Funciones de Pertenencia Para Lógica Difusa Utilizando MATLAB

*Edgar Andres Gutierrez Caceres*

*Abstract*— Para el desarrollo de la presente práctica se realizaron gráficas de algunos de los conjuntos difusos más representativos. Se utilizó el software MATLAB para elaborar los programas, de forma que se pudieran introducir valores diferentes en los parámetros cada vez que se corre un programa. También se incluye un ejemplo de una aplicación que puede ser representada con una gráfica de conjuntos difusos.

# INTRODUCCIóN

E

l software MATLAB es una lenguaje técnico de alto nivel con un ambiente interactivo para el desarrollo de algoritmos, la visualización de los datos, el análisis de datos, y el cómputo numérico. Usando MATLAB, se pueden solucionar problemas que computan técnicos más rápidamente que con los lenguajes de programación tradicionales, tales como C, C++, y FORTRAN. Se puede utilizar MATLAB en una amplia gama de usos, incluyendo procesamiento de señal y de imagen, comunicaciones, diseño del control, modelación, el análisis financiero y biología de cómputo.

Las cajas de herramientas adicionadas extienden el ambiente de MATLAB para solucionar clases particulares de problemas en estas áreas de aplicación. MATLAB también proporciona un número de características para documentar y compartir el trabajo.

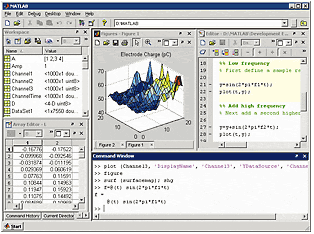


Fig. 1. MATLAB permite analizar gráficas de forma interactiva.

Para comenzar a utilizar el software es necesario conocer su funcionamiento básico y entender su lógica. A continuación se muestran algunos ejemplos de operaciones comunes:

Los cálculos que no se asignan a una variable en concreto se asignan a la variable de respuesta por defecto que es ans (del inglés, answer):

>>2+3

ans =

5

Sin embargo, si el cálculo se asigna a una variable, el resultado queda guardado en ella:

>>x=2+3

x =

5

Para conocer el valor de una variable, basta teclear su nombre:

>>x

x =

5

Si se añade un punto y coma (;) al final de la instrucción, la máquina no muestra la respuesta...

>>y=5\*4;

... pero no por ello deja de realizarse el cálculo.

>>y

y =

20

Las operaciones se evalúan por orden de prioridad: primero las potencias, después las multiplicaciones y divisiones y, finalmente, las sumas y restas. Las operaciones de igual prioridad se evalúan de izquierda a derecha:

>>2/4\*3

ans =

1.5000

>>2/(4\*3)

ans =

0.1667

Se pueden utilizar las funciones matemáticas habituales. Así, por ejemplo, la función coseno,

>>cos(pi) % pi es una variable con valor predeterminado 3.14159...

ans =

-1

o la función exponencial

>>exp(1) % Función exponencial evaluada en 1, es decir, el número e

ans =

2.7183

Otro ejemplo de función matemática: la raíz cuadrada; como puede verse, trabajar con complejos no da ningún tipo de problema. La unidad imaginaria se representa en MATLAB como i o j, variables con dicho valor como predeterminado:

>>sqrt(-4)

ans =

0+ 2.0000i

El usuario puede controlar el número de decimales con que aparece en pantalla el valor de las variables, sin olvidar que ello no está relacionado con la precisión con la que se hacen los cálculos, sino con el aspecto con que éstos se muestran:

>>1/3

ans =

0.3333

>>format long

>>1/3

ans =

0.33333333333333

>>format % Vuelve al formato estándar que es el de 4 cifras decimales

Para conocer las variables que se han usado hasta el momento:

>>who

Your variables are:

ans eps x y

o, si se quiere más información (obsérvese que todas las variables son arrays):

>>whos

Name Size Bytes Class

ans 1x1 8 double array

eps 1x1 8 double array

x 1x1 8 double array

y 1x1 8 double array

Grand total is 4 elements using 32 bytes

Para deshacerse de una variable

>>clear y

>>who

Your variables are:

ans eps x

Los ejemplos anteriores son útiles porque son operaciones realizadas comúnmente, pero son sólo una parte minúscula de las posibilidades que ofrece.

