Research on PID Control of Quadrotor Drones Based on MATLAB

Qin Zhilun

Civil Aviation University of China, Yichang, Hubei, 443000, China
Corresponding author’s e-mail: 18671738460@163.com

Abstract

Four-rotor UAV is a very practical and widely used UAV. In this paper, the development status and future trend of Four-rotor UAV at home and abroad are introduced, and then the transformation matrix from ground coordinates to airframe coordinates is deduced by using Euler equation. In the process of dynamic modeling, the plane position reference coordinate system and rotation angle reference coordinate system are selected to analyze the external forces and moments on the body, and the linear motion equation and the angular motion equation are written in parallel. On the basis of UAV dynamics model, the classical PID control method is used to control the attitude of the inner loop and the position of outer loop. According to simulation results, the flight controller can keep the UAV flying stably.

Keywords: MATLAB; PID control; Four-rotor UAV; Linear motion equation; Angular motion equation

Chapter 1: Introduction

1.1 Background and Significance of this study

With the rapid advancement of aerospace and aviation, multi-rotor drones have experienced swift development. Thanks to their simple mechanical structure and flight versatility, they are applicable to a variety of production and life scenarios.

Due to the complex characteristics of quadrotor drones, the design of their control systems poses particular challenges. The focus of this research is to establish an accurate mathematical model and to design an effective controller.

1.2 Main Research Content

The main aim of this study is to explore the underactuated characteristics of quadrotor drones and investigate their flight postures by establishing their dynamical models. The control system will be designed using classical PID control methods to manage the motion attitudes.

Chapter 2: Mathematical Model of Quadrotor Drones

Quadrotor drone control systems are underactuated, nonlinear dynamical systems with four control inputs and six degrees of freedom in outputs. Various variables within the system are interdependent. Drones are affected by various unpredictable environmental factors during flight; therefore, establishing a mathematical model for their dynamical system is crucial.

2.1 Principles of Quadrotor Drone Flight

The direct power source for quadrotor drones comes from motors. There are two types of rotor installation configurations: the X-configuration and the cross-configuration\cite{1}. Changes in the flight posture of the quadrotor drone will involve six variables: three translational components generated by translational motion and three angular components generated by rotational motion. Control over these variables was achieved by altering the rotational speeds of motors.

Figure 2.1: Structural Framework of Quadrotor Drone

Due to the coupling between the rotational and translational motions of the quadrotor drone, altering its attitude angles allows it to fly according to setted trajectory. The varying speeds of its four
motors enable the quadrotor drone to freely fly in space. Figure 2.1, as mentioned above, depicts the structural framework of quadrotor drone.

2.2 Mathematical Model of the Quadrotor Aircraft

2.2.1 Modeling Assumptions

The quadrotor control system is a nonlinear control system, making it hard to construct an exact math model. For the sake of research convenience, the following assumptions are made:

(1) The drone has a symmetrical external structure with uniformly distributed mass.
(2) The geometric center coincides with body coordinate system origin.
(3) Gravitational effects due to the distance between objects are ignored; the gravitational force remains constant.

(4) Thrust in every directions is directly proportional to motor speed square.
(5) The gravitational forces keep constant during flight.
(6) The airflow is stable; friction torques are ignored.
(7) The quadrotor performs low-speed and low-angle flight.

As shown in Figure 2.5, under the influence of lift, the drone body generates pitch, roll, yaw torques. Upon the relationships among driving force, lift, these three torques, and the relationships among six degrees of freedom induced by changes in drone’s flight posture, a Newton-Euler model was developed. This model allows for the calculation and output of six acceleration quantities. These acceleration quantities, after undergoing double integration, yield the position of the drone body.

2.2.2 Linear Motion Equations of the Quadrotor Drone

The quadrotor drone has six degrees of freedom, corresponding to the parameters $x, y, z, \phi, \theta, \psi$. The meanings of these parameters can be seen from Table 2.2.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x$</td>
<td>Refer to horizontal direction $x$</td>
</tr>
<tr>
<td>$y$</td>
<td>Refer to horizontal direction $y$</td>
</tr>
<tr>
<td>$z$</td>
<td>Refer to horizontal direction $z$</td>
</tr>
<tr>
<td>$\phi$</td>
<td>Roll angle, angular acceleration is $\phi$</td>
</tr>
<tr>
<td>$\theta$</td>
<td>Pitch angle, angular acceleration is $\theta$</td>
</tr>
<tr>
<td>$\psi$</td>
<td>Yaw angle, angular velocity is denoted as $\psi$</td>
</tr>
</tbody>
</table>

In 3D space, a rigid body rotating around origin has three degrees of freedom, described in full by three generalized coordinates. If the three coordinate quantities rotate according to right-hand rule, their basic rotations are as follows.

