Incremental PID

Incremental PID control does not directly use the error value as input to calculate the control output, but uses the increment of the control amount (i.e. the difference between the previous moment and the current moment) as input.

Incremental discrete PID formula

$$\triangle U_k = K_p[e(k) - e(k-1)] + K_i * e(k) + K_d[e(k) - 2e(k-1) + e(k-2)]$$

- $\triangle$Uk: incremental output

- e(k): current deviation

- e(k-1): last deviation

- e(k-2): last deviation

- Kp: proportional term parameter

- Ki: integral term parameter

- Kd: differential term parameter

Difference between positional and incremental PID

> **positional PID**

The positional PID calculates the current deviation value (difference between target value and actual value), integral value (sum of past errors) and differential value (error change rate), and then adds these three parts to obtain the PID output.

> **incremental PID**

The incremental PID calculates the change in the difference between the current error and the last error to obtain the PID output increment.

```
Position PID needs to accumulate the integral term, which is suitable for the
occasions where the system has high requirements for steady-state error.
Incremental PID does not need to accumulate the integral term, which is suitable
for the occasions where the system has high requirements for response speed.
```

Code implementation

Incremental PID is speed closed-loop control. Speed closed-loop control is to measure the speed information of the motor according to the number of pulses obtained per unit time, and compare it with the target value to obtain the deviation. Then we control the deviation proportionally, integrally, and differentially to make the deviation approach 0.

```
int Incremental_PI (int Encoder, int Target)
{
static float error, Pwm, Last_error, Last_last_error;
error = Target-Encoder; // Calculate deviation
Pwm + = Incremental_KP * (error-Last_error) +
Incremental_KI * error +
Incremental_KD * (error-2 * Last_error + Last_last_error); // Incremental PID
controller

Last_error = error; // Save the last deviation
Last_last_error = Last_error; // Save the last deviation

return Pwm; // Incremental output
}
```

Software code

Since the relevant peripheral driver tutorial has been introduced before, it will not be introduced here!

```
Product supporting materials source code path: Attachment → Source code summary →
3.PID_Course → 05.Increment_PID
```

Experimental phenomenon

The Increment_PID.hex file generated by the project compilation is located in the OBJ folder of the Increment_PID project. Find the Increment_PID.hex file corresponding to the project and use the FlyMcu software to download the program into it.

After the program is successfully downloaded: the serial port will print the button status and the currently set encoder increment (corresponding to the motor rotation speed). The button can switch the LED status and control the speed of the motor.

```
When using the serial port debugging assistant, you need to pay attention to the
serial port settings. If the settings are wrong, the phenomenon may be
inconsistent.
```