Photo: Ravi Patel

Improvement of Relative Humidity Control at AppalAIR

Shawn Beekman

Department of Physics and Astronomy, Appalachian State University, Boone NC

The Importance of Atmospheric Aerosols

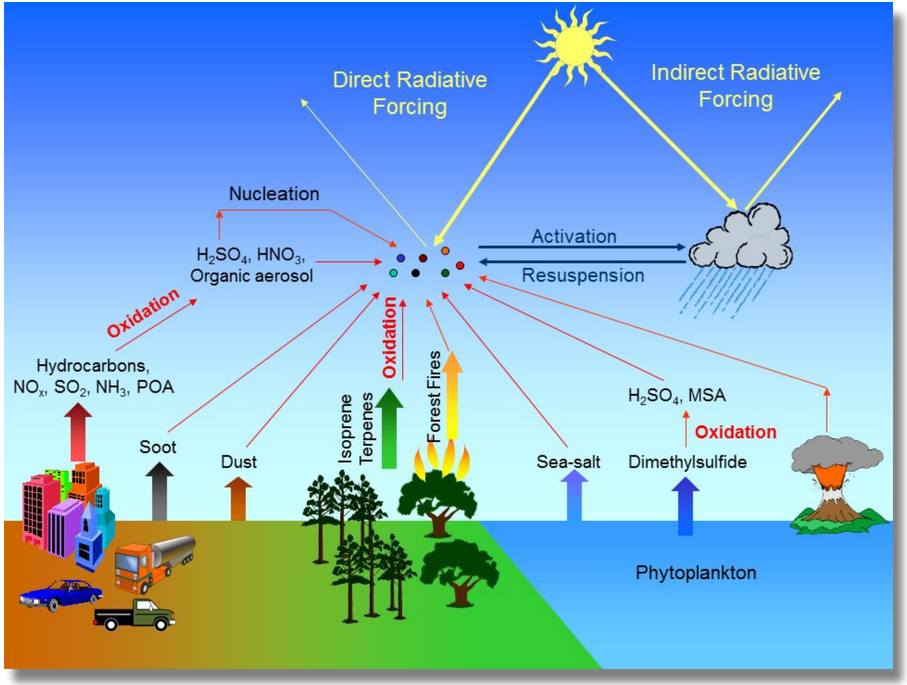


Figure 1: Image from Pacific Northwest National Laboratory [1]

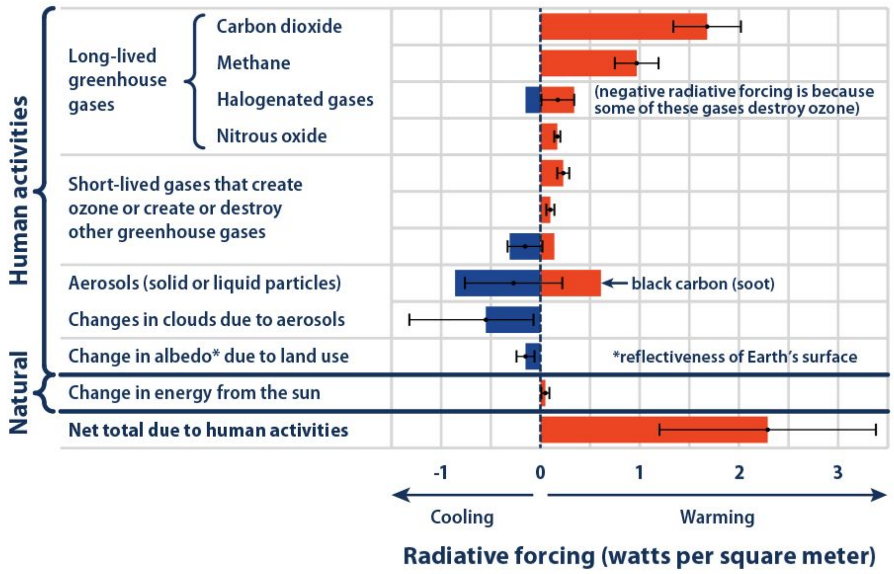
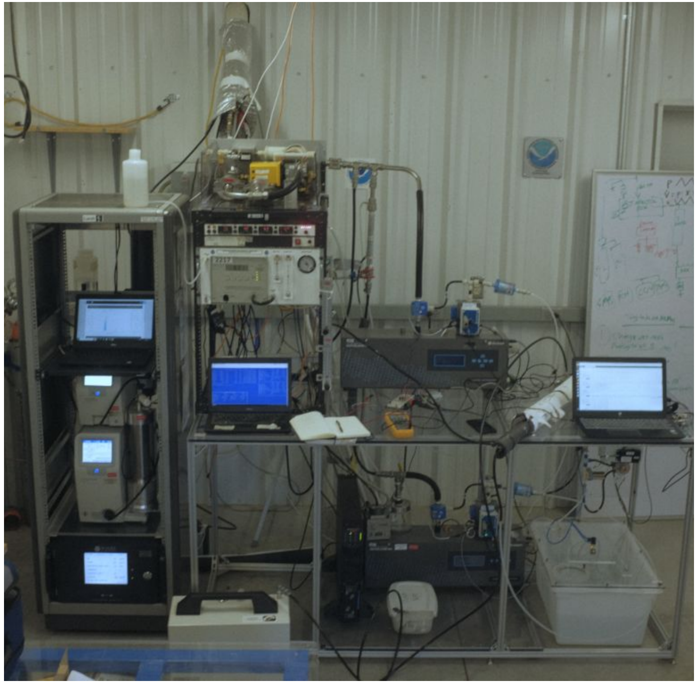


