Indian Institute of Technology Gandhinagar

Heat and Mass Transfer
ES 311

Course Project Final Report

ARDUINO BASED TEMPERATURE SENSOR

Group Number: **A3**

Abhiraj Mohan - 20110004
Ajay Kannaujiya - 22120021
Ayush Gupta - 20110031
Ayush Bhalala - 20110037
Dablu Kumar - 20110049
Dwip Dalal - 20110063
Kush Patel - 20110131
Priya Gupta - 20110147
Krish Raj - 20110160
Shantanu - 20110191
Somesh Pratap Singh - 19110206
Thejus Vinod - 20110216

Under the Guidance of
Prof. Soumyadip Sett

TABLE OF CONTENTS

1 Abstract

Temperature control and temperature sensors are essential aspects of a lot of industries. Many complex phenomena in these industries are precisely monitored because of their temperature dependence. Devices and sometimes whole functioning units fail because of uncontrolled and overshoot temperature values. We present a small-scale model to control and monitor the surface temperature of a thermoelectric cooler. We used a heat sink with constant feedback from the PID controller to maintain the surface temperature of the Peltier. As a significant improvement, we have used a motor driver for analog control. Further on, a comprehensive study on the variation of system parameters and control strategy is performed and presented. With proper planning and analysis, our approach and experiments hold the potential to be scaled to applicable industrial scale.

Keywords: Thermoelectric cooler, PID controller, Heat sink unit, Feedback controller

2 Introduction

Temperature control and temperature sensors are essential aspects of a lot of industries. Taking the example of the Food and medicine industries [1], any small changes in temperature or increase in the surrounding humidity during its production might adversely affect the manufacturing process and ultimately degrade the quality of the product [2]. The need for constant monitoring and control of temperature has been in demand for quite a long time, and new methods are being developed to make that process as efficient as possible. Recently, many machines have components that work on the principle of the Peltier effect. These components are called thermoelectric coolers. They generate a heat flux from one side to another such that the first surface experiences a drop in temperature and the other has a temperature rise. Quite often, this is a fast process and consumes electrical power. The temperature rise and fall are sometimes too fast, damaging the setup. So to make sure that these changes are gradual, it is necessary to use a heat sink, a passive heat exchanger used to transfer heat from mechanical surfaces to fluids like air.

Arduino Uno is commonly used to monitor temperature because it is a readily available open-source hardware development board efficiently. Using Arduino, one can send and receive signals from different components in the circuit, which opens up the window of opportunity to control the temperature of a system. Many specific details, including the amount of voltage to be supplied and the on/off signal, can be controlled using Arduino. Even though much research has been done already, many industries still use costly heat-controlling units. Since thermoelectric coolers and set-up for their temperature control is expensive for their specified efficiency, there is a need for an economical alternate option. The project aims at using Arduino Uno to make an inexpensive and robust temperature sensor for the Peltier using a heat sink. The secondary aim of this project is also to measure the humidity of the surrounding using a humidity sensor.

3 Motivation

Many processes around us involve monitoring and maintaining temperatures of different surfaces. Some of these are very subtle, and we often ignore the magnificent engineering aspect behind them. Laptops and desktops have become essential to our personal and professional lives. Billions of different calculations, data readings, and data transfers happen in fractions of seconds. It is commonly observed that when there is a lot of load on the system, a fan inside the central processing unit rotates faster, generating an audible sound. The computer system works on electrical power and has many components that heat up gradually. Cooling them is necessary because the excess temperature might damage the circuit or harm the semiconductors inside. As the fan rotates faster, it sucks in cold air and blows out hot air, thus cooling the system.

Air conditioners also face heating issues. Although this is because of low refrigerant and dirt inside the components, examples like these make us wonder how exactly the temperature of a particular object is maintained efficiently. After reviewing different articles, we concluded that it is very costly to repair the temperature-controlling units and sometimes requires good maintenance. There is a need to make it affordable and efficient. Concepts of heat and mass transfer are comprehensive and broad; however, one of the main takeaways of it is studying the temperatures of different objects subjected to different conditions at varying times. Considering the above reasons and given that Arduino is an essential component for engineering students to learn more about mechatronics, Simulink programming, and other engineering skills, we were naturally more inclined to work on a project involving Arduino programming.

4 Problem Statement

To control the surface temperature of a thermoelectric cooler using an Arduino-based sensor.

5 Experimental Method

5.1 Experimental apparatus

The apparatus for our project experiment includes the following components:

- First and foremost, the **Thermoelectric Cooler** (which shows the Peltier Effect), of which we are measuring the temperature using a temperature sensor (**NTC** Thermistor, operating range -50℃ to +100℃), initially we planned to use an **LM35** (operating range, -55℃ to 150℃) temperature sensor.
- Then, we used an **Arduino UNO** Board, the main controlling unit as a microcontroller. It controls the signals running through different components. Then we have a Heat sink and Fins, which are used for efficient heat dissipation from the Thermoelectric cooler. We used an **SMPS** (Switch Mode Power supply) as the power source for the experimental setup.
- During the initial phase, we used a solid state relay to control the circuit, and eventually, we switched to a **Motor Driver** (Cytron DC motor Driver rated 5V-30 Amps). Finally, to measure the humidity, **DHT22** (operating range: -40℃ to 80 ℃ at 5V) is used, which is a combined sensor for both temperature and humidity. To verify the temperature recording of the sensors, we used a Digital temperature meter. Finally, to connect all the circuit elements along with Arduino UNO, we used BreadBoard and Jumper wires.