MATLAB ofrece una gran variedad de herramientas gráficas. También pueden dibujarse funciones. Así:

>>fplot('sin(x)',[0 2\*pi]) % Dibuja la función seno en el intervalo [0,2\*pi]

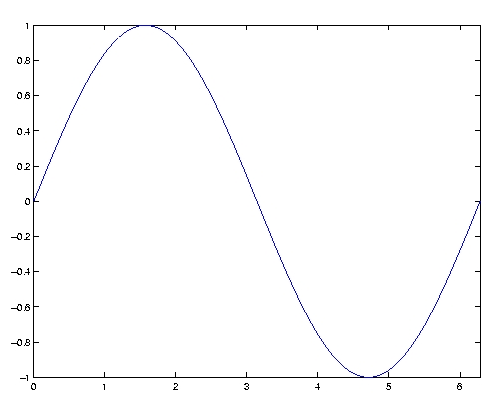


Fig. 2. Gráfica de una función senoidal en MATLAB

# Desarrollo

## Objetivo

* Conocer la programación de las funciones de pertenencia difusas mas representativas en MATLAB.

## Generación de Función Característica tipo Hombro derecho

Se puede señalar que tomando como ejemplo una función en específico se puede uno dar cuenta de las similitudes que presentan con otras funciones, es decir, si tomamos la función característica de la función triangular podemos identificar ciertos parámetros a simple vista que nos ayudarán en la construcción de diferentes funciones, como es el caso de una función del tipo hombro o saturación derecha.

Se puede deducir que el primer comportamiento que lleva la grafica de tipo triangulo constituye la misma subida que presenta la grafica de hombro y cuando llega al punto máximo con valor de pertenencia de uno, se mantiene constante hasta el final tal y como se ve en la Figura 3.

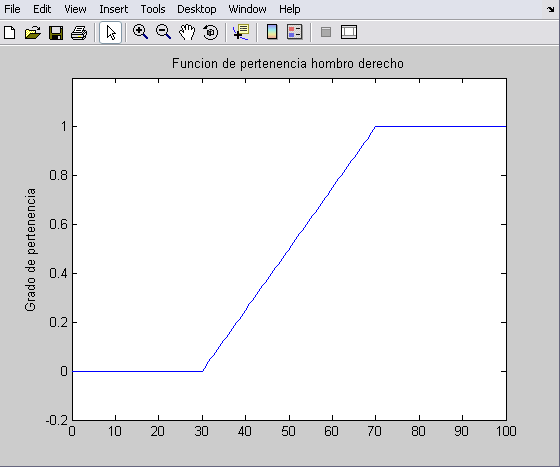
****

Figura 3 .- Gráfica correspondiente al tipo hombro derecho obtenida en el software de Matlab .

Para la función de tipo hombro hay dos parámetros que son los que dan la pauta de su comportamiento, los cuales hemos denominado como punto de “inicio”, que corresponde al punto en donde se empieza a elevar con pendiente constante desde cero hasta el segundo punto denominado como “centro” o en su defecto “final” que corresponde al punto en el que su incremento ha llegado al valor máximo en el grado de pertenencia, es decir “uno” y su valor permanece constante hasta el final de la grafica o hasta el último de los puntos que se estudian de esa función característica, si nos fijamos con detenimiento sobre la Figura 3 podemos ver que el valor que se escogió para el punto inicial de esta grafica corresponde al valor de 30 en el eje “X” mientras que el valor final o centro corresponde al valor en el eje “X” de 70.

Para el ejemplo en particular mostrado en la Figura 3, se fijaron unos valores, los cuales sabíamos con anterioridad que estos no cambiarían como es el ejemplo de los valores del eje “Y” delimitados en -0.2 para lograr ver la función con claridad pasar por el valor de cero y un tope máximo de 1.2 el cual la gráfica jamás rebasaría debido a que el mayo valor que puede obtener en grado de pertenencia es “uno”.; orto valor igual de importante para la programación es la de el paso, es decir cada cuando se haría una nueva medición y nosotros le dimos un valor arbitrario de 0.5 para cada uno de nuestras funciones.

