Advances in Computer, Signals and Systems (2024)
Clausius Scientific Press, Canada

DOI: 10.23977/acss.2024.080619
ISSN 2371-8838 Vol. 8 Num. 6

Research on Visual Flight of Cascade Control of Quadrotor UAV Based on Coordinate Transformation

Fang Zeping[a,*], Guo Chengquan[b], Liu Wenyuan[c]

School of Automation and Electrical Engineering, Zhongyuan Institute of Technology, Zhengzhou, China

[a]3943@zut.edu.cn, [b]420873751@qq.com, [c]2926611914@qq.com

[*]Corresponding author

Keywords: Quadrotor UAV, Visualization, Cascade control, Backstepping control, PID, Simscape

Abstract: Aiming at the problem that it is difficult for quadrotor UAV to directly observe the real-time flight posture and debug the flight control algorithm when flying at high altitude, the structural relationship and motion analysis of quadrotor UAV are established based on coordinate transformation theory, and the cascade control algorithm of PID and backstepping controller is adopted, and the flight control of quadrotor UAV is simulated and verified visually by Solidworks and Matlab/Simscape. In order to analyze and compare, the body model of quadrotor UAV is built under Simulink and Simscape respectively, and an 8-shaped expected flight trajectory based on trigonometric function is designed. The experimental results show that the quadrotor UAV body constructed by Simulink and Simscape respectively has the same flight effect under the control of the same algorithm. Generally, only numerical simulation results are obtained under Simulink. Under Simscape, not only numerical results can be obtained, but also 3D visual simulation results can be given. The flight ability of quadrotor UAV is related to the control algorithm and the expected tracking trajectory, and the expected control goal can be achieved by selecting different control algorithms or adjusting the parameters of the control algorithm. The tracking trajectory flight effect of quadrotor UAV is closely related to the response time. The control algorithm with shorter response time has better tracking effect. The response time of the control algorithm in this paper is short, and the tracking effect is good. The simulation system in this paper is intuitive, accurate and reliable, and can study and test the control algorithm well, which is of great value to the research of quadrotor UAV and control algorithm.

1. Introduction

The flight control algorithm of quadrotor UAV is one of the key contents of the research. Many control algorithms are applied to quadrotor UAV, such as PID control, sliding mode control, neural network control, backstepping, fuzzy control, fractional order and so on [1,2]. In recent years, the research on control algorithm of quadrotor UAV has changed from single algorithm to compound algorithm, and compound algorithm has become a research focus. In this paper, the cascade control strategy of PID and backstepping control method is adopted, the inner loop adopts backstepping method as the attitude angle control of the aircraft, and the outer loop adopts PID algorithm to control the position of the UAV. Because it is one of the main research contents and problems to determine the control parameters and adjust the parameter values of PID algorithm and backstepping control method, researchers need to design appropriate debugging platforms and methods to improve the development efficiency and reduce the development cost of the control algorithm of quadrotor UAV[3]. Therefore, a nonlinear quadrotor UAV simulation environment is needed to further verify the control algorithm. Yin Qiang and others established the visualization system of quadrotor UAV by combining FlightGear simulation software and MATLAB/Simulink. Zhou Dexin et al. designed an all-digital simulation platform for verifying flight control system by C++ language, and the visual display window was displayed by Primary Flight Display interface in civil aircraft [1]. The three-dimensional visual simulation system in reference [1] can not only simulate with numerical data, but also give the real-time pose (position and attitude) of UAV intuitively. Reference [4] mentioned several popular simulation environments of quadrotor UAV, such as AirSim and Gazebo. However, they are not based on MATLAB, and control researchers are more familiar with MATLAB. Although the simulation environment of quadrotor UAV based on Gazebo can be connected to MATLAB using the robot system toolbox, it needs complicated operations. Robotics and Perception Group provides a simulation environment based on Gazebo. However, for control researchers who are used to using MATLAB, they need to pay extra effort to learn and use Gazebo environment.

In this paper, MATLAB/Simscape is used to model the quadrotor UAV. Based on the coordinate transformation theory, the structural relationship and motion analysis of quadrotor UAV are established. The cascade control algorithm of PID and backstepping controller is adopted, and the flight control of quadrotor UAV is visually simulated and verified by Solidworks and MATLAB/Simscape.

2. Mathematical Model of Quadrotor UAV

The quadrotor UAV is a helicopter configuration, with four motors installed at both ends of two arms that are symmetrically crossed, as shown in Figure 1. Figure 1 shows the architecture of "X" UAV. Two opposite motors on the same arm rotate in the same direction, and another pair of adjacent motors rotate in opposite directions [5].

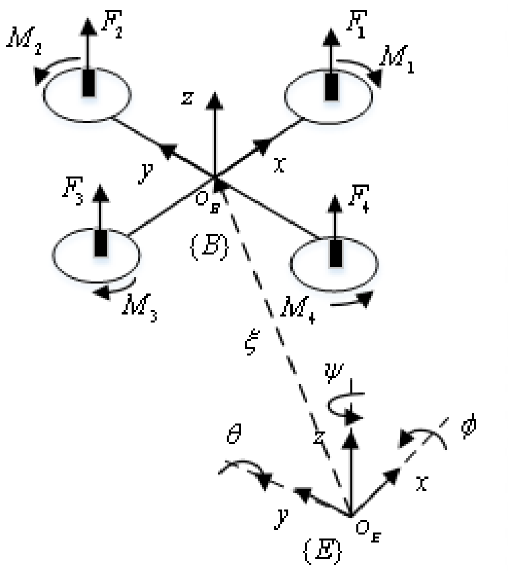


Figure 1: Structure and flight principle of quadrotor UAV.

Based on Newton's law of translation and rotation, we establish the mathematical model of quadrotor UAV. In equation (1), ϕ, θ, ψ are the roll angle, pitch angle and yaw angle of the quadrotor UAV respectively. $\dot{\phi}$, $\dot{\theta}$, $\dot{\psi}$ are the roll angle velocity, pitch angle velocity and yaw angle velocity respectively. x, y, z are x direction position, y direction position and height respectively. $\dot{x}$, $\dot{y}$, $\dot{z}$ are the x direction, the y direction and the speed of the z direction respectively. m is the quality of quadrotor UAV. g is the acceleration of gravity.

$$\begin{aligned}
&\ddot{x} = [(\cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi)U_z - k_{dx}\dot{x}]/m \\
&\ddot{y} = [(\cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi)U_z - k_{dy}\dot{y}]/m \\
&\ddot{z} = [(\cos\phi\cos\theta)U_z - k_{dz}\dot{z} - mg]/m \\
&\ddot{\phi} = [(I_y - I_z)\dot{\theta}\dot{\psi} + U_\phi - k_{d\phi}\dot{\phi}^2 - I_r\bar{\Omega}\dot{\theta}]/I_x \\
&\ddot{\theta} = [(I_z - I_x)\dot{\phi}\dot{\psi} + U_\theta - k_{d\theta}\dot{\theta}^2 + I_r\bar{\Omega}\dot{\phi}]/I_y \\
&\ddot{\psi} = [(I_x - I_y)\dot{\phi}\dot{\theta} + U_\psi - k_{d\psi}\dot{\psi}^2]/I_z
\end{aligned} \tag{1}$$

In equation (1), k_{dx}, k_{dy} and k_{dz} are the translational resistance coefficients in the direction of, x, y and z respectively. $k_{d\phi}$, $k_{d\theta}$, $k_{d\psi}$ are aerodynamic coefficients. I_x, I_y, I_z are the rotational inertia of the quadrotor UAV around the axis x, y, z, respectively. I_r is the moment of inertia of the motor.

In equation (1), $\bar{\Omega}$ is the algebraic sum of angular velocities of four propellers.

$$\bar{\Omega} = -\Omega_1 + \Omega_2 - \Omega_3 + \Omega_4 \tag{2}$$

In equation (2), Ω_1, Ω_2, Ω_3 and Ω_4 are the angular velocities of the four propellers respectively.

