Components & Experiments Toward a Pollen Neutralization System

Ziqi Guo, Mel Siegel

CMU-RI-TR-18-10

April 2018

Robotics Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

Abstract

Pollen Neutralization System aims to maintain desired indoor temperature, humidity and pollen density, based on the hypothesis that certain temperature and humidity help reduce pollen density.

The system displays real-time values of temperature and humidity sensor (DHT22), dust sensor (DSM501A), and predicts the trends of temperature and humidity. Fuzzy control strategy was used to eliminate overshoot of temperature control. Relay ports were reserved to control Heating Ventilation and Air Conditioning (HVAC), Humidifier, and High Efficiency Particle Arrestance Filter (HEPA). Moreover, a previously published experiment by Barnes et al. was referred to testify the relationship between temperature, humidity and pollen density. Another self-designed experiment was compared and analyzed, where pollen density is surrogated by dust ratio. Those two experiments demonstrate that temperature and humidity have minimal influence on pollen density and dust ratio.

Acknowledgement

I would like to show my gratitude to Dr. Mel Siegel and Rachel Burcin who always provide great help and support. I am honored to have them as my mentors. Special thanks to Dr. Mel Siegel for his detailed instructions on Pollen Neutralization System and valuable comments on this Technical Report.

I also thank for Suryansh and Isaac who provided insight and experience of how to detect practical problems which need to be solved and propose research topics. Pollen Neutralization System was proposed during discussion with them.

Contents

1 Introduction **1**

2 Hardware components **1**
2.1 DHT22 2
2.2 DSM501A 4

3 Software **6**
3.1 Trend Prediction 6
3.2 Fuzzy Control 6

4 Experiment **8**
4.1 Barnes Experiment 8
4.2 Self-Designed Experiment 8
4.3 Analysis 12

5 Conclusion **13**

6 Appendix **14**
6.1 DHT22 Datasheet 14
6.2 DSM501A Datasheet 14

1 Introduction

Nowadays, roughly 7.8% of people 18 and over in the U.S. have hay fever.[1] Some of people get hay fever because of the exposure to pollen for a period of time, usually 3-5 years. Hay fever is a type of inflammation in the nose which occurs when the immune system overreacts to allergens in the air. Signs and symptoms include a runny or stuffy nose, sneezing, red, itchy, and watery eyes, and swelling around the eyes.[2] The fluid from the nose is usually clear. Symptoms onset is often within minutes following exposure and they can affect sleep, the ability to work, and the ability to concentrate at school.[3]

It is also inevitable that indoor room exists pollen. People may gradually get hay fever and suffer from it, even always stay indoor. Therefore a Pollen Neutralization System is needed. The system should be able to maintain desired environment to prevent hay fever.

For this project, Pollen Neutralization System detects temperature, humidity and pollen density. It also predicts trends of temperature and humidity. Fuzzy control strategy was utilized for temperature control. Based on the hypothesis that temperature, humidity affect pollen density, a previously published experiment[5] was referred to testify the relationship between temperature, humidity and pollen density. Another self-designed experiment was compared and analyzed.

The report is organized as the following. Section 2 shows hardware components of Pollen Neutralization System. In section 3, methods of trend prediction and fuzzy control strategy are introduced. Section 4 describes two experiments. Section 5 gives conclusion. And section 6 includes appendix documents: codes for Pollen Neutralization System, Arduino Mega documents, DHT22 and DSM501A sensor data-sheets.

2 Hardware components

The system, Figure 1, is composed by Arduino Mega with LCD screen, temperature and humidity sensor (DHT22), dust sensor (DSM501A), and relay. Detailed information of Arduino Mega, DHT22 and DSM501A are referred in Appendix.

