Abstract:

Power generation systems utilize the heat transfer properties of material, and are important in daily life since everything we use from smartphones, to cars, and lighting in our rooms need power to run. All of these power generation systems rely on heat transfer mechanisms to remove wasted thermal energy sufficient to maintain peak efficiency through running at a certain temperature. Efficiency of the cooling mechanism in power generation systems is important since it can impact the size, form factor and other important factors for consumers. If the system has high efficiency but requires a high initial cost to purchase the material necessary, then it may not be worth the investment for them. Understanding the effect of material chosen on heat transfer, and the environment it's used in, and other parameters can help us decide whether we're using the appropriate selection for the given application. To do this, we'll use the lumped capacitance model whereby a system is treated as a number of discrete locations, where each lump is isolated from other lumps in its temperature. This was done since the biot number being examined was theoretically calculated within the range of a lumped capacitance model's feasibility. In this work, we studied the effect of four different cooling scenarios, composed of two different materials (aluminum and acrylic) and two different convective environments (still and convected via a fan). To accomplish this, a heat lamp increased the temperature of the given sample to a constant temperature. Then, the temperature change was measured given free and forced air flow. The temperature of the sample, and ambient temperature during the entire experiment were measured and interpreted to determine biot numbers and heat transfer coefficients. From our results, it can be concluded that forced air convection, and use of a thermally conductive material does increase the ratio of heat flux to the energy on its surface (heat transfer coefficient), and subsequently the ratio of the thermal resistances inside of a body and at the surface of a body. The lumped capacitance model was an appropriate heat transfer model to use in some of the experimental scenarios taken as shown by low percentage differences between theoretical and experimental data, however other materials and convection modes require further refinement of the model, or need another heat transfer model all together. By knowing these approximate differences between materials, and type of convection heat transfer used, the work can be used to define cooling requirements, and choices for many components. For instance, a car's radiator can be sized appropriately given the surface area it will have and material it'll be made from. Such knowledge can help firms select the correct option when it comes to their own components.

Introduction:

Power generation systems are responsible for converting inputs such as electricity, or fuel to useful energy outputs. In formula one, power generation systems are used to efficiently convert different forms of energy inputs to maximize the performance of the race car and outperform competitors [1]. Within power generation systems, an important aspect that affects how efficient power is generated is the ability of the system to remove waste energy in the form of heat.

There are many different forms that heat energy arises in the power generation process. Conductive heat transfer is the flow of heat in a medium due to the transfer of kinetic energy from higher-energy particles to lower energy particles in direct interactions. Another method is radiative heat transfer, caused by the emission of electromagnetic waves generated by thermal motion of charged particles within the medium. Finally, convection is defined in its two forms within liquids: free and forced. In free convection, the fluid movement occurs due to pressure differences from density variations in the fluid in regions of varying temperatures. On the other hand, forced convection has a fan, pump or other form of input movement in order to accelerate the heat transfer of the fluid medium and its particles.

Within power generation systems, ways to keep the system cool while it functions at it's limit ensure maximum efficiency. To achieve this, manufacturers commonly use heat sinks and other cooling mechanisms to dissipate heat away from the system. For instance, firms can select the right parameters for their required heat transfer needs. This can impact the noise level of CPU fans in personal computers if manufacturers could select a scenario with smaller heat dissipating shrouds, and smaller fans [2]. Furthermore, batteries in our remotes, cars, and other personal electronics use heat transfer to ensure maximum efficiency range as the working window of the device [3]. In this study, the physical phenomena being studied is heat transfer and how it is affected by choosing different materials and different convective flow environments.

Being able to measure, and parameterize different forms of heat transfer can help enable large technological advances. By calculating biot numbers and decay rates - measurements of heat transfer in a convective environment - based on the experimental data acquired and, using a lumped capacitance thermal model. In summary, by analyzing the effect of forced vs. free convection, and aluminum vs. acrylic samples, it was determined that the combination of forced convection and aluminum give the highest heat transfer coefficient of 151.1 - most readily cooling down internally w.r.t. the amount of surface heat dispersed. Subsequently, switching to an acrylic material of the same forced convection mode resulted in a heat transfer coefficient of 81.28. And finally, switching to free convection resulted in a heat transfer coefficient of 50.37, and 29.83 respectively for aluminum and acrylic. These results enable manufactures and firms to select the right material given the environment it is used in (forced vs. free convection). In this experiment, a heat lamp was used in conjunction with thermistors measuring

ambient air temperature, and sample temperature to evaluate the change in temperature compared to the time taken for that change.

[1]https://www.mercedesamgf1.com/en/news/2018/10/insight-five-examples-why-f1-is-accelerating-the-future/
[2] https://users.metu.edu.tr/home102/itari/wwwhome/OzturkTariIEEETCAPT2008.pdf
[3] https://www.nrel.gov/transportation/energy-storage-performance.html

Results:

