

THE EFFECTS OF PID PARAMETERS ON TEMPERATURE CONTROL OF UNINSULATED ENVIRONMENT

*Ahmet TOP^{*1}, Cem HAYDAROĞLU², Muammer GÖKBULUT³*

^{*1,3}Department of Electrical and Electronics Engineering, Faculty of Technology, Firat University, Elazığ, Turkey

²Department of Electrical and Electronics Engineering, Faculty of Engineering, Dicle University, Diyarbakır, Turkey

* Corresponding author; atop@firat.edu.tr

In this study, temperature of an uninsulated environment was controlled by PID for on-off and various parameter values using Arduino micro controller. A medium with high heat transmission and with no insulation materials on the junctions and the surface to prevent heat transfer into the external environment, is heated by an AC voltage-operated resistor. Temperature of the heated environment is taken with LM35 temperature sensor and converted to 0-10V industrial standards. Temperature information converted to voltage value was applied as an analogue input to the Arduino Due micro controller and resistance voltage was set via semiconductor switching path as a result of PID operation. Despite deteriorating experimental results such as heat loss, it is seen that the reference temperature is captured by the PID.

Key words: Temperature control, PID control, Arduino

1. Introduction

Temperature is a characteristic that expresses the value of heat taken on by an object. If an object takes on heat, its temperature rises; if it gives off heat, its temperature falls. Its measurement unit is centigrade degree (°C). It is a physical magnitude which can be controlled with low sampling frequencies. During this control, a device such as a resistance heater enabling heat discharge is used. A temperature sensor is utilized to convert ambient temperature to electrical magnitude and transfer it to a microcontroller. Upon this feedback, microcontroller controls the heater (and the system fan in some systems) [1].

Temperature control is a parameter control frequently used in the industry and in everyday life. It is essential that this control be efficiently conducted in simple processes such as increasing or maintaining the system's heat at a certain temperature in thermostats, as well as in various chemical processes, in industrial applications requiring very sensitive control of ambient temperature and in monitoring of environments such as incubators in medical rooms. The aim of temperature control can be to maintain the ambient temperature at a certain value, as well as to maintain it at a temperature range depending on the position and intended purpose of the designed system or facility. In each case, a desired value, a difference to be calculated from data obtained from the environment, and an efficient controller to process this difference will be available [2,3].

In a study, Bolat E. performed Matlab-simulink based real-time temperature control by using different types of auto-tune PID methods. Ziegler-Nichols step response method was observed to be a better method in proportional control, however, time-delay control produced better results in proportional + integral control [4].

Sefa İ. and Kahraman T. proposed an approach called “multipoint temperature measurement” to improve ambient temperature comfort in climatization. They measured temperatures from different points and sent the resulting values to a PIC18F452 processor via radio frequencies. They performed climate control using PI controller [5].

Aldemir A. and Hapoğlu H. conducted wireless temperature control experiments in a process simulator by using three different parameter adjustment methods that apply PID algorithm. They determined Kc, Ti and Td coefficients with three methods: Cohen-Coon, Ziegler-Nichols and trial-and-error. They concluded that trial-and-error is the ideal method to reach the best result [6].

In the present method, an uninsulated environment of 1 mm thick metal plates was prepared. Ambient temperature was measured with temperature sensor, transferred to the Arduino Due micro-controller, and the error between reference value and measured value was processed with on/off control through P, PI, PD and PID algorithms for various Kp, Ki and Kd values, and the firing duration of the semiconductor was adjusted. Ambient temperature was set to the desired value with a resistor whose voltage value varied depending on this firing duration, and results were examined for each control.

2. Control System

In general sense, the term control is used to mean to balance, direct or manage as desired. It is the performance of control processes without any direct human interference, by a decision-making mechanism established around the process intended to be controlled [7]. The reaction expected from the controlled system may be, in general, operating and stopping the system with on-off method, maintaining the system output at a value set by the user, or monitoring a user-detected change with minimal errors. Controllers can usually be grouped into two main categories according to the processing of error signal in the controller: two-position (on-off output) or continuous output controllers [8].

2.1. On-Off Control

In this control method, the system is either completely on or completely off around the reference value [9].

2.2. Automatic Control Forms

In automatic control logic, whatever control system is used as the controller, it must control the system continuously and accurately against the set value entered. Various control types are available to perform this sensitive study required in process control and minimize the error value [9]. These control types are Proportional + Integral (PI), Proportional + Derivative (PD) and Proportional + Integral + Derivative (PID) control. PID controllers are composed of three basic controllers and calculated as control signal $u(t)$:

$$u(t) = K_p \cdot e(t) + K_i \cdot \int_0^t e(t) \cdot dt + K_d \frac{d}{dt} e(t) \quad (1)$$

K_p is proportional coefficient, K_i is integral coefficient, K_d is derivative coefficient and $e(t)$ is error. This controller is generally effective on both transient and steady state control criteria. The relevant control parameter can be reset and used as PI or PD controller as well [8].

3. System Architecture

The architecture of microcontroller temperature control system, the general diagram of which is given in Figure 1, is composed of the components shown in the diagram below. Hardware components in this diagram are managed with microcontroller software.

