An Adaptive PID based Water temperature control of a Thawing Unit for Artificial Insemination of Livestock Animals

Mehul Shah[a], Amit Choksi[b], Rajendra Chhasatiya[c], Ritesh Prajapati[c], Dipen Parmar[d]

[a]Asso. Professor,GCET, EC Dept, [b]Assist. Professor,BVM, EC Dept.[c] System Level Solution, V.V.Nagar [d] Gavyam

Abstract

Thawing is the process to liquefy frozen or sex-sorted semen precisely at 37DegC precisely taken out from ~200DegC from Liquid Nitrogen Container during the artificial insemination process of livestock animals like cow and buffalo. Traditional method employed in the field does not guarantee temperature control of this process. Use of a Thawing unit is encouraged to improve the success rate of breeding livestock animals. This work presents a low cost and precise solution of design of a thawing unit. A prototype of the thawing unit is developed to control water temperature for liquefy frozen semen using Arduino Uno. Simple on-off control does not regulate water temperature at precise 37DegC, hence a PID control mechanism is used to provide automated temperature control. The prototype of thawing unit is tested with various PID parameters to see the temperature fluctuation with respect to set temperature of 37DegC. Experimental result suggests using an adaptive PID mechanism to control water temperature precisely at 37DegC instead of a basic PID control. With the use of adaptive PID control, water temperature of the thawing unit is controlled with the limit less than +-0.5 degree centigrade of set value temperature.

Keywords: Thawing Unit , PID Control, Temperature Control, Water Heater Design, Adaptive control

1. Introduction

Artificial Insemination (AI) in livestock animals is a technique in which sperm are collected from the male animal, processed, stored, and artificially introduced into the female animal's reproductive tract at proper time for purpose of conception [1]. Phenomenal growth of Artificial Insemination is due to improvement in fertility resulting from various methods [2],[3].

Thawing is the process to liquefy frozen or sex-sorted semen at 37DegC precisely taken out from ~200DegC from Liquid Nitrogen Container during the artificial insemination process of animals like cow and buffalo. Thawing semen in warm water (35-38 °C) for 40 seconds is the most used thawing procedure [4]. Use of a Thawing unit for controlling water temperature is the most effective way for the artificial insemination process. It equips users to set the correct temperature for the thawing process and can also display live temperature readings to minimize errors during semen thawing process. Various manufacturers have made such products, but none guarantees accurate temperature control of water while artificial insemination process. In this paper accurate temperature control of water with the help of proportional-integral-derivative (PID) method is implemented using Arduino Uno board.

Many researchers have applied PID control methods for patents, software packages and commercial hardware modules [5]. Authors in [6] has applied PID control of room temperature using microcontrollers. PID control methods are successfully applied to control temperature of water inside electric kettle for making tea and coffee[7]. Authors in [8] have applied PID temperature control using LABVIEW and Arduino and compared their method with on-off control. In [9] a PID based temperature of water dispenser is controlled using Arduino Uno and DS18B20 temperature sensor using Ziegler-Nichols tuning method. The experimental work in this paper uses an adaptive PID control method for precise temperature control of a thawing process used for AI.

To achieve precise temperature control, a PID control algorithm is implemented using Arduino Uno. A basic PID control algorithm is implemented to achieve precise water temperature control with different values of proportional, integral, and derivative parameters. An adaptive PID control algorithm is also implemented to achieve further accuracy of water temperature control within range of +-0.5 degrees of temperature.

Section II discusses heater design for a thawing unit and various issues related to its design. Section III presents basics of PID control, its implementation on Arduino Uno and two algorithm implementations on Arduino for precise temperature control. Section IV discusses various experimental results and analysis and section V conclude the paper.

2. Design of Heater for a Thawing Unit

In this section first various issues related to heater design are described and then heating time required for practical Heater design are described.