Figure 2: Climate Change Indicators: Climate Forcing | US EPA [2]

The Appalachian Atmospheric Interdisciplinary Research Facility (AppalAIR)

Aerosol Supersite representative of the Southeastern US

- Regional pollution transport
- Air quality
- Meteorology
- Climate Change



Measuring RH Dependent Light Scattering

1. Ensure global comparability (40% - 90% RH, WMO/GAW, 2003)
2. Contribute to limited deliquescent/efflorescent research.. eventually

Scattering Coefficient (σsp) is used to model direct climate forcing by aerosols

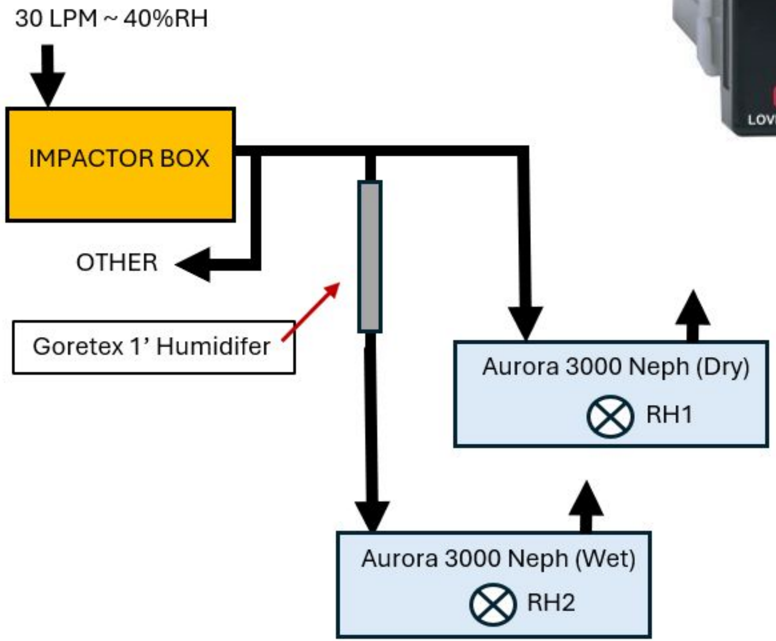
Scattering Enhancement Factor:

$$f\,(\mathrm{RH}) = \frac{\sigma_{\mathrm{sp}}(\mathrm{RH})}{\sigma_{\mathrm{sp}}(\mathrm{RH{=}dry})}$$

