Mendoza Hernandez Yair Alberto
3BM4
Tercer departamental

P3.1.FIS Sistema Mandani de tipo hipotético

MARCO TEORICO

Lógica difusa

La lógica difusa (también llamada lógica borrosa) se basa en lo relativo de lo observado como posición diferencial. Este tipo de lógica toma dos valores aleatorios, pero contextualizados y referidos entre sí. Así, por ejemplo, una persona que mida dos metros es claramente una persona alta, si previamente se ha tomado el valor de persona baja y se ha establecido en un metro. Ambos valores están contextualizados a personas y referidos a una medida métrica lineal.

Fue formulada en 1965 por el ingeniero y matemático Lotfi A. Zadeh.

Funcionamiento

La lógica difusa (fuzzy logic, en inglés) se adapta mejor al mundo real en el que vivimos, e incluso puede comprender y funcionar con nuestras expresiones, del tipo «hace mucho calor», «no es muy alto», «el ritmo del corazón está un poco acelerado», etc.

La clave de esta adaptación al lenguaje se basa en comprender los cuantificadores de cualidad para nuestras inferencias (en los ejemplos de arriba, «mucho», «muy» y «un poco»).

En la teoría de conjuntos difusos se definen también las operaciones de unión, intersección, diferencia, negación o complemento, y otras operaciones sobre conjuntos (ver también subconjunto difuso), en los que se basa esta lógica.

Para cada conjunto difuso, existe asociada una función de pertenencia para sus elementos, que indica en qué medida el elemento forma parte de ese conjunto difuso. Las formas de las funciones de pertenencia más típicas son trapezoidal, lineal y curva.

Se basa en reglas heurísticas de la forma SI (antecedente) ENTONCES (consecuente), donde el antecedente y el consecuente son también conjuntos difusos, ya sea puros o resultado de operar con ellos. Sirvan como ejemplos de regla heurística para esta lógica (nótese la importancia de las palabras «muchísimo», «drásticamente», «un poco» y «levemente» para la lógica difusa):

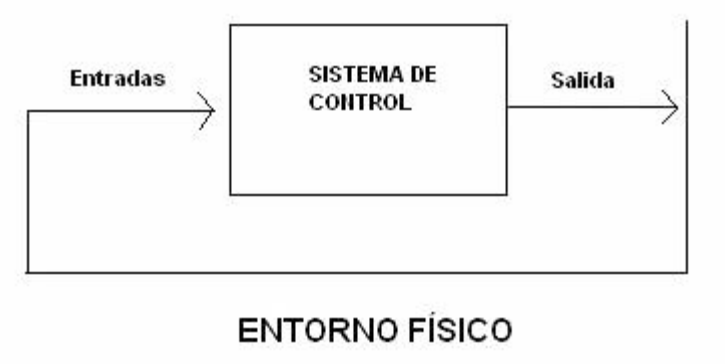
- SI hace muchísimo frío. ENTONCES aumento drásticamente la temperatura.
- SI voy a llegar un poco tarde. ENTONCES aumento levemente la velocidad.

Los métodos de inferencia para esta base de reglas deben ser sencillos, versátiles y eficientes. Los resultados de dichos métodos son un área final, fruto de un conjunto de áreas solapadas entre sí (cada área es resultado de una regla de inferencia). Para escoger una salida concreta a partir de tanta premisa difusa, el método más usado es el del centroide, en el que la salida final será el centro de gravedad del área total resultante.

Las reglas de las que dispone el motor de inferencia de un sistema difuso pueden ser formuladas por expertos o bien aprendidas por el propio sistema, haciendo uso en este caso de redes neuronales para fortalecer las futuras tomas de decisiones.

Los datos de entrada suelen ser recogidos por sensores que miden las variables de entrada de un sistema. El motor de inferencias se basa en chips difusos, que están aumentando exponencialmente su capacidad de procesamiento de reglas año a año.

Un esquema de funcionamiento típico para un sistema difuso podría ser de la siguiente manera:



En la figura, el sistema de control hace los cálculos con base en sus reglas heurísticas, comentadas anteriormente. La salida final actuaría sobre el entorno físico, y los valores sobre el entorno físico de las nuevas entradas (modificado por la salida del sistema de control) serían tomadas por sensores del sistema.

Por ejemplo, imaginando que nuestro sistema difuso fuese el climatizador de un coche que se autorregula según las necesidades: Los chips difusos del climatizador recogen los datos de entrada, que en este caso bien podrían ser la temperatura y humedad simplemente. Estos datos se someten a las reglas del motor de inferencia (como se ha comentado antes, de la forma SI... ENTONCES...), resultando un área de resultados. De esa área se escogerá el centro de gravedad, proporcionándola como salida. Dependiendo del resultado, el climatizador podría aumentar la temperatura o disminuirla dependiendo del grado de la salida.

Desarrollo y solución propuesta

La práctica consiste en realizar un sistema de inferencia difuso de tipo Mandani el cual será utilizado para encontrar la respuesta de un sistema hipotético propuesto en clase

El sistema propuesto para la realización de esta práctica consiste de 2 entradas caracterizadas por A y B en un rango de valores de 0 a 10 con una resolución de 0.1, cada una de las variables denotadas por dos conjunto difusos de tipo trapezoidal se definieron como A1 Y A2 para el conjunto A y B1 y B2 para el conjunto B por otra parte la salida está caracterizada por dos conjuntos trapezoidales también denotados por C1y C2, esta variable de salida C tiene un rango de -5 a 5. (Fig.1)

Una vez que tenemos caracterizado nuestro universo de trabajo se procede a fuzzificar los valores en las variables de entrada (Fig2.) este proceso se hace mediante el ingreso de valores propuesto por el usuario, dichos puntos se fuzzifican obteniendo su correspondiente grado de pertenencia en cada una de los conjuntos propuestos, posterior a la fuzzificacion se hace la evaluación de las reglas de inferencias propuestas para la resolución de este sistema, estas

reglas se representan mediante una tabla de inferencia,(tabla 1) en el algoritmo estas reglas se evalúan mediante mínimos

Tabla 1.Tabla de inferencia del sistema hipotético

	B1	B2
A1	C1	C2
A2	C2	C1

Al obtener el mínimo valor del resultado de las dos variables de entrada evaluadas en las regla difusa se obtiene el mínimo con respecto al conjunto de salida propuesto ,obteniendo un valor para cada una de muestras, estos cuatro valores generados por mínimos se comparan mediante máximos pero solo a los elementos que implican un cambio en el mismo conjunto de salida obteniendo entonces dos valores de estos máximos, estos valores generados implican el valor máximo al que los conjuntos se deben de cortar para obtener el conjunto a defuzzificar por ello entonces se hace uso de los valores generados por los máximos para recortar los conjuntos de C y posterior al corte se aplica un máximo sobre ambos conjuntos recortados para obtener la unión de ambos conjuntos (fig 3)

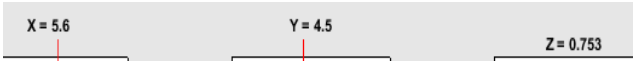
Una vez que obtuvimos este conjunto se utiliza una defuzzificacion por centroide el cual consiste en realizar la siguiente operación matemática

