Chapter 12
iTCLab Temperature Monitoring and Control System Based on PID and Internet of Things (IoT)

Basuki Rahmat
Universitas Pembangunan Nasional "Veteran" Jawa Timur, Indonesia

Minto Waluyo
Universitas Pembangunan Nasional "Veteran" Jawa Timur, Indonesia

Tuhu Agung Rachmanto
Universitas Pembangunan Nasional "Veteran" Jawa Timur, Indonesia

Mohamad Irwan Afandi
Universitas Pembangunan Nasional "Veteran" Jawa Timur, Indonesia

Ni Ketut Sari
Universitas Pembangunan Nasional "Veteran" Jawa Timur, Indonesia

Helmy Widyantara
Institut Teknologi Telkom Surabaya, Indonesia

Harianto Harianto
Universitas Dinamika, Indonesia

ABSTRACT

The rapid increase in applications that combine modern concepts and innovations, due to the development of the internet of things (IoT) and cloud computing around the world, make all areas of life continue to move towards an advanced and intelligent society. This innovation continues to enter almost all fields, ranging from simple to complex innovations. In this chapter, IoT is used as a means for tuning PID parameters, when the error does not converge to zero. The experimental results show that the PID parameter tuning process can be done through IoT. And the results are quite encouraging, as an alternative way of tuning PID parameters.

DOI: 10.4018/978-1-6684-5629-3.ch012

INTRODUCTION

Internet of things (IoT) is a developing region in which billions of keen objects are associated with one another utilizing the web to share information and assets (Rahmat, Moeljani, Widjajani, Sudiyarto, & Harianto, 2021). IoT innovation permits objects around us to be associated with the internet network. Where each object associated with the internet can be accessed anytime and anyplace. For illustration, able to remotely turn on and off machines at domestic (lights, TVs, stoves, radiators, etc.) as long as the equipment is associated with the IoT cloud and an online association is accessible. In general, the IoT architecture consists of an Application Layer, Middleware Layer, Network Layer, and Physical Layer (Rahmat et al., 2021, and Ravidas, Lekidis, Paci, & Zannone, 2019). The basic system of the IoT consists of 3 components, namely hardware or physical (things), internet connection, and cloud data center as a place to store or run the application (Rahmat et al., 2021). Some of the research on IoT and its application can be read in the following papers: Internet of things based attendance system design and development in a smart classroom (Eridani, Widianto, Windasari, Bawono, & Gunarto, 2021), Simplified automatic VAR/Power factor compensator using fuzzy logic based on internet of things (Luqman, Lestari, & Setiawan, 2019), Development of a smart parking system based on internet of things using object-oriented analysis and design method (Maulana, Adhy, Bahtiar, & Waspada, 2020), Smart Greetthings: Smart Greenhouse Based on Internet of Things for Environmental Engineering (Sofwan et al., 2020), Development of Controller for Internet of Things Based Anti Pollution Smart Toll Gate System (Syafei, Afiq, Wahyudi, & Hidayatno, 2020), LoRa Gateway as Internet of Things (IoT) Infrastructure Components on Undip Vocational School (Tadeus, Yuniarto, & Mangkusasmito, 2020), and lastly Door and light control prototype using Intel Galileo based Internet of Things (Windarto & Eridani, 2017).

In this paper, IoT is used as an alternative tuning of the proportional integral and derivative (PID) parameters of the internet-based temperature control lab (iTCLab) temperature control system. iTCLab is a temperature control kit for feedback control applications with an ESP32 Microcontroller, LED, two heaters, and two temperature sensors. The heater power output is adjusted to maintain the desired temperature set-point. Thermal energy from the heater is transferred by conduction, convection, and radiation to the temperature sensor. Heat is also transferred from the device to the environment.

This iTCLab kit is inspired by BYU (Brigham Young University) TCLab Products (BYU, 2018), one of the private campuses in Provo, Utah United States. This iTCLab kit is a miniature control system in a pocket that can be used as a practical IoT learning package, IoT programming, and IoT-based control system practice. This kit can also be used to learn system dynamics and control systems, Arduino and Python programming, Machine Learning programming, and others. One of the control system strategies that can be applied to this iTCLab Kit is PID control.