Once semen straws are injected for AI process from liquid nitrogen container at -200 degree Celsius, water temperature of a thawing unit momentarily reduced to very low temperature level. The thawing unit therefore uses a heater for keeping water temperature precisely at 37DegC for a pre-defined time of 30-40 second after insertion of semen straws.

An aluminum pipe is used as the core material of a heater in which the water is going to fill. Aluminum has high thermal conductivity and can be heated very fast. A nickel-chromium wire is used as a heating coil. Between the pipe and coil, an insulation made from Teflon is placed. Teflon paper is used between pipe and coil as insulation so that the current through the coil does not pass through the aluminum pipe and current leakage does not happen.

At the bottom of the pipe, the DS18B20 Temperature sensor [10] has been fixed to sense the water temperature for control purposes. The core functionality of the DS18B20 is its direct-to-digital temperature sensor. The resolution of the temperature sensor is user- configurable to 9, 10, 11, or 12 bits, corresponding to increments of 0.5°C, 0.25°C, 0.125°C, and 0.0625°C, respectively. The default resolution at power-up is 12-bit. The main objective of the heater is to supply heat to the water inside the aluminum tube and distribute it uniformly throughout its length.

Heat supply to the water is controlled through controlling current to the heater. An Arduino Uno is used to supply Pulse Width Modulated (PWM) current on one of its General-Purpose Input-Output (GPIO) pins as shown in fig-1. The heater has a maximum current carrying capacity of 2 Ampere, while Arduino can supply a maximum of 40 milliamp, therefore a TIP120 Darlington transistor is used as shown in fig-1 to increase the current capacity up to 2 amperes.

