

# PID algorithm basics

## 1. Introduction

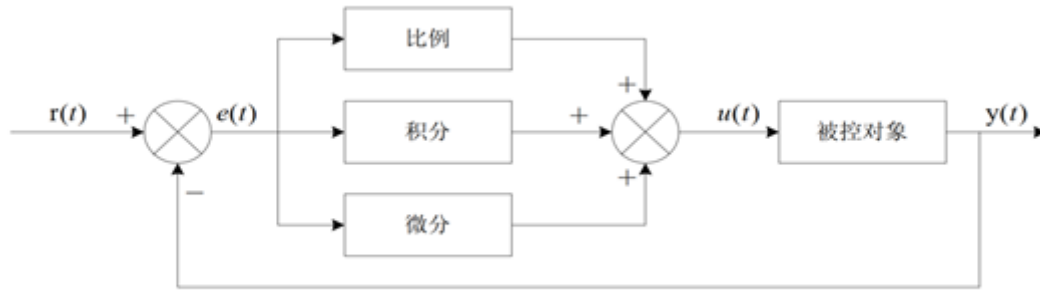
PID is to perform proportional, integral and differential operations on the input deviation, and the superposition results of the operations are used to control the actuator. The formula is as follows:

$$u(t) = K_p[e(t) + \frac{1}{T_i} \int_0^t e(t)dt + Td\frac{de(t)}{dt}]$$

It consists of three parts:

- P is the proportion, which is the input deviation multiplied by a coefficient;
- I is the integral, which is the integral operation of the input deviation;
- D is the differential, which is the differential operation of the input deviation.

The following figure shows a basic PID controller:



### (1) Proportional part

The mathematical expression of the proportional part is:  $K_p \cdot e(t)$

In the analog PID controller, the role of the proportional link is to respond to the deviation instantly. Once the deviation occurs, the controller immediately produces a control action to change the control amount in the direction of reducing the deviation. The strength of the control effect depends on the proportional coefficient. The larger the proportional coefficient, the stronger the control effect, the faster the transition process, and the smaller the static deviation of the control process; however, the larger the proportional coefficient, the easier it is to produce oscillation, which will destroy the stability of the system. Therefore, the proportional coefficient must be selected appropriately to achieve a short transition time, small static error and stable effect.

Advantages: Adjust the open-loop proportional coefficient of the system to improve the steady-state accuracy of the system, reduce the inertia of the system, and speed up the response speed.

Disadvantages: If only the P controller is used, an excessively large open-loop proportional coefficient will not only increase the overshoot of the system, but also reduce the stability margin of the system, or even make it unstable.

### (2) Integral part

The mathematical expression of the integral part is:

$$\frac{K_p}{T_i} \int_0^t e(t)dt$$

The larger the integral constant, the weaker the accumulation effect of the integral. At this time, the system will not produce oscillation during the transition; however, increasing the integral constant will slow down the elimination process of the static error, and the time required to eliminate the deviation will also be longer, but it can reduce the overshoot and improve the stability of the system. When  $T_i$  is small, the integral effect is stronger. At this time, oscillation may occur in the system transition time, but the time required to eliminate the deviation is shorter. Therefore,  $T_i$  must be determined according to the specific requirements of the actual control.

Advantages: Eliminate steady-state error.

Disadvantages: The addition of the integral controller will affect the stability of the system and reduce the stability margin of the system.

### (3) Differential part

The mathematical expression of the differential part is:

$$K_p * T_d \frac{de(t)}{dt}$$

The role of the differential link is to prevent the change of the deviation. It is controlled according to the change trend (speed of change) of the deviation. The faster the deviation changes, the greater the output of the differential controller, and it can be corrected before the deviation value becomes larger. The introduction of the differential action will help reduce overshoot, overcome oscillation, and make the system stable. It is especially beneficial for high-order systems. It speeds up the tracking speed of the system. However, the differential effect is very sensitive to the noise of the input signal. For those systems with large noise, differentiation is generally not used, or the input signal is filtered before the differential takes effect. The effect of the differential part is determined by the differential time constant  $T_d$ . The larger the  $T_d$ , the stronger its effect of suppressing the deviation change; the smaller the  $T_d$ , the weaker its effect of resisting the deviation change. The differential part obviously has a great effect on the stability of the system. Proper selection of the differential constant  $T_d$  can make the differential effect optimal.