- Método del centroide

$$z^* = \frac{\int \mu_C(z) \cdot z dz}{\int \mu_C(z) dz}$$

Esta operación nos regresar el valor del centroide a este procedimiento se le conoce como defuzzificacion y el resultado de esta nos da el valor que se genera para el control (fig. 4) y el cual se ve reflejado mediante la superficie de control (fig. 5)una vez terminado este algoritmo podemos encontrar para cualquier punto que ingresemos su valor de defuzzificacion para corroborar el correcto funcionamiento se usa el toolbox de MatLab para lógica difusa en donde se caracteriza el sistema igual a como se hizo en el algoritmo y al comprarlos nos podemos dar cuenta como el valor defuzzificado corresponde al mismo tanto en el algoritmo realizado como en el toolbox por lo cual decimos que esta correcto el algoritmo . Los resultados comparados los presentamos a continuación

```
Ingrese el valor de X: 5.6
Ingrese el valor de Y: 4.5
Z : 0.75663
```



Mendoza Hernandez Yair Alberto
3BM4
Tercer departamental

Algoritmo

Para el sistema Mandani

```
clc,clear all,close all
%definición de parametros de ejes X Y
xmin=0;ymin=0;
xmax=10;ymax=1;
ejex=xmin:0.1:xmax;
TDA=TrapecioD(ejex,3,7);
TIA=TrapecioI(ejex,3,7);

TDB=TrapecioD(ejex,4,6);
TIB=TrapecioI(ejex,4,6);

xmins=-5;
xmaxs=5;
ejexs=xmins:0.1:xmaxs;
TDC=TrapecioD(ejexs,-1,1);
TIC=TrapecioI(ejexs,-1,1);

Ldf=length(TDA);

figure
subplot(3,1,1),plot(ejex,TIA),hold on,plot(ejex,TDA),grid on,legend('A1','A2')
title('Conjunto de entrada A');
axis([xmin,xmax,ymin,ymax]),ylabel('Grado de pertenencia'),xlabel('X')

subplot(3,1,2),plot(ejex,TIB),hold on,plot(ejex,TDB),grid on,legend('B1','B2')
title('Conjunto de entrada B');
axis([xmin,xmax,ymin,ymax]),ylabel('Grado de pertenencia'),xlabel('Y')

subplot(3,1,3),plot(ejexs,TIC),hold on,plot(ejexs,TDC),grid on,legend('C1','C2')
title('Conjunto de salida C');
axis([xmins,xmaxs,ymin,ymax]),ylabel('Grado de pertenencia'),xlabel('Z')

Val1=(input('Ingrese el valor de X: ')*10)+1;
Val2=(input('Ingrese el valor de Y: ')*10)+1;
Px1=TDA(Val1);Py1=TDB(Val2);
Px2=TIA(Val1);Py2=TIB(Val2);
figure
subplot(2,1,1),plot(ejex,TIA,ejex,TDA), hold on
,plot(ejex(Val1),Px1,'*r',ejex(Val1),Px2,'*b'),grid on
title('Fuzzyficacion'),xlabel('X'),ylabel('Grado de pertenencia'),...
    legend('A1','A2','PpA2','PpA1')
subplot(2,1,2),plot(ejex,TIB,ejex,TDB), hold on
,plot(ejex(Val2),Py1,'*r',ejex(Val2),Py2,'*b'),grid on
xlabel('Y'),ylabel('Grado de pertenencia'),legend('B1','B2','PpB2','PpB1')
%%
%Reglas
       r1=min(TIA(Val1),TIB(Val2));
       R1=min(r1,TIC);
       r2=min(TIA(Val1),TDB(Val2));
       R2=min(r2,TDC);
       r3=min(TDA(Val1),TIB(Val2));
       R3=min(r3,TDC);
       r4=min(TDA(Val1),TDB(Val2));
       R4=min(r4,TIC);

       Rec1=max(max(R1,R4));
       Rec2=max(max(R2,R3));
```

```
        RC=max(r1,r4);
        RC2=max(r2,r3);

 for i=1:Ldf
    if TIC(i)<=Rec1
       TICr(i)=TIC(i);
    else
       TICr(i)=Rec1;
    end

    if TDC(i)<=Rec2
       TDCr(i)=TDC(i);
    else
       TDCr(i)=Rec2;
    end

 end

 ConjDef=max(TDCr,TICr);
 Deff=sum(ConjDef.*ejexs)/sum(ConjDef);
 figure
 subplot(2,1,1),plot(ejexs,TDCr,'--r','LineWidth',2),hold on ,plot(ejexs,TICr,'-
.b','LineWidth',2),grid on
 title('Conjuntos Recortados'),
xlabel('Z'),ylabel('Pertenenencia'),legend('C2','C1')
 subplot(2,1,2),plot(ejexs,ConjDef,'g','LineWidth',1),grid on
 title('Union de conjuntos
(Max)'),xlabel('Z'),ylabel('Pertenenencia'),legend('Conjunto recortado')
 figure
 plot(ejexs,ConjDef,'m','LineWidth',1.5),hold on...
     ,plot([Deff,Deff],[0,1],'r--','linewidth', 1.5)...
     ,plot(Deff,ConjDef,'b.','linewidth', 0.001),grid on,legend('Conjunto
recortado fuzzificado','Eje del centroide')
 title('Defuzzificacion');

 disp(['Z : ',num2str(Deff)])
```

Para la superficie

```
clc,clear all,close all
%definición de parámetros de ejes X Y
xmin=0;ymin=0;
xmax=10;ymax=1;
ejex=xmin:0.1:xmax;
TDA=TrapecioD(ejex,3,7);
TIA=TrapecioI(ejex,3,7);
TDB=TrapecioD(ejex,4,6);
TIB=TrapecioI(ejex,4,6);
xmins=-5;
xmaxs=5;
ejexs=xmins:0.1:xmaxs;
TDC=TrapecioD(ejexs,-1,1);
TIC=TrapecioI(ejexs,-1,1);
Ldf=length(TDA);
it=1;
for i=1:Ldf
    for j=1:Ldf
       r1=min(TIA(i),TIB(j));
       R1=min(r1,TIC);
       r2=min(TIA(i),TDB(j));
       R2=min(r2,TDC);
       r3=min(TDA(i),TIB(j));
```

```
        R3=min(r3,TDC);
        r4=min(TDA(i),TDB(j));
        R4=min(r4,TIC);

        Rec1=max(max(R1,R4));
        Rec2=max(max(R2,R3));

         for k=1:Ldf
             if TIC(k)<=Rec1
                TICr(k)=TIC(k);
             else
                TICr(k)=Rec1;
             end

             if TDC(k)<=Rec2
                TDCr(k)=TDC(k);
             else
                TDCr(k)=Rec2;
             end
         end

          ConjDef=max(TDCr,TICr);
          Deff(it)=sum(ConjDef.*ejexs)/sum(ConjDef);
it=it+1;
    end
end

[X,Y]=meshgrid(ejex);
ute=1;
for k=1:Ldf
    for j=1:Ldf
        Z(k,j)=Deff(ute);
        ute=ute+1;
    end
end
figure
surf(X,Y,Z);xlabel('X'); ylabel('Y'); zlabel('Z');
title('Superficie de control'), colormap('cool');
```

