Car speed control (PI)

The tutorial mainly introduces the speed control of the balancing car.

```
If the car has an inclination, the upright control will accelerate the car in the inclination direction. We can use this feature of the car to control the speed.
```

Speed measurement principle

The speed of the car can be configured with the development board encoder interface for motor encoder reading. By reading the motor encoder value, the number of pulse signals per unit time is obtained, which is reflected as the motor speed.

```
The motor rotation speed needs to be calculated in combination with the motor parameters
```

PI controller

The main function of differential (D) control in PID control is to reduce the dynamic error and oscillation of the system, and the main function of integral (I) control is to eliminate static error, so we can establish a speed PI closed-loop control, output an angle to make the car reach the target speed, realize cascade PID control, let the output of speed control as the input of angle control, and the output of angle control directly acts on the motor.

Implementation code

```
int Velocity_PI(int encoder_left,int encoder_right)
{
static float velocity,Encoder_Least,Encoder_bias,Movement;
static float Encoder_Integral;

Movement=Move_X;
//=================Speed PI controller=====================//
Encoder_Least =0-(encoder_left+encoder_right); //Get the latest speed deviation = target speed (zero here)-measured speed (sum of left and right encoders)
Encoder_bias *= 0.84; //First-order low-pass filter
Encoder_bias += Encoder_Least*0.16; //First-order low-pass filter to slow down speed changes
Encoder_Integral +=Encoder_bias; //Integrate displacement Integration time: 5ms
Encoder_Integral=Encoder_Integral+Movement; //Receive remote control data, control forward and backward
if(Encoder_Integral>8000) Encoder_Integral=8000; //Integral limit
if(Encoder_Integral<-8000) Encoder_Integral=-8000; //Integral limit
velocity=-Encoder_bias*Velocity_Kp/100-Encoder_Integral*Velocity_Ki/100; //Speed control

if(Turn_Off(Angle_Balance,battery)==1) Encoder_Integral=0;//Clear integration after motor is turned off
return velocity;
}
```

Software code

Basic balance car PID control basics: 08-13 tutorial only provides one project file.

```
Product supporting materials source code path: Attachment → Source code summary → 3.PID_Course → 08-13.Balanced_Car_PID
```