

RESEARCH

Open Access

Design of two-wheel self-balancing vehicle based on visual identification



Boping Zhang and Guoxi Wu*

Abstract

Poor user experience is caused by the current unstable operation of the two-wheel self-balancing vehicle control system, the high failure rate, and the poor accuracy and sensitivity of the control system (prone to deviation). Therefore, the work studied the two-wheel self-balancing vehicle system with the visual recognition of vehicle attitude. Kalman filtering and PID control algorithm were adopted to reduce the high frequency interference of the accelerometer and the low frequency error of the gyroscope, improve the response speed and control precision of motor to error, and ensure vertical control. The road images were acquired by a charge-coupled device vision sensor with the edge information extracted by the Laplacian operator, which ensures autonomous navigation control of a balance car. Based on the calculation of the distance between road axis and the existence detection of forward obstacles, the early warning mechanism was established through the calculation of the group agent to improve the safety performance of a balance vehicle. Experiments showed the improved control system has good stability, fast walking speed, strong anti-interference, and high security.

Keywords: Two-wheel balance car, CCD, Visual identification, Kalman filtering, PID algorithm

1 Introduction

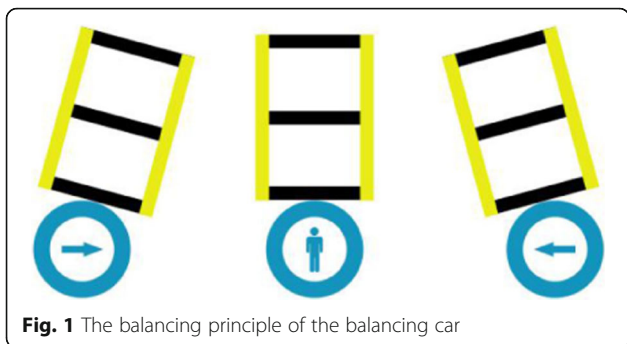
The two-wheel self-balancing vehicle has a simple structure, flexible movement, easy driving, and convenient carrying, meeting the needs of energy saving and environmental protection. It is popular with more and more green travel enthusiasts and has become an important branch of mobile robot research. The automatic balance vehicle is also called body-sensing car or Segway. Based on “dynamic stability”, the gyroscope and acceleration sensor inside the vehicle are used to detect the change of vehicle attitude, and the servo control system is used to adjust the motor for a balance system. The gyroscope detects the change of the gravity center of a human body. After the fast calculation by a single-chip micro-computer, the motor is driven to make the balance vehicle advance or retreat, with the turning realized by direction calculation. When a balance vehicle leans forward/backwards/in the left and right direction, the motor rotators rotate forward/backwards/in the left and

right direction to give the system a forward/backwards/left and right acceleration. When the balance vehicle reaches equilibrium, the motor automatically turns off (See Fig. 1).

The control system of existing balance vehicles has unstable operating conditions, high failure rate, and low accuracy and sensitivity during handling (prone to deviation), which results in poor user experience [1–3]. The work studied the two-wheel self-balancing vehicle system with the visual recognition of vehicle attitude, which solves accurate vertical control, visual autonomous navigation, and early warning mechanism. Kalman filtering and process identifier (PID) control algorithm are used to reduce the high frequency interference of the accelerometer and the low frequency error of the gyroscope, improve the response speed and control precision of the motor to error, and ensure good vertical control. Road images were obtained by the charge-coupled device (CCD) vision sensor of the balancing vehicle, performing

* Correspondence: guoxiwu11@163.com

School of Information Engineer, Xuchang University, Xuchang 461000, Henan, China



grayscale equalization enhancement, median filtering, binarization, and the extraction of Laplacian operator edge information. It ensured autonomous navigation of balancing vehicle. Existence detection was performed by the calculation of distance between road edge and axis and the frontal obstacles with gray feature. The early warning mechanism of the system was established by group agent calculation, which improved the safety of balance vehicle. It is expected that the market acceptance will be improved. Experiments showed the improved control system can stably control the balance vehicle, with fast traveling speed, strong anti-interference, and high safety.

2 Proposed method

Generally, long-time use for skilled users leads to inattention, increasing the possibility of traffic accidents. It is necessary for the judgment of vehicle attitude. When the vehicle is on the edge of the traffic warning line or there is a forward obstacle, an alarm is automatically issued to remind the user of the road condition. This requires a vision system, which uses the camera of the vehicle to capture images. Image processing algorithms are used to detect road edges and extract the road axis, calculating the deviation between the position of balance vehicle and the edge. An alarm message is issued when the deviation exceeds a certain threshold.

2.1 Grayscale equalization enhancement of road images

A large amount of color information is in the image of each road captured by a CCD vision sensor. Since the on-board system has high real-time requirements, it is necessary to convert color space into gray space by gradation transformation [4], as shown in Eq. (1).

$$\text{Gray}(i, j) = 0.3\text{Red}(i, j) + 0.59\text{Green}(i, j) + 0.11\text{Blue}(i, j) \tag{1}$$

Figure 2 shows the original image of the road and the converted grayscale image.

Road images have high gray level due to light changes. The contrast between the road marking line and pavement is seriously low, and sometimes white pavement appears at the end of the road [5]. In order to reduce the interference of light changes, the work used histogram equalization enhancement to reconstruct gray level transformation. The contrast was increased by stretching the area of grayscales in the original image, which makes the details of the image clear and highlights depth information of the road image.

The grayscale histogram is the statistical relation between the grayscales in the digital image and the frequency of its occurrence [6], which can be defined as

$$P(k) = \frac{n_k}{n} \quad k = 0, 1, 2, \dots, L-1 \quad \sum_{k=0}^{L-1} P(k) = 1 \tag{2}$$

where k is the k th gray level in the image $f(m, n)$; n_k is the number of pixels whose gray level is k in $f(m, n)$; n is the total number of pixels of the images; and L the grayscale series. Equalization is actually the transformation of monotone function, that is,

$$g = (g_{\max} - g_{\min}) + g_{\min} \tag{3}$$

After the equalization of grayscales of the road image, the dynamic range of the gray level is larger, highlighting the road marking line. The road markings of the interested image are highlighted after binarization (see Fig. 3).





Fig. 3 The enhanced image of the road

2.2 Road edge detection

For road information, the road edge should be extracted first. After denoising the image, it provides convenience to extract image edge. Recently, there are many methods of edge extraction, including Roberts, Prewitt, Sobel, Isotropic Sobel, Laplacian, and Candy operators [7, 8], wherein Sobel and Candy operators are used in visual navigation. Sobel operator uses the first derivative, which makes the extracted image edges blurred. It is not conducive for further extraction of navigation parameters. Candy operator transforms edge detection into the detection of the maximum of function, which has been widely used because the extracted edges are more accurate. However, it processes more data, which affects the real-time performance of visual navigation of balance vehicle.

The Laplacian operator is a non-directional operator, which uses the second derivative. Thus, the amount of calculation is relatively small, with clear and accurate extracted edges. It is also suitable for improving the image blurring caused by the diffuse reflection of light. Although the operator is sensitive to noise, the image has been substantially suppressed from noise points after preprocessing. Therefore, the Laplacian operator is used to extract image edges. The Laplacian operator is a

second derivative operator, and image edges are detected by the second derivative of the gray level. The range of the pulsed edge can be determined by detecting the upper and lower peaks of the first derivative. The rising and falling edges of the corresponding pulse, namely two edge points of the image, can be determined by detecting zero-crossing of second derivative of pulse profile. For the discrete function $f(x, y)$ of the digital image, its Laplacian operator at the pixel point (x, y) is defined as Eq. (4) [9].

$$\begin{aligned} \nabla^2 f(x, y) &= \nabla_x^2 f(x, y) + \Delta_y^2 f(x, y) \\ &= f(x-1, y-1) + f(x-1, y) \\ &\quad + f(x-1, y+1) + f(x, y-1) \\ &\quad + f(x, y+1) + f(x+1, y-1) \\ &\quad + f(x+1, y) \\ &\quad + f(x+1, y+1) - 8f(x, y) \end{aligned} \quad (4)$$

In the image processing, the Laplacian operator is implemented by template convolution as follows:

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} \quad (5)$$

Since the original image acquired by the video sensor is not intuitive with noise, it is necessary to denoise and binarize before the extraction of road information. The binarization sets the gray level of pixel points in the image to 0 or 255, so the image has only black and white visual effects. It is to simplify the image information and to change the gray levels into 0 and 255 (black and white) for image detection. Figure 4 shows the road image after binarization. Figure 4a and b show the road edges and the road axis after binarization, respectively. The background color of the road is white, and the curbs on both sides are black, with a centerline in the middle of the road.

The coefficients of the corresponding center pixel are positive in the template, and those of other neighboring pixels are negative. The sum of all coefficients is zero, so

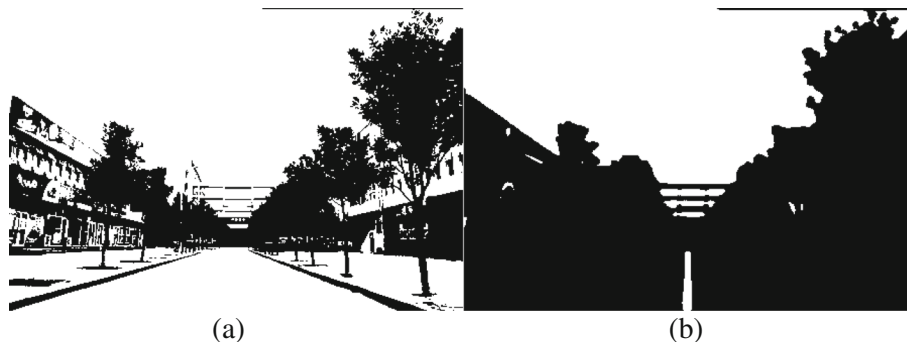


Fig. 4 The road image after binarization. **a** The area of interest is the edge of the road. **b** The area of interest is the central line of the road

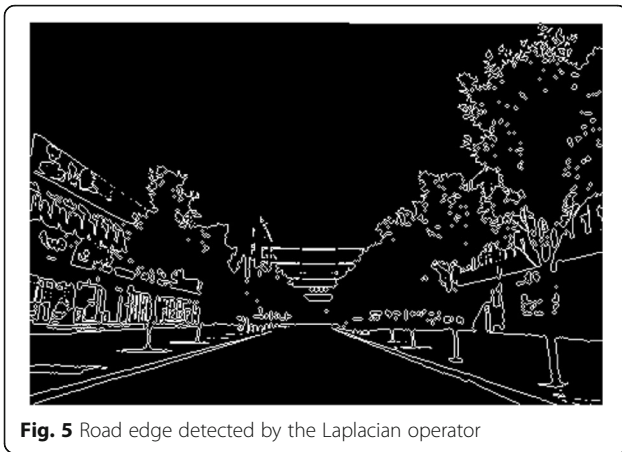


Fig. 5 Road edge detected by the Laplacian operator

the deviation of gray scale does not occur. The Laplacian operator is used for the detection of image edges, and Fig. 5 shows the results. The edge extraction of the Laplacian operator is more accurate, which is convenient for extracting the parameters of relative position of the simulated road.

2.3 Recognition of road axis

The centerline of the road is calculated from road edge data extracted in Section 2.2. It needs to find the black dividing line on both sides. After analysis, the system scans from the white road area where self-balancing vehicle is located to either side until the black line is

scanned or reaching the end of the line. There are three cases for the scanning method:

(1) The black lines are scanned on both sides. The position of the centerline is the arithmetic mean of the positions of the black lines on both sides.