The most famous control system in the industry is the PID. PID combines three proportional, integral, and derivative control actions. Each of these control actions has certain advantages, where the proportional control action has the advantage of a very fast rise time, the integral control action has the advantage of reducing errors, and the derivative control action has the advantage of reducing errors or reducing overshoot. The purpose of combining these three control actions is to produce output with fast research time and small errors. Several studies on the application of PID control systems can be read in the following papers: PID controller tuning by differential evolution algorithm on EDM servo control system (Andromeda, Yahya, Samion, Baharom, & Hashim, 2013), Design of Adaptive PID Controller for Fuel Utilization in Solid Oxide Fuel Cell (Darjat, Sulistyo, Triwiyatno, & Julian, 2018), Design and simulation of PID controller for lower limb exoskeleton robot (Munadi, Nasir, Ariyanto, Iskandar, & Setiawan, 2018), Design of Gain-Scheduled Fuzzy PID Controller for AFR Control System of SI-Based Motorcycle Engine Model (Panjaitan, Kurniahadi, Triwiyatno, & Setiawan, 2020), and Development of hovercraft prototype with stability control system using PID controller (Riyadi, Rahmando, & Triwiyatno, 2017).

The PID controller in its work automatically adjusts the control output based on the difference between the set-point (SP) and the measured process variable (PV), as the control error e(t). The controller output value u(t) is transferred as a system input. Each relationship used is as shown in Equations (1) and (2) (BYU, 2018).

$$e(t) = SP - PV \tag{1}$$

$$u(t) = u_{bias} + K_c e(t) + \frac{K_c}{\tau_I}\int_0^t e(t)\,dt - K_c \tau_D \frac{d(PV)}{dt} \tag{2}$$

$$u(t) = u_{bias} + K_c e(t) + \frac{K_c}{\tau_I}\sum_{i=1}^{n_t} e_i(t)\Delta t - K_c \tau_D \frac{PV_{n_t} - PV_{n_t - 1}}{\Delta t} \tag{3}$$

Equation (2) when expressed in digital form as shown in Equation (3) (BYU, 2018).

From Equation (3) it can be seen that three parameters determine the success of the control process, namely gain Kc, integral time constant I (τ_I), and derivative time constant D (τ_D). The process of searching or setting or tuning to obtain the best Kc,

τ_I, and τ_D values is generally called the tuning process. In this paper, a remote tuning process method is proposed using an internet connection via the MQTT protocol.

PID controller setup generally involves controlling four variables:

- Rise time: the amount of time it takes for the initial output of the system to rise past 90% of its desired value.
- Overshoot: the number by which the initial response exceeds the set-point value.
- Resolving time: the amount of time it takes for the system to converge to the set-point value.
- Steady-state error: the measured difference between the system output and the set-point value.

The purpose of the PID controller is to take the input value and maintain it at a certain set point over time. However, if the values for the three PID parameter controller loops are selected incorrectly, the system will become unstable through one of the numbers of failure modes. Usually, it involves output deviant with or without oscillation and is limited by the physical characteristics of the control mechanism, including actuator disconnection, sensor and encoder burnout, etc.