3. Cascade Control Strategy of Quadrotor UAV Flight

In this paper, the two-stage cascade control strategy of PID and backstepping control algorithm is adopted, as shown in Figure 2. Its main goal is to control the position { $x(t)$, $y(t)$, $z(t)$ } and attitude { $\phi(t)$, $\theta(t)$, $\psi(t)$ } of quadrotor UAV to track to the expected value { $x_d(t)$, $y_d(t)$, $z_d(t)$, $\phi_d(t)$, $\theta_d(t)$, $\psi_d(t)$ }. The first stage control loop adopts PID algorithm, and calculates the expected roll angle ϕ_d and pitch angle θ_d for the second stage control loop according to the expected x_d and y_d. The second stage control loop adopts the backstepping control algorithm to calculate the corresponding thrust of the required altitude and attitude, and then gives the control input of the system U_z, U_ϕ, U_θ and U_ψ [5]. In this paper, $x_d(t)$, $y_d(t)$, $z_d(t)$ are an 8-shaped expected flight trajectory values in the direction of, x, y and z or hovering flight altitude respectively.

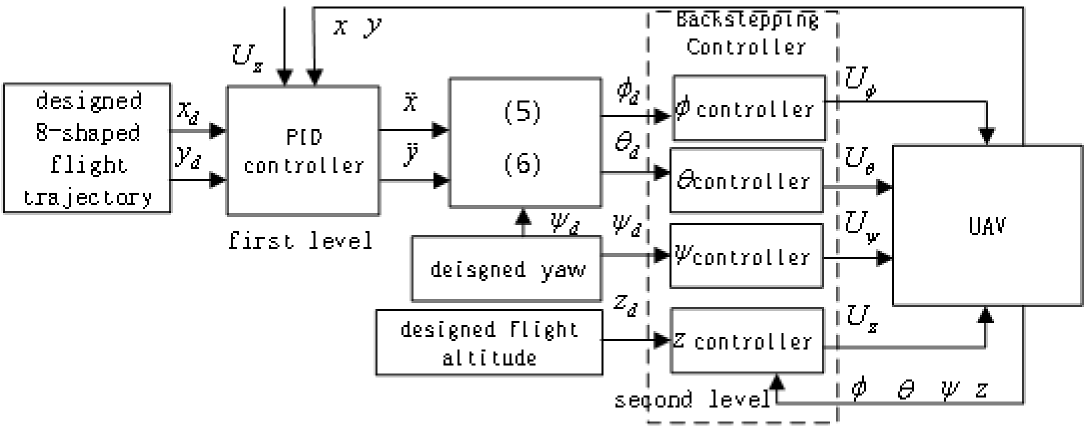


Figure 2: The overall scheme of cascade control strategy of PID and backstepping control.

3.1. Design of Backstepping Controller

In this section, the backstepping control algorithm is applied to the quadrotor UAV model established in the previous section. It is not accidental to choose this algorithm. The main advantages of this algorithm are: it ensures the stability of Lyapunov, it ensures the robustness and all properties of the required dynamics, and it ensures that all system nonlinearities are handled [6].

According to the backstepping control algorithm, U_z, U_ϕ, U_θ and U_ψ are

$$\begin{aligned} U_z &= m(g + k_{dz}x_6 - \ddot{x}_{5d} - p_5\dot{e}_5 - e_5 + p_6 e_6)/(\cos x_7 \cos x_9) \\ U_\phi &= (a_2 x_{10}\bar{\Omega} - a_1 x_{10} x_{12} + k_{d\phi} x_8 + \ddot{x}_{7d} + p_7 \dot{e}_7 + e_7 - p_8 e_8) a_6 \\ U_\theta &= (-a_4 x_8 \bar{\Omega} - a_3 x_8 x_{12} + k_{d\theta} x_{10} \ddot{x}_{9d} + p_9 \dot{e}_9 + e_9 - p_{10} e_{10}) / a_7 \end{aligned} \tag{3}$$

$$U_\psi = (a_5 x_8 x_{10} + k_{d\psi} x_{12} + \dot{x}_{11d} + p_{11}\dot{e}_{11} + e_{11} - p_{12}e_{12}) / a_8$$

In equation (3), $a_1 = (I_y - I_z)/I_x$, $a_2 = I_r/I_x$, $a_3 = (I_z - I_x)/I_y$, $a_4 = I_r/I_y$, $a_5 = (I_x - I_y)/I_z$, $a_6 = l/I_x$, $a_7 = l/I_y$, $a_8 = 1/I_z$.

In equation (4), e_i is the position and attitude tracking deviation, and its expression is

$$e_i = \begin{cases} x_{id} - x_i & i \in \{1,3,5,7,9,11\} \\ x_i - x_{(i-1)d} - p_{(i-1)}e_{(i-1)} & i \in \{2,4,6,8,10,12\} \end{cases} \tag{4}$$

wherein, $\forall i \in [1,12]$, satisfy $p_i > 0$. x_{id}, $x_{(i-1)d}$ respectively, are state variables and expected values.

p_1 - p_8 are undetermined parameters for the inversion control algorithm. There are many undetermined parameters, which bring many difficulties to the flight control debugging of quadrotor UAV. Therefore, it is necessary to design a debugging platform for the flight control system of quadrotor UAV.

3.2. PID Controller Design

The position subsystem (x and y direction) of quadrotor UAV is the underactuated part of its dynamics system. The quadrotor UAV must move in x and y direction through its roll angle ϕ and pitch angle θ changes [6].

Since the stable roll and pitch controller equation (3) has been established, the system can be intuitively reacted by adding controllers, so that the roll and pitch dynamics can be automatically adjusted to drive the quadrotor UAV to the required x and y positions. PID controller is used as the outer loop control subsystem.

ϕ_d and θ_d are the expected values of the roll angle and pitch angle subsystems of the controller are obtained. ϕ_d and θ_d are respectively

$$\begin{bmatrix} \phi_d \\ \theta_d \end{bmatrix} = \frac{m}{U_z} \begin{bmatrix} -\sin\psi & \cos\psi \\ -\cos\psi & -\sin\psi \end{bmatrix} \begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix} \tag{5}$$

The $\ddot{x}$ and $\ddot{y}$ in equation (5) are obtained by a PID controller, they are

$$\begin{aligned} \ddot{x} &= k_{px}(x_r - x) + k_{dx}(\dot{x}_r - \dot{x}) \\ \ddot{y} &= k_{py}(y_r - y) + k_{dy}(\dot{y}_r - \dot{y}) \end{aligned} \tag{6}$$

In equation (6), k_{px} and k_{dx} are the proportional parameter and differential parameter of the direction x, respectively. k_{py} and k_{dy} are the proportional parameters and differential parameters of the direction y, respectively.

4. Flight Modeling Based on Simscape

4.1. Coordinate Transformation Principle

Coordinate system selection and transformation are the mathematical foundations and key contents of motion analysis and control for unmanned aerial vehicles. Multiple coordinate transformations are required between the coordinate systems of various components of quadrotor UAV [7].

4.1.1. Pose (position and posture)

The pose (position and posture) description diagram is shown in Figure 3. Given a point p in a coordinate system $\{A\}$, the position vector ${}^{A}p$ is usually used to determine the position of the point in space.

$$
{}^{A}p = [p_x, p_y, p_z]^T
$$

Where, p_x, p_y, p_z are the three coordinate components of a point p in the coordinate system {A}.

In order to describe the orientation of each object, it is necessary to define a coordinate system fixed to this object. As shown in Figure 3, the orientation of space object B can be described by a 3×3 rotation matrix ${}^{A}_{B}R$ composed of the direction cosine of three unit principal vectors fixed to the coordinate system {B} relative to the reference coordinate system {A}.

$$
{}^{A}_{B}R = \begin{bmatrix} \cos\alpha_x & \cos\alpha_y & \cos\alpha_z \\ \cos\beta_x & \cos\beta_y & \cos\beta_z \\ \cos\gamma_x & \cos\gamma_y & \cos\gamma_z \end{bmatrix}
$$

Among them, α_i, β_i and γ_i are the directional angles of the ${}^{A}x$, ${}^{A}y$ and ${}^{A}z$ axes, respectively, with $i = x$, y, z.

