UNIDAD PROFESIONAL INTERDISCIPLINARIA EN INGENERÍA Y TECNOLOGÍAS AVANZADAS

Neourofuzzy systems

"Practice 6"

STUDENT: ZARAZUA AGUILAR LUIS FERNANDO

GROUP: 2MM13

TEACHER: BLANCA TOVAR CORONA

Introduction

Mamdani Fuzzy Inference System
A system of inference is one that allows us to give certain conclusions from a set of rules if-then evaluated with the input values in the system that we are interested in controlling, one of these inference systems is the Mamdani which was introduced by Mamdani and Assilian in 1975, this system is characterized by using fuzzy sets as a consequence of the inference rules. [1]

The steps that follow this method are:

- Evaluate input membership functions with input values.
- Perform the max-min composition for all rules of inference.
- Perform the max-min composition for each output membership function with the results obtained in the previous step.
- Obtain the defuzzified value by doing the centroid of the result in the previous step.

Humidity and Temperature Control in a Greenhouse

In the next problem it is necessary to control the speed that is in a direct current motor using 2 variables that are temperature and humidity, for this a controller will be used to regulate the engine speed correctly and sensors of temperature and humidity, on which the correct speed of the engine will be known.

To implement the control system a DHT11 humidity sensor can be used, this sensor is powered at 5V and by the OneWire protocol communicates with the microcontroller, this sensor has a small size and by means of a resistance NTC measures the humidity.

For the measurement of the temperature the sensor LM35 can be used, this sensor gives us an analog value from 0 to 1.5 Volts where the temperature is 100 times the voltage, this sensor has an accuracy of 1 ° C.

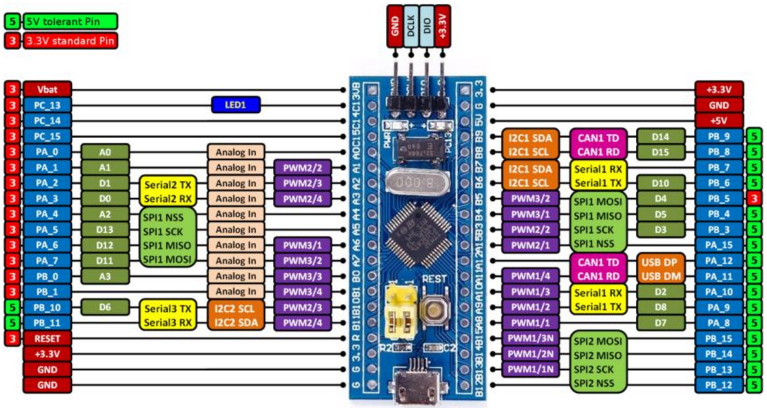
As an actuator, a 12 volt DC motor with 300rpm and a pair of 1.3Kg cm can be used, it has consumption of less than 800mA. [2]

Objective

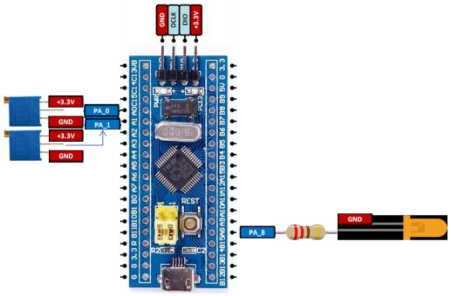
Use the Mamdani method to control the speed that we have in the fan that regulates the temperature and humidity in a garden of radishes.

Development

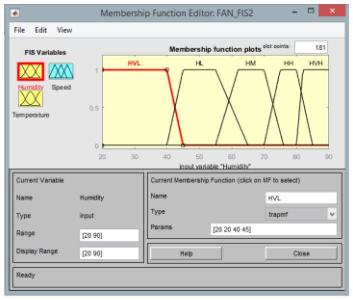
To develop the system was used the arduino IDE simulating the temperature and humidity inputs using 2 potentiometers that are able to manipulate both variables, in the solution of this problem the microcontroller stm32c108t was programmed, this microcontroller gives us a more exact calculation in the time to perform the comparisons and the defuzzification, giving us a result more accurate and fast than a typical 8 bits microcontroller of arduino.



