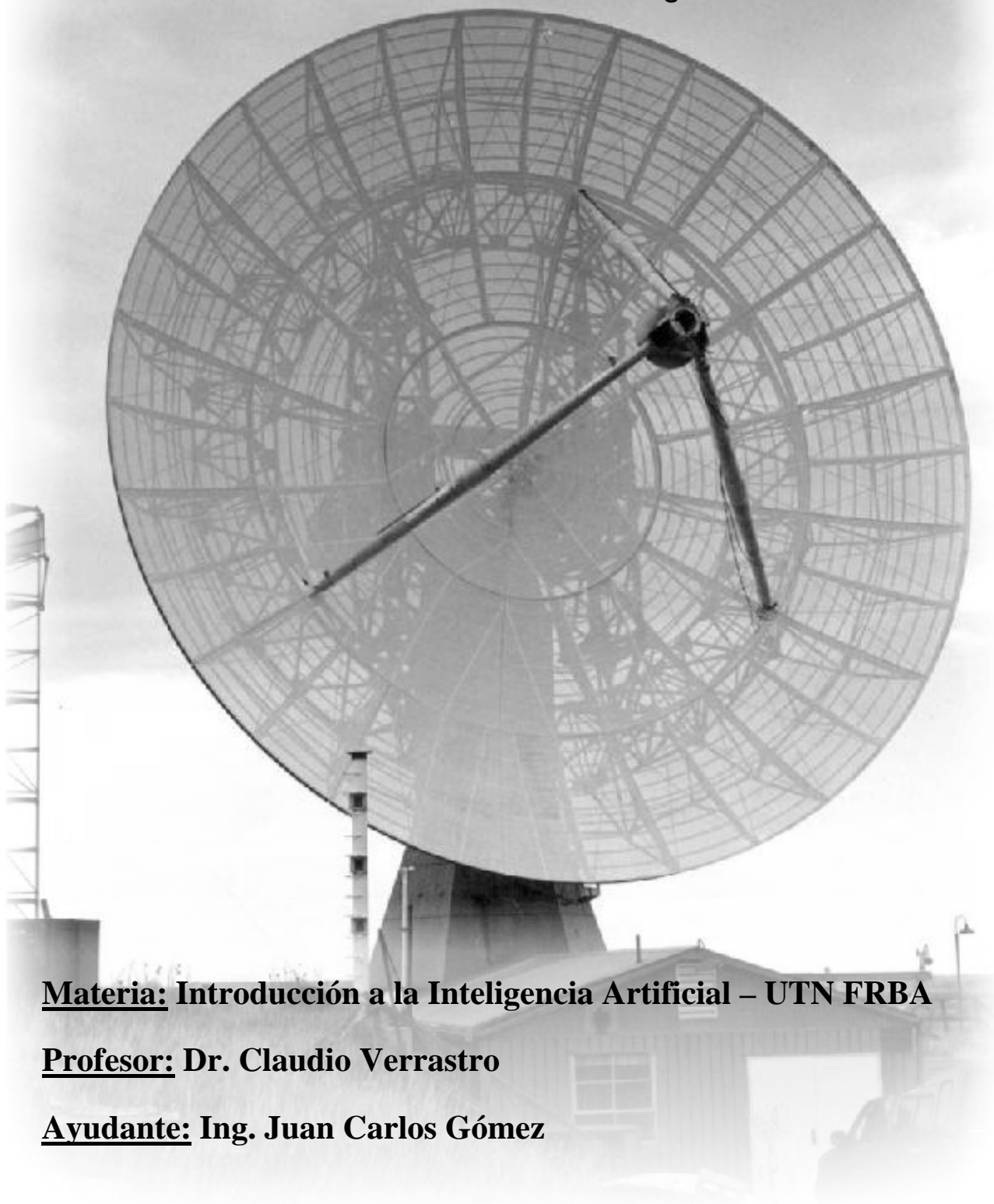


Trabajo Práctico N^{ro} 2 : Implementación de Fuzzy Logic con CubiCalc

“Modelo de un radar de seguimiento”



Materia: Introducción a la Inteligencia Artificial – UTN FRBA

Profesor: Dr. Claudio Verrastro

Ayudante: Ing. Juan Carlos Gómez

Alumno: Gustavo Damián Gil

Fecha: 18-07-07

Índice de contenidos:

<u>Enunciado de la Problemática y Marco Teórico</u>	3
<u>Reglas de activación utilizadas</u>	8
<u>Gráficas de errores en estado estacionario</u>	9
<u>Conclusiones para la implementación del sistema</u>	10
<u>Detalle del programa GUS6.CBC y GUS7.CBC</u>	11
 <u>Anexo:</u>	
<u>¿Que es la lógica difusa?</u>	12

Enunciado de la Problemática:

Se debe implementar un sistema de control difuso para una aplicación específica, simulando dicha aplicación con su controlador difuso empleando la herramienta CubiCalc 2.0 .

Para realizar la práctica propuesta, se planteó por mi parte al Dr. Verrastro realizar el controlador del pedestal de un radar de seguimiento (tracking) del tipo Height Finder, el cuál sirve para realizar la medición en altura (elevación) y dirección (azimuth) del los blancos (targets), pero no su distancia al radar. Se emplea como radar secundario para guiar misiles.

Según las recomendaciones del docente, se realizó el modelo matemático de la planta lo mas simplificado posible, y empleando el mismo modelo de planta tanto para los movimientos el eje X (elevación) como para el eje Y (azimuth).

{ Planta de mi radar de seguimiento:
=====

Modelo del sistema de posicion del radar en una dimension:

$$\bar{F}(t) = M \cdot \bar{a} + B \cdot \bar{v}$$

$$\bar{F}(t) = M \cdot \ddot{x} + B \cdot \dot{x}$$

$$F(s) = M \cdot X(s) \cdot S^2 + B \cdot X(s) \cdot S$$

$$F(s) = X(s) \cdot (M \cdot S^2 + B \cdot S)$$

F=fuerza

M=masa

a=aceleracion

B=coeficiente de viscosidad

v=velocidad

X(s)	1
F(s)	S.(S.M+B)

NOTA: tanto X(s) como F(s) son
pequeñas variaciones de
distancia y fuerza.

Aquí podemos apreciar la trasferencia en el dominio de Laplace, de nuestra planta.

Recordando la clasificación que describe a los sistemas a lazo abierto en régimen estacionario, podemos asegurar que es un sistema “tipo 1”, ya que posee un polo simple en el origen. Lo que acarrea entre sus consecuencias, que ante una entrada del tipo escalón unitario ($1 / S$), éste tipo de sistemas no presenta error estacionario, y frente a una entrada del tipo rampa ($1 / S^2$), dicho sistema si presenta error estacionario.

Mas adelante, veremos gráficamente en el CubiCalc, como se puede percibir éste efecto.

Es importante destacar que cuando se realiza tracking, siempre se emplean sistemas “tipo 1” y contadas veces “tipo 2”, ya que los sistemas “tipo 0” pese a ser mas simples, tienen error en estado estacionario frente a una entrada del tipo escalón, pero por sobre todas las cosas, tienen grandes tiempos de respuesta, ya que son de respuesta lenta.

Los sistemas “tipo 2” son mas veloces que los “tipo 1”, pero son mas propensos a las oscilaciones y no es algo deseado para nuestra aplicación en particular.

Ahora descomponemos la funcion tranferencia en facciones parciales para luego antitransformar y obtener la tranf. en el dominio del tiempo.

$$\frac{X(s)}{F(s)} = \frac{1}{S.M.(S+B/M)} = \frac{K1}{S.M} + \frac{K2}{(S+B/M)}$$

$$K1 = \lim_{s \rightarrow 0} \frac{1}{(S+B/M)} = 1/B$$

$$K2 = \lim_{s \rightarrow -B/M} \frac{1}{S.M} = -1/B$$

$$T(s) = \frac{X(s)}{F(s)} = \frac{1/B}{S.M} + \frac{-1/B}{(S+B/M)}$$

$$T(t) = 1/(B.M).u(t) - 1/B.e^{-B.t/M}.u(t)$$

$$\boxed{T(t) = (1/(B.M) - 1/B.e^{-B.t/M}).u(t)}$$

Tranferencia en el dominio del tiempo.