Let $x$-axis fixed, rotate by angle $\phi$ around $x$ axis, and represent the new coordinates as Equation (1).

$$
\begin{bmatrix}
  x_2 \\
  y_2 \\
  z_2
\end{bmatrix} =
\begin{bmatrix}
  1 & 0 & 0 \\
  0 & \cos \phi & \sin \phi \\
  0 & -\sin \phi & \cos \phi
\end{bmatrix} \begin{bmatrix}
  x_1 \\
  y_1 \\
  z_1
\end{bmatrix}
$$

(1)

Let $y$-axis fixed, rotate by angle $\theta$ around $y$ axis, and represent the new coordinates as Equation (2).

$$
\begin{bmatrix}
  x_2 \\
  y_2 \\
  z_2
\end{bmatrix} =
\begin{bmatrix}
  \cos \theta & 0 & -\sin \theta \\
  0 & 1 & 0 \\
  \sin \theta & 0 & \cos \theta
\end{bmatrix} \begin{bmatrix}
  x_1 \\
  y_1 \\
  z_1
\end{bmatrix}
$$

(2)

Let $z$-axis fixed, rotate by angle $\psi$ around $z$-axis, and represent the new coordinates as Equation (3).
The core idea of Euler angle coordinate transformation is that one coordinate system can be expressed by three spatial rotations of another reference coordinate system. The relevant transformation matrices for rotations around the x, y, z axes have been derived above. Let us denote these by Equations (4), (5), and (6).

\[
C_x = \begin{bmatrix}
1 & 0 & 0 \\
0 & \cos \phi & \sin \phi \\
0 & -\sin \phi & \cos \phi
\end{bmatrix}
\]

\[
C_y = \begin{bmatrix}
\cos \theta & 0 & -\sin \theta \\
0 & 1 & 0 \\
\sin \theta & 0 & \cos \theta
\end{bmatrix}
\]

\[
C_z = \begin{bmatrix}
\cos \psi & \sin \psi & 0 \\
-\sin \psi & \cos \psi & 0 \\
0 & 0 & 1
\end{bmatrix}
\]

Combining Equations (4), (5), (6), the attitude matrix expressed by Euler angles is given by Equation (7).

\[
C = C_x C_y C_z
\]

\[
F_T = C_b^T F_T^b = \sum_{i=1}^{4} F_i^b
\]

The gravitational force G was expressed in matrix form as:

\[
G = \begin{bmatrix} 0 & 0 & mg \end{bmatrix}^T
\]

The air resistance is:

\[
F_D = K_D \begin{bmatrix} x \\ y \\ z \end{bmatrix}
\]

In Equation (13), \(K_D = \text{diag}(K_{Dx}, K_{ Dy}, K_{Dz})\) represents the drag coefficient matrix. Thus, the net force is:

\[
F = F_T - F_D - G
\]

Applying force Law \(F = ma\), we can combine it with Equation (14) to obtain Equation (15).
\[m \ddot{x} = \left( \sin \theta \cos \phi \cos \psi + \sin \phi \sin \psi \right) \sum_{i=1}^{4} F_i^b - K_{Dx} x\]
\[m \ddot{y} = \left( \sin \theta \cos \psi \sin \phi \right) \sum_{i=1}^{4} F_i^b - K_{Dy} y\]
\[m \ddot{z} = \cos \theta \sin \phi \sum_{i=1}^{4} F_i^b - K_{Dz} z - mg\]  

\[2.2.4 \text{ Angular Motion Equations of the Quadrotor}\]  

According to Euler’s equations, in an inertial system, the linear and angular motion equations of drone can be expressed as Equation (16):

\[
\begin{align*}
F_s &= \frac{dP}{dt} = m \frac{d\dot{V}}{dt} \\
M_s &= \frac{dL}{dt} = J \frac{dw_{body}}{dt}
\end{align*}
\]

\[16\]

\[
\begin{align*}
F_s &= \left[ \begin{array}{c} F_{s_x} \\ F_{s_y} \\ F_{s_z} \end{array} \right] \\
M_s &= \left[ \begin{array}{c} M_{s_x} \\ M_{s_y} \end{array} \right]
\end{align*}
\]

