EVALUACIÓN PRIMER DEPARTAMENTAL: DISEÑO DE UN SISTEMA DE INFERENCIA DE TIPO MAMDANI PARA EL CONTROL DE UN CLIMATIZADOR

Instituto Politécnico Nacional
"La Técnica al Servicio de la Patria"

Ingeniería Telemática

Lógica Difusa

Abarca Romero José Ángel

Profesora

Gonzalez Navarro Yesenia Eleon

2TM9

Índice

1. Objetivo

- Diseñar un sistema de inferencia de tipo Mamdani que para la resolución de un problema a mediante un programa en Python.

2. Propuesta de problema a solucionar

Suponiendo que deso controlar la temperatura en una habitación con flujo estático de aire, coloco un sistema acondicionador que regula la temperatura del aire que arroja, así como su velocidad dependiendo de la sensación térmica que puede percibir mediante sus sensores. Los sensores se encargan de medir la temperatura seca y la humedad relativa de la habitación, las cuales se usan para determinar la sensación térmica de la habitación. El sistema, de forma automática, toma la decisión de aumentar o disminuir el flujo del aire, así como de la temperatura que este tiene.

3. Desarrollo

3.1. Caracterización de los universos de entrada

3.1.1. Temperatura seca

Suponiendo que el sistema de aire acondicionado se instala en una habitación cuya temperatura oscila entre 0 y 50 C°, podemos definir una variable llamada "temperatura seca", la cual está definida entre 0 y 50. Entiéndase temperatura seca como la temperatura que se puede censar en el aire sin tomar en consideración la radiación calorífica de los objetos que rodean el ambiente así su humedad relativa y la velocidad del aire. Dentro de esta variable definimos los siguientes conjuntos: *Frio, Templado, Caliente y Muy Caliente.*

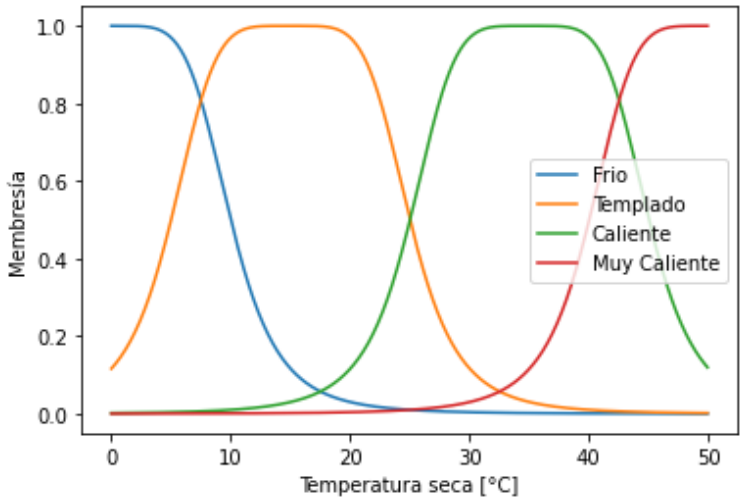

Figura 1: Universo de entrada 1: Temperatura seca en °C.

3.1.2. Humedad relativa

Se define la humedad relativa como la relación entre la fracción molar del vapor de agua en el aire y la fracción molar del vapor de agua en el aire saturado a la misma temperatura. Se denota por la letra griega Φ y es adimensional. El coeficiente se encuentra en el rango de 0 a 100, siendo el 100 el valor máximo de saturación que puede alcanzar el vapor de agua en el aire con relación al máximo valor que podría tener si estuviera saturado. Con esta variable de entrada generamos los siguientes conjuntos: *Seco, Húmedo y Zona de Comfort.*

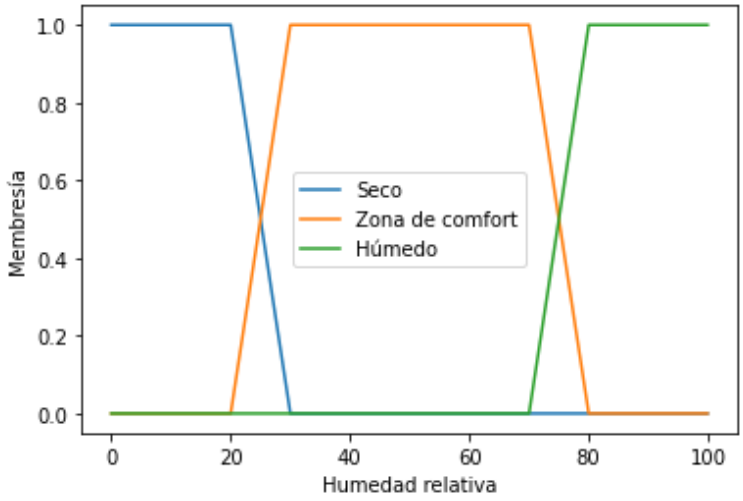

Figura 2: Universo de entrada 2: Humedad relativa (adimensional).