(2) The black line is scanned on one side due to light or other factors on one side, or the other side of the road not photographed (caused by turning). The value of the moment is deduced based on the trend of the values of the first few moments on the side. The centerline of road is calculated by the method in Case (1).

(3) The black line is not scanned on both sides. This is because the camera passes through the intersection or is subject to strong external factors. Then, the current centerline position is directly estimated based on the trend of the past centerline.

According to the three cases, Fig. 6 shows the specific implementation flow. Finally, the centerline position is compared with the horizontal coordinate of the midpoint of the image to determine where the vehicle is, thus obtaining direction deviation for direction control.

Figure 7 shows the recognition results of the centerline of road.

When the road environment is not good, the edge detection and centerline detection signals of the road are not clear, thus requiring further judgment. The road information including boundary position and road width is stored into an array. When the edge position is detected, we select the difference between the current position and the previous boundary position rather than edge

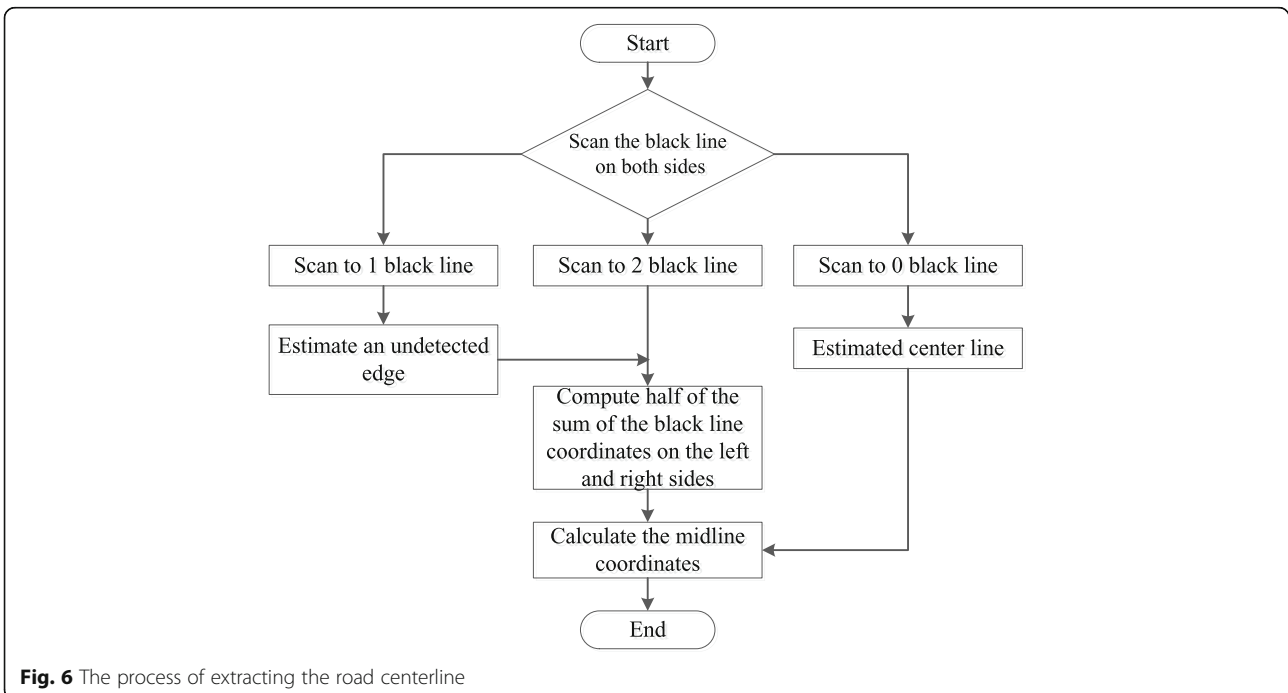


Fig. 6 The process of extracting the road centerline

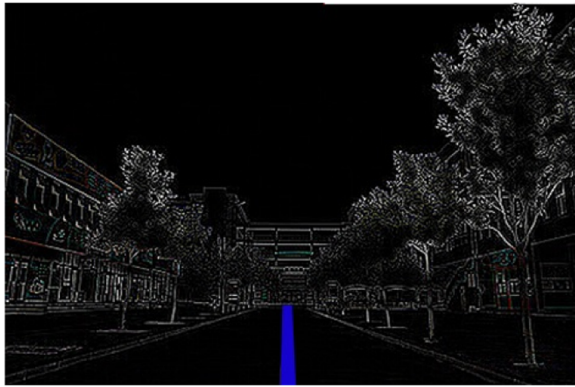


Fig. 7 Extraction results of road centerline

position for calculation. If the difference is less than a predetermined threshold, the location is considered to be a boundary; otherwise, it is not. Since the road boundary is continuous, the period of the CCD collecting boundary information is short (usually about 10 ms). When the balance vehicle travels at a certain speed, the boundary positions that the CCD continuously collect do not change too much [10]. Thus, the change is within a certain controllable range, namely the threshold is determinable.

After determining the boundary position, the centerline of road can be found. The centerline position is equal to the mean of the sum of the left and right boundary values. The direction deviation can be obtained by subtracting the expected value of the position of balance vehicle from the current centerline value of the road. Through the PD algorithm, the speed control output value, duty ratio of pulse-width modulation, can be calculated. In order to ensure that the vertical control is not affected, the subroutine call cycle for the directional control output value is

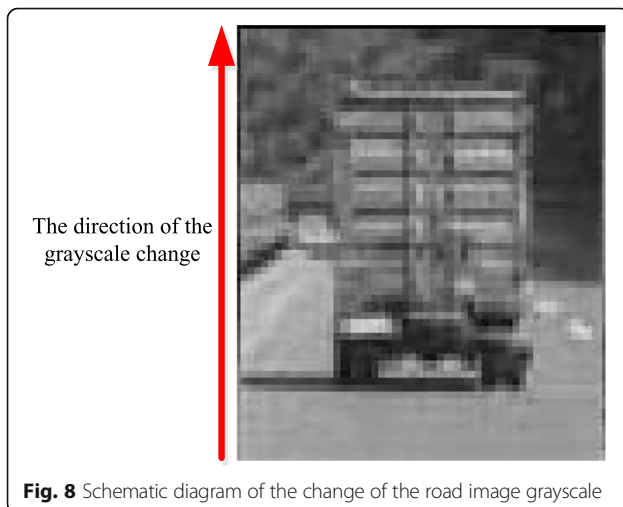


Fig. 8 Schematic diagram of the change of the road image grayscale

set to 10 ms. The obtained directional control output is divided into 10 parts, outputting one tenth of directional control output value each millisecond.

2.4 Existence detection of forward obstacles combined with grayscale

The existence detection of forward obstacles collects road traffic environment through CCD vision sensors. Visual cognition is used to simulate the habits and characteristics of human vision, with rich video images for identifying whether there are forward obstacles. Since the self-balancing vehicle travels fast, the algorithm should have real time.

Under the condition of high visibility, the bottom edge of the forward vehicle tail is affected by sunlight, projecting a shadow area on the road surface. The brightness of other non-projection road area is higher than that of the shadow area, that is, the gray level of other areas is larger than that of pavement image area under the bottom of the vehicle. Figure 8 shows the image pixel of pavement is in continuous variation of grayscale from the near to the distance. The pixel gray level near the bottom of the vehicle is significantly lower than that of the surrounding pixels, with large falling of grayscale distribution. In the determined forward vehicle detection and tracking area, the abrupt change of road image grayscales [11] can be utilized to calculate the average gray level of each row of pixels from the bottom to the top of the road image. It can preliminarily determine whether there is a possible target vehicle in the road image by searching for the area with large grayscale mean.

$$G(r) = \frac{1}{b(r)_r - b(r)_l + 1} \sum_{c=b(r)_l}^{b(r)_r} g(r, c) \quad (6)$$

where $G(r)$ is the mean of grayscales, $b(r)_l$ is the left-end pixel coordinate of the r th row in the search area, $b(r)_r$ is the right-end pixel coordinate of the L th row in the search area, and $g(r, c)$ is the gray level of pixel point (r, c) . In the area of forward vehicle detection, tracking and searching, the row where $G(r)$ has a step change corresponds to the lowermost edge of the vehicle (assuming the forward target has been detected). If without step change, it is considered that there is no forward target vehicle in the current detection area [12].

The grayscale mean is combined with the Laplacian operator for the edge detection on the forward obstacle. Figures 9, 10, 11, and 12 show the result.

2.5 Group warning mechanism based on vision of car body

The information detection of the road image lays the foundation for vehicle attitude estimation. The balance vehicle warns when approaching the road axis, edges, or forward obstacles, which is the multi-information source



Fig. 9 Sample training set

fusion method. The work used a multi-agent system model for early warning [13]. Agent is an abstract entity with autonomic characteristics, which has perception, problem solving, and communication with the outside world. Group agent solves the problem after negotiation. The group agent model in the work was composed of the inference engine (for sensing environmental factors and their changing feature perceptrons, with the ability to reason and solve problems) and the actuator for environment or other agents (see Fig. 13).

Group agent is a distributed autonomous system composed of multiple agents, completing tasks through negotiation. Each agent plays a different role in system operation. The early warning model of group agent collaboration in the work included perceptual agent, decision agent, and behavior agent, based on predesigned distance thresholds for early warning of possible potential dangers to avoid collision accidents.

The threshold is set according to the CCD sensor, and TSL1401 was used in the work with 128-pixel linearity. The CCD acquisition feedback is the gray level of 128 points on a horizontal line of the road, with the range of 0–255, the black gradation of 0 and the white gradation of 255. The gray level of 128 points was stored in an

array, and the maximum and minimum of grayscales in the array were found by comparison. The mean was calculated, used as a preset threshold. Figure 14 shows the warning mechanism of a group agent of cameras.

3 Experimental results

During the operation of the two-wheel self-balancing vehicle, it is necessary to complete vertical control, speed control, direction control, and dangerous attitude warning, which are achieved by changing the speed of the wheels.

3.1 Vertical control

The vehicle has its mechanical balance point in the vertical direction to main a short balance and tilt forward or backward. According to Newton's laws of motion, an opposite direction is needed for balance, but there is no way for the outside world to exert force on it. According to D'Alembert's principle, when the vehicle tilts forward, a forward acceleration is applied to ensure the balance; otherwise, a backward acceleration is applied [14–16].

Figure 15 shows the vertical control of the balancing vehicle in the actual system. When the vehicle jitters,



Fig. 10 The results low-pass filtering



Fig. 11 The results of image binarization

the noise is large, and the output signal of the accelerometer cannot stably reflect the inclination of the vehicle. The gyroscope can measure the angular velocity when the vehicle is jittering, but there are some slight deviations and drifts for the angular velocity signal. After the integral operation, the error increases with time. MMA7260 is used to measure the acceleration of the vehicle, and MPU6050 to measure the angular velocity. Through the Kalman filtering [17], the two can be combined to obtain the real-time, accurate position angle. The angle deviation value θ , obtained by subtracting the angle at the equilibrium position from the real-time angle, is the direction and magnitude of tilt. As the input of PD algorithm, θ can calculate the output value pulse-width modulation (PWM). The motor is driven to accelerate or decelerate the vehicle to maintain the vehicle vertical. The larger θ means the larger duty ratio of PWM, the faster motor speed, and the greater acceleration.