Mendoza Hernandez Yair Alberto
3BM4
Tercer departamental

Graficas

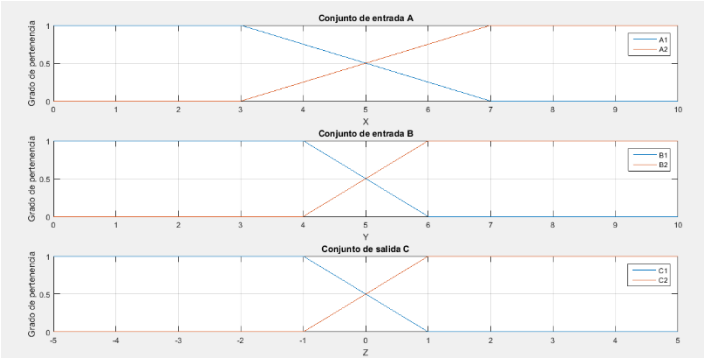


Fig.1 caracterización del universo de trabajo

```
Ingrese el valor de X: 5.6
Ingrese el valor de Y: 4.5
Z : 0.75663
```

Fig. Ingreso de valores propuestos por el usuario y resultado dado por el algoritmo

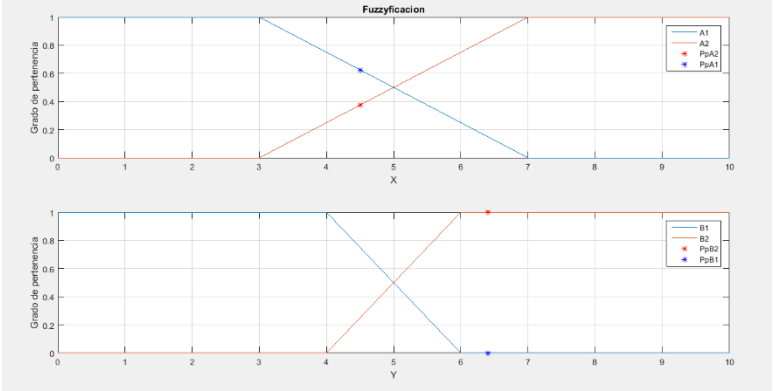


Fig.2. fuzzificacion de los conjuntos de entrada

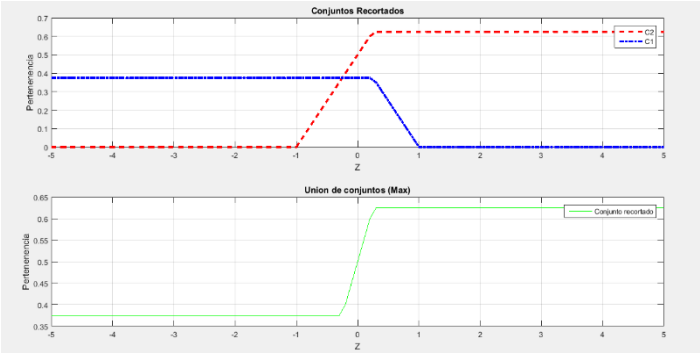


Fig.3. Recorte de conjuntos de salida indicados por el valor de las reglas difusas y unión de ambos mediante máximos

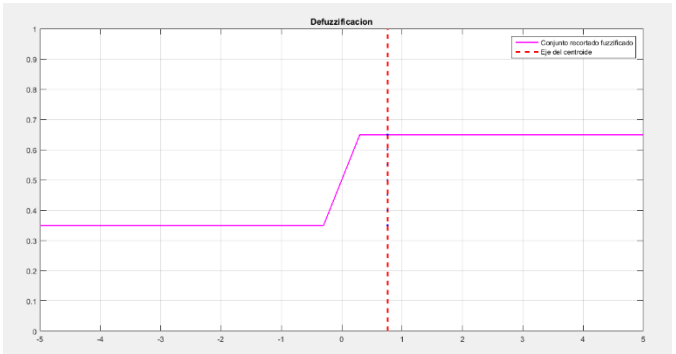


Fig.4 Defuzzificacion del conjunto recortado por el método de centroide

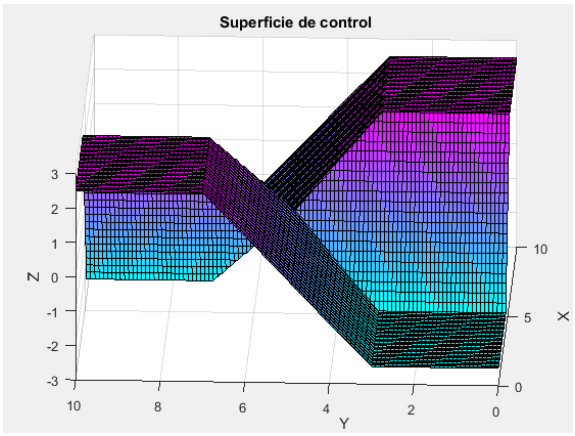


Fig.5. Superficie de control generada por el barrido de todos los puntos de los conjuntos de entrada

