AJAPAS

African Journal of Advanced Pure and Applied Sciences (AJAPAS)

Online ISSN: 2957-644X

Volume 2, Issue 4, October-December 2023, Page No: 340-351

Website: *https://aaasjournals.com/index.php/ajapas/index*

معامل التأثير العربي 2023: (1.55) SJIFactor 2023: 5.689 ISI 2022-2023: 0.557

Real Time Temperature Control Using LabVIEW Based on PID Algorithm and Power Electronics

Haithem Ali Ali [1 *], Noura Mohammed Salem [2], Retaj Mohammed Abdullah [3], Sumaya Basher Abdullah [4]

[1,2,3,4] Control Engineering Department, College of Electronic Technology, Bani Walid, Libya

[*]*Corresponding author: haamaliali@gmail.com*

Received: October 17, 2023	Accepted: November 26, 2023	Published: December 17, 2023

Abstract:

There are many temperature control systems for either heating or cooling, the most famous of which is the ON-OFF control system. Most of these systems draw a high current when starting, which leads to a large consumption of energy, this affects the overall efficiency of the system, and the response of the system is not completely as required, as an overshoot occurs in response due to continuous operation and stopping. Some systems are not greatly affected by the overshoot rate that occurs in the response of the system, and some systems, if the temperature control is not very accurate, will lead to the failure of the entire system and will entail major problems. An example of this is the incubator, which is a machine that adjusts the appropriate temperature and humidity that the eggs need to hatch. Any change in temperature causes problems and defects and reduces the hatching rate a lot, so this process requires constant monitoring of the temperatures in the incubator. In this project, the focus was on solving two problems: the problem of energy consumption in heating devices and the problem of override in the system response. By designing the feedback control system to maintain the system temperature within a particular range based on a feedback signal from the temperature sensor, as a result, the power consumption will be reduced, the overshoot will be eliminated, and the system response will be obtained. LabVIEW program was used to draw the system response. The parameters of the controller were modified using trial and error method to obtain good system characteristics. System results show that the controller was able to adjust the temperature without an overshoot.

Keywords: LabVIEW, PID Controller, Temperature Control, Power Electronics, Arduino

Cite this article as: H. A. Ali, M. N. Salem, R. M. Abdullah and S. B. Abdullah, "Real Time Temperature Control Using LabVIEW Based on PID Algorithm and Power Electronics," *African Journal of Advanced Pure and Applied Sciences (AJAPAS)*, vol. 2, no. 4, pp. 340–351, October-December 2023.

تصميم وتنفيذ نظام للتحكم ومراقبة الحرارة بإستخدام المتحكم PID مع LabVIEW

هيثم علي علي[1*]، نورا محمد سالم[2]، رتاج محمد عبدالله[3]، سمية بشير عبدالله[4]

[1،2،3،4] قسم التحكم الآلي، كلية التقنية الإلكترونية، بني وليد، ليبيا

الملخص
يوجد العديد من انظمة التحكم في درجة الحرارة إما للتسخين أو التبريد، ومن أشهرها نظام التحكم في التشغيل والإيقاف تسحب معظم هذه الأنظمة تياراً عالياً عند بدء التشغيل مما يؤدي إلى استهلاك كبير في الطاقة، وهذا يؤثر على كفاءة النظام الكلية واستجابة النظام لا تكون بالشكل المطلوب تماماً حيث يحدث overshoot في الاستجابة بسبب التشغيل والإيقاف المستمر. بعض الأنظمة لا تتأثر بشكل كبير بنسبة ال overshoot الذي يحدث في استجابة النظام والبعض الاخر من الأنظمة إذا لم يكن فيها التحكم في درجة الحرارة بشكل دقيق جداً فسيؤدي ذلك إلى فشل النظام بالكامل وسيترتب على ذلك مشاكل كبيرة. مثال على ذلك فقاسة البيض وهي الة تعمل على ضبط درجة الحرارة الرطوبة المناسبة التي يحتاجها البيض لكي يفقس. أي تغير في درجات الحرارة يسبب مشاكل وعيوب ويخفض نسبة التفقيس كثيراً. لذلك تتطلب هذه العملية مراقبة درجات الحرارة في الفقاسة بشكل مستمر. في هذا المشروع، تم التركيز على حل اثنان من المشاكل: مشكلة استهلاك الطاقة في الأجهزة ومشكلة ال overshoot في استجابة النظام عن طريق تصميم نظام التحكم في التغذية الراجعة للحفاظ على درجة حرارة النظام ضمن نطاق معين بناءًا على إشارة تغذية راجعة من حساس درجة الحرارة. نتيجة لذلك، سيتم تقليل استهلاك الطاقة وسيتم التخلص من ال overshoot والحصول على الاستجابة المطلوبة للنظام. تم استخدام برنامج اللابفيو لرسم استجابة النظام. تم تعديل بارامترات المتحكم باستخدام طريقة التجربة والخطأ وذلك للحصول على خصائص جيدة للنظام. النتائج العملية تظهر أن النظام كان قادر على ضبط درجات الحرارة بدون زيادة أو نقصان عن القيمة المرغوبة (overshoot).

الكلمات المفتاحية: لابفيو، إلكترونات القدرة، PID Controller، اردوينو، التحكم في الحرارة.