3.2 Speed control

3.2.1 Speed control method

In order to stabilize the vehicle, it is necessary to measure the actual velocity with closed-loop control. The

encoder counts the number of pulses rotated by the two wheels to calculate the true velocity. The velocity deviation value is obtained by subtracting the true velocity from the desired velocity. Then, through the PI algorithm, the output value of velocity control, the duty ratio of PWM [18], can be calculated. As it needs to ensure that the velocity control does not affect the vertical control, the calling cycle of the subroutine for obtaining the output value of velocity control is set to 100 ms. The output value is divided into 100 parts, outputting 1% of the output value of velocity control every millisecond. Moreover, the calculated output value is limited in the program to minimize the influence of velocity control on vertical control.

3.2.2 PID algorithm

PID algorithm [19] is a control method in which the proportional (P), integral (I), and differential (D) are linearly combined to form the controlled quantity for adjusting the controlled object. The output of the proportional adjustment is proportional to the input deviation, and as long as the deviation occurs, a proportional adjustment can occur in time. The output of the integral adjustment is proportional to the integral of input



Fig. 12 Results of edge detection of obstacles ahead combined the gray average with the Laplacian operator

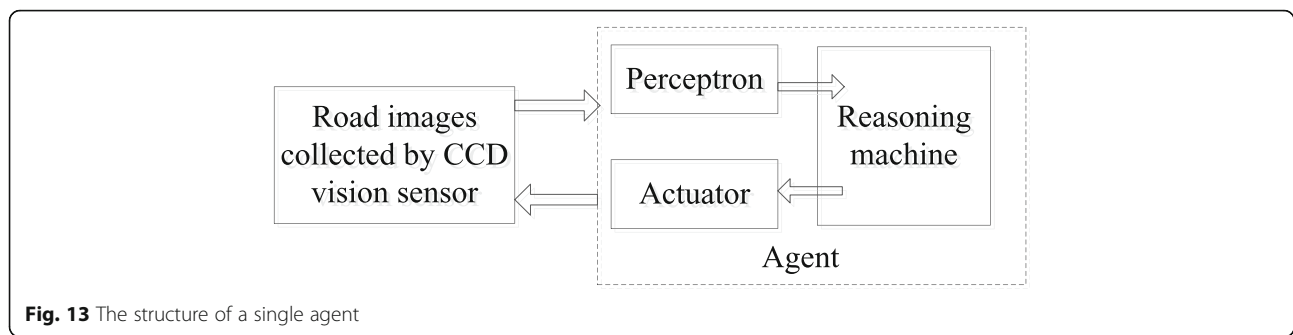


Fig. 13 The structure of a single agent

deviation. The output of differential adjustment is proportional to the differential of input deviation. If the proportional and the integral are combined, PI algorithm is formed. If the proportional and the differential are combined, PD algorithm is formed. To adjust the parameters of PID, it should successively adjust the proportional, integral, and differential parameters. If the adjustment is not obvious, the proportional parameters should be increased. If the adjustment fluctuates back and forth for a long time, the integral parameters should be increased. If the frequency of the fluctuation of the adjustment is large, the differential parameter should be increased.

(1) Smooth noise suppression of syncretic median filtering of sensor data.

As the sensor is disturbed by complicated physical factors, and the electric signal output by the sensor is also disturbed by the external electromagnetic field, the output signal generates ripple or oscillation to interfere with the PID controller [20]. Therefore, it is necessary to use the filtering algorithm for sensor signal. Sampling and processing sensor data in advance ensure the efficiency and reliability of the PID controller.

① First-order low-pass filtering

The velocity measured by the motor encoder uses the first-order complementary filtering to stabilize the output velocity of motor encoder, reducing the impact of abrupt change of velocity on the vertical control of vehicle.

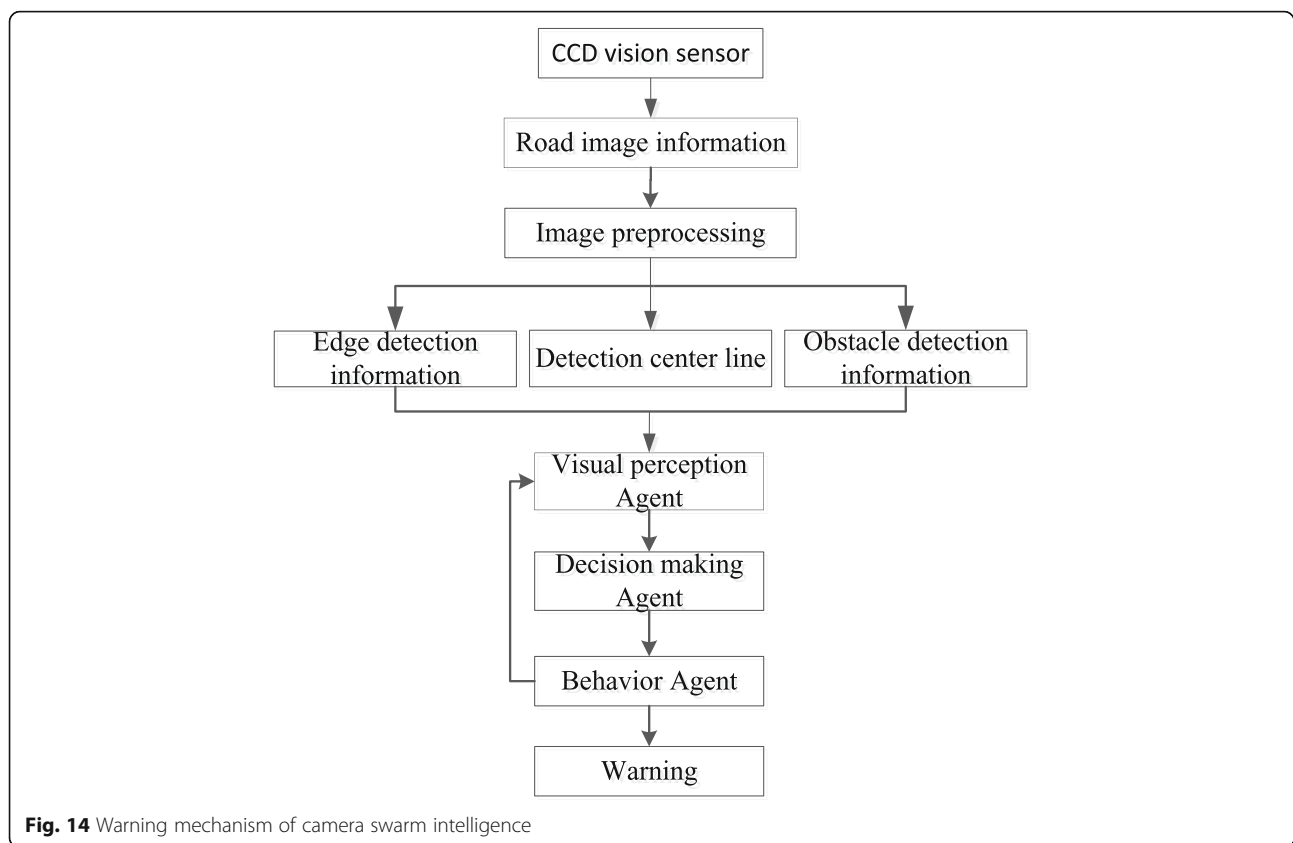


Fig. 14 Warning mechanism of camera swarm intelligence

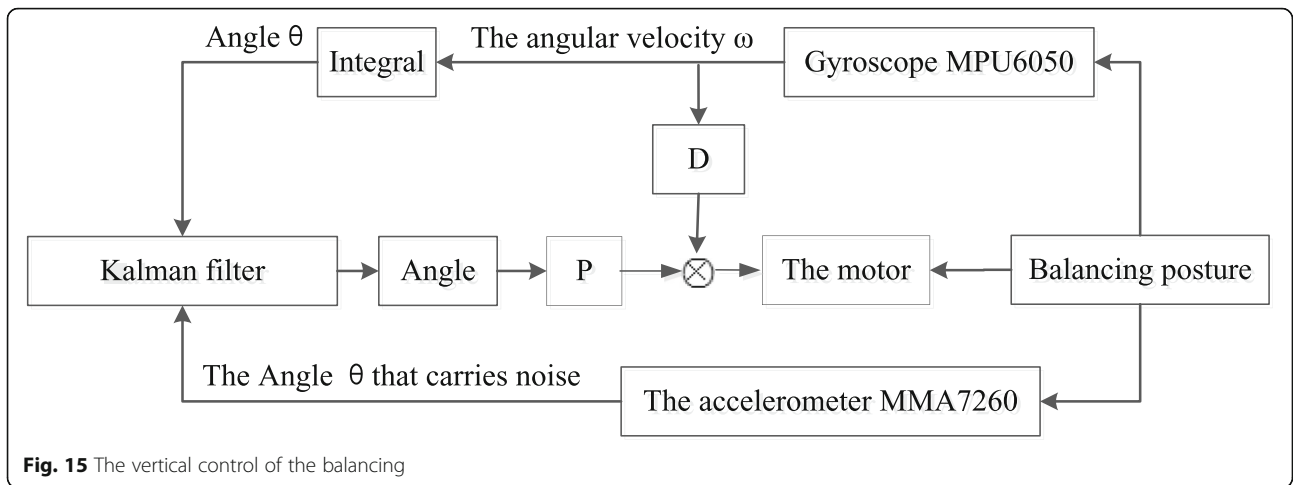


Fig. 15 The vertical control of the balancing

The following is the code fragment of first-order low-pass filtering:

```

main.c
speed *=0.8;
speed +=speed_L*0.2;
speed_OUT+=speed;
main.c

```

② Kalman filtering

The data output by the gyroscope uses Kalman filtering, making the obtained angle smoother and suppressing burr and jitter in the waveform, which makes the control efficient [21].

The following is the code fragment of first-order low-pass filtering:

```

control.c
void Kalman_Filter(float Accel,float Gyro)
{
angle+=(Gyro - Q_bias) * dt;
Pdot[0]=Q_angle - PP[0][1] - PP[1][0];
Pdot[1]=-PP[1][1];
Pdot[2]=-PP[1][1];
Pdot[3]=Q_gyro;
PP[0][0] += Pdot[0] * dt;
PP[0][1] += Pdot[1] * dt;
Angle_err = Accel - angle;
PCt_0 = C_0 * PP[0][0];
PCt_1 = C_0 * PP[1][0];
E = R_angle + C_0 * PCt_0;
K_0 = PCt_0 / E;
K_1 = PCt_1 / E;
t_0 = PCt_0;
t_1 = C_0 * PP[0][1];
t_1 = C_0 * PP[0][1];
PP[0][0] -= K_0 * t_0;
PP[0][1] -= K_0 * t_1;
PP[1][0] -= K_1 * t_0;
PP[1][1] -= K_1 * t_1;
angle += K_0 * Angle_err;
Q_bias+= K_1 * Angle_err;
angle_dot = Gyro - Q_bias;
}
control.c

```

Kalman filtering estimates an expected value PP [0]

according to the last sampling result, and then verifies covariance based on the parameters of the sensor, thus achieving good filtering.

(2) PID algorithm of the cascade control

① Positional PI algorithm of the speed loop

In the velocity control, the PI control algorithm is used to integrate the displacement to ensure that the balancing vehicle does not drift left and right.