The connection diagram used was the next:



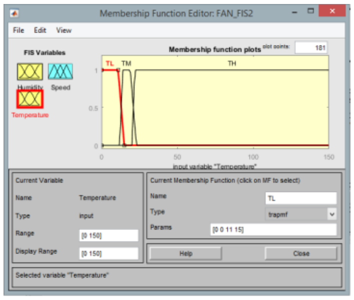
Knowing the temperatures at which the vegetable garden can maintain the vegetables and their average humidity, the following membership functions were proposed for both inputs and output.



Current Variable
Name Humidity
Type input
Range [20 90]
Display Range [20 90]

Current Membership Function (click on MF to select)
Name HVL
Type trapmf
Params [20 20 40 45]

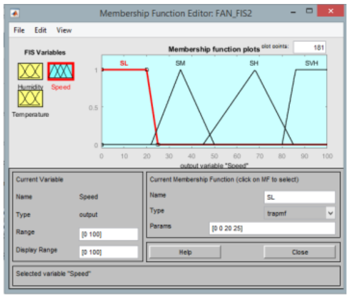
Ready



Current Variable
Name Temperature
Type input
Range [0 150]
Display Range [0 150]

Current Membership Function (click on MF to select)
Name TL
Type trapmf
Params [0 0 11 15]

Selected variable "Temperature"



Current Variable
Name Speed
Type output
Range [0 100]
Display Range [0 100]

Current Membership Function (click on MF to select)
Name SL
Type trapmf
Params [0 0 20 25]

Selected variable "Speed"

In order to obtain the desired speed result, the algorithm is programmed to know the values of the output membership functions which are only calculated when the program is started to do it faster, then the humidity and temperature membership functions are evaluated in the value reading of the sensors, already obtained these values a max min composition is made the for each one of the rules as described in the table.

Humidity Temp	Very Low	Low	Medium	High	Very High
Low	SM	SM	SL	SL	SM
Medium	SH	SM	SL	SM	SH
High	SVH	SVH	SVH	SH	SH

In order to get that composition, the minimum between each of the combinations with the entries of the inference table is calculated, saving each in the vector corresponding output function, then the maximum is calculated for each of these vectors in this case are 4 maximum values.

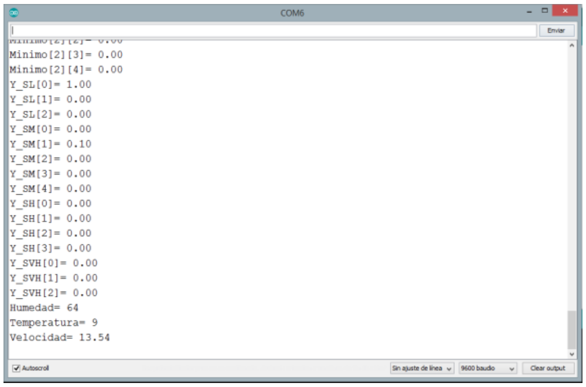
In order to obtain the final cutline, each maximum value is compared with its corresponding membership function, obtaining the minimum between these 2, having a cutline for each membership function that we have in the output, finally obtaining the maximum between all the cutlines and the centroid of this cutline corresponds to the output or defuzzificated value.

With the implementation of this algorithm we can know the output value of the motor, which is also shown using the serial interface that owns the Arduino ide and with the commands Serial.print (variable), Serial.println ("text"), with which we can print text by the serial interface.