Advantages: Make the system's response speed faster, reduce overshoot, reduce oscillation, and have a "predictive" effect on dynamic processes.

## 2. Selection of PID algorithm

Digital PID control algorithms can be divided into position PID and incremental PID control algorithms. So before we decide which PID algorithm to use, we should first understand its principle:

- Position PID algorithm:

$$u(k) = K_p e(k) + K_I \sum_{i=0} e(i) + K_D [e(k) - e(k-1)]$$

$e(k)$ : User-set value (target value) - Current state value of the controlled object

Proportional P:  $e(k)$

Integral I:  $\sum e(i)$  Accumulated error

Differential D:  $e(k) - e(k-1)$  This error - Last error

That is, the position PID is the actual position of the current system, and the deviation from the expected position you want to achieve, for PID control

Because there is an error integral  $\sum e(i)$ , it is accumulated all the time, that is, the current output  $u(k)$  is related to all the past states, and the accumulated value of the error is used; (error  $e$  will have an accumulated error), the output  $u(k)$  corresponds to the actual position of the actuator, once the control output is wrong (there is a problem with the current state value of the controlled object), a large change in  $u(k)$  will cause a large change in the system. When the integral term of the position PID reaches saturation, the error will continue to accumulate under the action of the integral. Once the error begins to change in the opposite direction, the system needs a certain amount of time to exit the saturation zone. Therefore, when  $u(k)$  reaches the maximum and minimum, the integral action must be stopped, and there must be integral limit and output limit. Therefore, when using the position PID, we generally use PD control directly. The position PID is suitable for objects whose actuators do not have integral components, such as the upright and temperature control systems of servos and balancing cars.

Advantages: Position PID is a non-recursive algorithm that can directly control actuators (such as balancing cars). The value of  $u(k)$  corresponds to the actual position of the actuator (such as the current angle of the car), so it can be well applied in objects where the actuators do not have integral components.

Disadvantages: Each output is related to the past state. When calculating,  $e(k)$  must be accumulated, and the calculation workload is large.

- Incremental PID:

$$\Delta u(k) = u(k) - u(k-1) = K_P[e(k) - e(k-1)] + K_I e(k) + K_D[e(k) - 2e(k-1) + e(k-2)]$$

Proportional P:  $e(k) - e(k-1)$  this error-last error

Integral I:  $e(k)$  error

Differential D:  $e(k) - 2e(k-1) + e(k-2)$  this error-2\*last error+last error

Incremental PID can be well seen from the formula that once  $K_P$ ,  $T_I$ ,  $T_D$  are determined, the control increment can be obtained by the formula by using the deviation of the three previous and subsequent measurement values. The control amount  $\Delta u(k)$  corresponds to the increment of the position error in recent times, not the deviation from the actual position. There is no error accumulation. In other words, no accumulation is required in incremental PID. The determination of the control increment  $\Delta u(k)$  is only related to the last three sampling values. It is easy to obtain a better control effect through weighted processing, and when a problem occurs in the system, the incremental type will not seriously affect the work of the system.

Summary: Incremental PID is an increment of the position PID. At this time, the controller outputs the difference between the position values calculated at two adjacent sampling moments. The result is an increment, that is, the control amount needs to be increased (negative value means reduction) on the basis of the previous control amount.

Advantages: ① The impact of malfunction is small, and the erroneous data can be removed by logical judgment when necessary.

② The impact is small when switching between manual and automatic, which is convenient for disturbance-free switching. When the computer fails, the original value can still be maintained.

③ No accumulation is required in the formula. The determination of the control increment  $\Delta u(k)$  is only related to the last three sampling values.

Disadvantages: ① The integral truncation effect is large, and there is a steady-state error;

② The overflow has a large impact. Some controlled objects are not suitable for incremental control;

- Differences between incremental control and position control

(1) Incremental control does not require accumulation. The determination of the control quantity increment is only related to the most recent deviation sampling values. The calculation error has little impact on the control quantity calculation. The position control algorithm uses the accumulated value of past deviations, which is prone to produce large accumulated errors.