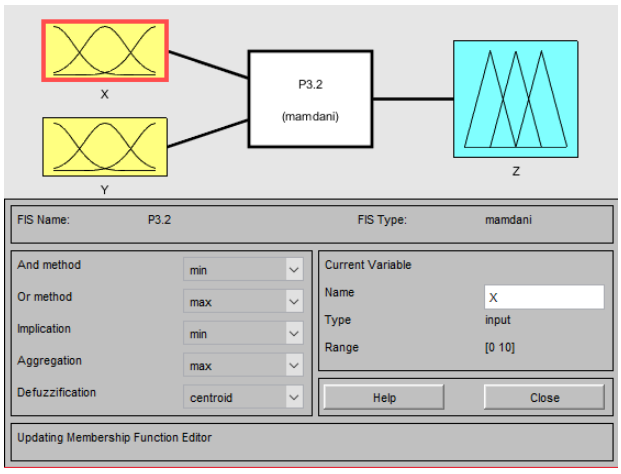


Fig.6.Sistema difuso generado por el toolbox de MatLab

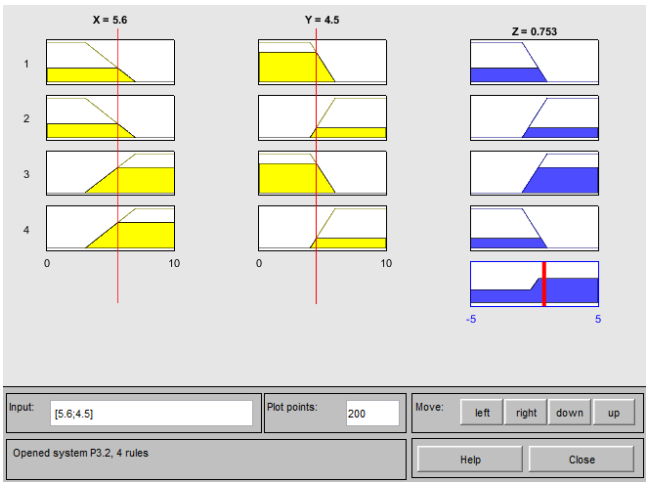


Fig.7 prueba de puntos en el sistema generado por el toolbox

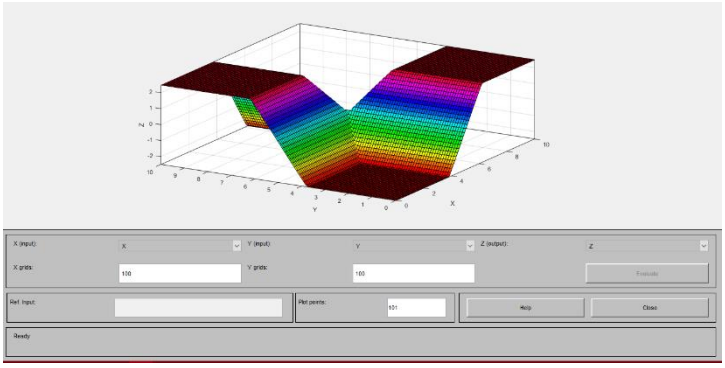


Fig.8. Superficie de control generado por el toolbox de MatLab

Conclusiones

En esta práctica se hace uso de la lógica difusa para encontrar un sistema de control dedicado a un sistema hipotético, en esta práctica se conocieron los principios básicos y las reglas principales para los sistemas difusos de tipo Mandani se pudieron observar además como se hace uso de las reglas de inferencia y de las variables lingüísticas para la descripción de sistemas, mediante la aplicación del toolbox de MatLab se pudo comprobar el correcto funcionamiento del algoritmo propuesto en el cual obtuvimos los mismos resultados en ambos sistemas y la superficie de control cumple con las mismas características , se ingresaron los valores en los conjuntos de entrada x=5.6 y y=4.5 y obtuvimos a la salida un único valor de 0.75 tanto en el algoritmo como en el toolbox .

Mendoza Hernandez Yair Alberto
3BM4
Tercer departamental

P3.2.FIS Sistema Mandani control de un generador a partir de la posición de una veleta

Desarrollo y solución propuesta

El objetivo de la práctica consiste en realizar un sistema difuso de tipo Mandani para el control de un generador de energía eólico el cual depende de la posición de una veleta para obtener como respuesta la velocidad a la que debe girar y el sentido al que debe girar para lograr aprovechar el viento lo más posible

Para la realización de esta práctica se propusieron dos conjuntos de entrada los cuales dependían de la posición del generador en el instante actual y en la posición en la que se encuentra la veleta, la posición del generador es implícitamente una conexión retroalimentada debido a que esta misma variable implica también la misma salida del sistema. Los dos conjuntos de entrada a su vez fueron caracterizados por 5 subconjuntos difusos de tipo trapezoidal los cuales están descritos usando como variable lingüística los puntos cardinales siendo las magnitudes descritas de los subconjuntos los grados que se tienen respecto al giro haciendo un espacio de 0 a 359 ° para ambos , como conjuntos de salida tenemos dos los cuales son como ya se mencionó antes la velocidad de giro del generador y la orientación que este adoptara respecto a la veleta , el conjunto de salida de la velocidad esta descrito por 3 tipos de velocidades siendo estas alta media y baja en un rango de 0 a 10 por otro lado el conjunto de la posición del generador esta descrito por el sentido de giro horario anti horario o no moverse .

Una vez que tenemos los conjuntos descritos y nuestro universo de trabajo bien estructurado procedemos a realizar la fuzzificacion de los valores ingresados por el usuario que generaran las salidas correspondientes para cada caso (sentido de giro, velocidad) esta fuzzificacion se hace para cada conjunto por lo cual se obtiene el grado de pertenencia que tiene cada subconjunto en los puntos ingresados anteriormente.

Posterior a fuzzificar se evalúan las reglas difusas generadas para los dos conjuntos de salida estas reglas se muestran mediante las siguientes tablas de inferencia

Tabla 2. Tabla de inferencia para el conjunto de salida de sentido de giro

NOv	NM	AH	AH	H	NM
Ov	AH	NM	AH	AH	H
Sv	H	H	NM	AH	AH
Ev	AH	H	H	NM	AH
NEv	NM	AH	H	H	NM

Tabla 3. Tabla de inferencia para el conjunto de salida de velocidad

	NOg	Og	Sg	Eg	NEg
NOv	B	B	A	M	B
Ov	M	B	M	A	M
Sv	A	M	B	M	A
Ev	M	A	M	B	M
NEv	B	B	A	M	B

Las tablas descritas anteriormente nos dan los valores correspondientes de las evaluación de las reglas en el algoritmo esta evaluación se hace mediante mínimos para después obtener el máximo de estos mínimos pero solo se hacen máximos para los puntos que modifican los mismos conjuntos de salida generando finalmente 3 puntos de corte para los conjuntos de salida tanto para el conjunto de velocidad como para el conjunto de sentido de giro después de encontrar y recortar los conjuntos de salida necesitamos generar un solo conjunto y esto se hace mediante una unión aplicada por medio de los puntos máximos de los conjuntos recortados finalmente obtenemos entonces el conjunto al que deseamos defuzzificar

La defuzzificacion al conjunto generado anteriormente se hace mediante el método de centroide para lo cual se hace uso de la fórmula que se presenta a continuación

- Método del centroide

$$z^* = \frac{\int \mu_C(z) \cdot z dz}{\int \mu_C(z) dz}$$

El valor que nos arroja la defuzzificacion es entonces el valor de interés para nuestro control difuso, es importante recordar que tenemos dos conjuntos de salida por lo tanto tendremos la evaluación de 25 reglas para la salida de velocidad y otras 25 de sentido de gira por lo cual entonces generaremos dos superficies a defuzzificar y obviamente 2 resultados que se representaran mediante 2 superficies de control

Para la generación de las superficies de control ser realiza el mismo procedimiento que el descrito anteriormente pero el ingreso de los puntos de prueba se substituye por un barrido de todo el universo de trabajo generando entonces como salida un vector con todos los valores defuzzificados para cada punto del universo generados este vector solo resta generar la superficie mediante el ploteo de los dos conjuntos de entrada con las salidas defuzzificadas igual que en la práctica anterior se corroboran los resultados tanto de las superficies como de la defuzzificacion con el toolbox de MatLab con lo cual se obtuvo un resultado satisfactorio.

