Problem Chosen	2023	Team Control Number
A	ShuWei Cup Summary Sheet	2023101519958

Modeling and Optimization Control of Compound Helicopters

Abstract

This article utilizes the aerodynamic parameters and body parameters of compound helicopters to conduct a mechanical analysis of compound helicopters, establish the dynamics differential equations of helicopters, and consequently derive a model for changes in helicopter attitude angles. It also provides attitude angle data for different time points and specifies control values for various helicopter components to ensure the smooth completion of helicopter flight missions.

Question 1: This question requires the establishment of a model for changes in the helicopter's pitch angle and the provision of specific pitch angles at certain moments. Under the given basic conditions, we analyze the pitch moment acting on the helicopter. We consider the pitch moment as the sum of the aerodynamic moment generated by the rotor and the pitch moment of the horizontal tail. We calculate these two moment components separately to obtain the total moment. Based on the relationship between moments and angular acceleration, we establish the helicopter's dynamics differential equation. By solving this differential equation, we obtain the pitch angle change model.

Question 2: This question requires the establishment of models for changes in three different attitude angles of the helicopter and the provision of specific attitude angles at certain moments. Under the given basic conditions, we analyze the moments acting on the helicopter. We consider the pitch moment as the sum of the aerodynamic moment generated by the rotor and the pitch moment of the horizontal tail, the roll moment as the sum of the aerodynamic moment generated by the rotor and the roll moment of the vertical tail, and the yaw moment as the sum of the aerodynamic moment generated by the rotor and the yaw moment of the vertical tail. We calculate these moment components separately to obtain the total moments in different directions. Based on the relationship between moments and angular velocity, we establish dynamic differential equations. By solving these differential equations, we obtain the attitude angle change models.

Question 3: This question requires the control of different components of the helicopter to complete horizontal flight missions in two scenarios: high-speed and low-speed. Assuming the helicopter is flying at a constant speed, in the low-speed scenario, attitude angle control is primarily achieved through the coaxial rotor, while in the high-speed mode, attitude angle control is mainly accomplished through the elevator and rudder. We select different control inputs to establish dynamic differential equations for the helicopter in these two scenarios and obtain attitude angle change models. The optimization objective is to minimize the sum of the three attitude angles at different times. We use the particle swarm optimization algorithm to obtain the optimal control inputs for various components.

Question 4: This question requires the control of different components of the helicopter to complete horizontal flight and forward acceleration tasks while the helicopter accelerates forward. We divide the flight process into low-speed, medium-speed, and high-speed stages and choose different control inputs at each stage. We establish dynamic differential equations for the helicopter in these different stages and derive attitude angle change models. The optimization objective is to minimize the sum of the three attitude angles at different times. We use the particle swarm optimization algorithm to obtain the optimal control inputs for various components at different times.

Keywords: Rotational Theorem, Ordinary Differential Equations, Particle Swarm Optimization Algorithm

Content

1. Introduction

1.1 Background

Helicopters have a wide range of uses, including transportation, patrolling, and rescue missions. However, traditional helicopters only have a rotor, and during high-speed flight, the rotor is susceptible to turbulence, which can lead to unstable helicopter flight. In contrast, compound helicopters equipped with both fixed wings and a rotor can significantly enhance the high-speed flying capabilities of helicopters. The rotor generates lift for tasks such as takeoff, hovering, and landing, while the propulsion system generates thrust for forward flight. Compound helicopters represent an important research direction in the field.

1.2 Work

In order to obtain the attitude angle change model for the helicopter, we perform force analysis on the helicopter to establish dynamic differential equations. This allows us to derive how the helicopter's attitude angles change over time. Then, under different velocity conditions, we select different control inputs with the objectives of completing flight missions and ensuring flight stability. We optimize these control inputs to ultimately obtain the optimal flight control strategy.