Introduction

Heating systems are widely used in the processes of industries for efficient processes and good quality products. The controls automatically turn the heating on and off based on settings input by the user, to ensure maximum comfort. Modern electrical appliances such as air conditions, water pumps, fans, heaters, etc. These systems draw a high current when starting, which leads to a large consumption of energy. This will affect the efficiency of the system and the life span of the device. Developing control systems in heaters' applications have been drawn huge attention, there are many studies focusing on this topic. ON-OFF Control system, on-off control is the simplest form of feedback control. An on-off controller simply drives the manipulated variable from fully closed to fully open depending on the position of the controlled variable relative to the setpoint. A common example of on-off control is the temperature control in a domestic heating system. When the temperature is below the thermostat setpoint, the heating system is switched on and when the temperature is above the setpoint the heating switches off. There is, however, a bit of subtlety applied in practical on-off systems. If the heating switches on and off the instant, the measured temperature crossed the setpoint, an overshoot will occur in the system response. Figure 1 shows the response of the on-off control system.

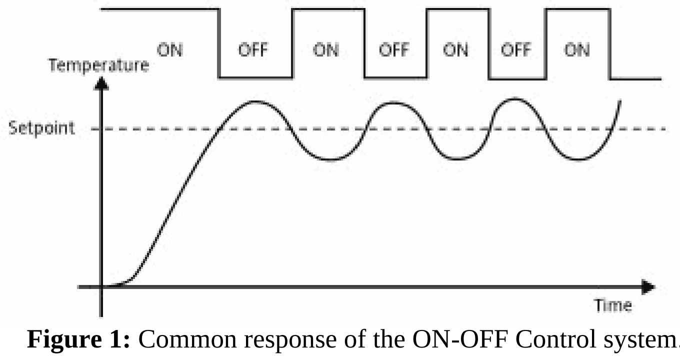

Figure 1: Common response of the ON-OFF Control system.

Nowadays, energy consumption is one of the most difficult challenges facing the world. One of the solutions to this problem is the intelligent control of energy consumption in electrical appliances. In the existing literature, there are many research papers for different systems and ways working on control the temperature and reduce the energy consumption. We studied several papers and here few of important contributions are presented.

One of the papers that has been published discusses the design of an Arduino Uno-based microcontroller system for automatic temperature management. The hardware used to interact with the computer was the temperature sensor LM35 and Arduino Uno, and the temperature in the room is controlled. Temperature is shown on an LCD display using the hardware A1 pin and an analog pin that uses pulse width modulation (PWM). [1]. Muhammad [2] developed a PID control strategy for monitoring and managing the temperature of a heating element, which is sensed by a thermocouple as the measuring device, using an Arduino microcontroller and Virtual Instrumentation

(VI) software called LabVIEW. The experimental results demonstrate that a PID controller can monitor and regulate heater temperature in order to maintain a specified reference temperature. The temperature control procedure for a hair drier was modeled in [3]. This design contains the Arduino board as the primary control device, and the Arduino interfaced thermocouple with MAX6675 will be used to measure temperature. A heater is used to warm the air that is being blown out of the dryer's primary air inlet, fan, and exhaust. The thermocouple uses a zero crossing detector to determine the output voltage based on the temperature of the reflected air. This device electrically triggers the TRIAC angle to turn on and off the heater, thereby managing the amount of heat power. The experimental analysis continually controls the hair dryer temperature, and as a result, the fan speed is controlled by regulating the TRIAC angle. The experimental results showed that the hair dryer system temperature is effectively controlled. By combining sensors, PID controllers, LCD displays, DC motors, and fan control systems, Okpagu et al. proposed creating a temperature control system for egg incubators. Due to the necessity of monitoring the embryo and its progress, this type of incubator system is crucial, and temperature control and monitoring were therefore crucial components of it [4]. The concept and design of temperature distribution control for infant incubator use was suggested by Wayan et al. To ensure a baby's healthy development in this system, it is crucial to maintain a specific temperature within the room. In the examination of the experiment employing the microcontroller-based system for temperature measurement and control, humidity was also taken into consideration. This application for baby care and health turned out to be really significant [5].

We conclude from the previous literature that all research was intended to control the temperature in different ways. Therefore, in this paper, a feedback control system is developed to maintain the heaters' temperature within a particular range based on a feedback signal from a temperature sensor. As a result, power consumptions will be reduced, the overshoot will be eliminated and the required system response will be obtained. The temperature will be controlled using the Arduino board with power electronics (Triac, Diac, etc.) away from the use of classic technologies (Relay, Contactors, etc.) to get a better performance of temperature control systems in terms of the required heat accuracy and reduce energy consumption. The PID algorithm is implemented on the Arduino board, and the Arduino is linked with the LabVIEW program to display the practical results to monitor the results in real time.

This work is arranged as follows: Section 2, which is devoted to the block diagram of the system with a brief explanation about the system components. Section 3, covers results and discussion of this work and section 4 of this paper presents the conclusion of the performed work.

System block diagram

Figure 2 shows the complete system diagram of the temperature control system and its controller board with two inputs (temperature sensor and zero crossing detector) and with two outputs (heating and cooling circuits). The PID algorithm is installed in the Arduino board and used as the main brain of the whole system; it receives data from the input devices and then updates the output devices via Arduino board. The goal of the system is to be able to receive the required temperature from the user through the LabVIEW program, then work to adjust it using heating and cooling circuits without an overshoot in the output while reducing the power consumed by the cooling and heating circuits. Figure 3 shows the prototype of the real time temperature control using LabVIEW Based on the PID algorithm and power electronics.