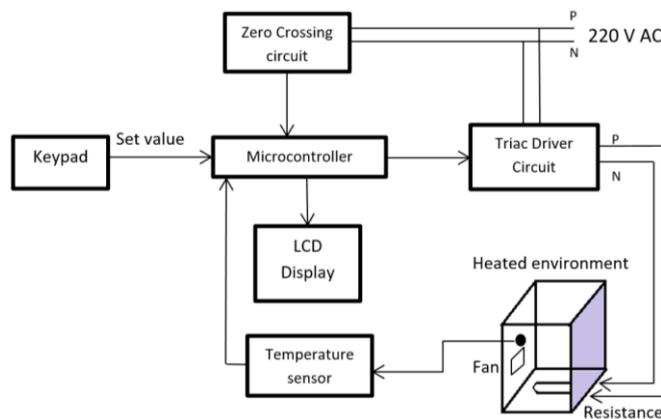


Figure 1. General diagram of microcontroller temperature control

3.1. Sensor and Microcontroller

Various types of sensors are available for temperature measurement. Regular sensors such as Thermocouple, PT100, thermistor and RTDs are widely used sensors [1]. In this study, a temperature sensor with analogue output shown in Figure 2 was used.

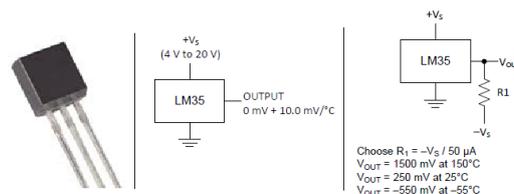


Figure 2. Temperature sensor

Temperature sensor has 4-20 V DC operating range and measures between -55°C and 150°C with 10 mV accuracy per degree [2]. This sensor produces 0 to 10 mV analogue outputs depending on our connection type. As microcontroller, Arduino Due microcontroller card -a AT91SAM3X8E microcontroller based development platform- was used.

3.2. Heated Environment and Heater

To observe the system reaction, a 17x30x31 cm structure in the form of a rectangular prism constructed from 1 mm thick metal plates was selected. The environment was not insulated so that the outcomes of deteriorating effects on the system could also be observed. A resistor was placed 4 cm high

from the prism floor. A 26x16 cm corrugated dry-air tube resistor with 46-ohm resistance, operating with 220V AC voltage and generating 1050 W power under nominal voltage condition was used to adjust the ambient temperature. The sensor was mounted next to the fan.

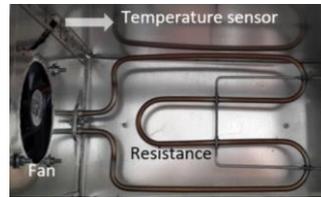


Figure 3. Interior of the heated environment and resistor

3.3. Dimmer Circuit

Dimmer circuit is a single-phase firing circuit. In Dimmer circuit, while one thyristor is used for half-wave control, two thyristors can be connected in an inverse-parallel mode in full-wave control [10]. In the present study, full-wave mains voltage was controlled for the voltage adjustment of resistor and, therefore, BT138 series Triac which has a voltage capability up to 600 V 12 A and can be used in such fields as motor control, interior lighting and heating control was used [11]. In order to isolate Arduinoile AC voltage, an integrated optocoupler was used to trigger Triac. This integrated optocoupler has a 400 V photo triac driver output. Control driver input can be performed with 5 V DC [12].

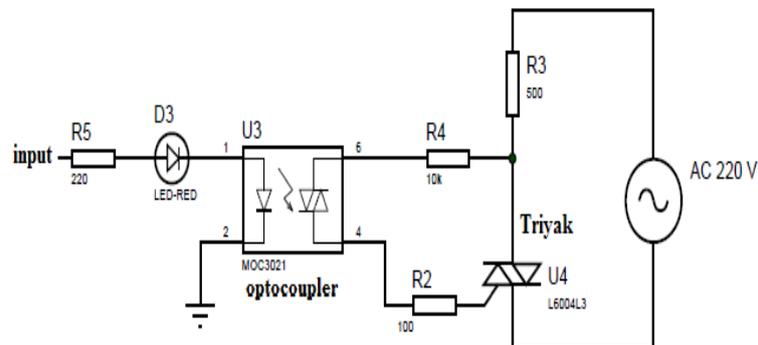


Figure 4. Dimmer circuit

3.4. Zero Crossing Circuit

In the control performed by using Triac, phase angle of the mains voltage was controlled and effective value of the output voltage was changed. Therefore, zero crossings of the mains signal were detected and "zero crossing pulses" were generated therefrom [13]. In a study conducted by Irmak E., Calpbiniçi A. and Güler N., they processed sinusoidal current and voltage signals through zero crossing detector to observe the delay between current and voltage and determined the time difference between them [14]. The zero crossing detector circuit built for this study is shown in Figure 5 and Figure 6.

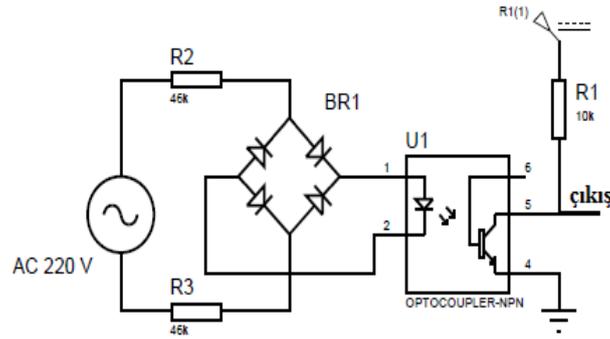


Figure 5. Zero crossing detector circuit

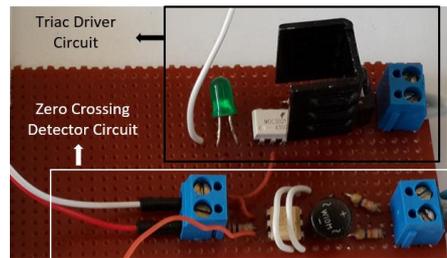


Figure 6. Zero Crossing Detector and Triac Driver Circuit

3.5. Set Input and Value Reading

3x4 keypad was used to enter the desired temperature value in the system. LCD screen was used to observe the set value entered when the system starts operating, and the set value and instantaneous measurements during operation. whole system is shown in Figure 7.