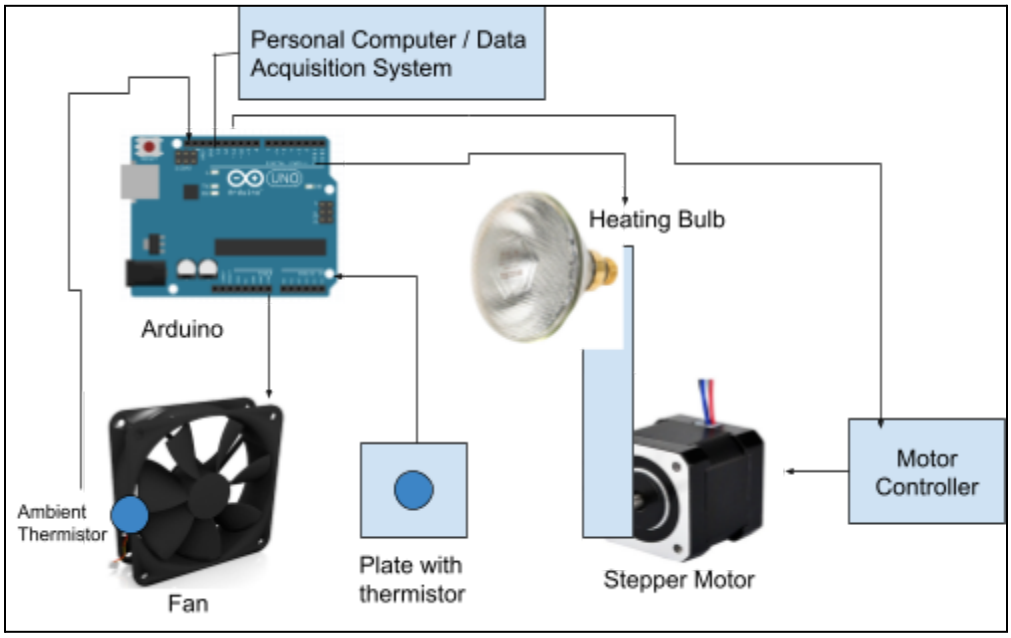

Figure 1. Experiment methodology diagram. Components highlighted are necessary to conduct the experiment as stated, however not conclusive.

The experiment was conducted via combining Matlab and Arduino programs to bring a halogen light bulb heat source on a aluminum mount, close to the plate to heat it. When a target temperature of 35 degrees celsius of the plate is reached for several seconds, the program raises the heat source away from the plate, and lets the system cool under convective (fan on) or still (fan off) conditions until the system approaches 3 degrees C of the ambient temperature. The ambient temperature of the room and the temperature of the plate are measured via thermistors as a function of time. The thermistor measuring the plate temperature is attached to the plate via thermal paste to minimize inaccuracies. Calibration for the thermistor has been implemented in the Arduino code, such that the data returned is in degrees C. The experiment was repeated three times, and then averaged to reduce random errors impacting the study. In order to evaluate the results appropriately, the assumption used in calculating biot and heat transfer coefficient numbers was the lumped capacitance model. This model assumes that heat transfer occurring has the temperature difference inside each lump as negligible, thereby treating each lump as discrete data points. This was chosen as appropriate since the expected Biot number was less than 0.1 according to calculations made [appendix], which is within the range of feasibility for the lumped capacitance model.

Upon measurement of the 4 scenarios, the datasets were plotted knowing that the curve plotted represents ln(θ) vs tau whereby $\theta = (T - T_\infty)/(T_0 - T_\infty)$ is the dimensionless surface temperature and $\tau = kAt/(\rho cdV)$ is the dimensionless time. In the dimensionless parameters given above, k is the thermal conductivity, ρ the density, c the

specific heat capacity, d the depth of the plate, V is the plate volume, A is the plate surface area; t is the time; T is the temperature of the plate, T_∞ is the temperature of the air, and T_0 is the initial temperature of the plate.

