www.editada.org

International Journal of Combinatorial Optimization Problems and Informatics, 16(4), Sep-Dec 2025, 79-95. ISSN: 2007-1558.
https://doi.org/10.61467/2007.1558.2025.v16i4.1134

Design and Implementation of a Low-Cost Thermal Chamber with PID Control and Kalman Filter State Estimation

Julio Noel Pacheco Serrano [1], Alan Alonso Porras Martínez [1], Carlos Daniel Zamora Pérez [1], José Jaime Trejo Olguín [1], Adael Amaury Ángeles López [1], Evelin Gutiérrez Moreno[2], Amadeo Manuel Hernández Hernández[1]

[1] Departamento de Ingeniería Eléctrica y Electrónica, Tecnológico Nacional de México, Campus Pachuca, México.
[2] Departamento de Ingeniería Mecatrónica, Universidad Politécnica de Pachuca.
l22200615@pachuca.tecnm.mx, l222006@pachuca.tecnm.mx, l22200629@pachuca.tecnm.mx, l22200626@pachuca.tecnm.mx, l22200580@pachuca.tecnm.mx, evgutierrez@upp.edu.mx, amadeo.hh@pachuca.tecnm.mx

Abstract. An automatic temperature control system is an important application used in almost all modern gadgets, smart homes and industries; however, there are almost no available prototypes to practise with, and the ones that do exist cost more than an average student in Mexico can afford. This project is based on the implementation of a digital PID controller, realised through widely studied mathematical concepts, applied to a fan that blows air into an enclosed area housing an incandescent bulb, thereby controlling the internal temperature. A Kalman filter was used to reduce the thermocouple noise. The performance of the controller was tested by observing its ability to maintain the desired temperature setpoint steadily. This test highlighted the PID controller's ability to reach and maintain the desired temperature, validating its effectiveness as both an educational tool and a method of control for industrial applications.

Keywords: Arduino Uno, Temperature sensor, Air Temperature Control, Performance, Kalman filter

Article Info
Received May 09, 2025
Accepted July 2, 2025

1 Introduction

Within any applicable context whether educational, industrial, or commercial, the practical understanding of temperature control systems is important because it can be used to train users operating such systems. The acquisition of commercial equipment tends to be costly and inaccessible, limiting learning opportunities. For this reason, there is a need to develop an educational, economical, and functional model that allows users to experiment with and apply the principles of temperature control.

The proposed prototype uses an Arduino UNO to regulate the temperature within a confined space. It relies on inexpensive electronic components, including a K-type thermocouple, a MAX6675 module and a DC fan. A PID control algorithm, complemented by a Kalman filter, modulates the fan speed to achieve the desired temperature. This approach offers an affordable technical solution while providing scope for refinement and supporting educational development.

In the development process, it is possible to implement a single control mode. For example, Busu et al. (2023) describe an experiment in which a proportional (P) controller was tuned heuristically to select the optimal gain, followed by proportional–integral (PI) and proportional–integral–derivative (PID) controllers. Each controller was tested three times with the selected gains until satisfactory performance was achieved. To enhance robustness as the prototype evolves, an adaptive PID strategy should be adopted, as noted by Vázquez Leiva et al. (2024), because it provides greater tolerance to disturbances than conventional methods. Mathematically, the controller gains are denoted by KpK (Dorf & Bishop, 13th ed.). Proportional control stabilises the system against small disturbances, an integral term is required to eliminate steady-state error, and a derivative term further improves response (Canduela Ilundain, 2022). Even with all three terms, measurement noise can affect the output. A Kalman filter, which

estimates the system state in the presence of disturbance, can mitigate this. Using this approach, the prototype can iteratively correct errors over time. For example, Artieda Cruz (2021) reported an estimation error of ±1.34 °C with a simple sensor, demonstrating the Kalman filter's effectiveness.

Other researchers have achieved steady-state temperatures of 45.0±0.07 °C and 55.0±0.11 °C for low- and high-temperature hyperthermia experiments, respectively, with overshoots below 1.1 °C (Shikano, Tonthat & Yabukami, 2021). Final variations between 1.1 and 1.3 °C allow comparison with other sensors through manufacturer specifications. A crucial phase in development is system identification (MathWorks, 1999–2001), as the system's response reflects the proportional, integral and derivative parameters (López Sobrino, 2019), which are tuned manually for temperature changes. This characterisation enables comparison between the model and experimental results (Bañuelos et al., 2000) and allows real-time visualisation.

For benchmarking, the LM35 sensor—with an accuracy of ±1 °C and an output of 10 mV / °C over –55 °C to 150 °C (Coto Barriere, 2017)—was compared with the Type K thermocouple, which operates between 0 and 1 070 °C (Evolution Sensors, 2025). Maintaining a stable temperature is challenging (Pérez & López, 2017), and cooling rates often slow markedly towards the end of the process (Díaz-Montes et al., 2021). Temperature reduction depends on how the PID interacts with the setpoint (e.g. 28 °C) and therefore requires tuning of the proportional gain, integral and derivative times (Cruz, 2014). In addition, the number of PWM cycles applied to the fan must be considered, as these high- and low-signal pulses regulate ventilation in a semi-open system.

Some designs employ a proportional time-generator circuit built with an LM555 timer to produce a 5-second base pulse (Bañuelos et al., 2000), but digital implementation on an Arduino UNO allows easier adjustment. The module uses a 10-bit ADC to sample the output signal y(t)y(t)y(t), while an 8-bit PWM output acts as a DAC for the control signal u(t)u(t)u(t) (Sozański, 2023). Implementing the control in software makes it possible to display real-time results on an LCD in both Celsius and Fahrenheit (Raju et al., 2020).

Evaluating the cost-to-performance ratio underscores the prototype's educational value. Although sophisticated commercial modules exist, their high price limits their use in education (Alviz Blanco, 2022). The design emphasises a lightweight and compact structure, long service life, rapid thermal response, quiet operation, high reliability and low energy consumption, while avoiding environmentally harmful fluids (Kherkhar et al., 2022), making it well suited for teaching and experimentation.

Optimal control, derived from the calculus of variations applied to the Hamilton-Jacobi-Bellman equation, was implemented to regulate temperature within a finite volume place and was further compared with a PID control in order to assess the difference in energy consumption between the two, with the optimal control having a clear superiority enhancing thermal efficiency without compromising product quality (Calderon-Lopez et al., 2023).

A thorough explanation of the physical system is included, considering its structural design using prefabricated and recycled parts, as well as the electronic components required. In educational settings, practical understanding of temperature control systems is essential for the training of engineering students and those in related disciplines. However, acquiring commercial equipment for such purposes is often costly and inaccessible for many institutions. Therefore, there is a clear need to develop a didactic, cost-effective, and functional model that enables students to experiment with and understand the principles of temperature control in a practical and effective manner.

It is proposed that it is possible to develop a low-cost temperature control system based on an Arduino UNO platform, capable of maintaining the temperature inside a confined area at a desired value with acceptable precision, and that this system can serve as a suitable educational tool for students learning about control systems.