Modelo del controlador difuso: (le ingresamos un delta X , y le sacamos un delta F)
=====

X aprox. 0	=>	Fuerza aprox. 0
X > 0	=>	Fuerza > 0
X < 0	=>	Fuerza < 0

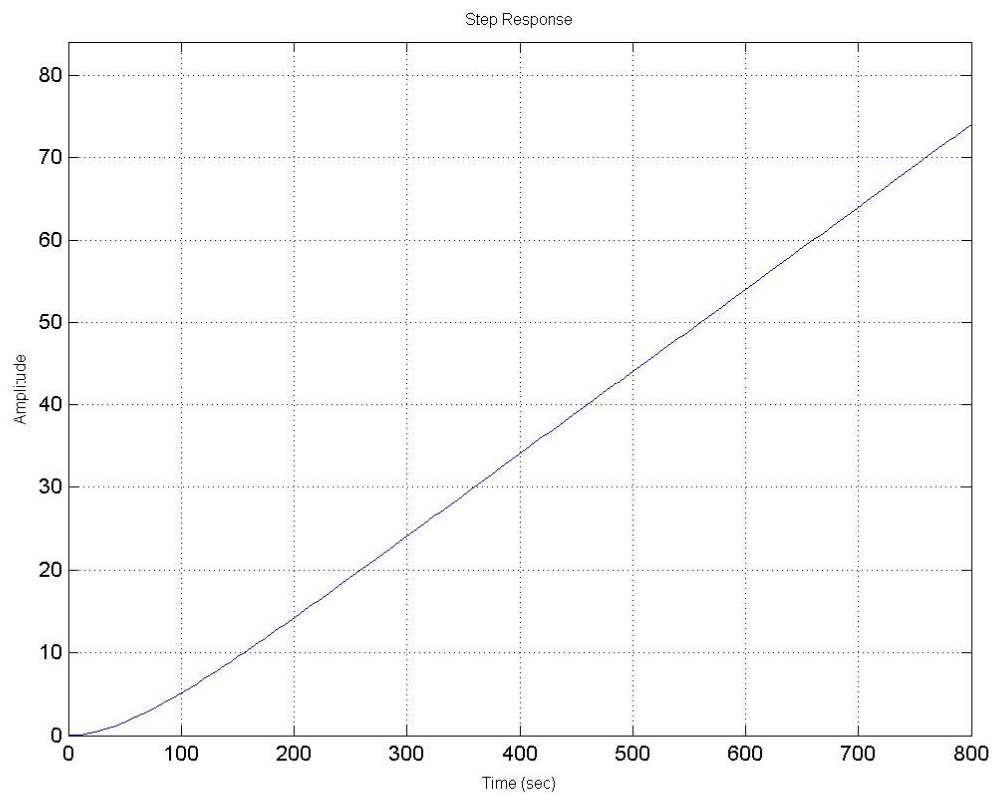
Como se puede apreciar, estas cuestiones se encuentran detalladas en el programa realizado en CubiCalc, para tener de un vistazo toda la información necesaria para modificarlo.

Ahora graficaremos la respuesta a una entrada tipo escalón unitario al modelo de nuestra planta con Matlab, para verificar que es un sistema "tipo 1".

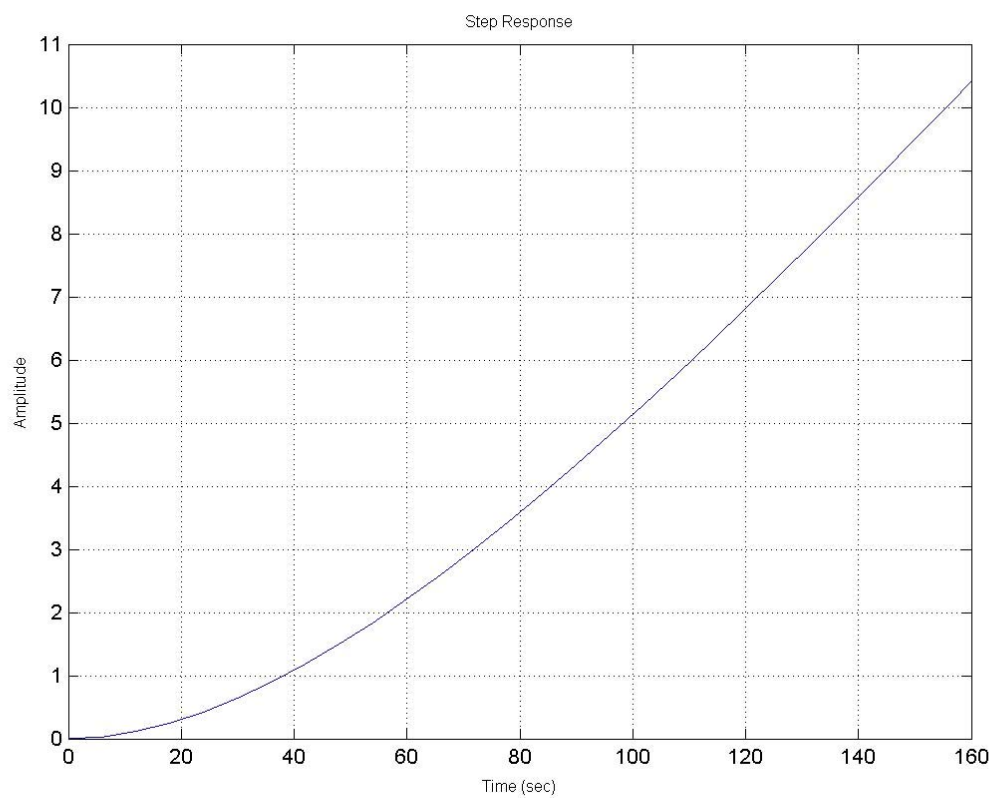
```
>> n=[1];
```

```
>> d=[600 10 0]; % coeficientes del polinomio M.s^2+B.s+0
```

```
>> step(n,d)
```

