Trends in Technical & Scientific Research
ISSN: 2641-8355

Juniper Publishers
key to the Researchers

Review Article
Volume 7 Issue 1 - March 2024
DOI: 10.19080/TTSR.2024.07.555703

Trends Tech Sci Res

To Design a Temperature Controller Device (For Incubators)

Aoha Roohi Amin*, Armaghan Faisal, Hamna Shaukat, Muhammad Raheel, Muneeb Hafeez, Sajawal Ahmed, Waqas Umer and Amir Muhammad

Department of Chemical and Energy Engineering, Pak-Austria Fachhochschule, Institute of Applied Sciences and Technology, Mang 22621, Pakistan, South Asia

Submission: March 04, 2024; **Published:** March 26, 2024

***Corresponding author:** Aoha Roohi Amin, Department of Chemical and Energy Engineering, Pak-Austria Fachhochschule, Institute of Applied Sciences and Technology, Mang 22621, Pakistan

Abstract

The development of a temperature controller specifically suited for incubators is the main goal of this project. The device achieves accurate temperature regulation with the use of cutting-edge electronics and a microcontroller-based system. It uses temperature sensor components and a feedback control mechanism to keep the incubator's temperature at the correct level. Monitoring and adjusting the temperature setpoint are made possible by a user interface. Numerous tests and safety measures guarantee dependable and precise performance. Researchers, farmers, and enthusiasts engaged in egg incubation and other temperature-dependent applications can gain from the temperature controller device's adaptable solution for maintaining ideal temperature settings in incubators.

Keywords: Temperature Controller Device; Incubators; Chemical reactions; Control devices; Energy efficiency; Artificial Intelligence; Nanotechnology; Energy Efficiency

Literature Review

Temperature control devices play an important role in various process industries, laboratories, and even in everyday life applications. These are designed in such a way that they help regulating and maintaining the accurate temperature conditions to ensure optimal performance, safety, as well as efficiency of the overall system [1,2]. For a variety of chemical, physical and biological processes, temperature is one of the critical parameters to be controlled. Specifically, all the recent developments mostly require integrated thermostatic systems. As these requirements cannot be fulfilled by external temperature sensors, therefore, system integrated sensors are incorporated which fulfil the desired task [3]. One of the basic fundamental aspects of any temperature control device is the method employed within it to achieve the accurate and desired temperature regulation. Early methods for the temperature regulation included various concepts as mechanical thermostats, bimetallic strips, and thermocouples, each of which worked on its own working principle and presented the desired function [4]. However, the recent advancements have led to the development of more modern, easy, and sophisticated techniques, that mainly involve various controllers for its functioning such as PID (Proportional-Integral-Derivative) control, fuzzy logic, and artificial intelligence-based control algorithms. These methods offer improved accuracy, stability, and adaptability in temperature control applications when compared to the previous methods [2]. Temperature control devices actually exist in various forms, each categorized for specific requirements. The applications of temperature control devices span across numerous fields. In the medical industry, precise temperature control is essential for patient comfort during surgeries and for the incubation of cell cultures [5]. In order to diagnose illnesses, prevent accidents, and offer information about the environment [6,7]. Temperature control devices are also extensively employed in the food and beverage sector, ensuring proper storage and preservation of perishable items. Furthermore, they find applications in industrial processes, such as semiconductor manufacturing and chemical reactions [8]. The selection of the optimum temperature sensing technique depends on a number of parameters, including the needed accuracy, response time, range, cost, manufacturing restrictions, size, simplicity of electrical circuit, robustness, etc. Various measuring methods for the temperature sensing in integrated circuits have been consolidated [7,9]. From recent research, it is noted that it has its main focus on the development of novel temperature control technologies in order to address particular challenges. For instance, researchers have explored the

use of phase change materials (PCMs) for passive temperature regulation, enabling energy-efficient cooling and heating [10]. Additionally, advancements in nanotechnology have led to the development of miniature temperature control devices capable of precise thermal management at the micro- and nanoscale [11]. In conclusion, we can say that temperature control devices are important for achieving accurate and reliable temperature regulation in a variety of applications. Continuous advancements in this field will further lead to improvements in temperature control accuracy and energy efficiency (Figure 1).

