Design of Fuzzy PID Temperature Control System

Guokun Xie, Kai Zheng and Yajuan Jia

Xi'an Traffic Engineering Institute

Abstract. Temperature plays an important role in daily life, industry production and scientific experimentation. However, because of the multi characteristics of nonlinearity, large lag, and time-varying of control objects, there will be great difficulties in establishing accurate mathematical models with mathematical methods. In order to achieve well controlling effects, this paper proposes a mature controlling system. This system combines PID algorithm and fuzzy control to design a fuzzy PID intelligent controller, which could process complex data and complex systems simultaneously. The high flexibility of fuzzy control and the high precision of PID control of this system could solve the influence of large inertia and interference of the temperature control system. This design will adopt the STC89C52 MCU as the main control core. Meanwhile, the fuzzy PID controller are combined with fuzzy control and PID control to control the temperature. In this paper, we first introduce the hardware composition of PID temperature control system. Secondly, we use proteus to build the general circuit diagram, and then use KEIL software to program again. Finally, we simulate the system and analyze the simulation results by the simulink module in MATLAB, and finally achieve the expected results.

1 Hardware design of PID temperature control system

1.1 The structure of the system hardware circuit

The PID temperature control system are composed of seven modules: temperature acquisition module, temperature setting module, MCU control module, external storage module, alarm module, motor drive module and temperature display module [1]. The overall block diagram of the system is shown in Fig.1.
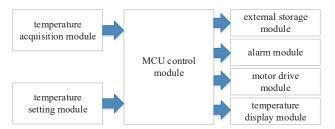

Figure 1. The overall block diagram of the proposed system.

The role of the seven modules is shown in Table 1.

Table 1. System hardware module.

(1) MCU module: The place where the system is used to operate other modules. It is the most important part of the PID temperature control system.
(2) Temperature detection: It transmits the temperature of the detected object to the microcontroller.
(3) Temperature setting module: You can preset a temperature you want to reach.
(4) Temperature limit alarm module: If the temperature display is higher or lower than the set temperature, it will start alarming for better human-machine communication.
(5) Motor drive: It consists of two parts: A heating device and a heat sink.
(6) External memory: Stores the highest and lowest values of the set temperature.
(7) Temperature display: Displays the currently set temperature value.

1.2 The general circuit diagram of system

Using the proteus to build the general circuit diagram, the circuit is composed of MCU, crystal oscillator circuit, button circuit, display circuit, reset circuit, temperature sensor and cooling circuit components [2]. Its circuit is shown in Fig.2.
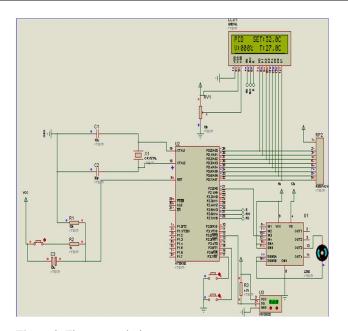
Figure 2. The system design.

The MCU is the control core of the circuit and the crystal oscillator circuit can provide a working frequency to the MCU to make the MCU work normally. When the MCU runs fault or crash, the resetting circuit realizes to enter the initial state. The display circuit is mainly used for display information, where it is mainly used to display the preset temperature and acquisition temperature of the system, for people to view the current temperature information. In this design, you can also check whether the design is running correctly. The button circuit mainly uses the hand to press the preset value, which is the preset value on the display circuit. The temperature sensor is related to temperature acquisition. It is mainly used for the collection of temperature indication, and then the collected data is transmitted to the MCU for analysis and processing; when the current temperature is higher than the preset temperature, the MCU analyzes the result and gives a command, and then transmits to the cooling circuit, thereby the circuit executes this command to reach the temperature it wants.

2 Physical display

The system takes the STC89C52 MCU as the core part, and the temperature sensor selects the full-featured DS18B20 sensor. The specific operation is to control the on/off of the relay through the MCU to realize the heating piece work to reach the preset temperature, and the LCD1620 displays the measured temperature [3]. The physical map is shown in Fig.3.

Figure 3. Physical operation diagram.

3 Software design

3.1 Main program module

The software flow chart mainly describes the physical operation and the sequence of actions. The flow chart is shown in Fig.4.
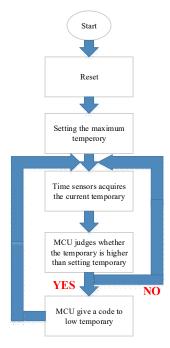

Figure 4. Program flow chart of the main program module.