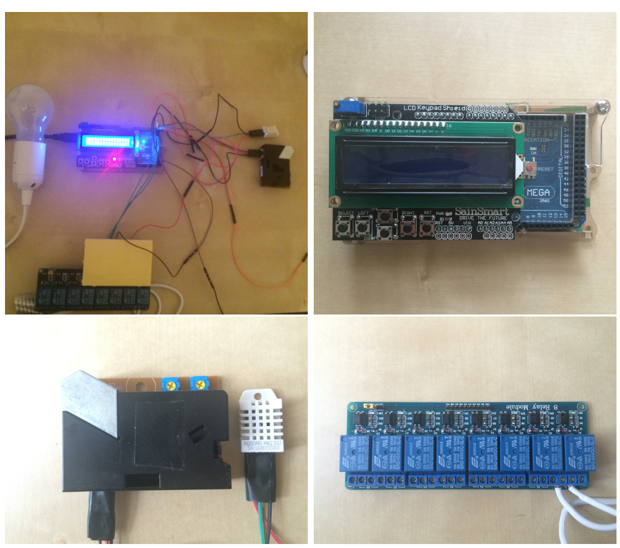

Figure 1: The sub-pictures include overview of Pollen Neutralization System (left-top), Arduino Mega with shielded LCD and protection (right-top), DSM501A and DHT22 sensors (left-bottom), relay (right-bottom).

2.1 DHT22

DHT22 pins	
1	VCC
2	DATA
3	NC
4	GND

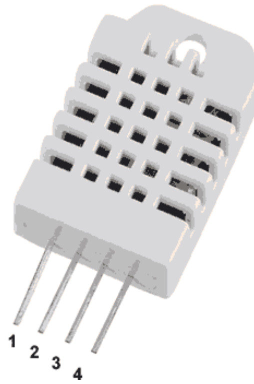

Figure 2: temperature and humidity sensor (DHT22)

AM2303/DHT22 is digital-output temperature humidity sensor. It outputs calibrated digital signal, and utilizes exclusive digital-signal-collecting-technique and humidity sensing technology, assuring its reliability and stability. DHT22 will directly output temperature (Celsius), relative humidity (Percentage) and display through LCD screen without noise.

Technical details of DHT22:

- Supply Voltage: 3-5V
- Current: 2.5mA
- Humidity: 0-100%, 2-5% accuracy
- Temperature: -40 to 80^oC, 0.5^oC accuracy

To display data through LCD, LiquidCrystal and DHT libraries need to be downloaded from Arduino Libraries. Here is the test code from Arduino Tutorials, circuit is shown as Figure 3, result is shown as Figure 4:

```
#include <LiquidCrystal.h> // includes the LiquidCrystal Library
#include <dht.h> // includes the Digital Humidity & Temperature Library for DHT22


#define dataPin 22
LiquidCrystal lcd(1, 2, 4, 5, 6, 7); // Creates an LCD object. Parameters: (rs, enable, d4, d5, d6, d7)
dht DHT;

void setup() {
  lcd.begin(16,2); // Initializes the interface to the LCD screen, and specifies the dimensions (width and height) of the display
}

void loop() {
  int readData = DHT.read22(dataPin);
  float t = DHT.temperature;
  float h = DHT.humidity;
  lcd.setCursor(0,0); // Sets the location at which subsequent text written to the LCD will be displayed
  lcd.print("Temp.: "); // Prints string "Temp." on the LCD
```

```
    lcd.print(t); // Prints the temperature value from the sensor
    lcd.print(" C");
    lcd.setCursor(0,1);
    lcd.print("Humi.:  ");
    lcd.print(h);
    lcd.print(" %");
    delay(2000);
}
```

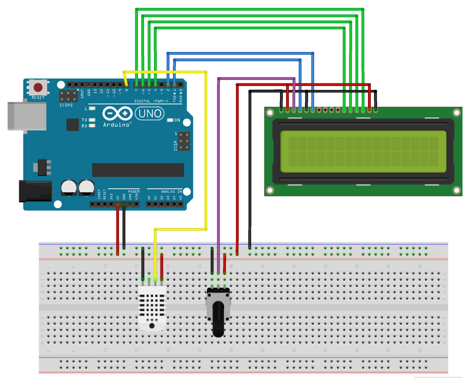

Figure 3: This is the circuit of DHT22 on Arduino UNO and proto-board. Pollen Neutralization System uses Arduino Mega instead.

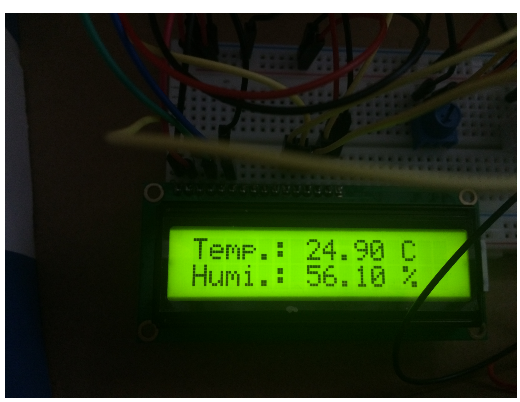

Figure 4: This is the result of DHT22 test code.