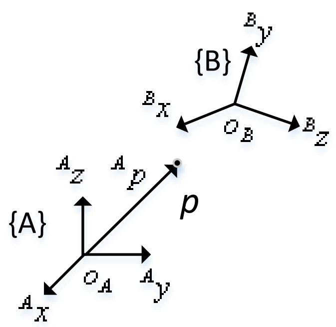


Figure 3: Position and attitude description.

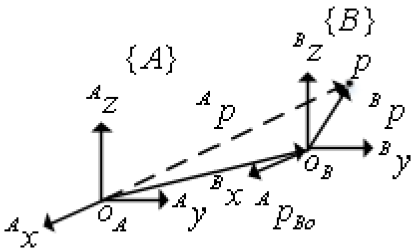
4.1.2. Translation Coordinate Transformation and Homogeneous Translation Coordinate Transformation

We describe the position and posture relationship of objects through the transformation between their coordinate systems. Coordinate transformation between one object and another usually includes translation transformation, rotation transformation and compound transformation.

As shown in Figure 4, the coordinate systems {A} and {B} have the same orientation, but their origins are not coincident. The position vector of a point p in two coordinate systems satisfies the following equation.

$$^{A}p = {}^{B}p + {}^{A}p_{Bo}$$

Where, ^{A}p, ^{B}p are the position vectors of the point p in the coordinate system {A} and {B} respectively. $^{A}p_{Bo}$ describes the position of the origin of coordinate system {B} relative to coordinate system {A}.



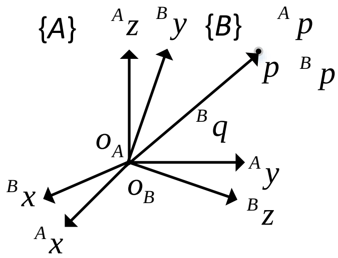


Figure 4: Translation transformation. Figure 5: Rotation transformation.

The point p is described by vector $a\vec{i} + b\vec{j} + c\vec{k}$. $\vec{i}$, $\vec{j}$, $\vec{k}$ is the unit vector on the axis x, y, z. The translation homogeneous transformation of this point p is

$$Trans(a,b,c) = \begin{bmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{7}$$

4.1.3. Rotation Coordinate Transformation and Rotation Homogeneous Coordinate Transformation

As shown in Figure 5, coordinate systems {A} and {B} have the same origin but different orientations, so the position vector relationship of point p in the two coordinate systems is

$$^{A}p = {}^{A}_{B}R\,{}^{B}p$$

$^{A}_{B}R$ is called rotation transformation matrix, which describes the orientation of coordinate system {B} relative to coordinate system {A}.

Corresponding to the rotation transformation of axes x, y and z with a rotation angle of θ, their homogeneous rotation matrices are

$$Rot(x,\theta)=\begin{bmatrix}1&0&0&0\\0&\cos\theta&-\sin\theta&0\\0&\sin\theta&\cos\theta&0\\0&0&0&1\end{bmatrix},\quad Rot(y,\theta)=\begin{bmatrix}\cos\theta&0&\sin\theta&0\\0&1&0&0\\-\sin\theta&0&\cos\theta&0\\0&0&0&1\end{bmatrix},$$

$$Rot(z,\theta)=\begin{bmatrix}\cos\theta&-\sin\theta&0&0\\\sin\theta&\cos\theta&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix} \tag{8}$$

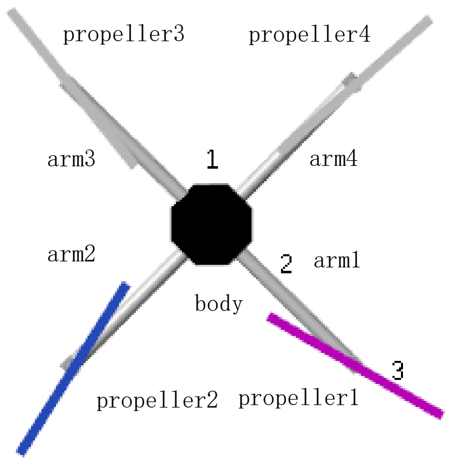
4.1.4. Coordinate Definition of Quadrotor UAV System

The selection of coordinate system is the basis of kinematics modeling of quadrotor UAV. The definitions of ground coordinate system {E} and aircraft-body coordinate system {B} are shown in Figure 1. Based on the above coordinate transformation method, the transformation from the ground coordinate system to the aircraft-body coordinate system is to rotate around their respective coordinate axes in the order of yaw angle ϕ, pitch angle θ and roll angle ψ, and the transformation matrix is

$$R_{\psi\theta\phi}=\begin{bmatrix}\cos\theta\cos\psi & \cos\theta\sin\psi & -\sin\theta\\(\sin\phi\sin\theta\cos\psi-\cos\phi\sin\psi) & (\sin\phi\sin\theta\sin\psi+\cos\phi\cos\psi) & \sin\phi\cos\theta\\(\cos\phi\sin\theta\cos\psi+\sin\phi\sin\psi) & (\cos\phi\sin\theta\sin\psi-\sin\phi\cos\psi) & \cos\phi\cos\theta\end{bmatrix} \tag{9}$$

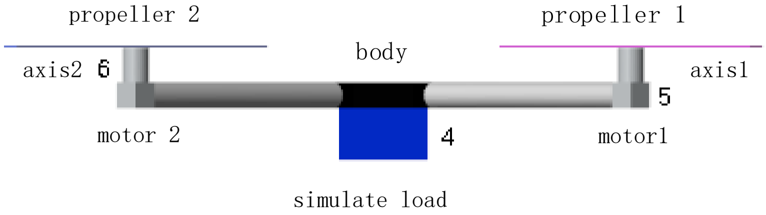
4.2. Structure Model of Quadrotor UAV

We simplify the components of real UAV, regardless of the components of flight control system such as circuit board, wires and GPS antenna, and build a simplified model of quadrotor UAV in Simscape, which mainly includes the body, four arms, four propellers, four motors, four motor shafts and a simulated load (including batteries). This model is the basis for building Simscape model, as shown in Figure 6.



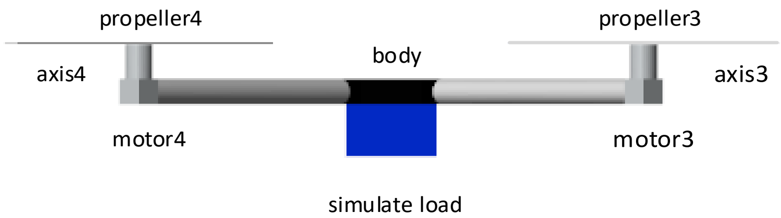
1- organism, 2- machine arm, 3- propeller

(a) Top view



4- load, 5- motor, 6- spindle

(b) Front view



(c) Rear view

Figure 6: Simplified model of quadrotor UAV.

4.3. Quadrotor UAV Flight System

The flight system of quadrotor UAV is shown in Figure 7. The system mainly consists of flight data input, flight controller, aircraft model and basic Simscape configuration. The flight data input module sets the flight altitude, the 8-shaped trajectory, the fixed-point hovering data and the flight course.