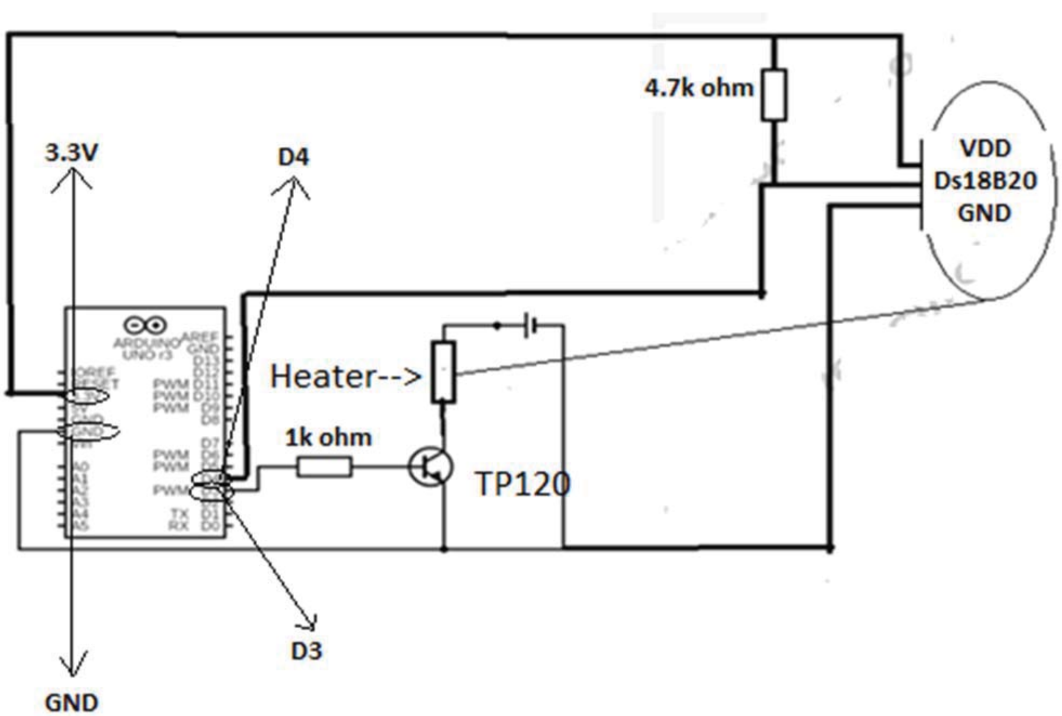

Fig. 1. Basic Circuit for Temperature control

2.1. Heater Design and Heating time calculation

Aluminum pipe is used as a core material of the Heater in which water will be filled. The pipe is 14 cm in length. Its outer Diameter is 20mm and inner diameter is of 15.5mm so thickness is of 2.25mm. Total volume of water inside the pipe is 25 ml. (volume = pi*r^2*length).
For this dimension of pipe, a 1.2-meter long 21-gauge nichrome wire is wounded as a heating coil, having a resistance of 3.1 ohms. 21-gauge nichrome wire has 2.6 ohm/meter resistance. So, 1.2-meter wire will have 3.1-ohm resistance.

To calculate the time required to heat 25ml of water from 19 degrees Celsius to 37 degrees Celsius using 21-gauge nichrome wire wounded on an aluminum pipe, following calculations are performed.

The heat Q required to raise the temperature of the water, Q = m•C•ΔT, Where m = mass of water, C = specific heat capacity of water and ΔT = Temperature difference. In this case, m = 25ml = 0.025kg (because the density of water is 1g/ml), C = 4.18 joule/g°C (Specific heat capacity of water at room temperature) and ΔT = 37-19 = 18°C
So, Q = (0.025kg)*(4.18 j/g°C)*(18°C) = 1.88kJoule

Now the power output of the heating coil P = VI, so P = (5.8) *(1.8) (in our case V=5.8, I=1.8) = 10.44W
Now practically the heating coil will lose some of this power to the aluminum pipe, So useful power P will be 6.264W, after assuming 40% of power dissipation.

Therefore, now the time t = Q/P=1.88kJ/6.264W=301 seconds to heat water pipe at level of 37 deg C.
Therefore, the heater will take approximately 301 seconds or 5 minutes to raise the temperature from 19 to 37 degree Celsius.

3. PID based Temperature control of the Heater

PID control is a feedback mechanism used in a control system. This type of control is also termed as three-term control and is implemented by a PID Controller. The proportional, derivative, and integral parameters can be expressed as Kp, Kd and Ki respectively. By these three parameters, how much a process variable (temperature for example) deviates from the desired set point value can be calculated and controlled. All these three parameters have effect on rise time, settling time, overshoot, and the steady state error of the process variable. Therefore, different control actions for specific work can be achieved with these parameter settings. For PID control the actuating signal at the input of a process block consists of proportional error signal added with derivative and integral of the error signal as shown in fig.2.

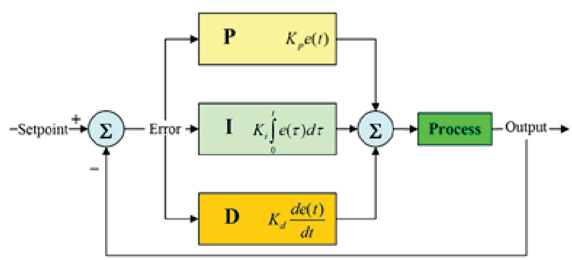

Fig. 2. PID Control block diagram

PID controller consists of three sub-units: an Arduino Uno R3; DS18B20 temperature sensor; and heater with transistor driver TIP 120 circuit as shown in fig. 3. The TIP120 acts like a switch, which is turned on and off by the PWM signal (Pin 3 of Arduino) at the base. A 1000-ohm resistor is added between the base of TIP120 and the PWM pin NO. 3. DS18B20 sensor read the temperature data from the heater. The power is provided to the circuit with the help of a +12V external power supply. The value set by push button is set by set value variable defined in the program in present implementation.