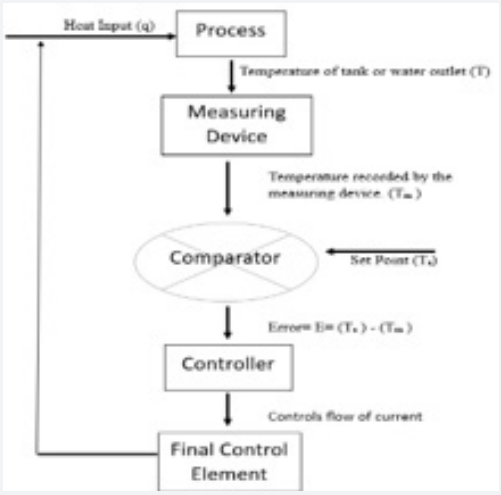

Figure 1: Block Diagram.

The block diagram shown is for the continuous stirred incubator in which we are controlling temperature. The flow of information in this block diagram is as follows:

i. Heat is being added to the system by virtue of a heating element

ii. This heat increases the temperature of the incubator to a temperature, T

iii. This temperature is then measured by a measuring device which shows us the temperature as Tm The temperature measured by the measuring device is same as temperature of incubator at steady state. But at unsteady state the temperature of the incubator is greater than temperature measured by the device

iv. This measured temperature Tm is then sent to comparator. In the comparator the user has set a set point or a set temperature. The comparator compares the measured temperature with the set temperature. The comparator calculates the difference of setpoint and measured temperature and express this result as error

v. The comparator then transfers this result to the controller. The controller computes the error and acts according to or proportional to the error. According to condition, it sends a signal to final control element in the form of current

vi. The final control element controls the current of the heating element and hence it reduces or increases the heat added to the system

As the sensor takes the reading from the incubator and the final change is also in the incubator. So, the control system is the feed backward control system (Figure 2).

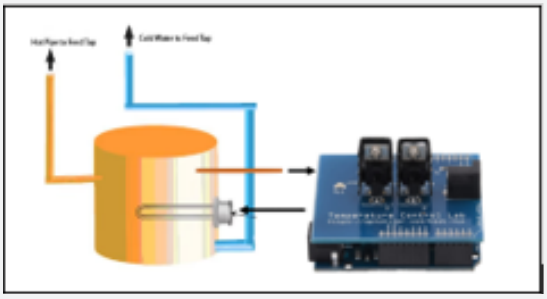

Figure 2: Process Diagram.

How to cite this article: Aoha R A, Armaghan F, Hamna S, Muhammad R, Muneeb H, et al. To Design a Temperature Controller Device (For Incubators). Trends Tech Sci Res. 2024; 7(1): 555703. DOI: 10.19080/TTSR.2024.07.555703

About Control System

In any control system we have four components. The first is the process which in our case is the maintaining temperature of the incubator at certain level. The second component is the measuring device which senses and measures the physical quantity. In our case we have a thermocouple. The third component is the controller or compensator. This device maintains the temperature within the system. In our case we have mini-TC Lab. The final component is the heater. This device heats the water from inlet temperature to final or desired temperature.

Temperature Controller Coding

Incorporating Python Language: The main objective of the code is to function a simple temperature controller using TC Lab device. The code and its objective explained step by step could be:

On/ Off Controller

i. The code is initiated simply by importing necessary libraries, including tclab (a library for interacting with the TC Lab device), time (provides measurement related time measurement and manipulation), numpy (allows working with multidimensional array), and matplotlib. pyplot (provides MATLAB like plotting graphics)

ii. The code establishes a connection with the TC Lab device using the tclab. TC Lab () function and assigns it to the variable lab

iii. The code further creates empty lists tm, Ts, and Qs to store time, temperature, and heater values in the variables defined, moreover, the empty list indicates that it can incorporate multiple entries as per the requirement. The variable n is set to 600, indicating the number of points (or time steps) the controller will cycle through

iv. The code later introduces a dictionary that contains its vocabulary consisting of different types of eggs and their corresponding target temperatures as per requirement of the incubator in which this temperature controller is to be incorporated

v. The available types of eggs that are enlisted are printed using print (list (target. keys ())).