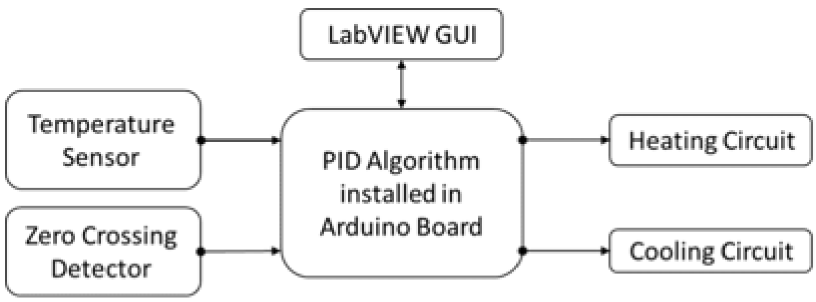

Figure 2: Block diagram of the real time temperature control system

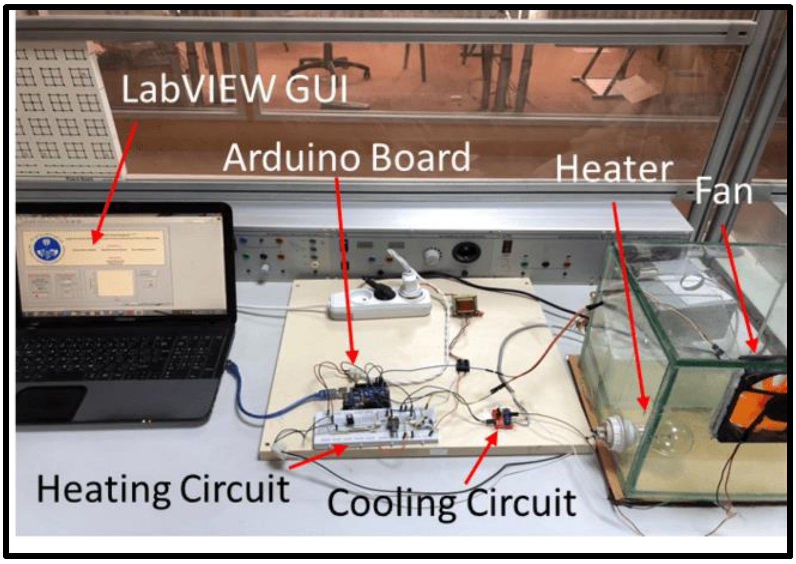

Figure 3: The prototype of the real time temperature control

1) System Hardware

The physical parts of a system are referred to as the hardware. Any component of the system that we can touch is referred to as system hardware. These are the main mechanical, electrical, and electronic components that make up the system. Hardware in this system includes, for instance, the Arduino board, Temperature sensor, MOSFET driver, etc.

- **Temperature Sensor**

DS18B20 is a Temperature Sensor 1-Wire interface developed by Dallas Semiconductor Corp. The sensor normally comes with two type factors. One that comes with box TO-92 looks much like a regular transistor. Another one in a rugged probe type that can be more effective when testing anything far away, underwater or below level. The temperature sensor DS18B20 is reasonably reliable and does not require additional components to run. With a precision of ±0.5 ° C, it can calculate temperatures from -55 ° C to +125 ° C. Figure 4 is shows the shape of DS18B20 Temperature Sensor [6].

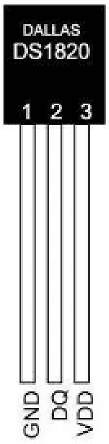

Figure 4: The DS18B20 temperature sensor

- **Zero Crossing Detector**

The instantaneous point in an alternating current signal where no voltage is present is known as the zero crossing. This often happens twice every cycle in a sine wave or other simple waveform. It is an instrument for determining the point at which the voltage in either direction crosses zero. Systems that transmit digital data over AC circuits, such modems, home automation control systems, and Digital Command Control systems used in Lionel and other AC model trains, depend on the zero-crossing. Another technique in speech processing to determine the basic frequency of speech is counting zero-crossings. With regard to providing the necessary supply to the other circuits, the base circuit will be displayed as it is in (Figure 5). It is necessary and desirable for the other circuit components to have a voltage standard of 9 V, which is achieved by the transformer. Furthermore, prior to the output half wave from the x-axis bridge entering the Arduino, the opto-coupler that produces the correct pulse at the zero crossing with it is exhibited in (Figure 6).

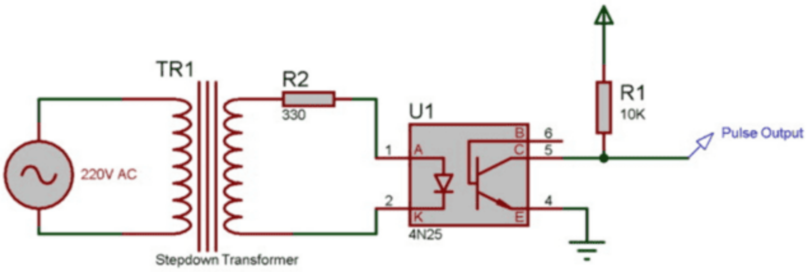

Figure 5: The Zero Crossing Detector Circuit

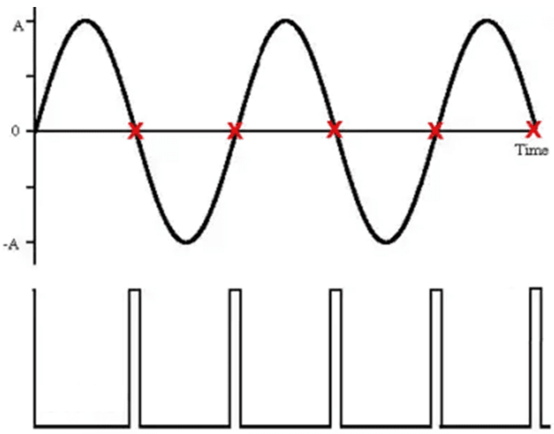

Figure 6: The Output of Zero Crossing Detector Circuit

- **Phase Control Driver**

AC phase angle control is a method through which we can control or chop an AC sine wave. The circuit shown in Figure 7 was used, relying on the Triac and the Diac to control the voltage sent to the heater. The firing angle of the switching device is varied following a zero-crossing detection, resulting in an average voltage output that change proportionally with the modified sine wave; the figure 8 describes phase control method.