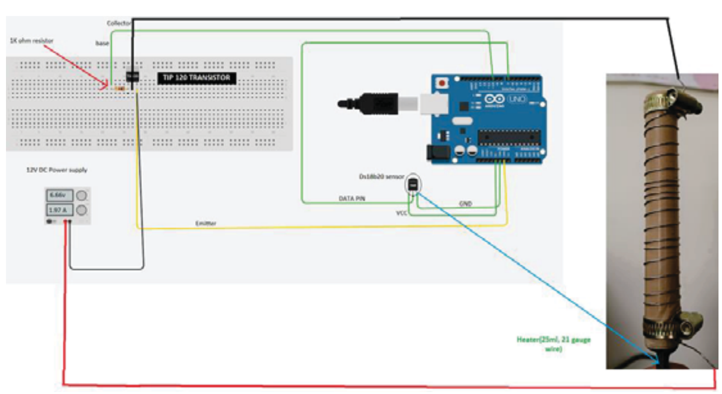

Fig. 3. Arduino based Temperature control of Heater

3.1 PID Control Algorithm

The PID temperature control works using a mathematical formula to calculate the difference between the current temperature and setpoint and then it tries to deliver the required power to ensure the target temperature remains constant. This also reduces overshoots that can be difficult to found in the traditional on-off control mechanism.

The operation of the PID based water temperature control system is as follows:

- The measured temperature with DS18B20 is compared with the setpoint temperature.
- The PID algorithm calculates the output signal based on the error between the setpoint and the measured temperature.
- The output control signal is sent to the PWM pin of Arduino, which controls the heater temperature.

A basic PID Control algorithm is used to implement a simple PID control strategy. Based on experimentation of this basic PID control with various Kp, Ki and Kd values an adaptive PID algorithm is also implemented to achieve precise water temperature control. Both of these algorithms are described next using pseudocode.

```
Algorithm 1: Basic PID Controller

    Require: Sampling_Time, Temperature_Set_point, kp, ki, and kd
    Start
    Set Error = 0
    Set Sampling_Time =0
    Set Sum_Error = 0
    Set Last_Error = 0
    Set Temperature_Set_point = 37
    Define kp, ki, and kd
    Pass the Temperature_Set_point, kp, ki, and Kd to the standard PID controller
    Find Sampling_Time = (Current_Time – Last_Time) * 1o^-3
    Compute Error = Temperature_Set_point – Past_Output
    Compute Sum_Error = Sum_Error + (Error * Sampling_Time)
    Compute Rate_Error = (Error - Last_Error)/Sampling_Time
    Set Last_Error = Error
    Set Last_Time = Current_Time
    Find Output = kp * Error + ki * Sum_Error + kd * Rate_Error
    Return Output
    End PID Control Process

Algorithm 2: Adaptive PID Controller

    Require: Sampling_Time, Temperature_Set_point
    Require: aggkp, aggki, and aggkd
    Require: conskp, conski and conskd
    Start
    Set gap = 0
    Set PID Output
    Set Input
    Set Temperature_Set_point = 37
    Compute Error = Temperature_Set_point – Input
    if (gap < 2)
    Compute PID Output using conservative conskp, conski and consKd
    else
    Compute PID Output using aggressive aggkp, aggki and aggkd
    end
    Return Output
    End Adaptive PID Control Process
```

4. Results and Analysis

An experimental setup as shown in figure 4 is implemented on Arduino Uno to control temperature of the heater using PID Control algorithm. Temperature of the water and current consumption data are collected using cool term software through UART interface of the board. Collected data are stored in csv files and then analyzed using MATLAB software.