Algoritmo

Para el sistema Mandani

```
clear all, clc, close all;
Ldf=359;
x=linspace(0,359,Ldf);
NOv=trapmf(x, [0 0 9 81]);
Ov=trapmf(x,  [12 83 101 173]);
Sv=trapmf(x,  [99 171 189 260]);
Ev=trapmf(x, [189 260 278 350]);
NEv=trapmf(x,  [278 350 368 440]);

y=linspace(0,359,Ldf);
```

```
NOg=trapmf(y, [0 0 9 81]);
Og=trapmf(y,  [9 81 99 171]);
Sg=trapmf(y,  [99 171 189 260]);
Eg=trapmf(y, [188 260 278 350]);
NEg=trapmf(y,  [278 350 368 440]);

s1=linspace(-5,5,Ldf);
ah=trapmf(s1,  [-5 -5 -4 0]);
nm=trapmf(s1,   [-4 -0.5 0.5 4]);
h=trapmf(s1,   [0 4 5 5]);

V2=linspace(0,10,Ldf);
b=trapmf(V2, [0 0 1 4]);
m=trapmf(V2, [1 4 6 9]);
a=trapmf(V2, [6 9 10 14]);

subplot(2,2,1)
plot(x,NOv,x,Ov,x,Sv,x,Ev,x,NEv); title('Angulo de la veleta (Input)'),grid on
xlabel('Grados'),ylabel('Grados de
pertenencia'),legend('NOv','Ov','Sv','Ev','NEv')
subplot(2,2,2)
plot(y,NOg,y,Og,y,Sg,y,Eg,y,NEg); title('Angulo del generador (Input)'),grid on
xlabel('Grados'),ylabel('Grados de
pertenencia'),legend('NOg','Og','Sg','Eg','NEg')
subplot(2,2,3)
plot(s1,ah,s1,nm,s1,h); title('Direccion de desplazamiento (Output)'),grid on
xlabel('Sentido de giro'),ylabel('Grados de
pertenencia'),legend('Antihorario','No mover','Horario')
subplot(2,2,4)
plot(V2,b,V2,m,V2,a); title('Revoluciones por minuto (Output)'),grid on
xlabel('Velocidad [m/h]'),ylabel('Grado de
pertenencia'),legend('Baja','Media','Alta')

Val1=input('Ingrese el valor del Angulo de la veleta: ')+1; %0-200
Val2=input('Ingrese el valor del Angulo del generado:  ')+1;
Px1=NOv(Val1);Py1=NOg(Val2);
Px2=Ov(Val1);Py2=Og(Val2);
Px3=Sv(Val1);Py3=Sg(Val2);
Px4=Ev(Val1);Py4=Eg(Val2);
Px5=NEv(Val1);Py5=NEg(Val2);
figure
subplot(2,1,1),plot(x,NOv,x,Ov,x,Sv,x,Ev,x,NEv), hold on ...

,plot(x(Val1),Px1,'r*',x(Val1),Px2,'g*',x(Val1),Px3,'b*',x(Val1),Px4,'c*',x(Val1)
,Px5,'m*'),grid on
title('Fuzzyficacion'),xlabel('Angulo de la Veleta'),ylabel('Grado de
pertenencia')
legend('NOv','Ov','Sv','Ev','NEv','PNOv','POv','PSv','PEv','PNEv')
subplot(2,1,2),plot(y,NOg,y,Og,y,Sg,y,Eg,y,NEg), hold on...

,plot(y(Val2),Py1,'r*',y(Val2),Py2,'g*',y(Val2),Py3,'b*',y(Val2),Py4,'c*',y(Val2)
,Py5,'m*'),grid on
xlabel('Angulo del generador'),ylabel('Grado de pertenencia')
legend('NOg','Og','Sg','Eg','NEg','PNOg','POg','PSg','PEg','PNEg')
%%
        vc=min(NOv(Val1),NOg(Val2));                vc=min(Sv(Val1),NOg(Val2));
        g1=min(vc,nm);                              g3=min(vc,ah);
        gg1=min(vc,b);                              gg3=min(vc,a);

        vc=min(Ov(Val1),NOg(Val2));                 vc=min(Ev(Val1),NOg(Val2));
        g2=min(vc,ah);                              g4=min(vc,h);
        gg2=min(vc,b);                              gg4=min(vc,m);
```

```
        vc=min(NEv(Val1),NOg(Val2));
        g5=min(vc,nm);
        gg5=min(vc,b);

        vc=min(NOv(Val1),Og(Val2));
        g6=min(vc,ah);
        gg6=min(vc,m);

        vc=min(Ov(Val1),Og(Val2));
        g7=min(vc,nm);
        gg7=min(vc,b);

        vc=min(Sv(Val1),Og(Val2));
        g8=min(vc,ah);
        gg8=min(vc,m);

        vc=min(Ev(Val1),Og(Val2));
        g9=min(vc,ah);
        gg9=min(vc,a);

        vc=min(NEv(Val1),Og(Val2));
        g10=min(vc,h);
        gg10=min(vc,m);

        vc=min(NOv(Val1),Sg(Val2));
        g11=min(vc,h);
        gg11=min(vc,a);

        vc=min(Ov(Val1),Sg(Val2));
        g12=min(vc,h);
        gg12=min(vc,m);

        vc=min(Sv(Val1),Sg(Val2));
        g13=min(vc,nm);
        gg13=min(vc,b);

        vc=min(Ev(Val1),Sg(Val2));
        g14=min(vc,ah);
        gg14=min(vc,m);

        vc=min(NEv(Val1),Sg(Val2));
        g15=min(vc,ah);
        gg15=min(vc,a);

        vc=min(NOv(Val1),Eg(Val2));
        g16=min(vc,ah);
        gg16=min(vc,m);

        vc=min(Ov(Val1),Eg(Val2));
        g17=min(vc,h);
        gg17=min(vc,a);

        vc=min(Sv(Val1),Eg(Val2));
        g18=min(vc,h);
        gg18=min(vc,m);

        vc=min(Ev(Val1),Eg(Val2));
        g19=min(vc,nm);
        gg19=min(vc,b);

        vc=min(NEv(Val1),Eg(Val2));
        g20=min(vc,ah);
        gg20=min(vc,m);

        vc=min(NOv(Val1),NEg(Val2));
        g21=min(vc,nm);
        gg21=min(vc,b);

        vc=min(Ov(Val1),NEg(Val2));
        g22=min(vc,ah);
        gg22=min(vc,b);

        vc=min(Sv(Val1),NEg(Val2));
        g23=min(vc,h);
        gg23=min(vc,a);

        vc=min(Ev(Val1),NEg(Val2));
        g24=min(vc,h);
        gg24=min(vc,m);

        vc=min(NEv(Val1),NEg(Val2));
        g25=min(vc,nm);
        gg25=min(vc,b);

        Recs1=max(max([g1;g5;g7;g13;g19;g21;g25]));%No mover
        Recs2=max(max([g2;g4;g6;g10;g11;g12;g17;g18;g23;g24]));%Antihorario
        Recs3=max(max([g3;g8;g9;g14;g15;g16;g20;g22]));%Horario

        Recv1=max(max([gg1;gg5;gg6;gg7;gg10;gg13;gg19;gg21;gg25]));%baja
        Recv2=max(max([gg2;gg4;gg8;gg12;gg14;gg16;gg18;gg20;gg22;gg24]));%media
        Recv3=max(max([gg3;gg9;gg11;gg15;gg17;gg23]));%alta

for i=1:Ldf
   if nm(i)<=Recs1
      nmr(i)=nm(i);
   else
      nmr(i)=Recs1;
   end

   if ah(i)<=Recs2
      ahr(i)=ah(i);
   else
      ahr(i)=Recs2;
   end
```

```
    if h(i)<=Recs3
       hr(i)=h(i);
    else
       hr(i)=Recs3;
    end
 end

  for i=1:Ldf
    if b(i)<=Recv1
       br(i)=b(i);
    else
       br(i)=Recv1;
    end

    if m(i)<=Recv2
       mr(i)=m(i);
    else
       mr(i)=Recv2;
    end

    if a(i)<=Recv3
       ar(i)=a(i);
    else
       ar(i)=Recv3;
    end
 end

 ConjDefs=max([nmr;ahr;hr]);
 Deffs=sum(ConjDefs.*s1)/sum(ConjDefs);
 ConjDefv=max([br;mr;ar]);
 Deffv=sum(ConjDefv.*V2)/sum(ConjDefv);

 %para sentido
 figure
 subplot(2,1,1),plot(s1,nmr,'--r','LineWidth',2),hold on ,plot(s1,ahr,'-
.b','LineWidth',2),plot(s1,hr,'g','LineWidth',2),grid on
 title('Conjuntos Recortados'), xlabel('Sentido de giro
'),ylabel('Pertenenencia')
 legend('No mover','Antihorario','Horario')
 subplot(2,1,2),plot(s1,ConjDefs,'m','LineWidth',1),grid on
 title('Union de conjuntos (Max)'),xlabel('Sentido de
giro'),ylabel('Pertenenencia');
 legend('Conjunto a defuzzificar ')
 figure
 plot(s1,ConjDefs,'m','LineWidth',1.5),hold on...
     ,plot([Deffs,Deffs],[0,1],'r--','linewidth', 1.5)...
     ,plot(Deffs,ConjDefs,'b.','linewidth', 0.001),grid on;
 title('Defuzzificacion Direccion '),xlabel('Sentido de
giro'),ylabel('Pertenenencia');
 legend('Conjunto defuzzificado','Eje del centroide');
 %para velocidad
 figure
 subplot(2,1,1),plot(V2,br,'--r','LineWidth',2),hold on ,plot(V2,mr,'-
.b','LineWidth',2),plot(V2,ar,'g','LineWidth',2),grid on
 title('Conjuntos Recortados'), xlabel('velocidad'),ylabel('Pertenenencia')
  legend('V.Baja','V.Media','V.Alta')
 subplot(2,1,2),plot(V2,ConjDefv,'m','LineWidth',1),grid on
 title('Union de conjuntos (Max)'),xlabel('Velocidad'),ylabel('Pertenenencia')
  legend('Conjunto a defuzzificar ')
 figure
 plot(V2,ConjDefv,'m','LineWidth',1.5),hold on...
     ,plot([Deffv,Deffv],[0,1],'r--','linewidth', 1.5)...
```

```
    ,plot(Deffv,ConjDefv,'b.','linewidth', 0.001),grid on;
 title('Defuzzificacion Velocidad'),xlabel('Velocidad
[m/h]'),ylabel('Pertenenencia');
 legend('Conjunto defuzzificado','Eje del centroide');

 disp(['Zs : ',num2str(Deffs)])
 disp(['Zv : ',num2str(Deffv)])
```