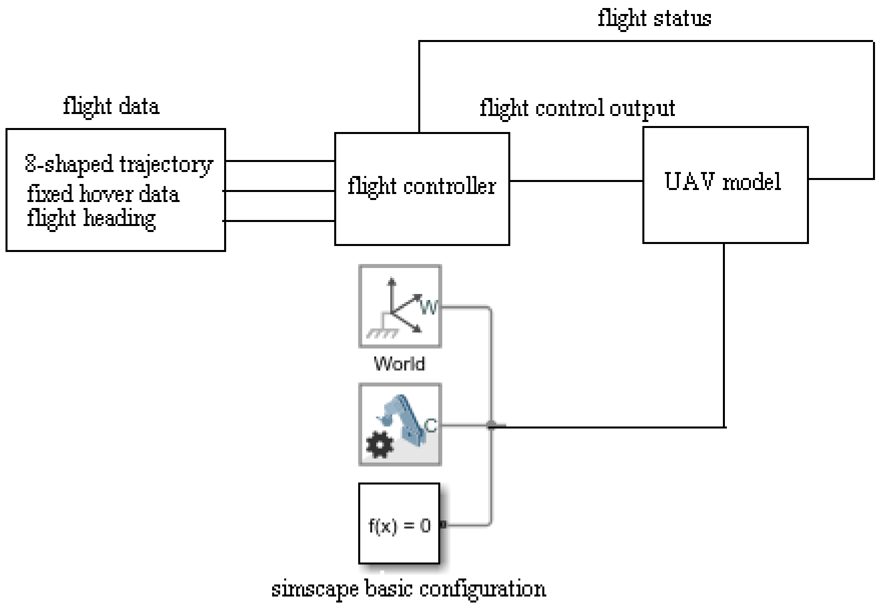


Figure 7: Structure diagram of flight system under Simscape.

We constructed a quadcopter UAV model using the Simscape module. The flight data input module and flight controller are constructed by Simulink module.

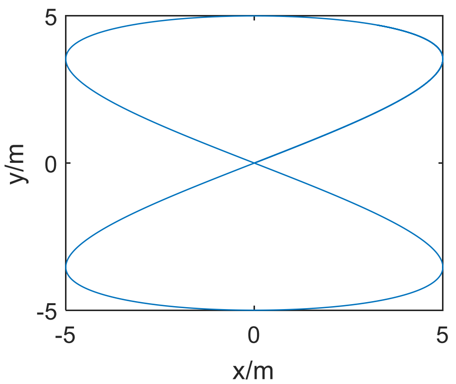
4.4. The 8-shaped Flight Trajectory

The navigation and autonomous driving of quadrotor UAV require the generation of achievable and computationally fast reference trajectories. The 8-shaped trajectory flight is one of the important contents of training and essential skills for manual control of quadrotor UAV. The 8-shaped trajectory flight in the text refers to the automatic control of the quadrotor UAV to fly along an 8-shaped trajectory in the x and y planes after reaching a certain altitude.

The 8-shaped trajectory, as shown in Figure 8, is generated by a sine function, and its expression is

$$\begin{aligned} x_d &= 5 \times \sin(t) \\ y_d &= 5 \times \sin(0.5t) \\ z_d &= const \end{aligned} \tag{10}$$

In equation (10), t is time. x_d, y_d and z_d are the expected trajectory values in the x, y, and z directions, and their values vary with simulation time. The amplitudes of x_d and y_d in equation (10) are 5.



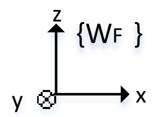


Figure 8: The expected 8-shaped trajectory. Figure 9: Front view of world coordinate system.

4.5. Qquadrotor UAV Model Based on Simscape

Simscape is modeled by physical network method, which is suitable for simulating the system composed of real physical components. In this paper, Simscape is used to build a quadrotor UAV model. According to the simplified structure and composition of the real quadrotor UAV, the corresponding modules in Simscape are used to simulate the real physical components.

Refer to Figure 7 for the simulation results based on the Simscape quadrotor UAV model. Under the given flight data, the Simscape model of quadcopter unmanned aerial vehicle is controlled by the aircraft controller to achieve the 8-shaped flight, and then the real unmanned aerial vehicle flight simulation research is conducted.

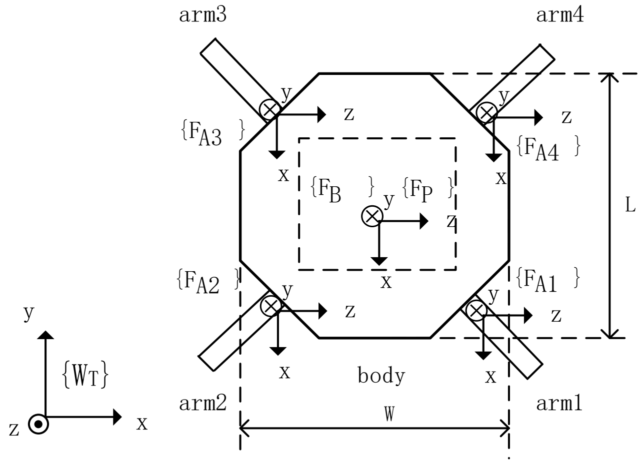
4.5.1. Coordinate System Definition

As can be seen from section 4.1, the orientation of quadrotor UAV is described by the coordinate system fixed to each component. The coordinate system of each component of quadrotor UAV is the basis of the transformation between their coordinate systems. The world coordinate system $\{W_F\}$ is an inertial reference system. As shown in Figure 9, it is the global reference system of the coordinate system of each part of the model, and it is in an absolute static state.

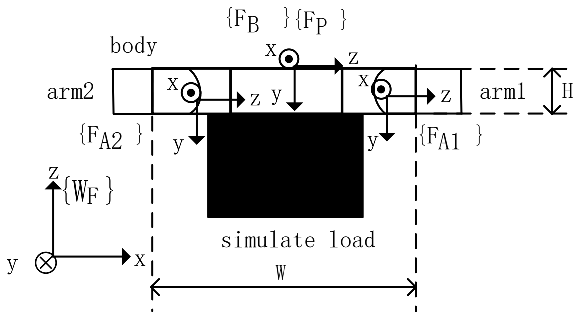
< note > the three axes of the coordinate system in Simscape are orthogonal and arranged according to the right-hand rule. In the coordinate system, in order to describe the positive directions of the three axes of the coordinate system, "$\odot$" is used to indicate that the positive direction of the axis is vertical outward, and "$\otimes$"is used to indicate that the positive direction of the axis is vertical inward.

(a) The body coordinate system

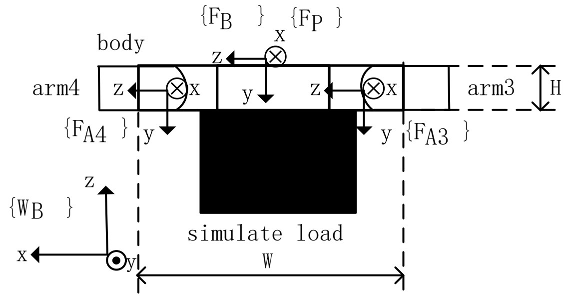
Referring to the simplified model of quadrotor UAV, its four arms and simulated load are physically connected with the body respectively. The body is an octagonal prism. Every other side surface is connected with the bottom surface of one arm, and a total of four side surfaces are connected with the bottom surfaces of four arms. Each arm is a cylinder with an upper bottom surface and a lower bottom surface, the upper bottom surface is connected with a motor, and the lower bottom surface is connected with the body. The simulated load is a cube, which has six faces: front, back, left, right, upper and lower, which are connected with the body. One end face of each arm and one end face of simulated load are physically connected with the body respectively. According to the physical connection between the body, the arm and the simulated load, in order to describe their positions and postures, respective coordinate systems are defined on each component, as shown in Figure 10. Due to the occlusion and overlap between multiple coordinate systems, it is difficult to determine the relationship between the coordinate systems from one perspective, so usually, the definition diagrams of each coordinate system from different perspectives are given. $\{W_T\}$, $\{W_F\}$ and $\{W_B\}$ are the top view, front view and back view of the world coordinate system respectively. $\{F_B\}$ is the body coordinate system, $\{F_P\}$ is the simulated load coordinate system, and $\{F_{A1}\}$ - $\{F_{A4}\}$ is the coordinate system of arm 1 - arm 4.



(a) Top view



(b) Front view



(c) Rear view

Figure 10: Various coordinate systems on the machine body.

In Figure 10(a), W and L are the width and length of the body respectively. H in Figure 10(c) is the width of the body.