To address the identified problem, the development of a low-cost temperature control system is proposed, one that can maintain stable thermal conditions within an enclosed space, while also functioning as an accessible didactic tool for engineering students. To achieve this goal, a simple and functional physical structure using a wooden confined enclosure is to be designed, along with the selection and integration of low-cost electronic components such as a type K thermocouple and the MAX6675 converter for accurate temperature measurement. A PID control algorithm will be developed and implemented to regulate the speed of a fan based on internal temperature variations. A Kalman filter will be incorporated to enhance the reliability of the acquired data, an LCD display will be added to show real-time temperature, and finally, the system's performance will be evaluated through experimental testing to validate its effectiveness as a technical solution and educational resource.

To build a low-cost temperature control system that gives schools and universities a practical, easy-to-use tool for teaching the basics of control systems and let students, get hands-on with theoretical concepts, helping them develop a deeper and more applied understanding of how these systems work this affordable model makes learning about temperature control more accessible and relatable.

Additionally, it provides a detailed guide on the integration of sensors, actuators, and control algorithms within an open hardware platform such as Arduino, which may serve as a foundation for future developments and adaptations in similar projects.

First, the physical construction of the prototype is carried out, which includes designing the thermal chamber as a closed space, placing the incandescent bulb as the heat source, installing the 12-volt fan for air regulation, and incorporating a PVC pipe for internal air recirculation. To install the LCD display so the temperature can be checked quickly and without much effort. To connect the type K thermocouple to the MAX6675 module, which communicates with the Arduino UNO board that takes care of running the control algorithm. To finish the wiring at this point and check that everything works properly. To write the system's program by implementing a PID algorithm, adjusting its gains through trial and error, which might need several rounds of fine-tuning. To include a Kalman filter that helps smooth out the temperature readings and makes the system more stable. To set up the PWM control for the fan and arrange the data display on both the LCD screen and the serial monitor, making sure the information is clear and easy to follow.

Finally, the experimental testing phase is conducted, during which the system's behavior is evaluated under various temperature conditions. Timed tests are performed to observe the response speed to temperature changes, stabilization time, and control accuracy relative to the desired value. These results allow for an analysis of the system's performance and the identification of possible adjustments or improvements in its operation.

2 Methodology

In the present study, a prototype for temperature control is developed, designed to monitor and regulate thermal variations in a partial enclosure environment. For this purpose, a digital PID controller, programmed in Arduino IDE, is implemented, which allows a fairly accurate temperature regulation. Temperature readings are taken using a K-type thermocouple and a MAX6675 analog-to-digital converter, which allows a reliable reading. For thermal regulation, this system incorporates a 12 V DC fan, which adjusts the air flow according to the needs of the environment. This study has two main objectives: first, to optimize thermal control in semi-open systems, providing an efficient solution for industrial and scientific applications; and second, to develop an economical and accessible tool, which can be integrated in different types of environments that require temperature regulation.

The main objective of this project is to evaluate the response of the PID controller by verifying the accuracy of the K-type thermocouple sensor measurements, the performance of the control algorithm in temperature stability and tuning will be analyzed. This section provides an overview of the design development and construction of the prototype used for the simulation of a semi-open environment and the collection of captured measurement information. The prototype is designed to store inside a passive device that when turned on emits heat and generates a temperature rise inside the space. To regulate the temperature, a fan is driven by a control action processed and sent by a PWM, then passed through a power stage that allows the safe operation of the fan.

Based on these considerations, the most suitable components were selected for the correct operation and temperature measurement within our prototype, in addition to ensuring the reliability in the operation of the fan and the control and measurement devices. Due to this, the following components were selected:

1. Thermal chamber built with 3 mm MDF wood and dimensions 0.25 × 0.25 × 0.40 m. The box includes a 100 W incandescent bulb with his soquet, for the heat production.
2. Type K thermocouple sensor with amplifier module Max6675 compatible with Arduino for temperature measurement.
3. hd44780 16x2 LCD display with I2C module for the displaying of both current and desired temperature.
4. Generic Arduino UNO as the main controller of the system.
5. 12 VDC 8 W fan for air recirculation process.
6. 12 VDC power supply.
7. TIP41C transistor for power interface.
8. Computer for Arduino power supply, data logging, and input of the desired temperature.

The correct functionality of the prototype will allow accurate and reliable measurements, so it is designed to be almost airtight and allow air flow inside, since an external air inlet could cause alterations in the temperature control. That is why the fan is strategically installed next to the incandescent bulb that causes the temperature increase and, for the same purposes, the design has heat dissipation openings in the upper part of the prototype, which provide an outlet for the hot air by convection, which means that the air, when heated, becomes less dense and therefore rises above the cold ambient air, in addition to a circular flow provided by a structured piping system on the outside of the prototype, but maintaining the airtightness inside it.

The construction of this prototype begins with the construction of the base composed of 6 sides made of MDF type wood bonded with silicone sealant that ensures airtightness to prevent air leakage. Our passive component consists of a 100W incandescent bulb that is fixed, at the bottom of the prototype, on a ceramic cap that will isolate the heat generated by the bulb on the wooden base. Subsequently, a circular space is made where the fan will be installed, allowing the entry of cold air from the environment, thus regulating the temperature inside the prototype.

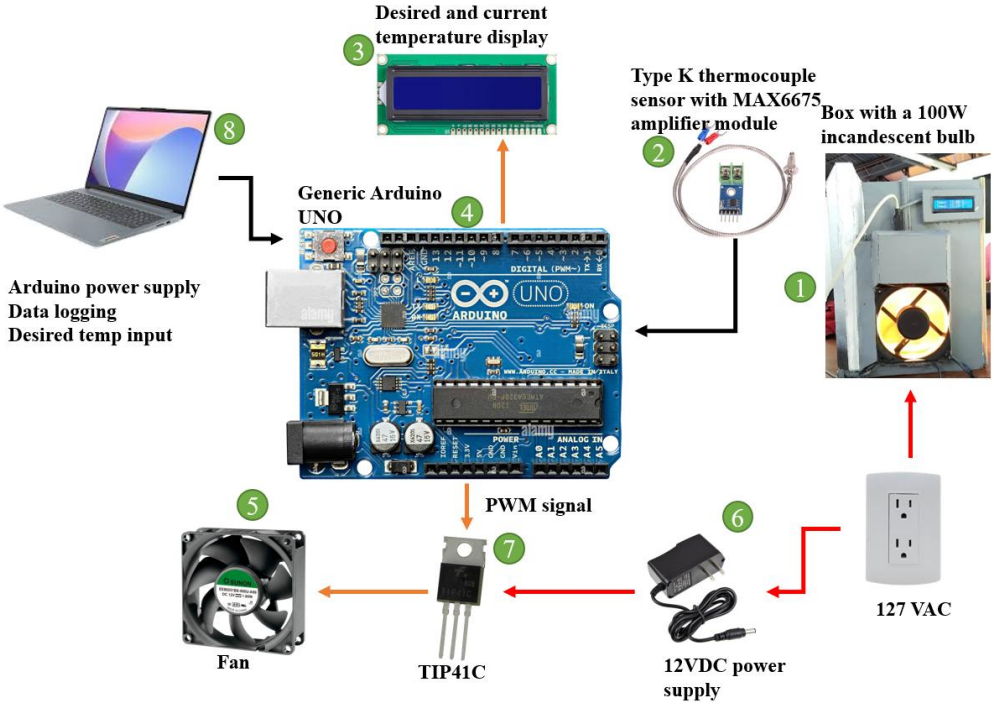

Fig. 1. Graphic diagram and interconnection of the components.