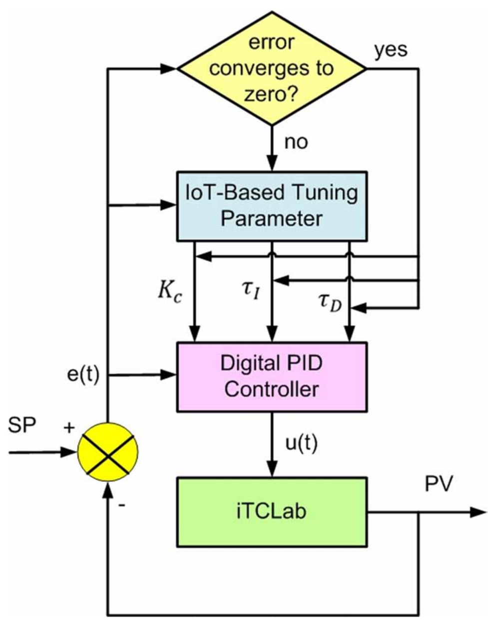
BACKGROUND

A control system can be said to be perfect if it can perform disturbance rejection (capable of returning the process that has been affected by disturbances back to a stable state) or also able to overcome the tracking set-point (the controller can produce a stable state even though we change the set point value). One of the tasks of the controller component is to reduce the error signal, namely the difference between the value of the set-point (SP) and the measured process variable (PV). This is consistent with the objective of the control system to get the measured process variable (PV) equal to the value of the set-point (SP). The faster the system reaction follows the set-point and the smaller the error that occurs, the better the performance of the applied control system. If the difference between the set-point value and the value of the measured process variable is relatively large, a good controller should be able to observe this difference to immediately produce appropriate control actions to affect the plant. Thus, the system rapidly changes the plant output until the difference between the set-point (SP) and the measured process variable (PV) is as small as possible. This is where the PID parameter tuning plays an important role to produce the best control performance. So far, the parameter tuning process is only carried out in places where the control process is carried out. This paper proposes a remote PID parameter tuning process using Internet of Things (IoT) technology. Thus, this study provides an alternative solution to the PID parameter tuning process, to produce the best control system performance.

METHODOLOGY

As explained above, the basic Internet of Things (IoT) system consists of 3 things, namely hardware or physical (objects), an internet connection, and a cloud data center as a place to store or run applications. This paper proposes a PID parameter tuning technique through IoT. The plant used for testing is the internet-Based Temperature Control Lab (iTCLab) kit. Cloud IoT as the MQTT Broker we use is hivemq.com. The port and protocol information from the broker will be used as the setting of the microcontroller program on the iTCLab system. The illustration of the PID parameter tuning system architecture in the iTCLab Temperature Control System is shown in Figure 1. While the temperature control method via IoT uses PID on the iTCLab Temperature Control System as shown in the figure 1. From this control method, the authors intended to use general PID parameter tuning using the Ziegler-Nichols method or a trial using arbitrary parameter values. When the

Figure 1.



output of the iTCLab system does not immediately enter the expected set-point or the error does not converge to zero, then another tuning alternative is needed. In this study, remote tuning is proposed using IoT technology via an internet connection.

The three PID parameters are played by being given intuitively the values that are expected to produce the best control performance, which can be given directly in the parameter settings via a cellphone using the IoT MQTT Panel. If the output error decreases or the output approaches the set-point the parameter values are retained. The advantage of tuning parameters through IoT is they can be controlled and monitored remotely. The same technique can be applied to general control systems using PID controllers for solving other cases. This technique, apart from being an alternative solution for remote control and monitoring, also presents its challenges for further development.

SOLUTIONS AND RECOMMENDATIONS

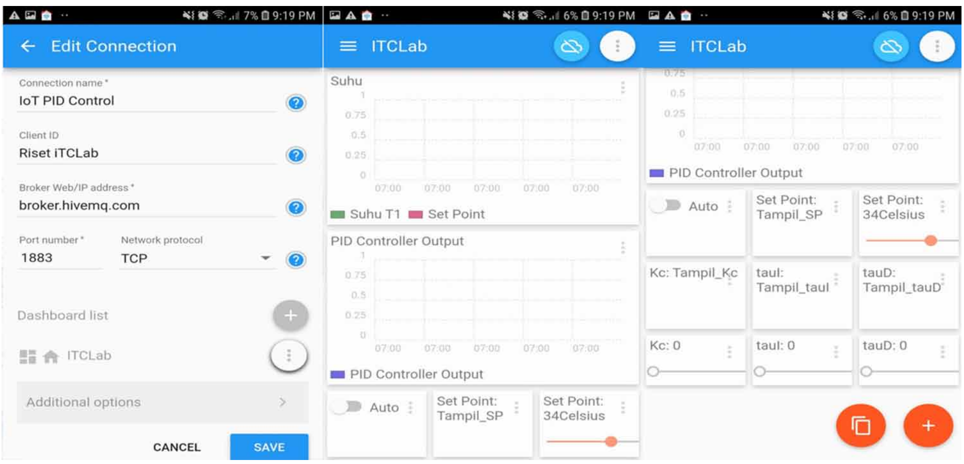
The kit used for testing in this experiment is the internet-based temperature control lab (iTCLab) kit. The iTCLab is a printed circuit board (PCB) shield that connects to an ESP32 microcontroller. The iTCLab shield has two transistors as heaters and two LM35 temperature sensors. The process exhibits second-order dynamics and the two adjacent heaters create a compact multivariate control system. The ESP32 microcontroller includes a 10-bit analog to digital converter (ADC) to measure the voltage of the temperature sensors in 1024 (2^{10}) discrete analog levels and Pulse Width Modulation (PWM) with 256 (2^8) levels to change the output to the heaters and LED.

