DOI: 10.54254/2755-2721/2026.BJ31650

Integrated Research on Design and Implementation of an Embedded Quadcopter UAV System Based on STM32

Yige Luo

School of Transportation Engineering, Central South University, Changsha, China
1830961927@qq.com

Abstract. With the rapid development of embedded technology, Micro-Electro-Mechanical Systems (MEMS), and intelligent control algorithms, quadcopter unmanned aerial vehicles (UAVs) have demonstrated broad application prospects in fields such as aerial photography, agricultural plant protection, environmental monitoring, logistics distribution, and teaching and scientific research due to their simple structure, flexible control, and strong vertical take-off and landing capabilities. In response to current issues in quadcopter UAV control systems—such as high cost, complex algorithms, and weak anti-interference capabilities microcontrollers. The system employs the STM32F103C8T6/STM32F4 series as the core controller, utilizes the MPU6050 attitude sensor to acquire flight data, integrates quaternion-based attitude calculation and Kalman filtering algorithms to achieve high-precision attitude estimation, adopts a cascade PID control strategy to stabilize flight attitude, and enables real-time communication with the remote controller via NRF24L01/Si24R1 wireless modules. In terms of software architecture, a layered design philosophy is adopted, and the FreeRTOS real-time operating system is integrated for multi-task scheduling, thereby enhancing system real-time performance and reliability. Through hardware construction, software programming, system integration, and flight testing, the designed UAV system is verified to possess good stability, fast response, and strong anti-interference capabilities. This study not only provides a complete technical solution for developing low-cost, high-performance quadcopter UAVs but also establishes theoretical and practical foundations for subsequent algorithm optimization and functional expansion.

Keywords: amplitude modulation, Synchronous demodulation, Spectrum aliasing, Butterworth filter, Signal distortion

1. Introduction

The drone has two pairs of symmetric rotors and can hover and move freely, with flexible and stable flight [1]. As an important unmanned aerial vehicle, drones are playing an increasingly important role in military reconnaissance, civilian aerial photography, agricultural spraying, disaster relief and other fields [2] . Among them, quadrotor drones have become one of the hotspots in current drone research due to their advantages such as simple mechanical structure, good maneuverability, vertical takeoff and landing, and hovering. However, quadrotor drones belong to underactuated and strongly coupled nonlinear systems, and their flight control is vulnerable to external environmental

DOI: 10.54254/2755-2721/2026.BJ31650

interference, posing high requirements for the robustness and real-time performance of control algorithms [3].

In recent years, with the continuous improvement of the performance of ARM Cortex-M series microcontrollers and the decline in the cost of MEMS sensors, the design of drone flight control systems based on embedded platforms has gradually become the mainstream [4]. STM32 series microcontrollers are widely used in various drone projects due to their high performance, low power consumption, rich peripheral resources and complete ecosystem. At the same time, attitude solution algorithms such as quaternion method, Kalman filter and classical PID control algorithms have also been deeply studied and applied in drone control [5].

Based on a comprehensive analysis of multiple relevant research papers, this paper aims to design and implement an embedded quadrotor drone system based on STM32. The system covers the entire process of hardware selection, circuit design, software architecture, attitude fusion algorithm, control strategy and system testing, and strives to achieve stable flight and precise control of the drone on the premise of controllable cost, providing a complete reference solution for teaching practice, scientific research exploration and industrial applications.

2. System architecture

The micro quad-rotor drone consists of an STM32F103C8T6 main control system, an NRF24L01 wireless communication module, an MPU6050 gyroscope module, motors and a battery. The micro quad-rotor drone adopts an X-shaped layout design, which has the advantages of flexible control, fast motion acceleration and high robustness. The remote controller establishes communication with the quad-rotor drone through a wireless communication mode in the 2.4GHz wireless frequency band to realize the remote operation of the drone's motion, so as to facilitate the control of the drone's motion trajectory. The real-time state of the drone is obtained jointly by the MPU6050 gyroscope module and the BMP280 barometer, and the attitude data of the drone is obtained through algorithm analysis. The main control calculates the output control quantity and converts it into a PWM wave to output to the motors for operation. In the attitude control part, the micro quad-rotor drone uses a cascade PID to control its attitude. Thus, the stable flight and control of the drone are completed. At the same time, MATLAB/Simulink simulation modeling is used to verify the efficiency of the Kalman filter. The overall system design block diagram of the quad-rotor drone is shown in Figure 1.