Para la construcción de esta gráfica nos basamos en la definición de la misma que esta descrita en la siguiente ecuación 1.1

 (1.1)

Función de pertenencia del tipo hombro o saturación derecha

La interpretación de la ecuación 1.1 es que el grado de pertenencia dentro de la función esta representado por el valor de “X” y los parámetros de inicio y fin de los que hablábamos en párrafos anteriores corresponden a los símbolos  y  respectivamente; esto significa que para valores menores a X el valor de partencia a la función tiene un valor de “cero”, mientras que para los valores entre a y b se presenta una línea recta con inclinación o pendiente positiva dada por la ecuación (x-)/(-) y para todos los valores mayores que  su grado de pertenencia será “uno”.

La descripción del código utilizado en Matlab para esta función de pertenencia se encuentra descrito en el Anexo A.1 de este mismo escrito.

## Generación de Función Característica tipo Hombro izquierdo.

La grafica correspondiente a la función de pertenencia de tipo hombro o saturación izquierda es simplemente un reflejo de la función de tipo saturación derecha, donde los primeros valores de la grafica tendrán un valor de pertenencia máximo (“uno”) y en el parámetro de “inicio” empieza a descender de manera lineal al incrementarse los valores del eje “X” hasta llegar al punto de “final” donde ha alcanzado su mínimo valor de pertenencia y mantiene este valor hasta el final de la gráfica o los puntos analizados dentro de la función. Véase Figura 4.

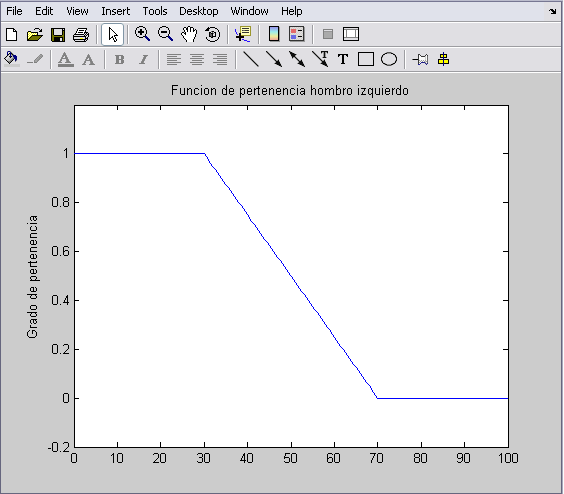
****

Figura 4 - Gráfica de tipo hombro izquierdo generada en Matlab.

Podemos ver que la grafica de la Figura 4 lleva un comportamiento inverso o negativo a la función de hombro derecho, y que se puede modelar tomando en consideración los mismos parámetros manipulados en la función anterior; que para este caso para el valor de “inicio” y “fin” corresponden a los valores en el eje “X” de 30 y 70 respectivamente.

La función de pertenencia tipo hombro izquierdo quedaría de la siguiente manera. Véase ecuación 2.1

 (2.1)

Función de pertenencia del tipo hombro o saturación izquierda

Para cada valor que sea menor a  le corresponde un valor de pertenencia de “uno” y permanecerá en ese estado hasta que “X” obtenga un valor entre  y que como podemos ver en la ecuación 2.1 le correspondería un valor con pendiente negativa y si analizamos con mas detenimiento la forma de la ecuación con respecto a la que se presentó en la ecuación 1.1 podemos ver que el cambio se presenta en el orden de los factores del numerados, los cuales se encuentran invertidos; al llegar a un valor mayor o igual a  se ha alcanzado el valor mínimo de pertenencia en esa función específica.

Para ver el código usado para la generación de la función de hombro izquierdo véase el Anexo A.2

## Generación de Función Característica tipo pi.

La grafica tipo pi aunque en principio parece ser más complicada que las anteriores, resulta ser solamente una combinación de las dos formas de saturación vistas anteriormente lo cual resulta en menor dificultad para su modelación. Véase Figura 5

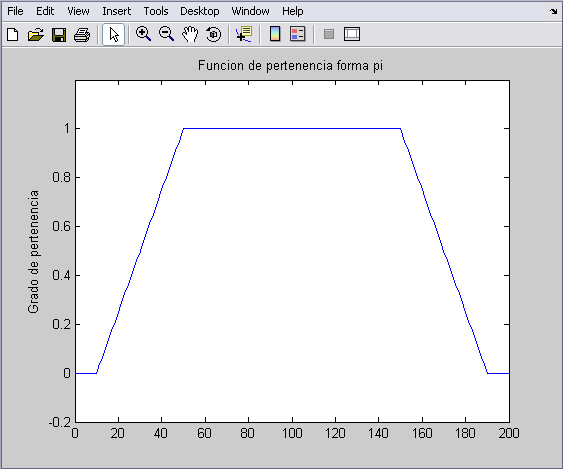


Figura 5.- Gráfica correspondiente a la funcion de pertenencia tipo Pi obtenida en Matlab