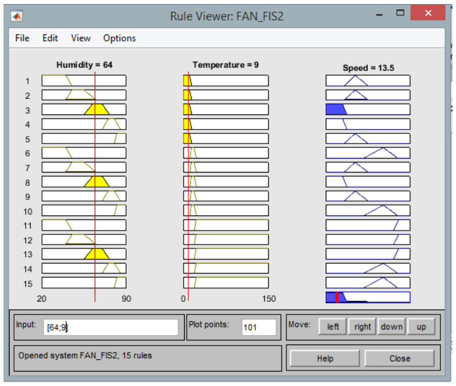
Results

In order to corroborate the output, we compared the value obtained with the one given by the fuzzy system editor in matlab, having very similar results. The output value was represented by a pwm in a led, however this pwm can be connected to an H bridge (to separate power and control stages) with which we can control the average voltage of the dc motor in the fan and thus obtain the desired speed.

For different combinations, these values were obtained:



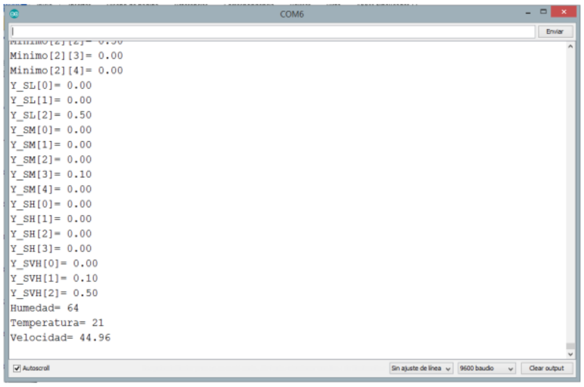
```
Minimo[2][2]= 0.00
Minimo[2][3]= 0.00
Minimo[2][4]= 0.00
Y_SL[0]= 1.00
Y_SL[1]= 0.00
Y_SL[2]= 0.00
Y_SM[0]= 0.00
Y_SM[1]= 0.10
Y_SM[2]= 0.00
Y_SM[3]= 0.00
Y_SM[4]= 0.00
Y_SH[0]= 0.00
Y_SH[1]= 0.00
Y_SH[2]= 0.00
Y_SH[3]= 0.00
Y_SVH[0]= 0.00
Y_SVH[1]= 0.00
Y_SVH[2]= 0.00
Humedad= 64
Temperatura= 9
Velocidad= 13.54
```



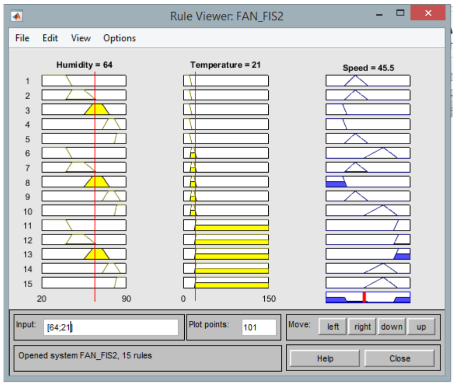
Input: [64;9]