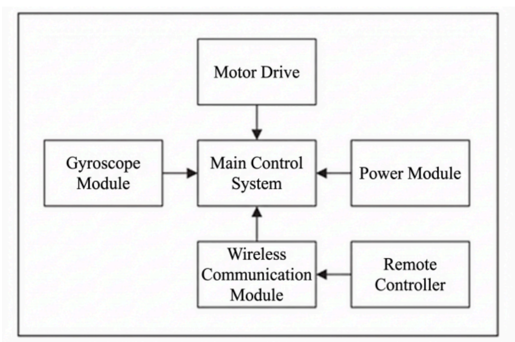

Figure 1. Overall design diagram of the drone

DOI: 10.54254/2755-2721/2026.BJ31650

3. Hardware design of the drone

3.1. STM32 main control system

This design uses an STM32F103RCT6 single-chip microcomputer as the main control board. The STM32 has rich resources. It can not only complete various functions by configuring peripherals through programs, but also read data by using STM32 and control the motor speed according to the control algorithm to ensure the stable flight of the aircraft. The design of the STM32 main control circuit mainly includes a microcontroller circuit, a clock circuit, a power supply circuit and a reset circuit, as shown in Figure 2 specifically.

First, the microcontroller, that is, the MCU, is the core of the minimum system circuit in the single-chip microcomputer, and its main task is to be responsible for data processing and program operation; second, the working speed of the single-chip microcomputer is represented by the signal pulse provided by the crystal oscillator, and the work of the crystal oscillator and capacitor in the clock circuit is to provide the clock signal required for the operation of the microcontroller; third, the power supply circuit is to provide a stable power supply for the whole circuit so that the circuit can work normally; fourth, when the single-chip microcomputer system needs to restart because the program cannot run normally during operation, the reset circuit can ensure that the microcontroller starts correctly when powered on; fifth, the program download port is the channel for programmers to burn the corresponding code into the controller. For the convenience of development and debugging, the program port usually uses common interfaces such as JTAG, SWD, ISP, etc.

First, complete the construction of the software execution environment of the system according to the processing flow of USB driver connection, STM32 protocol setting, drone electrical communication configuration analysis, and monitoring task priority setting, and then combine relevant hardware device components to realize the smooth application of the STM32-based drone electrical equipment monitoring system.

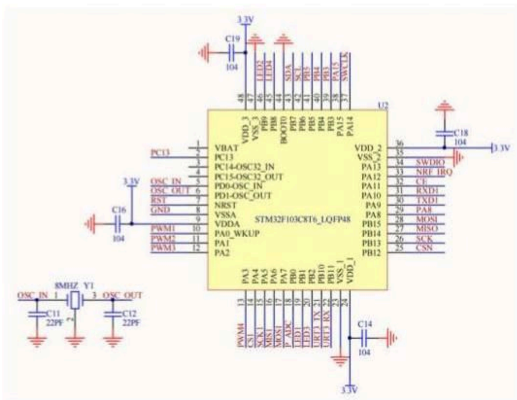

Figure 2. STM32 hardware diagram

3.2. Design of the attitude sensor circuit

The attitude sensor enables the drone to fly in the air with a relatively stable attitude. To ensure that the micro-drone can fly stably and maintain at the expected altitude, it is necessary to cooperate with the altitude controller to test the altitude of the drone. At the same time, the attitude sensor needs to collect the operation data of the drone in the air, first transmit it to the STM32F103C8T6 main controller through the IIC bus, and then the main controller will filter and fuse the obtained data,

DOI: 10.54254/2755-2721/2026.BJ31650

calculate the corresponding attitude angle through formula calculation, and finally output the data to the upper computer through serial communication.

