Study on self balancing car

Huang Jian[1,a,*]

[1] XiJing University, Xi'an 710123, China

[a] 565200245@qq.com

*corresponding author

Keywords: Self balancing car; PID algorithm; MPU6050

Abstract: Control from the balance of the car is relatively complex, which can be divided into balance, speed and direction of the three control, the balance control in the main, the control is similar to the inverted pendulum, to study its stress, construct its transfer function. MPU6050 is used to measure its rotation angle as a feedback quantity to form a closed loop system. PID algorithm is used to control the speed of the motor, and the corresponding PWM value is calculated. The balance, speed and direction of the motor can be controlled. The design has the advantages of fast response and high robustness.

1. Introduction

In recent years, self balancing vehicles have the advantages of convenience, shortcut and so on. They have been widely used. Among them, the effective control of them has become the focus of research in recent years, and has been gradually paid attention to. But its control quantity is more and more difficult, and how to control it effectively, different methods are put forward. In this paper, we simplify the complex problem, decompose it into three control, such as balance, speed and direction, and then carry out linear superposition, which can effectively control it.

2. Theoretical Analysis

The transfer function is shown in figure 1. The ratio and differential coefficients K1 and K2. are introduced as follows:

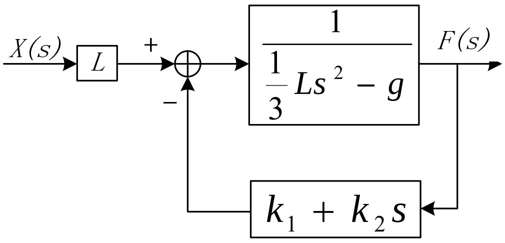

Figure 1 transfer function diagram

$$H_1(S) = \frac{1}{\frac{1}{3}Ls^2 - g + k_1 + k_2 s} \tag{1}$$

There are two poles in formula (1):

$$S_{p1} = \frac{-3k_2 \pm \sqrt{9k_2^2 - 12L(k_1 - g)}}{2L} \tag{2}$$

To eliminate the 2 poles, you must make K2>0, K1>g.

3. Software Program

In this design, under KEIL 5, programming with C language. Complete the PD control of the balance car. Here is part of the code：

```
 int TIM1_UP_IRQHandler(void)
 {
  if(TIM1->SR&0X0001)
    {
                 TIM1->SR&=~(1<<0);
                 Flag_Target=!Flag_Target;
          if(Flag_Target==1)
    {
               Get_Angle(Way_Angle);
                   return 0;
               }
           Encoder_Left=-Read_Encoder(2);
     Encoder_Right=Read_Encoder(4);
Get_Angle(Way_Angle);
               Led_Flash(100);
         Voltage=Get_battery_volt();
               Key();
         Balance_Pwm =balance(Angle_Balance,Gyro_Balance);
     Velocity_Pwm=velocity(Encoder_Left,Encoder_Right);
 Turn_Pwm    =turn(Encoder_Left,Encoder_Right,Gyro_Turn);
 Moto1=Balance_Pwm-Velocity_Pwm+Turn_Pwm;
 Moto2=Balance_Pwm-Velocity_Pwm-Turn_Pwm;
 Xianfu_Pwm();
 if(Turn_Off(Angle_Balance,Voltage)==0)
  Set_Pwm(Moto1,Moto2);                              }
 return 0;
 }
 int velocity(int encoder_left,int encoder_right)
 {
    static float Velocity,Encoder_Least,Encoder,Movement;
     static float Encoder_Integral;
     float kp=80,ki=0.40;
```

```
   if(1==Flag_Qian)       Movement=-90;
  else if(1==Flag_Hou)    Movement=90;
    else  Movement=0;
  Encoder_Least =(Encoder_Left+Encoder_Right)-0;
  Encoder *= 0.7;
  Encoder += Encoder_Least*0.3;
   Encoder_Integral +=Encoder;
   Encoder_Integral=Encoder_Integral-Movement;
 if(Encoder_Integral>10000)   Encoder_Integral=10000;
 if(Encoder_Integral<-10000)  Encoder_Integral=-10000;
  Velocity=Encoder*kp+Encoder_Integral*ki;
if(Turn_Off(Angle_Balance,Voltage)==1)   Encoder_Integral=0;
 return Velocity;
 }
 void Set_Pwm(int moto1,int moto2)
 {
              if(moto1>0)                  AIN2=1,                 AIN1=0;
              else          AIN2=0,                 AIN1=1;
              PWMA=myabs(moto1);
         if(moto2>0)      BIN1=0,                 BIN2=1;
              else       BIN1=1,                  BIN2=0;
              PWMB=myabs(moto2);
 }
```

4. Summary

This paper expounds the working principle and method of using PID to control the self balanced vehicle, and designs the hardware circuit. Software programming is carried out under KEIL, and the rotation angle of the car is collected by MPU6050, and a closed loop system is formed. The PID algorithm is used to control the balance, speed and direction of the car, and realize the self balanced control of the car. It has certain practical value.

References

[1] Qichang, Yuan Hongyue, Jin Xudong. The two wheeled self balanced vehicle speed sensorless balance control based on [J]. control engineering, 2013,20 (4): 618-622.

[2] Wen Jiabin, Ma Chen Wei. Design of fuzzy PI control system for brushless DC motor [J]. proceedings of the world's electric power and control, 2016, 20 (3): 102-108.

[3] Guo Wei, Chen Yifan, Zhou Li, Ni Li. Study on control of fractional order PI predictive function of Brushless DC motor [J]. Journal of Wuhan University of Technology, 2013,35 (3): 130-

135.

[4] Jiang Wenbiao, Wu Jian. Research on double closed loop PI control technology of DC motor [J]. machine tool and hydraulic technology, 2012,40 (11): 21-24.

[5] Wang Xia, Zhu Jingwei. Brushless DC motor speed control system based on fuzzy PI control [J]. electric drive, 2014,44 (1): 63-67.