3.2. Caracterización de los universos de salida

Como mencioné en el planteamiento del problema, el aire acondicionado tiene dos variables de salida: la velocidad con la que sale el flujo del aire y la temperatura que éste tiene.

3.2.1. Temperatura del aire

A diferencia de la variable "temperatura seca" planteada anteriormente, la temperatura del aire que sale del sistema de ambientación tiene un rango de operación más limitado, pues de forma convencional, estos operan entre 16 y 28 C°. Para este caso en específico, consideraré que la temperatura del aire que arroja el dispositivo oscila entre 15 y 30 C°. Dentro de esta variable consideraré los siguientes conjuntos *Frio*, *Templado* y *Caliente*.

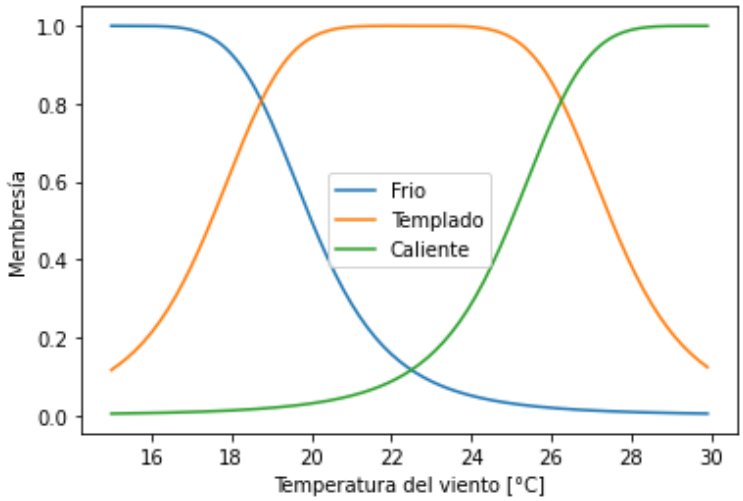

Figura 3: Universo de salida 1: Temperatura del flujo de aire en °C.

3.2.2. Velocidad del flujo de aire

Para medir qué tan rápido irá el flujo de aire utilizaré la escala Beaufort, la cual asigna números en una escala que va del 0 al 12, siendo 0 "calma" y 12 "huracán". Como es de esperarse, el sistema de refrigeración no requiere de la consideración de la escala completa, por lo que tomaré únicamente los valores 0, 1, 2 y 3, los cuales son: *calma, aire ligero, brisa ligera y brisa suave* cuyas velocidades son 0.5 m/s, 1.5 m/s, 3 m/s y 6 m/s respectivamente.

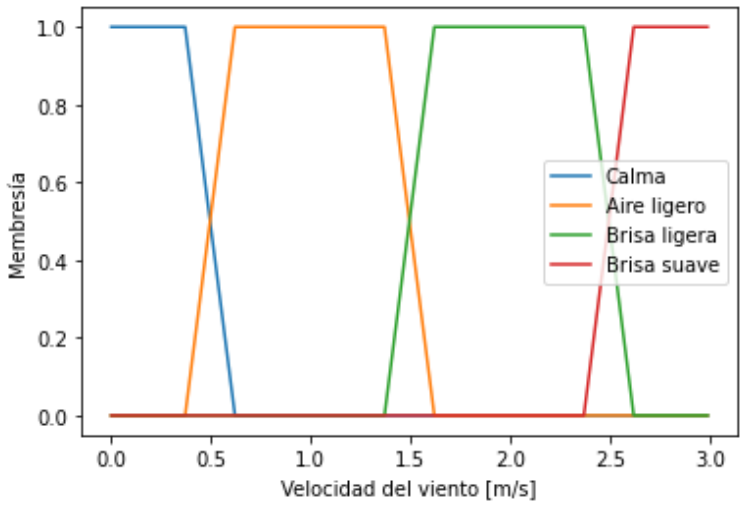

Figura 4: Universo de salida 2: Velocidad del flujo del aire en metros/segundos.

3.3. Tablas de inferencia hacia los conjuntos de salida

Las tablas de inferencia son las que determinan las reglas "si - entonces". Al tener dos variables difusas a la salida del sistema, diseñé dos tablas de inferencia por separado.

Humedad/Temperatura	Frío	Templado	Caliente	Muy caliente
Seco	Caliente	Templado	Frio	Frio
Zona de comfort	Templado	Templado	Templado	Frio
Húmedo	Caliente	Templado	Frio	Frio

Cuadro 1: Tabla de inferencia hacia Temperatura del aire.