En la Figura 5 se encuentra gráficamente la función de pertenencia descrita por la forma tipo pi, en donde para efectos visuales se le han dado valores simétricos; para este tipo de funciones se necesitan controlar cuatro parámetros, los cuales hemos denominado como “inicio” al punto donde la función empieza a tener una pendiente positiva y por ende un valor de pertenencia mayor que cero y que para en este caso en particular tiene un valor en el eje de las “X” de 15; el segundo valor que controlamos es el valor que nosotros hemos llamado “Centro-Min” dando referencia al valor centro mínimo que corresponde al primer punto de la gráfica con valor de pertenencia de “uno” que para este ejemplo pertenece al valor en “X” de 50.

Así como hay un “centro-Mínimo” tenemos un valor “Centro- Max” refiriéndose al punto central correspondiente al último valor de pertenencia “uno” en la función, que para esta gráfica concierne al valor en “X” de 150 y empieza a decrecer de manera lineal su grado de pertenencia con pendiente negativa hasta llegar al punto donde su grado de pertenencia es de “cero” el cual hemos llamado “fin” y su valor permanece constante para valores mayores en el eje “X”.

El modelo que mejor describe el comportamiento de la función de tipo pi se muestra en la ecuación 3.1

 (3.1)

La ecuación 3.1 dice que el grado de pertenencia de la variable “X” estará en función de los parámetros , ,  y , de otra manera su valor será de “cero” siendo para nuestro caso particular  como nuestra variable “inicio”,  “centro-Min”,  “centro-Max” y por último  como nuestra variable ”fin” favor de ver el código de la función en el Anexo A.3

*D) Generación de Función Característica tipo Triangular*

La función triangular la realizamos en clase como un ejemplo de programación en Matlab. Ésta a su vez también se puede visualizar como la unión de las dos pendientes de las gráficas de las funciones de los hombros izquierdo y derecho.

Para la realización de ésta gráfica solo utilizamos tres parámetros. El primero denominado “inicio” que se refiere al punto en donde empieza la pendiente positiva. El segundo denominado “centro” el cuál es el punto en donde termina la pendiente positiva y empieza la negativa. El tercero es llamado “fin” y corresponde al punto en donde acaba la pendiente negativa. Véase Figura 6