Using the world coordinate system $\{W_T\}$ as the reference coordinate system, other coordinate systems can be obtained by translating a certain distance and rotating a certain angle along and around the x, y, and z axes of the world coordinate system $\{W_T\}$. For example, the orientation of the body coordinate system $\{F_B\}$ can be obtained by first rotating 90 °around the x axis of the world coordinate system $\{W_T\}$, and then rotating 90 ° around its z axis. Similar methods can be used to obtain the orientation of the simulated load coordinate system $\{F_B\}$. The coordinate systems $\{F_{A1}\}$ to $\{F_{A4}\}$ of arm 1 to 4 have the same orientation as the body coordinate system $\{F_B\}$. The poses of $\{F_B\}$ and $\{F_P\}$ are consistent.

As shown in Figure 10(a), the top view can only provide the x-axis and z-axis of $\{F_B\}$, $\{F_P\}$, $\{F_{A1}\}$-$\{F_{A4}\}$, and cannot provide their y-axis. Therefore, it is necessary to refer to Figures 10(b) and 10(c) to determine their y-axis. Therefore, referring to Figure 10, using the body coordinate system $\{F_B\}$ as the reference coordinate system, the positions of the body coordinate system $\{F_B\}$, the simulated load coordinate system $\{F_P\}$, and the coordinate systems $\{F_{A1}\}$ to $\{F_{A4}\}$ of the arm 1 to 4 can be obtained through translation transformation, as listed in Table 1. In this way, by combining translational and rotational transformations, the pose of each component on the body can be described

Table 1: Translation transformation table.

coordinate	along the x axis	along the y axis	along the z axis
Simulated load	0	0	0
arm 1	translation $\sqrt{2}L/4$	translation $H/2$	translation $\sqrt{2}L/4$
arm 2	translation $\sqrt{2}L/4$	translation $H/2$	translation $-\sqrt{2}L/4$
arm 3	translation $-\sqrt{2}L/4$	translation $H/2$	translation $-\sqrt{2}L/4$
arm 4	translation $-\sqrt{2}L/4$	translation $H/2$	translation $\sqrt{2}L/4$

According to Table 1, the transformation relationship between the simulated load, four arms and the body coordinate system is established. According to the translation transformation matrix and rotation transformation matrix, the simulated load and the position and posture of four manipulators in the body coordinate system are described. In Simscape, the Rigid Transform module is used to realize translation transformation and rotation transformation. In this module, the Axis attribute is used to set the rotation axis, and its value is a vector, and the Angle attribute is used to set the rotation angle value around the axis.

When the Axis attribute values are $[1,\ 0,\ 0]$, $[0,\ 1,\ 0]$, and $[0,\ 0,\ 1]$ respectively, rotate the Angle angle around the x-axis, y-axis, and z-axis, and their rotation transformation matrices can be calculated using equation (8). If the Axis attribute value is $[1,\ 1,\ 1]$, it is the angle of rotation of the vector around the origin to point (1, 1, 1) by Angle degrees. The Translation property is used to design the translation distance along the coordinate axis. It is a 1×3 vector that sets the translation distance along the x, y, and z axes respectively.

(b) Body model

Simscape is modeled based on the principle of physical network. In Simscape model, the connection line represents the connection between the actual physical components, and it does not need to specify the flow direction and information flow. If physical components can be connected, so can their models. Simscape simulates the physical system layout. Physical network method automatically solves all the traditional problems of variables and directionality through its penetrating variables and undirected physical connections. According to the principle of physical network and the relationship between various coordinate systems on the body, we design the Simscape model of the body, as shown in Figure 11. The B end of the rigid body transformation module is the input end of the base coordinate system, and its F end is the output end of the transformed coordinate system. Rigid body transformation module usually completes the coordinate system transformation from B end to F end. The coordinate transformation matrix from F to B is the inverse coordinate transformation matrix from B to F [8,9].

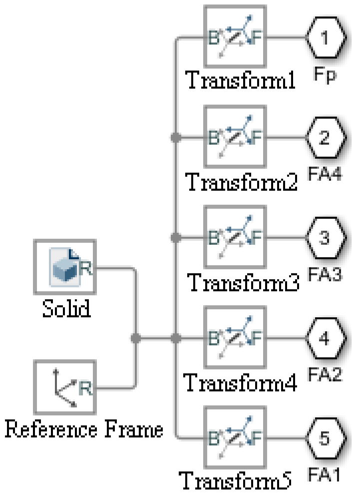


Figure 11: Model diagram of body.

(c) Coordinate system of manipulator arm 1

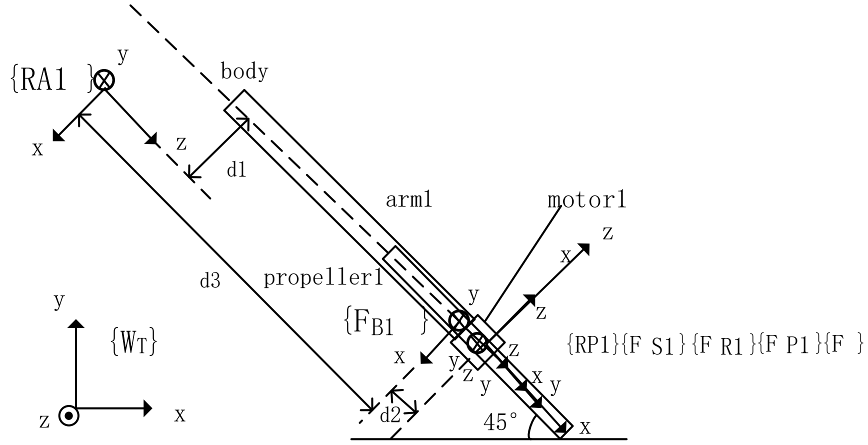
Taking one of the four arms of quadrotor UAV as an example, we construct the coordinate system of arm 1. The two ends of each arm are respectively connected with the body and the motor.

The coordinate system definition diagram of arm 1 is shown in Figure 12. In Figure 12(a), $\{W_T\}$ is the top view of the world coordinate system. From Figure 6, it can be seen that the motor 1, rotation shaft 1, and propeller 1 are vertically installed and connected from bottom to top. In Figure 12(a), it is difficult to describe the coordinate system between them. $\{RA1\}$ is the reference coordinate system for arm 1, and it is used as the reference coordinate system for the other end of arm 1, motor 1, and rotation shaft 1. $\{F_{S1}\}$ is the coordinate system on axis 1. $\{F_{R1}\}$ is the coordinate system on motor 1. $\{RP1\}$ is the reference coordinate system on propeller 1, and $\{F_{P1}\}$ is the coordinate system of propeller 1 after rigid body transformation. In Figure 12(a), $\{RP1\}$, $\{F_{S1}\}$, $\{F_{R1}\}$, and $\{F_{P1}\}$ overlap. $\{F_{B1}\}$ is the coordinate system where one end of arm 1 is connected to motor 1, and its z-axis is aligned with the direction of arm 1.

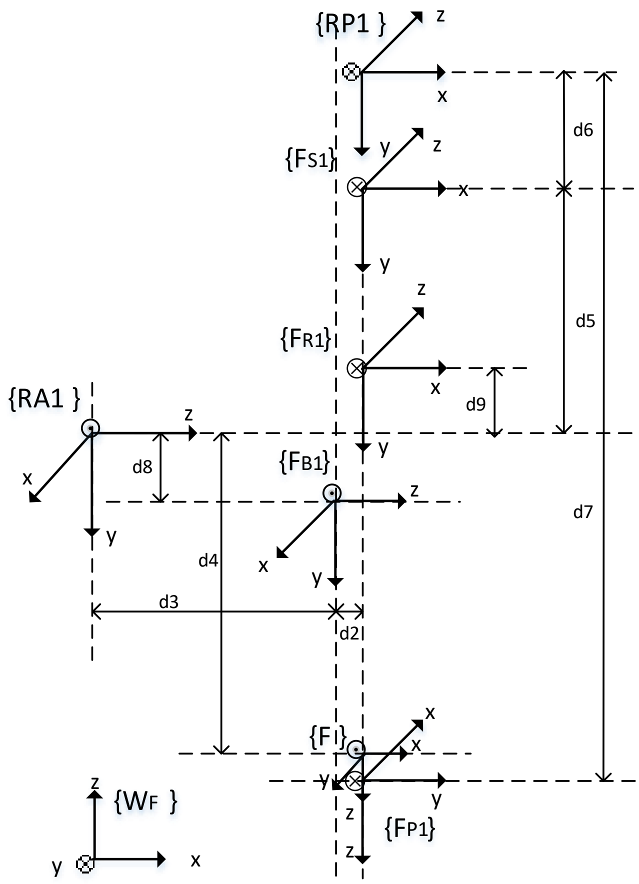
Using $\{RA1\}$ as the reference coordinate system, translate $d3$ along its z-axis and $d1$ along its negative x-axis to obtain the x-axis and z-axis of the coordinate system $\{F_{B1}\}$. Because the y-axis information cannot be seen from the top view, the translation of the y-axis cannot be determined. From Figure 12(b), it can be seen that by translating D8 distance along the y-axis of $\{RA1\}$, the y-axis of $\{F_{B1}\}$ can be determined.