2.2 DSM501A

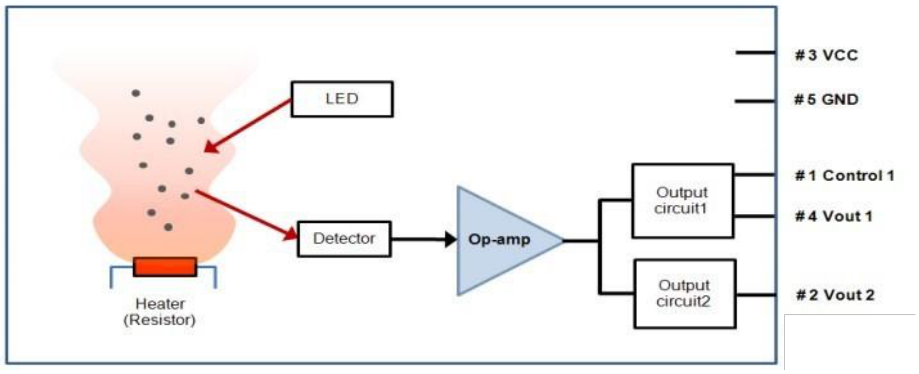

Figure 5: dust sensor (DSM501A)

Since a pollen sensor which specifically responses to pollen and acheives accurate measurement for pollen density is too expensive, we used dust sensor DSM501A instead. The dust sensor module DSM501A is a low-cost and compact-sized particle density sensor. As Figure 5, DSM501 consists of : Light Emitting Diode (LED) Lamp; Detector; Signal amplifier circuit; Output drive circuit 1; Output drive circuit 2; Heater (Resistor). The sensor displays a maximally high signal when no dust particles are present in the sampling volume of its optics, and a low signal when dust is present. The measure of dust concentration is based on the ratio of low signal time to overall sampling time of 30000 milliseconds.

Technical details of DSM501A:

- Supply Voltage: 4.5-5.5V
- Current: 2.5mA
- Detectable Particle Size: $1\mu m (munimum)$

Here is the test code from seeed WIKI:

```
#include<string.h>

int pin = 8;//DSM501A input D8
unsigned long duration;
unsigned long starttime;
unsigned long endtime;
unsigned long sampletime_ms = 30000; // overall sampling time (millisecond)
unsigned long lowpulseoccupancy = 0;
float ratio = 0; //ratio of low signal time to overall sampling time of 30000
    milliseconds
float concentration = 0;

int i=0;
void setup()
{
  Serial.begin(9600);
  pinMode(8,INPUT);
  starttime = millis();
}
void loop()
{

```

```
  duration = pulseIn(pin, LOW); //obtain duration time of one low signal from pin (microsecond)
  lowpulseoccupancy += duration;
  endtime = millis();
  if ((endtime-starttime) > sampletime_ms)
  {
    Serial.print("duration:");
    Serial.print(duration);
    Serial.print("\n");
    ratio = lowpulseoccupancy/(sampletime_ms*10.0);  // percentage 0~100% (only output integer-bits in Pollen Neutralization System)
    concentration = 1.1*pow(ratio,3)-3.8*pow(ratio,2)+520*ratio+0.62; // using spec sheet curve
    Serial.print("lowpulseoccupancy:");
    Serial.print(lowpulseoccupancy);
    Serial.print("\n");
    Serial.print("ratio:");
    Serial.print(ratio);
    Serial.print("\n");
    Serial.print("DSM501A:");
    Serial.println(concentration);
    Serial.print(";\n\n");

    lowpulseoccupancy = 0;
    starttime = millis();
  }

}
```