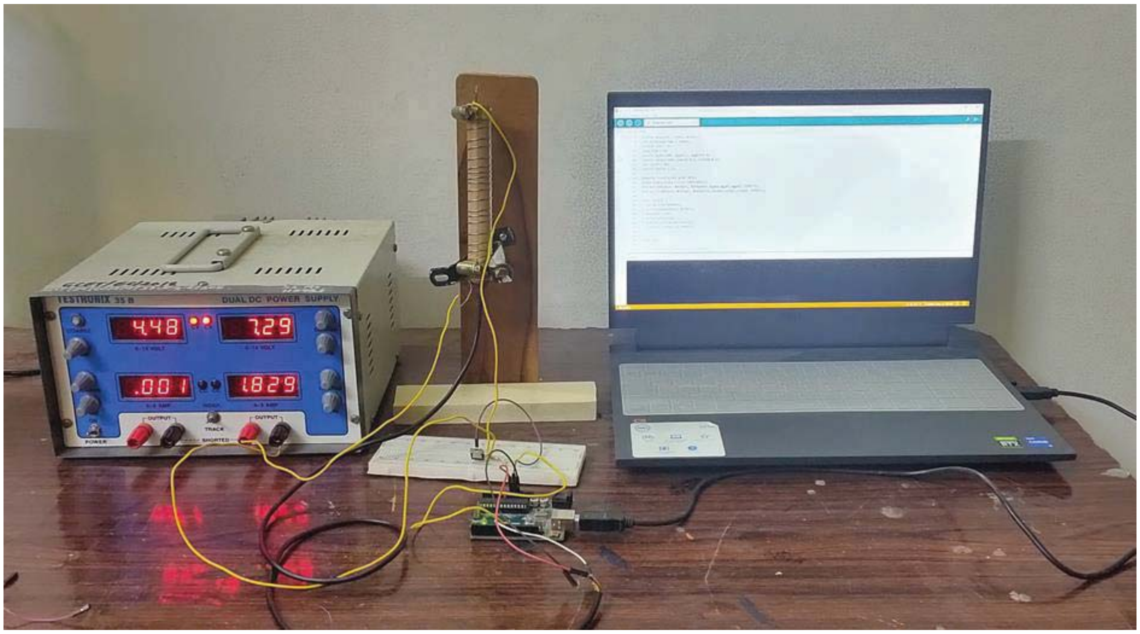

Fig. 4. Experimental setup for Temperature control

A basic PID Control algorithm is implemented first by varying different values of Kp, Ki and Kd, while keeping others values constant to see parameter's effect on rise time and over shoot values of the control temperature. Experiments are performed with below mentioned values of all the parameters.

- Values of Kp are changed from 400 t0 450 while keeping Ki at 5 and kd at 0.
- Value of Ki is changed from 2 to 50 while keeping Kp at 400 and kd at 0.
- Value of Kd is changed from 1.5 to 2 while kp and ki fixed at 600 and 8 respectively.

Results of above experiments are analysed with MATLAB on different CSV files of collected data and final summary of important observations are made in table-1.

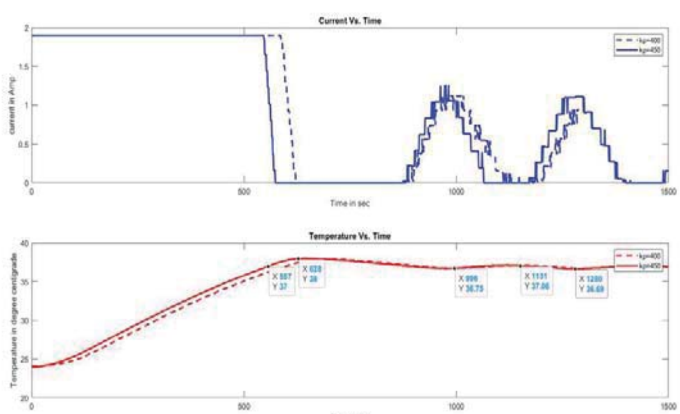

Fig. 5. Effect of increasing kp on rise time and overshoot (ki and kd constant at 5 and 0)

Experimental result as shown in fig. 5 suggest that as Kp values are increased, the rise time (Time to reach set temperature value) is reduced and first overshoot is marginally increased. This is in accordance to standard observation in the reference literature [5].