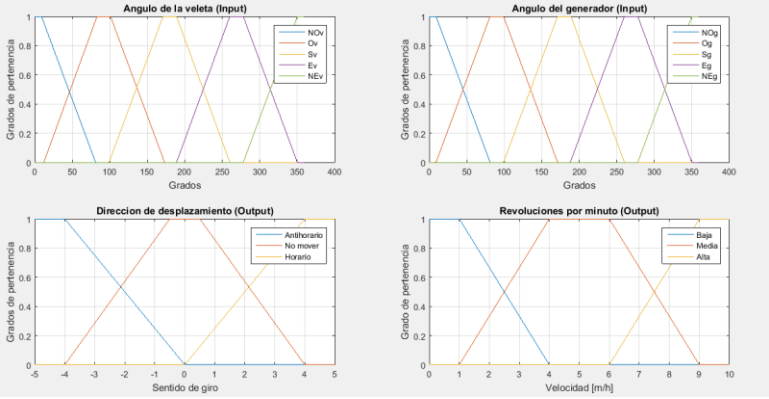
Para la superficie

```
clc,clear all,close all
Ldf=360;
x=linspace(0,359,Ldf);
NOv=trapmf(x, [0 0 9 81]);
Ov=trapmf(x,  [12 83 101 173]);
Sv=trapmf(x,  [99 171 189 260]);
Ev=trapmf(x, [189 260 278 350]);
NEv=trapmf(x,  [278 350 368 440]);

y=linspace(0,359,Ldf);
NOg=trapmf(y, [0 0 9 81]);
Og=trapmf(y,  [9 81 99 171]);
Sg=trapmf(y,  [99 171 189 260]);
Eg=trapmf(y, [188 260 278 350]);
NEg=trapmf(y,  [278 350 368 440]);

s1=linspace(-5,5,Ldf);
ah=trapmf(s1,  [-5 -5 -4 0]);
nm=trapmf(s1,   [-4 -0.5 0.5 4]);
h=trapmf(s1,   [0 4 5 5]);

V2=linspace(0,10,Ldf);
b=trapmf(V2, [0 0 1 4]);
m=trapmf(V2, [1 4 6 9]);
a=trapmf(V2, [6 9 10 14]);
%%
it=1;
for i=1:Ldf
    for j=1:Ldf
        vc=min(NOv(i),NOg(j));

         g1=min(vc,nm);
        gg1=min(vc,b);

        vc=min(Ov(i),NOg(j));
        g2=min(vc,ah);
        gg2=min(vc,b);

        vc=min(Sv(i),NOg(j));
        g3=min(vc,ah);
        gg3=min(vc,a);

        vc=min(Ev(i),NOg(j));
        g4=min(vc,h);
        gg4=min(vc,m);

        vc=min(NEv(i),NOg(j));
        g5=min(vc,nm);
        gg5=min(vc,b);

        vc=min(NOv(i),Og(j));
        g6=min(vc,ah);
        gg6=min(vc,m);

        vc=min(Ov(i),Og(j));
        g7=min(vc,nm);
        gg7=min(vc,b);

        vc=min(Sv(i),Og(j));
        g8=min(vc,ah);
        gg8=min(vc,m);

        vc=min(Ev(i),Og(j));
        g9=min(vc,ah);
        gg9=min(vc,a);

        vc=min(NEv(i),Og(j));
        g10=min(vc,h);
        gg10=min(vc,m);

        vc=min(NOv(i),Sg(j));
        g11=min(vc,h);
        gg11=min(vc,a);

        vc=min(Ov(i),Sg(j));
        g12=min(vc,h);
        gg12=min(vc,m);

        vc=min(Sv(i),Sg(j));
```

```
        g13=min(vc,nm);
        gg13=min(vc,b);

        vc=min(Ev(i),Sg(j));
        g14=min(vc,ah);
        gg14=min(vc,m);

        vc=min(NEv(i),Sg(j));
        g15=min(vc,ah);
        gg15=min(vc,a);

        vc=min(NOv(i),Eg(j));
        g16=min(vc,ah);
        gg16=min(vc,m);

        vc=min(Ov(i),Eg(j));
        g17=min(vc,h);
        gg17=min(vc,a);

        vc=min(Sv(i),Eg(j));
        g18=min(vc,h);
        gg18=min(vc,m);

        vc=min(Ev(i),Eg(j));
        g19=min(vc,nm);
        gg19=min(vc,b);

        vc=min(NEv(i),Eg(j));
        g20=min(vc,ah);
        gg20=min(vc,m);

        vc=min(NOv(i),NEg(j));
        g21=min(vc,nm);
        gg21=min(vc,b);

        vc=min(Ov(i),NEg(j));
        g22=min(vc,ah);
        gg22=min(vc,b);

        vc=min(Sv(i),NEg(j));
        g23=min(vc,h);
        gg23=min(vc,a);

        vc=min(Ev(i),NEg(j));
        g24=min(vc,h);
        gg24=min(vc,m);

        vc=min(NEv(i),NEg(j));
        g25=min(vc,nm);
        gg25=min(vc,b);

        Recs1=max(max([g1;g5;g7;g13;g19;g21;g25]));%No mover
        Recs2=max(max([g2;g4;g6;g10;g11;g12;g17;g18;g23;g24]));%Antihorario
        Recs3=max(max([g3;g8;g9;g14;g15;g16;g20;g22]));%Horario

        Recv1=max(max([gg1;gg5;gg6;gg7;gg10;gg13;gg19;gg21;gg25]));%baja
        Recv2=max(max([gg2;gg4;gg8;gg12;gg14;gg16;gg18;gg20;gg22;gg24]));%media
        Recv3=max(max([gg3;gg9;gg11;gg15;gg17;gg23]));%alta

for l=1:Ldf
   if nm(l)<=Recs1
      nmr(l)=nm(l);
   else
      nmr(l)=Recs1;
   end

   if ah(l)<=Recs2
      ahr(l)=ah(l);
   else
      ahr(l)=Recs2;
   end

   if h(l)<=Recs3
      hr(l)=h(l);
   else
      hr(l)=Recs3;
   end
end

 for k=1:Ldf
   if b(k)<=Recv1
      br(k)=b(k);
   else
      br(k)=Recv1;
   end

   if m(k)<=Recv2
      mr(k)=m(k);
```

```
    else
       mr(k)=Recv2;
    end

    if a(k)<=Recv3
       ar(k)=a(k);
    else
       ar(k)=Recv3;
    end
  end
     ConjDefs=max([nmr;ahr;hr]);
     Deffs(it)=sum(ConjDefs.*s1)/sum(ConjDefs);
     ConjDefv=max([br;mr;ar]);
     Deffv(it)=sum(ConjDefv.*V2)/sum(ConjDefv);
     it=it+1;
    end
end
[Xs,Ys]=meshgrid(x);
ute=1;
for i=1:Ldf
    for j=1:Ldf
        Zs(i,j)=Deffs(ute);
        ute=ute+1;
    end
end
figure
mesh(Xs,Ys,Zs);xlabel('X'); ylabel('Y'); zlabel('Z'),title('Superficie sentido');

[Xv,Yv]=meshgrid(y);
ute=1;
for i=1:Ldf
    for j=1:Ldf
        Zv(i,j)=Deffv(ute);
        ute=ute+1;
    end
end
figure
mesh(Xv,Yv,Zv);xlabel('X'); ylabel('Y'); zlabel('Z'),title('Superficie
velocidad')
```