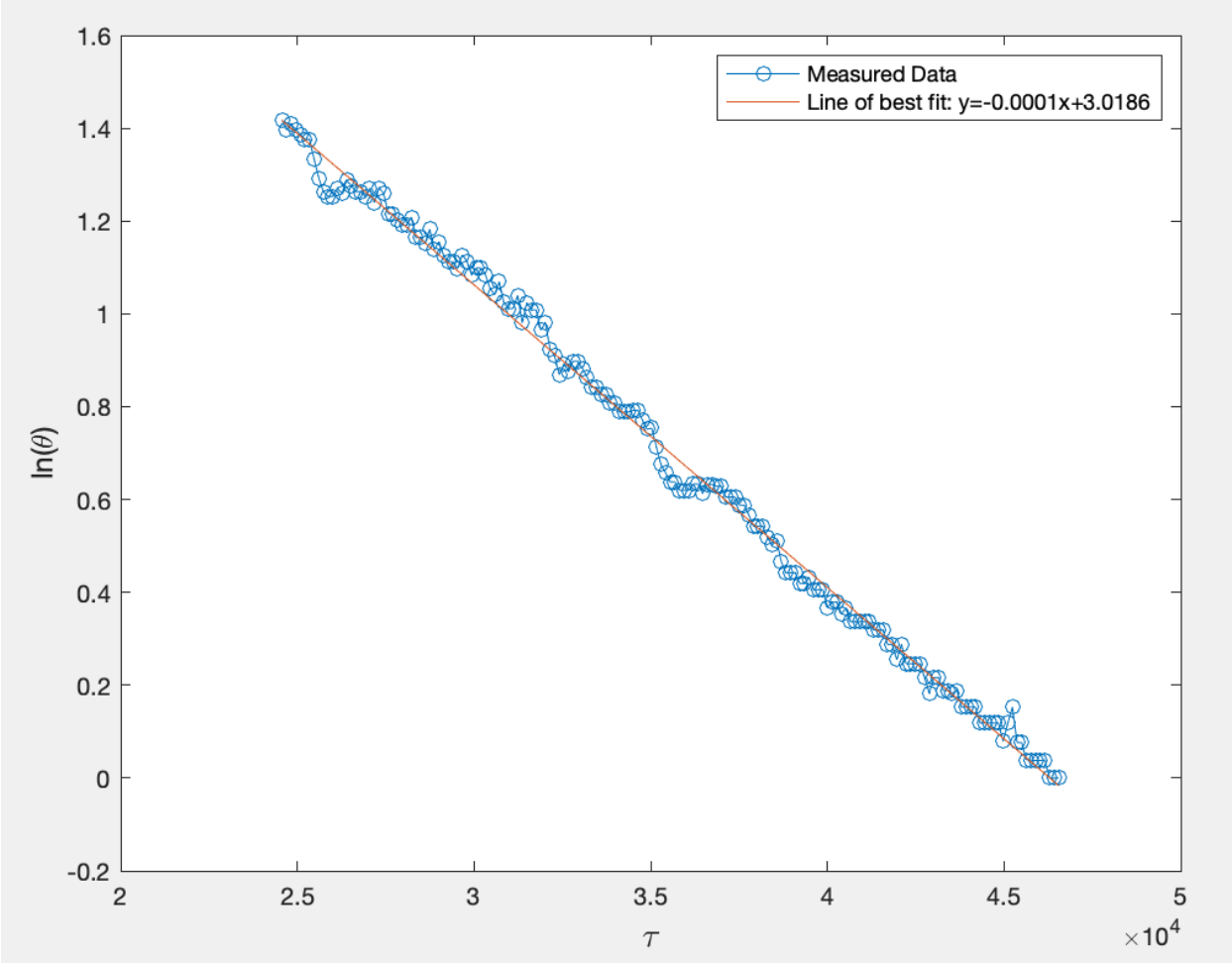

Figure 2) Thin square plate of Aluminum (d = 0.81 mm thick by 38.1 mm in width) under free convection airflow where data recorded was ambient temperature, sample temperature, and time. Measured Data was recorded at discrete intervals, and stringed together as indicated by the plot. Note: only constant cooling portion of graph is depicted

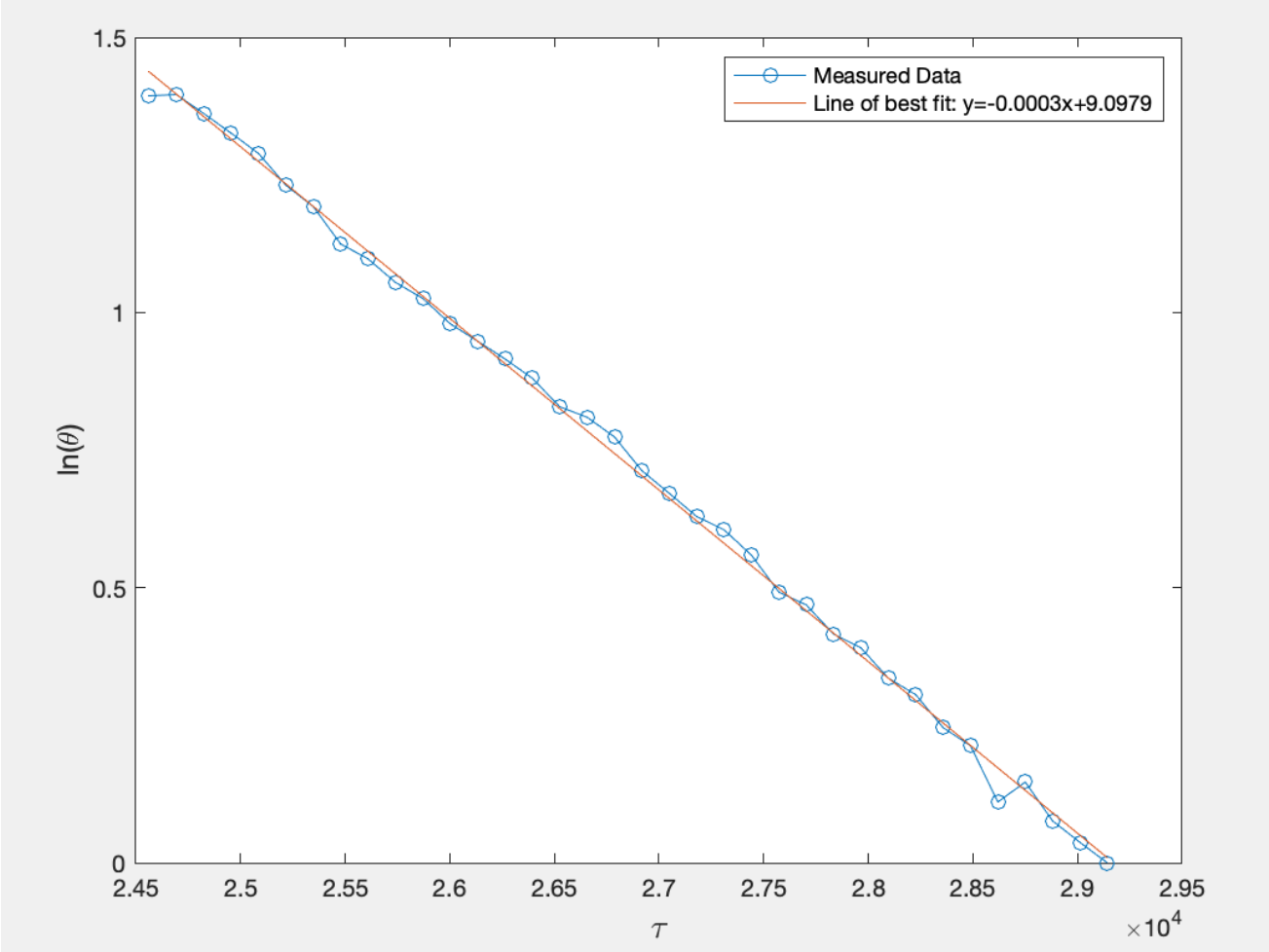

Figure 3) Thin square plate of Aluminum (d = 0.81 mm thick by 38.1 mm in width) under forced convection airflow where data recorded was ambient temperature, sample temperature, and time. Measured Data was recorded at discrete intervals, and stringed together as indicated by the plot. Note: only constant cooling portion of graph is depicted