2. Problem analysis

2.1 Analysis of question one and two

Questions one and two are essentially about the dynamic analysis of the helicopter, with the only difference being the different sets of basic data provided for each question. Additionally, question one only requires considering the pitch moment, while question two involves considering moments in all three directions. Therefore, we apply the same approach to analyze both questions. We treat the pitch moment as the sum of the aerodynamic moment generated by the rotor and the pitch moment of the horizontal tail, consider the roll moment as the sum of the aerodynamic moment generated by the rotor and the roll moment of the vertical tail, and view the yaw moment as the sum of the aerodynamic moment generated by the rotor and the yaw moment of the vertical tail. We calculate these moment components separately to obtain the total moment in different directions. Then, based on the relationship between moments and angular velocities, we establish dynamic differential equations. By solving these differential equations, we obtain the attitude angle change model.

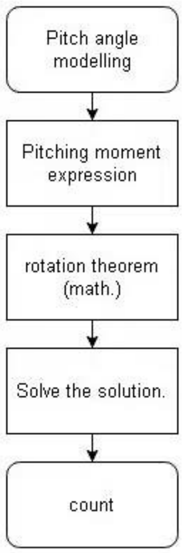


Fig. 1.Flowchart for the Solution to the First Question

2.2 Analysis of question three

Question three requires us to control the helicopter to accomplish fast horizontal flight tasks smoothly. This question is built upon the attitude angle change model established in questions one and two, with the constants given in questions one and two replaced by control variables. We represent the attitude angles using these variables, with the objective of minimizing the sum of attitude angle values at different times. We use the Particle Swarm Optimization algorithm to obtain the optimal control strategy.

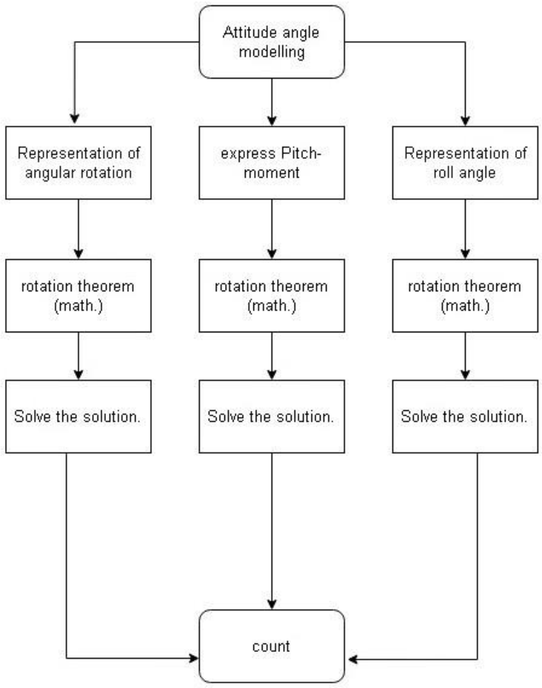


Fig. 2.Flowchart for the Solution to the Second Question

2.3 Analysis of question four

Question four introduces a variation based on question three, where we need to control the helicopter to complete an accelerated flight task. Unlike question three, which separately considered high-speed and low-speed scenarios, question four combines different velocity conditions within a single flight process. However, we can still segment time and select different control variables for low-speed, medium-speed, and high-speed time segments. It's important to note that in the previous questions, we assumed the helicopter was flying at a constant speed without considering the torque generated by the propulsion system. In question four, as the helicopter accelerates, we need to consider the torque generated by the propulsion system. With the objective of minimizing the sum of attitude angle values at different times, we use the Particle Swarm Optimization algorithm to obtain the optimal control strategy.