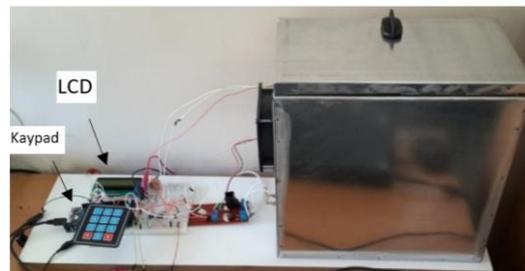


Figure 7. General View of the System

4. The Effects of On/Off and PID Controllers on Heat Control

In temperature control system, temperature range of ambient temperature to 100°C can be controlled. Experimental study results were taken from the serial port screen in the software interface of the microcontroller. These results were transferred to Matlab and plotted against time.

With the software generated by using microcontroller, on/off, proportional, proportional + integral, proportional + derivative and proportional + integral + derivative controls were performed. Effects of various parameter values on system reaction were studied with the controllers. Due to the absence of a cooler in system control, ambient temperature was taken as the initial temperature. Final temperature (set value) was determined as 60°C in all controllers.

Graphs of changing temperature of the resistor in the heated environment after application of full voltage (220 V AC) and half voltage (110 V AC) are given in Figures 8 and 9.

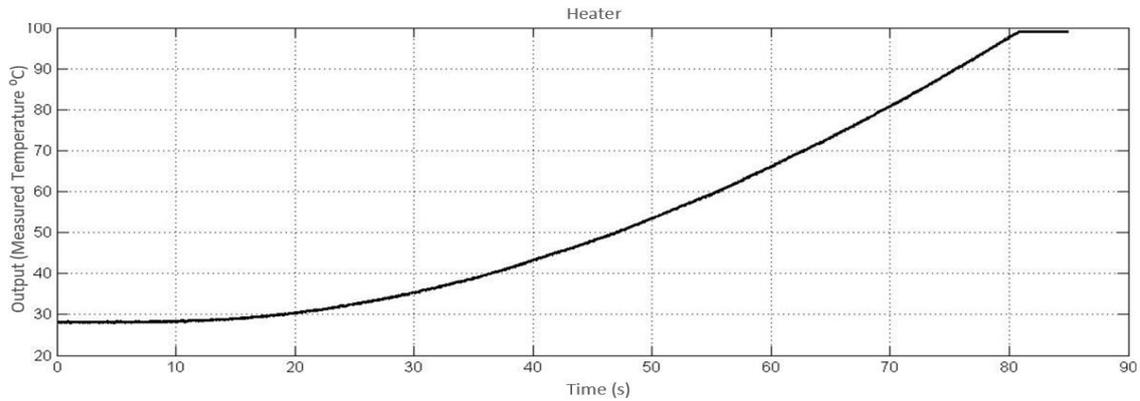


Figure 8. Output characteristic obtained from the application of 220 V input voltage to the heater

When 220 V AC voltage was applied to the heater, ambient temperature started increasing from 29°C and reached steady state at about 100°C. Time between transient state and steady state was measured as 81 seconds.

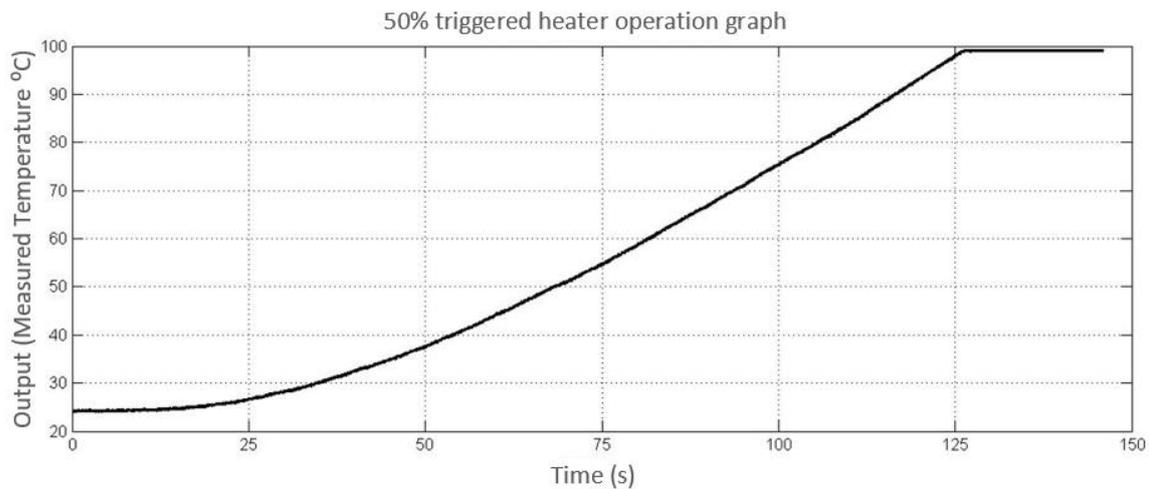


Figure 9. Output characteristic obtained from the application of 110 V input to the heater

When Triac driver circuit was triggered at 90°, the voltage applied to the heater was 110 V. In this case, it was observed to increase from ambient temperature to 100°C in 128 seconds.