Figure 16 shows the velocity feedback is positive. After setting the velocity, the motor breaks vertical balance. Thus, the vehicle moves forward and backward [22]. If the deviation after velocity sampling is e and the sampling frequency is n , PI control algorithm is simplified as Eq. (7).

$$O(n) = K_p e(n) + K_i \sum_{j=0}^n e(n) \tag{7}$$

The code snippets of PI control are as follows:

```

control.c
Encoder_Least=(Encoder_Left+Encoder_Right)-0;
Encoder *= 0.8;
Encoder += Encoder_Least*0.2;
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||Flag_Stop==1) Encoder_Integral=0;
return Velocity;
control.c

```

② Positional PD algorithm of angle loop

PD control is adopted in the vertical control, with fast response and short adjustment time. It is more suitable for the system with large inertia. Figure 17 shows the vertical feedback uses the negative feedback by referring to the tilt angle and angular velocity of the gyroscope. When deviating from the set angle, the vehicle makes it return to the origin [23].

It is supposed that θ is the angle collected by the gyroscope, β is the angle at the equilibrium position, and angular velocity is the differential of angular deviation.

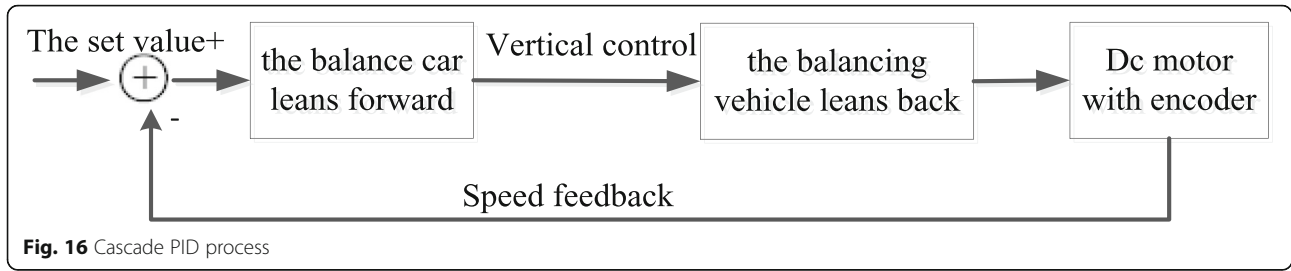


Fig. 16 Cascade PID process

$$O(k) = K_p e(\theta - \beta) + K_d \theta' \tag{8}$$

The following is a code snippet of vertical PD control:

```
control.c
int balance(float Angle,float Gyro)
{
    float Bias,kp=310,kd=1.3;
    int balance;
    Bias=Angle-ZHONGZHI;
    balance=kp*Bias+Gyro*kd;
    return balance;
}
control.c
```

```
control.c
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;
control.c
```

4 Hardware structure design of balance vehicle

In this design, popular balance vehicle is taken as the control background platform. Gyro sensor, motor speed encoder, and ultrasonic sensor are equipped for data acquisition [25]. The sensor data are conducted with attitude feedback and algorithm output taking STM32 processor as the control center. DC motor driven by integrated H-bridge is used for output control to form a complete closed-loop control system [26].

A micro operating system is used for the software control. Taking the clock output of MPU as control time base and 10 ms as rotation scheduling time, the data flow of the gyroscope is received for the Kalman filtering fusion and attitude analysis of software, analyzing the calculation speed of encoder. Advanced PID algorithm for cascade feedback control smoothly controls the motor output adjustment attitude, thus improving the real-time angle and stability of the balance vehicle as a whole.

③ Cascade PID control algorithm for a balancing vehicle

The core of the balancing vehicle is divided into vertical loop control and velocity loop control, which adopt the PID control, wherein vertical control is dominant. Velocity loop control is used as an aid to change the velocity of balancing vehicle to control the motion of the vehicle. Combined with the advanced PID and synchronous working, a cascade PID controller is formed to control the vehicle [24].

Figure 18 shows based on the control of vertical loop, the positive feedback interference from the velocity loop is added. In terms of cascade PID in Eqs. (7) and (8), and the velocity loop as the input of the vertical loop, the following can be obtained.

$$u(n) = K_p e * \theta + K_d \theta' - K_p * \left[K_{p1} * e(n) + K_i \sum_{j=0}^n e(n) \right] \tag{9}$$

The following is a code snippet of cascade PID:

4.1 Circuit structure of balance vehicle

4.1.1 Circuit diagram of balance vehicle

Figure 19 shows the circuit diagram of a self-balancing vehicle control system designed in the work [27], including Main controller 1, Motor drive circuit 3, Gyroscope circuit 5, Display circuit 4 and Power supply 2. Motor

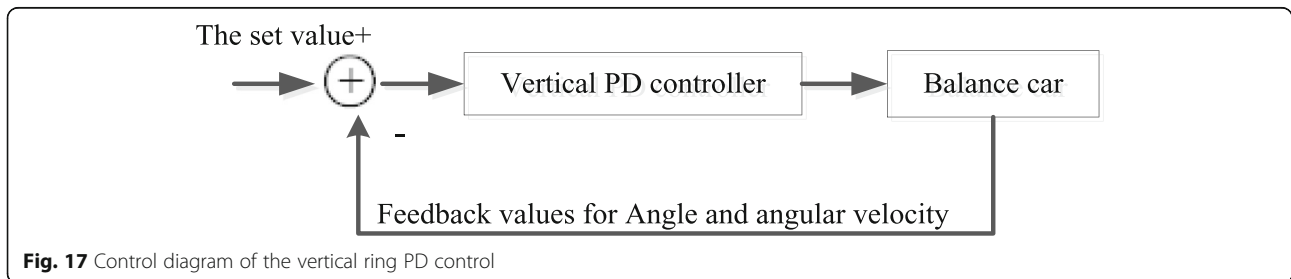


Fig. 17 Control diagram of the vertical ring PD control

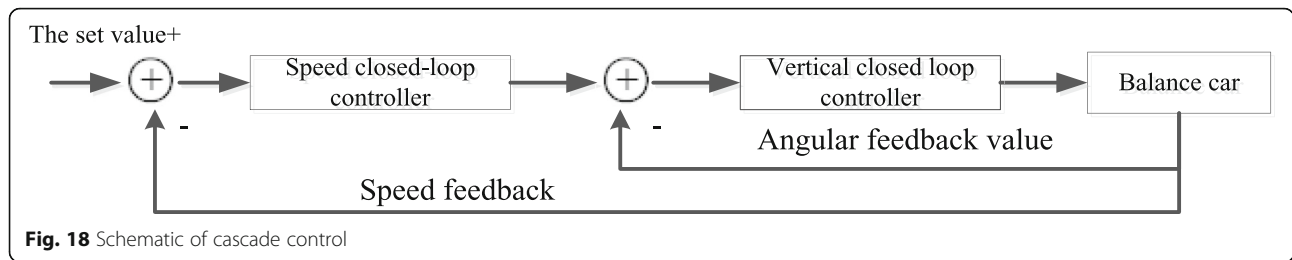


Fig. 18 Schematic of cascade control

drive circuit 3, Gyro circuit 5 and Display circuit 4 are connected to Main controller 1.

4.1.2 Control system STM32

The MCU selects the STM32C8T6 packaged in the LQFP48 pin of the Cortex-M3 core (see Fig. 20). With main frequency of 72 MHz and high-speed computing core, it does not waste too much residual IO. Most peripherals of a single-chip microcomputer are used to achieve the highest cost performance [28].

4.1.3 Vision subsystem design

A self-balancing vehicle vision system circuit includes core chip, filtering, clock, and output signal circuits. In the work, we select the chip OV7620 (see Fig. 21) [28]. The filtering circuit is composed of various capacitors and inductors to ensure the stability of data acquisition by a camera. The clock circuit is provided by a 27-MHz external crystal oscillator to ensure proper operation of OV7620. The output signal circuit mainly realizes data transmission between the camera and controller, setting working mode of the camera by the controller.

4.1.4 Perception subsystem design

Perception subsystem is used to detect the body posture of the self-balancing vehicle. The system perceives the external environment, realizing feedback control. When the self-balancing vehicle moves, the controller obtains the posture information of balance vehicle from the attitude sensor, thus calculating the running state of balance vehicle. According to the corresponding PID algorithm, the motor is controlled to complete the erection, acceleration, and turning functions. Accurate attitude information must be provided to achieve balance control [29–31]. The accuracy of detection module has a great influence on the balance control of system. The system uses the inertial sensor MEMS gyroscope MPU6050 and accelerometer MMA7260 for attitude detection.

(1) Accelerometer design

MMA7260 is a miniature capacitive acceleration sensor with temperature compensation and signal conditioning. In this system, the two pins g-select1 and g-select2 are simultaneously earthed to select the precision of 800 mV/g (See Fig. 22) [32].

The controller voltage is 5 V, and the accelerometer is 3.3 V. Therefore, a 5 V-to-3.3 V conversion circuit is required. Based on the sensitivity of attitude chip to noise, the MIC5205BM chip is used for 3.3 V voltage

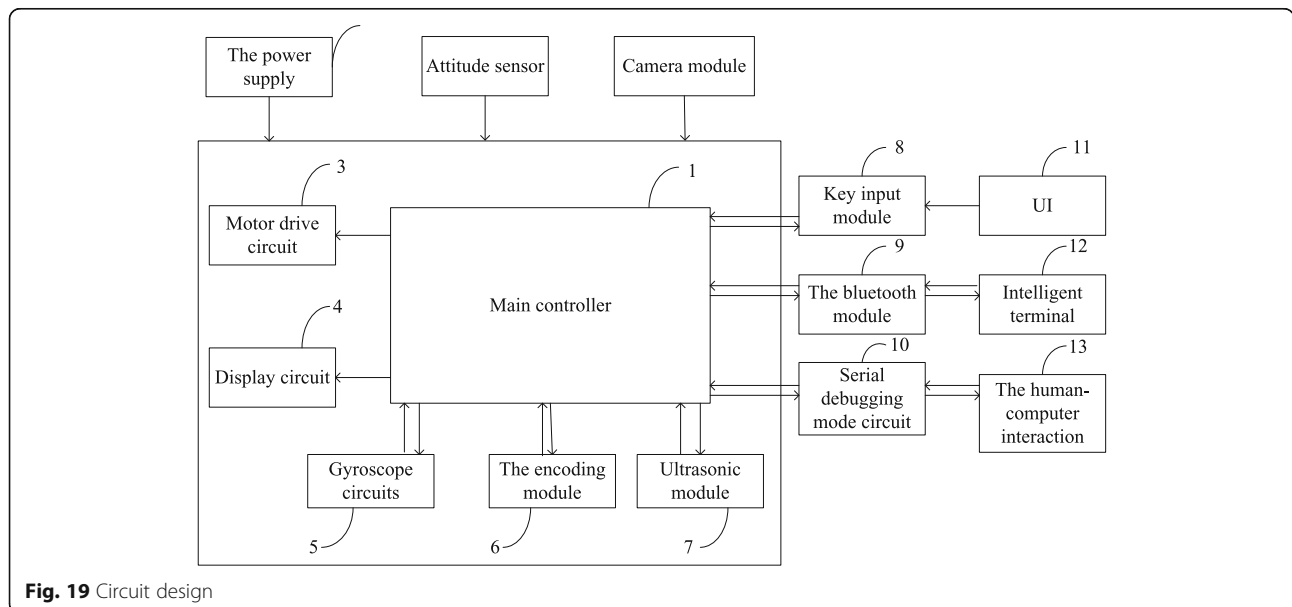


Fig. 19 Circuit design

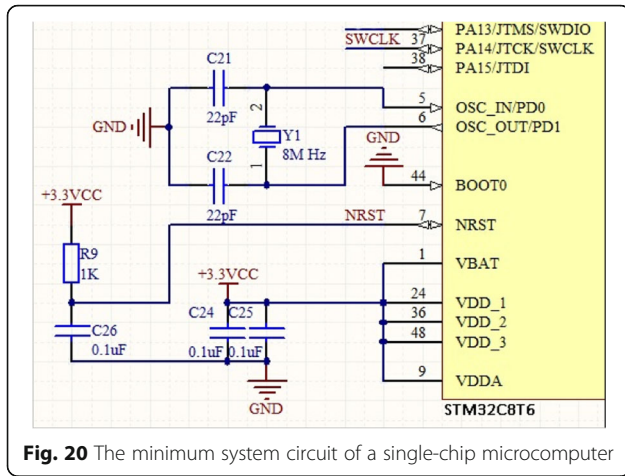


Fig. 20 The minimum system circuit of a single-chip microcomputer

conversion. The chip has ultra-low noise output and very low voltage drop. Figure 23 shows the circuit diagram [32].