Specifications for TEC-12706

Material	Alloy Steel
Maximum Temperature Difference	~ >65℃
Operating power	60W, (12V - 5Amps)
Dimensions	40 x 40 x 4 mm^3

5.2 Development of the Experimental Setup

For our experiment, we have selected a Thermoelectric cooler (TEC1 12706) and connected the heat sink with a hotter surface of Peltier to dissipate the heat and avoid damage on the hotter surface. At the same time, we connected the fins to the colder side of Peltier. The thermal paste establishes thermal contact between the heat sink and fins. The double-sided Teflon tapes insulate four sides of the Peltier for 1-D conduction. The temperature sensor is attached to the upper surface of the colder surface between the fins assuming temperatures are the same at each point of the fins. The heat sink and Peltier are connected because we want to turn on/off the Peltier together (at the same time). To regulate the voltage supply of Peltier, we have attached the motor driver.

5.3 Experimental Procedure

1. The setup assembly and connection should be checked before starting the experiment.
2. The voltage supply of 12 V is provided through the SMPS to start the Thermoelectric cooler and the heat sink simultaneously.
3. A temperature gradient exists between the two surfaces of the Peltier just after supplying the voltage.
4. The temperature is measured by the temperature sensor connected to the colder surface of the Peltier.
5. The measurement of the temperature is sent to the PID controller.
6. The PID controller decides whether the voltage supply should be ON or OFF.
7. The signal is sent to the motor driver, which regulates the temperature.

8. The temperature sensor is continuously sending the signal to the PID controller. Therefore the voltage supply is provided when the temperature is above a limit, and the voltage supply is switched OFF when the temperature falls below the limit.

6 Analytical/Numerical Analysis

The thermoelectric cooling system is based on the 'Peltier effect'. A cooling effect (Qc) will occur on the lower side of the thermocouple (where heat is absorbed), and a heating effect (Qh) will occur on the upper side of the thermocouple when a direct current (DC) flows through a circuit consisting of various metal conductors (where heat is expelled). This is known as the 'Peltier effect'. The efficiency of a Thermoelectric cooler module is a factor of

1. Amount of heat transferred
2. Temperature difference between its sides
3. Applied value of direct current

Heat produced or absorbed per Peltier element is

$$Q_p = \alpha * I \;\ldots\ldots\ldots\ldots (1)$$

Where α accounts for the amount of heat transported by a single charge at the cold side.

$$\alpha = (\alpha_1 - \alpha_2).T \;\ldots\ldots\ldots\ldots\ldots (2)$$

α_1 and $\alpha_2$respectively the thermoelectric voltage coefficient of semiconductor type p and n junctions. Whereas T is the temperature on the relevant side of the Thermoelectric cooler.

Other effects contributing to the working of the thermoelectric cooler are the Joule effect, Seebeck effect, and Thomson effect. Peltier and Joule effects occur in coupling and have to be accounted for together.

The rate of thermal energy generation by the Joule effect is:

$$Q_J = \frac{1}{2} I^2 R \;\ldots\ldots\ldots\ldots\ldots\ldots (3)$$

Q_J is the thermal energy generation rate due to Joule heating. And, R is the resistance of the thermoelectric cooler.

Amount of heat transfer from the hot side to the cold side is:

$$Q_e = \kappa_{th}(T_h - T_c) \text{ .... (4)}$$

κ_{th} is the thermal conductivity coefficient of the material used in a thermoelectric cooler. T_h, T_c are the temperatures of the hot and cold sides, respectively.

Heat transfer rate between the two sides of the thermoelectric cooler is calculated using the following equations:

Use equations (1) , (3), (4), we get

$$Q_c = \alpha.I - \frac{1}{2}I^2R - \kappa_{th}(T_h - T_c) \text{ .... (5)}$$

$$Q_h = \alpha.I + \frac{1}{2}I^2R - \kappa_{th}(T_h - T_c) \text{ .... (6)}$$

Q_c also represents the cooling production by the Peltier junction.