The circuit diagram of the MPU6050 attitude sensor is shown in Figure 3. INT is the interrupt interface of the MPU6050.

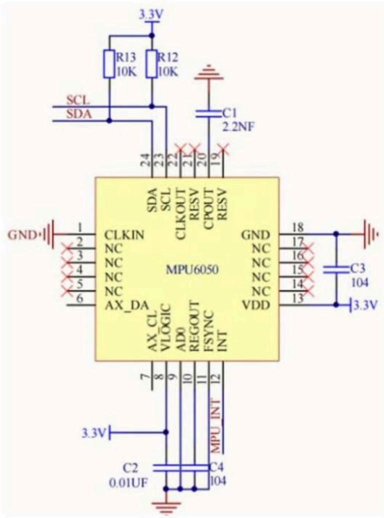

Figure 3. MPU6050 hardware diagram

3.3. Wireless communication module selection and circuit

The drone wireless communication module selects NRF24L01 as the communication system. NRF24L01 communicates with the main control STM32 through SPI. NRF24L01 has the advantage of strong anti-interference ability, which can ensure that the communication will not be interrupted at a certain distance and ensure that the drone will not be out of control due to its influence. During the communication process, the transmitter of the remote control encodes and modulates the data to be transmitted and sends it out in the form of a wireless signal through the NRF24L01 module. After receiving the signal, the NRF24L01 module at the receiving end of the drone demodulates and decodes the signal and provides the data to the control system at the receiving end for processing and response.

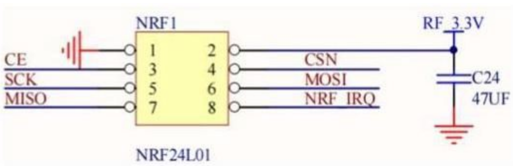

Figure 4. NRF24L01 hardware diagram

3.4. Motor system

The four-axis uses a micro high-speed 720 hollow cup brushed motor, which can reach 45000 RPM under full power operation. To provide enough current to drive the motor, an NMOS design is selected for the motor drive circuit. The main chip timer generates PWM pulses to control the on and off of the NMOS, thereby achieving the control of the brushed motor. Because in the PID control of the drone, the motor needs to be constantly in the startup and decay modes. When the motor is in the decay mode, that is, when the motor moves from the running state to the stop state,

DOI: 10.54254/2755-2721/2026.BJ31650

since the motor is an inductive load, the current cannot change suddenly. When the voltage applied to both ends of the motor is disconnected, the back electromotive force generated by the motor is very likely to damage the NMOS tube. Therefore, when the motor needs to stop, in addition to disconnecting, a freewheeling circuit needs to be added, that is, a freewheeling diode is added beside the NMOS tube. As shown in Figure 5.

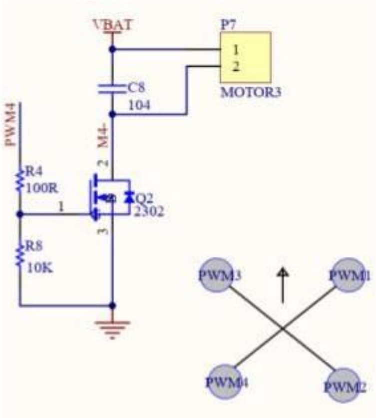

Figure 5. Motor hardware diagram

4. Software design of micro UAVs

4.1. Conversion relationship between euler angles and quaternions

The micro quad-rotor UAV uses the MPU6050 gyroscope sensor to obtain the attitude information of the UAV. However, the obtained angular velocity and angular acceleration information cannot be directly used. In this paper, a method of converting quaternions to Euler angles is adopted for the design.

Quaternions are numbers with a q = q0 + iq1 + jq2 + kq3 complex form, where q0, q1, q2 is a real number and i, j, k are imaginary units.

Euler angles are a representation method for describing the attitude of objects such as quadrotor drones. It can be divided into three angles: the φ roll angle(Roll) rotating around the X-axis, the θ pitch angle(Pitch) rotating around the Y-axis, and the ψ yaw angle(Yaw) rotating around the Z-axis. The Euler angles are shown in Figure 6.