4.1. On/Off Controller

Two on-off control applications were performed in the system.
1st measurement: Ambient temperature 31 °C, Set value: 60 °C, no hysteresis

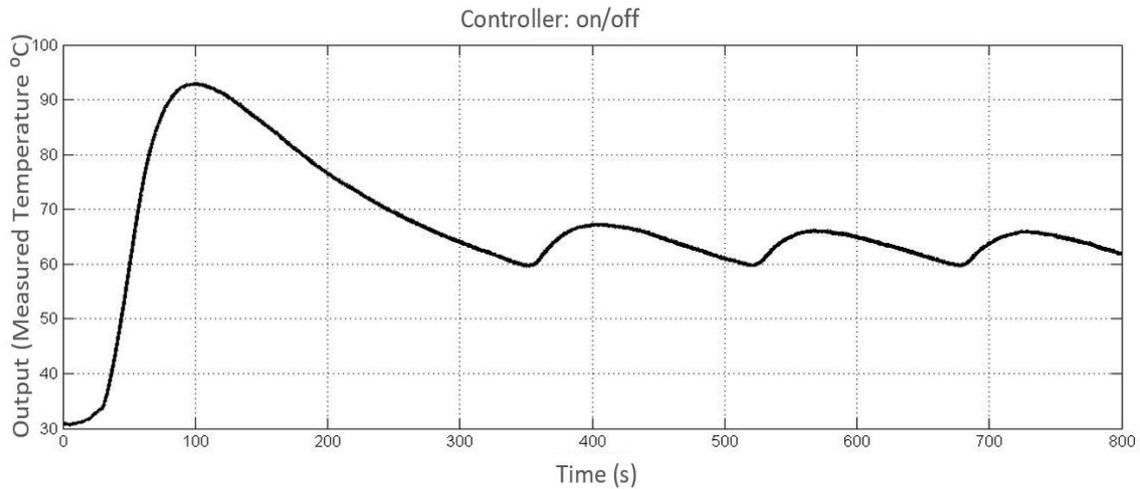


Figure 10. On/off controller, 31-60 °C, no hysteresis

In on/off control, the system switches to off-state when it reaches the reference value and to on-state when it falls below the reference value. Even if the system is switched off after reaching the reference value, the environment will be heated slightly more because the resistor temperature will not drop immediately. This is the reason of oscillation over the reference value in Figure 10. In this measurement, Rise time is 50.66 seconds, Maximum Overshoot is 33.05°C-55.08%, Peak time is 100.4 seconds and Settling time is 353 seconds.

2nd measurement: Ambient temperature 30 °C, Set value: 60 °C hysteresis: 3

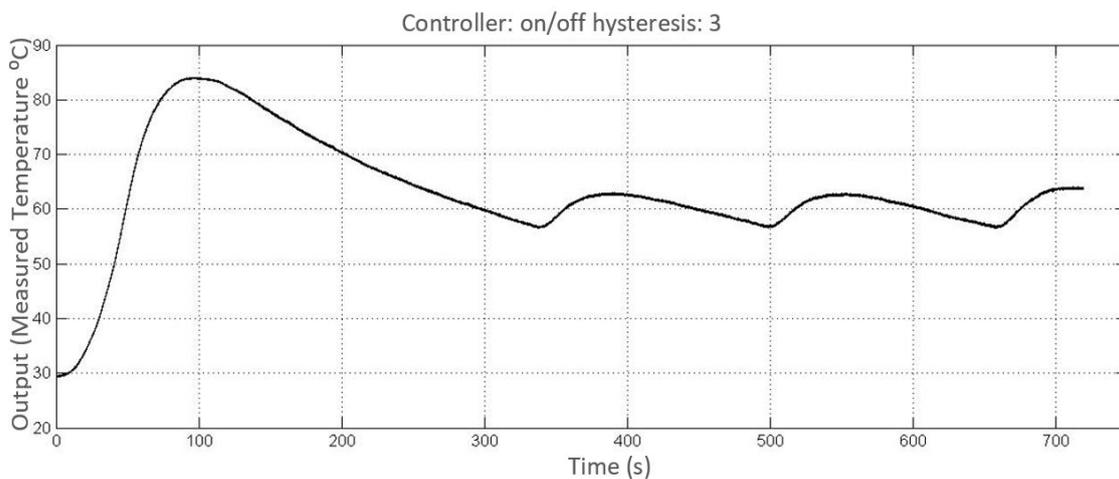


Figure 11. On/off controller, 30-60 °C, hysteresis:3

As seen in Figure 11, since a bandwidth both above and below the reference value was produced in case of hysteresis, not only overshoot decreased but also temperature became 60°C. In this measurement, rise time was 48.82 seconds, Maximum overshoot 24.12°C-40.2%, Peak time 97.21 seconds and Settling time 300.3 seconds.