$\{RP1\}$, $\{F_{S1}\}$, and $\{F_{R1}\}$ coordinate systems have the same orientation. As shown in Figure 12(a), the poses of these three coordinate systems overlap. By translating the distance d3+$d2$ along the z-axis of $\{RA1\}$, the z-axis of $\{RP1\}$, $\{F_{S1}\}$, and $\{F_{R1}\}$ can be obtained. By translating $d1$ along their negative x-axis, the x-axis of $\{RP1\}$, $\{F_{S1}\}$, and $\{F_{R1}\}$ can be obtained. However, due to pose overlap, the y-axis translation of $\{RP1\}$, $\{F_{S1}\}$, and $\{F_{R1}\}$ cannot be determined. From Figure 12(b), it can be seen that translating the $d9$ distance along the negative y-axis of $\{RA1\}$ can determine the y-axis of $\{F_{R1}\}$. Translating $d5$ distance along the negative y-axis of

$\{RA1\}$ can determine the y-axis of $\{F_{S1}\}$. By translating the negative y-axis of $\{RA1\}$ by a distance of $d5+d6$, the y-axis of $\{RP1\}$ can be determined.



(a) Top view



(b) Front view

Figure 12: Definition diagram of coordinate system of arm 1.

$\{W_F\}$ is the front view of the world coordinate system. $\{F_{R1}\}$ is the coordinate system on motor 1, with its x-axis aligned with the direction of arm 1. $\{F_{S1}\}$ is the coordinate system on axis 1, with its x-axis aligned with the direction of arm 1. $\{F_{P1}\}$ is the coordinate system on propeller 1, with its y-axis aligned with the direction of arm 1. $\{RP1\}$ is the reference coordinate system for propeller 1. The coordinate system $\{RP1\}$ is transformed to obtain the coordinate system $\{F_{P1}\}$. Using coordinate system $\{RP1\}$ as the reference coordinate system, translate $d7$ distance along its negative y-axis, first rotate 90° around its x-axis, and then rotate 90 ° around its y-axis to obtain coordinate system $\{F_{P1}\}$.

The coordinate system $\{RA1\}$ is transformed to obtain the coordinate system $\{F\}$. Translating d3+ $d2$ along the z-axis of $\{RA1\}$ yields the y-axis of $\{F\}$, while translating $d1$ along its negative x-axis yields the x-axis of $\{F_{R1}\}$. From Figure 11 (b), it can be seen that by translating $d4$ distance along the y-axis of $\{RA1\}$, the z-axis of $\{F\}$ can be determined. We rotate the x-axis of $\{RA1\}$ by 90°first, and then rotate it by 90°around its y-axis to obtain the coordinate system $\{F\}$.

To simulate propeller rotation, a Revolution Joint module is connected between coordinate system $\{F\}$ and coordinate system $\{F_{P1}\}$. Due to the fact that this rotary joint module can only rotate around its z-axis, it is necessary to ensure that the z-axis direction of coordinate system $\{F\}$ and coordinate system $\{F_{P1}\}$ are consistent. This requires performing a rigid transform on the coordinate system $\{RA1\}$ to obtain the coordinate system $\{F\}$, and similarly, performing a rigid transform on the coordinate system $\{RP1\}$ to obtain the coordinate system.

(d) Model of the manipulator arm 1

According to the coordinate definition of the robot arm 1, the Simscape model of the robot arm 1 is designed as follows.

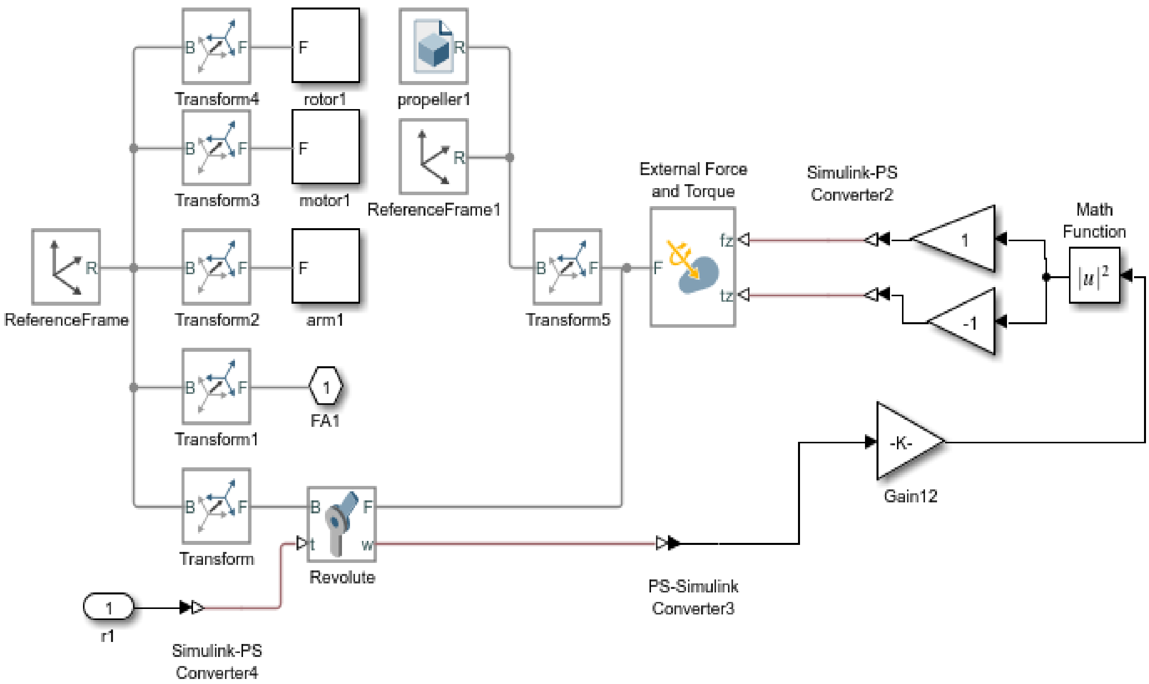


Figure 13: Model of Simscape of the arm 1.

In Figure 13, after the FA1 end is connected with the FA1 end in Figure 11, all Simscape models from the propeller 1 of the boom 1 to the body can be realized. In the same way, we can realize all the Simscape models from the propeller of the other three arms to the body, so we design and complete the Simscape model of the whole quadrotor UAV.

(e) The propeller rotate

In Figure 13, the t end of the rotary joint is a torque input port, and the w end is a rotational speed output port. The value of the torque input port determines the rotational torque of the rotating joint. According to the dynamic modeling principle of quadrotor UAV, the rotation of propeller produces upward lift.

The torque of each propeller rotating around the z axis is

$$M_i = K_d \omega_i^2$$

$i = 1,2,3,4$. K_d is the rotational resistance coefficient ($N \cdot m/(rad/s)^2$).

The lift F_i generated by the motor i is

$$F_i = K_l \omega_i^2$$

$i = 1,2,3,4$. F_i is proportional to the square of rotation angular velocity ω_i. K_l is the lift constant ($N/(rad/s)^2$).