Table 2.3 Symbol Explanation

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$F_s$</td>
<td>Net external force</td>
</tr>
<tr>
<td>$w_s$</td>
<td>Algebraic sum of the rotor speeds</td>
</tr>
<tr>
<td>$L$</td>
<td>Angular momentum</td>
</tr>
<tr>
<td>$W^b$</td>
<td>Angular velocity in body coordinate system</td>
</tr>
<tr>
<td>$I_{3x3}$</td>
<td>Unit matrix</td>
</tr>
<tr>
<td>$F^b_s$</td>
<td>Net external force in body coordinate system</td>
</tr>
<tr>
<td>$M_s$</td>
<td>Net torque about a certain rotation axis</td>
</tr>
<tr>
<td>$M_s^b$</td>
<td>Net external torque</td>
</tr>
<tr>
<td>$M_s^b$</td>
<td>Net torque along the $x$-axis</td>
</tr>
<tr>
<td>$M_s^b$</td>
<td>Net torque along the $y$-axis</td>
</tr>
<tr>
<td>$I$</td>
<td>Inertia tensor of the body</td>
</tr>
<tr>
<td>$m$</td>
<td>Total mass of aircraft</td>
</tr>
<tr>
<td>$l$</td>
<td>Distance of mass center to rotation axis</td>
</tr>
<tr>
<td>$b$</td>
<td>Lift coefficient of the rotor</td>
</tr>
<tr>
<td>$v$</td>
<td>Linear velocity component along the $y$-axis</td>
</tr>
<tr>
<td>$\nu^b$</td>
<td>Linear velocity vector relative to the body coordinate system</td>
</tr>
</tbody>
</table>

Due to the vectorial nature of force and torque, the net results for rigid body motion, both rotational and translational, can be described by Newton-Euler equations:

\[
\begin{align*}
F_s^b &= \frac{dP}{dt}_{rot} + W^b \times P = m \left( \dot{V}^b + W^b \times V^b \right) \\
M_s^b &= \frac{dL}{dt}_{rot} + W^b \times L = I W^b + W^b \times \left(I W^b\right)
\end{align*}
\]

Combining this with the motion of the aircraft, the above equation is represented in matrix form:

\[
\begin{pmatrix}
ml_{3x3} & 0_{3x3} \\
0_{3x3} & I
\end{pmatrix}
\begin{pmatrix}
\dot{V}^b \\
\dot{W}^b
\end{pmatrix} = \begin{pmatrix}
F^b_s \\
M^b_s
\end{pmatrix} - \begin{pmatrix}
W^b \times (mV^b) \\
W^b \times (I W^b)
\end{pmatrix}
\]

In Equation (18), linear motion equations are established. $W^b \times (mV^b)$, the rotational quantity around the body, has zero displacement in this system. Therefore, the linear motion equation is:

\[
mV = F_s
\]

For body coordinate system, angular motion equation remains unchanged:

\[
M^b_s = I W^b + W^b \times \left(I W^b\right)
\]

The quadrotor drone has excellent symmetry. Based on the definition of the area moment of inertia, we can deduce that $I_{yx} = I_{xy} = I_{yz} = I_{zy} = I_{xz} = I_{zx} = 0$, the moments of inertia about the $x$, $y$, $z$ axes are non-zero. The inertia matrix of body is represented as Equation (21):
\[
I = \begin{pmatrix}
I_{xx} & I_{xy} & I_{xz} \\
I_{yx} & I_{yy} & I_{yz} \\
I_{zx} & I_{zy} & I_{zz}
\end{pmatrix} = \begin{pmatrix}
I_x & 0 & 0 \\
0 & I_y & 0 \\
0 & 0 & I_z
\end{pmatrix}
\]

(21)

Here, \(I_x, I_y, I_z\) correspond to the moments of inertia about the \(x, y, z\) axes. \(W^b = \left[\begin{array}{c} w_x \\ w_y \\ w_z \end{array}\right]\), in which \(w_x, w_y, w_z\) are the components of vector \(W^b\) along the \(x, y, z\) axes.

\[
W^b \times (IW^b) = \begin{pmatrix}
 i & j & k \\
 w_x & w_y & w_z \\
 I_{xx} & I_{xy} & I_{xz}
\end{pmatrix} = \begin{pmatrix}
w_xw_y(I_x - I_y) \\
w_yw_z(I_y - I_z) \\
w_zw_x(I_z - I_x)
\end{pmatrix}
\]

(22)

Combining the above two equations and after simplification, we get:

\[
\begin{align*}
M_x^b &= I_xw_x + (I_z - I_y)w_yw_z \\
M_y^b &= I_yw_y + (I_z - I_x)w_zw_x \\
M_z^b &= I_zw_z + (I_x - I_y)w_xw_y
\end{align*}
\]

(23)

The lift torque on the drone body in the three axes is given by:

\[
\begin{align*}
M_{x1}^b &= l(F_x^b - F_y^b) \\
M_{y1}^b &= l(F_y^b - F_z^b) \\
M_{z1}^b &= -M_{x1}^b + M_{y1}^b + M_{x2}^b + M_{y2}^b
\end{align*}
\]

(24)

\(M_{ib}^b (i = 1, 2, 3, 4)\) represents the torque experienced by each rotor along the \(z\)-axis during flight and is expressed as:

Assuming the angular velocities \(iM_{ib}^b = dw_i^2\), where \(w_i = (i = 1, 2, 3, 4)\), then the individual lift force generated by each rotor was represented as:

\[
F_i^b = bw_i^2
\]

(26)

Here, \(b\) is the lift coefficient of the rotor, leading to:

\[
\begin{align*}
M_{x1}^b &= lb(w_x^2 - w_y^2) \\
M_{y1}^b &= lb(w_y^2 - w_z^2) \\
M_{z1}^b &= d(-w_x^2 + w_y^2 - w_z^2 + w_z^2)
\end{align*}
\]

(27)

During rotation, the object will experience gyroscopic effects. The quadrotor drone performs high-speed rotations in opposite directions between adjacent rotors during flight. When the direction of angular momentum changes due to a change in attitude, the rotor generates a torque. When the torques from all four rotors cannot cancel each other out, a gyroscopic torque is generated, causing the body to deviate. This can be expressed as:

\[
M_g^b = \sum_{i=1}^{4} W^b \times (I_i \Omega_i)
\]

(28)

Here, \(\Omega_i = \left[\begin{array}{c} 0 \\ 0 \\ (-1)^i w_i^2 \end{array}\right]\); \(I_i\) is the rotor’s moment of inertia. Simplifying, we get:

\[
M_g^b = I_i(-w_i + w_3 - w_4 + w_4) \begin{pmatrix} w_y \\ -w_z \\ 0 \end{pmatrix} = -I_iw_yw_z
\]

(29)

In this equation, \(w_i\) is the algebraic sum of the speeds of the four rotors, i.e., \(w_i = -w_1 + w_2 - w_3 + w_4\). The gyroscopic effects generated by the rotor speeds are solely dependent on the angular velocity.

From the above equations, the net torque can be determined as:

\[
\begin{align*}
M_x^b &= M_{x1}^b + M_{x2}^b = lb(w_x^2 - w_y^2) + (-I_iw_yw_z) \\
M_y^b &= M_{y1}^b + M_{y2}^b = lb(w_y^2 - w_z^2) + (-I_iw_yw_z) \\
M_z^b &= M_{z1}^b + M_{z2}^b = d(-w_x^2 + w_y^2 - w_z^2 + w_z^2)
\end{align*}
\]

(30)

In vector form, this can be written as:

\[
\begin{align*}
I_xw_x = (I_x - I_z)w_yw_z + I_yw_zw_x + lb(w_x^2 - w_y^2) \\
I_yw_y = (I_y - I_x)w_zw_x - I_xw_yw_z + lb(w_y^2 - w_z^2) \\
I_zw_z = (I_z - I_y)w_xw_y + d(-w_x^2 + w_y^2 - w_z^2 + w_z^2)
\end{align*}
\]

(31)

Combining the above analyses, Equations (15) and (31) represent its nonlinear motion equations.

2.2.5 Kinematic Model

In the case of translational motion of the quadrotor drone, assuming that the velocity components are known, \(V^b\) can be represented as a vector: \(V^b = (u, v, w)^T\).

Transforming \(V^b\) to the ground coordinate system is expressed as equation (32).

\[
\begin{align*}
x &= u \cos \theta \cos \psi + v(\sin \theta \cos \phi \cos \psi - \cos \theta \sin \psi) \\
&\quad + w(\sin \theta \cos \phi \cos \psi + \sin \phi \sin \psi) \\
y &= u \cos \theta \sin \psi + v(\sin \theta \sin \phi \sin \psi - \cos \phi \cos \psi) \\
&\quad + w(\sin \theta \cos \phi \sin \psi - \sin \phi \cos \psi) \\
z &= -u \sin \theta + v \sin \phi \cos \theta + w \cos \phi \cos \theta
\end{align*}
\]

(32)

The angular velocity vector of drone is \(W^b\), The
relationship between its three angular velocity components along the axes and the three angular rates in body coordinate system were described by Equation (33):

\[
W^b = \begin{bmatrix}
    w_x \\
    w_y \\
    w_z 
\end{bmatrix} = C_x C_y \begin{bmatrix}
    0 \\
    0 \\
    0
\end{bmatrix} + C_z \begin{bmatrix}
    \theta \\
    \phi \\
    \psi
\end{bmatrix} + \begin{bmatrix}
    0 \\
    0 \\
    0
\end{bmatrix} \phi
\]

\[
= \begin{bmatrix}
    1 & 0 & -\sin \theta \\
    0 & \cos \phi & \sin \phi \cos \theta \\
    0 & -\sin \phi & \cos \phi \cos \theta
\end{bmatrix} \begin{bmatrix}
    \phi \\
    \theta \\
    \psi
\end{bmatrix}
\]