Plot points: 101

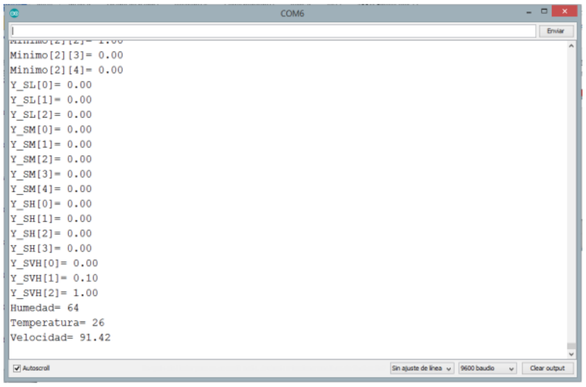
Opened system FAN_FIS2, 15 rules



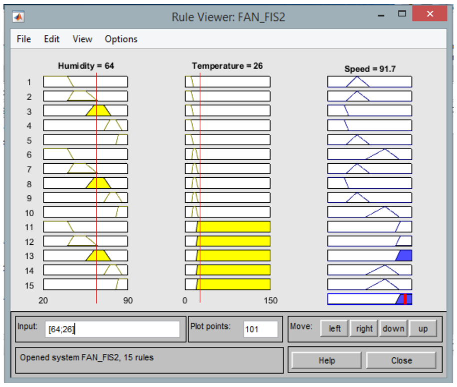
```
Minimo[2][2]= 0.50
Minimo[2][3]= 0.00
Minimo[2][4]= 0.00
Y_SL[0]= 0.00
Y_SL[1]= 0.00
Y_SL[2]= 0.50
Y_SM[0]= 0.00
Y_SM[1]= 0.00
Y_SM[2]= 0.00
Y_SM[3]= 0.10
Y_SM[4]= 0.00
Y_SH[0]= 0.00
Y_SH[1]= 0.00
Y_SH[2]= 0.00
Y_SH[3]= 0.00
Y_SVH[0]= 0.00
Y_SVH[1]= 0.10
Y_SVH[2]= 0.50
Humedad= 64
Temperatura= 21
Velocidad= 44.96
```



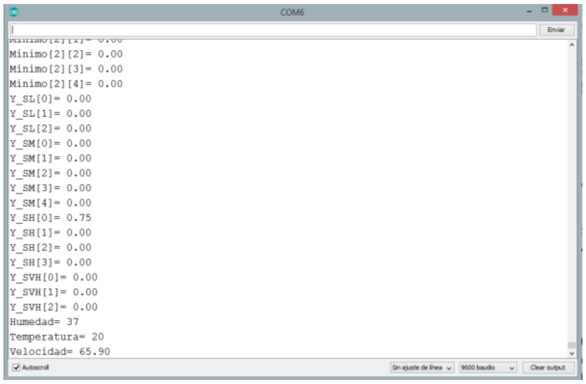
Input: [64;21]
Plot points: 101
Opened system FAN_FIS2, 15 rules



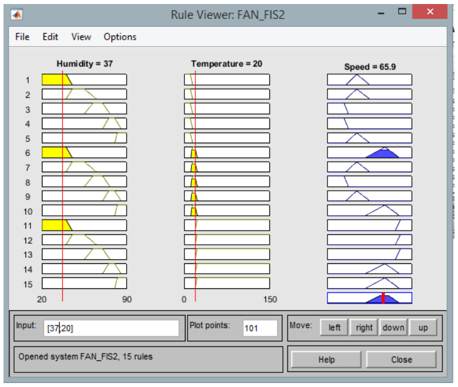
```
Minimo[2][2]= 1.00
Minimo[2][3]= 0.00
Minimo[2][4]= 0.00
Y_SL[0]= 0.00
Y_SL[1]= 0.00
Y_SL[2]= 0.00
Y_SM[0]= 0.00
Y_SM[1]= 0.00
Y_SM[2]= 0.00
Y_SM[3]= 0.00
Y_SM[4]= 0.00
Y_SH[0]= 0.00
Y_SH[1]= 0.00
Y_SH[2]= 0.00
Y_SH[3]= 0.00
Y_SVH[0]= 0.00
Y_SVH[1]= 0.10
Y_SVH[2]= 1.00
Humedad= 64
Temperatura= 26
Velocidad= 91.42
```



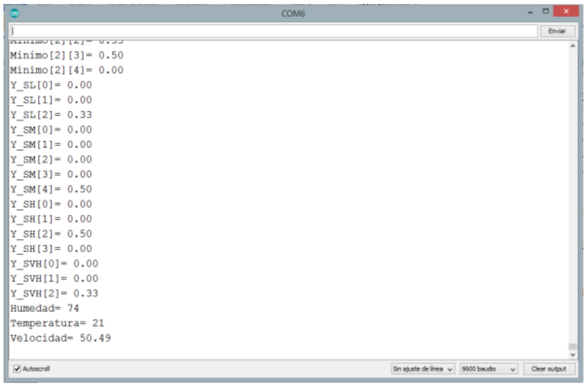
Input: [64;26]
Plot points: 101
Opened system FAN_FIS2, 15 rules