The transistor heaters are TIP120 NPN bipolar junction transistors (BJTs) in a TO-220 package. These transistors are commonly used in audio, power, and switching applications but not commonly as heaters. The TIP120 can act as both the switch and the heater. The two temperature sensors on the iTCLab are standard LM35 with an output voltage (mV) that is linearly proportional to temperature (+ 10-mV/°C Scale Factor) and no requirement for calibration. Typical sensor accuracy is 0.5°C Ensured Accuracy (at 25°C).

As a safety and equipment protection precaution, the ESP32 microcontrollers come preprogrammed to shut off the heaters if the temperature rises above 60°C. The heaters are powered by a 12V 2A power supply for a maximum power output of 24W. A USB cable connects the ESP32 to a computer for serial data communication. One TIP120 heater and one LM35 sensor are connected and with a thermal heat sink attached to the TIP120 transistor. The two heater units are placed in proximity to each other to transfer heat by convection and thermal radiation.

Furthermore, the program that must be embedded in the ESP32 microcontroller on the iTCLab system is adjusted to the settings provided by the MQTT broker or IoT Cloud. The MQTT broker used in this experiment is using the Public HiveMQ MQTT broker, hivemq.com. HiveMQ has a dashboard so we can see the amount of traffic on this broker. HiveMQ also maintains a list of MQTT client libraries that can be used to connect to HiveMQ. To be able to access this broker, we use broker settings: broker.hivemq.com, TCP port: 1883, or web socket port: 8000. An example of its use on the IoT MQTT panel is shown in figure 2 as well as the program used for tuning the PID parameters on this iTCLab Kit which is shown in the following coding https://io-t.net/itclab/files/09-IoT_PID_Control.ino.

Figure 2. IoT MQTT panel



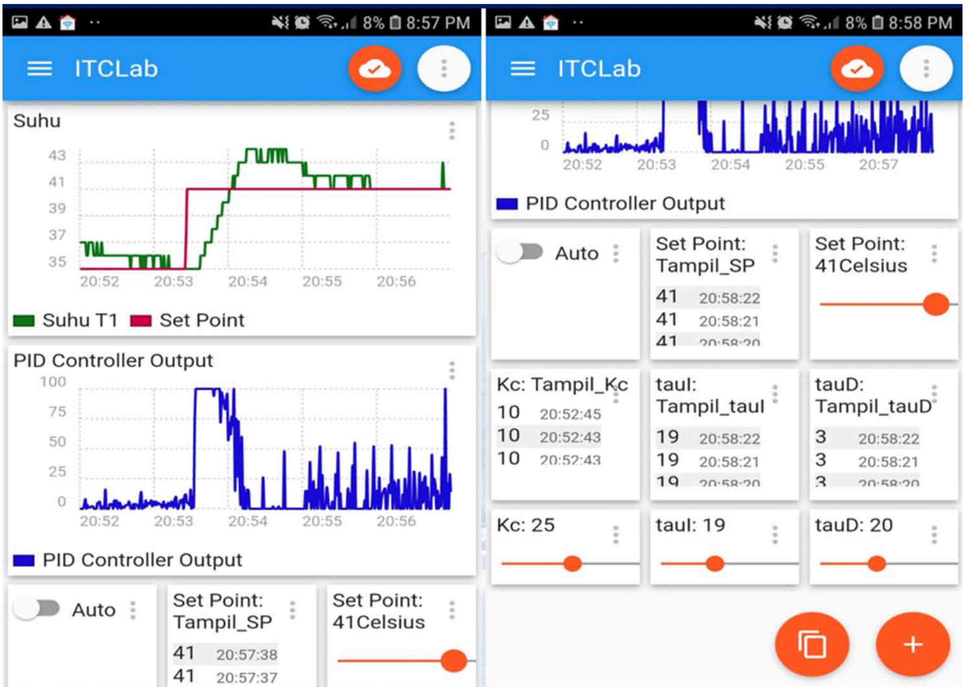
Some important code snippets include:

```
//Initial Setting
#include <WiFi.h>
#include <PubSubClient.h>
#include <Arduino.h>
const char* ssid = “wifi”; // Enter your WiFi name
const char* password =  “Password”; // Enter WiFi password
#define mqttServer “broker.hivemq.com”
#define mqttPort 1883
WiFiServer server(80);
WiFiClient espClient;
```

```
PubSubClient client(espClient);
// PID Controller
float pid(float sp, float Kc, float tauI, float tauD, float pv,
float pv_last, float& ierr, float dt) {
  float KP = Kc;
  float KI = Kc / tauI;
  float KD = Kc*tauD;
  // upper and lower bounds on heater level
  float ophi = 100;
  float oplo = 0;
  // calculate the error
  float error = sp - pv;
  // calculate the integral error
  ierr = ierr + KI * error * dt;
  // calculate the measurement derivative
  float dpv = (pv - pv_last) / dt;
  // calculate the PID output
  float P = KP * error; //proportional contribution
  float I = ierr; //integral contribution
  float D = -KD * dpv; //derivative contribution
  float op = P + I + D;
  // implement anti-reset windup
  if ((op < oplo) || (op > ophi)) {
    I = I - KI * error * dt;
    // clip output
    op = max(oplo, min(ophi, op));
  }
  ierr = I;
  Serial.println("sp="+String(sp) + " pv=" + String(pv) + "
dt=" + String(dt) + " op=" + String(op) + " P=" + String(P) + "
I=" + String(I) + " D=" + String(D));
  return op;
}
```