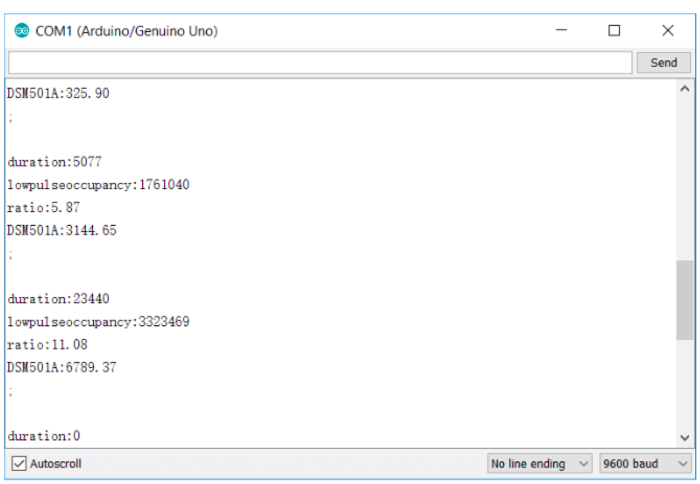

Figure 6: This is the result of DSM501A test code in Arduino UNO Serial Monitor without LCD screen. Pollen Neutralization System uses Arduino Mega with shielded LCD screen instead.

3 Software

3.1 Trend Prediction

Take temperature prediction for example. Select 6 consecutive values of temperature and compute the difference between each consecutive two. We obtain 5 gaps which would be'increase (↑)', 'decrease (↓)' or 'stable (=)'. We consider the mode of the 5 gaps as the trend prediction. Two examples are given as Table 1.

1	2	3	4	5	Trend
↑	↑	↑	↓	↓	increase

1	2	3	4	5	Trend
=	=	=	↓	=	stable

Table 1: examples of trend prediction

3.2 Fuzzy Control

Pollen Neutralization System integrates fuzzy control strategy for temperature control. Fuzzy control is a control system based on fuzzy logic mathematical system that analyzes analog input values in terms of logical variables that take on continuous values between 0 and 1.[4] The expectation was to eliminate overshooting of temperature control. Future experiments could be done to testify actual control effect.

Fuzzy Logic Toolbox in matlab was utilized to implement fuzzy control. Fuzzy control steps are as the following:

- Document the system's operational specifications and inputs and outputs.

  NL: Negative Large;

  NM: Negative Medium;

  NS: Negative Small;

  Z: Zero;

  PS: Positive Small;

  PM: Positive Medium;

  PL: Positive Large;

  Inputs: e is the error between current Temperature and desired Temperature,

  $$error = currentTemperature - desiredTemperature$$

  ec is error rate,

  $$errorRate = error_t - error_{t-1}$$

  Output: u is the control result, whose domain of discourse is [-3 -2 -1 0 1 2 3], which means [COOL_LARGE, COOL_MED, COOL_SMALL, OFF, HEAT_SMALL, HEAT_MED, HEAT_LARGE].

- Determine Membership functions, including Me, Mec, Mu, shown as Figure 7.

  Me: Membership Function of error;

  Mec: Membership Function of error rate;

  Mu: Membership Function of output;

Me	-15~-10 °C	-10~-5°C	--5~-2°C	-2~-2°C	2~5°C	5~-10°C	10~-15 °C
NL(e)	1	0.5	0.3	0	0	0	0
NS(e)	0	0.5	1	0.3	0	0	0
Z(e)	0	0	0.3	1	0.3	0	0
PS(e)	0	0	0	0	1	0.5	0.3
PL(e)	0	0	0	0	0.3	0.5	1

Mec	-10~-6°C	-6~-2°C	-2~2°C	2~-6°C	6~10°C
Negative(ec)	0.3	1	0	0	0
Zero(ec)	0	0.3	1	0.3	0
Positive(ec)	0	0	0.3	1	0.3

Mu	-3 (HEAT_ LARGE)	-2 (HEAT_ MED)	-1 (HEAT_ SMALL)	0 (OFF)	1 (COOL_ SMALL)	2 (COOL_ MED)	3 (COOL_ LARGE)
NL(u)	1	0.5	0	0	0	0	0
NM(u)	0	1	0.5	0	0	0	0
NS(u)	0	0.5	1	0.5	0	0	0
Z(u)	0	0	0.5	1	0.5	0	0
PS(u)	0	0	0	0.5	1	0.5	0
PM(u)	0	0	0	0	1	0.5	0
PL(u)	0	0	0	0	0	0.5	1

Figure 7: Membership Functions

- Document the rule set, shown as Figure 8.

  UC is the rule table for inputs, for example, if e is Positive Large and ec is Positive, then UC is Positive Large which needs cooler on.