The reset at the beginning sets each circuit in the initial state, and then sets the maximum value of the required temperature. At this time, the temperature sensor collects the data, and the collected data is transmitted to the MCU, and is compared with the preset temperature. When the collected data is higher than the preset temperature, the MCU sends a command, the cooling system is turned on, and the temperature sensor continuously collects the temperature data until the collected data is less than the preset temperature value, and the cooling system stops working. However, the temperature sensor should always detect the temperature and pass the data to the MCU for comparison. The MCU controls whether to start the cooling system.

3.2 PID control program

```
    struct _pid
    {
     float SetSpeed; //define the set value
     float ActualSpeed; //Define the actual value
     float err; // define the deviation value
     float err_last; //define the previous deviation value
     float Kp,Ki,Kd; //Define the proportional, integral,
and differential coefficients
     float voltage; //Define the voltage value (control the
actuator variable)
     float integral; //Define the integral value
    }      /* PID structure definition*/
    void PID_init()
    {
     pid.SetSpeed=0.0;
     pid.ActualSpeed=0.0;
     pid.err=0.0;
     pid.err_last=0.0;
     pid.voltage=0.0;
     pid.integral=0.0;
     pid.Kp=0.2;       //0.2
     pid.Ki=0.015;
     pid.Kd=0.2;
    }      /*PID control variable initialization*
    float PID_realize(float speed)
    {
     pid.SetSpeed=speed;
     pid.err=pid.SetSpeed-pid.ActualSpeed;
     pid.integral+=pid.err;

     pid.voltage=pid.Kp*pid.err+pid.Ki*pid.integral+pid.
Kd*(pid.err-pid.err_last);
     pid.err_last=pid.err;
     pid.ActualSpeed=pid.voltage;
     return   pid.ActualSpeed;}       /*Output speed
according to error*/
```

4 System simulation

4.1 Simulation diagram

The Simulink is used to build the simulation circuit. There are three adjustments of P, I and D in the circuit. As shown in Fig.5, The three parameters are changed to change the graphics.
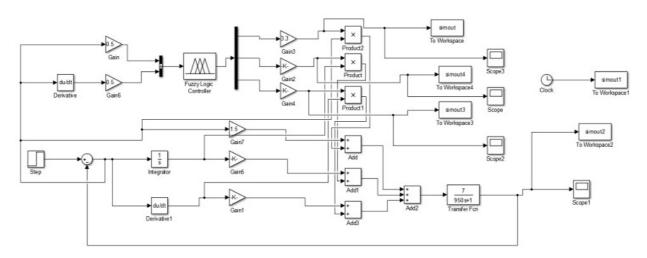

Figure 5. System diagram built in Simulink.

4.2 Simulation Results and Analysis

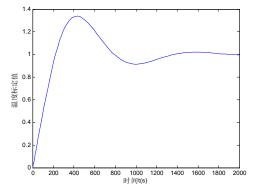We compare the conventional PID and fuzzy PID simulations as shown in Fig. 6 and 7 [4].

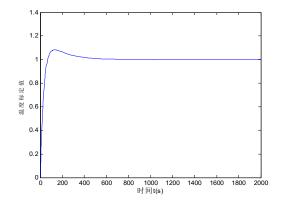**Figure 6.** Conventional PID simulations.

Figure 7. Fuzzy PID simulations.

Table 2 is compared with Fig. 6 and Fig. 7.

Table 2. Comparison of performance between conventional PID control and fuzzy PID control

Control mode	Overshoot σ%	Adjustment time t_s
Conventional PID control	30%	1400
fuzzy PID control	10%	400

It can be seen from Table 2 that this method has strong adaptability to the parameter time-varying system. Moreover, since the mathematical model of the general temperature control object can be described by the first-order inertia link, this control method is highly applicable to the time-varying system under the temperature control occasion. The measured results show that the algorithm of the fuzzy PID temperature control system is fast and stable.

5 Conclusion

The PID temperature control system and fuzzy PID temperature control system are introduced in the paper, and the results are simulated with MATLAB. The results show that the fuzzy PID control and PID control have simple operation, fast response and better stability. The overall performance of the system has also been improved.

References

1. Yujie Tang. Design of Digital Temperature Acquisition and Alarm System Based on DS18B20 [J]. Design and Manufacturing, 2014, (03).
2. Jinliang Sun. Principle and matching method of crystal oscillator circuit [J]. Home Appliance Technology, 2015, (05).
3. Yingying Guo. LCD1602 LCD screen [J]. Neijiang Technology, 2016, (07).
4. Wei Lin. Closed-loop Identification and PID Parameter Tuning of Cascade Control System [J]. Control Engineering, 2016, (01).