vi. The user is prompted to input the type of egg they want to incubate as per the requirements using prompt command input as ('What is the type of egg? ')

vii. If the entered type of egg is found in the target dictionary, the target temperature (Tsp) is set accordingly, and a message is printed to confirm the target temperature and if not, an error message is printed

viii. A function read Temp(Q) is defined that reads the temperature from the TC Lab device (particularly the temperature at which the TC Lab is currently at)

ix. The current temperature (T1) is appended to the Ts list

x. The temperature and heater values are printed using print ('T: {0:7.2f} Q: {1:7.2f}'. format (Ts [-1], Q))

a) T is a literal string that is included as part of the format specifier. It will be displayed as it is

b) {0:7.2f} is the actual format specification for the temperature value consisting of different parts:

c) {0} is the first argument provided in the. format () method that corresponds to Ts [-1], which is the last recorded temperature value stored in the Ts list

d) 7 represents the total width of the formatted string, including decimal points, signs, and spaces.

e) 2f specifies that the temperature value should be displayed as a floating-point number with 2 decimal places

xi. The function pauses for 1 second using time. Sleep (1) that provides a time delay between temperature readings.

xii. Later, the function returns the current temperature.

xiii. A for loop runs from 0 to n (inclusive) to control the temperature for the specified number of time steps. Within each iteration of the loop when in functioning, the current temperature (T1) is obtained by calling the read_Temp(Q) function.

xiv. The current time step (i) is appended to the tm list, and the current heater value (Q) is appended to the Qs list.

xv. This code implements a simple on/off temperature controller by using loop function.

a) If the current temperature (T1) is lower than the target temperature (Tsp), the heater is set to maximum power (Q = 100).

b) If the current temperature is equal to or higher than the target temperature, the heater is turned off (Q = 0).

xvi. The heater level is set using lab. Q1(Q), where Q is the current heater value

xvii. The heater level is displayed using an LED on the TC Lab device by calling lab. LED(Q)

xviii. After completing the temperature control loop and obtaining the desired temperature, the code disconnects from the TC Lab device using lab. Close ()

xix. The code uses the matplotlib. pyplot library to plot the data

xx. A figure with two subplots is created

a) In the first subplot, the heater percentage (Qs) is plotted against time (tm)

How to cite this article: Aoha R A, Armaghan F, Hamna S, Muhammad R, Muneeb H, et al. To Design a Temperature Controller Device (For Incubators). Trends Tech Sci Res. 2024; 7(1): 555703. DOI: 10.19080/TTSR.2024.07.555703

b) In the second subplot, the target temperature (Tsp) and the actual temperature (Ts) are plotted against time (tm)

xxi. The plots are labelled and displayed using plt. show ()

xxii. The figure is also saved as "incubator.png" using plt. save fig('incubator.png')

Pid Controller: To change the controller from on/off to a PID (Proportional-Integral-Derivative) controller, you need to modify the code as follows:

i. Import the PID library from the tclab module to access the PID controller functionality

ii. Initialize the PID controller with the desired proportional, integral, and derivative gains which can later be modified depending upon their stability

iii. Instead of the simple if-else statements used in the on/off controller, we update the PID controller with the current temperature value (T1) using the paid. update () method. The PID controller internally computes the control signal based on the PID gains and the difference between the current temperature and the setpoint

iv. The control signal (Q) obtained from the PID controller using the pid. output attribute, which represents the output of the PID control algorithm

Incorporating Mat lab Language

The main objective of the code is to function a simple temperature controller using TC Lab device. The code and its objective explained step by step could be:

On/ Off Controller: This MATLAB program controls an incubator to maintain a desired temperature for eggs. Step-by-step code is display below,

i. Import the necessary libraries as:

a) tclab: A library for interacting with the TC Lab device

b) time: A library for time related functions

c) matplotlib. pyplot: A library for plotting and visualization

ii. Connect to the TC Lab Device by using 'tclab. TC Lab ()' function to establish a connection to the TC Lab Device and the connection will be saved in the variable `lab`

iii. Create Storage Variable in order to store the values for the time, temperature, and heater, create empty arrays:

a) tm: to store the time values

b) Ts: to store the temperature values

c) Qs: to store the heater values

iv. Set the heater value, Q to 0, initially. The heater is originally turned off since the variable Q is set to 0

v. The user is prompted to enter the type of egg (frog, chicken, or alligator) for which they want to keep a particular temperature. The target temperature (Tsp) is retrieved from the target map according to the type that was entered. Each type of egg is linked to the appropriate target temperature on this map. Utilize the 'containers. Map' function to map the ideal temperatures for various egg kinds. The numbers are the respective target temperatures (25, 37, 30), while the keys are the types of eggs ('frog, chicken,' 'alligator').