Upon transformation, we get

\[
\begin{align*}
\phi &= w_x + (w_z \cos \phi + w_y \sin \phi) \tan \theta \\
\theta &= w_y \cos \phi - w_z \sin \phi \\
\psi &= \frac{1}{\cos \theta} (w_z \cos \phi + w_y \sin \phi)
\end{align*}
\]

\[
\theta = \pm \frac{\pi}{2} \theta = \pm \frac{\pi}{2}
\]

In the above equation, \(\cos \theta\) appears in the denominator. Near \(\cos \theta\), it is not possible to numerically solve for Euler angles using angular velocity. Therefore, a singularity in the Euler angle representation. This Equi is also represented as the rotational motion equation of system dynamics, reflecting the association of the three components of angular and the attitude angular velocity.

From the previous derivation, it is known that the mathematical model of the drone includes four sets of equations: the equations of motion, torque equations, navigation equations, and kinematic equations. After organizing, the nonlinear mathematical model of the quadrotor system during hover or slow flight is expressed as Equation (36):

\[
\begin{align*}
\dot{x} &= (\sin \theta \cos \phi \cos \psi + \sin \phi \sin \psi) \sum_{i=1}^{4} F_{bi}^b - K_{Dx} x \\
\dot{y} &= (\sin \theta \cos \phi \sin \psi - \sin \phi \cos \psi) \sum_{i=1}^{4} F_{bi}^b - K_{Dy} y \\
\dot{z} &= \cos \phi \sin \phi \sum_{i=1}^{4} F_{bi}^b - K_{Dz} z - mg \\
I_x \dot{w}_x &= (I_x - I_y) w_y^2 + I_z w_z^2 + lb \left( w_y^2 - w_z^2 \right) \\
I_y \dot{w}_y &= (I_y - I_x) w_x^2 - I_z w_z^2 + lb \left( w_x^2 - w_z^2 \right) \\
I_z \dot{w}_z &= (I_z - I_x) w_x w_y + d \left( -w_y^2 + w_z^2 - w_z^2 + w_z^2 \right) \\
x &= u \cos \theta \cos \psi + v \left( \sin \theta \cos \phi \cos \psi - \cos \phi \sin \psi \right) + w \left( \sin \theta \cos \phi \sin \psi + \cos \phi \sin \psi \right) \\
y &= u \cos \theta \sin \psi + v \left( \sin \theta \sin \phi \sin \psi - \cos \phi \cos \psi \right) + w \left( \sin \theta \cos \phi \sin \psi - \sin \phi \sin \psi \right) \\
z &= -u \sin \theta + v \sin \phi \cos \theta + w \cos \phi \cos \theta \\
\phi &= w_x + (w_z \cos \phi + w_y \sin \phi) \tan \theta \\
\theta &= w_y \cos \phi - w_z \sin \phi \\
\psi &= \frac{1}{\cos \theta} (w_z \cos \phi + w_y \sin \phi)
\end{align*}
\]

**2.2.6 Model Simplification**

The derived nonlinear mathematical model considers multiple physical effects. For the sake of research convenience, we assume that air resistance can be ignored. The moment of inertia of an object is related to the mass and volume of the object. Given the small volume and light weight of the quadrotor drone, its moment of inertia is small. For ease of analysis, the gyroscopic effects are also neglected.

Assuming that the pitch and roll angles of quadrotor are small, and rotation speed is also small, the system dynamics equations can be transformed into a standard unit matrix:

\[
\begin{bmatrix}
\dot{\phi} \\
\dot{\theta} \\
\dot{\psi}
\end{bmatrix} = \begin{bmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 1
\end{bmatrix} \begin{bmatrix}
\phi \\
\theta \\
\psi
\end{bmatrix}
\]

The driving force is described in terms of the rotor speed, that is, the power provided by the motor to the system is represented by the square sum of the rotor speeds. Considering the rotor speeds as input control variables,
the four virtual control input variables can be defined by Equation (38):

\[
\begin{align*}
U_i &= \sum_{j=1}^{3} F_{ij}^b = b \left( w_i^1 + w_i^2 + w_i^3 + w_i^4 \right) \\
U_2 &= l \left( F_{21}^b - F_{22}^b \right) = lb \left( w_2^1 - w_2^2 \right) \\
U_3 &= l \left( F_{31}^b - F_{32}^b \right) = lb \left( w_3^1 - w_3^2 \right) \\
U_4 &= -M_{D1}^b + M_{D2}^b - M_{D3}^b + M_{D4}^b = d \left( -w_4^1 + w_4^2 - w_4^3 + w_4^4 \right)
\end{align*}
\]

(38)

Assuming that the quadrotor structure is very symmetrical, and neglecting air resistance influence while performing small-angle motion, its nonlinear model can be simplified to Equation (39)[4].

\[
\begin{align*}
\phi &= U_1 \\
\theta &= U_3 \\
\psi &= U_4 \\
x &= \frac{U_1}{m} \left( \sin \theta \cos \phi \cos \psi + \sin \phi \sin \psi \right) \\
y &= \frac{U_1}{m} \left( \sin \theta \cos \phi \sin \psi - \sin \phi \cos \psi \right) \\
z &= \frac{U_1}{m} \cos \theta \cos \phi - g
\end{align*}
\]

(39)

Note: Here, \( x, y, z \) (displacements in the body coordinates) are the second derivatives of \( x, y, z \), which represent the accelerations in the three axes of the coordinate system.