(2) Design of gyroscope MPU6050

In Fig. 24, the MPU6050 integrates a three-axis acceleration and gyroscope in a $4 \times 4 \times 0.9$ QFN package to track fast motions. The data are transmitted to the single-chip microcomputer using the I²C bus. Hardware DMP in the MPU6050 can filter the gyroscope data for tuning output. It is a mature, cheap gyroscope chip.

4.1.5 MP2359 power module design

The MP2359 is a third-generation DCDC step-down chip using CMOS technology. Its performance exceeds that of LM2596. It normally works when the internal oscillation frequency reaches 1.4 MHz, and the inductance is 4.7 μ H (See Fig. 25). The SOT23-6 is used for packaging to minimize the PCB footprint. The MP2359 has an input voltage range of 4.5–24 V and a maximum output current of 1.2 A. It is more than enough for the 12-V-powered balancing vehicle control system to ensure better power supply system.

4.1.6 L298P motor driver L298P

The L298P driver is similar to the L298N, but the L298P is packaged in a POWER-SO20 package, which saves PCB size and avoids excessive motor drive volume (see Fig. 26). The L298P is integrated with dual H-bridge drive to bear 100 kHz PWM speed regulation. With maximum output current of 2 A and maximum current of 4 A, the single bridge is able to drive the motor of balance vehicle.

4.1.7 USB-to-serial-port circuit

Figure 27 shows the serial debugging circuit connected to computer. The integrated USB-to-serial-port chip CH340G is used as the drive to facilitate parameter debugging and observation of the vehicle.

4.1.8 Bluetooth communication module

The Bluetooth module is used to achieve Bluetooth communication with the smart terminal. The display circuit is an OLED display circuit. Bluetooth communication module HC-05 is used to realize the communication between the processor and mobile APP terminal. The mobile APP terminal can control the switching and movement of balance vehicle. The PID parameters of control algorithm can also be debugged in the development mode. The communication bus between Bluetooth communication module and processor is a serial port, which is an asynchronous communication interface. The design uses a communication protocol frame, including frame head, function code, data frame and check frame, to ensure reliable data transmission, avoiding misinformation of data. Operating voltage of the HC-05 Bluetooth module (5 V) is directly connected to the power module of 5 V output. A resistor of 1 to 2 K is connected in series between RX pin of module and TX pin of a single-chip microcomputer. Then, a voltage divider resistor is connected to the ground to prevent the RX pin from burning. Figure 28 shows the interface circuit and connection accessory diagram of the Bluetooth module.



Fig. 21 Schematic diagram and physical diagram of OV7620

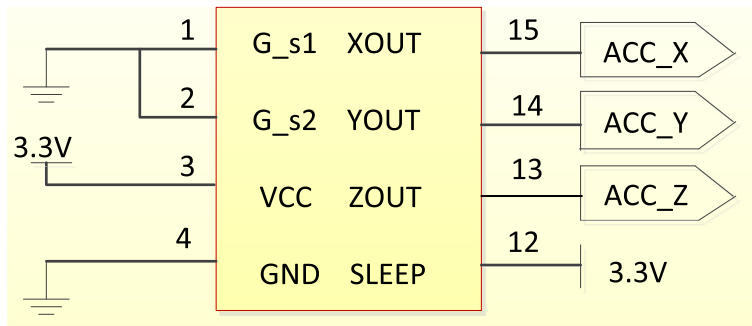


Fig. 22 Circuit diagram of accelerometer MMA7260

4.2 Circuit design details

The quality of circuit design is reflected in the details. This circuit design fully involves the circuit characteristics of the balance vehicle. According to circuit requirements, some peripheral circuits are added to accommodate the operation of the balance vehicle.

In Fig. 29, the level switching circuit allows a single-chip microcomputer with a 3.3-V level to communicate with other peripherals with a 5-V TTL level. It avoids the burnout of the controller or abnormal communication caused by voltage difference.

Figure 30 shows the photocoupling isolation circuit. The optocoupler has the characteristics of complete isolation and high-speed conduction. The motor drive chip L298P with a large current is isolated from a single-chip microcomputer, preventing the burnout of a single-chip microcomputer because of surge current generated by a motor.

Good hardware design of PCB ensures stable operation of software while adding functions required by the system. For high-current motors, the power supply is equipped with widened wire and plated copper. Important modules are interfaced with add-in modules in PCB design. The double insurance strategy is adopted to prevent circuit design errors from affecting the function of whole board.

According to the principle of top-level wiring, the 0R resistor jumper is used in multiple places of top layer to preserve complete ground layer, greatly increasing the connectivity integrity and anti-interference ability of the circuit (see Fig. 31).

In modular design of PCB, the circuit is divided into power supply, main control, gyroscope, and motor drive (see Fig. 32). Separated layout and wiring enhance the independence of the board module and overall anti-interference ability.

A 45° polyline is used on the top-level wiring to ensure a smooth signal. Meanwhile, the places with large currents are thickened, including the power and motor output (see Fig. 33).

4.3 Mechanical structure of the balance vehicle

The balance vehicle has an aluminum alloy chassis. The bottom of chassis is equipped with motor bracket to fix the motor. The core PCB control board is installed on screw holes of the chassis by a copper column to form a complete body (see Fig. 34).

The main parts are connected as follows.

The motor drive, gyroscope, and display circuits are respectively connected to the main controller, respectively. The motor drive circuit includes the motor and

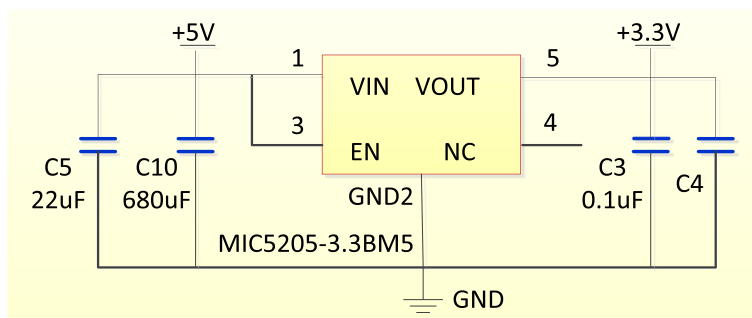


Fig. 23 Regulating circuit of 5V and 3.3V

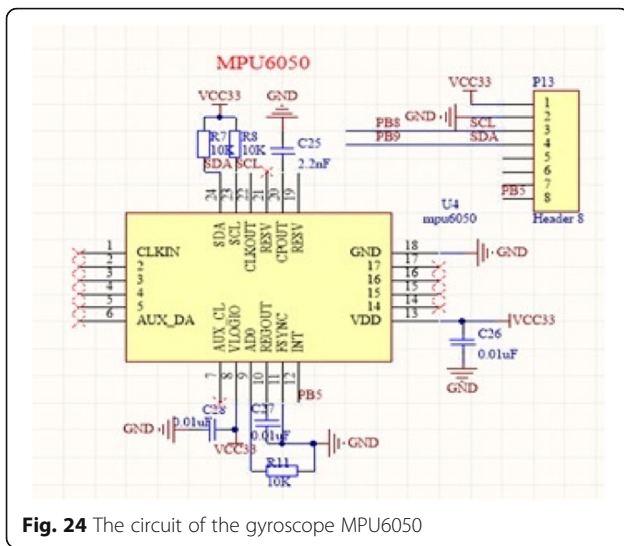


Fig. 24 The circuit of the gyroscope MPU6050

motor driver. The motor driver adopts a L298P drive chip, including an integrated dual H-bridge driver circuit.

The power circuit supplies the motor driver by converting the input DC into 5 VDC. The power includes power circuit and voltage conversion circuit. The power circuit is connected with a motor drive circuit. The voltage conversion circuit is connected with the main controller, gyroscope, and display circuits, respectively. The power circuit includes an MP2359 conversion chip. The power input terminal is connected to the anode of diode D11. The cathode of the diode D11 is connected to the power input voltage pin of a MP2359 conversion chip. The cathode of the diode D11 is simultaneously connected to the resistor R6 and the enable end of a MP2359 conversion chip. The switch pin of the MP2359 conversion chip is connected with the inductor L2 and the resistor R5 to insert the feedback pin of the MP2359 conversion chip. The resistor R5 is connected in parallel with the capacitor C18. The inductor L2 is 4.7 μ H, the resistor R6 100 k Ω , the resistor R5 73 k Ω , and the capacitor C18 15 pF.

The voltage conversion circuit converts 5 VDC from the power supply circuit into 3.3 VDC, which supplies the main controller, gyroscope, and display circuits. The gyroscope module is used to collect gravity and acceleration data, including gyroscope circuit connected by the MPU6050 chip. The voltage conversion circuit uses a AMS1117-3.3 regulator. The 5 VDC is connected to the anode of diode D1. The cathode of diode D1 is connected to the input of AMS1117-3.3. After short connection, two outputs of AMS1117-3.3 are connected in parallel to electrolytic capacitor C5 and capacitor C4. C5 and C4 are grounded at the other end. The AMS1117-3.3 outputs 3.3 VDC.

The main controller is STM32. The IO interfaces of STM32 are connected to the coding and ultrasonic modules, respectively. The USART1 interface is connected to the serial debugging circuit; the USART2 interface is connected to the Bluetooth module, which is wirelessly connected with the intelligent terminal. Smart terminal includes a mobile phone APP. The USART3 interface is connected to a key input module. Hall encoder is used for a coding module.

The serial port debugging circuit and key input module are connected with the human-machine interface. The serial port debugging circuit is connected to the STM32 main controller through a USB-to-serial-port chip. The USB-to-serial-port chip is CH340G.

The CCD-based optocoupling isolation circuit is connected to a main controller at one end and a motor drive circuit at the other end. It is used to completely isolate the motor drive chip L298P with a large current from the STM32 main controller.

Figure 35 shows the overall assembly diagram.

4.4 System parameter debugging

In the actual debugging process of balance vehicle, there are various unexpected problems in detail, which deepens the understanding of theoretical knowledge.