Graficas




```
Ingrese el valor del Angulo de la veleta: 40
Ingrese el valor del Angulo del generado:  30
Zs : -0.67969
Zv : 3.8605
``
```

Fig.1. Universo de discurso caracterización conjuntos de entrada y de salida

Mendoza Hernandez Yair Alberto
3BM4
Tercer departamental

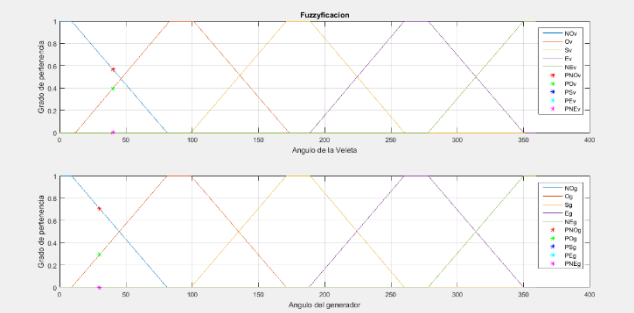


Fig.2.Fuzzificacion de los conjuntos de entrada

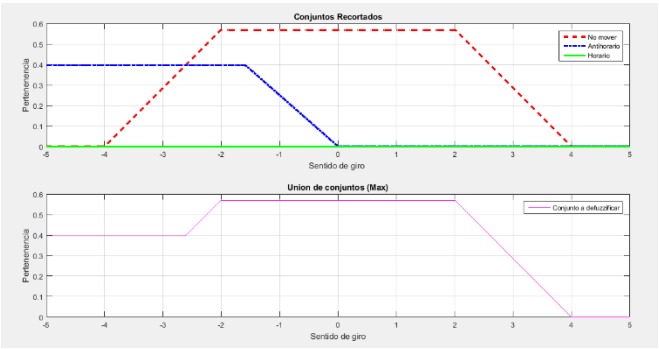


Fig.3.Recorte de los conjuntos de salida para el sentido de giro

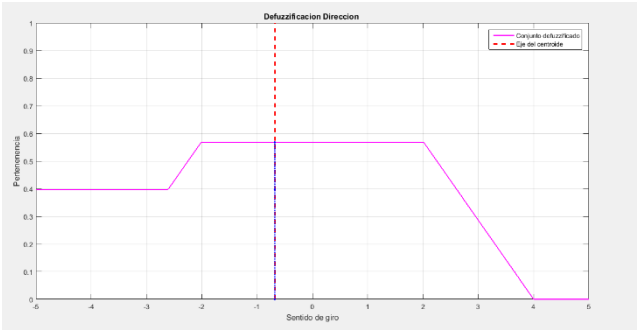


Fig.4.Defuzzificacion del conjunto recortado mediante el centroide

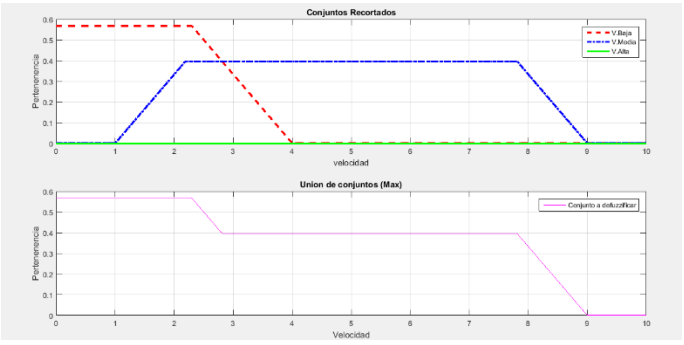


Fig.5.Recorte de los conjuntos de salida para la velocidad del generador

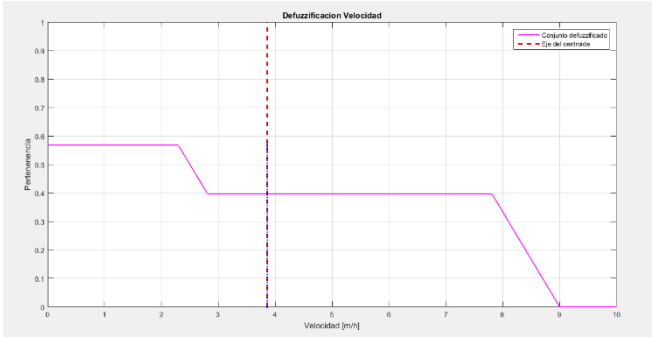


Fig.6.Defuzzificacion del conjunto recortado para los valores ingresados por el usuario para la velocidad

