Research on the Stability Control of Quadrotor Based on Arduino

Jing BAIa,*
College of Information Technology, Shanghai Jian Qiao University, Shanghai 201306, China

a15008@gench.edu.cn
*Corresponding author

Keywords: Quadrotor, Double closed loop PID control, Euler angle.

Abstract. This paper presented a modified PID control algorithm named double closed loop PID control algorithm, and verified this algorithm in practice. The new algorithm solves the problem which the miscalculation of Euler angle makes quadrotor unable to adjust flight attitude accurately and quickly. The double closed loop PID control algorithm selects angular velocity as inner loop, because angular velocity collected by gyroscope will not be affected by external interference, and selects angle as outer loop. Similarly, the new algorithm selects acceleration on Z axis as inner loop and selects height as outer loop when applied to regulate position. Finally, the results of simulation and flight experiment proved that the stability and anti-jamming capability of quadrotor are improved by the double closed loop PID control algorithm.

Introduction

The quadrotor is a kind of UAV (Unmanned Aerial Vehicle) with the features of small volume, light weight, simple structure, reliable performance, flexible control. It can be applied in traffic surveillance, aerial video, forest fire prevention and rescuing in complex environment. In recent years, because of those characteristics, the quadrotor has been widely used in civil and military fields, and became an important direction in the field of aerospace research in the world [1].

But, the system of quadrotor is non-linear, multi-variable, strong-coupled, less-driven and susceptible to interference. It’s difficult to obtain accurate real-time flight data to adjust the flight attitude quickly because of its features of small rotor, light weight, easy to be disturbed by many kinds of physics effect (aerodynamics, spiral effect and moment of inertia, etc.). Therefore, the key to keep the stability of the quadrotor is how to accurately and quickly obtain the external interference and then how to correct the error caused by these interference.

Principle of Flight

The quadrotor is mainly composed of a main control board and four electronic speed governors, motors, rotors in criss-cross construction. The main control board is mainly responsible for calculating the current flight attitude and controlling the electronic speed governors. The electronic speed governors adjust the running speed of the motor in order to control the four rotors, and then eventually control the flight attitude of quadrotor.
As shown in Fig. 1, take the cross flight mode for example, Rotor 1 is the head. Rotor 1 and 3 make counterclockwise rotation, while the rotor 2 and 4 make clockwise rotation, so as to balance the torque caused by the rotating of rotors.

![Figure 1. Structure chart of quadrotor.](image)

In spatial, there are 6 DOF (degree of freedom), which are translation and rotation along three axes [2]. In order to make the quadrotor move, there needs a thrust generated by increasing the speed of one of a set rotors (rotor 1, 3 or rotor 2, 4) and decreasing the speed of the other one of a set rotors simultaneously. For example, the rotational speed difference of rotor 2 and 4 is used to control the roll rate which can make quadrotor do the roll or left-to-right movement. The rotational speed difference of rotor 1 and 3 is used to control pitch rate which can make quadrotor do the pitch or front-to-back movement. The rotational speed difference of clockwise rotors (rotor 2, 4) and counterclockwise rotors (rotor 1, 3) is used to control yaw rate which can make quadrotor do the yaw movement. Increasing the speed of the 4 rotors simultaneously can make the quadrotor do the vertical movement. When the lift caused by the rotating of rotors is greater than the gravity, the quadrotor will rise vertically. On the contrary, the quadrotor will fall vertically. When the speeds of four rotors are equal, meaning the lift is equal to the gravity, the quadrotor will be in a steady hovering state [3]. The corresponding speed adjustment of four motors to different flight modes as shown in Table 1.