Humedad/Temperatura	Frío	Templado	Caliente	Muy caliente
Seco	Aire ligero	Aire ligero	Brisa ligera	Brisa suave
Zona de comfort	Aire ligero	Calma	Brisa ligera	Brisa suave
Húmedo	Aire ligero	Aire ligero	Brisa ligera	Brisa suave

Cuadro 2: Tabla de inferencia para Velocidad del viento

En base a la tabla de inferencia 1 propongo las siguientes reglas "si - entonces":

1. Si el aire es seco y su temperatura fría, entonces la temperatura de salida es Caliente.
2. Si el aire es seco y su temperatura es templada, entonces la temperatura de salida es Templada.
3. Si el aire es seco y su temperatura es caliente , entonces la temperatura de salida es Fría.
4. Si el aire es seco y su temperatura es muy caliente , entonces la temperatura de salida es Fría.

5. Si el aire está en zona de comfort y su temperatura fría, entonces la temperatura de salida es Templada.
6. Si el aire está en zona de comfort y su temperatura templada, entonces la temperatura de salida es Templada.
7. Si el aire está en zona de comfort y su temperatura caliente, entonces la temperatura de salida es Templada.
8. Si el aire está en zona de comfort y su temperatura muy caliente, entonces la temperatura de salida es Fría.
9. Si el aire es húmedo y su temperatura fría, entonces la temperatura de salida es Caliente.
10. Si el aire es húmedo y su temperatura es templada, entonces la temperatura de salida es Templada.
11. Si el aire es húmedo y su temperatura es caliente , entonces la temperatura de salida es Fría.
12. Si el aire es húmedo y su temperatura es muy caliente , entonces la temperatura de salida es Fría.

Ahora muestro las reglas "si - entonces" para la tabla de inferencia 2:

1. Si el aire es seco y su temperatura fría, entonces la velocidad del viento es Aire ligero.
2. Si el aire es seco y su temperatura es templada, entonces la velocidad del viento es Aire ligero.
3. Si el aire es seco y su temperatura es caliente , entonces la velocidad del viento es Brisa ligera.
4. Si el aire es seco y su temperatura es muy caliente , entonces la velocidad del viento es Brisa suave.
5. Si el aire está en zona de comfort y su temperatura fría, entonces la velocidad del viento es Aire ligero.
6. Si el aire está en zona de comfort y su temperatura templada, entonces la velocidad del viento es Calma.
7. Si el aire está en zona de comfort y su temperatura caliente, entonces la velocidad del viento es Brisa ligera.
8. Si el aire está en zona de comfort y su temperatura muy caliente, entonces la velocidad del viento es Brisa suave.
9. Si el aire es húmedo y su temperatura fría, entonces la velocidad del viento es Aire ligero.
10. Si el aire es húmedo y su temperatura es templada, entonces la velocidad del viento es Aire ligero.
11. Si el aire es húmedo y su temperatura es caliente , entonces la velocidad del viento es Brisa ligera.
12. Si el aire es húmedo y su temperatura es muy caliente , entonces la velocidad del viento es Brisa suave.

3.4. Programación del sistema de inferencia en Python

Con base a las reglas "si - entonces" generadas para las tablas de inferencia 1 y 2, modifiqué el programa que realicé para la práctica 1.2. Los cambios relevantes consisten en ajustar el tamaño de las tablas de inferencia y aplicar correctamente las reglas "si - entonces" para lo requerido por este ejercicio. A continuación muestro las gráficas correspondientes al proceso de desdifusificación de un valor difuso para cada uno de los universos de salida a partir de dos valores de entrada $x = 45$ y $y = 67$.

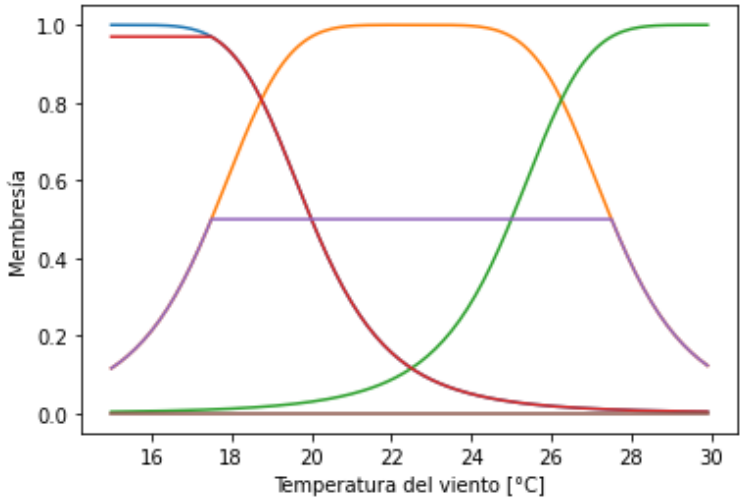

Figura 5: Cortes en los conjuntos del universo de salida "Temperatura del flujo de aire".