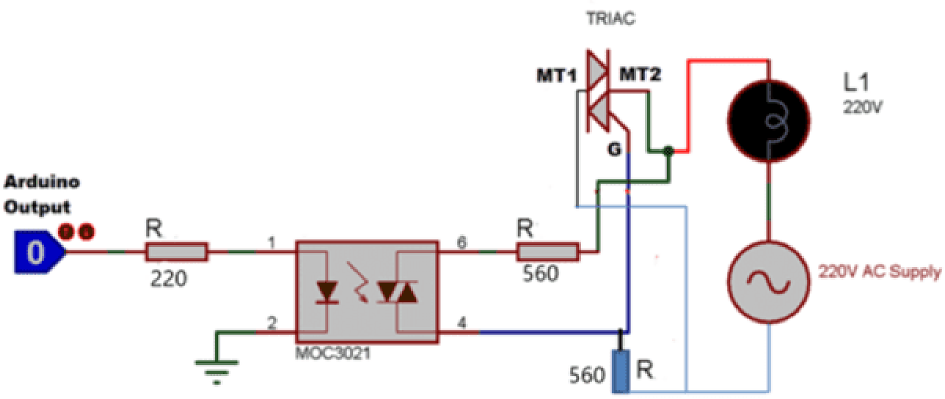

Figure 7: The Firing Circuit of TRIAC Element

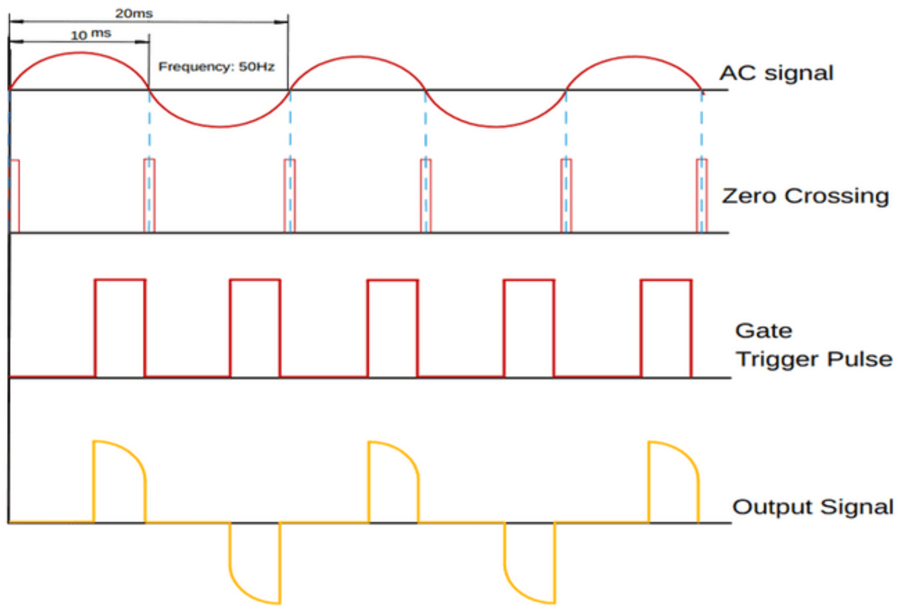

Figure 8: The Principle Working of Phase Shift Control

From figure 8, we start with our AC input signal, as you can see. The zero-crossing signal is the next in line; it causes an interrupt every 10 milliseconds. Next is the gate trigger signal. After receiving a triggering signal, we wait before delivering the trigger pulse for a set period. The longer we wait, the lower the average voltage can be, and vice versa. In the final experiment, the PID algorithm will be responsible for the time of triggering the Triac based on the changes that occur in the reading of the temperature sensor.

- **MOSFET Transistor**

A MOSFET transistor is one of the classifications of FET field transistors, which is an acronym for (Field Effect Transistor). The MOSFET transistor consists of three external terminals: the drain terminal, the gate terminal, and the source terminal. Where a metal electrode and another layer called a metal oxide insulator, is placed between the terminal of the gate and what is inside the transistor, to control the states of the transistor indirectly using a technique similar to the field effect. The PID algorithm sends the appropriate signal to the gate of mosfet, so that the mosfet opens a suitable path for the electrical current to the fan to decrease the temperature of the room. The circuit diagram of Mosfet with Arduino board is shown in Figure 9.