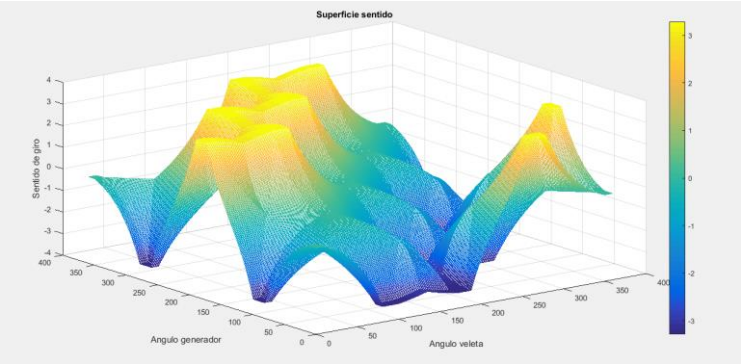
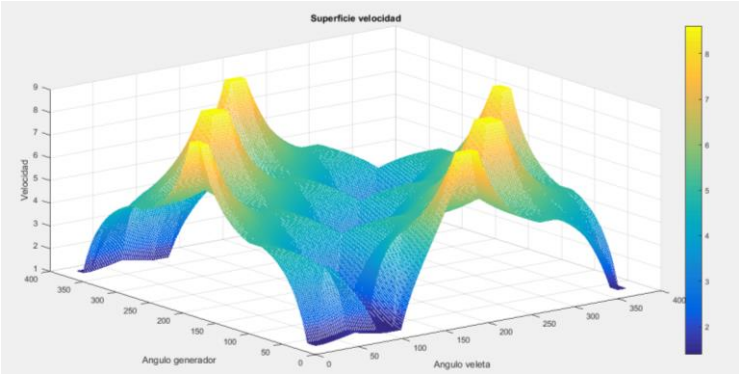


Fig.7.Superficie generada para el sentido de giro

Fig.8.Superficie generada para la velocidad



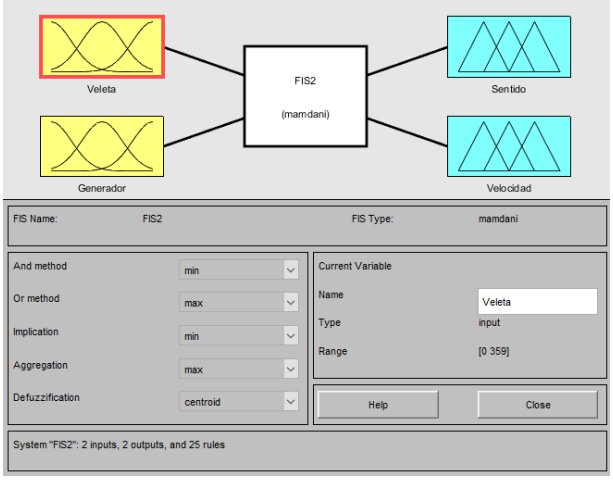


Fig.9.Sistema de tipo Mandani generado por el toolbox para el control de un generador

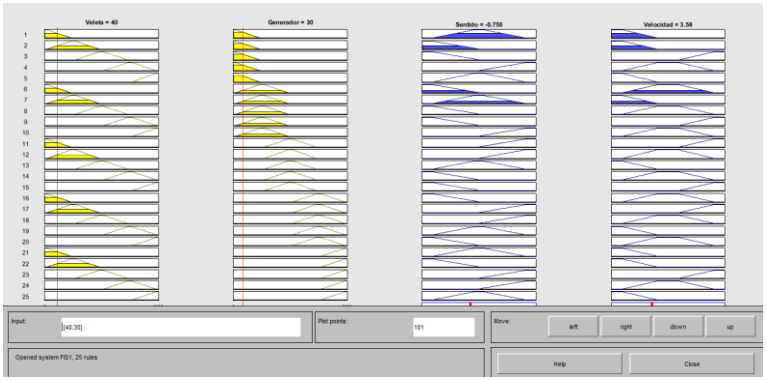


Fig.10.Comprobacion de los puntos de entrada evaluados en el sistema generado por el toolbox

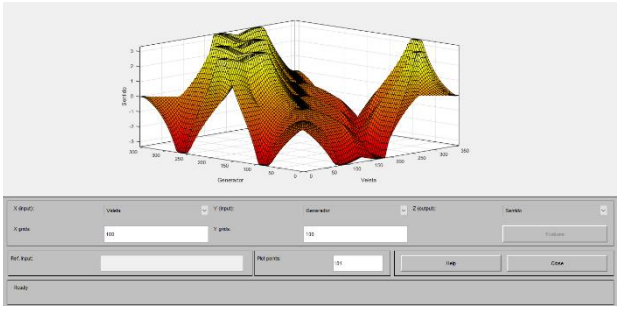


Fig.11.Superficie generada para el sentido de giro mediante le toolbox de MatLab

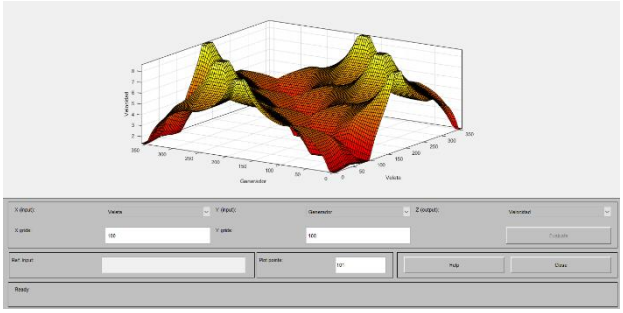


Fig.12.Superficie generada para la velocidad mediante le toolbox de MatLab

Conclusiones

En esta práctica se puede observar el uso de los sistemas difusos para generar un sistema de control aplicado a un caso real y que para la realización de este no es necesario nada más que la lógica difusa esta herramienta resulta muy eficiente para realizar sistemas de control sin meter un alto grado de evaluaciones matemáticas que pueden hacer un sistema muy complejo como en el caso de un control clásico el uso de la lógica difusa para realizar este tipo de control es práctico y eficiente en esta práctica además se hace la comparación entre las dos formas de generar el sistema de control tanto por el algoritmo propuesto como por la herramienta propia de MatLab los resultados que se obtuvieron para el ingreso de un solo punto fueron lo suficientemente cercanos como para inferir que el sistema de control funciona idénticamente uno del otro podemos observar como para nuestro algoritmo se hizo el ingreso de los puntos 40 y 30 grados obteniendo valores defuzzificados de -0.67 y 3.8 al ingresar los mismos valores a reglas del toolbox y los resultados defuzzificados fueron -0,75 y 3.58 si bien se encontraron una diferencia al corroborar las superficies para ambos casos se vio que el sistema correspondía sin ningún problema

REFERENCIAS

www.esi.uclm.es/www/cglez/downloads/docencia/2011.../LogicaDifusa.pdf