A piping system will allow the circular flow of air, following the laws of thermodynamics, the inlet and outlet of the piping system is installed in front of the fan, one inlet on the upper right side and the other on the lower left side on the same side of the fan, in this way the hot air rises and when it cools down it returns through the pipes at the bottom. In addition to this, heat dissipation openings are built in the upper part, which also allow the hot air to escape, and a partially polarized glass is installed in the front part to be able to visualize what is happening inside the prototype. Inside, at the top, is the type K thermocouple, and its connections go directly to the MAX6675 module that is connected and installed next to the Arduino, and an LCD display shows the measured temperature values in real time.

Fig. 2. Prototype temperature control of a semi-open system.

A 12 VDC eliminator is included, providing an adequate power supply to energize the elements of the system. The fan is connected to a control stage driven by a TIP41C transistor to establish communication between the Arduino program and the fan motor, ensuring a secure power supply with the necessary pulse width modulation.

Once all the physical elements are installed, the MAX6675 module is mounted and connected to the analog outputs of the Arduino. For the programming and calibration of the sensors, a PID controller was implemented in the Arduino. The proportional and derivative gains are selected with the objective of allowing a fast response to the system error; however, it is considered that it cannot be so high due to the possible oscillations that may occur. The gain of the derivative control is used to anticipate the variations and reduce the impulse; for this reason, a small value is selected, since it can amplify the noise and generate unwanted fluctuations. However, for the integral gain, a mathematical method known as "Trapezoid Rule" is used, which consists of finding the area under the curve of a signal, using trapezoids instead of rectangles, since this way gives a more approximate calculation. Similarly, a Kalman filter is programmed to suppress disturbances and allow accurate readings from the sensor.

Finally, calibration of the sensor is performed using a temperature measuring device as an exemplary parameter. In this case, a temperature measuring multimeter is used as the reference. The sensor's arduino library provides the possibility to set a fixed offset value to calibrate the thermocouple, simplifying the process. The PID controller (Proportional-Integral-Derivative) is widely used in process control. The combined action of these three control actions brings together the capabilities of each. The mathematical equation of the PID controller is:

$$u(t) = K_p e(t) + K_i \int_0^t e(t)dt + K_d \frac{de(t)}{dt} \tag{1}$$

The PID control action allows for the elimination of steady-state error by achieving optimal relative stability of the control system. Optimizing relative stability entails a transient response with short acquisition times, meaning a rapid convergence to the desired value, and a small overshoot without excessive peaks that could affect system performance. In the implementation of the integral component within the Arduino code, the trapezoidal rule was employed, which is a numerical integration method used to approximate the value of a definite integral and is based on the idea of approximating the area under the curve using trapezoids instead of rectangles, thus providing a more accurate estimate.

$$\int_0^t e(\tau)\,d\tau = \sum_{k=0}^{n} \frac{h\left(e(k)+e(k-1)\right)}{2} \tag{2}$$

To this end, the derivative is applied to both sides. On the right side, the derivative is utilized by definition using limits.

$$\frac{de(t)}{dt} = \lim_{h\to 0} \frac{e(k)-e(k-h)}{h} \tag{3}$$

The one-dimensional Kalman filter is a simplified version of the general Kalman algorithm. To derive the gain equations of the Kalman filter, statistical and mathematical concepts must be analyzed, which will be described below. The term Hidden State is defined as the internal condition of a system that cannot be observed or measured directly. Instead, it must be inferred from observable data through the application of mathematical models and estimation techniques.
The estimated state of the system (mean) can be expressed as

$$\mu = \frac{1}{N}\sum_{n=1}^{N} V_n \tag{4}$$

Variance is a measure of the dispersion of a data set concerning its mean,

$$\sigma_u^2 = \frac{1}{N}\sum_{n=1}^{N} (X_n - \mu)^2 \tag{5}$$

The standard deviation is the square root of the variance

$$\sigma_u = \sqrt{\frac{1}{N}\sum_{n=1}^{N} (X_n - \mu)^2} \tag{6}$$

To estimate the state of the system, multiple measurements are taken and averaged. At time n, the estimate $\hat{x}_{n,n}$ would be the average of all previous measurements.

$$\hat{x}_{n,n} = \frac{1}{n}\sum_{i=1}^{n} (z_i) \tag{7}$$

Where x is the true value of the system state (that is, the value of the temperature for our system), z_n is the measurement value at instant n, and $\hat{x}_{n,n}$ is the estimate of x that was made at time n (the estimate is made after taking z_n measurements). Since the dynamic model is considered to be constant, the estimate of the future state $n+1$ of x would be equal to the estimate of x at time n, that is,

$$\hat{x}_{n+1,n} = \hat{x}_{n,n} \tag{8}$$

The estimation is made at the moment n, immediately after obtaining the measurement z_n. Therefore, $\hat{x}_{n+1,n}$ is considered a predicted state. Now, to estimate $\hat{x}_{n,n}$, it is necessary to recall all historical measurements, so we only need to use the previous estimation and add a small adjustment, thus saving memory in the controller. For this purpose, the following mathematical operation is employed.
Knowing that the formula for the average is the summation of n terms divided by the number of measurements.

$$\hat{x}_{n,n} = \frac{1}{n}\sum_{i=1}^{n} (z_i) \tag{9}$$

We apply the summation of the $n-1$ previous terms plus the last measurement.

$$\hat{x}_{n,n} = \frac{1}{n}\left(\sum_{i=1}^{n-1} (z_i) + z_n\right) \tag{10}$$

We distribute $1/n$ and multiply and divide the first term by $n-1$.

$$\hat{x}_{n,n} = \frac{1}{n}\frac{n-1}{n-1}\sum_{i=1}^{n-1} (z_i) + \frac{1}{n} z_n \tag{11}$$

Rearranging reveals that the term $\frac{1}{n-1}\sum_{i=1}^{n-1}(z_i)$ is the average of the previous instant $n-1$.

$$\hat{x}_{n,n} = \frac{n-1}{n}\frac{1}{n-1}\sum_{i=1}^{n-1}(z_i) + \frac{1}{n}z_n \tag{12}$$

Rewriting the summation as its estimate, we obtain

$$\hat{x}_{n,n} = \frac{n-1}{n}\hat{x}_{n-1,n-1} + \frac{1}{n}z_n \tag{13}$$

Distributing the term $\frac{n-1}{n}$

$$\hat{x}_{n,n} = \hat{x}_{n-1,n-1} - \frac{1}{n}\hat{x}_{n-1,n-1} + \frac{1}{n}z_n \tag{14}$$

And finally, when reordered, we obtain

$$\hat{x}_{n,n} = \hat{x}_{n-1,n-1} + \frac{1}{n}\left(z_n - \hat{x}_{n-1,n-1}\right) \tag{15}$$

$\hat{x}_{n-1,n-1}$ is the estimated state of x at moment $n-1$, based on the measurement at moment $n-1$. Now, based on $\hat{x}_{n-1,n-1}$, $\hat{x}_{n,n-1}$ will be obtained, which is the predicted state of x at moment n. That is to say, $\hat{x}_{n-1,n-1}$ is extrapolated to moment n. Since the dynamic model is considered static, the predicted state of x is equal to the estimated state of x, namely,

$$\hat{x}_{n,n-1} = \hat{x}_{n-1,n-1} \tag{16}$$

Based on the above, the estimation of the current state $\hat{x}_{n,,n}$ can be written as

$$\hat{x}_{n,n} = \hat{x}_{n,n-1} + \frac{1}{n}\left(z_n - \hat{x}_{n-1,n-1}\right) \tag{17}$$

This equation is referred to as the state update equation.