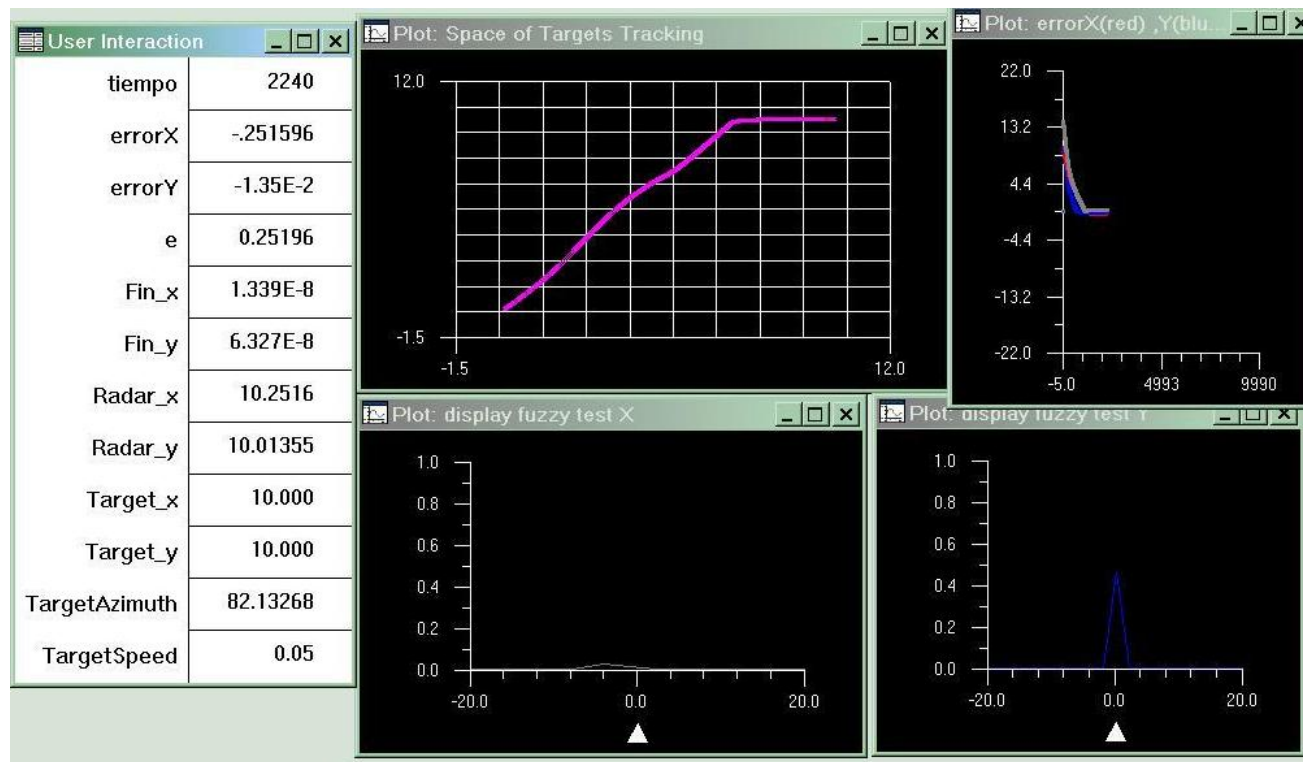


Haciendo un zoom:



Ahora mostraremos un par de screenshots de mi programa de simulación con CubiCalc, el cuál se ejecuta empleando el archivo GUS6.CBC.

Éste es nuestro modelo para un blanco estacionario en el espacio:

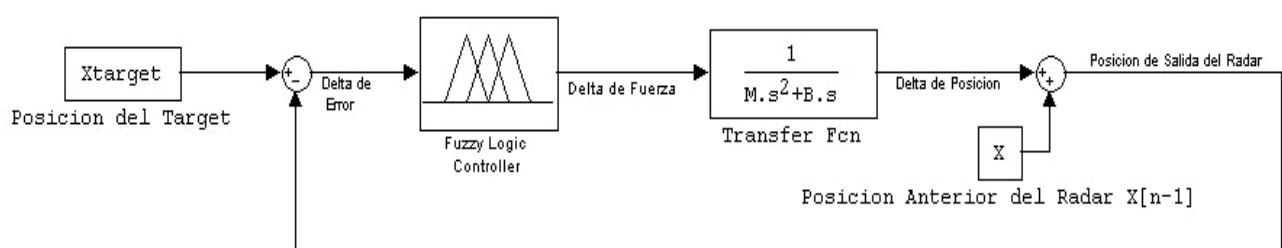


Describiendo el resultado del modelo, tendremos en el panel de la izquierda, las variables mas importantes del sistema, indicando su valor en cada instante. Si se detiene la simulación, se puede modificar a voluntad los valores y ver como responde el sistema.

La grilla que va desde -1,5 hasta 12 (tanto en el eje x, como en el eje y), y se encuentra recorrida por una línea de color fuxia, es el espacio de cobertura de nuestro radar bidimensional, y dicha línea es el camino que realizó nuestro radar para enfocar a un blanco estacionario ubicado en la coordenada (10;10), desde su situación de reposo en el origen de coordenadas.

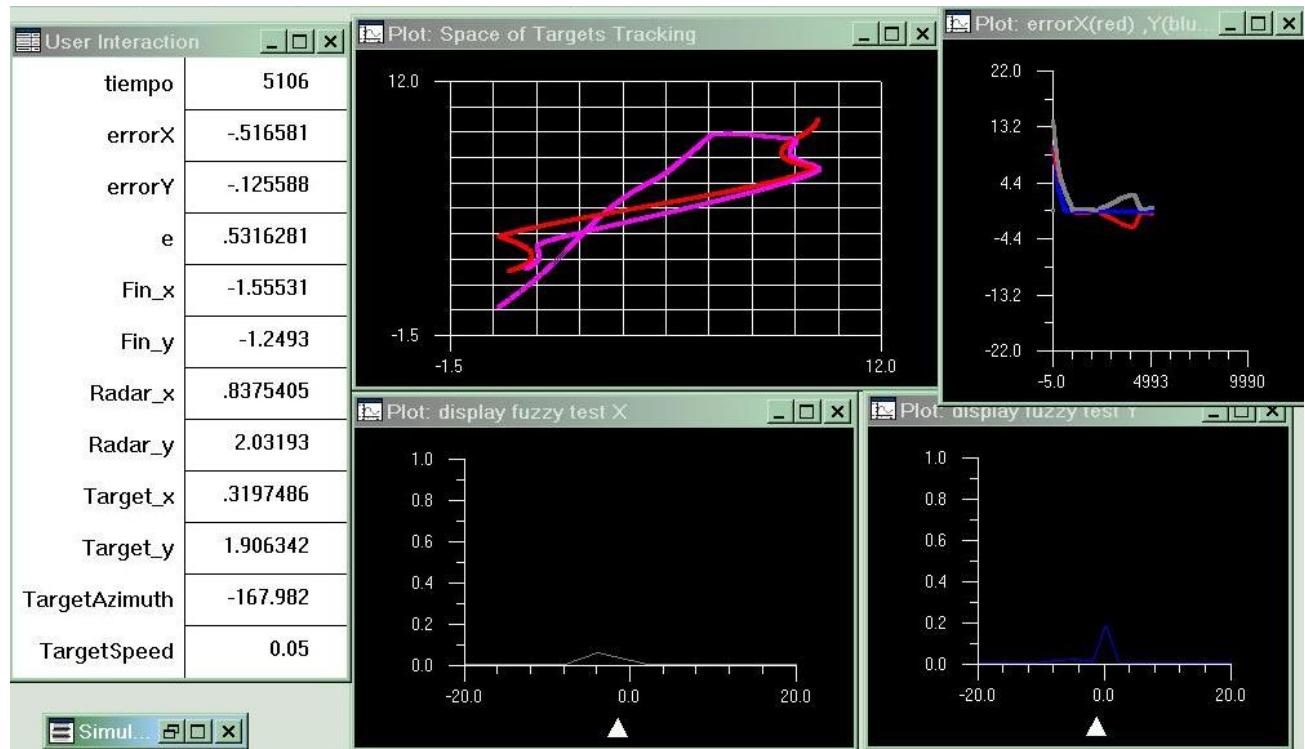
El gráfico de su derecha, describe el error que estamos cometiendo al enfocar nuestro blanco en función del tiempo. La línea roja es el error de azimuth, la línea azul es el error cometido en elevación, y la línea gris es la suma de la composición de ambos errores en valor absoluto y por ende siempre será mayor a cero.

Los dos gráficos inferiores muestran el resultado de Fin_x (force input x), y Fin_y (force input y) respectivamente, que son las dos salidas de mi controlador difuso, y son diferenciales de fuerza. Éstos diferenciales de fuerza, son las entradas de la planta, o sea las entradas actuantes en mi sistema de orientación de la antena del radar.