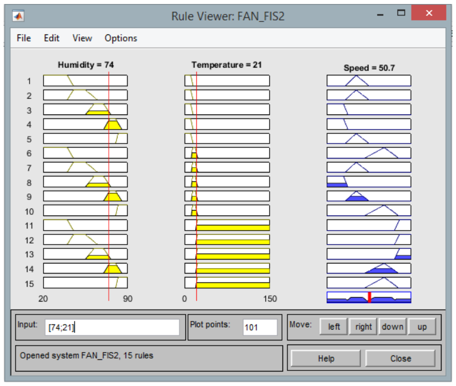
```
Minimo[2][1]= 0.00
Minimo[2][2]= 0.00
Minimo[2][3]= 0.00
Minimo[2][4]= 0.00
Y_SL[0]= 0.00
Y_SL[1]= 0.00
Y_SL[2]= 0.00
Y_SM[0]= 0.00
Y_SM[1]= 0.00
Y_SM[2]= 0.00
Y_SM[3]= 0.00
Y_SM[4]= 0.00
Y_SH[0]= 0.75
Y_SH[1]= 0.00
Y_SH[2]= 0.00
Y_SH[3]= 0.00
Y_SVH[0]= 0.00
Y_SVH[1]= 0.00
Y_SVH[2]= 0.00
Humedad= 37
Temperatura= 20
Velocidad= 65.90
```



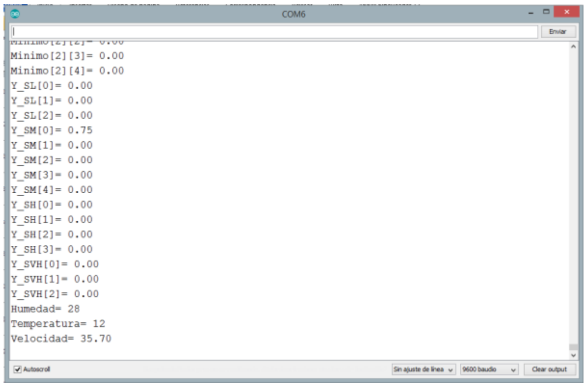
Input: [37;20]
Plot points: 101
Opened system FAN_FIS2, 15 rules



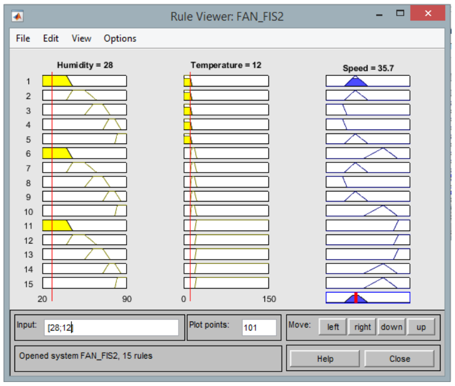
```
Minimo[2][2]= 0.33
Minimo[2][3]= 0.50
Minimo[2][4]= 0.00
Y_SL[0]= 0.00
Y_SL[1]= 0.00
Y_SL[2]= 0.33
Y_SM[0]= 0.00
Y_SM[1]= 0.00
Y_SM[2]= 0.00
Y_SM[3]= 0.00
Y_SM[4]= 0.50
Y_SH[0]= 0.00
Y_SH[1]= 0.00
Y_SH[2]= 0.50
Y_SH[3]= 0.00
Y_SVH[0]= 0.00
Y_SVH[1]= 0.00
Y_SVH[2]= 0.33
Humedad= 74
Temperatura= 21
Velocidad= 50.49
```



Input: [74;21]
Plot points: 101
Opened system FAN_FIS2, 15 rules



```
Minimo[2][2]= 0.00
Minimo[2][3]= 0.00
Minimo[2][4]= 0.00
Y_SL[0]= 0.00
Y_SL[1]= 0.00
Y_SL[2]= 0.00
Y_SM[0]= 0.75
Y_SM[1]= 0.00
Y_SM[2]= 0.00
Y_SM[3]= 0.00
Y_SM[4]= 0.00
Y_SH[0]= 0.00
Y_SH[1]= 0.00
Y_SH[2]= 0.00
Y_SH[3]= 0.00
Y_SVH[0]= 0.00
Y_SVH[1]= 0.00
Y_SVH[2]= 0.00
Humedad= 28
Temperatura= 12
Velocidad= 35.70
```



Input: [28;12]

Plot points: 101

Opened system FAN_FIS2, 15 rules

Conclusions

Using an inference system such as the Mamdani we can obtain the level of membership that has in the output with the rules proposed giving us a more accurate and simple result than using a more complex model of the system.