Other relevant physical parameters are:

Electric Voltage (V):

$$V = \alpha.(T_h - T_c) + I.R \text{ .... (7)}$$

Input electric power (W):

$$P = \alpha.I.(T_h - T_c) + I^2R \text{ .... (8)}$$

Also, if we take the Seebeck effect (electricity is generated between a thermocouple when the two ends of it are subjected to a temperature gradient) into consideration:

$$\Delta T = \frac{(S.I.T_h - \frac{1}{2}I^2R)}{(S.I + \kappa_{th})} \text{ .... (9)}$$

I in such a case would become

$$I = \frac{V - \alpha.\Delta T}{R} \text{ .... (10)}$$

This would further go on to effect the cooling production rate and the new cooling production rate taking Seebeck effect into account is given by [3][4][5][6]:

Use equations (5), (9) and (10) to give:

$$Q_{c.Seebeck} = \frac{\alpha.U.T_c}{R} + \frac{\alpha^2.T_c.\Delta T}{R} + \kappa_{th}\Delta T - \frac{U^2}{2.R} + \frac{(\alpha.\Delta T)^2}{2.R} \text{ .... (11)}$$

As applied by Chen and Sun., 1999 and Guo et. al. 2020 [7][8], Thomson effect which is the absorption of heat when some electric current passes through a circuit made up of single material having a temperature gradient along its length.

Some other phenomena to take into consideration are

The convective heat transfer rate from a Peltier surface (say hot side to the surrounding) is given by

$$q''_{natural} = h_{natural}(T_h - T_{ambient}) \text{ .... (12)}$$

And when the fan is turned on in the heat sink:

$$q''_{heatSink} = h_{forced}(T_h - T_{ambient}) \text{ .... (13)}$$

Where $h_{forced} > h_{natural}$

For the general condition of convective heat transfer at the fin tip, we can find the temperature profile within the fin (Assumptions: 1D conduction and the analysis is valid for given instants) (Note that steady state assumption can not be applied because the T(x) here is varying with time i.e. we are trying to control the temperature)

The two boundary conditions that are deployed are (x = 0 at the fin base and x = L at the fin tip)

1. $-k_{th}\frac{\partial T}{\partial x}|_{x=L} = h(T_{x=L} - T_{ambient})$

2. T at (x = 0) = Specified temperature at the base (T_b)

The fin temperature profile is [9]

$$\frac{\theta}{\theta_b} = \frac{T - T_{infinity}}{T_{b-} T_{infinity}} = \frac{cosh(m(L-x)) + \frac{h}{mk}(sinh(m(L-x)))}{cosh(mL) + \frac{h}{mk}(sinh(mL))} \text{ ……………………. } (14)$$

Where $m = \sqrt{\frac{hP}{kA_C}}$

P is the perimeter of the fin = surface area/unit length. And, A_c is the fin surface area

One of the most important performance aspects is the Coefficient of Performance (COP) of the Peltier element:

$$COP_{Peltier} = \frac{q_{dot}}{W} = \frac{|-k_{th}\frac{\partial T}{\partial x}|}{Rated\ power\ of\ the\ module}$$

$\frac{\partial T}{\partial x}$ evaluated for two points on the base plate and the sink

7 Control strategy - PID controller

In this case, we want to control the surface temperature, so we shall implement a control loop that will take in the current state of the system and then calculate the error and, based on the error, augment the input. After going through the corpus of literature, we listed some widely used control strategy:

1. ON - OFF (also called Bang–bang) controller
2. Proportional controller (P)
3. Integral controller (I)
4. Proportional + Integral controller (PI)
5. Proportional +Derivative Controller (PD)
6. Proportional +Integral + Derivative Controller (PID)

On further inspection we came to the conclusion that PID outperforms all the other traditional controllers. A PID controller is a feedback control system used to regulate the temperature of an object, such as a heating or cooling device. The input to the controller is the difference between desired and actual set point.

PID Consists of three controllers:

1. Proportional Controller

   It is a linear feedback control system where the correction is applied to the controlled variable, and the size of the correction is proportional to the difference between the desired value (set temp) and the measured value (current temp).
2. Integral Controller

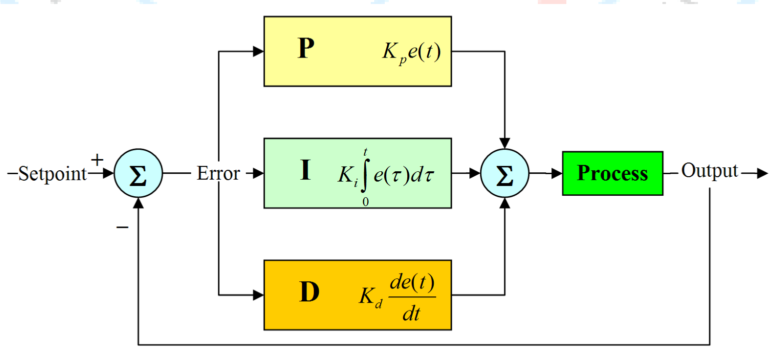
   It is mostly used along with the Proportional controller, and it helps in eliminating the steady-state error that occurs with a proportional controller.
3. Derivative Controller

   In this case, the control action is done in a way that the controller's output is proportional to the rate at which the error signal changes over time.