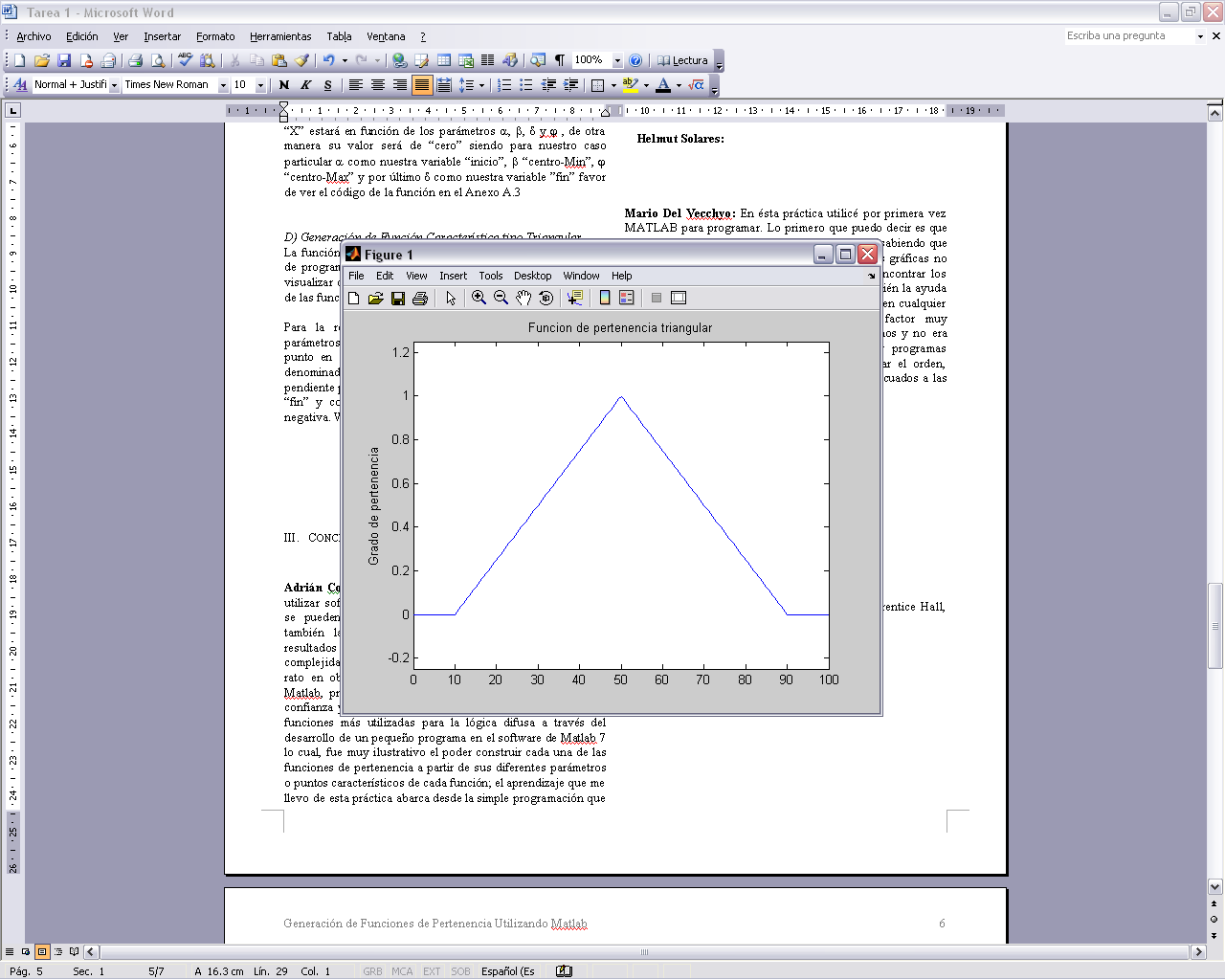


Figura 6 – Gráfica correspondiente a la función de pertenencia tipo Triangular obtenida en Matlab

Para la programación de ésta gráfica utilizamos las siguientes ecuaciones. Véase ecuación 4.1

 (4.1)

En el caso de la función triangular, consideramos  como el punto de “inicio”. El carácter es considerado como el “centro”. Mientras que es el punto “final”. Las dos primeras ecuaciones mostradas en 4.1 se refieren a una pendiente positiva y una negativa, respectivamente.

En la Figura 6 se puede apreciar claramente los valores de dichos parámetros. Cualquier valor antes del punto “inicio” es igual a cero. Entre los puntos “inicio” y “centro” tenemos una pendiente positiva dada por la primera ecuación de 4.1. Entre el punto “centro” y “final” tenemos una pendiente negativa representada por la segunda ecuación de 4.1. Además de que cualquier valor mayor al punto “final” tendrá un valor de cero.

En éste caso podemos apreciar que los valores dados en el programa corresponden a: inicio =10, centro=50, final=90. Esto dentro de un rango de 0 a 100 en el eje X.

En el anexo A.4 se encuentra el programa realizado para dicha función con sus respectivos comentarios.

*E) Generación de Función Característica tipo “S” o Sigmoidal*

La gráfica correspondiente a la función característica tipo “s” es la única de las anteriores que no se relaciona mucho con las formas de éstas. Esto es debido a que cuenta con un punto de inflexión ubicado en el centro de la gráfica. Sin embargo, obteniendo las ecuaciones es sencillo programar la función.

Para su programación utilizamos tres parámetros. El primero se ubica en el punto donde la gráfica empieza a crecer en el eje Y, es decir en el punto donde empieza a tener una pendiente exponencial y es denominado como “inicio”. El segundo parámetro, llamado “inflexión”, es en donde se encuentra el punto de inflexión de la gráfica. Por ultimo el tercer parámetro es denominado el “final” de la gráfica después del punto de inflexión.