In Simscape, the External Force and Torque module is used to provide the lift and torque of the propeller. In Figure 13, r1 is the control signal of the motor 1. S-PS module is a data conversion module from Simulink to Simscape, and PS-S module is a data conversion module from Simscape to Simulink [10].

5. Simulation Results and Analysis

In this paper, the flight simulation results based on PID and PID-backstepping control algorithm under Simulink and Simscape are given and compared.

5.1. Flight Parameters and Simulation Parameters

Parameters of quadrotor UAV, PID controller and backstepping controller are listed in Table 2.

Table 2: Parameters of quadrotor UAV and controller.

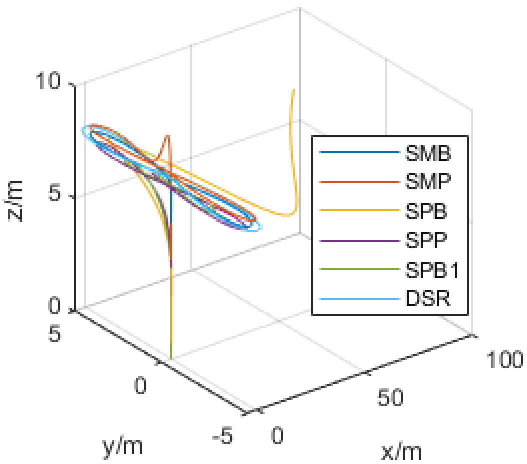
parameter	value	parameter	value	parameter	value	parameter	value
m	1.623	I_z	0.0915	K_{dy}	3.3	p_5	4.3589
g	9.81	I_r	0.006	p_1	4.3589	p_6	13.53
l	0.24	K_{px}	1.85	p_2	8.41	p_7	2.54
I_x	0.047	K_{py}	1.8	p_3	4.3589	p_8	1.2
I_y	0.047	K_{dx}	3.1	p_4	8.0	$k_{dx,y,z,\phi,\theta,\psi}$	0.3729

5.2. Flight Results of the 8-shaped Trajectory

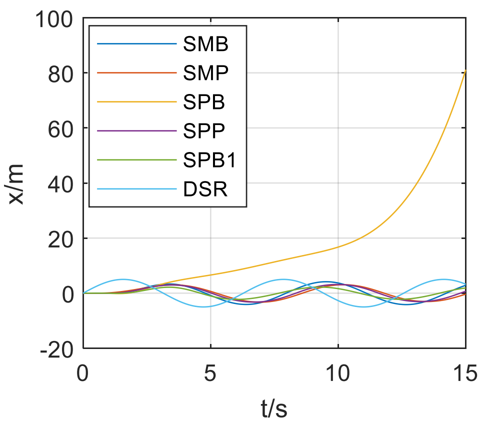
According to Equation (10), the 8-shaped flight trajectory is generated as the expected flight trajectory. The flight simulation results of the 8-shaped trajectory of quadrotor UAV are shown in Figure 14. Figure. 14(a) is a flight trajectory diagram of the 8-shaped trajectory. Figure 14 (b) to 14(d) are the 8-shaped flight trajectory diagrams in the x, y and z directions, respectively. Figure 14(e) is the 8-shaped flight trajectory diagram of xy plane. From the simulation results, the actual flight trajectory of the UAV can be divided into two stages. The first stage is that the UAV takes off from the point (0,0,0) to the desired altitude, and the second stage is to control the quadrotor UAV to fly according to the 8-shaped trajectory at the predetermined altitude. Setting the amplitudes of xd and yd in equation (9) to 3 can generate another 8-shaped expected flight trajectory. In Figure 14, SPB1 is the simulation result of a quadrotor UAV flying along this expected trajectory.

As can be seen from Figure 14, SMP curve reflects that the flying height of UAV exceeds the expected height, followed by SPB and SPP curve. SMB curve reflects that the flying height of UAV is consistent with the expected flying height. Generally speaking, it shows that all four curves can control unmanned flying to a predetermined altitude.

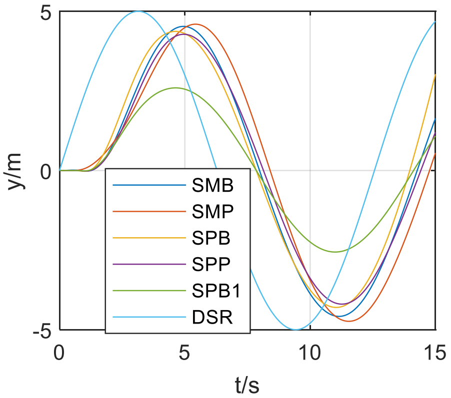
As can be seen from Figure 14(b), the SPB curve is in the x direction, and the actual flight trajectory is quite different from the expected trajectory. As can be seen from Figure 14(e), the actual flight trajectory of SPB curve is quite different from the 8-shaped expected flight trajectory, indicating that the UAV cannot be controlled to fly according to the 8-shaped expected flight trajectory. However, the error between the actual flight trajectory of SPB1 curve and the expected flight trajectory is small, which reflects that the quadrotor UAV can track the expected flight trajectory well. Therefore, the above results show that the flight results of quadrotor UAV following the expected flight trajectory are related to the geometric parameters of the expected flight trajectory.



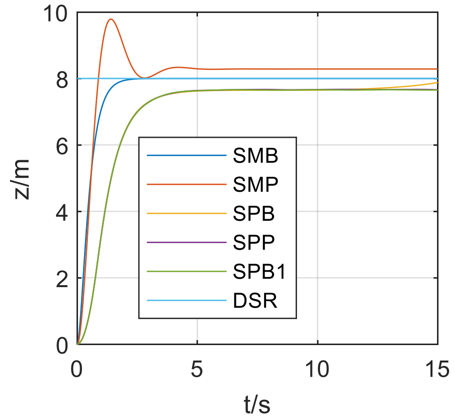
(a) The 8-shaped flight path



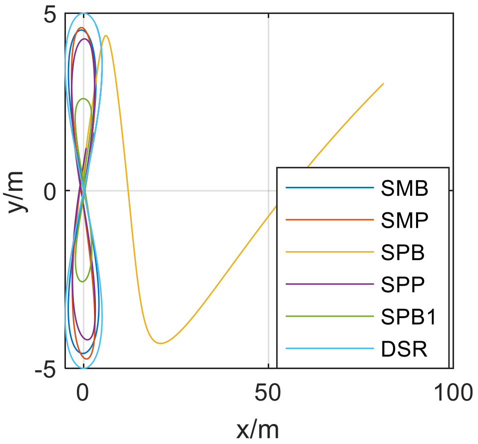
(b) The x-direction flight path



(c) The y-direction flight trajectory



(d) The z-direction flight trajectory



(e) xy plane flight path

Figure 14: Flight results of the 8-shaped trajectory.

5.3. 3D Visualization Results under Simscape

We use Simscape to generate 3D visualization results of quadrotor UAV model, as shown in Figure 6. In this section, the animation of 3D visualization results is used to analyze the results of fixed-point flight and an 8-shaped expected flight trajectory of quadrotor UAV. Firstly, the flight animation of the quadrotor UAV model is exported to a video file, then the video file is converted into a series of image frames, and finally, the image frames are superimposed according to Equation (11) to generate a 3D visualization result.