4.2. Proportional (P) Controller

1st measurement: Ambient temperature 31 °C, Set value: 60 °C Kp:1

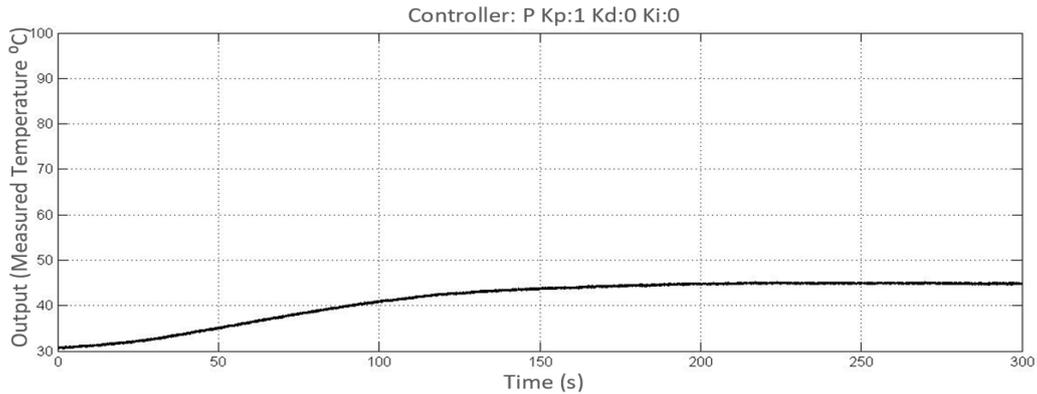


Figure 12. Proportional controller, 31-60 °C, Kp:1 temperature-time graph

When the proportional coefficient was adjusted to 1, ambient temperature became steady at 45°C after 200 seconds. A steady state error of 15°C occurred.

2nd measurement: Ambient temperature 31 °C, Set value: 60 °C Kp: 2.25

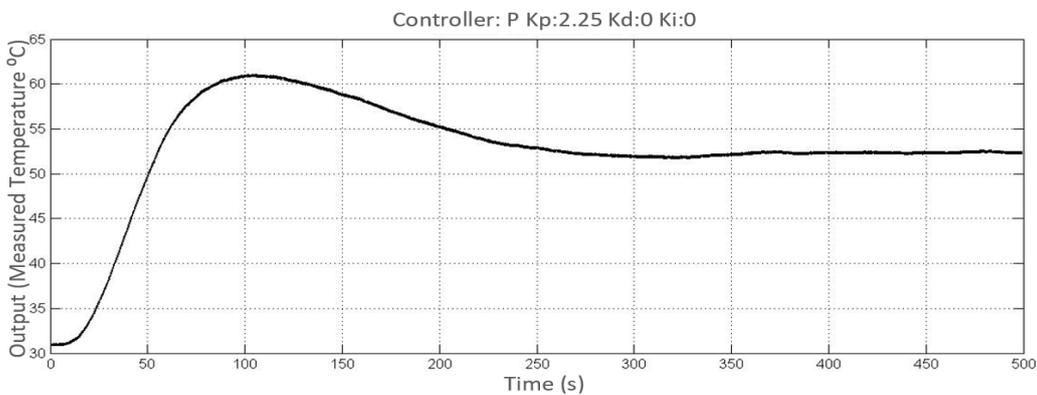


Figure 13. Proportional controller, 31-60 °C, Kp:2.25 temperature-time graph

When Kp coefficient was set to 2.25 in order to increase the rise time and to reduce the steady state error, rise time was 84.77 seconds, peak time 105.3 seconds, maximum overshoot 1.07°C - 1.78%, settling time 369.2 seconds and steady state error 7.57°C.

4.3. Proportional + Integral (PI) Controller

1st measurement: Ambient temperature 30 °C, Set value: 60 °C Kp:2.25 Ki:2

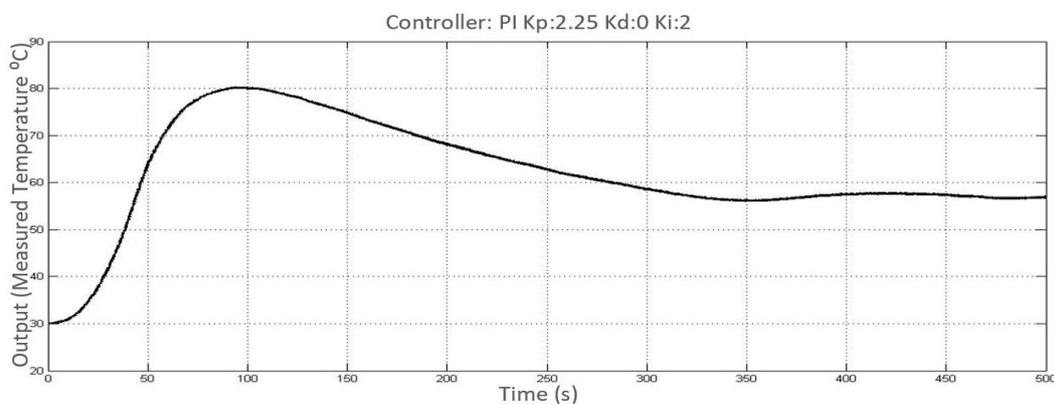


Figure 14. Proportional + Integral controller, 30-60 °C Kp:2.25 Ki:2 temperature-time graph