Finally, the results of controlling using a PID controller, and tuning PID parameters remotely using IoT are shown in figure 3. From the control results displayed on the cellphone using the IoT MQTT Panel, it can be seen that the temperature output was overshoot, but then convincingly went to the set-point.

Figure 3.



CONCLUSION

It has been tested in an internet-Based Temperature Control Lab kit (iTCLab). IoT is used as a means of setting parameters for a Proportional Integral Derivative (PID) controller when the error does not converge to zero. The proposed method can be used as an alternative to manually tuning PID parameters remotely by utilizing IoT technology. First, a program is created that must be embedded in the iTCLab Kit. Then the settings are made on the cellphone using the IoT MQTT Panel. The experimental results show that the PID parameter tuning process can be done through IoT. And the results are quite encouraging, as an alternative way of setting PID parameters.

REFERENCES

Andromeda, T., Yahya, A., Samion, S., Baharom, A., & Hashim, N. L. (2013). PID controller tuning by differential evolution algorithm on EDM servo control system. *Applied Mechanics and Materials*. doi:1 doi:0.4028/www.scientific.net/AMM.284-287.2266

BYU. (2018). Apmonitor.com. Proportional Integral Derivative (PID) Control. *AP Monitor* http://apmonitor.com/pdc/index.php/Main/ProportionalIntegralDerivative

Darjat, D., Sulistyo, S., Triwiyatno, A., & Julian, E. (2018). Design of Adaptive PID Controller for Fuel Utilization in Solid Oxide Fuel Cell. *Proceedings - 2018 5th International Conference on Information Technology, Computer and Electrical Engineering, ICITACEE 2018*. IEEE. 10.1109/ICITACEE.2018.8576967

Eridani, D., Widianto, E. D., Windasari, I. P., Bawono, W. B., & Gunarto, N. F. (2021). *Internet of things based attendance system design and development in a smart classroom*. Indonesian Journal of Electrical Engineering and Computer Science. doi:10.11591/ijeecs.v23.i3.pp1432-1439

Luqman, A. N., Lestari, N. S., & Setiawan, I. (2019). Simplified automatic VAR/Power factor compensator using fuzzy logic based on internet of things. *Journal of Physics: Conference Series*, *1195*, 012016. doi:10.1088/1742-6596/1195/1/012016