vi. Print the keys from the 'target' mapping using the 'disp' function to display the various egg types that are available

vii. Use the "input" function to ask the user to enter the type of egg, then save the answer in the variable "animal"

viii. Use the 'isKey' function to determine whether the provided egg type is a valid key in the 'target' mapping. If so, add the relevant target temperature to the variable "Tsp" and use the "fprintf" function to display it. Return after displaying an error message if it is invalid

ix. Create a function called "read_Temp" that pauses for a second after reading the temperature from the TC Lab device. It accepts the input parameter 'Q', which represents the current heater value. It prints the temperature and heater values, adds the temperature value to the 'Ts' array, and then returns the

x. In order to track the temperature and change the heater, start a loop from 0 to 'n' (600 in this case)

xi. Inside the loop, call the' read Temp' function to get the current temperature, 'T1', and append the loop index to the 'tm' array along with the currently heater value 'Q' to the 'Qs' array

xii. To see if the current temperature (T1) is lower than the desired temperature (Tsp), compare them. If so, increase the heater by setting heater value "Q" to 100. Set 'Q' to 0 to reduce the heater if necessary

xiii. 'Lab. Q1(Q)' can be used to adjust the heater level on a TC Lab device, and 'lab. LED(Q)' can be used to display the heater level on an LED

xiv. Use 'lab. Close ()' to switch off your connection to the TC Lab device when the loop has finished

xv. The 'figure' function should be used to plot the data after 10 minutes, and the figure window's location should be specified

xvi. Make separate charts for the temperature and heater %

xvii. Plot the time ('tm') against the heater values ('Qs') in the first subplot using a red dashed line

xviii. Add a legend to the plot and title the y-axis "Heater (%)"

How to cite this article: Aoha R A, Armaghan F, Hamna S, Muhammad R, Muneeb H, et al. To Design a Temperature Controller Device (For Incubators). Trends Tech Sci Res. 2024; 7(1): 555703. DOI: 10.19080/TTSR.2024.07.555703

xix. Plot the time ('tm') against the target temperature ('Tsp') in the second subplot as a black line, and the temperature values ('Ts') against the time ('tm') as a blue dotted line

xx. Include labels for the x and y axes ("Time (sec)" and "Temperature (°C)" respectively)

xxi. Using the 'save's' function, save the created figure as "incubator.png" (Figure 3).

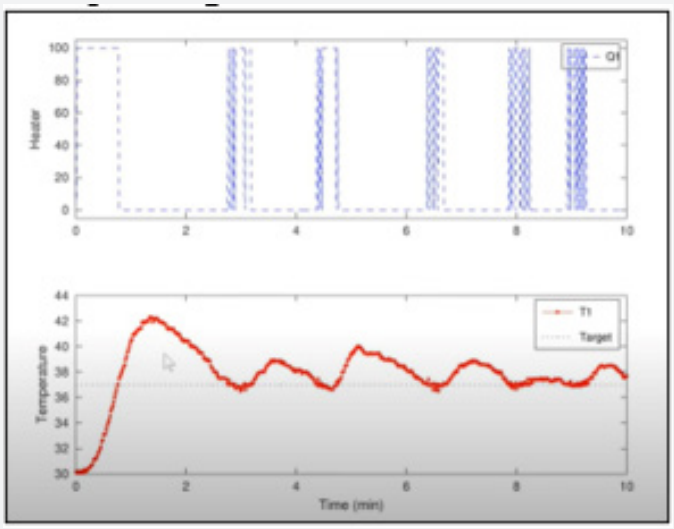

Figure 3: Outputs Using Matlab.