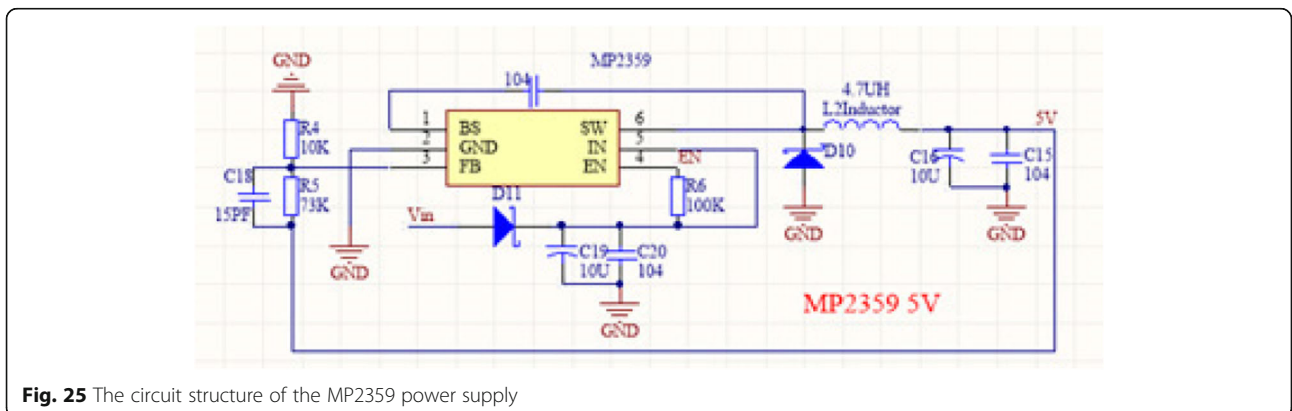


Fig. 25 The circuit structure of the MP2359 power supply

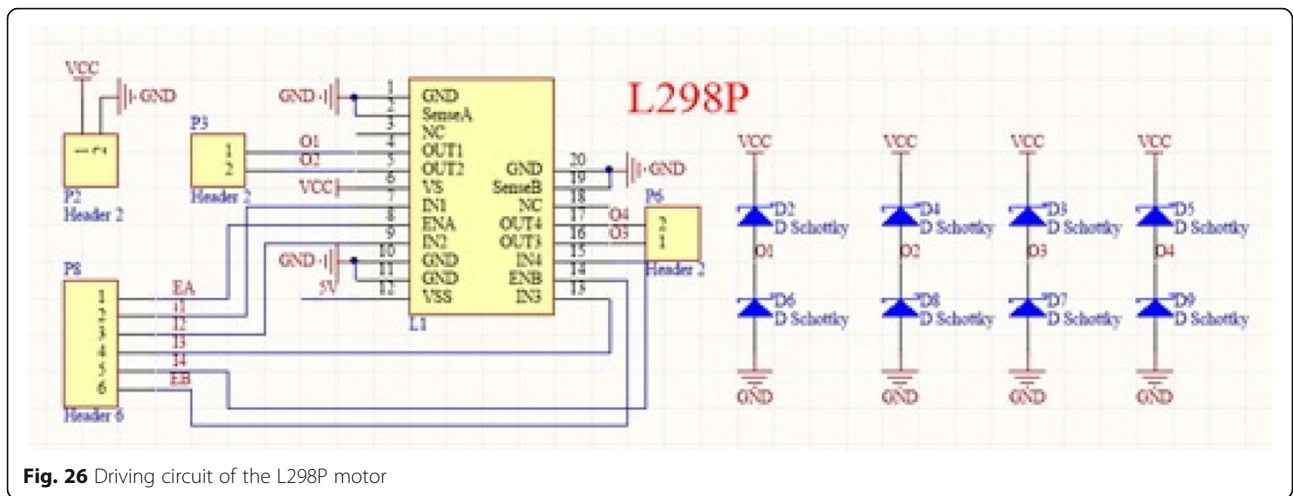


Fig. 26 Driving circuit of the L298P motor

4.4.1 Output of sensor MPU6050

The MPU6050 is an MPU for I²C communication. The program uses its I²C interface to receive six sets of raw data of three-axis acceleration and three-axis gyroscopes. The serial port is used for printout. The INT pin of MPU6050 outputs square wave of 200 Hz, which is used as the sampling time base of the MCUPID algorithm. Figure 36 shows the output waveform.

In Fig. 37, the gyroscope outputs 5-ms pulse square wave as the sampling time base of the controller digital PID. Meanwhile, the gyroscope DMP is normally initialized.

(1) First-order balance low-pass filtering output

The differences between the last sensed value fused by software and two filtering results are multiplied by the first-order filtering coefficient, and then added. Most smooth signals pass through the filter to obtain smooth waveform. Motor encoder outputs square wave by the Hall element. After first-order complementary filtering, the speed becomes relatively smooth, thus reducing the interference to vertical loop.

Filtering result = last filtering result × (1 – filter coefficient) + (last filtering result – sampling this time) × filter coefficient.

In Fig. 38, anonymous host computer is used to compare the original data of a motor speed encoder with the data subjected to first-order low-pass filtering. It is found that the first-order low-pass filtering significantly leads to smoother output waveform.

(2) Kalman filtering output

Kalman filtering algorithm involves the idea of optimal estimation. Physical quantities such as velocity, displacement, and acceleration cannot be mutated. After recording the measured value at the last moment and white Gaussian noise, we estimate the current state of the system by the formula. With a good filtering effect, this algorithm is widely used in temperature and missile speed changes.

Figure 39 shows that the data output by a gyroscope becomes smoother after Kalman filtering. The gyroscope easily fluctuates with environmental interference to produce smoother output (the red is the waveform using Kalman filtering).

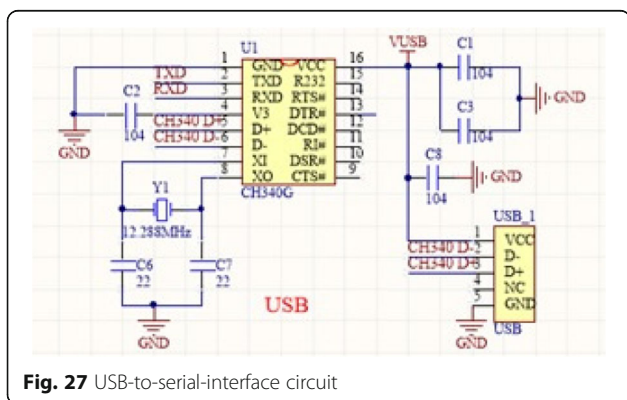


Fig. 27 USB-to-serial-interface circuit

4.4.2 Operation result of double closed-loop control motor

(1) Speed loop PID algorithm debugging

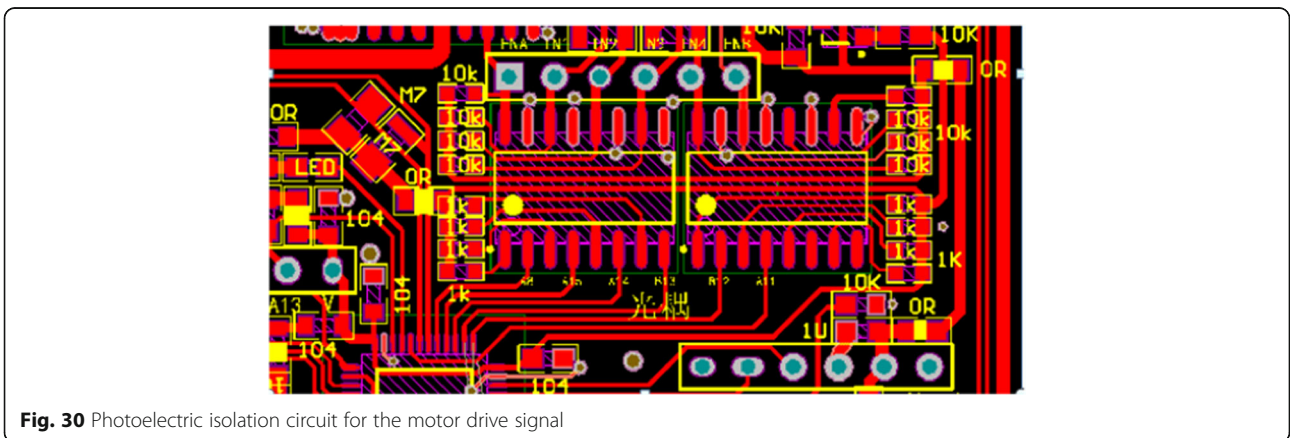
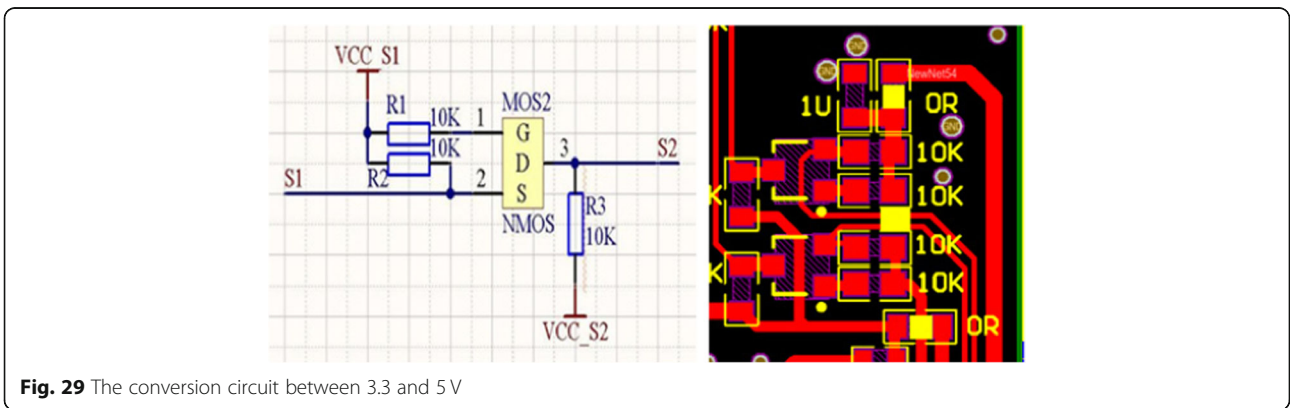
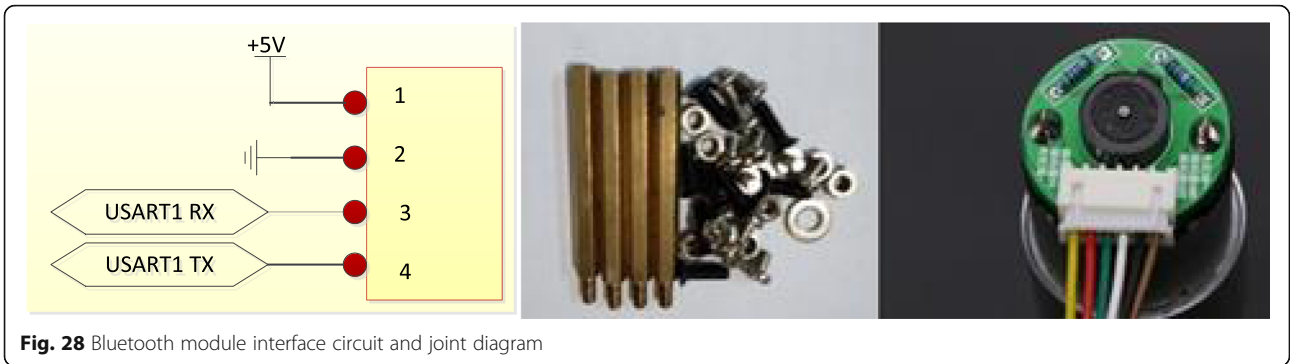
Positional PI control algorithm is used in speed loop to ensure the accuracy of speed integral displacement, reducing the steady-state error of system.

The PID algorithm is simulated by MATLAB. It is found that the PI control has slow response, and Parameter I greatly affects the system (see Fig. 40).

(2) Debugging of angle loop incremental PID algorithm

Angle ring adopts PD control. The control response speed is improved to derive better dynamic response characteristic of a balance car vehicle (see Fig. 41).