It can be seen that, \( x, y, z \) are related only to \( U_1 \), \( \phi \) is related to \( U_2 \), \( \theta \) is related to \( U_3 \), and \( \psi \) is related to \( U_4 \).

The virtual control input variables defined by Equation (38) can be represented in matrix form as shown in Equation (40):

\[
\begin{bmatrix}
U_1 \\
U_2 \\
U_3 \\
U_4
\end{bmatrix}
= \begin{bmatrix}
bw_1^2 + bw_2^2 + bw_3^2 + bw_4^2 \\
lbw_2^2 - lbw_2^2 \\
lbw_2^2 - lbw_2^2 \\
-dw_2^2 + dw_3^2 + dw_4^2
\end{bmatrix}
\begin{bmatrix}
b & b & b \\
0 -lb & 0 & lb \\
-lb & 0 & 0 \\
-d & -d & d
\end{bmatrix}
\begin{bmatrix}
w_1^2 \\
w_2^2 \\
w_3^2 \\
w_4^2
\end{bmatrix}

(40)

Typically, the forces \( U_i \) can be obtained based on the desired attitude. However, in the control program, it is necessary to calculate the speed control variable \( w_i^2 \), for each motor, thus requiring a transformation.

\[
\begin{bmatrix}
w_1^2 \\
w_2^2 \\
w_3^2 \\
w_4^2
\end{bmatrix}
= \begin{bmatrix}
b & b & b \\
0 -lb & 0 & lb \\
-lb & 0 & 0 \\
-d & -d & d
\end{bmatrix}^{-1}
\begin{bmatrix}
U_1 \\
U_2 \\
U_3 \\
U_4
\end{bmatrix}
= \frac{1}{4}
\begin{bmatrix}
\frac{1}{b} & 0 & -\frac{2}{lb} & -\frac{1}{d} \\
\frac{1}{b} & 0 & \frac{2}{lb} & -\frac{1}{d} \\
0 & 1 & 0 & 1 \\
1 & 0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
U_1 \\
U_2 \\
U_3 \\
U_4
\end{bmatrix}

(41)

\]

**Chapter 3: Design of the PID Controller**

**3.1 Introduction to PID Algorithm**

The PID controller has a simple structure, stable performance, easy adjustment, and reliable operation, making it widely applied in practical engineering scenarios[5].

![Figure 3.1: Structure of PID Controller](image)

In this experiment, the application principle of the PID controller in the quadrotor is as follows: An initial value is assumed and the current attitude angle data is obtained through the calculator, which is then summed with the initial value. By continuously adjusting the PID parameters, the attitude angle stability of quadrotor drone is increased, enabling stable flight. The expression for the PID algorithm is:

\[
U(t) = K_p e(t) + K_i \int_0^t e(\tau)d\tau + K_d \frac{d}{dt} e(t)
\]

(42)

The meanings of the various characters are shown in Table 3.1:
Generally, the PID controller was considered as a filter in the frequency domain system. Based on this property, it is used to control the device.

### 3.2 Controller Design and Simulation

The attitude control of the quadrotor consists of two control loops: inner and outer loops. Observing the simplified mathematical model, the change in attitude angle affects position change. The position control is treated as the outer loop, and attitude control is inner loop \[\text{[6]}\]. Observing its mathematical model, there are four input variables and six output variables, constituting an underactuated system. The variables are mutually influential, indicating coupling relationships \[\text{[6]}\]. The system diagram of the quadrotor’s position control and attitude angle control is shown in Figure 3.2.