Estimation of the current status = Prediction of the current status + Factor (Measurement - Prediction of the current status)

Fig. 3. Operation of the state update equation

In the language of the Kalman filter, the factor $1/n$ is known as the Kalman gain and, for practical applications, is replaced by K_n. Equation (13) is referred to as the state update equation. The subscript n indicates that the Kalman gain may vary with each iteration. Meanwhile, the Greek letter α_n is used to substitute K_n. Therefore, the state update equation will now be as follows.

$$\hat{x}_{n,n} = \hat{x}_{n,n-1} + \alpha_n(z_n - \hat{x}_{n,n-1}) \tag{18}$$

The term $(z_n - \hat{x}_{n,,n-1})$ is a residual measure, also called innovation. The innovation contains new information.

In our case, $1/n$ decreases as N increases. This means that, initially, there is not enough information about the temperature value; therefore, we base the estimation on the measurements. As we continue, each successive measurement contributes less to

the temperature in the estimation process, as $1/n$ decreases. At some point, the contribution of the new measurements would become insignificant.

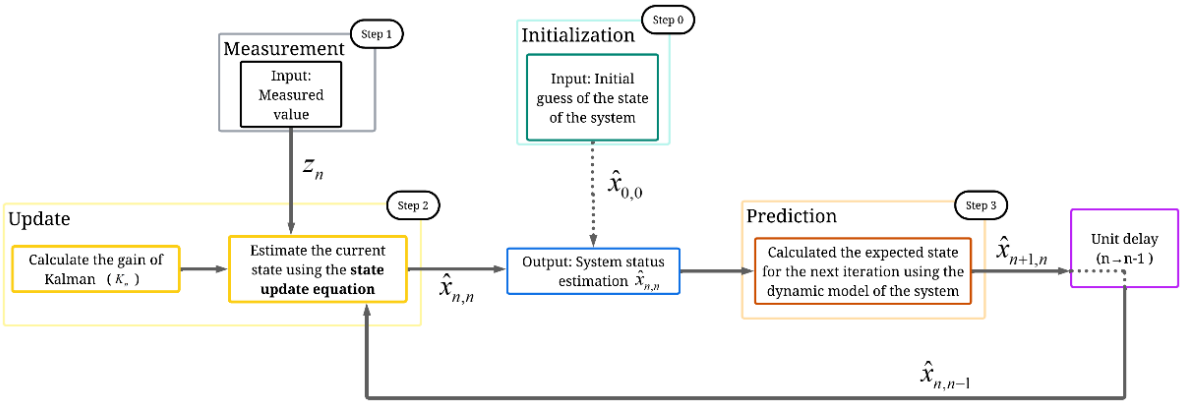

Fig. 4. Scheme of the Kalman estimation algorithm

The difference between measurements and the true value constitutes measurement errors. Since measurement errors are random, they can be described by the variance $\left(\sigma^2\right)$. The standard deviation σ of measurement errors is indeed the measurement uncertainty or measurement error. This measurement uncertainty will be denoted by $\boldsymbol{r}$. The variance of measurement errors may be provided by the supplier of the measuring equipment, calculated, or empirically derived through a calibration process. The difference between the estimate and the true value is the estimation error. The estimated error diminishes as more measurements are taken and converges towards zero, while the estimated value converges towards the true value. The estimation error is unknown; however, the uncertainty in the estimation can be estimated. The estimated uncertainty is denoted by $\boldsymbol{p}$.

The measurement uncertainty (r) is the variance of the measurement (σ^2).
When the measurement uncertainty is very large and the estimated uncertainty is very small, the Kalman gain is close to zero. On the other hand, when the measurement uncertainty is very small and the estimated uncertainty is very large, the Kalman gain is close to one. Therefore, we give a small weight to the estimate and a large weight to the measurement. If the uncertainty of the measurement is equal to the estimated uncertainty, then the Kalman gain is equal to 0.5. This can be seen by rewriting the equation $\hat{x}_{n,n} = \hat{x}_{n,n-1} + \alpha_n\left(z_n - \hat{x}_{n,n-1}\right)$ leaving it as follows

$$\hat{x}_{n,n} = (1-K_n)\hat{x}_{n,n-1} + K_n z_n \tag{19}$$

Kalman's gain tells us how much I wish to adjust my estimate based on a measurement. The following equation defines the update of the estimated uncertainty:

$$\hat{p}_{n,n} = (1-K_n)\hat{p}_{n,n-1} \tag{20}$$

Where K_n is the Kalman gain, $p_{n,n-1}$ is the estimated uncertainty calculated during the previous estimation, and $p_{n,n}$ is the estimated uncertainty of the current state. This equation updates the estimated uncertainty of the current state and is referred to as the covariance update equation. From the equation, it is observed that the estimated uncertainty always becomes smaller with each iteration of the filter, as $\left(1-K_n\right) \leq 1$. When the measurement uncertainty is large, the Kalman gain will be low, therefore, the convergence of the estimated uncertainty would be slow. However, when the measurement uncertainty is small, the Kalman gain will be high and the estimated uncertainty would converge rapidly towards zero.
In this model, the dynamics of the system are constant. Therefore, the extrapolation of the estimated uncertainty would be:

$$\hat{p}_{n+1,n} = \hat{p}_{n,n} \tag{21}$$

Where p is the estimation of the uncertainty (of the system temperature), this equation is known as the covariance extrapolation equation.

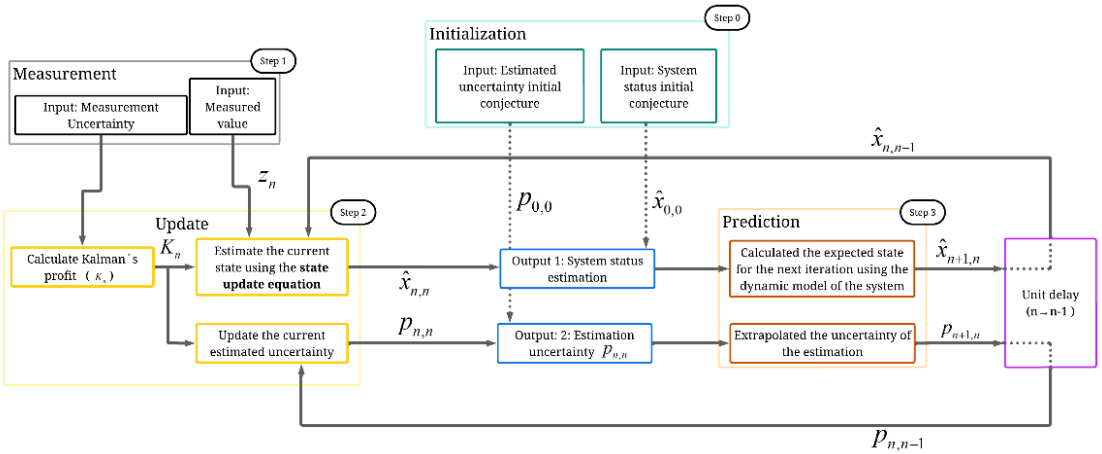

Fig. 5. Detailed block diagram of the Kalman filter operation.