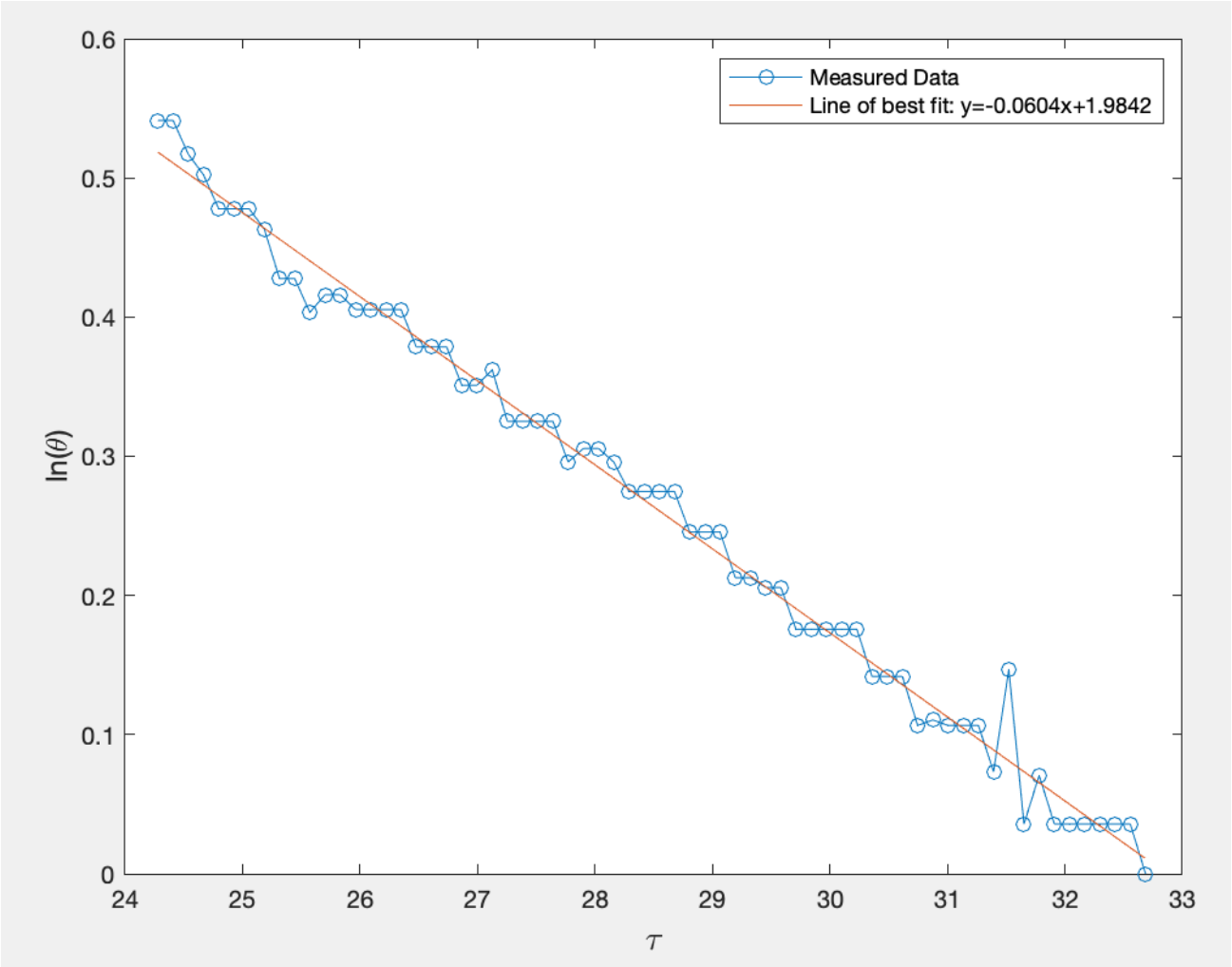

Figure 4) Thin square plate of Acrylic (d = 0.81 mm thick by 38.1 mm in width) under free convection airflow where data recorded was ambient temperature, sample temperature, and time. Measured Data was recorded at discrete intervals, and stringed together as indicated by the plot. Note: only constant cooling portion of graph is depicted