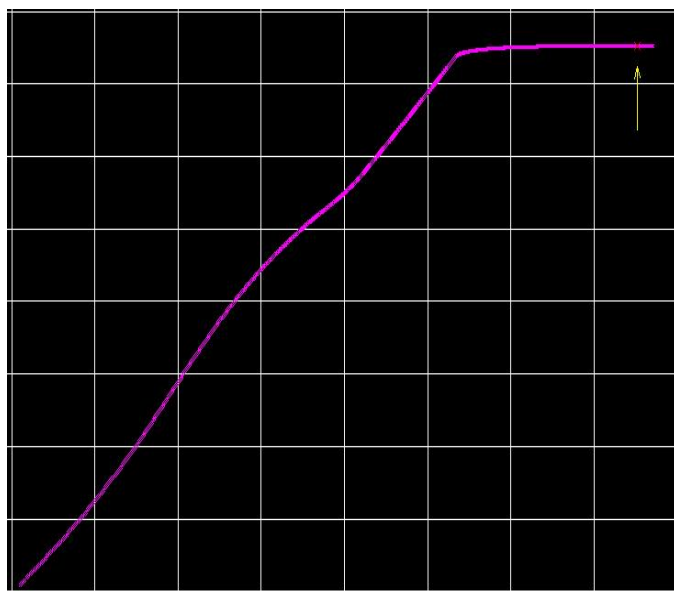
Los triángulos son la composición de las reglas que están actuando en ese momento, y el cursor de abajo, indica el nivel de la salida de la variable para ese instante.

Ahora empleando el archivo GUS7.CBC, podremos simular el modelo para un blanco de trayectoria similar a la que podría tener un misil dotado de contramedidas electrónicas para evitar ser derribado.



Los diversos paneles indican lo mismo que fue mencionado anteriormente, pero la diferencia radica en que la grilla (el espacio 2D de cobertura de nuestro radar), se encuentra recorrida por una línea roja que es la trayectoria del blanco, y otra fuxia que es la acción de seguimiento de nuestro radar mediante el controlador difuso.

Con respecto a los errores de los sistemas en estado estacionario, mencionados en la hoja número 3, podremos ver en detalle la simulación del funcionamiento para el sistema en la siguiente simulación:



Remarcado con una flecha amarilla, se puede apreciar el punto rojo que es la coordenada (10;10) de nuestro blanco estacionario.

Podemos apreciar que no se cometió error en elevación, pero sí se sucedió un pequeño sobrepaso en azimuth, debido al momento de inercia que tiene nuestra planta. (Recordemos que el movimiento de nuestra antena de 600Kg, debe ser lo mas rápido posible para alcanzar la ubicación del blanco.)

Ahora detallaremos el criterio tomado para las reglas de activación:

```
# Reglas de activacion del controlador
# Deberian ser iguales las del eje x a las del eje y , pero
# las hice diferentes para ver que pasaba|
```

```
# Reglas para el movimiento Azimuthal (EJE X)
```

```
If errorX is acceptable then make Fin_x OK;
```

```
If errorX is muchoDer then make Fin_x DerFuerte;
```

```
If errorX is muchoIzq then make Fin_x IzqFuerte;
```

```
If errorX is pocoIzq then make Fin_x IzqSuave;
```

```
If errorX is pocoDer then make Fin_x DerSuave;
```

```
If errorX is Izq then make Fin_x Izquierda;
```

```
If errorX is Der then make Fin_x Derecha;
```

```
# Reglas para el movimiento de Elevacion (EJE Y)
```

```
If errorY is acceptable then make Fin_y OK;
```

```
If errorY is muchoDer then make Fin_y Derecha;
```

```
If errorY is muchoIzq then make Fin_y Izquierda;
```

```
If errorY is derecha then make Fin_y Derecha;
```

```
If errorY is izquierda then make Fin_y Izquierda;
```

```
If errorY is pocoDer then make Fin_y DerSuave;
```

```
If errorY is pocoIzq then make Fin_y IzqSuave;
```

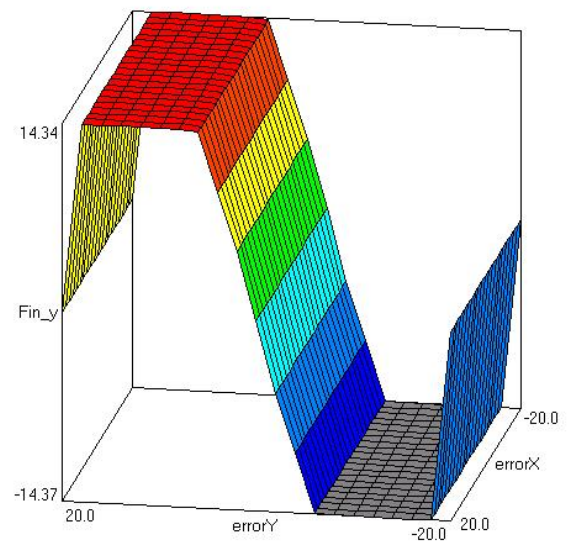
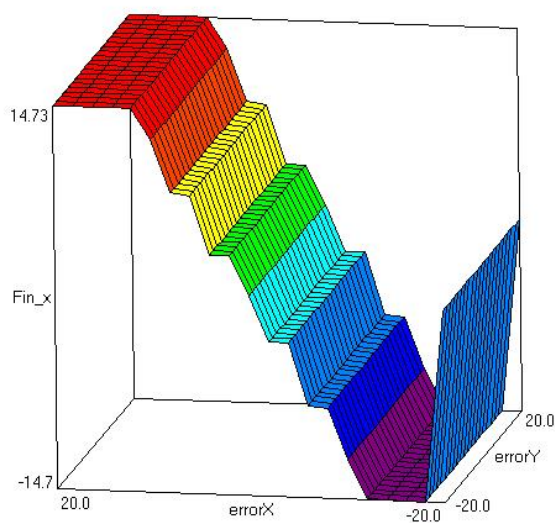
Veamos como se puede interpretar esto, con ejemplos concretos:

Primer regla: si la desviación que tenemos con respecto al lugar dónde nos encontramos apuntado “errorX”, es poca o aceptable para nosotros, entonces el controlador difuso decide que la variable de salida “Fin_x” no produzca cambios en el sistema (OK), para no modificar el lugar donde se apunta en ese instante.

Segunda regla: si la desviación que tenemos con respecto al lugar donde nos encontramos apuntado “errorX” es mucha y hacia la derecha, entonces el controlador difuso decide que la variable de salida “Fin_x” actúe en el sistema de forma de modificar fuertemente hacia la derecha, el lugar donde se apunta en ese instante.

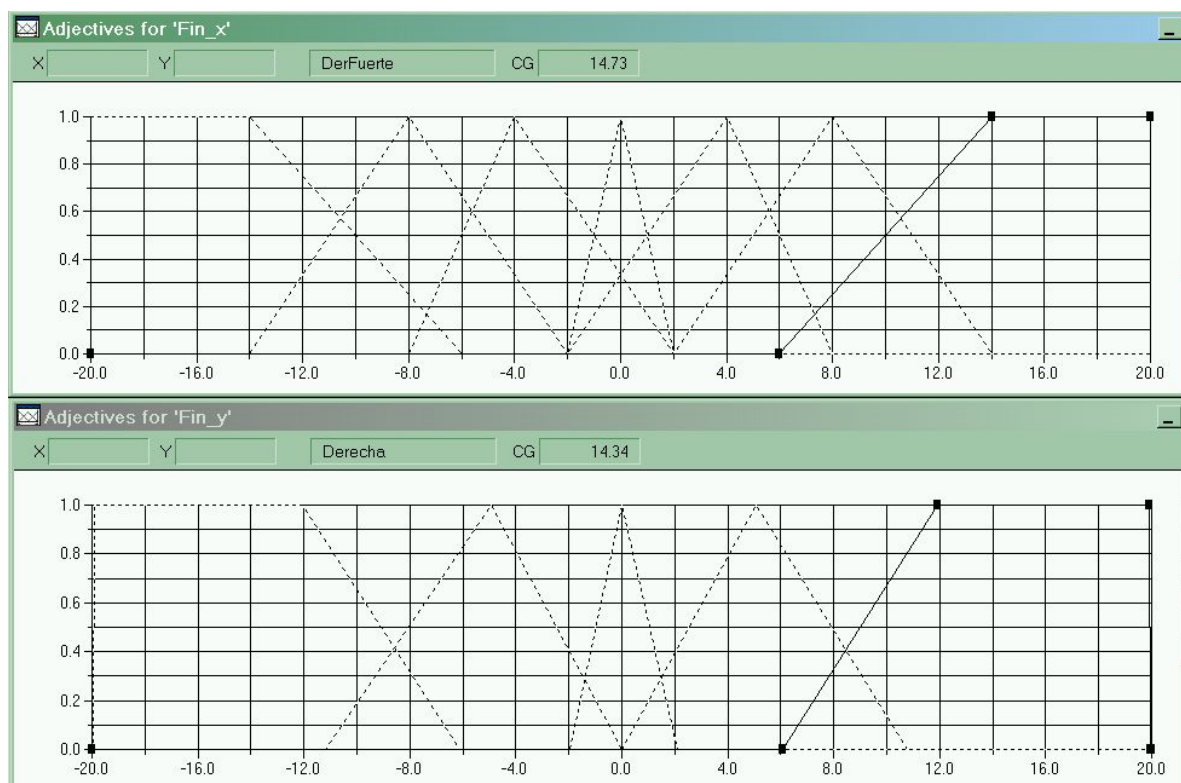
Y así con las demás reglas, nos resulta muy simple, ya que es como razonaría un ser humano. Los valores numéricos de los sensores de posición y fuerza son reemplazados por adjetivos, a los cuales los valores numéricos tienen cierto grado de pertenencia.