Los valores proporcionados al programa en un rango de 0 a 100 en el eje X fueron: “inicio” = 10, “inflexión” = 50, “final”= 90. Véase Figura 7

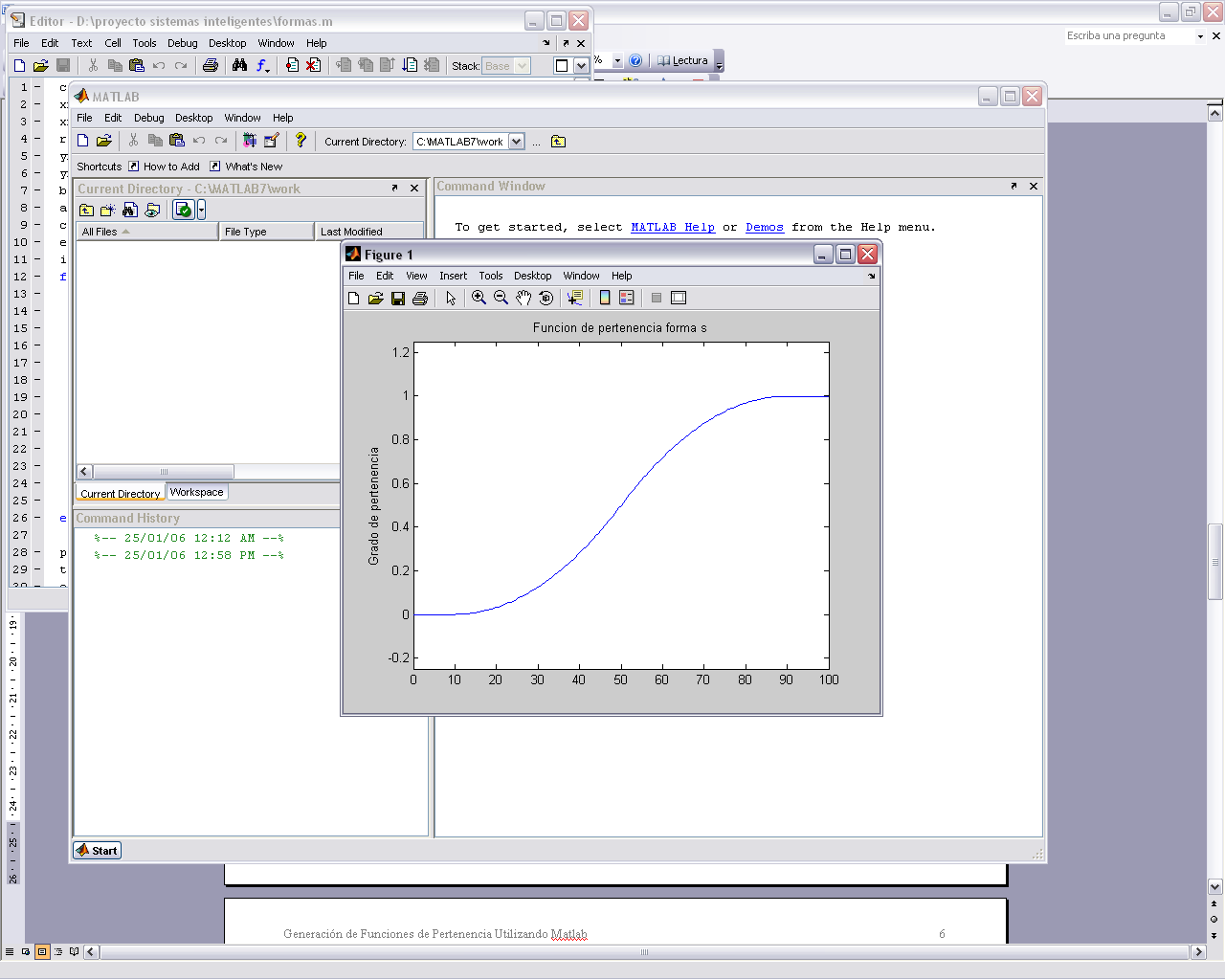


Figura 7 - Gráfica correspondiente a la función de pertenencia tipo “s” o sigmoidal obtenida en Matlab.

A continuación se muestran las ecuaciones 5.1 utilizadas para la realización de la gráfica.