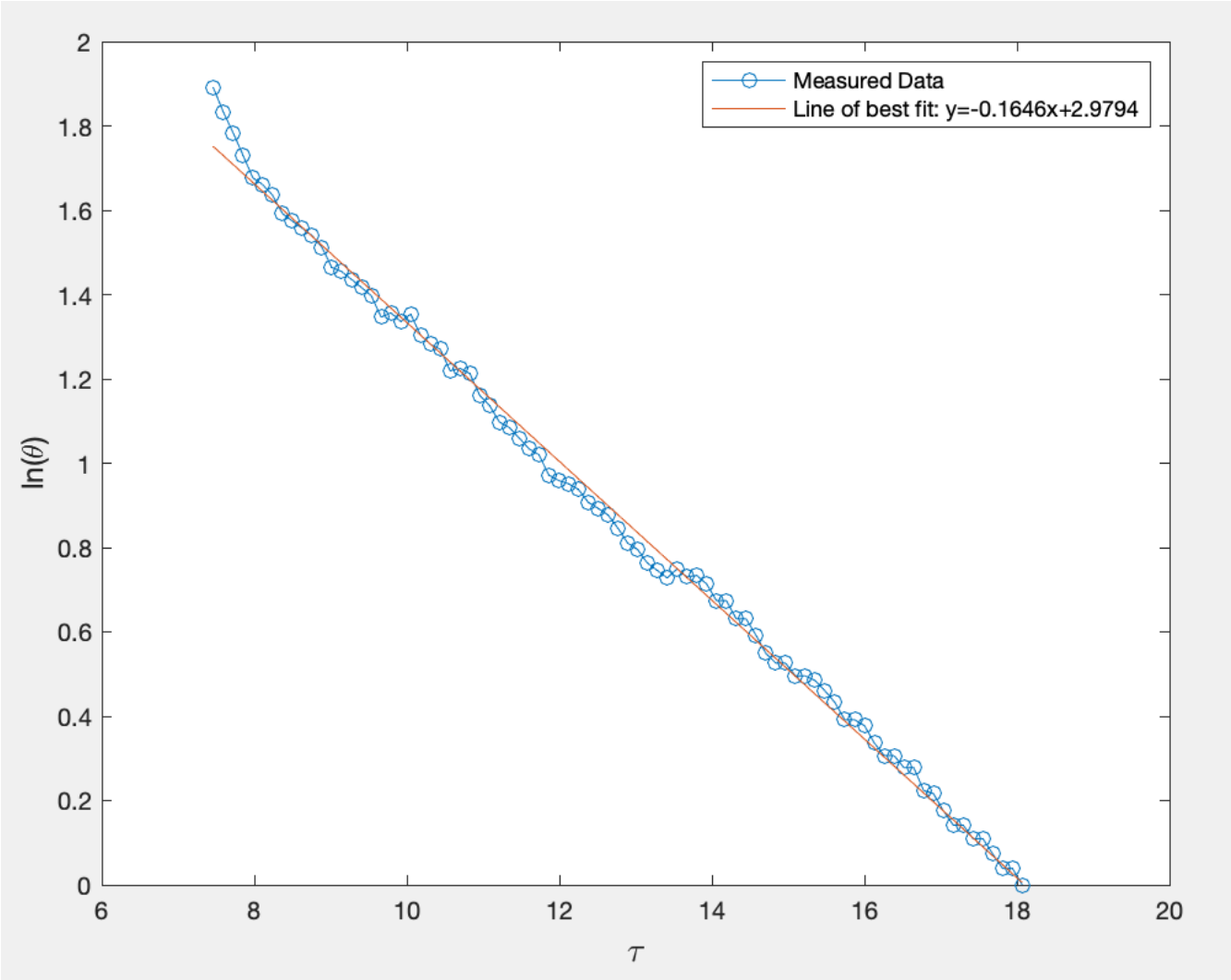

Figure 5) Thin square plate of Acrylic (d = 0.81 mm thick by 38.1 mm in width) under forced convection airflow where data recorded was ambient temperature, sample temperature, and time. Measured Data was recorded at discrete intervals, and stringed together as indicated by the plot. Note: only constant cooling portion of graph is depicted

To find the Biot number from the graphs above, the negative of the slope at cooling segments is taken. The relation between the Biot number and the heat transfer coefficient is: Bi=hL/k, where L is the characteristic length and k the conductivity. The Biot number is directly identified from the data while L and K are scientifically known constants and h is to be determined [Appendix]. From the graphs, we're able to define Biot Numbers and heat transfer coefficients as follows:

Table 1) Experimental Scenarios conducted, and their respective average Biot Number calculated. Each experimental scenario takes the average of 3 trials [Appendix]

Experimental Scenario Average	**Average Biot Number**
Aluminum free (Figure 2)	0.0001
Aluminum forced (Figure 3)	0.0003
Acrylic free (Figure 4)	0.0604
Acrylic forced (Figure 5)	0.1646

Table 2) Experimental Scenarios conducted, and their respective heat transfer coefficient. Each experimental scenario takes the average of 3 trials [Appendix]

Experimental Scenario Average	**Heat transfer coefficient**
Aluminum free (Figure 2)	50.37
Aluminum forced (Figure 3)	151.1
Acrylic free (Figure 4)	29.83
Acrylic forced (Figure 5)	81.28

The equation used to calculate sample standard deviation is:

$$S = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(xi - \overline{x})^2} \quad \textbf{(Equation 1)}$$

where S is the sample standard deviation, N is the number of observations, xi is the observed value of a single item, and $\overline{x}$ is the mean value of all the observations.

Furthermore, the percentage error of the sample using percentage deviation was calculated using the following formula:

$$\%\, error = \frac{|C - M|}{|M|} * 100 \quad \textbf{(Equation 2)}$$

where C is the calculated biot number using constant parameters as defined at the start of the experiment, and M is the measured biot using the plot results. This is multiplied by 100 given the ratio of absolute error with respect to the actual value in order to derive percentage error.

Table 3) Calculated errors for Biot Number using Eq.1 and Eq.2 from results obtained in Table 1. Results given to 4 significant figures as limited by the equipment used.

Experimental Scenario	**Sample Standard Deviation**	**% deviation**	**% error**
Aluminum free	0.0000001	0	0.00
Aluminum forced	0.0000053	2	200
Acrylic free	0.0019	3	67.6
Acrylic forced	0.0027	2	62.6

Table 4) Calculated errors for Heat transfer coefficient using Eq.1 and Eq.2 from results obtained in Table 2. Results given to 4 significant figures as limited by the equipment used.

Experimental Scenario	**Sample Standard Deviation**	**% deviation**	**% error**
Aluminum free	0.0002	5	10.02
Aluminum forced	0.0013	12	60.11
Acrylic free	0.0320	29	153.3
Acrylic forced	0.0612	24	378.2

Discussion:

Table 1 indicated Biot number increases from any material under free convection to forced convection. This suggests that our fan did work as expected, and did make a large percentage difference to the ability of a material to lose heat with respect to its shape and size. Additionally, switching from Aluminum to Acrylic resulted in higher biot numbers by 603.1% for free and 547.7% for forced convection suggested the acrylic had approximately six times the ratio of the internal resistance of its body to heat conduction, to its external resistance to heat convection, compared to aluminum

Table 2 shows that heat transfer is affected by both material, and convection mode. Free convection was again lower in magnitude than forced convection for both materials. Moreover, aluminum had higher heat transfer coefficients compared to acrylic, for both modes of convection. This confirms that the results are indeed consistent with what was expected since aluminum has a greater ratio of heat flow through a unit area and temperature difference, and higher thermal capacity in general as well - contributing to higher heat transfer coefficients.