Los gráficos que veremos a continuación son las superficies de decisión. Dichos gráficos correlacionan dos entradas con una salida para ver sus influencias. Nosotros podremos advertir con facilidad que las entradas del controlador difuso “errorX” y “errorY” son independientes entre si.

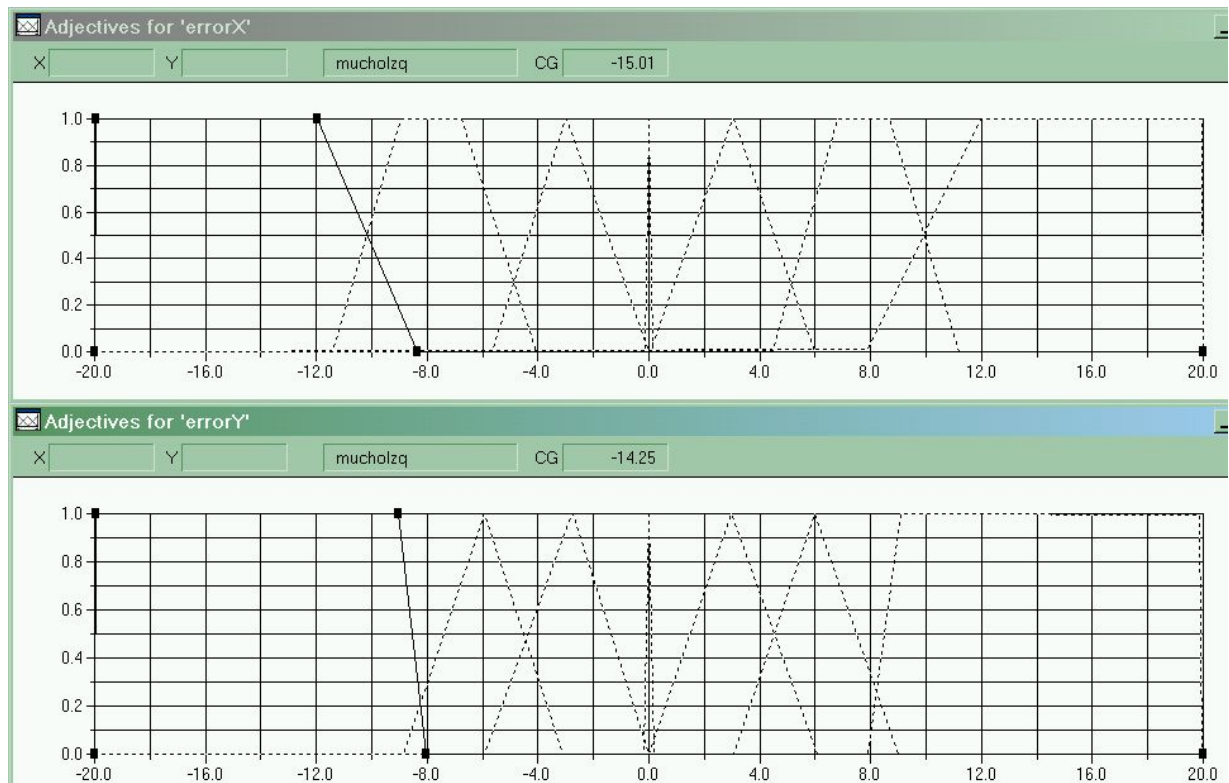


También se puede observar que ambos gráficos no son iguales, y sería lógico que “si” lo fueran, esto sucede por que he utilizado distinta cantidad de reglas de activación entre las salidas, para evaluar en forma cualitativa si resulta mas conveniente tener muchas reglas de activación o no.

En la salida “Fin_y” tenemos menos reglas de activación, lo que produce un cambio de pendiente mas pronunciado y homogéneo que en “Fin_x”.



Adicionalmente, podemos destacar que la salida que controla a Y, actúa frente a un error mayor que la salida X, pese a que las variables difusas de entrada “errorX” y “errorY” tienen la misma cantidad de reglas de activación, pero se sensibilizó las entradas de X como veremos a continuación.



Finalmente, podemos concluir que de éstos análisis realizados, que para nuestro sistema de seguimiento convendrá emplear la menor cantidad de reglas de activación para cada salida (como sucede en “Fin_y”), minimizando el tiempo de respuesta para el posicionamiento de blancos distantes, o que se mueven a gran velocidad, y emplear reglas de activación con buena sensibilidad en las entradas difusas como sucede en “errorX”, para comenzar la corrección de la trayectoria antes.

Detalle del algoritmo del programa:

El programa es relativamente corto, y se encuentran varias cosas duplicadas porque es un análisis bidimensional. Se debe recordar que el CubiCalc funciona como un LOOP, una vez que ejecutan todas las instrucciones, actualizando sus variables y empleando las reglas de decisión establecidas en forma de adjetivos en el menú PROYECT→RULES, y en forma gráfica en el menú PROYECT→ADJETIVES FOR ... comienza a ejecutarse nuevamente la primer instrucción.

Para acceder al programa vamos al menú PROYECT→SIMULATION, las otras opciones de inicialización, pre y post procesamiento también son parte del LOOP pero no fueron utilizadas, todo se realizó en el programa principal que es el mostrado a continuación:

```
# Constantes de mi sistema mecanico de movimiento de la antena:

M=600;           # 600kg suponiendo un reflector de 6,1m para trabajar a 2,8Ghz
B=10;            # 10kg/(m.s) viscosidad del actuador
TargetSpeed=0.05; # velocidad del misil

errorX=(Target_x-Radar_x);      # errorX (IN FUZZY) genera la salida Fin_x (OUT FUZZY)
errorY=(Target_y-Radar_y);      # errorY (IN FUZZY) genera la salida Fin_y (OUT FUZZY)
e=((Target_x-Radar_x)^2+(Target_y-Radar_y)^2)^0.5;      # para graficar en 2D - line 90

# Limitamos el espacio de operacion del misil (si y solo si se cumple la primer condicion,
# retorna el segundo argumento con el cual cambiamos la fase del misil, si no se cumple
# dejamos la fase como esta)

TargetAzimuth = Iif(Target_x >= 10.0, 180 - TargetAzimuth + (10 * uniform())-5.0, TargetAzimuth);
TargetAzimuth = Iif(Target_x <= 0.0, 180 - TargetAzimuth + (10 * uniform())-5.0, TargetAzimuth);
TargetAzimuth = Iif(Target_y >= 10.0, -TargetAzimuth + (10 * uniform())-5.0, TargetAzimuth);
TargetAzimuth = Iif(Target_y <= 0.0, -TargetAzimuth + (10 * uniform())-5.0, TargetAzimuth);
TargetAzimuth = angle180(TargetAzimuth);

# Funcion locura del misil (PODEMOS ALTERNAR ENTRE UN BLANCO FIJO O UNO MOVIL)

#Target_x = Target_x + 4*TargetSpeed*sin(cos(0.001*tiempo)*0.0001*tiempo)/10 * cosd(TargetAzimuth);
#Target_y = Target_y + 20*TargetSpeed/(100) * sind(TargetAzimuth);

Target_x = 10; # Punto Fijo
Target_y = 10;

tiempo=tiempo+1;

# ECUACIONES QUE DESCRIBEN LA TRANSFERENCIA DE LA PLANTA
#deltaRadarX=10*Fin_x*(1/(B*M))-(1/B)*exp(-B*tiempo/M)); #line 116 - Ecuaciones originales
deltaRadarX=10*Fin_x*(1/(B*M));                          #Ecuaciones Simplificadas
#deltaRadarY=10*Fin_y*(1/(B*M))-(1/B)*exp(-B*tiempo/M)); #line 118 - Ecuaciones originales
deltaRadarY=10*Fin_y*(1/(B*M));                          #Ecuaciones Simplificadas

# Salidas de la planta: delta de posicion del radar, lo que se debe mover desde donde se encuentra
#                          apuntando en ese momento para seguir al blanco.

Radar_x=(posicionAnteriorX+deltaRadarX);      # Line 155
posicionAnteriorX=Radar_x;

Radar_y=(posicionAnteriorY+deltaRadarY);      # Line 158
posicionAnteriorY=Radar_y;
```