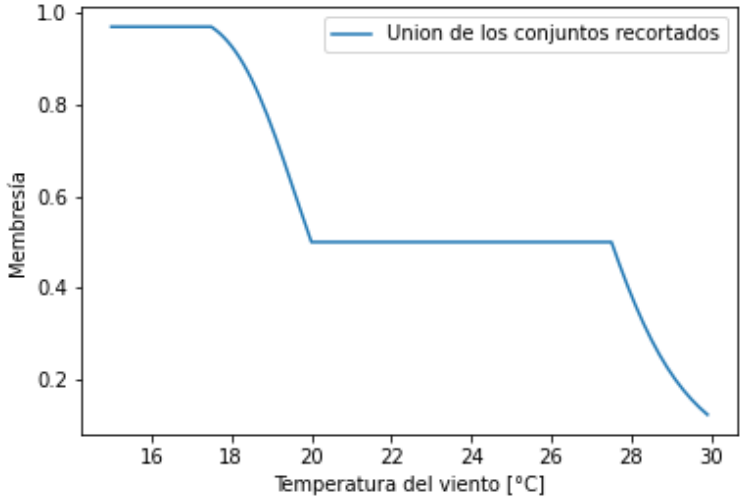

Figura 6: Unión de los conjuntos recortados

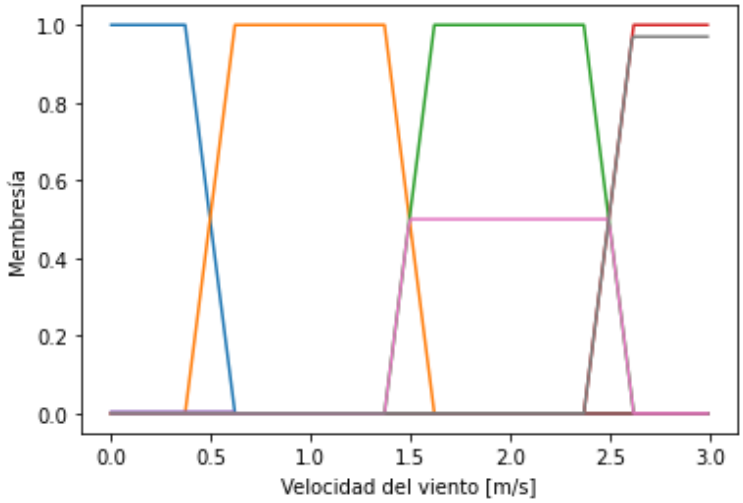

Figura 7: Cortes a los conjuntos del universo de salida "Velocidad del flujo de aire".

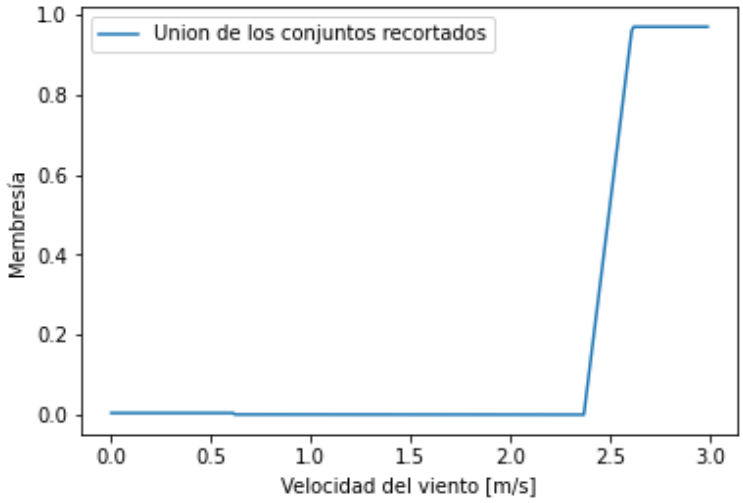

Figura 8: Unión de los conjuntos recortados

```
Ingrese el valor de la temperatura seca (0 - 50): 45
Ingrese el valor de la humedad relativa (0 - 100): 67
Tabla de inferencia:  [[0.0, 0.0, 0.0, 0.0], [0.0005416292886037812,
0.004098360655737705, 0.5, 0.9696969696969697], [0.0, 0.0, 0.0, 0.0]]
z1* =  20.86370103546008  [°C]
z2* =  2.72389655917784 [m/s]
```

Figura 9: Tabla de inferencia y valores desfusificados para $x = 45$ y $y = 67$.

En la siguiente sección muestro las superficies de control obtenidas.