**Figure 3.2 Block Diagram of Attitude Angle and Position Control System**

#### 3.2.1 Design of Position Loop Controller

Let \([x \ y \ z]\) be the given body coordinates, and the feedback position coordinates are the double integral of acceleration \([\ddot{x} \ \ddot{y} \ \ddot{z}]\) calculated through the model:

\[
\begin{align*}
\begin{bmatrix}
\ddot{x} \\
\ddot{y} \\
\ddot{z}
\end{bmatrix} &=
\begin{bmatrix}
U_1 \\
U_2 \\
U_3 \\
U_4
\end{bmatrix}
\end{align*}
\]

Let \(\theta, \phi, \psi\) be known quantities, thus:

\[
\begin{align*}
\begin{bmatrix}
\ddot{x} \\
\ddot{y} \\
\ddot{z}
\end{bmatrix} &=
\begin{bmatrix}
\sin \theta \cos \phi \cos \psi + \sin \phi \sin \psi \\
-\sin \theta \cos \phi \sin \psi - \sin \phi \cos \psi \\
\cos \theta \cos \phi - \sin \phi
g
\end{bmatrix}
\end{align*}
\]

After researching, the relevant parameters for the quadrotor UAV are collected as shown in Table 3.1.
Table 3.1 Quadrotor UAV Flight Parameters

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Unit</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Body mass</td>
<td>kg</td>
<td>1.485</td>
</tr>
<tr>
<td>Rotor lift coefficient</td>
<td>$N*S^2$</td>
<td>3.15e-5</td>
</tr>
<tr>
<td>Rotor drag coefficient</td>
<td>$N<em>m</em>S^2$</td>
<td>7.8e-7</td>
</tr>
<tr>
<td>Distance from motor to center</td>
<td>m</td>
<td>0.5</td>
</tr>
<tr>
<td>Rotational inertia to x-axis</td>
<td>$kg*m^2$</td>
<td>2.453e-3</td>
</tr>
<tr>
<td>Rotational inertia to y-axis</td>
<td>$kg*m^2$</td>
<td>2.453e-3</td>
</tr>
<tr>
<td>Rotational inertia to z-axis</td>
<td>$kg*m^2$</td>
<td>5.386e-2</td>
</tr>
</tbody>
</table>

Given the known quantities $x, y, z,$ and yaw angle $\psi$, we combine the above equations to calculate the roll angle $\phi$ and pitch angle $\theta$. Here we construct pseudo control variables as shown below:

\[
\begin{align*}
U_x &= K_p e_x + K_d \int e_x \, dt + K_i \int e_x \, dt + x \\
U_y &= K_p e_y + K_d \int e_y \, dt + K_i \int e_y \, dt + y \\
U_z &= K_p e_z + K_d \int e_z \, dt + K_i \int e_z \, dt + z
\end{align*}
\]

The block diagram of position controller model was exhibited in Figure 3-3. Here, $x_c, y_c, z_c$ correspond to the input ports of the given position quantities $x, y, z$. $x_g, y_g, z_g$ are the input ports for feedback positions $x_c, y_c, z_c$, $\text{ang}_z$ is the input port for yaw angle $\psi$, $\text{ang}_x, \text{ang}_y$ are the output ports for roll angle $\phi$ and pitch angle $\theta$, $U_1$ is the output port for the system’s virtual control input $U_1$.

![Figure 3.3 Position Controller](image-url)
The internal structure can be seen below.

![Diagram](image1.png)

**Figure 3.4**

**Subsystem**

3.2.2 Design of Attitude Loop Controller

Similar to the attitude control design, let $\phi, \theta, \psi$ be the given attitude angles. $\dot{\phi}, \dot{\theta}, \dot{\psi}$ are the feedback attitude angle accelerations, which after integration yield the feedback attitude angle values. According to the adopted control method, we construct pseudo control variables as follows:

\[
\begin{align*}
\phi &= \frac{U_2}{I_x} \\
\theta &= \frac{U_3}{I_y} \\
\psi &= \frac{U_4}{I_z}
\end{align*}
\]

The corresponding motor speed is

\[
\begin{pmatrix}
\omega_1^2 \\
\omega_2^2 \\
\omega_3^2 \\
\omega_4^2
\end{pmatrix} = \frac{1}{4} \begin{pmatrix}
\frac{1}{b} & 0 & -\frac{2}{bl} & -\frac{1}{d} \\
\frac{1}{b} & \frac{-2}{bl} & 0 & 1 \\
\frac{1}{b} & 0 & \frac{2}{bl} & -\frac{1}{d} \\
\frac{1}{b} & \frac{2}{bl} & 0 & 1
\end{pmatrix} \begin{pmatrix}
U_1 \\
U_2 \\
U_3 \\
U_4
\end{pmatrix}
\]

Here, we construct pseudo control variables and let $l=b=d=1$. Then, the motor speed is:

\[
\begin{pmatrix}
\omega_1^2 \\
\omega_2^2 \\
\omega_3^2 \\
\omega_4^2
\end{pmatrix} = \frac{1}{4} \begin{pmatrix}
1 & 0 & -2 & -1 \\
1 & -2 & 0 & 1 \\
1 & 0 & 2 & -1 \\
1 & 2 & 0 & 1
\end{pmatrix} \begin{pmatrix}
U_1 \\
U_2 \\
U_3 \\
U_4
\end{pmatrix}
\]

Based on this, the block diagram of the attitude control loop model was set up as Figure 3.4.

