Balance car parameter adjustment

The tutorial introduces how to adjust the PID parameters of the balance car to keep the car in a more stable balance state.

```
The tutorial only introduces the standard library project code
```

PID parameter adjustment

```
Users can first burn the program we provide to observe the effect and then
consider whether to adjust the parameters.
```

> **Description**

The code used for the balance car parameter adjustment is the Bluetooth remote control project source code. We can adjust the PID parameters through the APP.

```
Product supporting information source code path: Attachment → Source code summary
→ 4.Balanced_Car_base → 04.bluetooth_control
```

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

After the program is downloaded successfully: After the OLED displays the button prompt, press the KEY1 button to start the car balance function: the OLED will display the inclination of the balance car in real time!

```
The program has voltage detection. If the voltage is less than 9.6V, the low
voltage alarm is triggered and the buzzer will sound.
Common situations that trigger the voltage alarm:
1. The power switch of the development board is not turned on, and only the Type-
C data cable is connected for power supply
2. The battery pack voltage is lower than 9.6V and needs to be charged in time
```

Determine the mechanical median

Mechanical median: Ensure that the overall center of gravity of the car is in a stable position so that the car can maintain balance during movement.

Actual operation:

1. You need to clear all PID parameters first (that is, modify each Kp, Kd, Ki parameter to 0), then burn the program, press the KEY1 button according to the prompt to start the car balance function.

```
PID parameter source code location: pid_control.c
Modification content:
float Balance_Kp =0; // Range 0-288
float Balance_Kd =0; // Range 0-2
//Speed loop PI control parameter
float Velocity_Kp=0; // Range 0-72
float Velocity_Ki=0; // kp/200
//Steering loop PD control parameter
float Turn_Kp=0; // This is adjusted according to your needs, but the balance can be adjusted, which is related to the rotation speed
float Turn_Kd=0; // Range 0-2
```

2. Use your hand to hold the balance car and observe the MPU6050 angle value displayed on the OLED when the car is close to balance (that is, the hand is separated from the car, and the car can maintain a short balance state).
3. Change the approximate value of the MPU6050 angle displayed on the OLED to the mechanical median parameter.

```
Source code location of mechanical median: main.c
Modified content:
Mid_Angle = 1; // Here, it is assumed that the mechanical median is 1. When the car is balanced, the angle value of MPU6050 displayed on OLED is around 1
```

Determine PID polarity

After completing the mechanical median step, determine the polarity (polarity adjustment is mainly determined by the positive and negative of the parameter): only modify one PID parameter at a time, and the other parameters are 0.

```
For the convenience of adjustment, the PID parameters of the program are enlarged 100 times
```

Vertical ring polarity

The vertical ring uses Kp and Kd parameters.

> **Determine the polarity of the upright ring Kp**

Modify Balance_Kp=10000 (actual value is 100), Balance_Kd =0:

```
PID parameter source code location: pid_control.c
Modification content:
float Balance_Kp =10000; // Range 0-288
float Balance_Kd =0; // Range 0-2
```

After downloading and running the program:

If the car is tilted, it will accelerate in the tilted direction (maintaining the balance trend), indicating that the polarity of Kp is correct;

If the car is tilted, it will accelerate in the opposite direction of the tilt (accelerating the trend of falling), indicating that the polarity of Kp is opposite.

```
If the car polarity is negative, in order to modify and unify the parameters, it
is recommended that the parameters transmitted by Kp, Kd, and Ki are positive.
You can directly modify the positive and negative of the upright ring calculation
formula to ensure that the parameters we debug later are positive.
int Balance_PD(float Angle,float Gyro)
{
float Angle_bias,Gyro_bias;
int balance;
Angle_bias=Mid_Angle-Angle;
Gyro_bias=0-Gyro;
balance=-Balance_Kp/100*Angle_bias-Gyro_bias*Balance_Kd/100; // Modify the sign
in front of Balance_Kp
return balance;
}
```

> **Determine the polarity of the upright ring Kd**

Modify Balance_Kp=0, Balance_Kd =100 (actual value is 1):

```
PID parameter source code location: pid_control.c
Modification content:
float Balance_Kp =0; // Range 0-288
float Balance_Kd =100; // Range 0-2
```

After downloading the program and running it:

Place the car vertically, close to the mechanical midpoint: if the car is tilted, it will move in the tilted direction (maintaining the balance trend), indicating that the polarity of Kd is correct;

Place the car vertically, close to the mechanical midpoint: if the car is tilted, it will move in the opposite direction of the tilt (accelerating the trend of falling), indicating that the polarity of Kd is opposite.

```
If the car polarity is negative, in order to modify and unify the parameters, it
is recommended that the parameters transmitted by Kp, Kd, and Ki are positive.
You can directly modify the positive and negative of the upright ring calculation
formula to ensure that the parameters we debug later are positive.
int Balance_PD(float Angle,float Gyro)
{
float Angle_bias,Gyro_bias;
int balance;
Angle_bias=Mid_Angle-Angle;
Gyro_bias=0-Gyro;
balance=-Balance_Kp/100*Angle_bias-Gyro_bias*Balance_Kd/100; // Modify the sign
in front of Balance_Kd
return balance;
}
```

Speed loop polarity

The speed loop uses Kp and Ki parameters.