Initialization is step 0. It is performed only once and provides two parameters: the initial state of the system $\left(\hat{x}_{1,0}\right)$ and the initial uncertainty of the state $\left(p_{1,0}\right)$. Initialization is followed by prediction.

Step 1 is measurement. The measurement process will provide two parameters: the measurement of the system state $\left(z_n\right)$ and the measurement uncertainty $\left(r_n\right)$.

Step 2 is the status update. The status update process is responsible for estimating the current state of the system. The inputs of the state update process are: measured value $\left(z_n\right)$, measurement uncertainty $\left(r_n\right)$, estimation of the state of the previous system $\left(\hat{x}_{n,n-1}\right)$ and estimated uncertainty $\left(p_{n,n-1}\right)$.

According to the entries, the state update process calculates the Kalman gain and provides two outputs: the current state estimate of the system $\left(\hat{x}_{n,n}\right)$ and the estimated uncertainty of the current state $\left(p_{n,n}\right)$. These two parameters are the outputs of the Kalman filter.

The final step is step 3, which is the prediction. The prediction process extrapolates the current state of the system and the estimation error of the current state of the system to the next state of the system. In the first iteration of the filter, the initialization outputs are treated as the estimation and uncertainty of the previous state. In the subsequent iterations of the filter, the prediction outputs become the previous estimation and uncertainty.

The Kalman filter treats the estimate as a random variable, because of this the variance of the estimate must be extrapolated to the next state, this concept is defined as $p_{n,n}$. In this case, the extrapolation of the estimated uncertainty would be

$$p_{n+1,n} = p_{n,n} \tag{22}$$

Where p. is the estimated variation. The equation for extrapolating estimated uncertainty is known as the covariance extrapolation equation. The Kalman filter combines the previous state estimate with the measurement in a way that minimizes the uncertainty of the current state estimate. The current state estimate is a weighted average of the measurement and the previous state estimate.

$$\hat{x}_{n,n} = w_1 z_n + w_2 \hat{x}_{n,n-1} \tag{23}$$

$$w_1 + w_2 = 1 \tag{24}$$

Where w_1 and w_2 are the weighting factors for the measurement and estimation of the previous state. Equation (**23**) is rewritten as

$$\hat{x}_{n,n} = w_1 z_n + (1 - w_1)\hat{x}_{n,n-1} \tag{25}$$

The relationship between the deviations is given by:

$$p_{n,n} = w_1^2 r_n + (1 - w_1)^2 p_{n,n-1} \tag{26}$$

Where $p_{n,n}$ is described as the variance of the optimal combined estimation $\hat{x}_{n,n}$, $p_{n,n-1}$ is the variance of the previous estimation $\hat{x}_{n,n-1}$, y r_n (measurement uncertainty) is analogous to the variance of the measurement r_n. Initially, to find w_1, the drift is calculated $p_{n,n}$ homogeneously, obtaining the rate of change as a function of the variable w_1.

$$\frac{dp_{n,n}}{dw_1} = 2w_1 r_n - 2(1 - w_1) p_{n,n-1} = 0 \tag{27}$$

By equating equation 19 to 0, we can first expand the terms by multiplying the values found within the parentheses:

$$2w_1 r_n - 2p_{n,n-1} + 2w_1 p_{n,n-1} = 0 \tag{28}$$

Subsequently, the terms containing w_1 are grouped and factored, and the term $2p_{n,n-1}$ is moved to the right side of the equation.

$$2w_1(r_n + p_{n,n-1}) = 2p_{n,n-1} \tag{29}$$

Now finally, we simplify the like terms.

$$w_1(r_n + p_{n,n-1}) = p_{n,n-1} \tag{30}$$

and we solve for w_1, obtaining the result of w_1 as

$$w_1 = \frac{p_{n,n-1}}{r_n + p_{n,n-1}} \tag{31}$$

This value is substituted in the ec.25 and we obtain the estimation of the current state:

$$\hat{x}_{n,n} = \hat{x}_{n,n-1} + \frac{p_{n,n-1}}{p_{n,n-1} + r_n}(z_n - \hat{x}_{n,n-1}) \tag{32}$$

The equation for the one-dimensional Kalman gain is as follows.:

$$K_n = \frac{p_{n,n-1}}{p_{n,n-1} + r_n} \tag{33}$$

The measurement uncertainty (r_n) is the variance of the measurement σ_n^2.

Now, the variance of the current state estimation will be obtained. It is known that the relationship between the variances is given by

$$p_{n,n} = w_1^2 r_n + (1 - w_1)^2 p_{n,n-1} \tag{34}$$

The weight w_1 is a Kalman gain.

$$p_{n,n} = K_n^2 r_n + (1 - K_n)^2 p_{n,n-1} \tag{35}$$

Thus, the term $(1 - K_n)$ will now be determined.

$$(1-K_n)=\left(1-\frac{p_{n,n-1}}{p_{n,n-1}+r_n}\right)=\frac{p_{n,n-1}+r_n-p_{n,n-1}}{p_{n,n-1}+r_n}=\frac{r_n}{p_{n,n-1}+r_n} \tag{36}$$

We substitute this result into $p_{n,n}=K_n^2 r_n+(1-K_n)^2 p_{n,n-1}$

$$p_{n,n}=\left(\frac{p_{n,n-1}}{p_{n,n-1}+r_n}\right)^2 r_n+\left(\frac{r_n}{p_{n,n-1}+r_n}\right)^2 p_{n,n-1} \tag{37}$$

We squared the product of the corresponding terms

$$p_{n,n}=\frac{p_{n,n-1}^2 r_n}{(p_{n,n-1}+r_n)^2}+\frac{r_n^2 p_{n,n-1}}{(p_{n,n-1}+r_n)^2} \tag{38}$$

We factor the terms.

$$p_{n,n}=\frac{p_{n,n-1}r_n}{p_{n,n-1}+r_n}\left(\frac{p_{n,n-1}}{p_{n,n-1}+r_n}+\frac{r_n}{p_{n,n-1}+r_n}\right) \tag{39}$$

Knowing the value of $(1-K_n)$ and K_n, we rearrange the terms and obtain.

$$p_{n,n}=(1-K_n)\,p_{n,n-1}\left(K_n+(1-K_n)\right) \tag{40}$$

Finally, the covariance update equation is defined as

$$p_{n,n}=(1-K_n)\,p_{n,n-1} \tag{41}$$

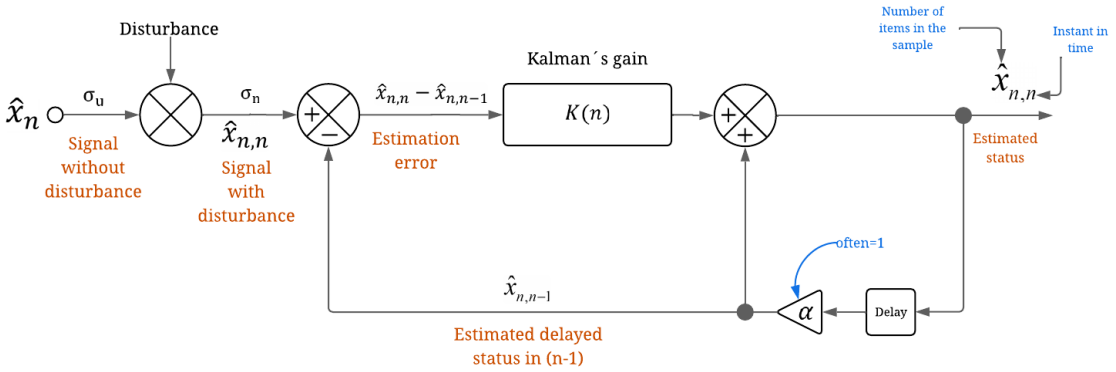

Fig. 6. Block diagram of the simplified Kalman filter.