The mathematical description of this controller is as follows:

$$u(t) = K_p e + K_d \frac{de}{dt} + K_i \int_0^t e(t)dt$$

And the Schematic diagram is as follow:



7.1 Code Explanation

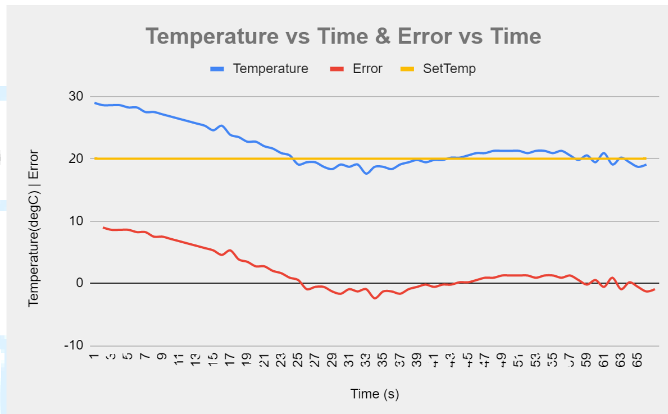
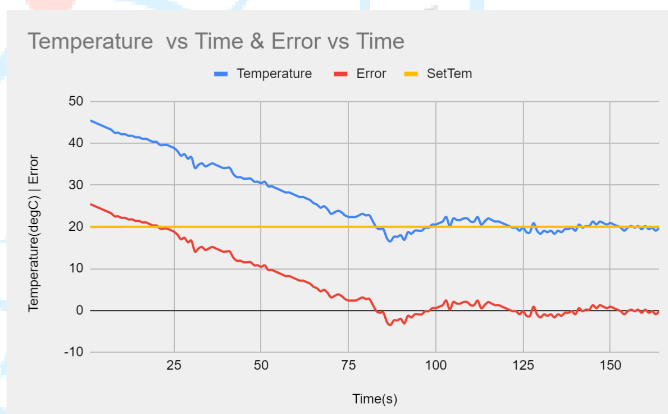
Initially we used a latch relay and implemented a naive Bang-Bang controller. The logic behind it is: if the current temperature is greater than the set temperature, then it would turn ON the thermoelectric cooler, and if the current temperature is lesser than the set temperature, then it would turn OFF the thermoelectric cooler. This didn't gave a good result and wasn't able to reach set temperature. So we then used the well know feedback controller: The PID Controller. The corresponding code can be found here:

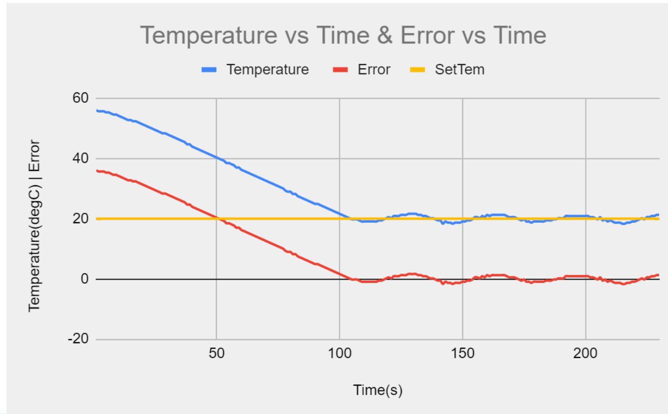
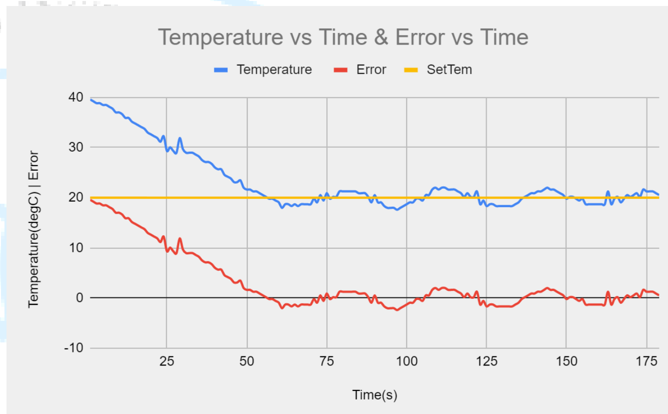
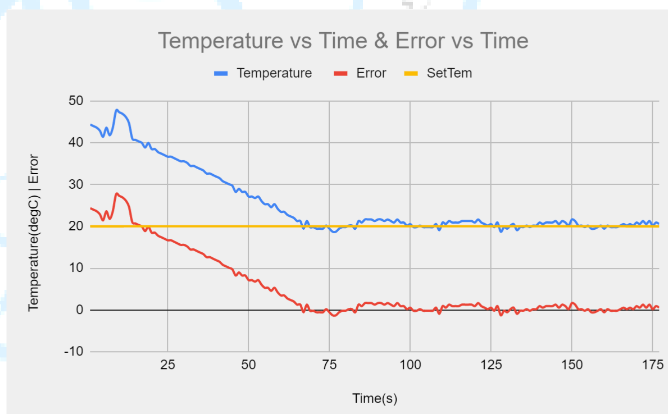
```
void Control_PID(double iTemp){
        if(iTemp>MaxTemp){
          analogWrite(relay, 255); // Temperature outside the bounds.
          Serial.println("Turn on the cooler at full speed.");
          return;
        }
        if((iTemp) >= SetTemp){
          if(bInRange==0){
            bInRange=1;
          }
        }else{
          if(bInRange==1){
            bInRange=0;
          }
      //PID subroutine
        float err = iTemp - SetTemp;
        s_integral += err*dt;
        float s_derivative = (err - previous_error)/dt;
        int U_in_ctrl = (K_P_ctrl*err + K_I_ctrl*s_integral +
K_D_ctrl*s_derivative);
        previous_error = err;
          if (U_in_ctrl<=255){
             if (U_in_ctrl > 0){
                analogWrite(relay, U_in_ctrl);
             }
             else
             {
                analogWrite(relay, 0);
             }
```