According to engineering experience, Kp and Ki parameters have a certain proportional relationship: Kp=Ki*200

> **Determine the speed loop Kp and Ki**

Modify Velocity_Kp=6000 (actual value is 60), Velocity_Ki=30 (actual value is 0.3):

```
PID parameter source code location: pid_control.c
Modified content:
float Velocity_Kp=6000; // Range 0-72
float Velocity_Ki=30; // Kp/200
```

After downloading the program and running it:

Turn the wheel of the balancing car. If the wheel speed is getting faster and faster, it means that the polarity of Kp and Ki is correct;

Turn the wheel of the balancing car. If the wheel speed is basically unchanged or there is resistance, it means that the polarity of Kp and Ki is opposite.

```
If the polarity of the car is negative, in order to modify and unify the
parameters, it is recommended that the parameters transmitted by Kp, Kd, and Ki
are positive. You can directly modify the positive and negative of the speed loop
calculation formula to ensure that the parameters we debug later are positive.
int Velocity_PI(int encoder_left,int encoder_right) { static float
velocity,Encoder_Least,Encoder_bias,Movement; static float Encoder_Integral;
if(g_newcarstate==enRUN)Movement=Car_Target_Velocity; else
if(g_newcarstate==enBACK)Movement=-Car_Target_Velocity; else Movement=Move_X;
Encoder_Least = 0-(encoder_left+encoder_right); Encoder_bias *= 0.84;
Encoder_bias += Encoder_Least*0.16; Encoder_Integral +=Encoder_bias;
Encoder_Integral=Encoder_Integral+Movement; if(Encoder_Integral>8000)
Encoder_Integral=8000;
if(Encoder_Integral<-8000) Encoder_Integral=-8000;
velocity=-Encoder_bias*Velocity_Kp/100-Encoder_Integral*Velocity_Ki/100; //
Modify the sign before Velocity_Kp and Velocity_Ki
if(Turn_Off(Angle_Balance,battery)==1) Encoder_Integral=0;
return velocity;
}
```

Steering ring polarity

The steering ring uses the Kp and Kd parameters (mainly Kd parameter debugging).

The Kp parameter is part of the Bluetooth remote control car. The polarity of this parameter is related to the direction of remote control steering. If the direction is opposite during remote control, the Kp polarity can be reversed.

If you want to modify the steering speed of the car, directly modify the Car_Turn_Amplitude_speed parameter.

> **Determine the speed loop Kd**

Modify Turn_Kp=0, Turn_Kd=100 (actual value is 1):

```
PID parameter source code location: pid_control.c
Modification content:
float Turn_Kp=0; // This is adjusted according to your needs, but the balance can
be adjusted without adjustment, which is related to the rotation speed
float Turn_Kd=100; // Range 0-2
```

After downloading the program and running:

Rotate the balance car, if the wheels inhibit our rotation, it means that the polarity of Kp and Ki is correct;

Rotate the balance car, if the wheels help us rotate, it means that the polarity of Kp and Ki is opposite.

```
If the car polarity is negative, in order to modify and unify the parameters, it
is recommended that the parameters transmitted by Kp, Kd, and Ki are positive.
You can directly modify the positive and negative of the speed loop calculation
formula to ensure that the parameters we debug later are positive.
int Turn_PD(float gyro) { static float Turn_Target,turn_PWM; float Kp=Turn_Kp,Kd;
//To modify the steering speed, please modify Car_Turn_Amplitude_speed
if(g_newcarstate==enLEFT)Turn_Target=-Car_Turn_Amplitude_speed; else
if(g_newcarstate==enRIGHT)Turn_Target=Car_Turn_Amplitude_speed; else if(g
_newcarstate == enTLEFT) Turn_Target=-50; else if(g_newcarstate == enTRIGHT)
Turn_Target=50; else { Turn_Target=0; } if(g_newcarstate==enRUN ||
g_newcarstate==enBACK ) { Kd=Turn_Kd; } else Kd=0;
turn_PWM=Turn_Target*Kp/100+gyro*Kd/100+Move_Z; // Modify the sign in front of Kp
and Kd
return turn_PWM;
}
```

Adjust PID parameters

After adjusting the polarity of PID parameters, you can adjust the size of PID parameters.

```
First adjust the upright ring, then the speed ring, and finally the steering
ring
```

Upright ring

First modify the Balance_Kp parameter and then modify the Balance_Kd parameter.

```
After each modification, you need to download the program to the development
board to verify the effect
```

Other PID parameters need to be set to 0 first, and then the Balance_Kp parameter needs to be increased separately until the balancing car has low-frequency jitters;

After low-frequency jitters occur, modify the Balance_Kd parameter until high-frequency jitters occur, and multiply the Balance_Kp and Balance_Kd values by 0.6 as the determined values for modifying the parameters.

Speed loop

Retain the previously confirmed PID data, and modify the Velocity_Kp and Velocity_Ki parameters synchronously, the ratio: Kp=Ki*200.

```
After each modification, you need to download the program to the development
board to verify the effect
```

Increase the Velocity_Kp and Velocity_Ki parameters to observe whether the balance car can be balanced in place. If the increase is such that the balance car shakes and the balance car will swing back sharply when pushed by hand, take the previous set of data as the determined value of the Velocity_Kp and Velocity_Ki parameters.

Turning loop

Retain the previously confirmed PID data, and modify the Turn_Kd parameter.

```
After each modification, you need to download the program to the development
board to verify the effect
```

Use Bluetooth to control the car to observe the effect of the car walking in a straight line, increase the Turn_Kd parameter, if the car shakes, take the previous Turn_Kd parameter as the determined value.

Notes

- To determine the PID polarity, other PID parameters need to be set to 0
- To adjust the PID parameters, you need to keep the previously determined data and then determine the new PID parameters. You do not need to set other parameters to 0
- The PID parameters do not have to be very accurate, as long as they are balanced and do not swing greatly

```
The program has voltage detection. If the voltage is less than 9.6V, the low
voltage alarm is triggered and the buzzer will sound.
Common situations that trigger the voltage alarm:
1. The power switch of the development board is not turned on, and only the Type-
C data cable is connected for power supply
2. The battery pack voltage is lower than 9.6V and needs to be charged in time
```