Maulana, M. F., Adhy, S., Bahtiar, N., & Waspada, I. (2020). Development of a smart parking system based on internet of things using object-oriented analysis and design method. *Journal of Physics: Conference Series*, *1524*(1), 012111. doi:10.1088/1742-6596/1524/1/012111

Munadi, M., Nasir, M. S., Ariyanto, M., Iskandar, N., & Setiawan, J. D. (2018). Design and simulation of PID controller for lower limb exoskeleton robot. *AIP Conference Proceedings*, 060008. doi:10.1063/1.5046300

Panjaitan, J. H. D., Kurniahadi, A., Triwiyatno, A., & Setiawan, I. (2020). Design of Gain-Scheduled Fuzzy PID Controller for AFR Control System of SI-Based Motorcycle Engine Model. *7th International Conference on Information Technology, Computer, and Electrical Engineering, ICITACEE 2020 - Proceedings*. IEEE. 10.1109/ICITACEE50144.2020.9239250

Rahmat, B., Moeljani, I. R., Widjajani, B. W., Sudiyarto, S., & Harianto, H. (2021). I-OT.Net As Internet of Things (IoT) Cloud in Internet-Based Control System Applications. *5th International Seminar of Research Month 2020*.

Ravidas, S., Lekidis, A., Paci, F., & Zannone, N. (2019). Access control in Internet-of-Things: A survey. *Journal of Network and Computer Applications*, *144*, 79–101. doi:10.1016/j.jnca.2019.06.017

Riyadi, M. A., Rahmando, L., & Triwiyatno, A. (2017). Development of hovercraft prototype with stability control system using PID controller. *Proceedings - 2016 3rd International Conference on Information Technology, Computer, and Electrical Engineering, ICITACEE 2016*. IEEE. 10.1109/ICITACEE.2016.7892421

Sofwan, A., Sumardi, S., Ahmada, A. I., Ibrahim, I., Budiraharjo, K., & Karno, K. (2020). Smart Greetthings: Smart Greenhouse Based on Internet of Things for Environmental Engineering. Proceeding - *ICoSTA 2020: 2020 International Conference on Smart Technology and Applications: Empowering Industrial IoT by Implementing Green Technology for Sustainable Development.* 10.1109/ICoSTA48221.2020.1570614124

Syafei, W. A., Afiq, R., Wahyudi, W., & Hidayatno, A. (2020). Development of Controller for Internet of Things Based Anti Pollution Smart Toll Gate System. 7th International Conference on Information Technology, Computer, and Electrical Engineering, ICITACEE 2020 - Proceedings. 10.1109/ICITACEE50144.2020.9239163

Tadeus, D. Y., Yuniarto, Y., & Mangkusasmito, F. (2020). LoRa Gateway as Internet of Things (IoT) Infrastructure Components on Undip Vocational School. *IOP Conference Series: Materials Science and Engineering*. IEEE. 10.1088/1757-899X/771/1/012009

Windarto, Y. E., & Eridani, D. (2017). Door and light control prototype using Intel Galileo based Internet of Things: (Case study: Embedded and robotics laboratory, department of computer engineering, Diponegoro University). *Proceedings - 2017 4th International Conference on Information Technology, Computer, and Electrical Engineering, ICITACEE 2017*. IEEE.

KEY TERMS AND DEFINITIONS

iTCLab: This stands for internet-Based Temperature Control Lab. ITCLab is a temperature control kit for feedback control applications with an ESP32 Microcontroller, LED, two heaters and two temperature sensors.

IoT: This stands for Internet of Things. IoT is a developing region in which billions of keen objects are associated with one another utilizing the web to share information and assets.

PID: This stands for Proportional, Integral, and Derivative. PID Controller is the most famous control system in the industry. PID combines three proportional, integral and derivative control actions.

PID tuning parameter: This is how to choose the best parameter values: proportional gain, integral time constant, and derivative time constant in order to achieve optimal tuning, and produce optimal controller performance.

MQTT broker: This is an intermediary entity that enables MQTT clients to communicate. Specifically, an MQTT broker receives messages published by clients, filters the messages by topic, and distributes them to subscribers.

IoT MQTT Panel: This is a mobile application used to manage and visualize IoT projects, based on the MQTT protocol.