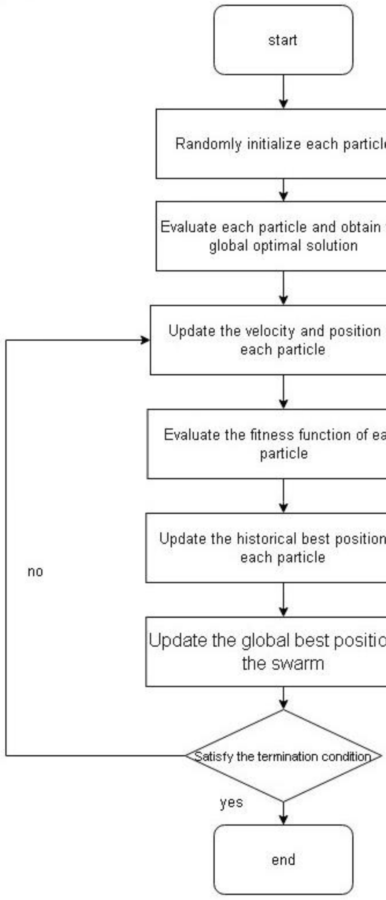


Fig. 3.Flowchart of Particle Swarm Optimization Algorithm

3. Symbol and Assumptions

3.1 Symbol Description

Symbol	Description
M_{total_y}	expression for thepitch moment
M_y	Pitch torque generated by coaxial rotors
M_h	Pitch torque generated by horizontal tail
k_y	Rotor roll coefficient
M_{total_x}	Roll torque
k_v	Vertical tail coefficient
u_e	Coaxial rotor longitudinal cyclic pitch
u_c	Coaxial rotor collective angle
u_{cd}	Coaxial rotor differential collective angle
k_h	Horizontal tail coefficient
M_{total_z}	Rotor yaw coefficient
u_{eh}	Elevator deflection value
u_{av}	Rudder deflection value

3.2 Fundamental assumptions

1. In questions one, two, and three, the thrust force of the propeller is zero.

4. Model

4.1 Question One:

HelicopterFlightModel

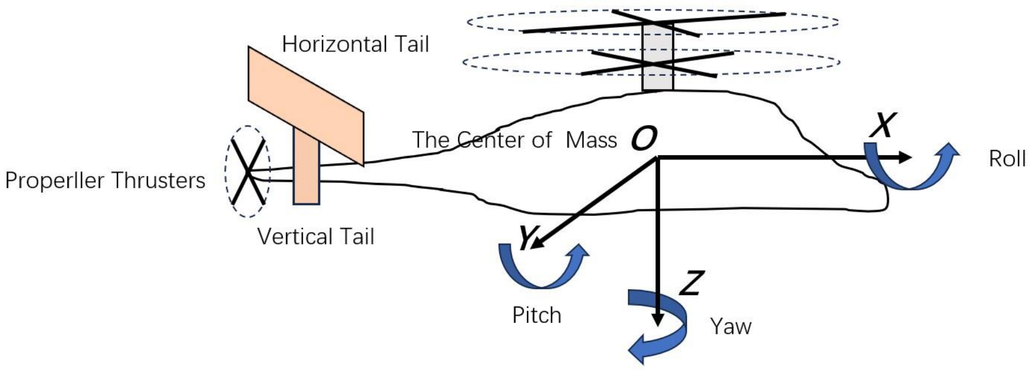


Fig. 4 . Coaxial helicopter configuration and roll, yaw, pitch schematic diagram

Establish a coordinate system fixed on the helicopter as shown in the diagram, where the angle of rotation around the y-axis is the pitch angle, the angle of rotation around the x-axis is the roll angle, and the angle of rotation around the z-axis is the yaw angle.

Parameter Fitting Function:

When calculating moments, helicopter parameters vary with changes in the rotor advance ratio. We need to have information about the helicopter parameters under different rotor advance ratios. The problem provides some data, but it is not sufficient for our use. Therefore, we perform a polynomial fitting of the helicopter parameters as a function of the rotor advance ratio. The fitting function is as follows:

Horizontal tail moment coefficient deviation

Elevator coefficient

Vertical tail moment coefficient deviation

Rudder coefficient

Roll deflection value

Lateral pitch roll factor

Differential total distance roll coefficient

Pitch deviation value

Longitudinal variable pitch coefficient

Total pitch coefficient

Differential total pitch coefficient

Yaw deviation value

Differential total pitch yaw coefficient

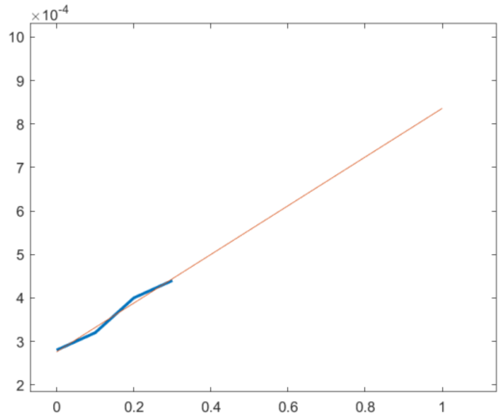


Fig. 5.Function Lateral pitch roll factor for Rotor advanced ratio

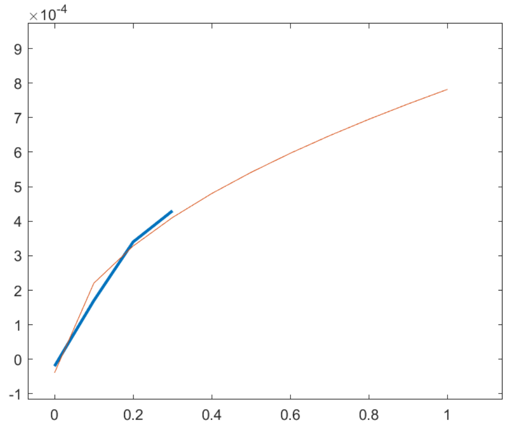


Fig. 6.Function Total pitch coefficient for Rotor advanced ratio

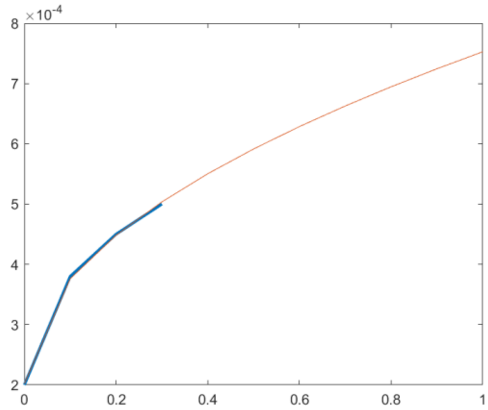


Fig. 7.Function Longitudinal variable pitch coefficient for Rotor advanced ratio

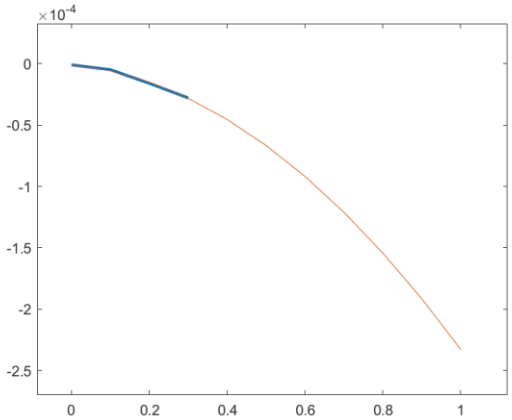


Fig. 8.Function Rudder coefficient for Rotor advanced ratio

Helicopter pitch moment

According to the question, the helicopter capable of generating pitch moments includes the coaxial rotor and the horizontal tail. Therefore, we decompose the pitch moment into the aerodynamic moment generated by the coaxial rotor and the torque generated by the horizontal tail as follows:

$$M_{total\_y} = M_y + M_h$$

Next, let's express the decomposed moments. Regarding the rotor moment, we have:

$$M_y = \frac{1}{2} \cdot k_y \cdot S_1 \cdot \rho \cdot v^2$$

Where:

$$k_y = f_2(p) + g_2(p) \cdot u_e + c_2(p) \cdot u_c + m \cdot u_{cd}$$

$$u_e = -3.4817$$

$$u_c = 0$$

$$u_{cd} = -2.1552$$

$$m = 5000$$

$$S_1 = \pi \cdot r^2, r = 6m$$

$$\rho = \rho_0 (1 - \frac{L \cdot h}{T_0})^{(\frac{g}{L \cdot R} - 1)}$$

$$\rho_0 = 1.225 kg / m^3$$
$$L = 0.0065 k / m$$
$$h = 3000 + v_v \cdot t$$
$$T_0 = 288.15 k$$
$$g = 9.80665 m / s^2$$
$$R = 287.05 J / kg$$