UC	PL(e)	PS(e)	Z(e)	NS(e)	NL(e)
Positive(ec)	7(PL)	7(PL)	7(PL)	6(PM)	4(Z)
Zero(ec)	6(PM)	5(PS)	4(Z)	3(NS)	2(NM)
Negative(ec)	4(Z)	2(NM)	1(NL)	1(NL)	1(NL)

Figure 8: UC: the rule table for inputs

- Determine the defuzzification method.

$$R = Me \times Mec \times Mu$$

  We obtain fuzzy vector

$$U^{'} = e^{'} \times ec^{'} \circ R$$

  $e^{'}$ and $ec^{'}$ is the fuzzy vector of e and ec. For example, e = -15, ec = -10, then $e^{'}$ = [1 0 0 0 0 0 0], $ec^{'}$ = [1 0 0 0 0].

Finally, we use weighted mean method as defuzzification method. For example, $U^{'}$=[0 0 0 0.3 0.3 0.5 1], domain of discourse of U is [-3 -2 -1 0 1 2 3]. We obtain u=$U^{'}$U/sum($U^{'}$)=2, which means COOL_MED.

And we obtain output table for temperature, shown as Figure 9:

Output		Error						
		-15~-10 °C	-10~-5°C	--5~-2°C	-2~-2°C	2~5°C	5~10°C	10~15°C
Error rate	-10~-6°C	-3 (HEAT_LARGE)	-3 (HEAT_LARGE)	-3 (HEAT_LARGE)	-3 (HEAT_LARGE)	-1 (HEAT_SMALL)	0 (OFF)	0 (OFF)
	-6~-2°C	-2 (HEAT_MED)	-2 (HEAT_MED)	-2 (HEAT_MED)	-2 (HEAT_MED)	-1 (HEAT_SMALL)	0 (OFF)	0 (OFF)
	-2~-2°C	-1 (HEAT_SMALL)	0 (OFF)	0 (OFF)	0 (OFF)	1 (COOL_SMALL)	1 (COOL_SMALL)	1 (COOL_SMALL)
	2~6°C	0 (OFF)	0 (OFF)	1 (COOL_SMALL)	1 (COOL_SMALL)	2 (COOL_MED)	2 (COOL_MED)	2 (COOL_MED)
	6~10°C	0 (OFF)	1 (COOL_SMALL)	1 (COOL_SMALL)	2 (COOL_MED)	3 (COOL_LARGE)	3 (COOL_LARGE)	3 (COOL_LARGE)

Figure 9: Fuzzy control strategy for temperature.

4 Experiment

We suppose that proper temperature, humidity will help alleviate pollen allergy, and pollen allergy simply depends on airborne pollen concentration. Therefore, the experiment is based on two hypothesis. First, the presence of high temperatures tends to result in increased airborne pollen concentrations. Second, the coincidence of periods of high relative humidity result in lowered airborne pollen concentrations.

4.1 Barnes Experiment

Barnes designed his experiment [5] to testify weather precipitation, wind speed, temperature and relative humidity influence airborne pollen concentrations, as the following:

"During the ragweed (RW) season for the years 1997 and 1998, 10 minute pollen collections were taken at least every 4 hours using an Allergenco MK-3 spore trap. Slides were fixed, and counted microscopically at 400X. During this same period, weather parameters were monitored by an Automated Weather Systems recording station located within a few meters of the collector. The ragweed season for this region begins in mid August and ends by mid October. Temperature patterns for the period demonstrated usual daily fluctuations with highs 13 to 35 C and lows 8 to 24 C. Relative humidity readings for the period varied between 25 and 100%. Highest RW values were seen after seasonal cooling in September. Daily rainfall for the period varied between 0 and 100 mm.
All samples from the collector were examined microscopically. And meteorological data was monitored within 20 meters of the collection site using a station obtained from Automated Weather Source (Gaithersburg, MD, USA)."

Barnes concluded that under normal weather conditions, temperature and relative humidity have minimal effects on pollen density; however, unstable atmospheric conditions such as the passing of a cold front has the greatest impact of all the weather-related events on airborne pollen density. Also, airborne ragweed pollen counts are generally lowest at 6:00 AM right around dawn and highest around noon after pollen grains dry out and become airborne.

4.2 Self-Designed Experiment

- Use incense to produce smoke, as Figure 10.