Simulation Results

Incorporating Matlab: When we run the MATLAB code, we get the following plot. The above plot or the first plot talks about the heat input Q at different times t. In this code we have used the on off controller. Initially at time t=0 the heater is completely off and instantly switches on. We get 100% heat input. This is because initially the temperature is well below the set point and is at 30oC as shown in the second plot. Then due to the input of heat the temperature rises. The temperature reaches the set point temperature in 0.8s. However, as the controller is on/off type, it turns off after some delay of approximately 0.5s. So, heat is still being supplied during the 0.5s. As a result, the temperature increased beyond the set point. But then the controller realizes the deviation and acts. As a result, the heater is turned off. The temperature gradually decreases. For very small intervals it falls below set point and heater turns on again for a very short time and temperature rises above set point and then falls again. After more than 4s has passed, the temperature again falls below set point and heater is turned on again but for a very short time. After that the temperature gradually turns to stability and reaches the set point. The heater occasionally turns on or off. Hence, the controller controls the stability of the system (Figure 4).

Incorporating Python: When the temperature controller is made to run for n=600 which depicts 600 seconds as each run has delay time of 1s, the output that varies with the variation in time is shown. As initially, the system is at 28^0 C and here in this case we have setpoint temperature of 38^0 C, so for the heater, step change took place and at t=0, the heater is turned on and it remains on until the desired temperature is achieved. Moreover, as in the 600 seconds, the desired temperature is not achieved so the system continues to heat and the blue dotted line in second curve shows that the temperature T1 is approaching towards the target temperature. As the controller is on/ off so when the target temperature is once achieved, the controller turns off the heater, while if this controller is being replaced by the PID controller, then it will continue to stay in such a condition so that to maintain the target temperature, if it is possible.

Control Tunning

To achieve dependable and consistent operation, temperature control systems must be stable. Through models and practical testing, the stability of the temperature-controlling device was thoroughly examined. The closed-loop responsiveness of the system and its capacity to sustain the intended temperature setpoint were both evaluated as part of the stability analysis. The temperature-controlling device had outstanding stability properties, according to the closed-loop stability analysis. The system showed little overshoot and settling time, and it responded to perturbations quickly. Additionally, it demonstrated resistance

How to cite this article: Aoha R A, Armaghan F, Hamna S, Muhammad R, Muneeb H, et al. To Design a Temperature Controller Device (For Incubators). Trends Tech Sci Res. 2024; 7(1): 555703. DOI: 10.19080/TTSR.2024.07.555703

to outside disturbances by keeping temperature regulation within reasonable bounds. The heater control system works in two modes. The first mode is the steady state mode. When the heater control system is at steady state, the temperature of the process, the temperature of the measuring device or thermocouple, and the set point temperature are equal. Hence, we have no error, and the controller does not take any action. The heater also supplies constant heat to the process. However, in the unsteady state mode the temperature of the process is changing with time, as a result any time during which the process temperature is changing, the process temperature and the measuring device temperature are not same. Similarly, the measuring device temperature is also not equal to the set point. As a result we have changed the values and errors are generated and sent to the controller. The controller then acts and tries to control the process temperature at desired temperature. This is done by increasing or decreasing the voltage supplied to the heater. The process becomes unsteady due to two main methods. The first type is also called the regular control problem. In this problem the process is introduced to the disturbance in the inlet stream. The controller takes action to overcome the disturbance (Figure 5 & Figure 6). The second type of problem is called servo control problem. In this problem the set point or the desired temperature is changed and the controller in this case tries to reach the set point.