2nd measurement: Ambient temperature 23 °C, Set value: 60 °C Kp:2 Ki:0.02

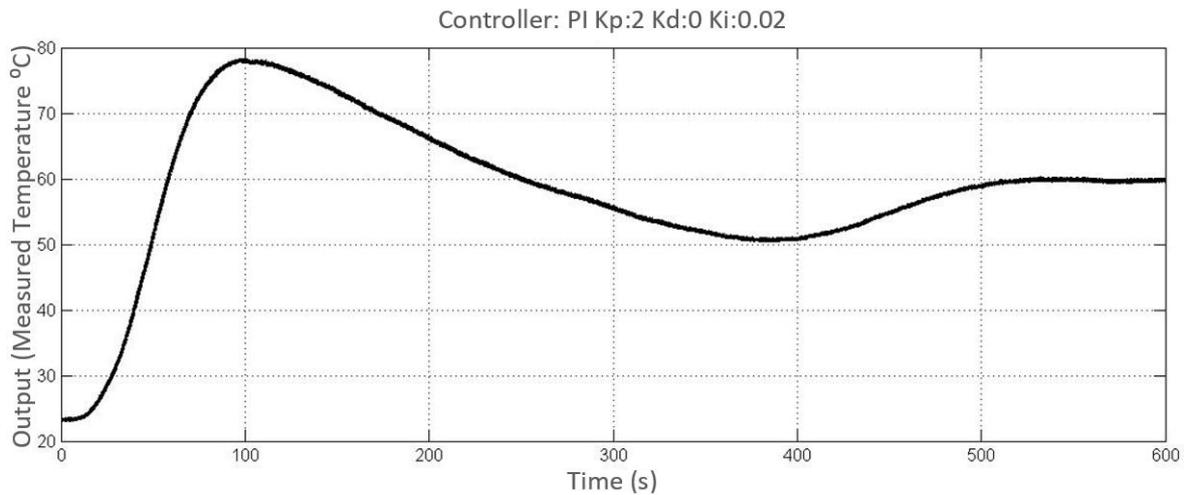


Figure 15. Proportional + Integral controller, 23-60 °C Kp:2 Ki:0.02 temperature-time graph

PI controller was used to eliminate steady state errors of the system in proportional control. Initially, when parameters were set to Kp:2.25 Ki:2, it was observed that rise time was 46.56 seconds, peak time was 94.99 seconds, maximum overshoot was 20.22°C - 33.7%, settling time was 370.1 seconds and steady state error was 3.47°C - 5.78%. When parameters were set to Kp:2 and Ki:0.02, rise time was 58.37 seconds, peak time was 96.32 seconds, maximum overshoot was 18.18°C – 30.3%, settling time was 534.2 seconds and steady state error was 0.09°C - 0.15%.

4.4. Proportional + Derivative (PD) Controller

1st measurement: Ambient temperature 32 °C, Set value: 60 °C Kp:3 Kd:0.2

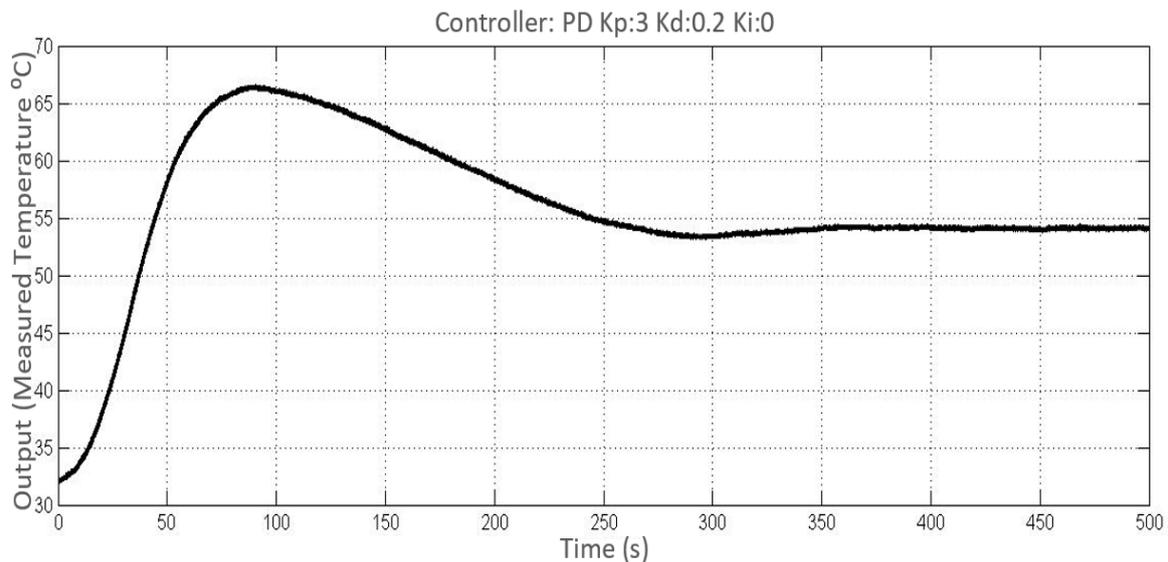


Figure 16. Proportional + Derivative controller, 32-60 °C, Kp:3 Kd:0.2 temperature-time graph