  Firstly, we would like to make dust ratio above normal state and record data of temperature, humidity and dust ratio. As shown in Figure 11, after incense was ignited, dust ratio increased. To visualize data, we could download Arduino Support from MATLAB. Then plot realtime data of temperature, humidity and dust ratio in matlab. Here is the matlab code:

```
s = serial('COM3');
set(s,'BaudRate',9600);
fopen(s);

interval = 10000;
t_pass = 1;
t = 1;
x = [];
y = [];
z = [];
flag = 0;
xlabel('Count')
ylabel('Data')
while(t<interval)
    if flag == 0
    a = str2num(fgetl(s))
    x = [x,a];
    subplot(2,2,1);
    plot(x,'r-*');
    title('Temperature');
    ylabel('C');
    axis([0 inf 10 50]);
    hold on
    drawnow;
    t = t+t_pass;
    flag = 1;
    elseif flag == 1
      b = str2num(fgetl(s))
       y = [y,b];
      subplot(2,2,2);

       plot(y,'b-o');
       title('Humidity');
       ylabel('%');
       axis([0 inf 0 100]);

       drawnow;
       flag = 2;
       t = t+t_pass;
    elseif flag == 2
      c = str2num(fgetl(s))
       z = [z,c];
       subplot(2,2,3);

       plot(z,'y-+');
       title('Dust Ratio');
       ylabel('%');
      axis([0 inf 0 30]);
       t = t+t_pass;
       drawnow;
    flag = 0;
    end
```

```
end
fclose(s);
}
```

Figure 10: To increase dust ratio, incense was used to produce smoke.
Compared with other kinds of smoke, like rope and cigarette, incense has no pollution and smells good.

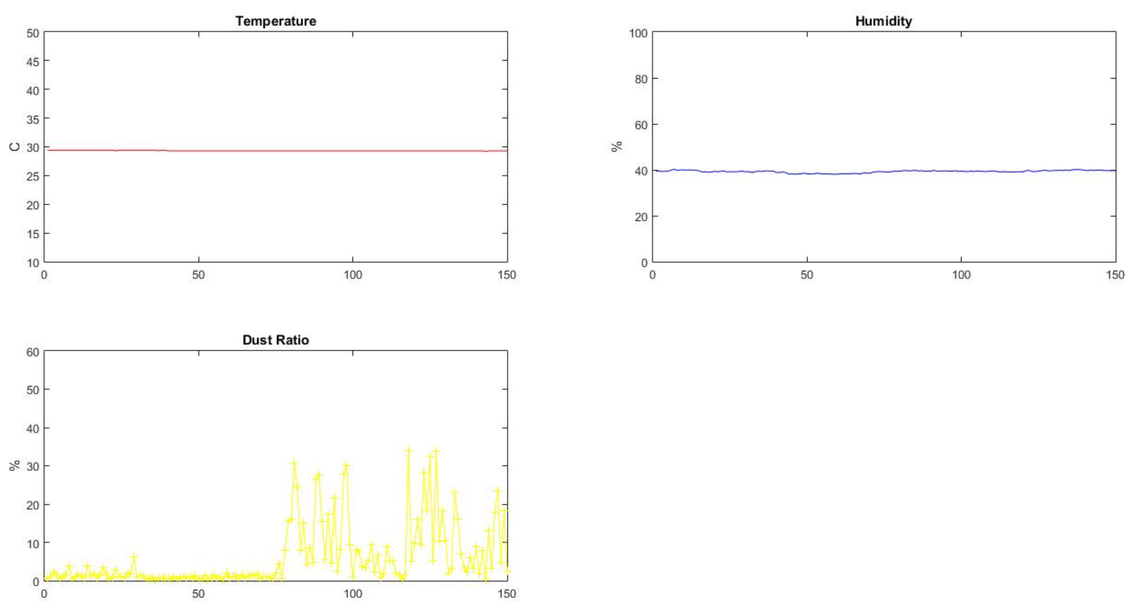

Figure 11: Output real time data of temperature, humidity and dust ratio in matlab.

- Use a 100W incandescent bulb to increase temperature. The result is shown as Figure 13.

Figure 12: One incandescent bulb (120v 100W) with extension cord was controlled by 5v relay. The bulb was used as a heater to increase temperature.