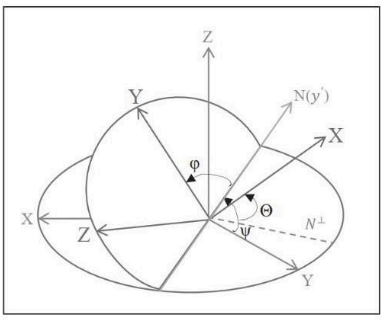

Figure 6. Illustration of attitude angles

The information obtained from the MPU6050 sensor of a micro quadrotor drone is used to calculate the corresponding quaternion representation through the following formula.

DOI: 10.54254/2755-2721/2026.BJ31650

$$q_0=\sin\frac{\phi}{2}\sin\frac{\theta}{2}\sin\frac{\psi}{2}+\cos\frac{\phi}{2}\cos\frac{\theta}{2}\cos\frac{\psi}{2} \tag{1}$$

$$q_1=\cos\frac{\phi}{2}\sin\frac{\theta}{2}\sin\frac{\psi}{2}-\sin\frac{\phi}{2}\cos\frac{\theta}{2}\cos\frac{\psi}{2} \tag{2}$$

$$q_2=\sin\frac{\phi}{2}\cos\frac{\theta}{2}\sin\frac{\psi}{2}+\cos\frac{\phi}{2}\sin\frac{\theta}{2}\cos\frac{\psi}{2} \tag{3}$$

Obtain the corresponding quaternion information and convert it to the corresponding Euler angles by using the formula.

$$\begin{bmatrix}\theta\\ \varphi\\ \psi\end{bmatrix}=\begin{bmatrix}\arcsin\left(2\left(q_0q_2+q_1q_3\right)\right)\\ \arctan\frac{2(q_0q_1+q_2q_3)}{1-2(q_1^2+q_2^2)}\\ \arctan\frac{2(q_0q_3+q_1q_2)}{1-2(q_3^2+q_4^2)}\end{bmatrix} \tag{4}$$

Get the gravity component in the attitude matrix calculated by the gyroscope (since the accelerometer measures values in the object coordinate system, we also need to extract the gravity component in the object coordinate system in the attitude matrix calculated using the angular velocity). The gravity components are denoted as Vx, Vy, and Vz, and the specific calculation method is as follows:

The attitude matrix calculated by quaternion for the transformation from the E frame (geographic coordinate system) to the b frame (body coordinate system) is:

$$c_E^b=\begin{bmatrix}1-2\left(q2^2+q3^2\right) & 2\left(q1q2+q0q3\right) & 2\left(q1q3-q0q2\right)\\ 2\left(q1q2-q0q3\right) & 1-2\left(q1^2+q3^2\right) & 2\left(q2q3+q0q1\right)\\ 2\left(q1q3+q0q2\right) & 2\left(q2q3-q0q1\right) & 1-2\left(q1^2+q2^2\right)\end{bmatrix} \tag{5}$$

Then the gravity component in the b system (object coordinate system) is extracted as

$$\begin{bmatrix}Vx\\ Vy\\ Vz\end{bmatrix}=\begin{bmatrix}1-2\left(q2^2+q3^2\right) & 2\left(q1q2+q0q3\right) & 2\left(q1q3-q0q2\right)\\ 2\left(q1q2-q0q3\right) & 1-2\left(q1^2+q3^2\right) & 2\left(q2q3+q0q1\right)\\ 2\left(q1q3+q0q2\right) & 2\left(q2q3-q0q1\right) & 1-2\left(q1^2+q2^2\right)\end{bmatrix}\begin{bmatrix}0\\ 0\\ 1\end{bmatrix}=\begin{bmatrix}2\left(q1q3-q0q2\right)\\ 2\left(q2q3+q0q1\right)\\ 1-2\left(q1^2+q2^2\right)\end{bmatrix} \tag{6}$$

Complementary filtering adds the error input to the Pid controller and the angular velocity measured by the gyroscope in the current attitude update to obtain a corrected angular velocity value. The obtained corrected angular velocity value is used to update the quaternion, thereby obtaining accurate attitude angle information.