This indicates that forced convection does greatly increase the ability of a sample to convect heat. Furthermore, aluminum with its higher thermal capacity is able to show greater thermal conductivity as well. Potential error sources include the measurement error of the experiment. To quantify this, the sample standard deviation was calculated as a measure of how dispersed the data is with respect to the mean. As indicated in table 3, sample standard deviation was calculated, and then percent deviation was also found to give context. All experimental scenarios had relatively low percent deviations, suggesting that the experiment was conducted with little random error.

However, the percentage error of the actual result compared to the theoretical result is large, for instance, approximately 200% per the aluminum under forced convection scenario. One possible explanation for these large percentage errors could be the Murata thermistor's calibration. It seems that the resistive load chosen for the voltage divider was constant at 5.63 ohms as provided [Appendix]. However, this should likely have been changed for each scenario as only the free convection aluminum experiment had zero percentage error for its biot number as shown in Table 3. Furthermore, the same scenario had the smallest percentage error for its heat transfer coefficient as shown in Table 4.

An assumption made at the start of the experiment was that the convective heat transfer coefficient h of air can range from 2.5 to 500 W/m^2K. This was used to select theoretical values of constants, and thereby in calculating biot numbers and heat transfer coefficients. This may not be the best method since the range is quite large, and unable to differentiate between convective modes. Having a larger dataset of values for heat transfer calculations to utilize in the real world could be a further application of this research in enabling better selection of heat transfer components.

For future experiments, outliers should be removed from the dataset using a suitable rejection criteria. This could be implemented using the two-sigma rejection criterion whereby any suspected data point which deviates two standard deviations from the sample mean can be regarded as outlier and therefore be taken out of the data set. This could help realize the portion of the graph with constant cooling since the current dataset becomes small when taking out transfer portions such as peak to constant cooling mode. Moreover, while statistical methods were used to approximate the error of the results, and verify they're within tolerance, this was only done for components with known tolerance. For instance, the aluminum plates could have variances within it's surface and thickness/density. This should be accounted for in further experiments by confirming the tolerance with the source of the material, and other components with unknown tolerances.

In conclusion, there's a need to measure heat transfer with instruments using proper calibration for the given experiment and models used to approximate them. Using our experiment, and the associated assumptions we made, aluminum with forced convection provided the best cooling efficiency to a power generation system. However the experiment showed that correctly calibrated murata thermistors could approximate the parameters for heat transfer more accurately. Enabling the above mentioned changes, and measuring further changes such as different fan speeds to simulate different convective environments, higher temperature points, and different materials could enable manufacturers to select the correct component for their given environment of heat transfer needs.

Acknowledgements:

I'd like to thank Andrew Enmark, Chaitanya Tambat, and Arnav Wayade for providing feedback on my first draft. Changes included error uncertainty analysis, and revisions to abstract.

APPENDIX:

Expected Biot and heat decay results using constants

Table 5) Table of Expected Biot numbers and decay rates, based on the experimental scenario described and using a lumped capacitance thermal model and the experimental parameters given in Table 1 below, where ρ denotes density, c specific heat capacity, and k thermal conductivity. As a ballpark range, the convective heat transfer coefficient h of air was considered in the range from 2.5 to 500 W/m^2K, where 50 was chosen

	ρ (kg/m^3)	***c* (J/kgK)**	***k* (W/mK)**	***Lc (m)***	***Ideal h air (W/m^2 K)***	**Biot number**	**Decay rate (W/mK)**
Aluminum	2707	879	204	0.000405	50	0.00010	235
Acrylic	1180	2000	0.2	0.000405	50	0.10125	73

* Ideal heat transfer coefficient gathered from https://www.sciencedirect.com/topics/engineering/thermal-conductivity-coefficient

Table 6) Three trials of experimental scenario: Aluminum free. Results given to 4 significant figures as limited by the equipment used.

Trial	**Biot Number**	**Convective Heat Transfer Coefficient**
1	0.0001	50.3525
2	0.0001	50.4654
3	0.0001	50.2713

Average	0.0001	50.3704

Table 7) Three trials of experimental scenario: Aluminum forced. Results given to 4 significant figures as limited by the equipment used.

Trial	**Biot Number**	**Convective Heat Transfer Coefficient**
1	0.0003	150.5432
2	0.0003	148.1329
3	0.0002	153.1943
Average	0.0003	151.11111

Table 8) Three trials of experimental scenario: Acrylic free. Results given to 4 significant figures as limited by the equipment used.

Trial	**Biot Number**	**Convective Heat Transfer Coefficient**
1	0.0525	28.54901
2	0.0725	29.34420
3	0.0592	30.19593
Average	0.0604	29.82716

Table 9) Three trials of experimental scenario: Acrylic forced. Results given to 4 significant figures as limited by the equipment used.

Trial	**Biot Number**	**Convective Heat Transfer Coefficient**
1	0.1332	89.193934
2	0.1920	80.490202
3	0.1853	82.305342
Average	0.1646	81.283951

Thermistor calibration. We use Murata thermistors, Part No: NXFT15XH103-FA2B100. Information on the thermistors can be found in the Murata Thermistor.pdf file included in the **'Lab 6'** folder of the files section of the course webpage. Highlighted sections of p. 7 and p. 31 have the necessary information, relevant to your thermistor model. In summary, the nominal resistance of your thermistor is R_0 = 10 kΩ at nominal temperature T_0 = 298.15 K (25 degrees C). The thermistors resistance R behaves as a function of temperature as $R = R_0 e^{B(1/T - 1/T_0)}$ where $B = 3380 \pm 1\%$ K and T is the temperature in degrees K. While the response is exponential, it's voltage output can be linearized using a voltage divider, as shown in Fig. A1, such that $V_{out} = V_{in}[R_{load}/(R_{th} + R_{load})]$, where R_{load} is chosen to be 5.63 kΩ, R_{th} is the thermistor resistance, and V_{in} is 5 V supplied from the Arduino. Fitting a line to the linearized relationship between temperature in Celsius and voltage, over a range of 25 to 50 degrees C, results in a sensitivity (calibration) relation: $T_c = V_{out}23.1273 - 16.7901$.