2nd measurement: Ambient temperature 28 °C, Set value: 60 °C Kp:3 Kd:4

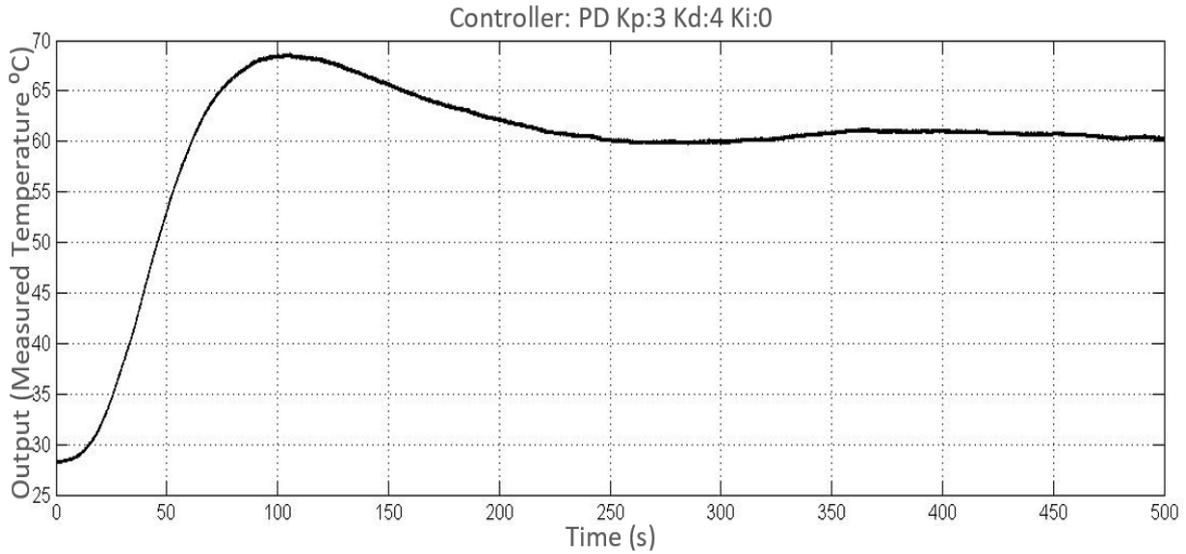


Figure 17. Proportional + Derivative controller, 28-60 °C, Kp:3 Kd:4 temperature-time graph

Derivational effect was used to fix the oscillations and overshoots occurring in the transient regime of the system. As shown in the figures above, steady state was not set to zero, but the overshoot that reached 75-80°C in proportional and proportional + integral control did not exceed 70°C here due to derivative effect.

4.5. Proportional + Integral + Derivative (PID) Controller

1st measurement: Ambient temperature 30 °C, Set value: 60 °C Kp:1.5 Kd:0.02 Ki:0.02

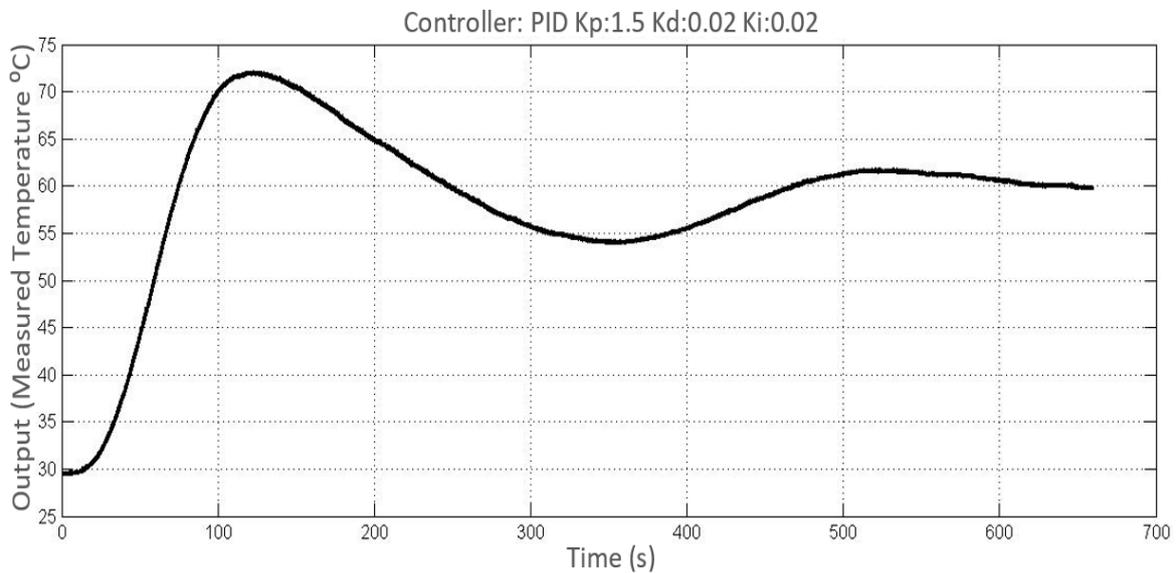


Figure 18. Proportional + Integral + Derivative controller, 30-60 °C, Kp:1.5 Kd:0.02 Ki:0.02 temperature-time graph