Depending on the processing capabilities that we implement in the circuit we will achieve more accurate results when using this method, also depending on the language it will simplify the realization of the program.

In order to obtain an optimal result you must choose well where the membership functions begin to grow to achieve a sufficiently high or low result.

References

[1] MAMDANI, E. H. y ASSILIAN, S. An experiment in linguistic synthesis with a fuzzy logic controller. *International Journal of Man-Machine Studies.* 1975, vol. 7, núm. 1, pp. 1-13.

[2] E. Barrera Martin, R. V. Herrero Niño and A. R. Meraz García, "Tesis "Invernadero Inteligente"," vol. 1, p. 103, 2014.

Annex

Code in Arduino

```
const int Sensor_Humedad=PA0;
const int Sensor_Temperatura=PA1;
const int Pin_PWM=PA8;
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin PB1 as an output.
  pinMode(PC13, OUTPUT);
  pinMode(Sensor_Humedad, INPUT_ANALOG);
  pinMode(Sensor_Temperatura, INPUT_ANALOG);
  pinMode(Pin_PWM, PWM);
  Serial.begin(9600);
  Serial.println("Comunicacion comenzada");
}
```

```
// the loop function runs over and over again forever
void loop() {
  int opcion,Humedad=43,Temperatura=50;
    //Speed   Triangle
    //  SL=Triangular(0 0 20 25);
    //  SM=Triangular(22 35 50);
    //  SH=Triangular(45 68 85);
    //  SVH=Trapezoidal(80 86  100 100);

  int i,j;
  float x;
    float Vector_X[201],F_SL[201],F_SM[201],F_SH[201],F_SVH[201];
    float F_SL2[201],F_SM2[201],F_SH2[201],F_SVH2[201],Cutline_R[201];
    for (i=0;i<201;i++){//Creación del vector del 0 al 100 para Cutlines.
       x=i/2;
       Vector_X[i]=x;
       F_SL[i]=Trapezoidal(0,0,20,25,x);
       F_SM[i]=Triangular(22,35,50,x);
      }
    for (i=0;i<201;i++){//Creación del vector del 0 al 100 para Cutlines.
       x=i/2;
       F_SH[i]=Triangular(45,68,85,x);
       F_SVH[i]=Trapezoidal(80,86,100,100,x);
      }
  while(1){
  if (Serial.available()>0){
      opcion=Serial.parseInt();  // recibe un byte de la memoria
intermedia serie
      if (Serial.read()==',')
      {
       if (opcion==1){//Modifica Humedad
          Humedad=Serial.parseInt();
          if (Serial.read()=='r'){
          Serial.print("Datos Recibidos Humedad=");
          Serial.println(Humedad);}
       }
       else{ //Modifica Temperatura
          Temperatura=Serial.parseInt();
          if (Serial.read()=='r'){
          Serial.print("Datos Recibidos Temperatura= ");
          Serial.println(Temperatura);}
       }
      }
  }
  //Read of sensors
  Humedad = map(analogRead(Sensor_Humedad), 0, 4095, 20, 90);
  Temperatura=map(analogRead(Sensor_Temperatura), 0, 4095, 0, 100);
  //Humidity Trapezoid
  float y1_HVL=Trapezoidal(20,20,40,45,Humedad);