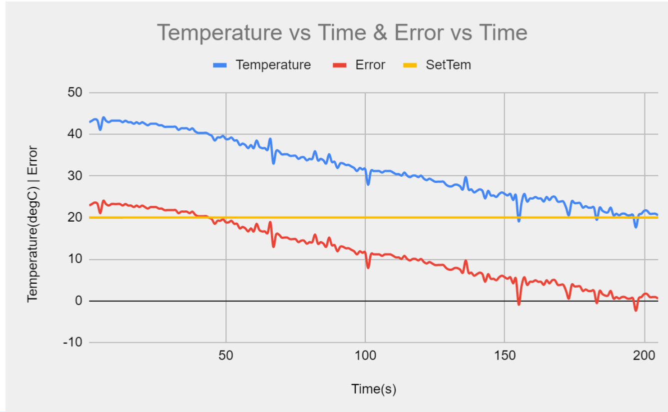
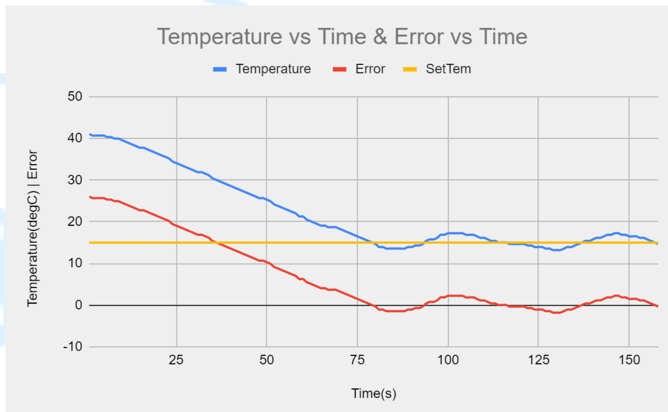
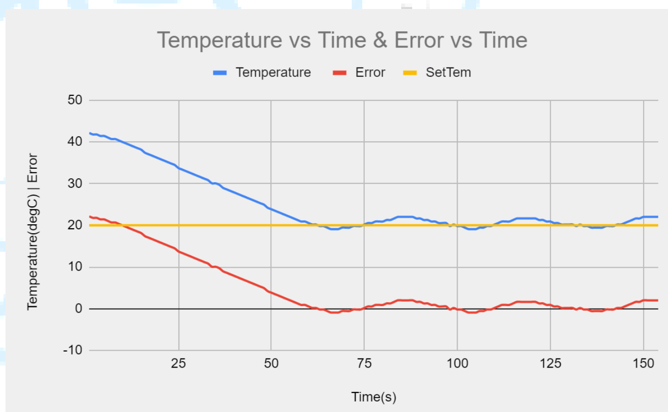
```
            }
            else{
               analogWrite(relay,255); // relay is turned on
            }
    }
```