2nd measurement: Ambient temperature 29 °C, Set value: 60 °C Kp:3 Kd:4 Ki:0.01

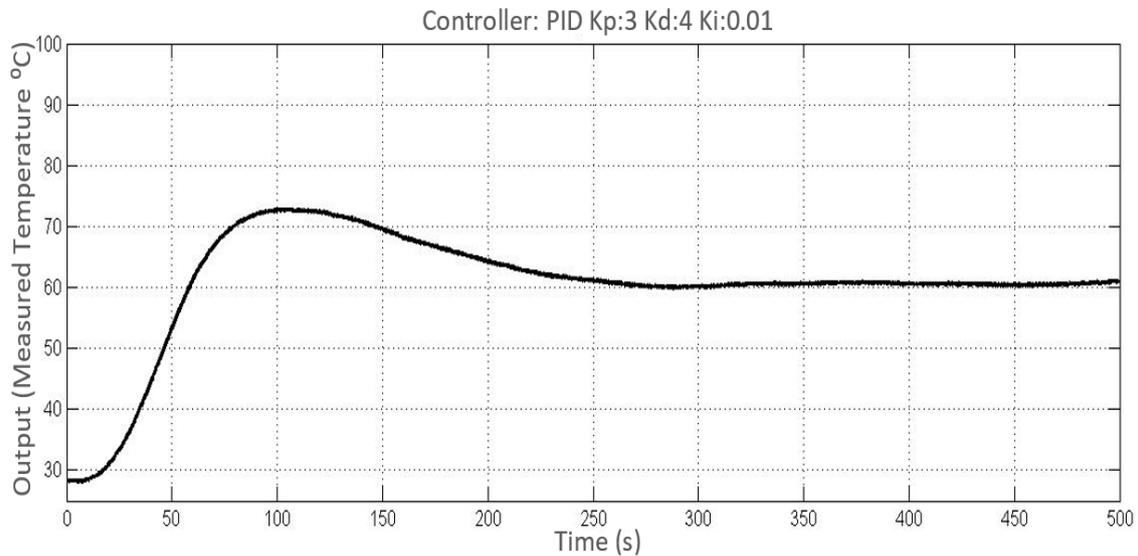


Figure 19. Proportional + Integral + Derivative controller, 29-60 °C, Kp:3 Kd:4 Ki:0.01 temperature-time graph

When parameters were chosen as Kp:1.5 Kd:0.02 Ki:0.02 in PID control, it oscillated for a while and then reached the set value in steady state, as shown in Figure 18 and 19. When parameters were selected as Kp:3 Kd:4 Ki:0.01, it did not oscillate and reached the set value without oscillation and with only one overshoot.

5. Conclusion

In the present study, the resistance used to heat the uninsulated environment was controlled with a Triac driver circuit for on/off, P, PI, PD and PID controller outputs obtained via software by Arduino microcontroller card, and the ambient temperature was maintained at the desired reference value. Reference value was taken as 60°C for all applications. Because measurements are made on different days and cooling is not used, ambient temperatures may vary. As predicted, system response oscillated over reference value in on/off control. When only proportional control was used and Kp coefficient was kept low at the beginning, ambient temperature reached steady state with a 25.28% error without reaching the reference value. When this coefficient was increased, 1.78% overshoot was seen in the transient regime, it then dropped below the reference value and reached steady state with a 7.57°C error. Proportional + integral control was used to fix the steady state error. Steady state error of the system was eliminated and temperature became steady at 60°C with this control; however, a 30.3% overshoot occurred in transient state of the system. When proportional + derivative control was used, it was observed that the overshoot in transient state dropped to 10.75% but a 5.83°C steady state error occurred. Proportional + integral + derivative control was used to fix both steady state and transient state of the system and to reach the desired reference value within the shortest time with the highest accuracy. By using PID control, reference value was captured with 21% overshoot and 0.48°C error. In conclusion, the effects of PID parameters on the system were examined and PID controller was observed to produce the ideal results.

6. References

- [1] Doğan, İ., Microcontroller Based Temperature Monitoring And Control, Newnes, England, 2002 171-219
- [2] Savas, Y., Sayısal Göstergeli Sıcaklık Ölçme ve Kontrol Cihazının Tasarım ve Yapımı, Yüksek Lisans Tezi, G. Ü. Fen Bil. Enstitüsü, Ankara, 1988, 1-44.
- [3] Kalender, O., MC 6802 Tabanlı Oransal Sıcaklık Kontrol Devresinin Tasarımı ve Yapımı, Yüksek Lisans Tezi, G. Ü. Fen Bil. Enstitüsü, Ankara, 1991, 1-77.
- [4] Bolat E., Real Time Temperature Control of Oven Using Matlab-SIMULINK, Proceedings of the 11th WSEAS International Conference on SYSTEMS, Greece, 2007
- [5] Sefa İ., Kahraman T., Klimalar İçin Yeni Bir Kontrol Sistemi, *Gazi Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi*, 2007, Cilt 22, No 3, 339-347
- [6] Aldemir A., Hapoğlu H., Kablosuz Sıcaklık Kontrolü İçin PID Ayarlama Yöntemlerinin Karşılaştırılması, *politeknik dergisi*, 2016, cilt 16, sayı 1, sayfa 9-19
- [7] Kıncay O., *Yapılarda otomasyon ve enerji yönetimi*, otomatik kontrol ders notu, Yıldız Teknik Üniversitesi, İstanbul
- [8] Gökbulut M., *Kontrol sistemlerinin analiz ve tasarımı*, ders kitabı seçkin yayıncılık, Ankara
- [9] Gürsel K., Ünlversal gelişmiş kontrol cihazları, PID parametrelerinin ayarlanması, Auto-Tune özellikleri, Elimko Elektronik İmalat ve Kontrol LTD.ŞTİ, Ankara, 18s
- [10] Ak, M., Tuna, M. Ve Ergün Amaç A., Bir Fazlı Tetikleme devreleri için eğitim amaçlı kullanıcı arayüzü tasarımı, *UMES'07-Ulusal Teknik Eğitim, Mühendislik ve Eğitim Bilimleri Genç Araştırmacılar Sempozyumu*, 2007, 309-312
- [11] NXP Semiconductors BT138 series Triacs datasheet.
- [12] Texas Instruments moc3020 thru moc3023 optocouplers/optoisalatorsdatasheet, soes025a – october 1986 – revisedapril ,1998
- [13] Tıgılı, İ.C., Mutfak tipi bir mikrodalga fırının mikroişlemci ile kontrolü, *BAÜ Fen Bil. Enst. Dergisi*, 2014, Cilt 16(2) 87-99
- [14] İrmak E., Calpbıncı A., Güler N., Orta Ölçekli Bir İşletmenin Enerji İzleme Sisteminin Tasarlanması, *Pamukkale Üniversitesi Mühendislik Bilimleri Dergisi*, 2012 Cilt 18, Sayı 2, Sayfa 123-131