<table>
<thead>
<tr>
<th>Flight mode</th>
<th>Motor 1</th>
<th>Motor 2</th>
<th>Motor 3</th>
<th>Motor 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vertical rise (fall)</td>
<td>Increase (decrease)</td>
<td>Increase (decrease)</td>
<td>Increase (decrease)</td>
<td>Increase (decrease)</td>
</tr>
<tr>
<td>Pitch up(down)/Front(back)</td>
<td>Decrease (increase)</td>
<td>Invariant (invariant)</td>
<td>Increase (decrease)</td>
<td>Invariant (invariant)</td>
</tr>
<tr>
<td>Roll left(right)/Left(right)</td>
<td>Invariant (invariant)</td>
<td>Decrease (increase)</td>
<td>Invariant (invariant)</td>
<td>Increase (decrease)</td>
</tr>
<tr>
<td>Yaw left(right)</td>
<td>Decrease (increase)</td>
<td>Increase (decrease)</td>
<td>Decrease (increase)</td>
<td>Increase (decrease)</td>
</tr>
</tbody>
</table>
Attitude Calculation

The system of quadrotor is unstable because it is a less-driven system with the features of nonlinearity, multi-variables, strong-coupling. The key to maintain the stability of the quadrotor is to detect, control and maintain the flight attitude. Flight attitude is generally presented by Euler angle, including yaw angle, pitch angle and roll angle. The attitude angle of the flying quadrotor is obtained by calculating the data which obtained from sensors of the main control board. The attitude angle is controlled and adjusted to keep the quadrotor stable.

Flight attitude represents the transformation relationship between the body coordinate system and geographic coordinate system. Define the flight attitude is the transformation from the body coordinate system to the geographic coordinate system. There are many ways to express the rotation transformation, including direction cosine, Euler angles, quaternion method. In this paper, the quaternion method is adopted. In fact, the quaternion method is a kind of calculation which transfer the angle to multidimensional space, whose amount of computation is greatly reduced compared to that of Euler angle or direction cosine methods.

Hardware Selection

Select the chip model MPU6050 as gyroscope and acceleration sensor, the chip model HMC5883L as magnetic sensor. The communication mode of the main control board selects IIC bus.

Due to the device error exists in the sensor, it should be amended before use. The gyroscope of MPU6050 has a zero drift, that is, in the case of the quadrotor does not move, the gyroscope will have an output. And the zero drift value is not sure, which will change with different environment. So, at the time of initialization, the zero drift of a gyroscope shall be adjusted. The method adopted in this paper is, for the pitch angle and roll angle, reading 100 zero drift values as the mean of zero drift at rest, then during working, the data read by gyroscope are subtracted from this mean to get the corrected data; for yaw angle, it is measured by magnetic sensor, which is the angle value read by MCU during the initialization of quadrotor after power-up.

Attitude Calculation by Quaternion Method

Initialize quaternion, refer with: Eq. 1.

\[
\mathbf{q} = \begin{bmatrix} q_0 \ q_1 \ q_2 \ q_3 \end{bmatrix}^T = \begin{bmatrix} 1 \ 0 \ 0 \ 0 \end{bmatrix}^T.
\]  

Obtain the acceleration \((a_x', a_y', a_z')\), and angular velocity \((\omega_x', \omega_y', \omega_z')\), of the three axis from the gyroscope.

Normalize the acceleration of three axis (transform into a unit vector), refer with: Eq. 2.

\[
\begin{align*}
\alpha_x &= \frac{a_x}{\sqrt{a_x^2 + a_y^2 + a_z^2}} \\
\alpha_y &= \frac{a_y}{\sqrt{a_x^2 + a_y^2 + a_z^2}} \\
\alpha_z &= \frac{a_z}{\sqrt{a_x^2 + a_y^2 + a_z^2}}.
\end{align*}
\]
Obtain the gravity unit vector of three axis from quaternion[4]. First, determine the coordinate transformation (direction cosine) matrix by quaternion, refer with: Eq. 3.

\[
C_b^R = \begin{bmatrix}
q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_0q_2 - q_1q_3) & 2(q_0q_2 + q_1q_3) \\
2(q_0q_2 + q_1q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2q_3 - q_0q_1) \\
2(q_0q_3 - q_1q_2) & 2(q_2q_3 + q_0q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2
\end{bmatrix}
\]

(3)

Then, according to the definitions of cosine matrix and Euler angle, the unit vectors of gravity \((v_x, v_y, v_z)\) are the three elements in the third line of the cosine matrix in equation, refer with: Eq. 4.