Let gx,gy,gz be the angular velocities of the three axes measured by the gyroscope and the corrected angular velocity values after filtering, and Kp be the complementary filtering coefficient. Then the calculation method for the corrected angular velocity is as follows:

$$gx=gx+Kp*ex+accex \tag{7}$$

DOI: 10.54254/2755-2721/2026.BJ31650

$$gy = gy + Kp * ey + accey \tag{8}$$

$$gz = gz + Kp * ez + accez \tag{9}$$

Finally, we can solve for the required attitude angles accordingly.

$$g1 = 2(q1q3 - q0q2) \tag{10}$$

$$g2 = 2(q2q3 + q0q1) \tag{11}$$

$$g3 = q0^2 - q1^2 - q2^2 + q3^2 \tag{12}$$

$$g4 = 2(q1q2 + q0q3) \tag{13}$$

$$g5 = q0^2 + q1^2 - q2^2 - q3^2 \tag{14}$$

then

$$\gamma = -\arcsin(g1) \tag{15}$$

$$\theta = \arctan(g2/g3) \tag{16}$$

$$\psi = \arctan(g4/g5) \tag{17}$$

The program statements are as follows:

$$\text{float } g1,\ g2,\ g3 \tag{18}$$

$$g1 = 2.0f * (q1 * q3 - q0 * q2) \tag{19}$$

$$g2 = 2.0f * (q0 * q1 + q2 * q3) \tag{20}$$

$$g3 = q0 * q0 - q1 * q1 - q2 * q2 + q3 * q3 \tag{21}$$

$$g4 = 2.0f^{\,*}\left(q1q2 + q0q3\right) \tag{22}$$

$$g5 = q0 * q0 + q1 * q1 - q2 * q2 - q3 * q3 \tag{23}$$

$$\text{pitch} = -\text{asinf}(g1) * 57.3f \tag{24}$$

DOI: 10.54254/2755-2721/2026.BJ31650

$$roll = atanf(g2/g3)\,{*}57.3f \tag{25}$$

$$yaw = atan2f(g4/g5)\,{*}57.3f \tag{26}$$

4.2. Kalman filter algorithm and simulation

The micro quad-rotor UAV uses the Kalman filter algorithm to filter the data of the MPU6050. During the process of obtaining the attitude by the MPU6050, its three-axis accelerometer and three-axis gyroscope are often disturbed by factors such as noise, resulting in inaccurate and large-error data obtained. The method of using the Kalman filter algorithm can effectively reduce the error.

The Kalman filter algorithm is a highly efficient recursive filter that provides optimal state estimation by continuously iterating and updating the state estimation values and covariance matrices [6]. The calculation process of the Kalman filter is as follows: 5. Complementary filtering. The error is input into the Pid controller and added to the angular velocity measured by the gyroscope in the current attitude update to obtain a corrected angular velocity value. The obtained corrected angular velocity value is used to update the quaternion, thereby obtaining accurate attitude angle information. Let gx,gy,gz be the angular velocities of the three axes measured by the gyroscope and the filtered angular velocity correction value, and Kp be the complementary filtering coefficient. Then the calculation method of the corrected angular velocity is as follows:

The first step is the calculation of the prior estimate value, as shown in Equation (27):

$$\hat{x}_{\bar{k}} = A\hat{x}_{k-1} + Bu_{k-1} \tag{27}$$

The second step is the calculation of the prior error covariance matrix, as shown in Equation (28):

$$P_k^- = AP_{k-1}A^T + Q \tag{28}$$

The third step is the calculation of the Kalman gain, as shown in Equation (29):

$$K_k = \frac{P_k^- H^T}{HP_k^- H^T + R} \tag{29}$$

The fourth step is the calculation of the posterior estimate value (optimal value), as shown in Equation (30):

$$\hat{x}_k = \hat{x}_{\bar{k}} + K_k\left(z_k - Hx_{\bar{k}}\right) \tag{30}$$

The fifth step is to update the error covariance matrix, as shown in Equation (31):

$$P_k = (I - K_k H)P_k^- \tag{31}$$

Then, the output of the state variable and the velocities (Omega 1 to 4) given by the attitude control part are used to estimate the state of the aircraft through the Kalman filter. The Simulink

DOI: 10.54254/2755-2721/2026.BJ31650

simulation of the Kalman filter is shown in Figure 7.