(3) Cascade advanced PID integration



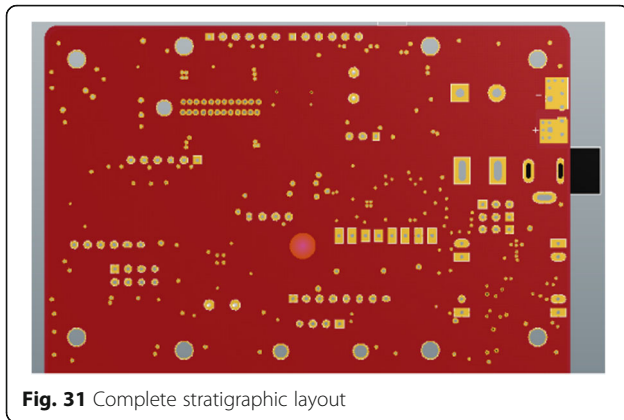


Fig. 31 Complete stratigraphic layout

Based on the idea of advanced PID, the PI control is conducted with gearshift integral in the cascade PID, and the PD control is incompletely differentiated. The two equations are integrated and fed back to the motor PWM output, thus improving the efficiency of cascade PID.

5 Discussion

PID parameter-tuning is complicated and time-consuming. The debugging and exploration of long-time PID needs a deep understanding. The PID coefficients of upright and speed loops of vehicle are separately calculated for overall observation and fine tuning.

When the D coefficient is 0, the sign of P is changed to observe the phenomenon. It is compared with a motor rotation phenomenon according to balance principle of a vehicle. When the motor has negative feedback for the upright loop, the parameter polarity of P is correct. After determining the sign of P , the calculation of the size range of the P starts. The approximate range of P is determined according to motor PWM fullness and range of angular deviation. (For example, in the vertical ring, the PWM is 7200 (100%); the angle of inclination is about $\pm 20^\circ$, so P should be in the range of 0–360). After that, P is continuously increased until the balance vehicle can be quickly

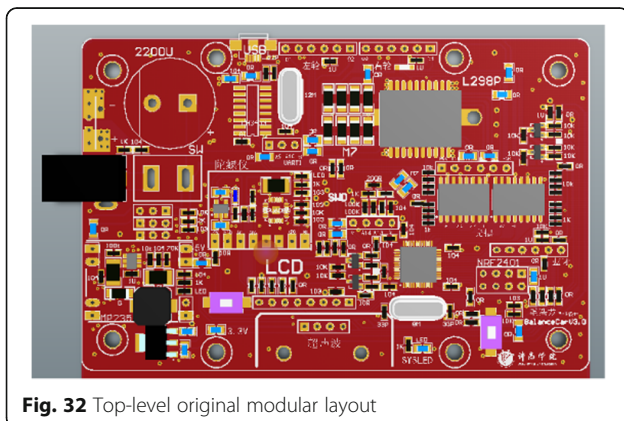


Fig. 32 Top-level original modular layout

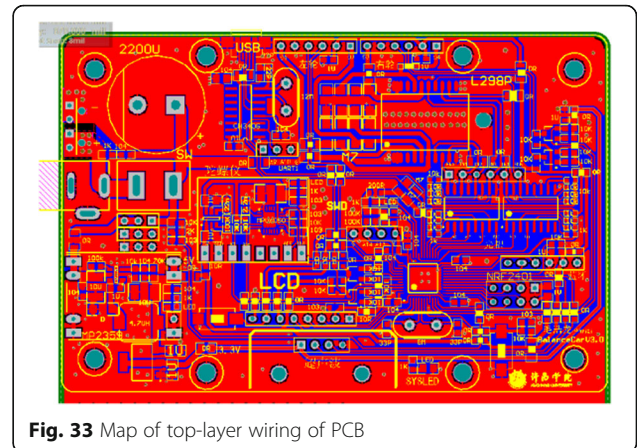


Fig. 33 Map of top-layer wiring of PCB

adjusted, thus causing repeated oscillation of the vehicle body. Here, P is close to the ideal value. Then, D is adjusted. Firstly, the sign of D is judged by observing the motor rotation feedback (positive/negative) according to negative feedback method of vertical loop. Then, the PD adjustment is closed to tune the parameter of PI adjustment. The balance vehicle is basically kept in the original position (without significant changes) by the PI adjustment. According to the experience, the PD parameters are 60% of the measurements; I value is 1/200 of a P value. Finally, the speed loop is integrated with vertical and steering loops for vehicle test. PID parameter-tuning is completed.

6 Conclusions

The work studied the two-wheel self-balancing vehicle system with the visual recognition of vehicle attitude, including self-balancing vehicle balance principle, hardware control circuit, Kalman filtering, PID control, vehicle CCD vision sensor acquisition technique, edge shape detection and analysis in machine vision technology. The main conclusions are described as follows.

(1) Firstly, the overall architecture design of system was given to then introduce the principles and peripheral circuit

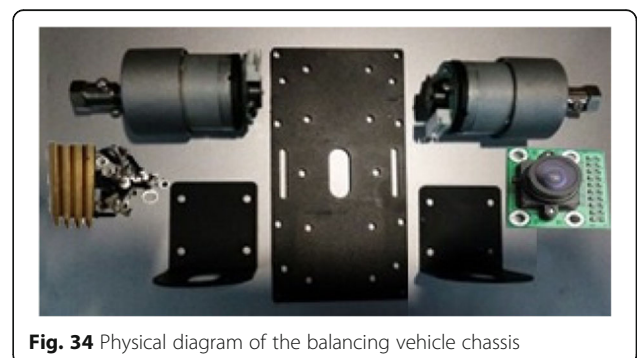


Fig. 34 Physical diagram of the balancing vehicle chassis

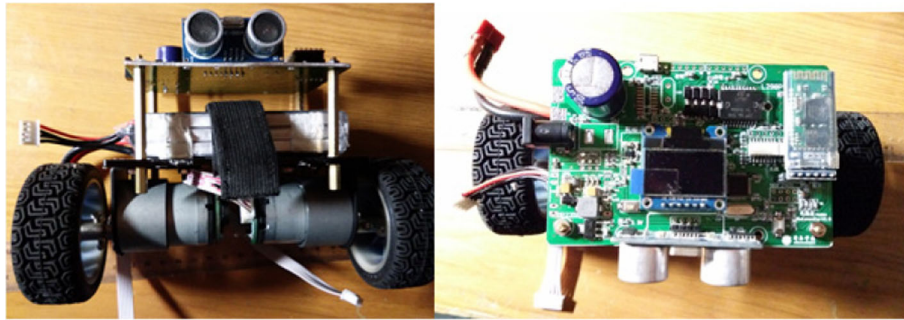


Fig. 35 The overall assembly diagram



Fig. 36 The output waveform of gyro sensors



Fig. 37 The interrupt waveform of gyro sensors INT

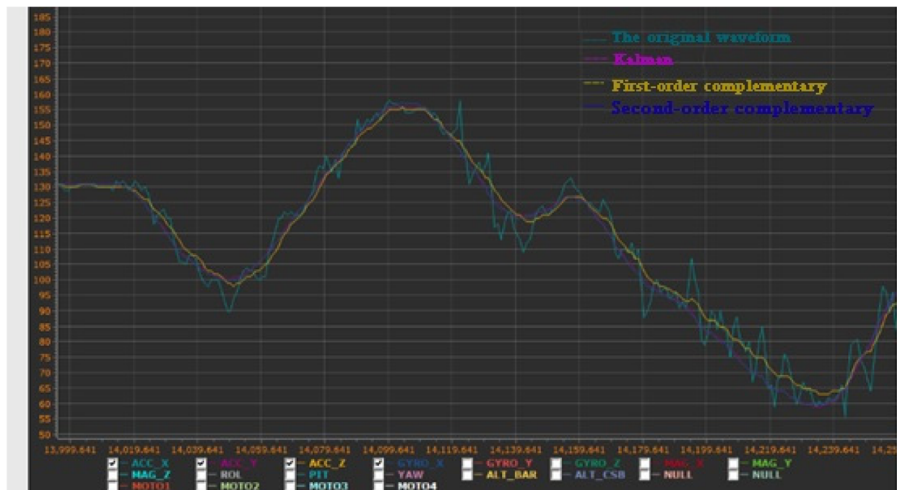


Fig. 38 First-order low-pass filtering waveform

design of a minimum system, sensor, camera, and liquid crystal in detail.

(2) After giving a general software algorithm framework of the system, the work explored the algorithms such as Kalman filtering and PID control according to the algorithm module.

(3) The traditional PID algorithm was improved.

Based on a differential saturation phenomenon in incremental PID algorithm, the unexecuted differential part was re-executed step by step. The system was easily interfered with high-frequency noise due to introduction of differential terms. Therefore, first-order low-pass filtering was often added to incomplete differentiation. The filtering algorithm was

added to the feedback term. First-order low-pass filtering was also added after the PID calculation term. This project introduced the serial PID control algorithm into self-balancing vehicle control algorithm. The serial PID control algorithm included speed loop position PI and angle loop position PD algorithms.

① Speed loop position PI algorithm

In the speed control, the PI control algorithm was adopted to strictly integrate the displacement. It was ensured that the balance vehicle did not drift left or right. The speed feedback was positive. After setting the speed, the motor broke the vertical balance, thus causing forward or backward motion of the vehicle.

② Angle loop position PD algorithm



Fig. 39 Waveform comparison result

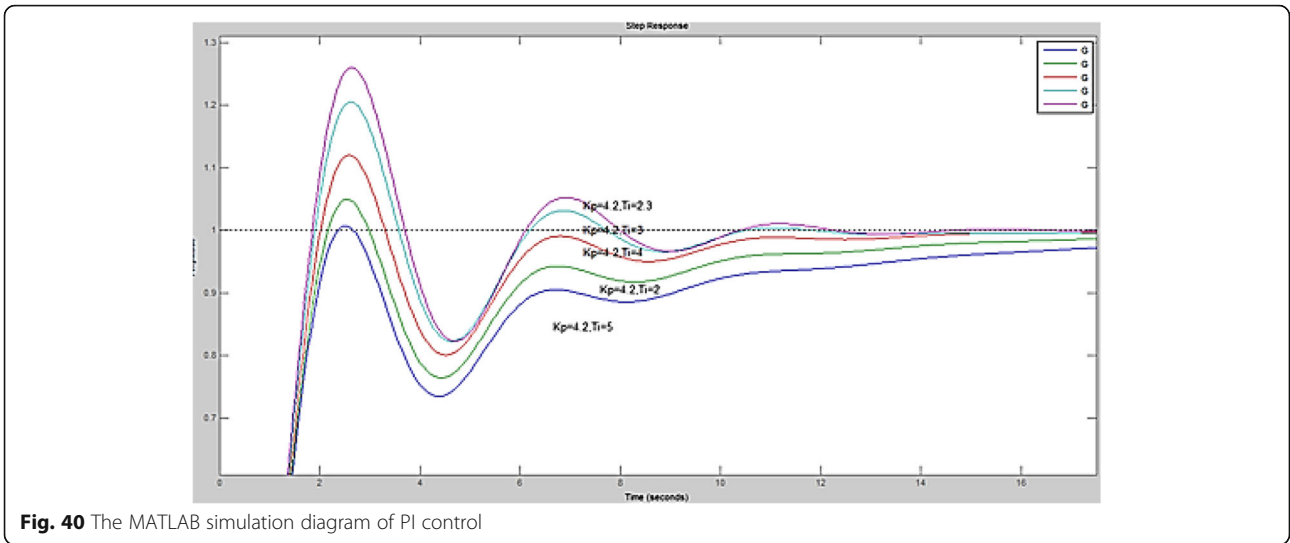


Fig. 40 The MATLAB simulation diagram of PI control

In vertical control, the PD control algorithm had a fast response speed and short adjustment time. It was more suitable for balance vehicle system with large inertia. The vertical feedback took the angle and angular velocity of a gyroscope as references based on the principle of negative feedback. When the vehicle deviated from the set angle, it tended to return to the origin.