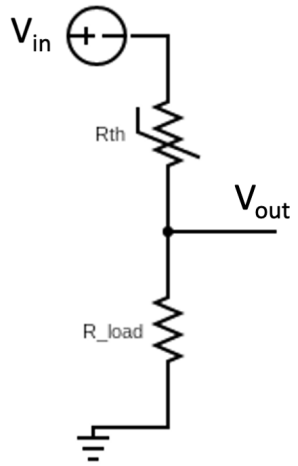

Figure A1: Thermistor and linearizing voltage divider circuit.

Experiment automation circuit. The key components of your automation setup are an Arduino, a custom Arduino add-on board (referred to as a "Shield"), and a 12V 2A DC power supply. This shield contains:

- A DRV8825 Stepper Motor Controller which runs a NEMA 17 stepper motor (Online model 17HS19-2004S1)
- A IRLZ34N MOSFET which controls an MR11 halogen reflector light bulb with 12 V and 35 W max
- A BS170 MOSFET which controls a 12V DC fan

These components are connected as shown in the diagram below in Figure A2.

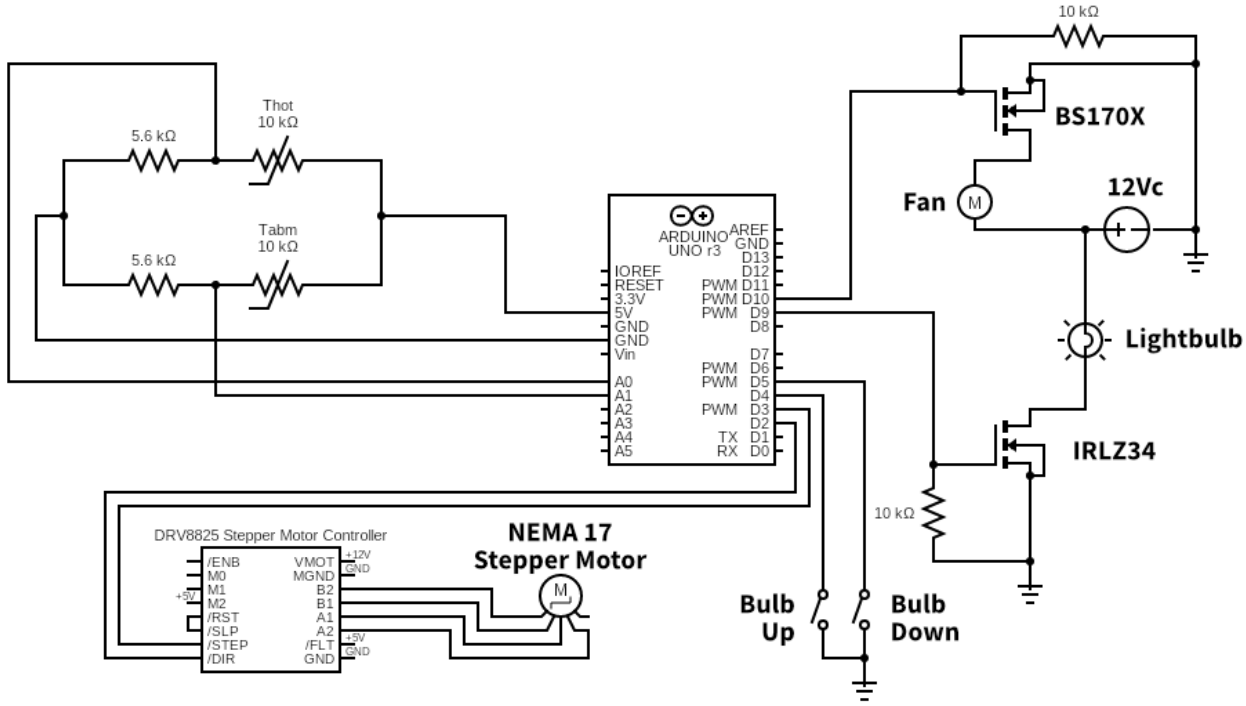

Figure A2: Experiment automation circuit diagram.

Matlab script (MAE170_Lab6_Fall2022.m)):

```
clc;
clear all;
close all;
instrreset;

%%
maxTemp = 35; % max temperature to achieve, in Celsius
ambTol = 3; % ambient / plate cutoff tolerance
avg=3; % number of repetitions
%%


for av = 1:avg

    disp('Connecting to Arduino...');
    s = serialport('COM13', 9600);
    pause(5);

    tempPlateC = [];
    tempAmbC = [];
    time = [];
    i = 1;

    writeline(s, string(maxTemp));
    heating = true;
```

```
    cooling = false;
    disp('Waiting for cycle to begin...');
    pause(5);
    figure(01);
    disp(['Beginning cycle ' num2str(av)])

    while heating || cooling
        out = char(readline(s));
        index1 = find(out==',',1);
        index2 = find(out==';',1);
        tPlate = str2double(out(1:index1-1));
        tAmb = str2double(out(index1+1:index2-1));
        milli = str2double(out(index2+1:end)) / 1E3;
        if heating & tPlate > maxTemp
            heating = false;
            cooling = true;
        end
        if cooling & tPlate < tAmb + ambTol
            cooling = false;
        end

        time(i) = milli;
        tempPlateC(i) = tPlate;
        tempAmbC(i) = tAmb;

        plot(time, tempPlateC, '-or', time, tempAmbC, '-ob')
        xlabel('Time (s)');
        ylabel('Temperature (C)');
        ylim([15 maxTemp+5]);
        title(['Convection over flat plate Iteration ' num2str(av)]);
        legend('Plate', 'Ambient')
        drawnow()

        i = i + 1;
        %         pause(deltaT);
    end
    disp('Cycle Complete! Saving Data...')
    save(['lab6' 'Average' num2str(av) '-' ...
        num2str(floor(now*1E3)) '.mat']);

    clear tempPlateC tempAmbC time s;
    close;
    if av ~= avg
        % Pause time is required to ensure that the Arduino has enough
        % time to reenter standby mode & be ready to receive the command
        % to begin the next cycle.
        disp('Waiting for test to reset...')
        pause(10);
        disp('Next cycle will begin in 5 seconds...')
        pause(5);
    end

end
disp('Test Complete!')
```