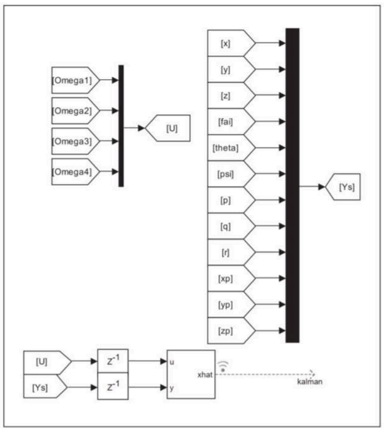

Figure 7. Simulink simulation diagram

MATLAB simulation is used to observe the efficiency of the Kalman filter. It can be seen from the simulation results that although the input command in the Z-axis direction changes continuously, the specified 0 values in the X-axis and Y-axis directions remain horizontal and unchanged. As shown in Figure 5, with the continuous change of the input value, the fitting degree of the Kalman estimation curve and the actual output curve is relatively high. This indicates that the attitude of the micro quadrotor UAV is well solved by the Kalman filter.

C language partial code:

```
// P.K_1_K_1 = P.K_K;
// return X.K_K;
//}
void kalman_1(struct _1_ekf_filter *ekf,float input)
{
   ekf->Now_P = ekf->LastP + ekf->Q;
   ekf->Kg = ekf->Now_P / (ekf->Now_P + ekf->R);
   ekf->out = ekf->out + ekf->Kg * (input - ekf->out);
   ekf->LastP = (1-ekf->Kg) * ekf->Now_P ;
}
```

4.3. Cascade PID algorithm

The micro UAV uses the PID control algorithm theory, and the PID proportional adjustment is used to gradually reduce the error. In this design, if the outer loop uses the angle loop for PI control and the inner loop uses the angular velocity loop for D control, the aircraft can correct the speed quickly when there is a large angle deviation and slowly when there is a small angle deviation. The principle of its PID control algorithm is shown in Figure 8.

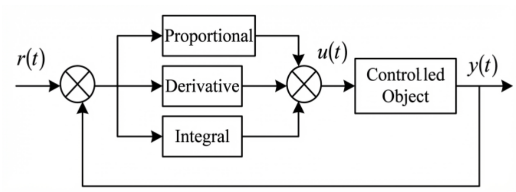

Figure 8. PID principle diagram

DOI: 10.54254/2755-2721/2026.BJ31650

The PID calculation formula is as shown in Equation (32):

$$u(t) = K_p e(k) + K_p \frac{T}{T_i} \sum_{j=1}^{k} e_j + K_p \frac{T_d}{T} [e(k) - e(k-1)] \quad (32)$$

In the formula, $u(t)$ represents the control quantity output by the controller; K_p represents the proportional coefficient; T_i represents the integral time constant; T_d represents the derivative time constant; $\sum_{j=1}^{k} e_j$ represents the cumulative sum of errors from the first sampling moment to the k -th sampling moment; $e(k) - e(k-1)$ represents the difference between the error at the current sampling moment and the error at the previous sampling moment, approximately representing the rate of change of the error.

The theory of the PID control algorithm is an improvement of intelligent algorithms and the rise of control algorithm adaptive parameter adjustment technology, making great progress in the research direction of performance in nonlinear control. In the PID parameter adjustment, the values of K_P , K_I and K_D can change with the error, and the parameters can be calculated and adjusted at any time. Therefore, there is no need to accurately mathematically model the controlled object, simplifying the relevant processes.