③ Cascade PID control algorithm of balance vehicle

PID control was used for a balance vehicle core, including vertical and speed loop controls, wherein the vertical loop control always dominated. As the auxiliary, speed loop control changed the speed to control the movement of the balance vehicle. Based on the idea of advanced PID, the two worked synchronously to form a cascade PID controller. Positive feedback interference of the speed loop was

added to the balance vehicle based on the control of the vertical loop.

(4) The balance vehicle system was equipped with an early warning system for visual recognition of vehicle attitude to improve safety and market acceptability. Road information was extracted by digital image processing methods such as image denoising and binarization. Therefore, we realized an early warning system based on swarm intelligence.

(5) The software and hardware modules of a system were gradually debugged and tested. Results showed that the self-balancing vehicle based on visual autonomous navigation worked well.

However, the system processor speed needs to be improved due to the complexity of road information. ARM +DSP can be used for subsequent development, which will improve the image and algorithm processing speed.

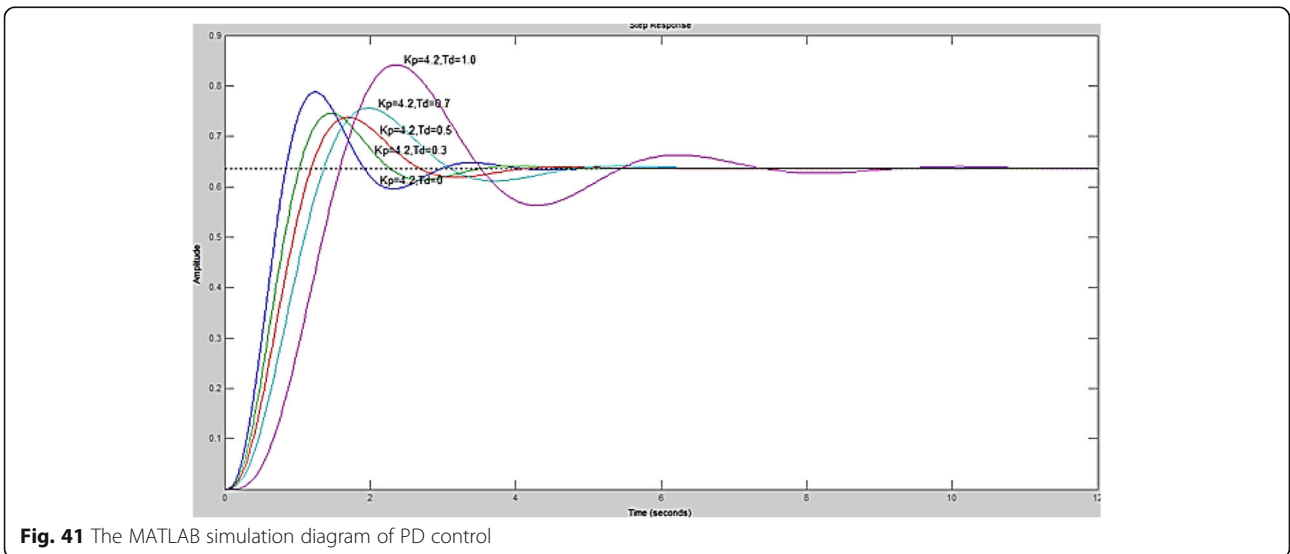


Fig. 41 The MATLAB simulation diagram of PD control

Abbreviations

ARM: Advanced RISC Machine; CCD: Charge-coupled device; DSP: Digital signal processor; PID: Process identifier; PWM: Pulse Width Modulation; USB: Universal Serial Bus

Acknowledgements

Thank for the editor and reviewers.

Funding

The paper is subsidized by science and technology key project of Henan Province, China. NO.172102210462.

Availability of data and materials

Data will not be shared; reason for not sharing the data and materials is that the work submitted for review is not completed. The research is still ongoing, and those data and materials are still required by my team for further investigations.

Authors' contributions

GW and BZ designed the research. BZ analyzed the data, wrote, and edited the manuscript. Both authors read and approved the final manuscript.

Ethics approval and consent to participate

We approved.

Consent for publication

We agree.

Competing interests

The authors declare that they have no competing interests.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Received: 30 October 2018 Accepted: 24 January 2019

Published online: 06 February 2019

References

1. J. Lin, D. Tingfei, X. Hui, Current loop cascade PID control design of self-balanced vehicle. *Microcontr. Embed. Syst.* **5**, 63–67 (2017)
2. Q.I.N. Fu-Kang, M.E.N.G. Fan-Xi, The control system design of two-wheeled self-balancing car based on linear CCD. *Dev. Innov. Mach. Electr. Prod.* **29**(4), 38–40 (2016)
3. W.U. Junfeng, S.I.N. Lei, Two-wheeled self-balancing robot control method study. *J. Harbin Eng. Univ.* **19**(6), 22–26 (2014)
4. W.A.N.G. Rongben, Y.U. Tianhong, G.U. Boyuan, et al., Research on linear lane mark identification and track method based on Edg. *Comput. Eng.* **32**(18), 195–196 (2006)
5. Y. Xining, D. Jianmin, G. Dezhi, Z. Banggui, et al., Research on lane detection based on improved Hough transform. *Comp. Meas. Contr.* **18**(2), 292–294 (2010)
6. X.U. Luping, *Digital image processing* (Science press, Beijing, 2007), pp. 125–153
7. P. Berkelman, M. Chen, J. Easudes, J. Hancock, M. Martin, A. Mor, E. Rollins, A. Sharf, J. Silberman, T. Warren, D. Babna. Design of a day/night lunar rover. *CMU-RI-TR-95-24*. 1995:1–126
8. J.C. McCall, M.M. Trivedi, Video-based lane estimation and tracking for driver assistance: survey, system, and evaluation. *IEEE Trans. Intell. Transp. Syst.* **7**(1), 20–37 (2006)
9. D. Demigny, T. Kamle, A discrete expression of Canny's criteria for step edge detector performances evaluation. *IEEE Trans. Pattern Anal. Mach. Intell.* **19**(11), 1199–1211 (1997)
10. Z. Bo-ping, H. Jian, The study of GPS vision navigation system of multi-stage real-time matching algorithm. *Sci. Technol. Eng.* **14**(13), 208–212 (2014)
11. X. Youchun, W. Rongben, L. Bing, J. Shouwen, "A vision navigation algorithm based on linear lane model", *Proceedings of the IEEE Intelligent Vehicles Symposium 2000* (Cat. No.00TH8511) (Dearborn, 2000), pp. 240–245. <https://doi.org/10.1109/IVS.2000.898348>
12. R. Keyhani, M. Deriche, E. Palmer, "A high impedance fault detector using a neural network and subband decomposition", *Proceedings of the Sixth International Symposium on Signal Processing and its Applications* (Cat.No. 01EX467) Vol. 2 (Kuala Lumpur, 2001), pp. 458–461. <https://doi.org/10.1109/ISSPA.2001.950179>
13. J. Kim, P.J. Bentley, A model of gene library evolution in the dynamic clonal selection algorithm. J. Timmis, P.J. Bentley (Eds.), *Proceedings of the 1st International Conference on Artificial Immune Systems (ICARIS'02)*, Canterbury, UK, 9–11 September 2002, (University of Kent at Canterbury Printing Unit, 2002), pp. 175–182
14. J.Kim, P.J. Bentley, U. Aickelin, J. Greensmith, G. Tedesco, J. Twycross, Immune system approaches to intrusion detection—a review[J]. *Natural Computing.* **6**(4), 413–466 (2007). <https://doi.org/10.1007/s11047-006-9026-4>
15. Z. Bo-ping, W. Guo-xi, Research on decision support system for allocation of education source based on CO-agent. *2012 Int. Conf. Technol. Manag. ICTAM 2012*(6), 727–734 (2012)
16. W.U. Guoxi, Z. Boping, Research on decision support system for civil water pollution control based on immune agent. *Comput. Eng.* **35**(10), 179–181 (2009)
17. Q. Yongyuan, Z. Hongyue, W. Shuhua, *Kalman filtering and integrated navigation principle* (Northwestern Polytechnical University Press, Xi An, 1998), pp. 33–55
18. S. Mingxuan, H. Baojian, *Iterative learning control* (National Defense Industry Press 5, Beijing, 1999), pp. 80–121
19. B. Zhigang, *Automatic adjustment system analysis and PID tuning* (Chemical Industry Press 5, Beijing, 2015), pp. 78–155
20. W. Lei, S. Wenzhong, PID controlling. *Autom. Instrument* **25**(04), 3–8 (2004)
21. L. Fing, Research on motor speed control system based on advanced PID control. *Technol. Innov. Appl.* **15**, 34–37 (2014)
22. M. Viteckova, A. Vitecek, 2DOF PI and PID controllers tuning. *IFAC Proc. Vol.* **43**(2), 343–348 (2010)
23. Z. Jinlong, X. Hui, et al., PID control based on fuzzy neural network for precision angular alignment. *Chin. J. Sci. Instrum.* **33**(03), 549–554 (2012)
24. H.-m. Li, X.-b. Wang, S.-b. Song, H. Li, Vehicle control strategies analysis based on PID and fuzzy logic control [J]. *Procedia Eng.* **137**, 234–243 (2016)
25. B. Zhang, Distributed SVM face recognition based on Hadoop. *Clust. Comput.* **11**, 1–8 (2017)
26. Y. Baijun, W. Xuechun, H. Yaqin, *Easy to play STM32 microcontroller* (Electronics Industry Press 5, Beijing, 1999), pp. 55–132
27. Š. Bucz, A. Kozáková, The sine wave tuning method: Robust PID controller design in the frequency domain. *J. Electric. Syst. Inform. Technol.* **2**(3), 358–367 (2015)
28. Q. Fu-Kang, L. Xin, D. Yao-Wei, P. Bo, L. Jun, L. Bo, The analysis of flowering prediction device based on the theories of TRIZ. *Dev. Innov. Mach. Electric. Prod.* **29**(3), 43–45 (2016)
29. Q. Zhou, Z. Xu, N.Y. Yen, User sentiment analysis based on social network information and its application in consumer reconstruction intention. *Comput. Hum. Behav.* (2018). <https://doi.org/10.1016/j.chb.2018.07.006>
30. Q. Zhou, Multi-layer affective computing model based on emotional psychology. *Electron. Commer. Res.* **18**(1), 109–124 (2018). <https://doi.org/10.1007/s10660-017-9265-8>
31. Q. Zhou, J. Luo, The study on evaluation method of urban network security in the big data era. *Intell. Autom. Soft Comput.* **24**(1), 133–138 (2018). <https://doi.org/10.1080/10798587.2016.1267444>
32. ZHOU Mu-dan KANG Kai CAI Pu-lang HUANG Tian-jian. Design and implementation of two-wheeled self-balancing vehicle control system. *Tech. Autom. Appl.*, 2014, 33 (10) :4–8

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► [springeropen.com](https://www.springeropen.com)