\[
\begin{align*}
v_x &= 2(q_1q_3 - q_0q_2) \\
v_y &= 2(q_2q_3 + q_0q_1) \\
v_z &= q_0^2 - q_1^2 - q_2^2 + q_3^2.
\end{align*}
\]

(4)

Calculate the error to correct the data obtain from gyroscope. \(a_x, a_y\) and \(a_z\) are the acceleration vectors in the body coordinate system, which measured by acceleration sensor. \(v_x, v_y\) and \(v_z\) are unit vectors of gravity calculated according to the integrated attitude data from gyroscope. The error vectors between acceleration vectors and gravity unit vectors are expressed by \(e_x, e_y\) and \(e_z\), which are the cross products of two vectors, refer with: Eq. 5.

\[
\begin{align*}
e_x &= a_yv_z - a_zv_y \\
e_y &= a_zv_x - a_xv_z \\
e_z &= a_xv_y - a_yv_x.
\end{align*}
\]

(5)

Integral correct and complementary filter the measurement of gyroscope. Put the error vectors \((e_x, e_y, e_z)\) into PI controller and then obtain the corrected angular velocity \((\omega_x, \omega_y, \omega_z)\) by adding the output of PI controller to the angular velocity measured by gyroscope in the new update cycle, refer with: Eq. 6.

\[
\begin{align*}
\omega_x &= \omega_{x0} + k_xe_x + k_\int_0^t e_x \, dt \\
\omega_y &= \omega_{y0} + k_ye_y + k_\int_0^t e_y \, dt \\
\omega_z &= \omega_{z0} + k_re_z + k_\int_0^t e_z \, dt
\end{align*}
\]

(6)

Update the quaternion using the corrected angular velocity \((\omega_x, \omega_y, \omega_z)\). Calculate Eq. 7 (quaternion differential equation) to update the quaternion to obtain Eq. 8, of which, the \(t\) is the measurement period.

\[
\begin{bmatrix}
q_0 \\
q_1 \\
q_2 \\
q_3
\end{bmatrix}_{t+\Delta t} = \begin{bmatrix}
q_0 \\
q_1 \\
q_2 \\
q_3
\end{bmatrix}_t + \frac{\Delta t}{2} \begin{bmatrix}
-\omega_xq_1 - \omega_yq_2 - \omega_zq_3 \\
+\omega_xq_0 - \omega_yq_3 + \omega_zq_2 \\
+\omega_yq_3 + \omega_xq_0 - \omega_zq_1 \\
-\omega_zq_2 + \omega_yq_1 + \omega_xq_0
\end{bmatrix}
\]

(7)
\[
\begin{align*}
q'_0 &= q_0 + \frac{t}{2}(-q_1\omega_x - q_2\omega_y - q_3\omega_z) \\
q'_1 &= q_1 + \frac{t}{2}(q_0\omega_x - q_3\omega_y - q_2\omega_z) \\
q'_2 &= q_2 + \frac{t}{2}(q_3\omega_x + q_0\omega_y - q_1\omega_z) \\
q'_3 &= q_3 + \frac{t}{2}(-q_2\omega_x + q_1\omega_y + q_0\omega_z)
\end{align*}
\] 

(8)

Normalize the quaternion, refer with: Eq. 9.

\[
\begin{align*}
q''_0 &= \frac{q'_0}{\sqrt{q^{'2}_0 + q^{'2}_1 + q^{'2}_2 + q^{'2}_3}} \\
q''_1 &= \frac{q'_1}{\sqrt{q^{'2}_0 + q^{'2}_1 + q^{'2}_2 + q^{'2}_3}} \\
q''_2 &= \frac{q'_2}{\sqrt{q^{'2}_0 + q^{'2}_1 + q^{'2}_2 + q^{'2}_3}} \\
q''_3 &= \frac{q'_3}{\sqrt{q^{'2}_0 + q^{'2}_1 + q^{'2}_2 + q^{'2}_3}}
\end{align*}
\]

(9)

Transform the quaternion to Euler angles according to the transformation relationship between them, refer with: Eq. 10, of which, \(\psi\) is the yaw, \(\theta\) is the pitch, and \(\varphi\) is the roll.