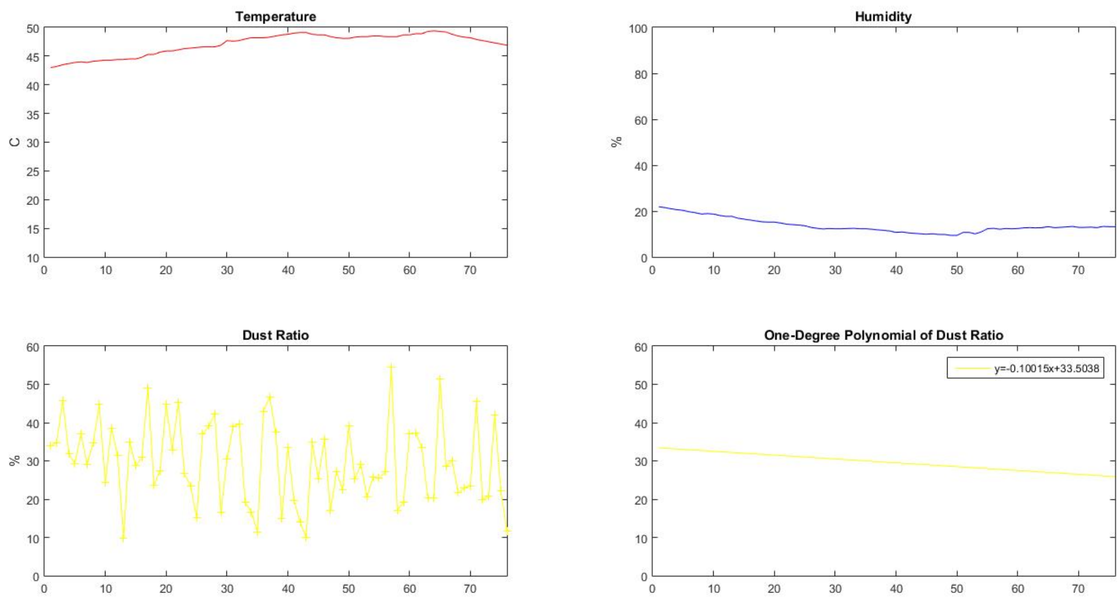

Figure 13: This is the result when temperature increased. One-degree polynomial of dust ratio (right-bottom curve) was given to observe the trend of dust ratio. The bulb was not able to only control temperature. It increased temperature, and decreased humidity simultaneously. But this does not influence experiment result. The slope was approximately zero.

- Use tissue with water and a 5v mini fan to increase humidity. The result is shown as Figure 15.