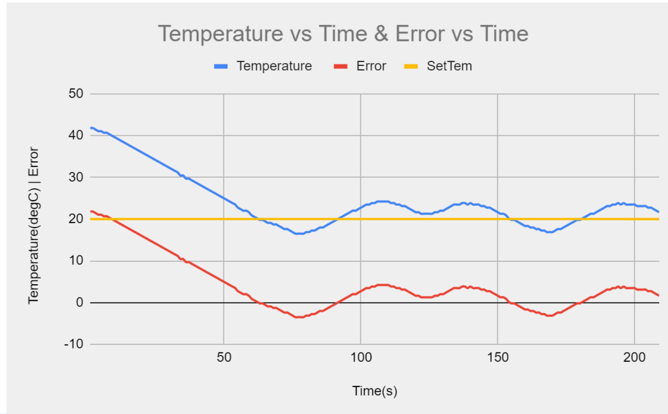
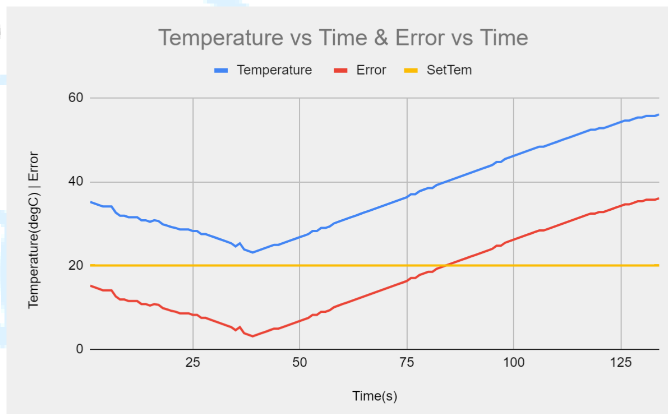
The core part of our code is the void Control PID() function that we have implemented in our system. This function's input is the temperature sensed by the sensor. Here, we have used the range concept in the PID algorithm. After defining the set temperature, we have bounded it in some range between maxTemp and minTemp. So, if the current temperature is greater than the maxTemp and less than the minTemp, then the Peltier will be turned on and off respectively by giving direct PWM to the Peltier through the motor driver. Suppose the current temperature is in the range. In that case, it will go into the PID subroutine, whereby calculating the error, it gets into the PID control algorithm, where the output from that will be the total error due to contributions from PID. This value is the controlled output from the subroutine according to the requirement for the Peltier to be turned on and off. Then, the motor driver will give the PWM-controlled voltage to the heat sink unit and Peltier in the same amount as the governing PID error. This is how the temperature is being controlled [10][11][12][13].

8 Key Results

	Experiments	Results
1	Initial Temperature = 28℃	
2	Initial Temperature = 45℃	

<table>
<tr><td>3</td><td>Initial Temperature = 56℃</td><td>
</td></tr>
<tr><td>4</td><td>By Increasing Range</td><td>
</td></tr>
<tr><td>5</td><td>By Decreasing Range</td><td>
</td></tr>
</table>

6	Copper Plate	
7	Set temperature = 15℃	
8	Value of I is small	

9	Value of I is Big	
10	Temperature change of Peltier without Heat Sink	

9 Comparison and Explanation of Results

A total of 9 different readings were taken with changing various parameters and conditions.

- For the first 3 experimental readings, the starting temperature was changed to values: 28℃, 45℃, and 56℃ respectively. In all these cases, the activation/deactivation temperature is ± 3℃ from the set temperature.
  - We can see that the time taken for each case to get to 20℃ is different, with the third case taking the most time(dipping below 20℃ after 104s), the second case taking 83s, and the first case only taking 25s. This is to be expected.
  - The rate of change of temperature (ΔT/t) is almost the same for the first two cases(≈0.3℃/s). But for the third case, ΔT/t ≈ 0.44℃/s. If we were to assume that the Heat Transfer Rate(Q) of the Peltier does not vary greatly within the given temperature range, we have to assume that some of the heat energy within the fins has dissipated into the atmosphere as radiation.

- But for the first two cases, since they were relatively closer to each other in temperature, the heat lost to the surroundings should be almost at the same proportionality that the Peltier cooler was taking away heat from the fins, i.e, since Radiation $\propto T^4$, the third case with a higher temperature would dissipate more heat into the surroundings.
- Also, the curve in the graph for the third case appears to have lower fluctuations. But this should be due to the fact that the temperature values in the third case are higher, so the fluctuation appears smaller in the graph.

- In the 4th and 5th experiments, the activation/deactivation temperatures are respectively set at ±5℃ and ±1℃ respectively.
  - In the 4th experiment, as the temperature of the fins falls to 20℃, the Peltier cooler cuts off when the temperature gets below 15℃ and the Peltier cooler will start working when the temperature hits 25℃. These values are 19℃ and 21℃, respectively, for experiment 5.
  - As we can observe, the oscillations once we reach the Set Temperature of 20℃ are much smaller in amplitude in the case of the second case compared to the first. This is because the activation/deactivation is much more sensitive in the latter experiment. Thus, we can infer that the lower difference in activation/deactivation temperature is much better at maintaining a steady temperature.
- In the 6th experiment, a copper plate was also introduced on the fins.
  - As we can observe from the graph, it takes almost 200s for the system to reach the set temperature.
  - The rate of change in temperature $\Delta T/t = 0.11$℃/s. This is because the heat transfer rate Q is not affected. Thus, it takes more time to transfer the heat as more material is in the system, assuming the Q of the Peltier Cooler does not vary greatly.
  - The equation is as follows
  - $Q = (m_{fins} * c_{fins} + m_{copperplate} * c_{copper})\Delta T$
- In the 7th experiment, we changed the Set Temperature to 15℃, with activation/deactivation at ±3℃ from the set temperature.

- Rate of change of temperature $\Delta T/t \approx 0.33$℃/s, as it took 80 seconds to drop from 41℃ to 15℃. The oscillation amplitudes seem similar to that of the 1st experiment, as both have ±3℃ activation/deactivation temperature.