If a single-loop PID is used to control the attitude angle integrated from the angular velocity, the integration will cause integration drift due to long-term integration. If the attitude angle obtained from the gravity sensor is used, it is easily affected by the vibration of the body. If the attitude angle obtained by calculating the six axes using quaternions is used, the control speed will be relatively slow (the quaternion operation continuously corrects the quaternion through the acceleration, which is a process). Therefore, a cascade PID is introduced, as shown in Figure 9.

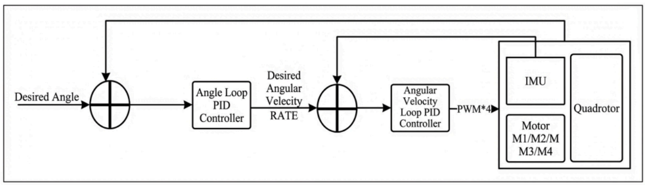

Figure 9. Cascade PID structure diagram

Desired Angle: The flight rudder amount given by the remote control. In the ideal state, the horizontal flight is 0 rudder amount.

Angle to PID: Perform angle-loop PID control on the attitude angle, and its output is used as the expected value for the inner-loop PID.

Feedback Current Angle: The angle obtained from the quaternion operation. Unit: degree.

Angular Velocity PID Control: Perform PID control on the three-axis angular velocity of the six-axis sensor, and its output is used as the PWM control signal for the motor.

Feedback/Current Angular Velocity: The measurement data of the three-axis angular velocity of the six-axis sensor, unit degree/second.

Motor: Controlled by PWM, the signal is 0~100%.

DOI: 10.54254/2755-2721/2026.BJ31650

PWM Partial Program Code:

```
void PID_Init(PID *pid,float p,float i,float d,float maxl,float maxOut)
{
   pid->kp=p;
   pid->ki=i;
   pid->kd=d;
   pid->maxIntegral=maxl;
   pid->maxOutput=maxOut;
}
void PID_Clear(PID *pid)
{
   pid->error=0;
   pid->lastError=0;
   pid->integral=0;
   pid->output=0;
}
typedef struct _CascadePID
{
   PID inner;
   PID outer;
   float output;
}CascadePID;
void PID_CascadeCalc(CascadePID *pid,float angleRef,float angleFdb,float speedFdb)
{
   PID_SingleCalc(&pid->outer,angleRef,angleFdb);
   PID_SingleCalc(&pid->inner,pid->outer.output,speedFdb);
   pid->output=pid->inner.output;
}
// P.K_1_K_1 = P.K_K;
// return X.K_K;
//}
void kalman_1(struct _1_ekf_filter *ekf,float input)
{
   ekf->Now_P = ekf->LastP + ekf->Q;
   ekf->Kg = ekf->Now_P / (ekf->Now_P + ekf->R);
   ekf->out = ekf->out + ekf->Kg * (input - ekf->out);
   ekf->LastP = (1-ekf->Kg) * ekf->Now_P ;
}
```

5. Hardware physical diagram

5.1. Host computer verification waveform

To verify the state test of the designed unmanned aerial vehicle, the flight control core is inserted on the debugging platform for the host computer flight attitude test. The test diagram of the unmanned aerial vehicle is shown in Figure 10.

DOI: 10.54254/2755-2721/2026.BJ31650

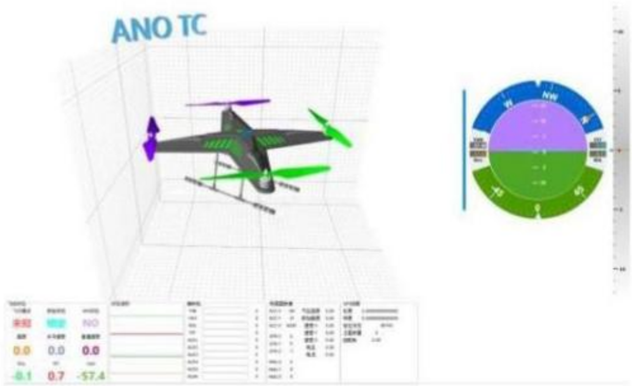

Figure 10. Host computer test diagram

At the same time, the waveform can be selected to view the filtering effect.