4. Resultados

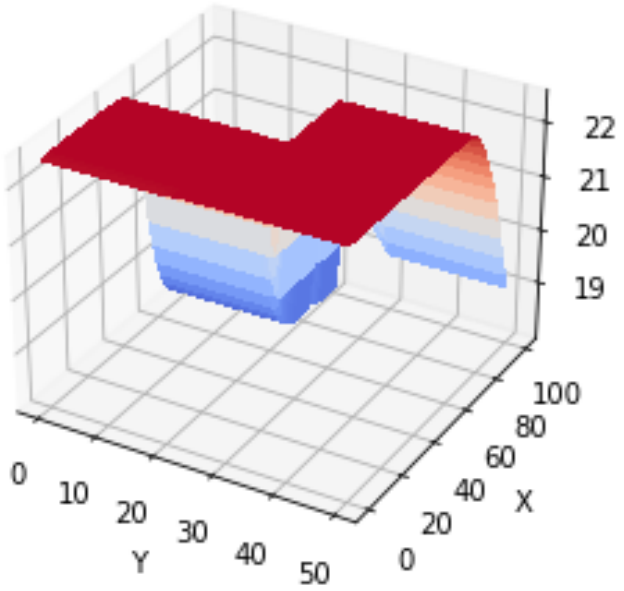

Figura 10: Superficie de control para "Temperatura del viento".

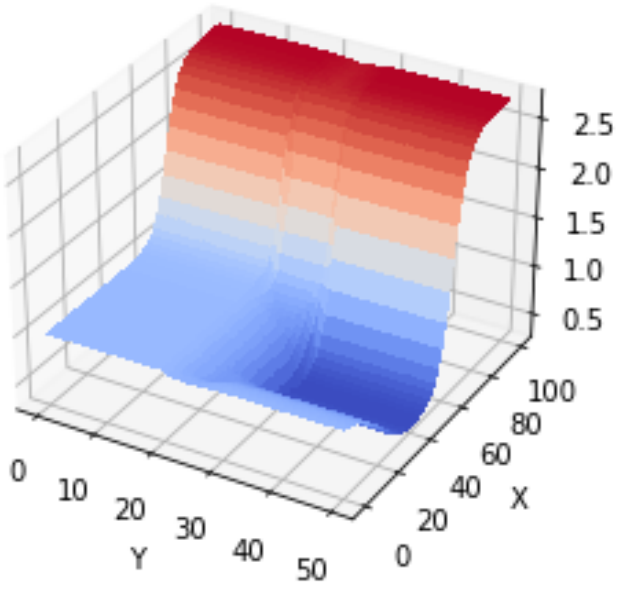

Figura 11: Superficie de control para "Velocidad del viento".

5. Conclusiones

A diferencia de las prácticas 1.1 y 1.2, para caracterizar las variables de este ejercicio utilicé rangos de operación distintos uno respecto del otro, pues su comportamiento está limitado de cierta forma para hacerlo coincidir con el comportamiento que tendría en la vida real. Con ello me di cuenta de la complejidad que conlleva hacer que las tablas de inferencia y las reglas "si - entonces" sean coherentes con el comportamiento real del sistema. Además, esto mismo requirió que se realizaran ajustes en el modo de operación de las funciones que había definido previamente, pues el hecho de que ciertas variables no inicien en 0 implica que ciertos desplazamientos horizontales se necesitan para hacer que los resultados tengan sentido.