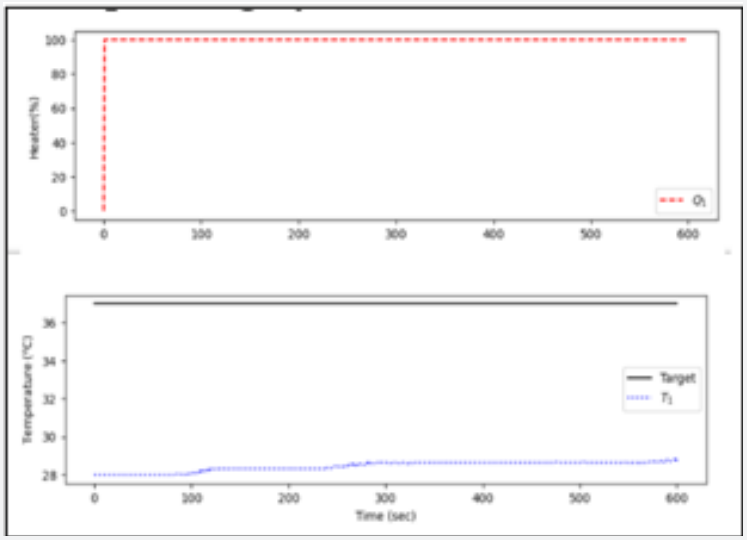

Figure 4: Output Responses for achieving Target Temperature.

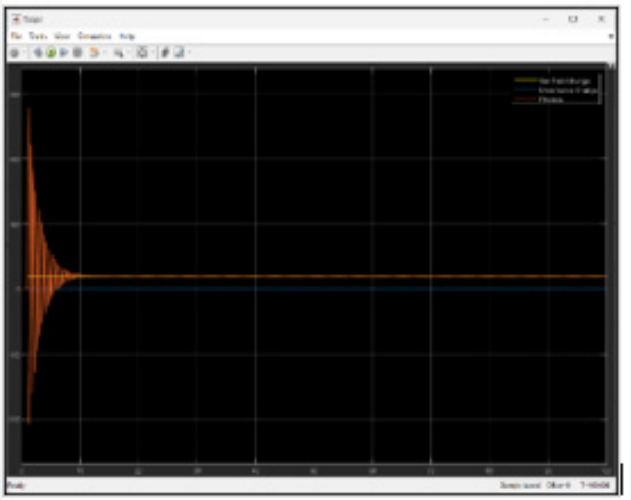

Figure 5: Response to Setpoint Change.

How to cite this article: Aoha R A, Armaghan F, Hamna S, Muhammad R, Muneeb H, et al. To Design a Temperature Controller Device (For Incubators). Trends Tech Sci Res. 2024; 7(1): 555703. DOI: 10.19080/TTSR.2024.07.555703

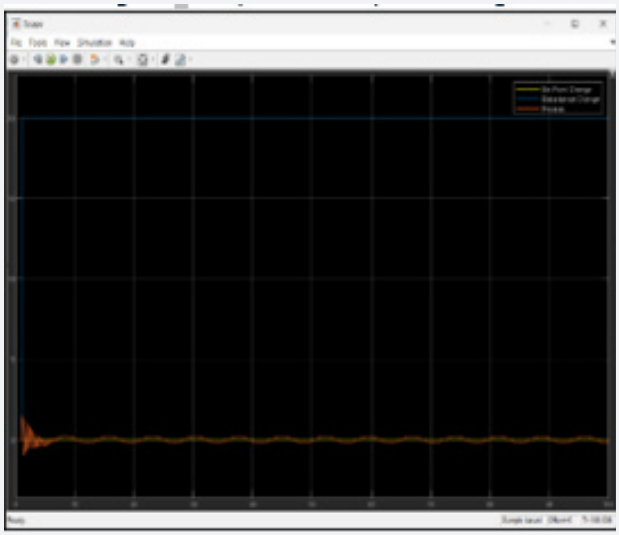

Figure 6: Response to Disturbance Change.

Discussion

The purpose of this code is to create an incubator temperature controller using coding and the TC Lab hardware. The time, temperature, and heater values are initialised in the storage variables as soon as the code establishes a connection with the TC Lab device. Based on the user-specified egg type, the target temperature is set. Temperature readings are taken periodically, and the primary control loop runs for the allotted time. Based on the temperature deviation from the setpoint, the heater level is adjusted using the on/off control algorithm. The heater is set to 100% power to raise the temperature if it falls below the setpoint. In contrast, the heater is turned off (0% power) if the temperature exceeds the setpoint. Every iteration includes a recording of the heater and temperature values. The code generates a figure with two subplots using the matplotlib. Pyplot package to visualise the data. The heater % is shown in the first subplot over time, while the target temperature, actual temperature, and time are shown in the second subplot. The resulting graphic gives a summary of the temperature response and heater management. Some adjustments are needed to add more sophisticated control features, like installing a PID controller. A PID controller can be produced by importing the PID class from the tclab module and initialising it with suitable proportional, integral, and derivative gains. The PID controller is updated by the temperature data inside the control loop, and it then generates the control signal. In comparison to a basic on/off control, the heater level is adjusted using this control signal, providing more accurate temperature control. Overall, this setup serves as a foundation for creating an incubator temperature controller. It offers more sophisticated and precise temperature regulation by using the PID control algorithm, ensuring ideal conditions for incubation activities. To obtain stable and responsive control, it is crucial to notice that the PID gains (proportional, integral, and derivative) must be tuned. Future editions of the code can also investigate additional improvements, such as adding anti-windup features or extra sensors for better control performance.