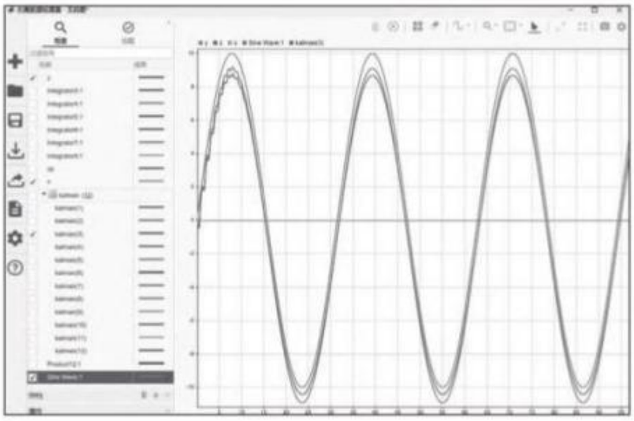

Figure 11. Filtered waveform diagram

5.2. Communication test

The micro quad-rotor UAV conducts a test on the accuracy of data transmission and reception in an open area without obstacles. By sending a specific set of test data from the ground terminal, the received data at the UAV receiver is compared with the sent data to calculate the accuracy of data transmission and reception. The transmission and reception test is shown in Figure 12.

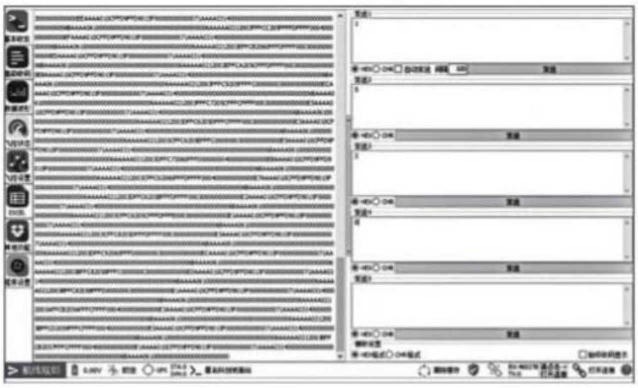

Figure 12. Communication transmission and reception diagram

DOI: 10.54254/2755-2721/2026.BJ31650

6. Conclusion

Through optimizing the hardware design and system integration, this quad-rotor aircraft system shows significant advantages in terms of stability, endurance, and communication reliability. The combination of 8520 motors and 13.5 cm propellers achieves efficient power output at a 16 cm wheelbase; with the 600 mAH battery, it can achieve continuous flight for about 10 minutes. The system is equipped with a calibrated MPU6050 sensor, effectively suppressing zero drift and improving flight attitude stability; the charging module has a safety protection function and can complete charging in about 54 minutes. The communication module has passed multiple environment tests, with a communication success rate of over 99% within 30 meters indoors and still maintaining a stable connection within 30 meters in an outdoor open environment, demonstrating good environmental adaptability. In summary, this system meets the expected design goals in terms of structural compactness, endurance time, flight stability, and control reliability, and has good experimental and application potential.

References

[1] Song Fengying, You Xiaoping. Design and Implementation of a Micro Quad-Rotor UAV Based on STM32 [J]. Journal of University of Science and Technology Liaoning, 2017, 40(02): 148-154.
[2] Li Qing. Design of a Quad-Rotor Aircraft System Based on STM32 [J]. Modern Manufacturing Technology and Equipment, 2021, 57(3): 81-82.
[3] He Feng. Design of a Small Quad-Rotor UAV Based on STM32+MPU6050 [J]. Computer Knowledge and Technology, 2020.
[4] Xiong Yingchao. Design and Optimization of a Quad-Rotor UAV [J]. Mechanical Engineering & Automation, 2022(05): 124-125+128.
[5] Zhang Lijuan. Attitude Angle Estimation and Control of a Quad-Rotor Aircraft [D]. Xi’an: Xi’an University of Science and Technology, 2016.
[6] Zhang Hao. Research on Key Technologies of the Control System of a Quad-Rotor Aircraft [D]. Harbin: Harbin Engineering University, 2017.