\[
\begin{align*}
\psi &= \arctan\left(\frac{2(q''_0q''_2 + q''_0q''_3)}{1 - 2(q''_2^2 + q''_3^2)}\right) \\
\theta &= \arcsin\left(\frac{2(q''_0q''_2 - q''_0q''_3)}{1 - 2(q''_2^2 + q''_3^2)}\right) \\
\varphi &= \arctan\left(\frac{2(q''_0q''_1 - q''_0q''_3)}{1 - 2(q''_1^2 + q''_2^2)}\right)
\end{align*}
\]

(10)

**Height Calculation**

The height can be calculated using the pressure-to-height transformation Eq. 11. The accurate pressure is obtained through second-order temperature compensation for the pressure which is collected by the pressure sensor.

\[
\text{Height} = 44330 \left(1 - \frac{\text{PresentPressure}}{\text{PressureBeforeDeparture}}\right)^{0.1903}
\]

(11)

**Control Algorithm**

In the current popular research of quadrotor, most controllers use PID control algorithm. The process of traditional PID control algorithm is shown in Fig. 2. Input data is the error between the expected angle and the angle which is collected by gyroscope and then calculated by main control board. Output data is the input data of the controlled object.
But, if quadrotor meet a sudden interference (such as wind, etc.) or magnetic-interference when it fly, the angle data collected by sensors will be distorted, which will cause miscalculation of the Euler angle. If only the angle to do PID control, it will make the input data incorrect, leading to reduce the stability of the system. While, the angular velocity collected by gyroscope will not be affected by external interference [5]. Therefore, double closed loop PID control algorithm is adopted in this paper, in which the angular velocity is added as the inner loop PID control.

Similarly, when regulating position (setting the height), the data collected by pressure sensor will also be affected by external interference. Adding acceleration as the inner loop can avoid this problem.

The process of double closed loop PID control algorithm is shown in Fig. 3 and Fig. 4.

For angle PID control, input data of the outer loop is the error between desired Euler angle obtained from the remote-controller and the Euler angle attitude-calculated according to the real-time angles obtained by gyroscope. Output data of the outer loop is the desired angular velocity. Input data of the inner loop of is the error between the angular velocity collected by gyroscope and the desired angular velocity (output data of the outer loop).

For height PID control, input data of the outer loop is the error between desired height obtained from the remote-controller and the height collected from pressure sensor. Output data of the outer loop is the desired acceleration. Input data of the inner loop is the error between the Z axis acceleration collected by gyroscope and the desired acceleration (output data of the outer loop).

For angle and height PID control, the inner loop outputs the throttle value to the electronic speed governors which controls the flight attitude and height by adjusting running-speed of the motor. But the throttle value must be amplitude-limited before being given to the throttle, so as to avoid the instability caused by the too large output of the throttle.
Positional PID control method is shown in Eq. 12.

\[ u(t) = k_p e(t) + k_i \int_0^t e(t) \, dt + k_d \frac{de(t)}{dt} \]  

(12)

Discrete PID control method is adopted in this algorithm, compared to positional PID control method, which the microprocessor only need to output the increment so as to avoid accidents caused by the sudden increase or sudden decrease of control variable. Discrete PID control method is shown in Eq. 13.

\[ u(t) = k_p e(t) + k_i \sum_{j=0}^t e(j)T + k_d \frac{e(t) - e(t-1)}{T} \]  

(13)

Based on the above analysis of Fig. 3 and Fig. 4, the Eq. 14 (outer loop of the controller) and Eq. 15 (inner loop of the controller) are gained. Similarly, the equation of height PID control algorithm is shown in Eq. 16 and Eq. 17.

\[ \text{AngelPIDOut}(t) = k_p e(t) + k_i \sum_{j=0}^t e(j)T + k_d \frac{e(t) - e(t-1)}{T} \]  

(14)

\[ \text{AngelRatePIDOut}(t) = k_p e(t) + k_i \sum_{j=0}^t e'(j)T + k_d \frac{e'(t) - e'(t-1)}{T} \]  

(15)

\[ \text{AltitudePIDOut}(t) = k_p e(t) + k_i \sum_{j=0}^t e(t)T + k_d \frac{e(t) - e(t-1)}{T} \]  