The operation process of the Kalman filter, within the Arduino code, consists of 7 important points.
The first is the state and measurement, where the state (shat) estimates the actual voltage of the sensor, and the measurement (xn) performs the analog reading converted to voltage. The second is to define the key variables, which are: P (estimation uncertainty), σ_u (sigma_u) the model noise (representing the small expected variation between steps), and σ_n (sigma_n) is the measurement noise (possible instability of the reading). The third is to calculate, at each step, the variance of all readings up to that point, to adjust the sensor's confidence. The fourth is the prediction, where the filter predicts what the new voltage value (shat) might be based on the previous value. The fifth is to assess how much trust can be placed in the measurement x_n compared to the prediction (shat). The sixth step, after obtaining a new measurement and determining how much to trust it (due to K), is to correct the voltage estimation. Finally, the seventh step involves sending the obtained information to the serial monitor, including the raw voltage (x_n) and the filtered voltage (shat). Subsequently, the cycle is repeated for the next reading.

3 Results and Discussion

In the present section, the experimental results have been discussed. The block diagram in discrete form for overall system of this device that is connected to the fan is illustrated in Figure 7, where T(k) is the air temperature inside the thermal chamber directly measured by the thermocouple and T_d is the desired temperature; e(k) is the error, at sampling time *k*, which is obtained by resting the desired temperature set by the user, and the estimated temperature $\hat{T}$, obtained with the Kalman's filter. This digital control can be treated as a continuous-time control system, even though it operates in discrete time due to several reasons but mainly because of two: high sampling frequency and slow system dynamics. In this case, the sampling time used in the controller is small (0.2) allowing a high sampling frequency; on the other hand, the environmental temperature changes so slowly, in contrast to the sampling interval, that the digital system effectively behaves like a continuous one. In addition, digital algorithms, such as the discrete-time PID used here, can be derived directly from their continuous-time counterparts using approximation methods (Euler's method, trapezoidal integration, or bilinear transformation), as previously explained.
For real-time results, the hardware prototype of this project is tested and a PC is used for data logging purpose, to verify the performance of the project.

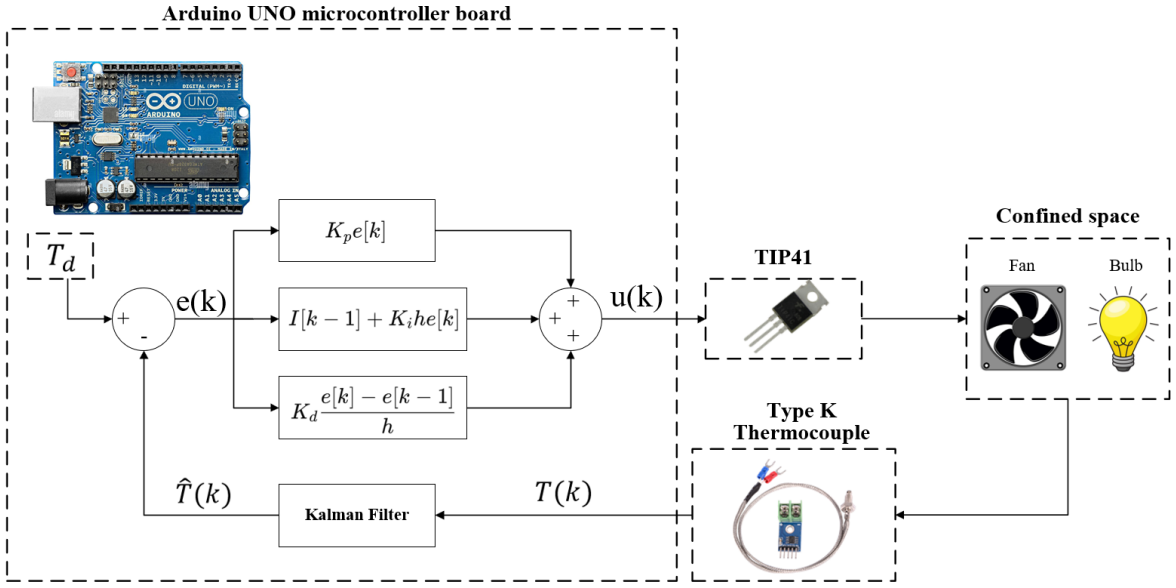

Fig. 7. Block diagram for the discrete form of the PID controller ran on Arduino.

The prototype of the developed device is depicted in Figure 8 where the LCD display prints both desired and current air temperature inside the thermal chamber. A type K thermocouple sensor probe is immersed inside the thermal chamber just above the incandescent bulb. The device is also equipped with a switch to enable user to turn on/off the bulb. The PID controller gains, as well as the desired temperature, are set directly in the Arduino code. An air recirculation tube is also added for a better performance.

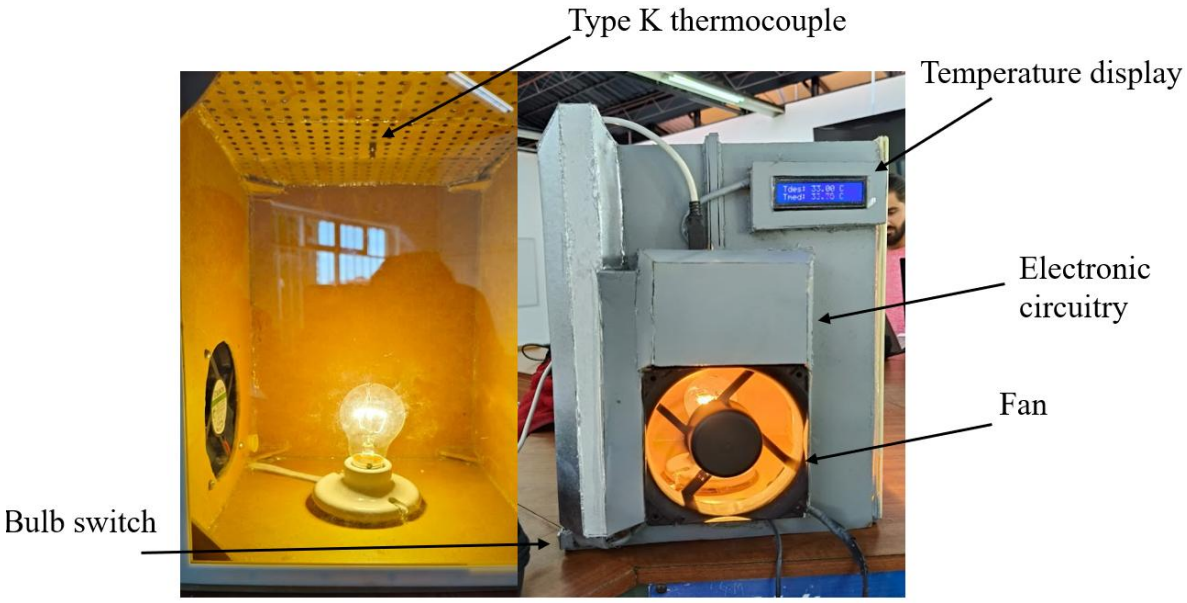

Fig. 8. Temperature control prototype.