(2) The incremental control algorithm obtains the increment of the control quantity. For example, in valve control, only the change in valve opening is output. The impact of false operation is small. If necessary, the output can be limited or prohibited through logical judgment, which will not seriously affect the operation of the system. The output of the position control directly corresponds to the output of the object, so it has a greater impact on the system.

(3) The incremental PID control outputs the control quantity increment and has no integral effect. Therefore, this method is suitable for objects with actuators with integral components, such as stepper motors, while the position PID is suitable for objects with actuators without integral components, such as electro-hydraulic servo valves.

(4) When performing PID control, the position PID needs integral limiting and output limiting, while the incremental PID only needs output limiting. Position PID and incremental PID are just two implementation forms of digital PID control algorithms, and they are essentially the same. The main difference is the storage method of the integral item. The position PID integral item is stored separately, while the incremental PID integral item is stored as part of the output. Various players on the Internet also have their own unique insights and opinions on the use of position and incremental methods. It depends on which algorithm is suitable for our specific application scenario.

### 3. PID parameter debugging

There are many methods for selecting PID controller parameters, such as trial and error, critical proportionality method, and expanded critical proportionality method. However, for PID control, parameter selection is always a very complicated task, and it requires continuous adjustment to obtain a more satisfactory control effect. Based on experience, the steps for determining the general PID parameters are as follows:

- Determine the proportional coefficient  $K_p$

When determining the proportional coefficient  $K_p$ , first remove the integral and differential terms of the PID, and set  $T_i=0$  and  $T_d=0$  to make it

Pure proportional regulation. The input is set to 60% to 70% of the maximum output allowed by the system, and the proportional coefficient  $K_p$  gradually increases from 0 until the system oscillates; conversely, the proportional coefficient  $K_p$  at this time gradually decreases until the system oscillation disappears. Record the proportional coefficient  $K_p$  at this time and set the proportional coefficient  $K_p$  of PID to 60% to 70% of the current value.

- Determine the integral time constant  $T_i$

After the proportional coefficient  $K_p$  is determined, set a larger integral time constant  $T_i$ , and then gradually reduce  $T_i$  until the system oscillates, and then conversely, gradually increase  $T_i$  until the system oscillation disappears. Record  $T_i$  at this time and set the integral time constant  $T_i$  of PID to 150% to 180% of the current value.

- Determine the differential time constant  $T_d$

The differential time constant  $T_d$  generally does not need to be set, it can be 0, at which point the PID adjustment is converted to PI adjustment. If it needs to be set, the method is the same as determining  $K_p$ , taking 30% of its value when there is no oscillation.

- System no-load and load joint adjustment

Fine-tune the PID parameters until the performance requirements are met.

Of course, this is just my personal debugging method, which is not necessarily suitable for everyone and every environment, and is only provided for your reference; however, there are also classic trial and error formulas for debugging PID circulating on the Internet, and I also posted them for your reference:

Find the best parameter setting, and check from small to large.

First the proportion, then the integral, and finally add the differential.

The curve oscillates frequently, and the proportional dial should be enlarged.

The curve floats around a big bend, and the proportional dial is turned down.

The curve deviation is slow to recover, and the integral time is reduced.

The curve fluctuation period is long, and the integral time is further extended.

The oscillation frequency of the curve is fast, so first reduce the differential.

The dynamic difference is large and the fluctuation is slow, so the differential time should be extended.

The ideal curve has two waves, the first high and the second low, four to one.

One look, two adjustments, and more analysis, the adjustment quality will not be low.

PID is a proportional (P), integral (I), and differential (D) control algorithm. It is not necessary to have all three algorithms at the same time. It can also be PD, PI, or even only P algorithm control. My most simple idea for closed-loop control before was only P control. Feedback the current result and then subtract it from the target. If it is positive, it will slow down, and if it is negative, it will speed up. Of course, this is only the simplest closed-loop control algorithm, which is still back to the summary of the position type and incremental type in the previous section. The specific reference should be made to our current control environment. Because of the differences in each control system, the parameters that can make our system achieve the most stable effect are of course OK.