 (5.1)

Para cualquier valor anterior a “inicio” o  el grado de pertenencia será cero. Para el siguiente se utiliza la segunda ecuación de 5.1 la cuál se refiere al tramo acotado entre “inicio” o  e “inflexión” o . Después encontramos el tramo ubicado entre “inflexión” o  y “final” o  caracterizado por la tercera ecuación de 5.1. Al finalizar tenemos que para cualquier valor mayor a “final” tendremos un grado de pertenencia igual a uno.

El programa completo se puede ver en el anexo A.5

# Ejemplo de aplicación

Si se dice "esta persona es muy alta" y otra persona contesta "tal vez, pero no tanto" estamos ante un fenómeno de lógica difusa: ¿hasta qué punto podemos decir que el concepto "alto" permite etiquetar a las personas? La lógica difusa nos permite resolver esta contradicción y asumirla como normal y, finalmente, trabajar con expresiones de ese tipo. Entonces diríamos que esa persona es miembro del conjunto de personas muy altas con un a relación de pertenecia del 30%, por ejemplo.

 Claro está que el escéptico podría decir que estadísticamente una persona es muy alta si mide 1.95 m, basándose en un argumento arbitrario, pero el resto de las personas tendrían todo el derecho a cuestionarlo. En la Figura 8 se representa el grado de pertenencia al conjunto de "jóvenes", en función de la edad. La línea roja representa el modelo tradicional, y el verde, el difuso.

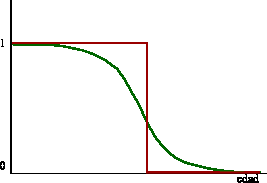


Figura 8 – Función de pertenencia de edades jovenes

# Referencias

1. Apuntes del curso de Sistemas Inteligentes.
2. Ogata, K. *Discrete-Time Control Systems*. Prentice Hall, 1995.
3. MatLab 7.0
4. <http://www.caos.tv/fuzzy.html>
5. Tarea funciones de membresía, curos sistemas inteligentes Prof. Pedro ponce, alumnos M. Adrián, Solares A. Helmut, Del Vecchyo Mario

Anexo A

## Programas de MatLab

Los diferentes programas desarrollados en MatLab que se utiliza en el proceso de la práctica y que permite realizar la programación de las diferentes funciones de pertenencia y su visualización gráfica se presenta a continuación.

A.1 este código que a continuación se enlista corresponde al generado para reproducir la función de saturación u hombro derecho.

%%%%%%%%%%%%%%%%%%

% funcion de tipo hombro derecho%

%%%%%%%%%%%%%%%%%%

clear all;% funcion para borrar variables de memoria

% definicion de parametros de los ejes X Y

xmin=input (' minimo='); %minimo valor eje x

xmax=input ('maximo=');

res=0.5;

ymin=0;

ymax=1;

% variables de entrada de la funcion

b=input('centro=');

a=input('inicio=');

ejex=xmin:res:xmax;

i=1;

% condiciones de pertenencia de funcion tipo hombro

for x=ejex;

if (x<a)

pertenhombroder(i)=0;

end

if (x>=a)&(x<b)

pertenhombroder(i)=(x-a)/(b-a);

end

if (x>=b)

pertenhombroder(i)=1;

end

i=i+1;

end

% parametros o codigo para graficar resultados

plot(ejex,pertenhombroder)

title('Funcion de pertenencia hombro derecho');

axis([xmin,xmax,ymin,ymax]);

ylabel('Grado de pertenencia');

% fin del codigo

Cada uno de los códigos listados en este anexo se encuentran debidamente comentados dentro del programa para su fácil comprensión.

Se puede identificar los comentarios del programa ya que son los únicos que vienen precedidos por un signo de porcentaje y se encuentran al principio de cada grupo de instrucciones.

A.2 Código generado en Matlab para la función de pertenencia de tipo hombro izquierdo.