6. Apéndices

Código para la generación de gráficas de los conjuntos de entrada y salida caracterizados {graficas.py}

```
import numpy as np
import matplotlib.pyplot as plt
x = np.arange(0, 50, 0.1)
y = np.arange(0, 100, 0.1)
z1 = np.arange(15, 30, 0.1)
z2 = np.arange(0, 3, 0.01)
# X, Y = np.meshgrid(x, y)
#Temperatura seca
A1 = np.zeros(len(x)) #Frio
A2 = np.zeros(len(x)) #Templado
A3 = np.zeros(len(x)) #Caliente
A4 = np.zeros(len(x)) #Muy caliente
#Humedad relativa
B1 = np.zeros(len(y)) #Seco
B2 = np.zeros(len(y)) #Zona de comfort
B3 = np.zeros(len(y)) #Húmedo
#Temperatura del aire
C1 = np.zeros(len(z1)) #Frio
C2 = np.zeros(len(z1)) #Templado
C3 = np.zeros(len(z1)) #Caliente
#Velocidad del viento
D1 = np.zeros(len(z2)) #Calma
D2 = np.zeros(len(z2)) #Aire ligero
D3 = np.zeros(len(z2)) #Brisa ligera
D4 = np.zeros(len(z2)) #Brisa suave
def funcionCampana(a,b,c,C,r):
a = a*10 #Ancho de la campana
b = b*10 #Razon de cambio
c = c*10 #Centro de la campana
for i in range(len(r)):
C[i] = 1/(1 + abs((i-c)/a)**(2*b))
return C
def funcionTrapezoidal(a,b,c,d,C,r):
a = a*10
b = b*10
c = c*10
d = d*10
for i in range(len(r)):
if i <= a:
C[i] = 0
elif a <i and i <= b:
C[i] = (i - a)/(b - a)
elif b <i and i <= c:
C[i] = 1
elif c <i and i <= d:
C[i] = (d - i)/(d - c)
elif d <i:
C[i] = 0
return C
```

```
#Generación de los conjuntos difusos
#Temperatura seca
A1 = funcionCampana(10,0.25,0,A1,x)
A2 = funcionCampana(10,0.25,15,A2,x)
A3 = funcionCampana(10,0.25,35,A3,x)
A4 = funcionCampana(10,0.25,50,A4,x)
plt.figure(1)
plt.plot(x,A1,x,A2,x,A3,x,A4)
plt.xlabel("Temperatura seca [°C]")
plt.ylabel("Membresía")
#Humedad relativa
B1 = funcionTrapezoidal(-1,-1,20,30,B1,y)
B2 = funcionTrapezoidal(20,30,70,80,B2,y)
B3 = funcionTrapezoidal(70,80,100,100,B3,y)
plt.figure(2)
plt.plot(y,B1,y,B2,y,B3)
plt.xlabel("Humedad relativa")
plt.ylabel("Membresía")
# Temperatura del viento
C1 = funcionCampana(5,0.25,0,C1,z1)
C2 = funcionCampana(5,0.25,7.5,C2,z1)
C3 = funcionCampana(5,0.25,15,C3,z1)
plt.figure(3)
plt.plot(z1,C1,z1,C2,z1,C3)
plt.xlabel("Temperatura del viento [°C]")
plt.ylabel("Membresía")
#Velocidad del viento
D1 = funcionTrapezoidal(-1*10,-1*10,0.37*10,0.62*10,D1,z2)
D2 = funcionTrapezoidal(0.37*10,0.62*10,1.37*10,1.62*10,D2,z2)
D3 = funcionTrapezoidal(1.37*10,1.62*10,2.37*10,2.62*10,D3,z2)
D4 = funcionTrapezoidal(2.37*10,2.62*10,3*10,3*10,D4,z2)
plt.figure(4)
plt.plot(z2,D1,z2,D2,z2,D3,z2,D4)
plt.xlabel("Velocidad del viento [m/s]")
plt.ylabel("Membresía")
#CZ = np.zeros((len(x),len(y)))
#Obtención de los grados de pertenencia a los conjuntos de los universos de entrada
valx = int(input("Ingrese el valor de la temperatura seca (0 - 50): "))