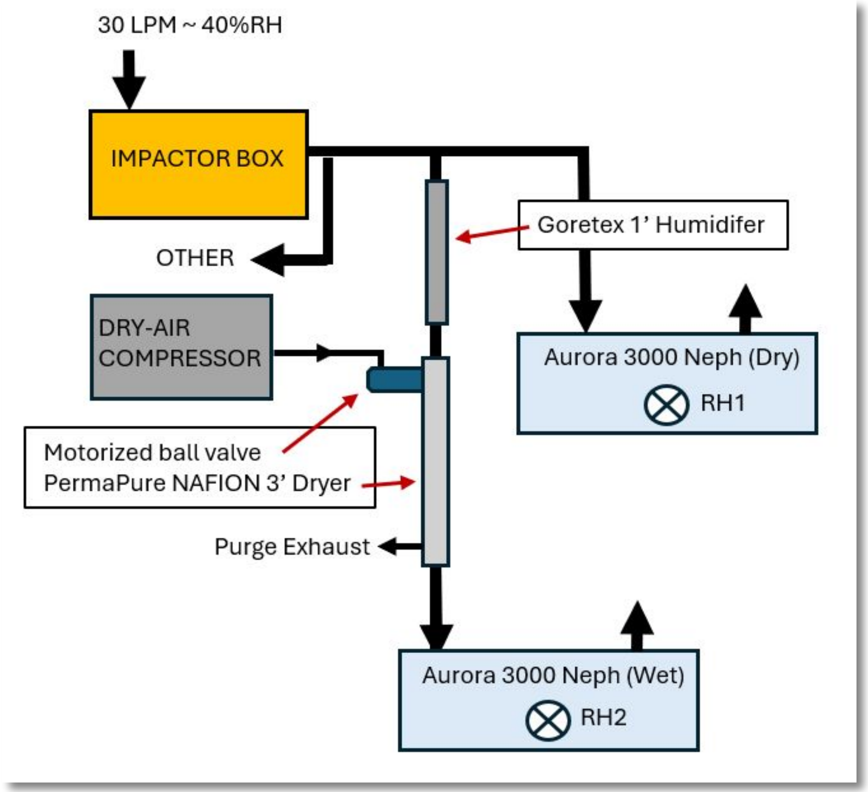
Ecotech Aurora 3000 [4]

How we measure f(RH)

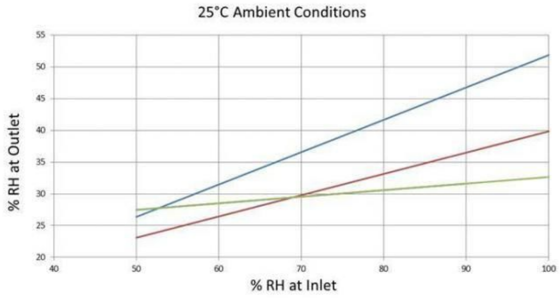
Experimental Setup:



Improvements to Measuring f(RH)



PermaPure Monotube Dryer 700



Symbol			
Inlet Flow	2LPM	5LPM	16.7LPM
Dryer Length	12 Inches 15 cm	24 Inches 60 cm	48 Inches 120 cm
Vacuum	23" Hg 779 mbar	20" Hg 677 mbar	23" Hg 779 mbar
Purge Flow	80%	80%	80%

Figure 3: MD-700 Dryer Specifications [3]

Dryer Capability

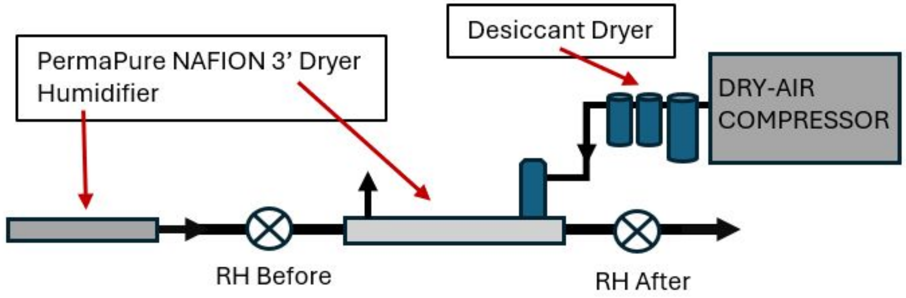
Good news:

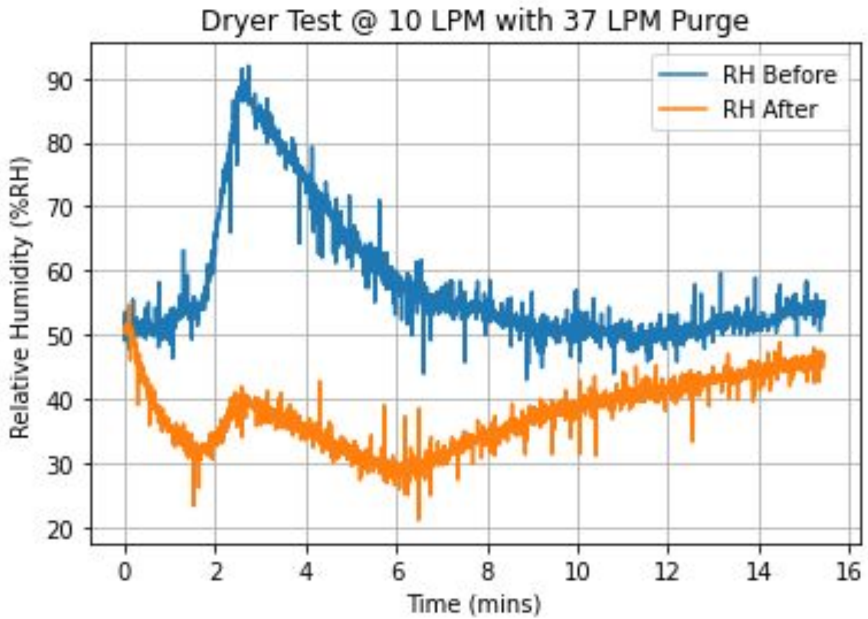
We were able to reduce %RH from 88% to 41%!

Bad news:

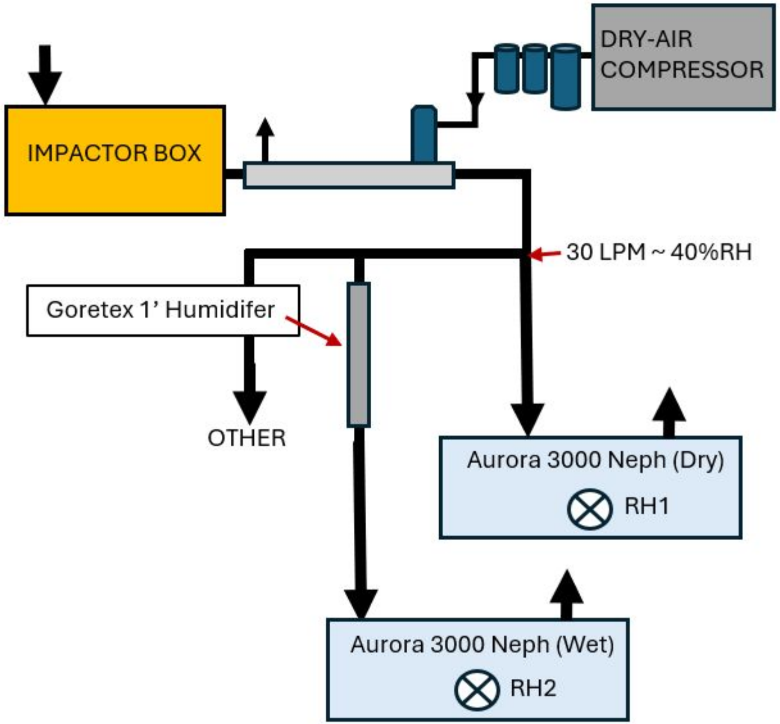
System is noisy AND responds slowly

PID requires filtering and fine tuning



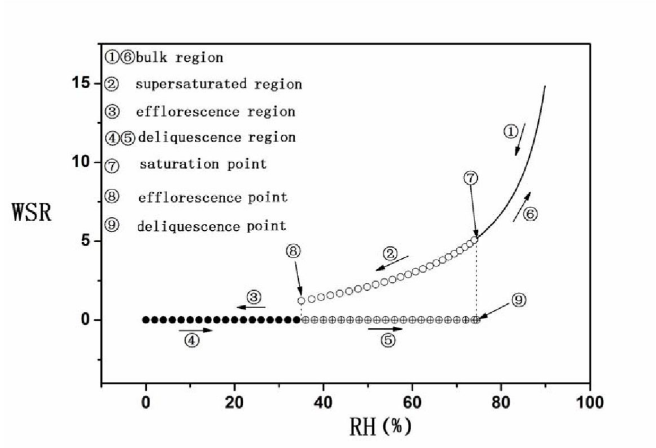


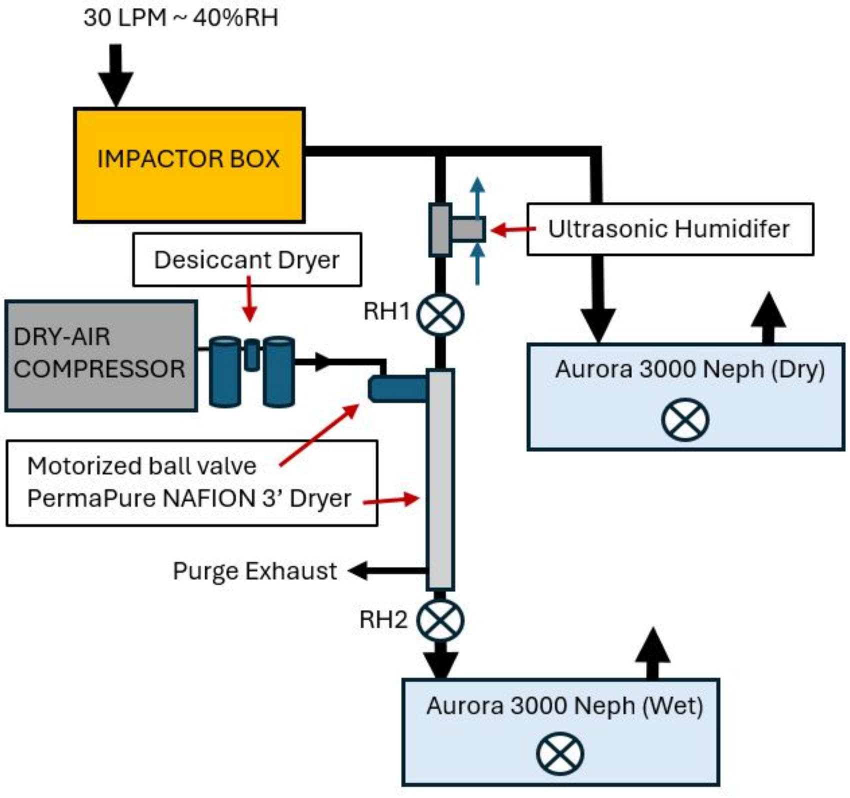
Simple Solution



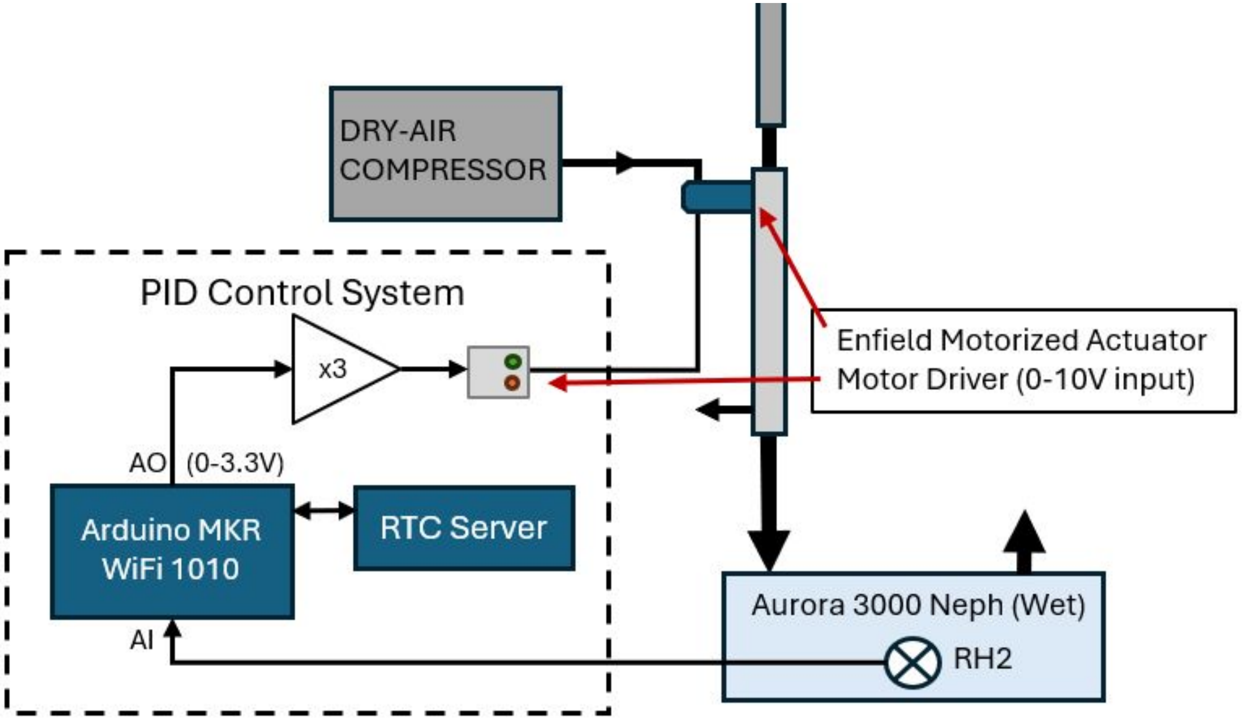
Future Work/Ideal Design

- ❖ Ultrasonic Humidifier
- ❖ Install two-stage RH
    - ➢ Observe deliquescence and efflorescence
    - ➢ Test with characterized particles





Control via Arduino/Software PID



Software Goals:

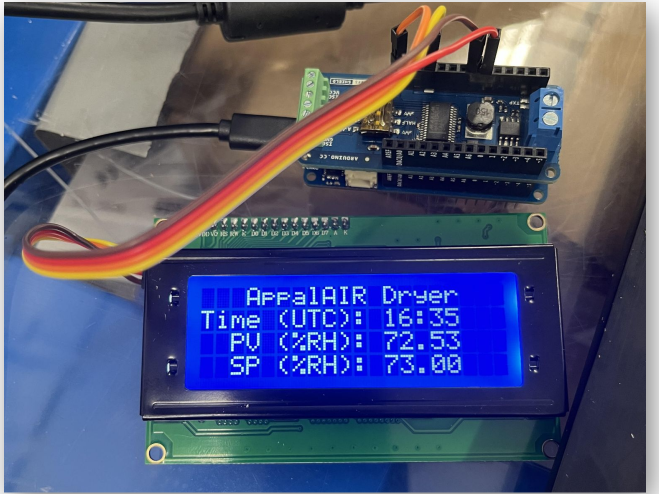
- ❖ Connects to WiFi for Time Sync
- ❖ Generates hourly SP ramp (%RH)
- ❖ Determines valve position to reach SP (PID)

Arduino Code / Documentation

```
void loop() {
  // Minute Alarm
  if (tick_RTC) {
    hour = rtc.getHours(); minute = rtc.getMinutes(); second = rtc.getSeconds(); // Read time from RTC
    updateSP();
    updateLCD();
    tick_RTC = false;
    integral = 0; // Reset integral every minute
  }

  // PID Alarm ~2s
  if (tick_PID) {
    // Average RH readings for PV
    PV = sum(PV_Array) / len(PV_Array);
    updateLCD();
    // Perform PID calculation
    output = pid(PV-SP);
    if (output > MAX_OUTPUT) { // Coerce
      output = MAX_OUTPUT;
    }
    else if (output < 0) {
      output = 0;
    }
    analogWrite(OUTPUT_PIN, output);
    tick_PID = false;
    sample = 0; // Reset samples
    PV_Array = {};