¿Que es la lógica difusa?

La lógica difusa (Fuzzy Logic) ha surgido como una herramienta lucrativa para el control de subsistemas y procesos industriales complejos, así como también para la electrónica de entretenimiento y hogar, sistemas de diagnóstico y otros sistemas expertos.

Lo difuso ha llegado a ser una palabra clave para vender. Los artículos electrónicos sin componentes difusos se están quedando gradualmente desfasados. La popularidad de la lógica difusa, produce que cada vez sea mas frecuente un sello "fuzzy logic" impreso sobre el producto.

En Japón la investigación sobre lógica difusa es apoyada ampliamente con un presupuesto enorme. En Europa y USA se están realizando esfuerzos para alcanzar el tremendo éxito japonés. Por ejemplo, la NASA emplea lógica difusa para el complejo proceso de maniobras de acoplamiento.

La lógica difusa es básicamente una ***lógica multievaluada***, 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. De esta forma se ha realizado un intento de aplicar una forma más humana de pensar en la programación de computadoras.

La lógica difusa se inició en 1965 por Lotfi A. Zadeh, profesor de ciencias de la computación en la Universidad de California, en Berkeley.

¿Qué es un conjunto difuso?

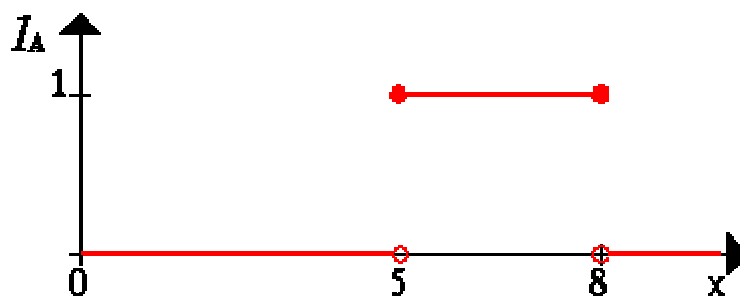
La noción mas básica de sistemas difusos es un sub-conjunto difuso. Significa que no se puede definir la pertenencia total de un elemento a un conjunto, sería como una pertenencia gradual al conjunto.

Veamos un ejemplo:

En primer lugar consideramos ***un conjunto X*** con todos los números reales entre 0 y 10, que nosotros llamamos ***el universo de discurso***. Ahora definimos un subconjunto A de X, con todos números reales en el rango entre 5 y 8.

$$A = [5,8]$$

Ahora mostramos el conjunto A por su ***función característica***, es decir esta función asigna un número 1 o 0 al elemento en X, dependiendo de si el elemento está en el subconjunto A o no. Esto conlleva a la figura siguiente:



Nosotros podemos interpretar los elementos que han asignado el número 1 como los elementos que están en el conjunto A, y los elementos que han asignado el número 0 como los elementos que no están en el conjunto A.

Este concepto es suficiente para muchas áreas de aplicación. Pero nosotros podemos encontrar fácilmente situaciones donde carece de flexibilidad. Para comprender este concepto veamos un ejemplo:

Queremos describir el conjunto de gente joven. Más formalmente nosotros podemos denotar

$$B = \{\text{conjunto de gente joven}\}$$

Como - en general - la edad comienza en 0, el rango mas inferior de este conjunto está claro. El rango superior, por otra parte, es más bien complicado de definir. Como un primer intento colocamos el rango superiora en 20 años, por decir algo. Por lo tanto nosotros definimos B como un intervalo denominado:

$$B = [0,20]$$

Ahora la pregunta es: ¿por qué alguien es en su 20 cumpleaños joven, y al día siguiente no? Obviamente, este es un problema estructural, porque si movemos el límite superior del rango desde 20 a un punto arbitrario podemos plantear la misma pregunta.

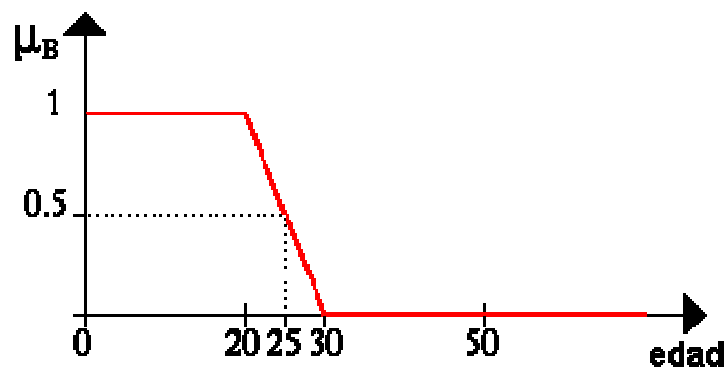
Una manera más natural de construir el conjunto B, estaría en suavizar la separación estricta entre el joven y el no joven. Nosotros haremos esto para permitir no solamente la (crispada) decisión "él/ella SI está en el conjunto de gente joven" o "él/ella NO está en el conjunto de gente joven", sino también las frases más flexibles como "él/ella SI pertenece un poquito mas al conjunto de gente joven" o "él/ella NO pertenece aproximadamente al conjunto de gente joven".

Pasamos a continuación a mostrar como un conjunto difuso nos permite definir una noción como "él/ella es un poco joven".

En nuestro primer ejemplo, codificamos todos los elementos del Universo de Discurso con 0 o 1. Una manera de generalizar este concepto está en permitir más valores entre 0 y 1. De hecho, permitiremos infinitas alternativas entre 0 y 1, denominando el intervalo: $Y_0 = [0, 1]$.

La interpretación de los números ahora asignados a todos los elementos del Universo de Discurso, es algo mas difícil. Por supuesto, el número 1 asignado a un elemento significa que el elemento está en el conjunto B y 0 significa que el elemento no está definitivamente en el conjunto el B. El resto de valores significan una pertenencia gradual al conjunto B.

Para ser mas concretos, mostramos gráficamente el conjunto de gente joven por su función característica.



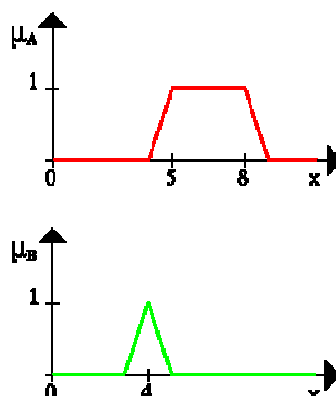
De esta forma, a los 25 años de edad todavía sería joven al grado de 50 por ciento.