(16)

\[ \text{AcceleratePIDOut}(t) = k_p e(t) + k_i \sum_{j=0}^t e''(j)T + k_d \frac{e''(t) - e''(t-1)}{T} \]  

(17)

Output data of the inner loop PID must be amplitude-limited before being given to the throttle, so as to avoid the instability caused by the too large output of the throttle. Therefore, limit the amplitude of the throttle from -100 to +100, as shown in Eq. 18 and Eq. 19.

\[ \text{Angel}_{\text{out}} = \text{AngelPIDOut}(t)(\text{LimitRange:} -100\sim +100) \] \hspace{1cm} (18)

\[ \text{Altitude}_{\text{out}} = \text{AcceleratePIDOut}(t)(\text{LimitRange:} -10\sim +10) \] \hspace{1cm} (19)
According to the throttle output Eq. 20, the throttle increment of four electronic speed governors is gained, after the attitude calculation, PID control, and amplitude limit.

$$\begin{align*}
\text{Throttle}_1 &= \text{Throttle}_\phi - \text{Throttle}_\psi - \text{Throttle}_\psi \\
\text{Throttle}_2 &= -\text{Throttle}_\phi + \text{Throttle}_\psi + \text{Throttle}_\psi \\
\text{Throttle}_3 &= -\text{Throttle}_\phi + \text{Throttle}_\psi + \text{Throttle}_\psi \\
\text{Throttle}_4 &= \text{Throttle}_\phi + \text{Throttle}_\psi + \text{Throttle}_\psi
\end{align*}$$

(20)

**Simulation and Experiment**

Use the debug soft MultiWiiConf for this simulation. When the quadrotor adopts traditional PID control algorithm, the oscillation of pitch angle, yaw angle and roll angle is shown in Fig. 5.

Assignments of each group for PID are shown in Table 2.

<table>
<thead>
<tr>
<th>Assignment</th>
<th>outer loop P</th>
<th>outer loop I</th>
<th>outer loop D</th>
<th>inner loop P</th>
<th>inner loop I</th>
<th>inner loop D</th>
</tr>
</thead>
<tbody>
<tr>
<td>Roll /X axis</td>
<td>3.3</td>
<td>0.03</td>
<td>23</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pitch /Y axis</td>
<td>3.3</td>
<td>0.03</td>
<td>23</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Yaw /X axis</td>
<td>6.8</td>
<td>0.045</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Height setting</td>
<td>6.4</td>
<td>0.025</td>
<td>24</td>
<td>7</td>
<td>0.045</td>
<td>2</td>
</tr>
<tr>
<td>GPS</td>
<td>0.15</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Position Regulation</td>
<td>3.4</td>
<td>0.14</td>
<td>0.053</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Navigation</td>
<td>2.5</td>
<td>0.33</td>
<td>0.083</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Level</td>
<td>9</td>
<td>0.01</td>
<td>100</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Magnetism</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

When regulating position, the oscillation of pitch angle, yaw angle and roll angle is shown in Fig. 6. It can be seen that the oscillation amplitude of the attitude angle is
much smaller when adopting double closed loop PID control algorithm, meaning that the system is more stable.

Figure 6. Oscillation of the attitude angle using double closed loop PID control algorithm when regulating position.

The experiment of the quadrotor in outdoors is shown in Fig. 7. The quadrotor is very stable when regulating position (setting height), and it can adjust quickly in flight, with strong ability to resist interference. The above proves that the flight control algorithm achieves the expected goal.

Figure 7. Flight experiment of the quadrotor in outdoors.

**Conclusion**

In this paper, the quaternion method is applied to calculate the attitude, and improvements have been made in control algorithm, which is changed from traditional PID control algorithm to double closed loop PID control algorithm. In the double closed loop PID control algorithm, angle is used as outer loop and angular velocity is used as inner loop for attitude PID control; height is used as outer loop and Z axis acceleration is used as inner loop for height PID control. It also proves that the double closed loop PID control algorithm greatly increases the stability of the system by comparison of data through simulation and experiment.
References