- In the 8th and 9th experiments, the K_p was changed with 0.001 in the first case and 0.01 in the second case.
  - As we can observe, the oscillations are a lot smaller in amplitude when K_p value is 0.001 when compared to 0.01. Thus reducing by K_p, the transient response time of the setup is improved and allows the setup to get much better at maintaining a steady temperature.
- In the 10th experiment, the heat sink was removed from the setup. We can observe that the temperature of colder surfaces started decreasing, but after some time, it started increasing gradually. We have done this observation in 2 other Peltier, but the result was the same. This happens because we have not insulated the four sides of Peltier, and the hotter surface temperature is higher than the colder surface. Without sufficient insulation, it can affect the colder surface; therefore temperature goes higher.

10 Summary and Conclusion

Based on the research conducted, it is found whatever we designed could control the surface temperature of thermoelectric coolers satisfactorily. The testing phases have been carried out to examine and ensure that the units function properly. An Arduino Serial Monitor acquired the real time data of the humidity and Peltier surface temperature. The accuracy and stability of the experimental setup for characterizing temperature and humidity control depended on the environment. The results demonstrated that controlling the surface temperature using Arduino is functioning as per the intended design, and meeting the overall project objectives.

The observed controller's response parameters demonstrated that the on/off controller responds faster but incurs oscillations with the overshoots. In the PI-controller, increasing the integral gain while keeping the proportional gain constant, we observed that the controller responds faster but with increments in the occurrence of overshoots too. The appropriate choice of the proportional and integral gains leads the system to settle with very small steady state error values and smaller overshoots.

We have constructed graphs using the data from various experiments to obtain a comprehensive understanding of the changing system parameters on the system. Thus, we can conclude that the time response plots indicate that the temperature fluctuations are within a small range 1°C, which is adequate for the experimental accuracy for the study of temperature control of surfaces. The project is successfully implemented within a timeframe.

11 Critical Analysis

11.1 Innovation

At first, using the relay module to perform this experiment presented some challenges. The frequency of the signal—specifically, how frequently the relay is turned on and off—influences the temperature sensor reading. Thus, in order to report the actual Peltier surface temperature, the temperature sensor also emits a random reading. We replaced the relay module with a Motor Driver to address this issue. A motor driver's main advantage over a relay is that it offers a PWM signal output, which enables digital writing using digital pins on Arduino boards. As a result, the temperature sensor could operate accurately without any problems. Also, the Peltier rating is 92 W, and the 12 V from SMPS. Thus, we need around 7.67 Amps, and the Motor driver is rated 30 V and 10 Amps, which is good for the Peltier rating.

The error is the difference between the observation and the set temperature, as we have previously discussed in the temperature vs. time and error vs. time graphs. Moreover, the error keeps getting smaller the further you go through time. Initially, an error was close to 10, but as you go further, the error gets smaller. The error increases when we attempt to raise the temperature while maintaining the set temperature for a new experiment. But as time passes, it gets closer to zero. Furthermore, it requires more time than before to approach close to zero.

As part of the innovation, we analyzed the system for several other experiments:

1. Humidity experiments (failed due to unavailability of the chamber)

2. Experiments to determine the effect of heat sink on the Peltier element (observed abrupt temperature changes, explanations given in the table above).
3. Experiments to show the heat transfer between the fin array and a metal piece kept on it.
4. Apart from these we have also experimented with increasing/decreasing the binary range, values of I and the Set temperature values.

11.2 Shortcomings

In the humidity experiment, we did not get the appropriate results because the humidity sensor only sensed the influence of fins and Peltier surfaces in the open air. If it were covered in the close chamber, we would have gotten the desired readings, but because of time and resources constraints, we could not make one.

Another issue was faced while tuning the PID controller. To match the dynamics of the process to be regulated, it must be tuned. The default values that designers use for the P, I, and D terms often don't produce the intended performance and can even cause instability and sluggish control operations. The PID controllers may be tuned using a variety of techniques, and the operator must pay close attention to choose the appropriate proportional, integral, and derivative gain values. In our experiment, we tried to match the tuning to achieve an optimal outcome, but the value we got was approximate because of experimental errors. However, in the overall results, this approximation doesn't affect greatly.

11.3 Challenges

We encountered various obstacles in the completion of this project. These are listed below.

- The tip of the NTC thermal sensor was round, so it was quite hard to adjust it to the fins of the thermoelectric Peltier cooler.
- The thermoelectric Peltier cooler we used at first showed abnormal behavior as it heated on both of its surfaces, while one of the surfaces should have been hot and the other one should have been cold.
- The digital thermal sensor did not show the accurate temperature of the surface. It also had a certain lag time to display temperature, which caused some errors.

- Relay works only with digital inputs and outputs. On the other hand, our input sensor generates continuous and analog signals. Thus, using a relay in our setup would not have worked as the frequencies of the input signal and the input for the relay would not have matched, and we would not have acquired the desired results. Instead, we required something that would have given continuous outputs in direct Pulse Width Modulation (PWM) form, not in digital form. There was also a problem with using the relay in the system. The relay's response time is quite more compared to the sensor analog input.
- Due to the starting temperature, the system had a minor delay in getting into a stable state.