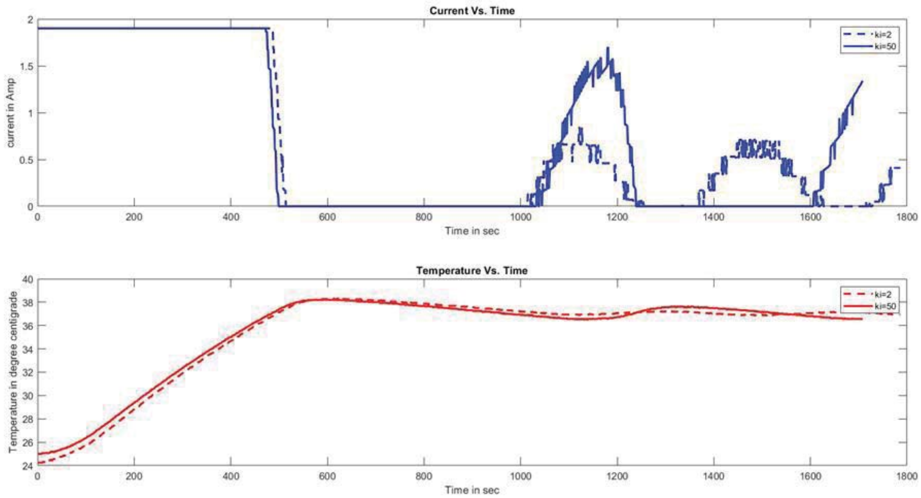

Fig. 6. Effect of increasing Ki on rise time and overshoot (Kp and Kd constant at 400 and 0)

Experimental result as shown in fig. 6 suggest that increase in ki value reduces rise time only marginally as compared to Kp. Overshoot has also increased marginally.

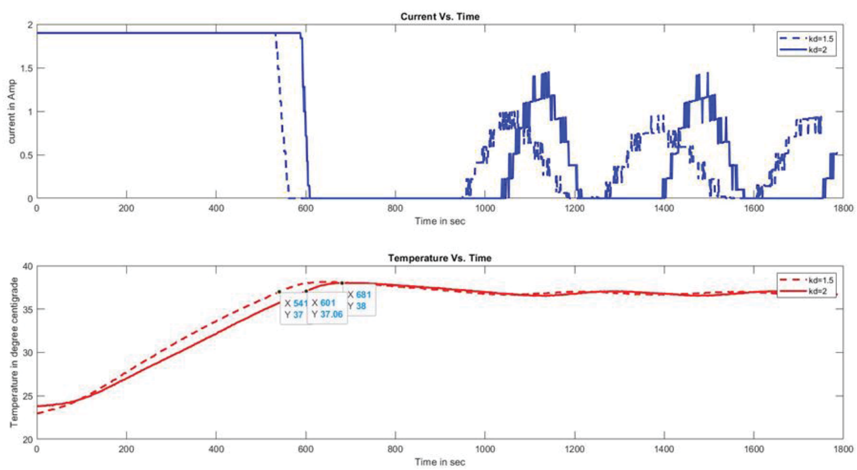

Fig. 7. Effect of increasing Kd on rise time and overshoot (Kp and Ki constant at 600 and 8)

As shown in fig.7 the overshoot has decreased.

Table 1. Summary of results plotted in fig. 5,6 and 7

Kp	Ki	Kd	Rise time	Maximum Overshoot	Maximum Undershoot
400	5	0	4:42 min	37.94 °C	36.75 °C
450	5	0	4:40 min	38 °C	36.75 °C
400	2	0	4.08 min	38.31°C	36.88°C
400	50	0	4.08 min	38.19 °C	36.56°C
600	8	1.5	4.36 min	38.13 °C	36.63-°C
600	8	2	5.02 min	38.00 °C	36.5-°C

Next an adaptive PID control is implemented for accurately controlling the temperature of the water temperature.