memA = [A1[valx*10],A2[valx*10],A3[valx*10],A4[valx*10]]
valy = int(input("Ingrese el valor de la humedad relativa (0 - 100): "))
memB = [B1[valy*10],B2[valy*10],B3[valy*10]]
#Composición max - min (Evaluación de las reglas "si-entonces")
#Tablas de inferencia (intersección - min)
tabla = [[np.minimum(memA[0],memB[0]),np.minimum(memA[1],memB[0]),
np.minimum(memA[2],memB[0]),np.minimum(memA[3],memB[0])],
[np.minimum(memA[0],memB[1]),np.minimum(memA[1],memB[1]),
np.minimum(memA[2],memB[1]),np.minimum(memA[3],memB[1])],
[np.minimum(memA[0],memB[2]),np.minimum(memA[1],memB[2]),
np.minimum(memA[2],memB[2]),np.minimum(memA[3],memB[2])]]
print("Tabla de inferencia: ",str(tabla))
#Agregación - max hacia Temperatura del viento
agregacionC1 = max(tabla[0][2],max(tabla[0][3],max(tabla[1][3],
max(tabla[2][2],tabla[2][3])))) #Frio
agregacionC2 = max(tabla[0][1],max(tabla[1][0],max(tabla[1][1],
```

```
max(tabla[1][2],tabla[2][1]))) #Templado
agregacionC3 = max(tabla[0][0],tabla[2][0]) #Caliente
#Agregación - max hacia Velocidad del viento
agregacionD1 = tabla[1][1] #Calma
agregacionD2 = max(tabla[0][0],max(tabla[0][1],max(tabla[1][0],
max(tabla[2][0],tabla[2][1]))) #Aire ligero
agregacionD3 = max(tabla[0][2],max(tabla[1][2],tabla[2][2])) #Brisa ligera
agregacionD4 = max(tabla[0][3],max(tabla[1][3],tabla[2][3])) #Brisa suave
aux1 = agregacionC1*np.ones(len(C1))
aux2 = agregacionC2*np.ones(len(C2))
aux3 = agregacionC3*np.ones(len(C3))
aux4 = agregacionD1*np.ones(len(D1))
aux5 = agregacionD2*np.ones(len(D2))
aux6 = agregacionD3*np.ones(len(D3))
aux7 = agregacionD4*np.ones(len(D4))
C1_r = np.minimum(C1,aux1)
C2_r = np.minimum(C2,aux2)
C3_r = np.minimum(C3,aux3)
D1_r = np.minimum(D1,aux4)
D2_r = np.minimum(D2,aux5)
D3_r = np.minimum(D3,aux6)
D4_r = np.minimum(D4,aux7)
plt.figure(5)
plt.plot(z1,C1,z1,C2,z1,C3,z1,C1_r,z1,C2_r,z1,C3_r)
plt.xlabel("Temperatura del viento [°C]")
plt.ylabel("Membresía")
plt.figure(6)
plt.plot(z2,D1,z2,D2,z2,D3,z2,D4,z2,D1_r,z2,D2_r,z2,D3_r,z2,D4_r)
plt.xlabel("Velocidad del viento [m/s]")
plt.ylabel("Membresía")
#Dedifusificación
#Temperatura del viento
z_d = 0
nume = 0
deno = 0
CT1 = np.maximum(C1_r,C2_r,C3_r)
plt.figure(7)
plt.plot(z1,CT1)
plt.xlabel("Temperatura del viento [°C]")
plt.ylabel("Membresía")
for i in range(len(z1)):
nume += CT1[i]*(i/10+15)
deno += CT1[i]
z_d = nume/deno
print("z1* = ", str(z_d), " [°C]")
# Velocidad del viento
z_d = 0
nume = 0
deno = 0
CT2 = np.maximum(np.maximum(D1_r,D2_r,D3_r),D4_r)
plt.figure(8)
plt.plot(z2,CT2)
plt.xlabel("Velocidad del viento [m/s]")
plt.ylabel("Membresía")
```

```
for i in range(len(z2)):
nume += CT2[i]*i/100
deno += CT2[i]
z_d = nume/deno
print("z2* = ", str(z_d), "[m/s]")
```