  }

  // Read RH sensor in between PID ticks
  while (sample < numSamples) {
    PV_Array[sample] = analogRead(INPUT_RH) * 100 * 3.3 / A2D_res; // Read voltage (0-1V) convert to %RH
    sample++;
  }
}
```




```
/*
  AppalAIR Dryer PID System

  This system is centered around the PermaPure MD-700 which operates using a purge flow of dried compressed air
  Using an RH sensor following the Dryer, this program uses PID functionality to calculate a voltage to control the
  Enfield valve, thus controlling purge flow rate and ultimatley the drying efficiency.

  Hardware:
  PermaPure Monotube Dryer 700 (MD-700)
  Arduino MKR WiFi 1010 (connected is the MKR485 Shield but unnecessary)
  Breadboard 3x OpAmp Circuit
  HD44780 LCD Display with I2C
  Enfield SS Motorized Needle Valve with FKM Seals (ENV-0375-SSF) with Motor Driver
  Werther PC120/4-C Air Compressor + NANPU 1/2" NPT 4 Stage Desiccant Dryer

  Software:
  WiFi is used to gather universal time and load into the on-chip RTC. This is used for accurate and synced interrupts
  Alarms are set for every dt (seconds) for the PID control, and every minute to update the SetPoint (to mimic CPD3)
  While the program awaits interrupts, it reads %RH data into an array, to average into PV before PID calculation

  Programmer: Shawn Beekman <shawnbeekman1@gmail.com>

  Last Updated April 15, 2024
*/
```