Arduino script (MAE170-Lab6-Fall2022.ino):
****NOTE** This script is being provided here for your reference - this code DOES NOT need to be uploaded to the Arduinos (it should already be present).**

```
/*  filename = MAE170-Lab6-Fall2022.ino
    MAE170 radiant heat source heat transfer experiment
    SWR 18 Feb 2022) - 11 Aug 2022
    MR11 bulb, 12V, 2.8A, 35W
    NEMA 17 step motor StepperOnline model 17HS19-2004S1
    Uses Pololu DRV8825 step driver @ 16:1 ustepping
*/

byte stp   =    3;     // step pin
byte dir   =    2;     // direction of rotation
#define down    5      // lamp down button
#define up      4      // lamp up button

#define OK      6      // OK button
#define Thot   A0      // heated thermistor
#define Tamb   A1      // ambient thermistor
#define bulb    9      // halogen bulb control
#define fan     10     // fan control

int  dlyon    = 100;   // uS delay for pulse hi
int  dlyoff   = 4000;  // uS delay for pulse lo
int  n90      = 800;   // # steps for up rotation
int  ScanDly  = 1000; // loop length, mS

int  cooling = 0;      // 0 = not in cooling mode, 1 = cooling
int i;          // reuseable int i
float  Tset;        // Max degrees C set point
int  Thold = 0;
int TholdSet = 15 ;    // how long to hold Tset (sec)
float Tplate, Tair;    // plate temp, air temp (deg C)
long Tstart;           // initialize elapsed millisecs

int standBy = 1 ;        // start in "standby mode" - accept button inputs, wait for matlab
serial command
int firstRun = 0;        // flag for a the first block of code that needs to be run for each
test

void rotate(byte x) {          // make 1 atep on the step motor
  digitalWrite(dir, x);        // set direction (x=0/1,up/dn)
  digitalWrite(stp, 1); delayMicroseconds(dlyon);  // pulse hi
  digitalWrite(stp, 0); delayMicroseconds(dlyoff); // pulse lo
}

void setup() {
  pinMode(stp, OUTPUT);       // pulse pin for step motor
  pinMode(dir, OUTPUT);       // motor direction 1=CCW, 0=CW
  pinMode(down, INPUT_PULLUP); // jogs bulb toward plate
  pinMode(up,  INPUT_PULLUP); // jogs bulb away from plate
  pinMode(OK,  INPUT_PULLUP); // OK when bulb is in heating position
  pinMode(bulb, OUTPUT);      // bulb on/off control
  pinMode(fan, OUTPUT);       // fan on/off control
  Serial.begin(9600);         // set serial baud rate

  digitalWrite(bulb, 0);         // turn off bulb
  digitalWrite(fan, 0);          // turn off fan
} // end of setup
```

```
void loop() {
  if (standBy) {
    digitalWrite(bulb, 0);      // turn off bulb
    digitalWrite(fan, 0);       // turn off fan
    if (Serial.available() > 0) {
      String s = Serial.readString(); // read matlab setpoint temperature in from serial
      Tset = s.toFloat();
      standBy = 0;
    }
    while (digitalRead(down) == 0) {
      rotate(1);   // rotate bulb toward plate
    }
    while (digitalRead(up)   == 0) {
      rotate(0);   // rotate bulb away from plate
    }
  } else {
    if (not firstRun) {
      for (i = 0; i < n90; i++) rotate(1); // rotate bulb down
      for (i = 0; i < 255; i++) {
        analogWrite(bulb, i);  // soft turn-on
        delay(20);
      }
      firstRun = 1;
    }


    if (cooling == 0) {                         // if not in cooling mode
      if (Tplate < Tset) {
        analogWrite(bulb, 150); // plate is below set point
      }
      if (Tplate > Tset) {
        analogWrite(bulb, 0); // plate is above set point
      }
    }

    int T1 = analogRead(Thot);                 // read T hot
    float T1V = 5.0 * T1 / 1023.;              // assumes Vusb = 5.00
    Tplate = T1V * 23.7585 - 19.064;           // cal equation

    int T2 = analogRead(Tamb);                 // read T amb
    float T2V = 5.0 * T2 / 1023.;              // assumes Vusb = 5.00
    Tair = T2V * 23.7585 - 19.064;             // cal equation

    Serial.print(Tplate, 2); Serial.print(",");     // plate temperature (C)
    Serial.print(Tair, 2); Serial.print(";"); // free-air temperature (C)
    Serial.print (millis());           // total mS
    Serial.println();

    if (cooling == 0) {                          // if not in cooling mode...
      if (abs(Tplate - Tset) < 1.0) Thold++;     // count # seconds when T-Tset < 1 degC
      if (Thold > TholdSet) {                    // shut down heating control
        cooling = 1;                            // flag in cooling mode
        digitalWrite(bulb, 0);                  // bulb off
        for (i = 0; i < n90; i++) rotate(0);    // rotate bulb up
        digitalWrite(fan, 1);                   // turns on fan, if switch enabled
      }
    } else {
      if (Tplate < Tair + 1.5) {  // if plate temp is close to ambient air temp
         cooling = 0;                // reset flags back to initial values to wait for next
command from MATLAB
        standBy = 1;
```

```
        firstRun = 0;
      }
    }
  }
  delay(ScanDly);
}  // end of loop
```