![Diagram](image2.png)

**Figure 3.4 Attitude Controller**

The attitude control module and motor conversion module (the rotor subsystem) are used to input the motor speeds into the quadrotor UAV system model. The overall block diagram of the model encapsulates the outer loop position , the inner loop attitude control model, the quadrotor UAV system model, as Figure 3.5.
The internal structure of that can be seen from Figure 3.6. Its role is to minimize the impact of noise and output error.

3.3 Method for Adjusting PID Parameters

The key to PID regulation lies in the tuning of its parameters. A commonly used tuning method is the “Quad Axis” method. After the PID controller model is well-constructed, it uses cascading PID with inner and outer loops, which embodies the stability and the response speed of drone, respectively. First, adjust the inner loop for stability. In conditions without oscillations, the P value is positively related to stability. If slight oscillations appear during the adjustment, the P value is generally proper, the D term is added for suppression. These two variables need to be coordinated, otherwise, it’s hard to realize stable effect. When adjusting the outer loop, don’t arbitrarily modify the parameters. Identify the issue based on the symptoms and then adjust the parameters to gradually achieve a stable effect.

In the initial design of this controller, the PID parameters were exhibited in Table 3.2. After optimization, that are as presented in Table 3.3.

Table 3.2 Initial PID Parameters for Quadrotor Controller(1)

<table>
<thead>
<tr>
<th>Channel</th>
<th>Proportional (P)</th>
<th>Integral (I)</th>
<th>Derivative (D)</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>100000</td>
<td>0</td>
<td>100000</td>
</tr>
<tr>
<td>y</td>
<td>28000</td>
<td>500</td>
<td>33000</td>
</tr>
<tr>
<td>z</td>
<td>160000</td>
<td>100000</td>
<td>100000</td>
</tr>
<tr>
<td>φ</td>
<td>100</td>
<td>2</td>
<td>200</td>
</tr>
<tr>
<td>θ</td>
<td>100</td>
<td>3</td>
<td>200</td>
</tr>
<tr>
<td>ψ</td>
<td>10000</td>
<td>1000</td>
<td>100000</td>
</tr>
</tbody>
</table>

3.4 Experimental Results Analysis

By comparing graphs of x, y, z position response curves in Figure 3.7 and the curves in Figure 3.8, a conclusion can be obtained that the simulated results stabilize after 4 seconds for every degree of freedom. The simulation results prove that this model is very suitable for quadrotor drone, verifying the reliability of PID controller. Since there are 6 PID controllers and there exist coupling among these six degrees of freedom, parameter tuning is somewhat challenging.
From the y-position1 curve in Figure 3.7, it can be seen that the curve’s stability is not very good. Subsequent adjustments were made to the PID parameters of the \( y \), \( z \), and \( \theta \) channels, while the PID parameters for the other channels remained unchanged. The comparison graphs of the PID position control simulation and the PID attitude control simulation after modifying the PID parameters were exhibited in Figure 3.7 and 3.8. The optimized curves show better stability, and after 4 seconds, the output of all six degrees of freedom basically remains stable. The optimized PID parameters as Table 3.3.
Table 3.3 Optimized PID Parameters for Quadrotor Controller(2)

<table>
<thead>
<tr>
<th>Channel</th>
<th>Proportional (P)</th>
<th>Integral (I)</th>
<th>Derivative (D)</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>100000</td>
<td>0</td>
<td>100000</td>
</tr>
<tr>
<td>y</td>
<td>40000</td>
<td>500</td>
<td>60000</td>
</tr>
<tr>
<td>z</td>
<td>165000</td>
<td>100000</td>
<td>100000</td>
</tr>
<tr>
<td>ϕ</td>
<td>100</td>
<td>2</td>
<td>200</td>
</tr>
<tr>
<td>θ</td>
<td>100</td>
<td>0.25</td>
<td>200</td>
</tr>
<tr>
<td>ψ</td>
<td>10000</td>
<td>1000</td>
<td>100000</td>
</tr>
</tbody>
</table>

Conclusion

During the preparation of this paper, a dynamic analysis was performed, and the dynamic equations for the quadrotor were established. These dynamic equations were then simplified for practical application. Based on the established model, a simulation model of the dynamic equations was built in MATLAB/Simulink using PID control methods. Continuous adjustments were made to the PID parameters to achieve stable control results. Through the design of a quadrotor UAV controller using PID control methods, a deeper understanding of MATLAB software was achieved. This work also enhanced the ability to write functions in MATLAB, strengthened the application of the Simulink toolbox, and solidified the capability to build Simulink models based on mathematical equations.

References

[1] [Online resource available at http://www.sohu.com/a/69928848_360440]