Additional Experience Gained

Troubleshooting EVERYTHING

Installing new instruments (nephelometers, CCN, SMPS/CPC)

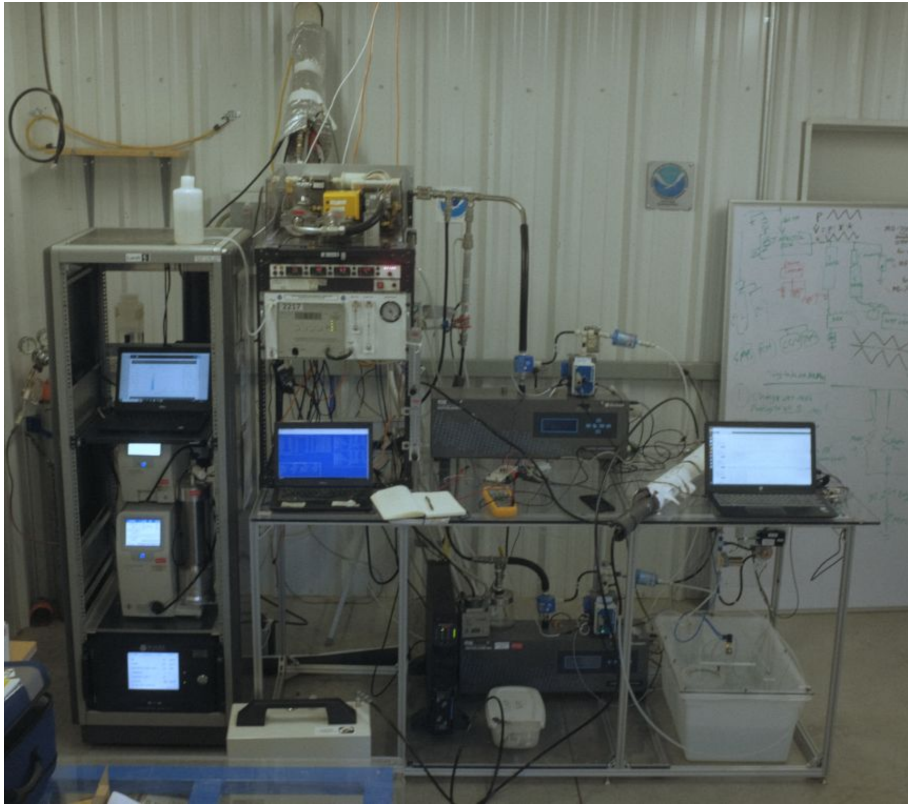
Performed instrument maintenance

Vaisala probe calibrations, flow/leak checks

Installed flow meters for troubleshooting

Assisted with SMPS closure studies

Fixed many system flow errors



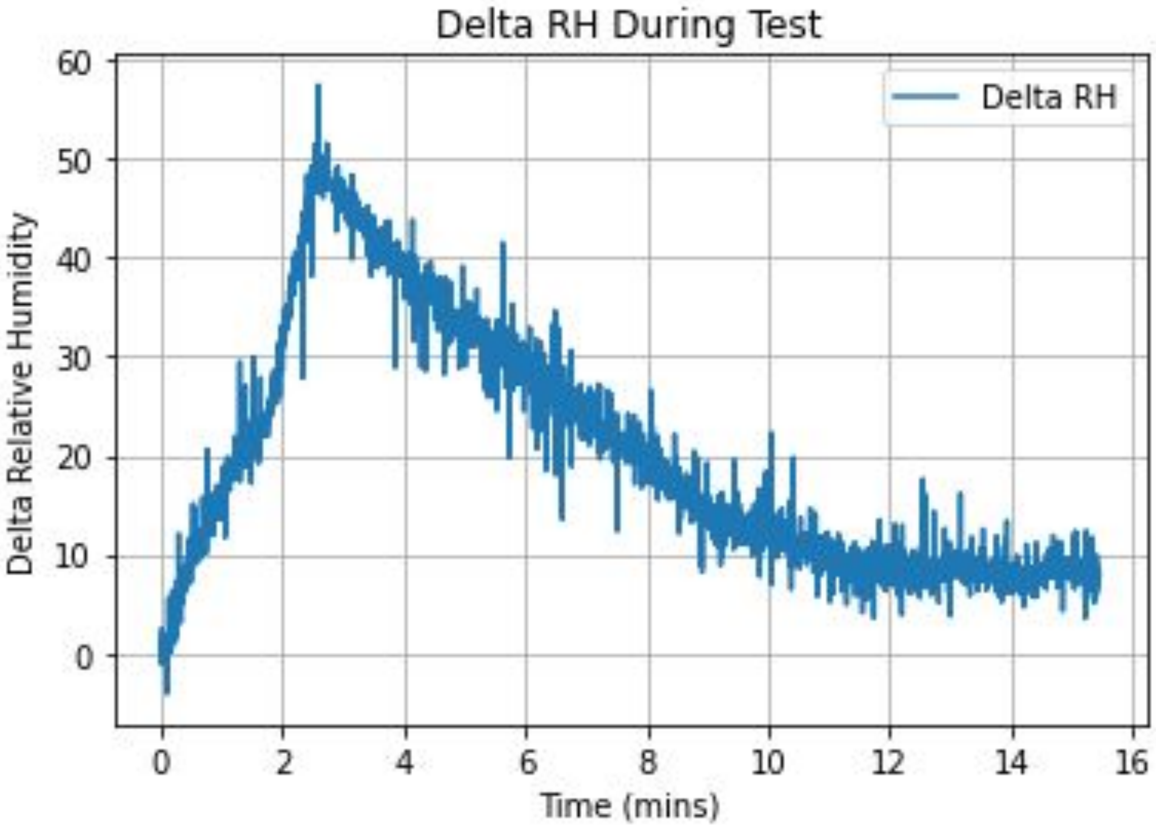
Acknowledgements

Special thanks to the wonderful AppalAIR team,

- Dr. James P Sherman
- Mx. Patrick Richardson
- Ethan Barber
- Jonathan Linderich

References

1. What are Aerosols? | BNL Newsroom
2. Climate Change Indicators: Climate Forcing | US EPA
3. Monotube Dryer 700 (MD-700) - Perma Pure
4. AMT - Measurement of relative humidity dependent light scattering of aerosols (copernicus.org)
5. ESRL Global Monitoring Laboratory - Global Radiation and Aerosols (noaa.gov)
6. Deliquescence and Efflorescence Processes of Aerosol Particles Studied by Molecular Spectroscopy | Semantic Scholar


Delta RH During Test
Delta RH
Delta Relative Humidity
Time (mins)
60
50
40
30
20
10
0
0
2
4
6
8
10
12
14
16

```
/////////////////////// MAIN CODE ///////////////////////////

void setup() {
  // Initialize Serial Ports/Wifi
  Serial.begin(9600);
  while (!Serial) {
    delay(250);
  }

  // Change Read Resolution to 12 bits for better precision
  analogReadResolution(A2D_bits);