  float y1_HL=Trapezoidal(40,45,55,65,Humedad);
  float y1_HM=Trapezoidal(55,62,70,76,Humedad);
  float y1_HH=Trapezoidal(70,75,80,85,Humedad);
  float y1_HVH=Trapezoidal(80,82,90,90,Humedad);
  Serial.print("y1_HVL= ");
  Serial.println(y1_HVL);
  Serial.print("y1_HL= ");
  Serial.println(y1_HL);
  Serial.print("y1_HM= ");
  Serial.println(y1_HM);
  Serial.print("y1_HH= ");
  Serial.println(y1_HH);
  Serial.print("y1_HVH= ");
  Serial.println(y1_HVH);
  //Temperature Trapezoid
  float y2_TL=Trapezoidal(0,0,11,15,Temperatura);
  float y2_TM=Trapezoidal(12,14,19,23,Temperatura);
  float y2_TH=Trapezoidal(20,22,150,150,Temperatura);
  Serial.print("y2_TL= ");
  Serial.println(y2_TL);
  Serial.print("y2_TM= ");
  Serial.println(y2_TM);
  Serial.print("y2_TH= ");
  Serial.println(y2_TH);
  float y1[5]={y1_HVL,y1_HL,y1_HM,y1_HH,y1_HVH};
  float y2[3]={y2_TL,y2_TM,y2_TH};


  float Y_min[3][5],var;
  for (i=0;i<3;i++){
     for (j=0;j<5;j++){
     delay(1);
     //Y_min[j,i]
     Serial.print("Minimo[");
     Serial.print(i);
     Serial.print("][");
     Serial.print(j);
     Serial.print("]= ");
     Y_min[i][j]=minimo(y1[j],y2[i]);
     Serial.println(Y_min[i][j]);
    }
  }

  float Y_SL[3]={Y_min[0][2],Y_min[0][3],Y_min[1][2]};
  float
Y_SM[5]={Y_min[0][0],Y_min[0][1],Y_min[0][4],Y_min[1][1],Y_min[1][3]};
  float Y_SH[4]={Y_min[1][0],Y_min[1][4],Y_min[2][3],Y_min[2][4]};
  float Y_SVH[3]={Y_min[2][0],Y_min[2][1],Y_min[2][2]};

  Serial.print("Y_SL[0]= ");
  Serial.println(Y_SL[0]);
  Serial.print("Y_SL[1]= ");
  Serial.println(Y_SL[1]);
  Serial.print("Y_SL[2]= ");
  Serial.println(Y_SL[2]);
  Serial.print("Y_SM[0]= ");
  Serial.println(Y_SM[0]);
  Serial.print("Y_SM[1]= ");
  Serial.println(Y_SM[1]);
  Serial.print("Y_SM[2]= ");
  Serial.println(Y_SM[2]);
  Serial.print("Y_SM[3]= ");
  Serial.println(Y_SM[3]);
  Serial.print("Y_SM[4]= ");
  Serial.println(Y_SM[4]);
  Serial.print("Y_SH[0]= ");
  Serial.println(Y_SH[0]);
  Serial.print("Y_SH[1]= ");
  Serial.println(Y_SH[1]);
  Serial.print("Y_SH[2]= ");
  Serial.println(Y_SH[2]);
  Serial.print("Y_SH[3]= ");
  Serial.println(Y_SH[3]);
  Serial.print("Y_SVH[0]= ");
  Serial.println(Y_SVH[0]);
  Serial.print("Y_SVH[1]= ");
  Serial.println(Y_SVH[1]);
  Serial.print("Y_SVH[2]= ");
  Serial.println(Y_SVH[2]);
  float Y_SL_max=maximo_Arreglo(Y_SL,3);
  float Y_SM_max=maximo_Arreglo(Y_SM,5);
  float Y_SH_max=maximo_Arreglo(Y_SH,4);
  float Y_SVH_max=maximo_Arreglo(Y_SVH,3);

  copiar_Arreglo(F_SL2,F_SL,201);
  copiar_Arreglo(F_SM2,F_SM,201);
  copiar_Arreglo(F_SH2,F_SH,201);
  copiar_Arreglo(F_SVH2,F_SVH,201);
  minimo_Arreglo(F_SL2,Y_SL_max,201);
  minimo_Arreglo(F_SM2,Y_SM_max,201);
  minimo_Arreglo(F_SH2,Y_SH_max,201);