To determine the mathematical model of the plant, the Gauss-Newton algorithm is applied to experimental data collected from the system in response to a step input. The model is characterized as a second-order transfer function as follows:

$$G(s)=\frac{2.4996}{2079.7279s^2+227.913s+1} \tag{42}$$

The results were consistent, yielding a correlation value of 87.8%. The closed loop system applying PID control is:

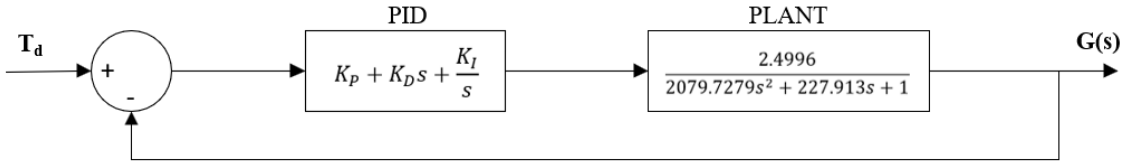

Fig. 9. Block diagram of the closed loop system with a PID controller.

Simplifying the system in Figure 9 and removing the feedback loop results in:

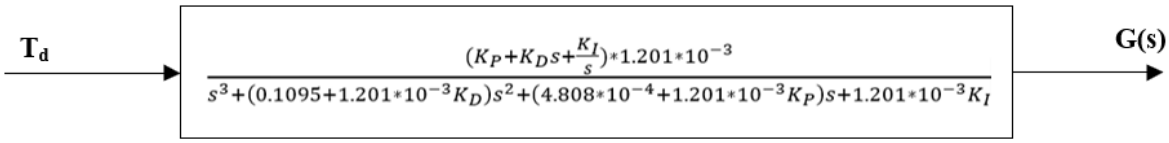

Fig. 10. Closed loop transfer function of the system.

The following complex conjugated poles are proposed for an underdamped response:

$$P_d(s)=(s+0.295558)(s-0.0928973-j0.272740)(s-0.0928973+j0.272740) \tag{43}$$

Multiplying the poles to obtain the characteristic polynomial:

$$P_d(s)=s^3+0.10976422s^2+0.0281038s+0.02453643 \tag{44}$$

Then the PID gains are obtained through pole assignment, equating the coefficients of the poles shown in Figure 10, with the ones obtained in Equation 44, yielding as follows: K_p=23, K_i=20.43, K_d=0.22. Which are the constants to be used in the digital PID.

The PWM signal is constrained in the coding between 150 and 255, where 150 represents the minimum voltage to activate the fan. The test is performed for about 3 minutes in which data is gathered for its post-analysis.

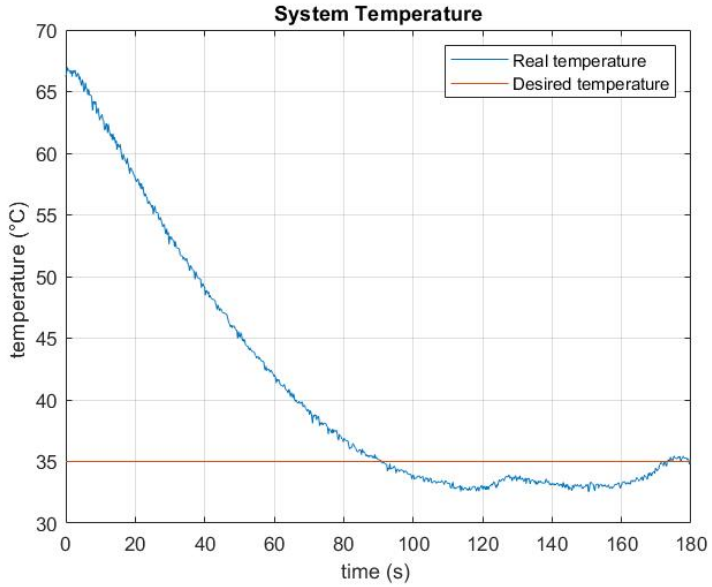

Fig. 11. Response of air temperature inside the thermal chamber using PID controller.

The temperature values plotted in Figure 11 are obtained using the Kalman filter. A 35° C temperature setpoint is stablished directly in the code. The value of the standard deviation σ_u of the noise-free signal is founded with a trial-and-error method estimating a parameter of 2800.

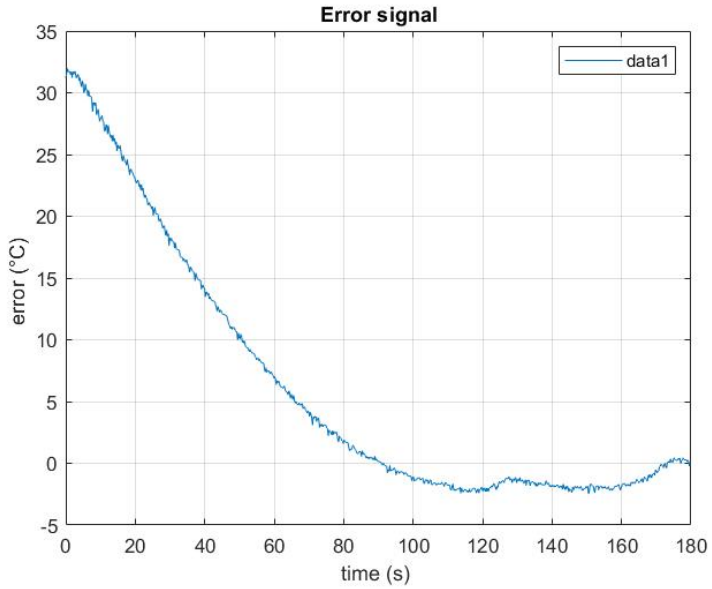

Fig. 12. Graph of the error signal.

In Figure 12 the error values are plotted and it can be seen how it reaches zero at around 90 seconds and stays close to it for the entire time of the test.