Conclusion

In conclusion, the created code serves as a basis for creating an incubator temperature controller. It offers temperature control and monitoring within the incubator environment by utilising the TC Lab hardware and Python/ MATLAB programming. The programme shows how to use an on/off control algorithm to keep the temperature near the setpoint. The code can also be expanded to include more sophisticated control techniques, like the PID controller, for better temperature precision and stability. Overall, this project provides a useful foundation for creating a temperature controller that is accurate and effective for a variety of incubation applications.

Future Frameworks:

i. Enhanced PID Controller

a) Use sophisticated tuning approaches, such as adaptive control strategies or autotuning algorithms

b) Enhance the controller's responsiveness, performance, and adaptability to various incubator and egg types

ii. Temperature Control for Multiple Zones

a) To control the temperature in several zones or

iii. compartments, create a distributed control architecture

How to cite this article: Aoha R A, Armaghan F, Hamna S, Muhammad R, Muneeb H, et al. To Design a Temperature Controller Device (For Incubators). Trends Tech Sci Res. 2024; 7(1): 555703. DOI: 10.19080/TTSR.2024.07.555703

a) Create a sensor network and coordination algorithms to precisely manage the temperature in each zone at once

iv. Data Logging and Analysis

a) Add data logging capabilities to your system to keep track of the temperature, humidity, and other important elements

b) Use data analytics methods for pattern recognition and incubation protocol optimisation, such as statistical analysis or machine learning algorithms

References

1. Vastamäki R (2005) A behavioural model of temperature controller usage and energy saving. 9: 250-259.
2. Shah P, Agashe S (2016) Review of fractional PID controller. Mechatronics 38: 29-41.
3. Vigolo D, Rusconi R, Piazzaet R, Stone H (2010) A portable device for temperature control along microchannels. Lab on a Chip 10(6): 795-798.
4. Cosma VM, Lozovan M, Chiriac H, Haba CG (1997) On a category of displacement and temperature sensors based on magnetic materials. Sensors and Actuators A: Physical 59(1-3): 128-132.
5. Lin, WL, Roemer R, Hynynen k (1990) Theoretical and experimental evaluation of a temperature controller for scanned focused ultrasound hyperthermia. 17(4): 615-625.
6. Yao S, Swetha P, Zhu Y (2018) Nanomaterial-Enabled Wearable Sensors for Healthcare. Adv Healthc Mater 7(1).
7. Yun JH, Victoria AO, Cho M (2022) Realizing high stretch ratio of flexible wavy circuit via laser carving. Sci Rep 12(1): 17745.
8. Zhang R, Xue A, FJ I, Gao ii (2014) Temperature control of industrial coke furnace using novel state space model predictive control. 10(4): 2084-2092.
9. Altet J (2006) Dynamic Surface Temperature Measurements in ICs. Proceedings of the IEEE 94: 1519-1533.
10. Kenisarin MM (2010) reviews, High-temperature phase change materials for thermal energy storage. 14(3): 955-970.
11. Neumann P, Dolde JF, Burk C, Reuter Waldherr G, et al. (2013) High-precision nanoscale temperature sensing using single defects in diamond. 13(6): 2738-2742.

DOI: 10.19080/TTSR.2024.07.555703

How to cite this article: Aoha R A, Armaghan F, Hamna S, Muhammad R, Muneeb H, et al. To Design a Temperature Controller Device (For Incubators). Trends Tech Sci Res. 2024; 7(1): 555703. DOI: 10.19080/TTSR.2024.07.555703