%%%%%%%%%%%%%%%%%%%

% Funcion de tipo hombro izquierdo %

%%%%%%%%%%%%%%%%%%%

%codigo para limpiar memoria

clear all;

% definicion de parametros ejes X Y

xmin=input (' minimo=');

xmax=input ('maximo=');

res=0.5;

ymin=0;

ymax=1;

% Variables de entrada funcion especifica

b=input('fin=');

a=input('centro=');

ejex=xmin:res:xmax;

i=1;

% grados de pertenencia de la funcion hombro izq

for x=ejex;

if (x<a)

pertenhombroizq (i)=1;

end

if (x>=a)&(x<b)

pertenhombroizq (i)=1-(x-a)/(b-a);

end

if (x>=b)

pertenhombroizq(i)=0;

end

i=i+1;

end

% codigo para generar resultados graficos

plot(ejex,pertenhombroizq)

title('Funcion de pertenencia hombro izquierdo');

axis([xmin,xmax,ymin,ymax]);

ylabel('Grado de pertenencia');

% fin del codigo

A.3 El siguiente código corresponde al desarrollado para la función de pertenencia de tipo pi.

%%%%%%%%%%%

% funcion de tipo PI %

%%%%%%%%%%%

clear all;% recomendable para lilmpiar toda

%variable que se queda en memoria

%rangos de valores para los ejes de la gráfica

xmin=input (' X\_minima=');

xmax=input ('X\_maxima=');

res=0.5;

ymin=0;

ymax=1;

%parametros de entrada para la funcion especifica

b=input('centro-Min=');

d=input ('centro-Max=');

a=input('inicio=');

c=input('fin=');

ejex=xmin:res:xmax;

i=1;

% condiciones de pertenencia de la funcion pi

for x=ejex;

if (x<a)|(x>=c)

pertenpi(i)=0;

end

if (x>=a)&(x<b)

pertenpi(i)=(x-a)/(b-a);

end

if (x>=b)&(x<d)

pertenpi(i)=1;

end

if (x>=d)&(x<c)

pertenpi(i)=1-(x-d)/(c-d);

end

i=i+1;

end

% codigo para desplegar grafica de la función

plot(ejex,pertenpi)

title('Funcion de pertenencia forma pi');

axis([xmin,xmax,ymin,ymax]);

ylabel('Grado de pertenencia');

% fin del codigo

A.4 El siguiente código corresponde al desarrollado para la función de pertenencia de tipo Triangular.

%%%%%%%%%%%%%%%

%Función de tipo Triangular %

%%%%%%%%%%%%%%%

clear all;

%definición de parametros de ejes X Y

xmin=input (' minimo=');%

xmax=input ('maximo=');

res=0.5;

ymin=0;

ymax=1;

%Variables de entrada de funcion triangular

b=input('centro=');

a=input('inicio=');

c=input('fin=');

ejex=xmin:res:xmax;

i=1;

for x=ejex;

%grado de pertenencia = 0

if (x<a)|(x>=c)

pertentri(i)=0;

end

%pendiente positiva

if (x>=a)&(x<b)

pertentri(i)=(x-a)/(b-a);

end

%pendiente negativa

if (x>=b)&(x<c)

pertentri(i)=1-(x-b)/(c-b);

end

i=i+1;

end

%código para generar gráfica

plot(ejex,pertentri)

title('Funcion de pertenencia triangular');

axis([xmin,xmax,ymin,ymax]);

ylabel('Grado de pertenencia');

A.5 El siguiente código corresponde al desarrollo para la función de pertenencia de tipo “S”

%%%%%%%%%%%

%Función de tipo "S"%

%%%%%%%%%%%

clear all;

%definicion parametros ejes X Y

xmin=input (' minimo=');

xmax=input ('maximo=');

res=0.5;

ymin=0;

ymax=1;

%Variables entrada

b=input('inflexion=');

a=input('inicio=');

c=input('fin=');

ejex=xmin:res:xmax;

i=1;

for x=ejex;

%grado de pertenencia = 0

if (x<a)

pertens(i)=0;

end

%tramo entre inicio e inflexión

if (x>=a)&(x<b)

pertens(i)=2\*(((x-a)/(c-a))^2);

end

%tramo entre inflexion y fin

if (x>=b)&(x<=c)

pertens(i)=1-2\*(((x-c)/(c-a))^2);

end

%grado de pertenencia = 1

if (x>c)

pertens(i)=1;

end

i=i+1;

end

%código para generar gráfica

plot(ejex,pertens)

title('Funcion de pertenencia forma s');

axis([xmin,xmax,ymin,ymax]);

ylabel('Grado de pertenencia');