Código para la generación de superficies de control {3d.py}

```
import numpy as np
import matplotlib.pyplot as plt
from matplotlib import cm
x = np.arange(0, 50, 0.1)
y = np.arange(0, 100, 0.2)
z1 = np.arange(15, 30, 0.1)
z2 = np.arange(0, 3, 0.01)
X, Y = np.meshgrid(x, y)
#Temperatura seca
A1 = np.zeros(len(x)) #Frio
A2 = np.zeros(len(x)) #Templado
A3 = np.zeros(len(x)) #Caliente
A4 = np.zeros(len(x)) #Muy caliente
#Humedad relativa
B1 = np.zeros(len(y)) #Seco
B2 = np.zeros(len(y)) #Zona de comfort
B3 = np.zeros(len(y)) #Húmedo
#Temperatura del aire
C1 = np.zeros(len(z1)) #Frio
C2 = np.zeros(len(z1)) #Templado
C3 = np.zeros(len(z1)) #Caliente
#Velocidad del viento
D1 = np.zeros(len(z2)) #Calma
D2 = np.zeros(len(z2)) #Aire ligero
D3 = np.zeros(len(z2)) #Brisa ligera
D4 = np.zeros(len(z2)) #Brisa suave
def funcionCampana(a,b,c,C,r):
a = a*10 #Ancho de la campana
b = b*10 #Razon de cambio
c = c*10 #Centro de la campana
for i in range(len(r)):
C[i] = 1/(1 + abs((i-c)/a)**(2*b))
return C
def funcionTrapezoidal(a,b,c,d,C,r):
a = a*10
b = b*10
c = c*10
d = d*10
for i in range(len(r)):
if i <= a:
C[i] = 0
elif a <i and i <= b:
C[i] = (i - a)/(b - a)
elif b <i and i <= c:
C[i] = 1
elif c <i and i <= d:
```

```
C[i] = (d - i)/(d - c)
elif d <i:
C[i] = 0
return C
#Generación de los conjuntos difusos
#Temperatura seca
A1 = funcionCampana(10,0.25,0,A1,x)
A2 = funcionCampana(10,0.25,15,A2,x)
A3 = funcionCampana(10,0.25,35,A3,x)
A4 = funcionCampana(10,0.25,50,A4,x)
#Humedad relativa
B1 = funcionTrapezoidal(-1,-1,20,30,B1,y)
B2 = funcionTrapezoidal(20,30,70,80,B2,y)
B3 = funcionTrapezoidal(70,80,100,100,B3,y)
# Temperatura del viento
C1 = funcionCampana(5,0.25,0,C1,z1)
C2 = funcionCampana(5,0.25,7.5,C2,z1)
C3 = funcionCampana(5,0.25,15,C3,z1)
#Velocidad del viento
D1 = funcionTrapezoidal(-1*10,-1*10,0.37*10,0.62*10,D1,z2)
D2 = funcionTrapezoidal(0.37*10,0.62*10,1.37*10,1.62*10,D2,z2)
D3 = funcionTrapezoidal(1.37*10,1.62*10,2.37*10,2.62*10,D3,z2)
D4 = funcionTrapezoidal(2.37*10,2.62*10,3*10,3*10,D4,z2)
CZ1 = np.zeros((len(x),len(y)))
CZ2 = np.zeros((len(x),len(y)))
#Obtención de los grados de pertenencia a los conjuntos de los universos de entrada
def difusificacion(valx,valy):
memA = [A1[valx],A2[valx],A3[valx],A4[valx]]
memB = [B1[valy],B2[valy],B3[valy]]
#Composición max - min (Evaluación de las reglas "si-entonces")
#Tablas de inferencia (intersección - min)
tabla = [[np.minimum(memA[0],memB[0]),np.minimum(memA[1],
memB[0]),np.minimum(memA[2],memB[0]),np.minimum(memA[3],memB[0])],
[np.minimum(memA[0],memB[1]),np.minimum(memA[1],
memB[1]),np.minimum(memA[2],memB[1]),np.minimum(memA[3],memB[1])],
[np.minimum(memA[0],memB[2]),np.minimum(memA[1],
memB[2]),np.minimum(memA[2],memB[2]),np.minimum(memA[3],memB[2])]]
#Agregación - max hacia Temperatura del viento
agregacionC1 = max(tabla[0][2],max(tabla[0][3],
max(tabla[1][3],max(tabla[2][2],tabla[2][3])))) #Frio
agregacionC2 = max(tabla[0][1],max(tabla[1][0],
max(tabla[1][1],max(tabla[1][2],tabla[2][1])))) #Templado
agregacionC3 = max(tabla[0][0],tabla[2][0]) #Caliente
#Agregación - max hacia Velocidad del viento
agregacionD1 = tabla[1][1] #Calma
agregacionD2 = max(tabla[0][0],max(tabla[0][1],
max(tabla[1][0],max(tabla[2][0],tabla[2][1])))) #Aire ligero
agregacionD3 = max(tabla[0][2],max(tabla[1][2],tabla[2][2])) #Brisa ligera
agregacionD4 = max(tabla[0][3],max(tabla[1][3],tabla[2][3])) #Brisa suave
aux1 = agregacionC1*np.ones(len(C1))
aux2 = agregacionC2*np.ones(len(C2))
aux3 = agregacionC3*np.ones(len(C3))
aux4 = agregacionD1*np.ones(len(D1))
aux5 = agregacionD2*np.ones(len(D2))
```

```
aux6 = agregacionD3*np.ones(len(D3))
aux7 = agregacionD4*np.ones(len(D4))
aux1 = agregacionC1*np.ones(len(C1))
aux2 = agregacionC2*np.ones(len(C2))
aux3 = agregacionC3*np.ones(len(C3))
aux4 = agregacionD1*np.ones(len(D1))
aux5 = agregacionD2*np.ones(len(D2))
aux6 = agregacionD3*np.ones(len(D3))
C1_r = np.minimum(C1,aux1)
C2_r = np.minimum(C2,aux2)
C3_r = np.minimum(C3,aux3)
D1_r = np.minimum(D1,aux4)
D2_r = np.minimum(D2,aux5)
D3_r = np.minimum(D3,aux6)
D4_r = np.minimum(D4,aux7)
z_d = np.zeros((1,2))
nume = 0
deno = 0
CT1 = np.maximum(C1_r,C2_r,C3_r)
CT2 = np.maximum(np.maximum(D1_r,D2_r,D3_r),D4_r)
for i in range(len(z1)):
nume += CT1[i]*(i/10+15)
deno += CT1[i]
z_d[0,0] = nume/deno
nume = 0
deno = 0
for i in range(len(z2)):
nume += CT2[i]*i/100
deno += CT2[i]
z_d[0,1] = nume/deno
return z_d
for i in range(len(x)):
for j in range(len(y)):
CZ1[i,j] = difusificacion(i,j)[0][0]
CZ2[i,j] = difusificacion(i,j)[0][1]
fig, ax = plt.subplots(subplot_kw={"projection": "3d"})
ax.set_xlabel('Y')
ax.set_ylabel('X')
ax.set_zlabel('Z')
surf = ax.plot_surface(X, Y, CZ1, cmap=cm.coolwarm,
linewidth=0, antialiased=False)
fig2, ax2 = plt.subplots(subplot_kw={"projection": "3d"})
ax2.set_xlabel('Y')
ax2.set_ylabel('X')
ax2.set_zlabel('Z')
surf = ax2.plot_surface(X, Y, CZ2, cmap=cm.coolwarm,
linewidth=0, antialiased=False)
```

Referencias

[1] Hojas técnicas Velocidad del aire – S&P. (2015, 9 de junio). Ventilation Systems – S&P. https://www.solerpalau.com/es-es/hojas-tecnicas-velocidad-del-aire/

[2] B. Granados Rojas P. N. Cortez Herrera Y. E. González Navarro. Cómo Diseñar un Sistema de Inferencia Difusa de Tipo Mamdani. 1st. La Laguna Ticomán, CDMX, MX: Boletín UPIITA, 2019.