12 Future Work

- The use of An NI Labview module or Agilent DAQs (a parallel validation sensor) will prevent benchmarking or commenting on the response time compared to state-of-the-art systems.
- The use of a humidity chamber would provide better humidity control.
- Tuning the PID algorithm using other advanced methods to make it more robust for the setup:
    1. MATLAB Auto Tuning using the Simulink feature on MATLAB by mathematically modeling the circuit. The first step would be to model the plant transfer function.
    2. Root Locus Method: A manual graphical pen and paper tool to find the values of P, I, and D to obtain the desired performance.

13 Acknowledgement

In the accomplishment of completion of our project on **'Arduino based Temperature Sensor'** we would like to convey our special gratitude to Mr. Soumyadip Sett, Assistant Professor, Mechanical Engineering at IIT Gandhinagar. Your valuable guidance and suggestions helped us in various phases of project completion. We will always be thankful to you in this regard. We would also like to thank our Teaching Assistant Aaqib Khan for their continuous motivation and support. We would also like to thank the team of Maker Bhavan for letting us build our project in it and also providing us with the tool that would be necessary for successful completion of the project.

14 References

[1] *Why temperature control is essential to the manufacturing sector*. Askews Refrigeration Services. (2020, March 10). Retrieved November 26, 2022, from https://www.askewsltd.com/blog/why-temperature-control-is-essential-to-the-manufacturing-sector/

[2] J, P. A. (2022, November 26). *Different applications and importance of temperature control units across industries*. Temperature Control Units: The Most Essential Component for Successful Projection of Industrial Operations. Retrieved November 26, 2022, from https://www.researchdive.com/blog/different-applications-and-importance-of-temperature-control-units-across-industries

[3] Rao, G. K., Subramanyam, M. V., & Satyaprasad, K. (n.d.). *Study on PID controller design and Performance Based on tuning techniques*.

[4] H.L. Tsai, J.M. Lin, Model building and simulation of thermoelectric module using matlab/simulink, J. Electron. Mater. 39 (2010) 2105–2111.

[5] N. Wang, M.M. Chen, H.Z. Jia, et al., Study of voltage-controlled characteristics for thermoelectric coolers, J. Electron. Mater. 46 (2017) 3097–3102.

[6] R. Hans, S.C. Kaushik, S. Manikandan, Experimental study and analysis on novel thermo-electric cooler driven by solar photovoltaic system, Appl. Sol. Energy. 52 (3) (2016) 205–210.

[4] Gupta, R., Das, C., Roy, A., Ganguly, R., & Datta, A. (n.d.). *Arduino based temperature and humidity control for condensation on ...* IEEE Xplore. Retrieved November 26, 2022, from https://ieeexplore.ieee.org/abstract/document/8405062

[5] Mota L, Mota A, Pezzuto C, Carvalho M, Lavorato M, Coiado L, Oliveira E. Development of a Surface Temperature Sensor to Enhance Energy Efficiency Actions in Buildings. Sensors. 2018; 18(9):3046. https://doi.org/10.3390/s18093046

[6] Abdelkrim Kherkhar, Younes Chiba, Abdelhalim Tlemçani, Hayati Mamur, Thermal investigation of a thermoelectric cooler based on Arduino and PID control approach, Case Studies in Thermal Engineering,Volume 36, 2022, 102249, ISSN 2214-157X.

[7] Zhenlin Chen, Zhongquan Sun, The principle and application of semiconductor refrigerator, Journal of Micro electronic technology (5) (1999) 63–65.

[8] D. Guo, Q. Sheng, X. Dou, Z. Wang, L. Xie, B. Yang, Application of thermoelectric cooler in temperature control system of space science experiment, Appl. Therm. Eng. 168 (2020), 114888.

[9] Incropera, F. P., DeWitt, D. P., Bergman, T. L., & Lavine, A. S. (1996). *Fundamentals of heat and mass transfer* (Vol. 6, p. 116). New York: Wiley.

[10] *Arduino PID temperature controller using MAX6675 K-thermocouple to Digital Converter IC*. Circuit Digest - Electronics Engineering News, Latest Products, Articles and Projects. (2021, November 1). Retrieved November 26, 2022, from https://circuitdigest.com/microcontroller-projects/arduino-pid-temperature-controller
[11] Bista, D. (2016). *VCU scholars Compass | Virginia Commonwealth University Research*.
[12] R. Kumar, S.K. Singla, V. Chopra, Comparison among some well known control schemes with different tuning methods, Journal of Applied Research and Technology, Volume 13, Issue 3, 2015, Pages 409-415, ISSN 1665-6423
[13] Agarwal, T. (2021, November 22). *PID controller : Working, types, Advantages & Its Applications*. ElProCus. Retrieved November 26, 2022, from https://www.elprocus.com/the-working-of-a-pid-controller/#:~:text=PID%20controllers%20are%20classified%20into,controller%20to%20regulate%20the%20method