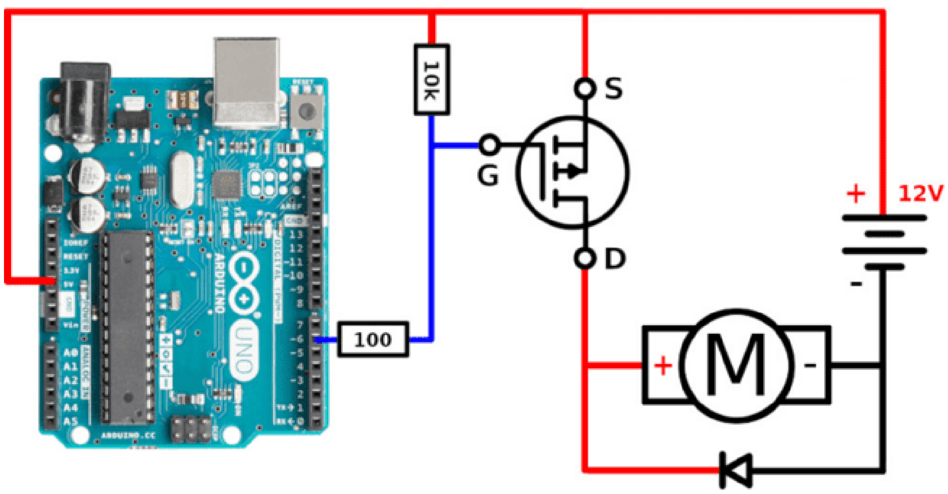

Figure 9: The Circuit Wiring of MOSFET with Arduino

- **Arduino Board**

Arduino is open-source electronics prototyping platform based on flexible, easy-to-use hardware and software. It is intended for artists, designers, hobbyists, and anyone interested in creating interactive objects or environments. Arduino can sense the environment by receiving input from a variety of sensors and can affect its surroundings by controlling lights, motors, and other actuators. The microcontroller on the board is programmed using the Arduino programming language (based on Wiring) and the Arduino development environment (based on Processing). Arduino projects can be stand-alone or they can communicate with software running on a computer [8]. The Arduino Uno is a microcontroller board based on the ATmega328 as shown in figure 10. It has 14 digital input/output pins (of which six can be used as PWM outputs), six analog inputs, one UART (hardware serial port), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with an AC-to-DC adapter or battery to get started [7].

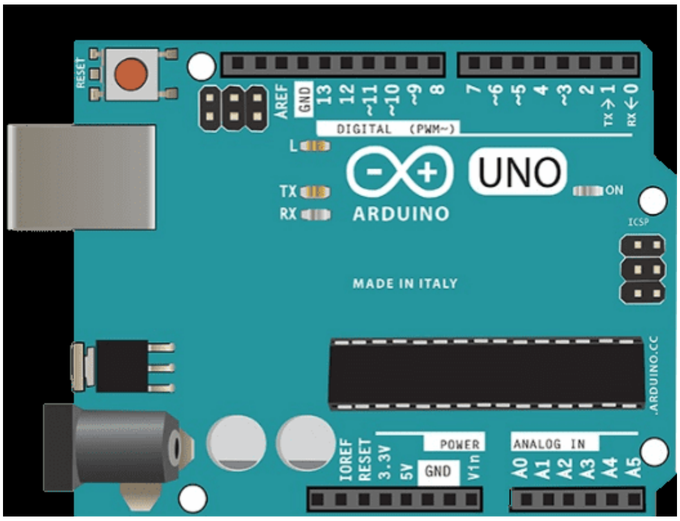

Figure 10: The Arduino UNO Board

2) System Software

The software is a set of guidelines, processes, and documentation that manages various tasks on a system. The PID algorithm in the Arduino IDE and GUI in LabVIEW program were the two pieces of software employed in this system.

- **PID Algorithm**

Proportional-integral-derivative (PID) control is the most common control algorithm used in industry and has been universally accepted in industrial control. The popularity of PID controllers can be attributed partly to their robust performance in a wide range of operating conditions and partly to their functional simplicity, which allows engineers to operate them in a simple, straightforward manner [8]. The block diagram of PID controller is shown in figure 11.

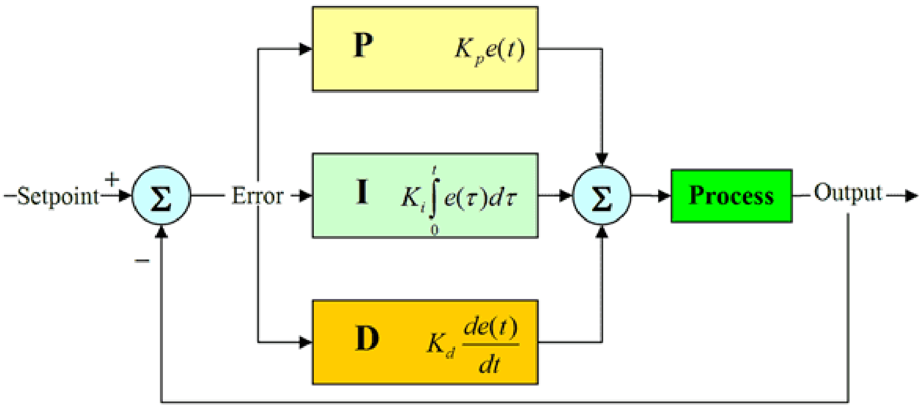

Figure 11: The Block Diagram of PID Controller.

As the name suggests, the PID algorithm consists of three basic coefficients; proportional, integral and derivative which are varied to get optimal response, the trial-and-error method was used to get optimal response.

- **Graphic User Interface LabVIEW**

The temperature control system needs an interface between the model and the user or operator, LabVIEW is a better solution, it is a virtual instrumentation software that allow the user to deal or monitor the process without the direct constant with the system. Figure 12 shows the GUI layout window for the system, this window provided with an empty field to enter the desired temperature value, the below field is to show the actual temperature value measured of the heater and the last field is to show the error value. This chart contains a heater temperature-monitoring graph. The chart contains two curves, red curve for actual value of the system response and the blue curve for the desired value.