Operaciones con conjuntos difusos

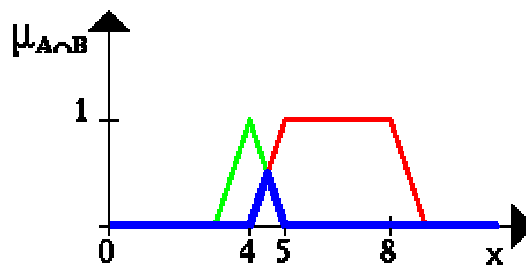
Las operaciones básicas sobre conjuntos difusos, son parecidas a las operaciones sobre conjuntos booleanos, nosotros también podemos intersectar, unificar y negar (complementar) conjuntos difusos.

L. A. Zadeh sugirió el operador mínimo para la intersección, y el operador máximo para la unión de dos conjuntos difusos. Es fácil ver que estos operadores coinciden con la unificación booleana, e intersección si nosotros únicamente consideramos los grados miembros 0 y 1.

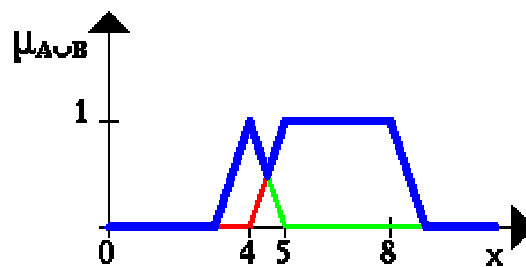
A fin de aclarar esto, mostraremos ejemplos. Sea A un intervalo difuso entre 5 y 8, y B un número difuso entorno a 4. Las figuras correspondientes se muestran a continuación:



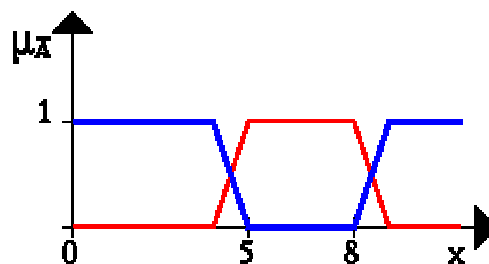
La figura siguiente muestra la operación AND (Y), o el resultado del operador mínimo, del conjunto difuso A y el número difuso B (el resultado es la línea azul).



La operación OR (O), es el resultado del operador máximo del conjunto difuso A con el número difuso B, se muestra en nuevamente, en línea azul.



Un ejemplo de complementación de A, la línea azul es la NEGACION del conjunto difuso A.



El control difuso

Los controladores difusos, son las aplicaciones mas importantes de la teoría difusa. Ellos trabajan de una forma bastante diferente a los controladores convencionales; *se utiliza el conocimiento del experto en el tema, en vez de generar las ecuaciones diferenciales para describir un sistema.*

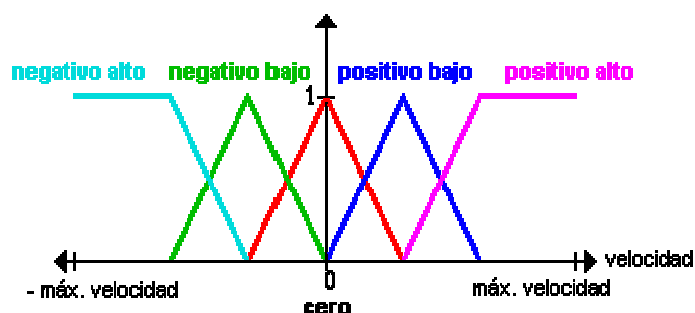
Este conocimiento puede expresarse de una manera muy natural, empleando las variables lingüísticas que son descritas mediante conjuntos difusos.

Ejemplo: El péndulo invertido

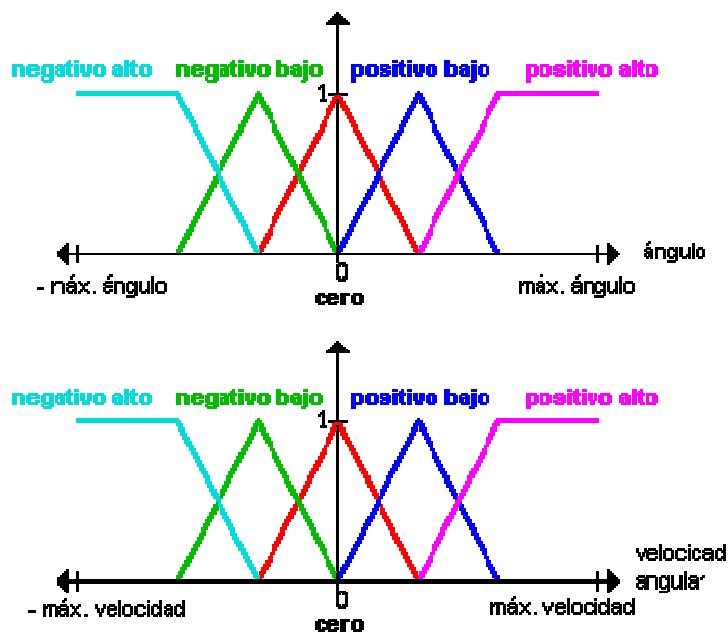
El problema está en equilibrar una escoba sobre nuestra mano (plataforma móvil), que puede moverse en dos únicas direcciones (para acotar la dificultad del ej.), a la izquierda o a la derecha.

Ante todo, nosotros tenemos que definir (subjetivamente) cual es la velocidad del movimiento pendular: alta, baja, etc. Esto se hace, para especificar las funciones pertenecientes al conjunto difuso:

- negativo alto (celeste)
- negativo bajo (verde)
- cero (rojo)
- positivo bajo (azul)
- positivo alto (morado)



Lo mismo se hace para el ángulo entre la plataforma y la escoba, y para la velocidad angular del ángulo:



Para hacerlo más fácil, suponemos que al principio la escoba está en una posición cercana a la central, para que un ángulo mayor de 45 grados en cualquier dirección no pueda ocurrir (por definición nuestra).

Ahora formularemos reglas, que dicen qué hacer en situaciones concretas (la inteligencia).

Considere por ejemplo que la escoba está en la posición central (el ángulo es cero), y no se mueve (la velocidad angular es cero). Obviamente esta es la situación deseada, y por lo tanto no tenemos que hacer nada (la velocidad es cero).

Consideremos otro caso: la escoba está en la posición central como antes, pero está en movimiento a baja velocidad en la dirección positiva. Naturalmente nosotros tendríamos que compensar el movimiento de la escoba moviendo la plataforma en la misma dirección a baja velocidad.

De esta forma, hemos constituido dos reglas, que podemos expresar como:

Si el ángulo es cero **y** la velocidad angular es cero, **entonces** la velocidad será cero.

Si el ángulo es cero **y** la velocidad angular es positiva baja, **entonces** la velocidad será positiva baja.

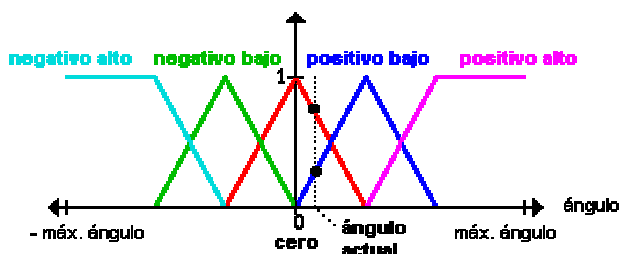
Podemos resumir todas las reglas aplicables en una tabla:

veloc	angulo				
	NA	NB	C	PB	PA
NA			NA		
NB			NB	C	
C	NA	NB	C	PB	PA
PB		C	PB		
PA			PA		

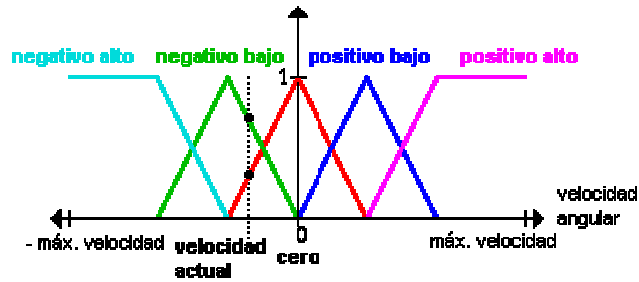
Dónde NA es abreviatura para negativa alta, NB para negativa baja, etc.