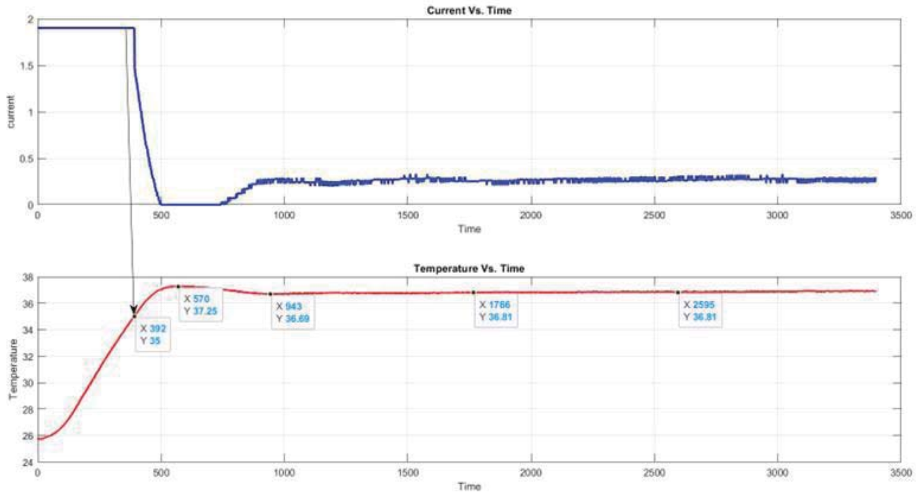

Fig.8 Adaptive PID control within Gap of 2degreeC

Experimental results suggest a precise control of temperature within the range of +- 0.5 degC.

5. Conclusion

Designing temperature control systems can be made easier with the assistance of Arduino Uno Hardware and software. Based on the experiments, it can be concluded that the system can maintain the temperature of water in the aluminum pipe at set level of 37 degrees and can maintain it stable with a range from 37.5 -36.5 degreec. Proper tuning of Kp, Ki and Kd values are crucial for accurate temperature of water at set value. Experimental result shows variation of Kp, Ki and Kd as per independent control of these parameters. However, for precise control of temperature with in a range of -+ 0.5 degC a use of Adaptive PID is required. The Kp, Ki and Kd parameters need to be fine-tuned for an optimum temperature control with both PID methods.

References

1. Foote, R. H. "The history of artificial insemination: Selected notes and notables." J. Anim. Sci 80 (2010): 1-10
2. Selk, Glenn. Artificial insemination for beef cattle. Oklahoma Cooperative Extension Service, 2004.
3. Yadav, Sarvajeet, et al. "Artificial Insemination in bovine and livestock management." 2022.
4. kumar Patel, Gaurang, et al. "Artificial insemination: A tool to improve livestock productivity." Journal of Pharmacognosy and Phytochemistry 6.6S (2017): 307-313.
5. Ang, Kiam Heong, Gregory Chong, and Yun Li. "PID control system analysis, design, and technology." IEEE transactions on control systems technology 13.4 (2005): 559-576.
6. Ranjan, Sudhir, Abhishek Sharma, and Puneet Chaudhary. "An effective temperature controller system using PID mechanism." 2014 Innovative Applications of Computational Intelligence on Power, Energy and Controls with their impact on Humanity (CIPECH). IEEE, 2014.
7. Shah, M. B. N., Zailany, N., Abidin, A. F. Z., Halim, M. F., Annuar, K. A., Azahar, A. H., & Yaakub, M. F. (2019). PID-based temperature control device for electric kettle. International Journal of Electrical and Computer Engineering, 9(3), 1683-1693.
8. Asraf, H. Muhammad, et al. "Development of experimental simulator via Arduino-based PID temperature control system using LabVIEW." Journal of Telecommunication, Electronic and Computer Engineering (JTEC) 9.1-5 (2017): 53-57.
9. Aisuwarya, Ratna, and Yulita Hidayati. "Implementation of ziegler-nichols PID tuning method on stabilizing temperature of hot-water dispenser." 2019 16th International Conference on Quality in Research (QIR): International Symposium on Electrical and Computer Engineering. IEEE, 2019.
10. Semiconductor, Dallas. "Programmable resolution 1-wire digital thermometer." Data Sheet DS18B20 (2002).

Acknowledgement

The authors would like to thank System Level Solutions and Gavyam Gentech Private Limited for helping this project work.