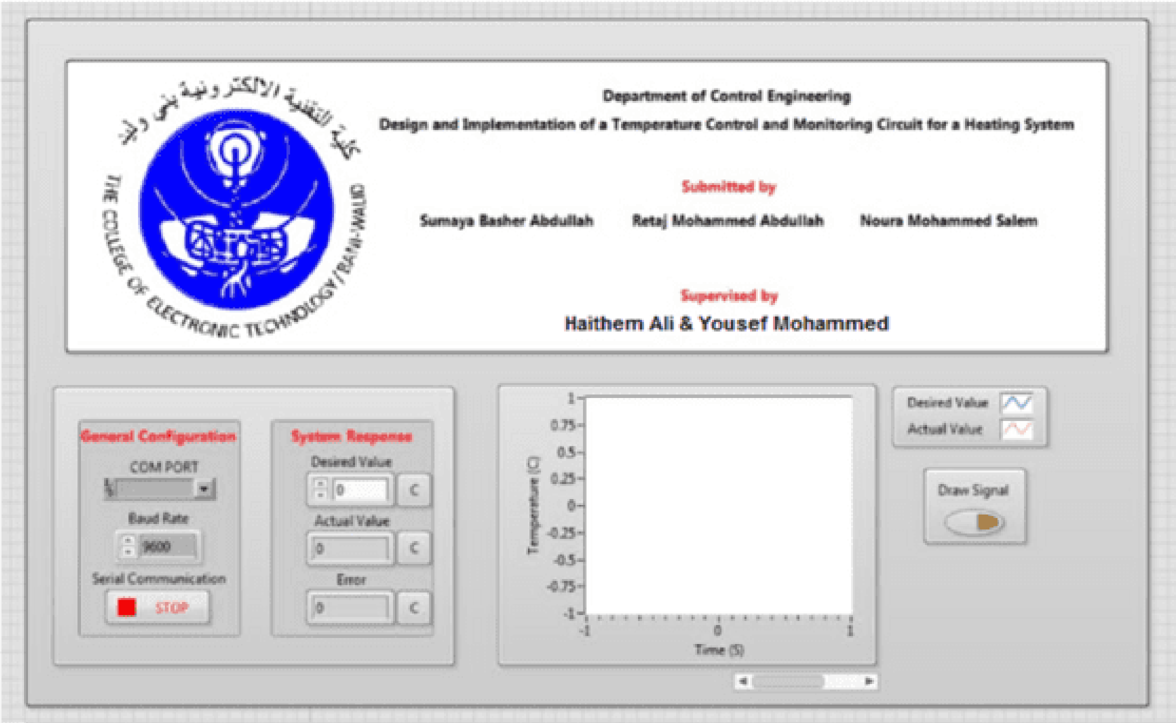

Figure 12: The LabVIEW GUI of the System

Results and discussion

In this section, the flow chart of this system will be shown through which the idea of the system works will be shown, followed by the consequences of several tests that have been applied to the system. The flow chart of this system is shown in figure 13.

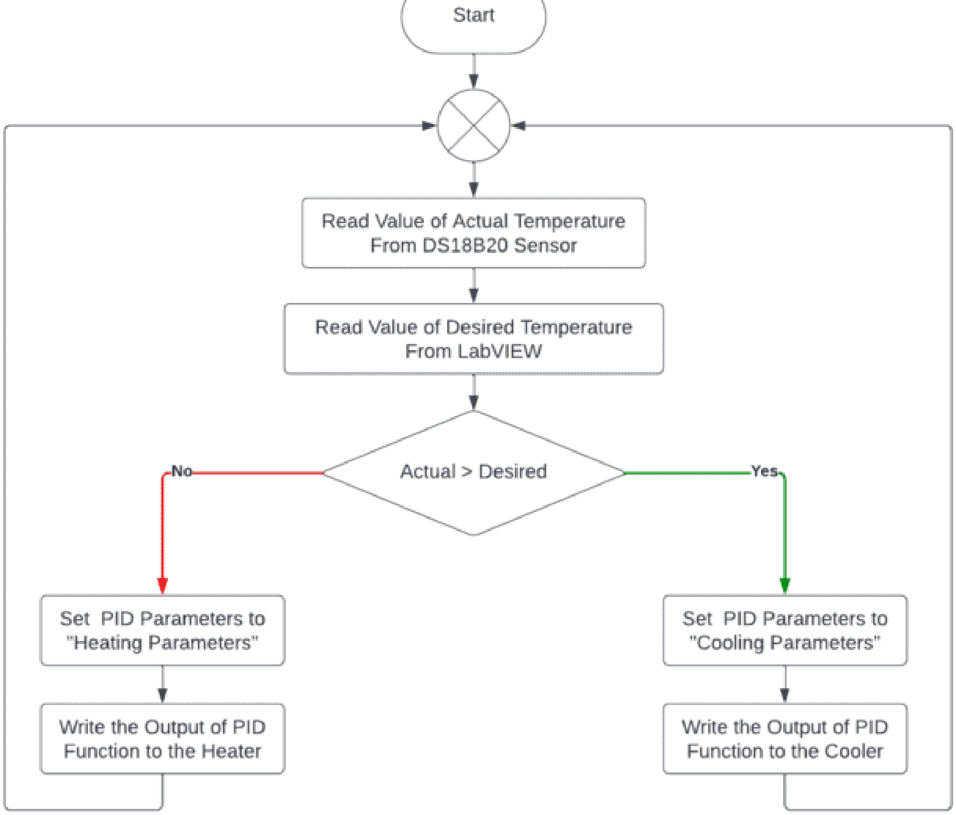

Figure 13: The Flow Chart of the system

From the figure 13, we notice that the system initially reads the actual temperature from the sensor and the required temperature through the LabVIEW program. After that, the system compares the real value and the required value to be able to make the final decision whether the system needs a cooling process or a heating process. During each cooling or heating process, the controller's parameters are loaded in the PID algorithm to obtain the best possible performance. The controller parameters were tuned in the case of cooling and heating using trial and error method.