Figure 14: Tissue with water and 5v mini fan was used to decrease humidity

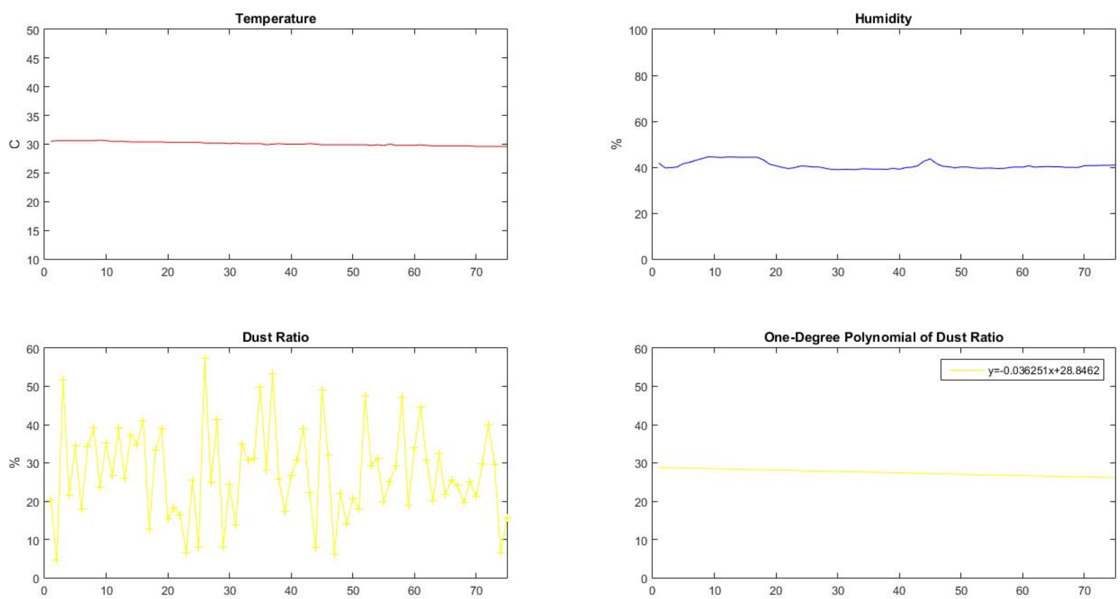

Figure 15: This is the result when humidity increased. One-degree polynomial of dust ratio (right-bottom curve) was given to observe the trend of dust ratio. The slope was effectively zero, which means dust ratio trend remained stable.

4.3 Analysis

As shown in Figure 11, the dust ratio increased with fluctuations. Since the DSM501A sensor was perpendicularly above incense, whose smoke is not stable, and the system was exposed in unsealed environment. The dust concentration is not uniformed. Though the dust ratio plot could not directly show whether it increased or decreased, polynomial could be utilized to analyze the trend of dust ratio.

In Figure 13 and 15, one-degree polynomial of dust ratio was given to observe the trend of dust ratio. When temperature increased, the slope of the one-degree polynomial line was -0.1. When humidity increased, the slope was -0.036. We suppose -0.036 ± 0.1 is effectively zero, which means the dust ratio trend remained stable. However the slope value when temperature increased was a little smaller than the slope value when humidity increased. Perhaps because the heating of bulb dispelled some smoke near the dust sensor.

Another biological factor may also affect the experiment result to testify the initial hypothesis, that temperature and humidity may influence pollen's biological viability, while smoke does not have the biological feature. As Barnes mentioned, numerous investigators (Solomon and Mathews, 1990[6]; Bianchi et al., 1959[7]; Ogden et al., 1969[8]) observed that ragweed pollen is expelled from the flower about sunrise (around 6:00 AM) and becomes airborne as the pollen dries out and the ambient wind distributes the grains. This is why pollen density usually gets lowest around dawn and reaches highest during the noon. Temperature and humidity may influence dry-out progress for pollen. However, smoke of incense does not share this feature.

5 Conclusion

Pollen Neutralization System displays real-time values of dust sensor (DSM501A), temperature and humidity sensor (DHT22) on LCD screen, and predicts the trends of temperature and humidity. Fuzzy control strategy was used for temperature control. Humidity and pollen density control could be developed in the future. Relay ports were reserved to control Heating Ventilation and Air Conditioning (HVAC), Humidifier, and High Efficiency Particle Arrestance Filter (HEPA). Furthermore, the scientific experiment shows that temperature and humidity have minimal effect on pollen density. And the self-designed experiment indicates that temperature and humidity do not influence dust ratio.

6 Appendix

6.1 DHT22 Datasheet

6.2 DSM501A Datasheet

References

[1] Summary Health Statistics for U.S. Adults: National Health Interview Survey, 2010. By Jeannine S. Schiller, M.P.H., Jacqueline W. Lucas, M.P.H., Brian W. Ward, PhD and Jennifer A. Peregory, M.P.H., Division of Health Interview Statistics.

[2] ”Environmental Allergies: Symptoms”. NIAID. April 22, 2015. Retrieved 19 June 2015.

[3] Wheatley, LM; Togias, A (29 January 2015). ”Clinical practice. Allergic rhinitis.”. The New England Journal of Medicine. 372 (5): 45663.

[4] Introduction to Fuzzy Control

[5] The effect of temperature, relative humidity and rainfall on airborne ragweed pollen concentrations. Aerobiologia, 2001, Volume 17, Number 1, Page 61 Charles Barnes, Freddy Pacheco, Julie Landuyt, Frank Hu, Jay Portnoy

[6] Solomon W. and Mathews K.: 1990, Aerobiology and inhalant aller- gens. In: Middelton Reed, Ellis Adkinson and Yunginger (eds), Allergy Principles and Practice, Volume 1 Third edition. Mosby, St Louis, pp. 312372.

[7] Bianchi D., Schwemmin D. and Wagner W.: 1959, Pollen release in common Ragweed (Ambrosia Artimesifolia). Botanical Gazzet 4, 253.

[8] Ogden E. and Raynor G.: 1960, Field evaluation of ragweed pollen samplers. Journal of Allergy 31, 307316.