PID example analysis

The tutorial takes the electric heater control system as an example: its goal is to keep the temperature of a thermostat (such as a smart water dispenser) constant at 50°C.

> **PID formula**

$$u(t) = K_p e(t) + K_i \int_t^0 e(t)\mathrm{d}t + K_d \frac{de(t)}{2}$$

Example Analysis 1

Assuming that the current system temperature is 20°C, we need to use the PID controller to adjust the power of the heater so that the temperature reaches and maintains 50°C.

Proportional (P) controller

The main function is to quickly reduce the error.

Adjusting the proportional coefficient Kp can make the heater power respond to the error quickly, thereby quickly raising the temperature to close to 50°C.

If Kp is set larger, the temperature will rise quickly, but it will fluctuate greatly around 50°C.

If Kp is set small, the temperature will rise slowly, but will fluctuate less around 50°C.

```
If only Kp is adjusted, the system may oscillate around the set point
```

Integral (I) controller

The main function is to eliminate steady-state error.

Adjusting the integral coefficient Ki can make the system gradually accumulate errors over a long period of time, eventually eliminating steady-state errors and stabilizing the temperature around 50°C.

```
If only Ki is adjusted, there may be no vibration, but the set point may not be
reached quickly due to the excessive integral effect, resulting in a certain
deviation between the temperature and the set point
```

Derivative (D) controller

The main function is to predict the trend of error changes.

Adjusting the differential coefficient Kd can make the system predict the trend of error changes, thereby adjusting the heater power in advance, reducing the accumulation of errors, making the temperature approach 50°C faster and reducing overshoot.

Summary

In general, the reasonable selection of the values of the three parameters Kp, Ki, and Kd can make the system strike a balance between fast response, elimination of steady-state errors, and suppression of oscillations, thus achieving system stability and performance optimization.