In the first test, the required value was 50°C and the system was under the influence of room temperature 27°C, in this case the system will works on heating until it reaches the temperature. The figure 14 shows Response of Heating DV = 50°C.

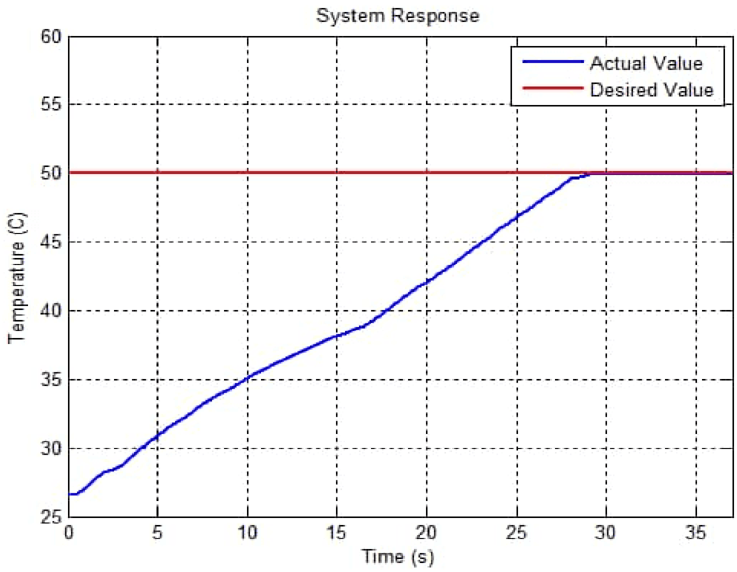

Figure 14: The System Response with Setpoint = 50 °C

We notice from the previous figure that the controller was able to drive the heater to track the temperature value required by the user. We also notice that there is no an overshoot in the temperature and this is one of the goals required from the beginning of the system design.

In the second test, the desired value was 25°C and the system temperature was 60°C, in this case the system will work on cooling until it reaches the desired temperature. The figure 15 shows Response of Cooling DV = 25°C.

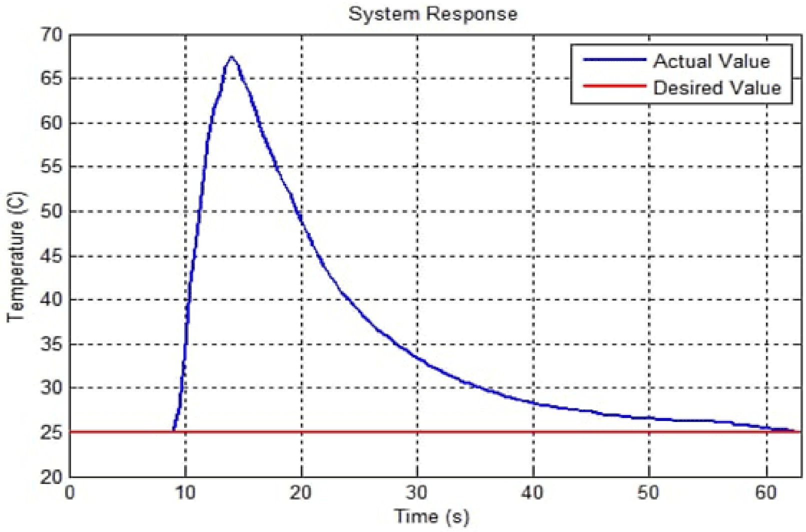

Figure 15: The System Response with Setpoint = 25 °C

We notice from the previous figure that the system detected a change in temperature after it was 60, it changed to 25, meaning that the process required to reach the required temperature is cooling, so the cooler was operated until it reached the required value without an undershoot in the temperature response.

In the third test, the system was under the influence of a room temperature of 27°C and the desired value was variable in this case the system will work on heating and cooling together, as shown in the following figure 16.

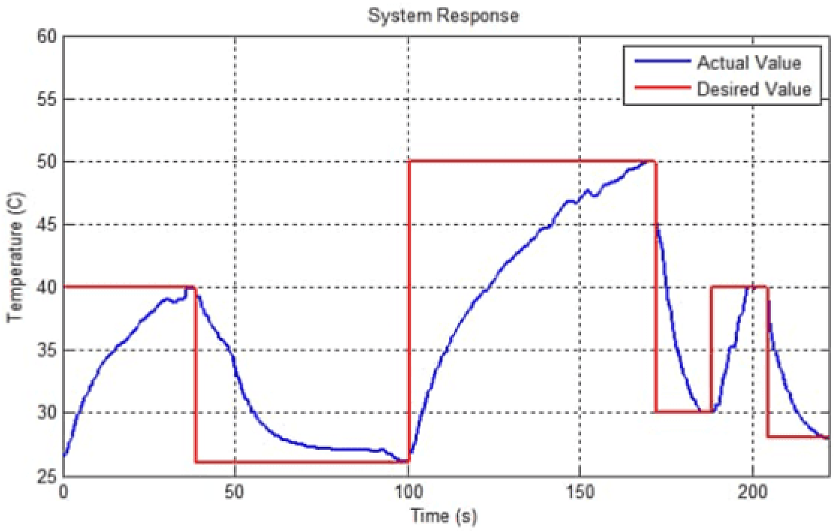

Figure 16: The System Response with Setpoint = Changed

Through the previous figure, the difficulty of the test was increased to ensure the efficiency of the controller in obtaining constantly changing temperatures over time, as we notice from the response that the system is successful in tracking the changing values required by the user, whether in cooling or heating operations, without an overshoot or undershoot in the output response.