5.3.1. Principle of 3D Visualization Results

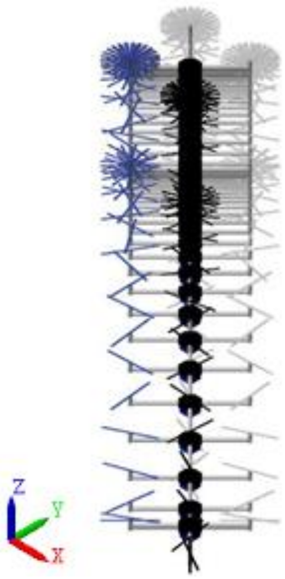
The current image frame and the previous accumulated image frame are superimposed. In the process of superposition, the pixel value of the same pixel point takes the accumulated image frame as the priority, and partial pixel superposition and partial pixel retention methods are adopted, specifically as follows.

$$D(i,j,k)=\begin{cases} I_2(i,j,k) & I_1(i,j,k) > I_2(i,j,k) \\ I_1(i,j,k) & I_1(i,j,k) < I_2(i,j,k) \end{cases} \tag{11}$$

Where, i, j, k are the index positions of pixel points, and their sizes are related to the size of the image. D is a temporary image frame, and $D(i,j,k)$ is a temporary pixel value. When the pixel value of the current image frame is smaller than that of the accumulated image frame, updating the pixel value of the accumulated image frame with the pixel value of the current image frame. On the contrary, the accumulated image frames are taken first, that is, the pixel values of the accumulated image frames are reserved. According to equation (11), the algorithm traverses the pixel values of the current image frame and assigns pixel frame D to pixel frame I. The algorithm repeats the previous process and continue to overlay with the next image frame until all image frames are overlaid.

5.3.2. 3D Visualization Results of Fixed-point Flight

The 3D visualization results of fixed-point flight of quadrotor UAV are shown in Figure 15. Figure 15 shows the 3D visualization results of vertical fixed-point flight, with fixed-point values of (0, 0, 1.5m) and (0,0,2m). Figure 15 is divided into two stages according to the set fixed-point position data. At the initial stage of take-off, the flight speed is slow, and there are many 3D images of UAV flight. At the middle stage, the flight speed is fast, and there are few 3D images of UAV flight. After the UAV arrives at the flight point, it flies at the fixed point, and there are many 3D images. The flying speed of UAV can be seen from the number or density of 3D images of UAV flying. When hovering at a fixed point, the number of 3D is large or dense.



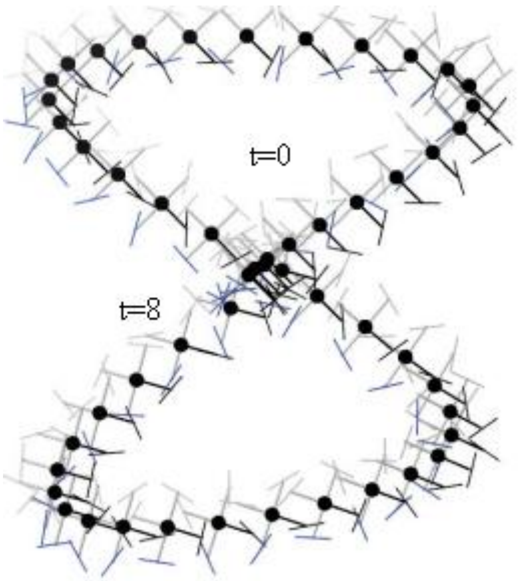


Figure 15: 3D results of fixed-point flight (axonometric drawing). Figure 16: 3D visualization results of the 8-shaped trajectory flight.

5.3.3. 3D Visualization Results of the 8-shaped Flight Trajectory

Pose is the core content of UAV flight control. Figures 14 only give the flight trajectory diagram, but they can't give the flight attitude of the UAV. Compared with Figure 15-16, Figure 14 not only gives the flying posture of UAV intuitively, but also can analyze the flying posture of UAV from multiple perspectives. As can be seen from Figure 16, the heading angle of quadrotor UAV can be observed from the top view. At t=0s, the initial heading angle is about 0 degree, and at t=8s, the heading angle is about 30 degree.

6. Discussion

For the purpose of analysis and comparison, the body models of quadrotor UAV are built under Simulink and Simscape respectively. By analyzing Figure 14-16, it can be concluded that:

(1) The body model of quadrotor UAV can be constructed by Simscape multi-body or Simulink module, and both of them have the same effect. In the existing literature, the simulation model is usually established under Simulink and the simulation results are given.

(2) Generally, the numerical simulation results can be obtained under Simulink. Under Simscape, not only numerical results can be obtained, but also 3D visual simulation results can be given. Using 3D visual simulation results, it is more intuitive to analyze the flight system.

(3) The flight ability of UAV tracking trajectory is related to the control algorithm and the expected tracking trajectory, and the expected control goal can be achieved by selecting different control algorithms or adjusting the parameters of the control algorithm.

(4) The tracking trajectory flight effect of quadrotor UAV is closely related to the response time. The control algorithm with shorter response time has better tracking effect. The response time of the control algorithm in this paper is short, and the tracking effect is good.

(5) Coordinate transformation theory has an important theoretical basis in the construction of Simscape model.

7. Conclusion

Aiming at the problem that it is difficult for quadrotor UAV to directly observe the real-time flight posture and adjust multiple parameters of flight controller when flying at a high altitude, this paper establishes the structural relationship and motion analysis of quadrotor UAV based on coordinate transformation theory, and completes a platform for visual experimental simulation and verification of flight control of quadrotor UAV by combining Solidworks and MATLAB/Simscape. In order to analyze and compare, the body model of quadrotor UAV is built under Simulink and Simscape respectively, and a kind of 8-shaped expected flight trajectories based on trigonometric function is designed respectively. Using the experimental platform in this paper, the simulation results show that the tracking flight ability of quadrotor UAV is related to the control algorithm and the expected tracking trajectory, and the tracking flight effect of quadrotor UAV is closely related to the response time. Therefore, the simulation system in this paper is intuitive, accurate and reliable, easy to debug and analyze the flight control system, which can improve the development efficiency and reduce the development cost of the control algorithm of quadrotor UAV, and can better study and test the flight control algorithm, which is of great value to the research of quadrotor UAV and flight control algorithm.

Acknowledgements

This work is supported by 2021 "Light of Textile" China Textile Industry Federation Higher Education Teaching Reform Research Project (5582021BKJGLX558), Zhongyuan Institute of Technology School Education Reform Project (2021ZGJGLX058), 2023 Horizontal Project: Research on UAV Flight Control System Visualization (22002542).

References

[1] Xu jiangyin, Zhao hongqiang, Deng Yu. (2017) Visual Trajectory Tracking Simulation System for Quadrotor UAV [J]. Computer Measurement and Control, 25(3), 130-134.
[2] Shang B, Zhang Y, Wu C, et al. (2018) Fractional Order Flight Control of Quadrotor UAS: an OS4 Benchmark Environment and a Case Study [C] // 2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV), 1-6.
[3] Koksal N, An H, Fidan B. (2020) Backstepping-based Adaptive Control of a Quadrotor UAV with Guaranteed Tracking Performance [J]. ISA Transactions, 105-112.
[4] Bo S, Wu C, Zhang Y, et al. (2018) Fractional Order Flight Control of Quadrotor UAS: A Simscape Benchmark Environment and a Case Study [C] // 2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV), 1-5.
[5] Matouk D, Gherouat O, Abdessemed F, et al. (2016) Quadrotor Position and Attitude Control via Backstepping Approach [C] // International Conference on Modelling, IEEE, 73-79.
[6] Farzaneh M M, Tavakolpour-Saleh A. (2020) Adaptive Trajectory Tracking Control of a Quadrotor Based on Iterative Learning Algorithm [J]. Journal of Engineering Technology and Applied Sciences, 1, 453-462.
[7] Zhang Xianmin. (2019) Robot Technology and Its Application (2nd Edition) [M]. Mechanical Industry Press.
[8] Cui Jintao, Zeng Qingdong, Wang Tianli, et al. (2015) Modeling and Simulation of Lithium Battery Based on Simscape [J]. Agricultural Equipment and Vehicle Engineering, 53(11), 31-35.
[9] Ji Jiang, Han Jianing, Meng Lifan. (2020) Modeling and Simulation of Quadrotor UAV Based on Simscape [J]. Testing Science and Instrument: English Edition, 11(2), 169-176.
[10] Derrouaoui S H, Bouzid Y, Guiatni M. (2021) PSO Based Optimal Gain Scheduling Backstepping Flight Controller Design for a Transformable Quadrotor [J]. Journal of Intelligent & Robotic Systems, 102(3), 1-25.