  minimo_Arreglo(F_SVH2,Y_SVH_max,201);
  maximo_Arreglos(Cutline_R,F_SL2,F_SM2,F_SH2,F_SVH2,201);//Guarda el
maximo en Cutline_R.
//float Velocidad=Defuzz(Vector_X,Cutline_R,201);
  float Velocidad_2=Defuzz_2(Vector_X,Cutline_R,201);
//  Serial.print("Vector_X= ");
//  Serial.println(Vector_X[70]);
//  Serial.print("F_SL= ");
//  Serial.println(F_SL2[70]);
//  Serial.print("F_SM= ");
//  Serial.println(F_SM2[70]);
//  Serial.print("F_SH= ");
//  Serial.println(F_SH2[70]);
//  Serial.print("F_SVH= ");
//  Serial.println(F_SVH2[70]);
  Serial.print("Humedad= ");
  Serial.println(Humedad);
  Serial.print("Temperatura= ");
  Serial.println(Temperatura);
  Serial.print("Velocidad= ");
//  Serial.println(Velocidad);
//  Serial.print("Velocidad 2= ");
  Serial.println(Velocidad_2);
  pwmWrite(Pin_PWM, map(Velocidad_2, 0, 100, 0, 255));
  delay(3000);
 }
}
void copiar_Arreglo(float A[],float B[],int l){
  int i;
  for(i=0;i<l;i++){
    A[i]=B[i];
  }
}
float minimo(float a,float b){
  if (a<b)
  return a;
  else
  return b;
}

float maximo(float a,float b){
  if (a>b)
  return a;
  else
  return b;
}
void minimo_Arreglo(float A[],float b,int l){
  int i;
  for(i=0;i<l;i++){
   if (b<A[i])
    A[i]=b;
  }
}
void maximo_Arreglos(float R[],float A[],float B[],float C[],float
D[],int l){
  int i;
  for(i=0;i<l;i++){
  float Y[4]={A[i],B[i],C[i],D[i]};
   R[i]=maximo_Arreglo(Y,4);
  }
}
float maximo_Arreglo(float A[],int l){
  int i,index;
  float y;
  for(i=0;i<l;i++){
     if(A[i]>y){
        y=A[i];
        index=i;
      }
   }
   return y;
}
float Defuzz(float X[],float A[],int l){
  int i;
  float dt=X[1]-X[0];
  float sum1=0;
  float sum2=0;
  for(i=0;i<l;i++){
      sum1=sum1+A[i]*X[i]*dt;
      sum2=sum2+A[i]*dt;
     }
  return sum1/sum2;
}
float Defuzz_2(float X[],float A[],int l){
  int i;
  float sum1=0;
  float sum2=0;
  for(i=0;i<l;i++){
      sum1=sum1+A[i]*X[i];
      sum2=sum2+A[i];
     }
  return sum1/sum2;
}
float Trapezoidal(float a,float b,float c,float d,float x){
  float y;
  if (x<a)
     y=0;
   else if (a<=x && x<b)
     y=(x-a)/(b-a);
   else if (b<=x && x<=c)
     y=1;
   else if (c<=x && x<d)
     y=(d-x)/(d-c);
   else
     y=0;
   if (c==d && x==c)
     y=1;
//   Serial.print("a= ");
//  Serial.print(a);
//  Serial.print(",b= ");
//  Serial.print(b);
//  Serial.print(",c= ");
//  Serial.print(c);
//  Serial.print(",d= ");
//  Serial.print(c);
//  Serial.print(",x= ");
//  Serial.print(x);
//  Serial.print(",y= ");
//  Serial.println(y);
   return y;
}

float Triangular(float a,float b,float c,float x){
   float y;
   if (x<a)
     y=0;
   else if (a<=x && x<b)
     y=(x-a)/(b-a);
   else if (b<=x && x<c)
     y=(c-x)/(c-b);
   else
     y=0;
   return y;
}
```