In the fourth test, the system was stable at 25°C and we affected it by disturbing external heating for a certain time and then turned it off, in which case the system was able to adjust itself and return to the desired value. The figure 17 shows system response with external heating disturbance.

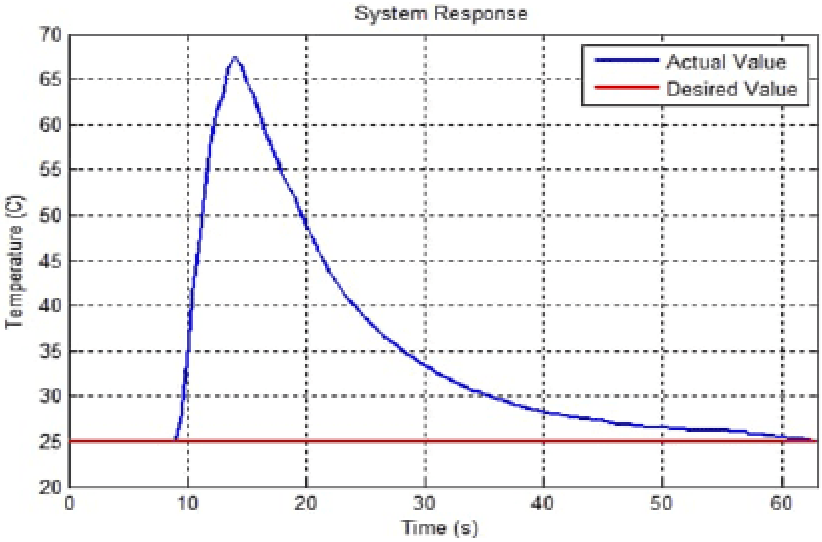

Figure 17: The System Response with External Heating Disturbance

The output response for the fourth test shown in the previous figure. Additional heat was applied into the room to test the system to detect and overcome any disturbance. As shown in the figure, the system was able to detect disturbance and overcome it without undershoot.

In the last test, the system was stable at a temperature of 50°C and we affected it by disturbing the external cooling for a certain time and then turned it off, in this case, also the system was able to adjust itself and return to the desired value. The figure 18 shows system response with cooling disturbance.

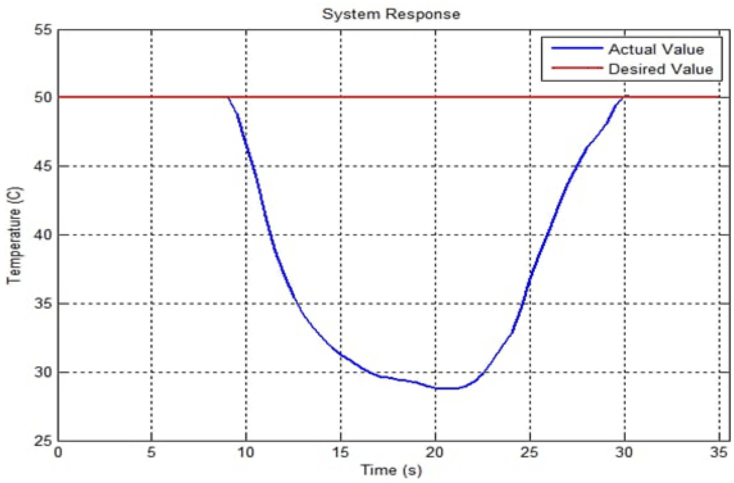

Figure 18: The System Response with External Cooling Disturbance

Conclusion

we conclude that the response of the system was good As it performed the required function of this project, we were able to control the starting current passing through the circuit and reduce the over-shot of the system's response in the case of heating and cooling, when the required temperatures were changed by the user "heating and cooling" at the same time, the system was able to track changes and give the required response at each value and when an external influence was added for a certain period of time, it had an effect on the system in the “cooling and heating” states, and the system was able to recover its normal state.

References

[1] Kyi Kyi Khaing, K. Srujan Raju, G. R. Sinha and Wit Yee Swe," Automatic Temperature Control System Using Arduino," Advances in Intelligent Systems and Computing. March 2020.

[2] Muhammad Asraf, H., K.A. NurDalila, A.W. Muhammad Hakim, and R.H. Muhammad Faizzuan Hon, "Development of Experimental Simulator via Arduino-based PID Temperature Control System using LabVIEW," Journal of Telecommunication, Electronic and Computer Engineering. 9 (1–5): 53–57 2017.

[3] Jaya B., Sushant P. and Siuli D., " Design and Development of Temperature Control Loop of Hair Dryer Using Arduino,". International Conference on Industry 4.0 Technology (I4Tech) Vishwakarma Institute of Technology, Pune, India. Feb 13-15, 2020

[4] Okpagu, P.E., and A.W. Nwosu, European Journal of Engineering and Technology 2 (7): 13–21 2016.

[5] Widhiada, W., D.N.K.P. Negara, and P.A. Suryawan, "Temperature Distribution Control for Baby Incubator System Using Arduino ATMega 2560,". Bali Indonesia 19 (20) part xv, 1748–1751 2017.

[6] DS18B20 Temperature Sensor Datasheet

[7] Khairul., A. B."Obsticale Avoidance Mobile Robot." B.SC. Dept. Elec. Eng. Pahang Univ., Malaysia.

[8] www.ni.com/ The PID Controller & Theory Explained