A continuación, mostraremos como estas reglas pueden aplicarse con valores concretos para el ángulo y velocidad angular. Para ello vamos a definir dos valores explícitos para el ángulo y la velocidad angular, y así operar con ellos.

Un valor actual para el ángulo:



Un valor actual para la velocidad angular:

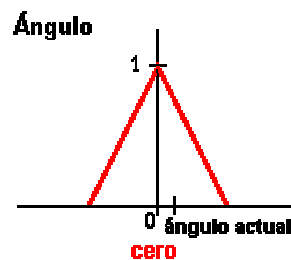


Ahora mostraremos como aplicar nuestras reglas a esta situación real.

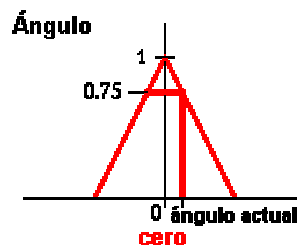
Veamos como aplicar la regla:

Si el ángulo es cero **y** la velocidad angular es cero, **entonces** la velocidad será cero. a los valores que hemos definido.

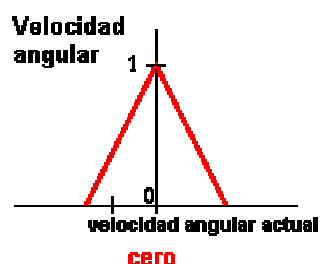
Esta es la variable lingüística "ángulo" donde nos centramos en el conjunto "cero", y el ángulo actual:



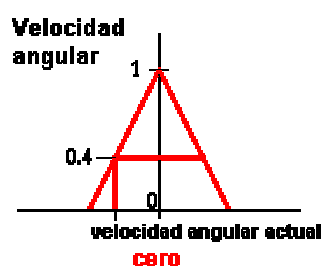
Nos damos cuenta que nuestro valor real pertenece al conjunto difuso "cero" en un grado de 0.75:



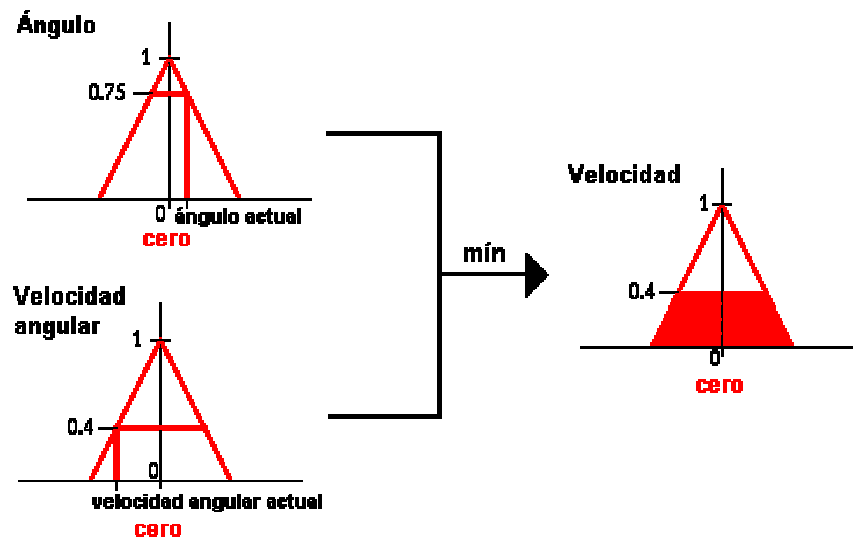
Ahora mostramos la variable lingüística "velocidad angular", donde nos centramos en el conjunto difuso "cero" y el valor actual de velocidad angular:



Nos damos cuenta que nuestro valor real pertenece al conjunto difuso "cero" en un grado de 0.4:

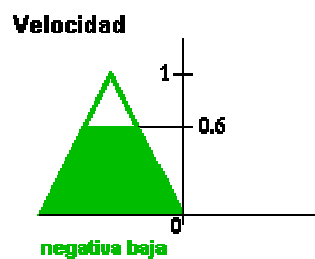


Como las dos partes de la condición de nuestra regla están unidas por una **Y** (operación lógica **AND**), calculamos el $\min(0.75, 0.4) = 0.4$, y cortamos el conjunto difuso "cero" de la variable "velocidad" a este nivel (según nuestra regla):



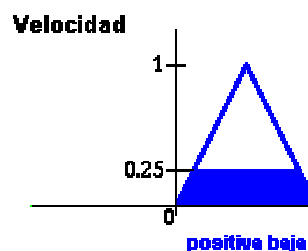
Por otra parte, el resultado de la regla

Si el ángulo es cero **y** la velocidad angular es negativa baja **entonces** la velocidad será negativa baja nos queda:



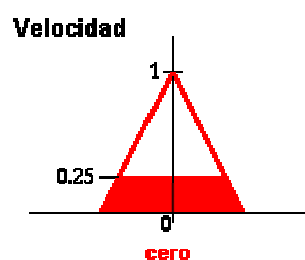
En otro caso, el resultado de la regla

Si el ángulo es cero **y** la velocidad angular es positiva baja **entonces** la velocidad será positiva baja

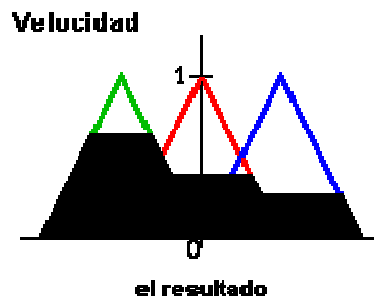


Analizando el resultado de la regla

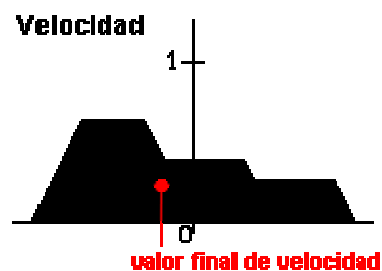
Si el ángulo es positivo bajo **y** la velocidad angular es negativa baja **entonces** la velocidad será cero



Estas cuatro reglas solapadas, desembocan en un único resultado:



El resultado del controlador difuso es un conjunto difuso (de velocidad), indicado por el área negra, así que debemos escoger un valor representativo como salida final (un valor escalar). Hay varios métodos heurísticos o de **defuzzification**, uno de ellos es tomar el centro de gravedad del conjunto difuso.



Este procedimiento completo, se denomina controlador de Mamdani.

Aplicaciones de la lógica difusa

Principalmente, miraremos la aptitud del control difuso en términos generales.

El empleo del control difuso es recomendable:

- Para procesos muy complejos, cuando no hay un modelo matemático simple.
- Para procesos altamente no lineales.
- Si el procesamiento del (lingüísticamente formulado), conocimiento experto puede ser recopilado.

El empleo del control difuso no es una buena idea si:

- El control convencional teóricamente rinde un resultado satisfactorio.
- Existe un modelo matemático fácilmente soluble y adecuado.

Definición de Variables lingüísticas

Un variable lingüística es un quintuple (X , $T(X)$, U , G , M).

Dónde:

X es el nombre de la variable.

$T(X)$ es el término conjunto (es decir, el conjunto de nombres de valores lingüísticos de X).

U es el universo de discurso.

G es la gramática para generar los nombres.

M es un conjunto de reglas semánticas para asociar cada X con su significado.