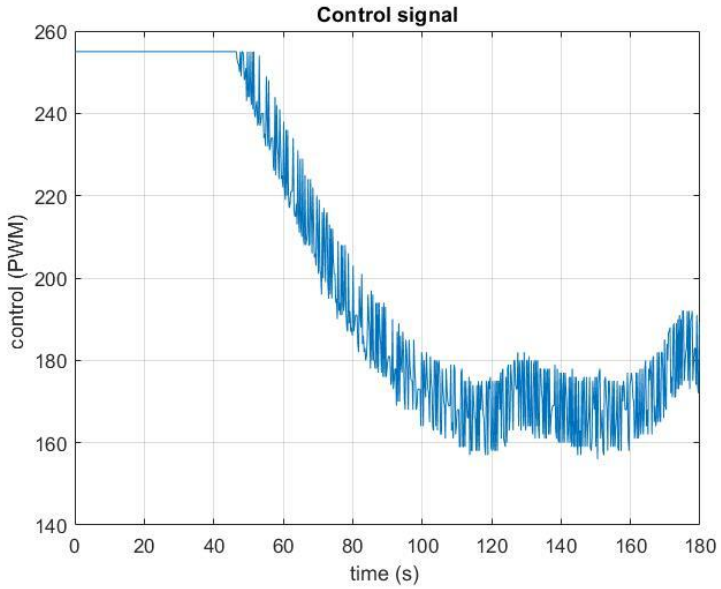

Fig. 13. Graph of the control signal given by the PID controller

As seen in Figures 11 and 12, the system achieved a response of 94 seconds to reach the set point, and a settling time of approximately 180 seconds, demonstrating practical viability. A 6.91% overshoot value is obtained according to the data logged. The percentage overshoot represents the amount that the response overshoots its steady-state (or final) value at the peak time, and is expressed as a percentage of the steady-state value.

Conclusions

The prototype of a didactic temperature-controlled thermal chamber was successfully fabricated and tested. Besides, the implementation of the PID system together with a Kalman filter in the temperature controller resulted in acceptable temperature regulation within a semi-open system, a configuration that could be scaled to industrial environments. The proposed gains were able to provide a response without oscillations, reflecting a certain robustness in the control system. This may imply a pole assignment on the negative real axis. The PID controller properly adjusted the control signal to reach and maintain the desired value, while the Kalman filter significantly improved the quality of the measurements provided by the sensor by reducing the noise present in the readings. As a result, the system achieved a partially fast response with a decent settling time and a low overshoot percentage.

This project underscored the importance of integrating theory, modeling, and real-world response analysis to achieve robust practical outcomes. The experience lays a foundation for future prototype upgrades and inspires the development of low-cost educational systems to support STEM students.

Future work could explore hardware enhancements like adding another sensor (different from the one used here) and averaging the data obtained from both; insulation material also needs to be considered in order to minimize heat exchange with the environment, reducing energy consumption and improving control stability. Similarly, adaptive PID algorithms may further optimize performance under dynamic conditions.

References

Alviz Blanco, E. R. (2022). *Design and implementation of an educational module for temperature control using PI, PID, and Smith Predictor controllers.*

Artieda Cruz, L. E. (2021). *FPGA-based implementation of a Kalman filter to reduce noise effect in temperature sensors* (Master's thesis). Escuela Superior Politécnica del Litoral (ESPOL). Retrieved from http://www.dspace.espol.edu.ec/handle/123456789/56407

Bañuelos, M., Castillo, J., Rayo, G., Quintana, T., Damián, R., & Pérez, J. (2000). *Didactic-type temperature PID controller.* Universidad Nacional Autónoma de México (UNAM). Retrieved from http://pronton.ucting.udg.mx/somi/memorias/didactica/Did-2.pdf

Busu, N. A., Mat Isa, N., Hariri, A., & Hussein, M. (2023). Comparison of P, PI and PID strategy performance as temperature controllers in active iris dampers for centralized air conditioning systems. *Journal of Advanced Research in Fluid Mechanics and Thermal Sciences, 102*(2), 143–154.

Calderon-Lopez, J. V., Gutiérrez-Moreno, E., Ordaz-Oliver, J. P., Ordaz-Oliver, M. O., & Montiel-Hernández, J. F. (2023). Implementation of optimal control for thermal regulation in finite-volume spaces described by second-order dynamics. *International Journal of Combinatorial Optimization Problems and Informatics, 14*(3), 147–156. https://doi.org/10.61467/2007.1558.2024.v15i1.391

Canduela Ilundain, J. I. (2022). *PID temperature control system for teaching automatic regulation subjects.* (Undergraduate project). [Institution not verified – no official URL found.]

Coto Barriere, E. A. (2017). *Temperature Control System. IEEE.* https://doi.org/10.1109/ICIEAC.2017.8076586

Cruz, G. (2014). A review of heating and temperature control systems for polymer extrusion. *Revista CINTEX, 19,* 127–143.

Díaz-Montes, E., Vargas-León, E. A., Garrido-Hernández, A., & Ceron-Montes, G. I. (2021). Simulation of heat transfer in the feeding container of a spray dryer and its temperature control. *Pädi Scientific Bulletin of Basic Sciences and Engineering of ICBI, 9,* 168–173.

Domínguez-Zenteno, J. E., De los Santos-Ruiz, I., Velasco-Bermúdez, S., Brindis-Velázquez, O., León-Orozco, V., Bermúdez-Hernández, R., & López-Estrada, F. R. (n.d.). *The Kalman Filter: A journey through its history, mathematical foundation, and practical applications. Revista Digital,* 56.

Dorf, R. C., & Bishop, R. H. (2017). *Modern Control Systems* (13th ed.). Pearson.

Evolution Sensors and Controls. (2025). *Type K thermocouple probe – 12 inch long, 1/8" diameter stainless steel MI cable ungrounded with transition to FEP lead wire and miniature connector.* Retrieved April 9, 2025, from https://evosensors.com/products/type-k-thermocouple-probe-12-inch-long-1-8-diameter-stainless-steel-mi-cable-ungrounded-with-transition-to-fep-lead-wire-and-miniature-connector

Kherkhar, A., Chiba, Y., Tlemçani, A., & Mamur, H. (2022). Thermal investigation of a thermoelectric cooler based on Arduino and PID control approach. *Case Studies in Thermal Engineering, 36,* 102249. https://doi.org/10.1016/j.csite.2022.102249

López Sobrino, D. (2019). *Construction of a greenhouse and implementation of different control techniques for humidity and temperature* (Master's thesis). Universidad Politécnica de Madrid (UPM). Retrieved from https://oa.upm.es

Raju, K. S., Govardhan, A., Rani, B. P., Sridevi, R., & Murty, M. R. (Eds.). (2020). *Proceedings of the Third International Conference on Computational Intelligence and Informatics: ICCII 2018* (Vol. 1090). Springer Nature.

Rodríguez-Cruz, A., Quijano-Castillo, C. I., Hernández-Bautista, G., Vázquez-Hernández, O. O., & Vélez-Díaz, D. (2018). Temperature control for egg incubation. *XIKUA Scientific Bulletin of the Escuela Superior de Tlahuelilpan, 6*(11).

Shikano, A., Tonthat, L., & Yabukami, S. (2021). A simple and high-accuracy PID-based temperature control system for magnetic hyperthermia using a fiber optic thermometer. *IEEJ Transactions on Electrical and Electronic Engineering, 16*(5), 807–809. https://doi.org/10.1002/tee.23373

Sozański, K. (2023). Low-cost PID controller for a student digital control laboratory based on Arduino or STM32 modules. *Electronics, 12*(15), 3235. https://doi.org/10.3390/electronics12153235

The MathWorks, Inc. (1999–2001). *Real-Time Windows Target User's Guide.*

Vázquez Leiva, J. H., Peláez Llano, A., & Vargas Fernández, J. E. (2024). Adaptive control of a three-phase motor for heavy load equipment using a Kalman filter. *Serie Científica de la Universidad de las Ciencias Informáticas, 17*(5), 84–102. Retrieved from https://publicaciones.uci.cu/index.php/serie/article/view/1598