  connectToWiFi(); Serial.println("Connected to WiFi");
  Udp.begin(localPort); Serial.println("UDP started");

  // Setup RTC
  rtc.begin();
  updateCurrentTime(); updateSP();
  rtc.setTime(hour, minute, second);

  // Minute Alarm
  rtc.setAlarmSeconds(58);            // 58s to account for delay - allows every interrupt to happen at 0s
  rtc.enableAlarm(rtc.MATCH_SS);
  rtc.attachInterrupt(RTCAlarm);

  // PID Alarm (Every dt seconds)
  for (int i = 0; i < 60; i += dt) {
    rtc.setAlarmSeconds(i);
  }
  rtc.enableAlarm(rtc.MATCH_SS);
  rtc.attachInterrupt(PIDAlarm);

  lcd.begin(); lcd.backlight(); lcd.clear();
  updateLCD();
```

```
/////////////////////////// FUCNTIONS ////////////////////////////////////////

// PID Function - feedback function to couple %RH with Actuator voltage
// Inputs: Error = PV - SP
// Outputs: Enfield Motorized Actuator Control Signal (to be converted to 0-3.3V)
double pid(double error) {
  double proportional = error;
  integral += error * dt;
  double derivative = (error - previous) / dt;
  previous = error;
  return (kp * proportional) + (ki * integral) + (kd * derivative);
}

// Update SetPoint based on the minute of hour
double updateSP() {
  if (minute == 59) {
    SP = setpoint[0];
  }
  else {
    SP = setpoint[minute+1];
  }
}

// RTC Alarm
void RTCAlarm() {
  tick_RTC = true;
}

// PID Alarm
void PIDAlarm() {
  tick_PID = true;
}
```