For the horizontal tail moment, we have:

$$M_h = k_h \cdot F \cdot S_2 \cdot d_1$$

where：

$$k_h = \sigma_h + \varphi_h(p) \cdot u_{eh}$$

$$u_{eh} = -9.0772 \times 10^{-7}$$

$$F = \frac{1}{2} \cdot \rho \cdot v^2$$

$$S_2 = 1m^2$$

$$d_1 = -3m$$

Finally, substituting back into the total moment equation, we obtain the expression for the pitch moment as:

$$M_{total\_y} = \frac{1}{2} \cdot k_y \cdot S_1 \cdot \rho \cdot v^2 + k_h \cdot F \cdot S_2 \cdot d_1$$

Pitch angle change model

According to the dynamics formula relating torque to angular acceleration：

$$\begin{cases} M_{total\_y} = J_y \cdot \dfrac{d^2\theta}{dt^2} \\ \theta(0) = 0, \dfrac{d\theta}{dt}(0) = 0 \end{cases}$$

Using the built-in dsolve function in MATLAB to solve this differential equation, we obtain the values of the attitude angles at various times as follows:

$$\theta(5) = 3.81640^{\circ}$$

$$\theta(10) = 15.2637^{\circ}$$

$$\theta(20) = 61.0504^{\circ}$$

4.2 Question two

Pitch moment: Same as in question one.

Pitch Angle Change Model: Same as in question one.

Using the built-in dsolve function in MATLAB to solve this differential equation, we obtain the values of the attitude angles at various times as follows:

$$\theta(5) = 3.81640^{\circ}$$

$$\theta(10) = 15.2637^{\circ}$$

$$\theta(20) = 61.0504^{\circ}$$

Rolling moment:

According to the question, the helicopter capable of generating roll moments includes the coaxial rotor and the rudder. Therefore, we decompose the roll moment into the aerodynamic moment generated by the coaxial rotor and the torque generated by the rudder as follows:

$$M_{total\_x} = M_x + M_{v2}$$

For the rotor torque

$$M_x = \frac{1}{2} \cdot k_x \cdot S_1 \cdot \rho \cdot v^2$$

where：

$$k_x = f_1(p) + g_1(p) \cdot u_a + c_1 \cdot u_{cd}$$

For the rudder torque：

$$M_{v2} = k_v \cdot F \cdot S_3 \cdot d_3$$

Whre:

$$S_3 = 0.5m^2$$

Finally, substituting back into the roll moment formula and using the data provided in the question, we obtain the expression for the roll moment.：

$$M_{total\_x} = \frac{1}{2} \cdot k_x \cdot S_1 \cdot \rho \cdot v^2 + k_v \cdot F \cdot S_3 \cdot d_3$$

Roll angle change model

According to the dynamics formula relating torque to angular acceleration：

$$\begin{cases} M_{total\_x} = J_x \cdot \dfrac{d^2\phi}{dt^2} \\ \varphi(0) = 0, \dfrac{d\varphi}{dt}(0) = 0 \end{cases}$$

Using the built-in dsolve function in MATLAB to solve this differential equation, we obtain the values of the attitude angles at various times as follows:

$$\varphi(5) = -0.390457$$

$$\varphi(10) = -1.56177$$

$$\varphi(20) = -6.24666$$

Yaw moment:

According to the question, the helicopter capable of generating yaw moments includes the coaxial rotor and the rudder. Therefore, we decompose the yaw moment into the aerodynamic moment generated by the coaxial rotor and the torque generated by the rudder as follows：

$$M_{total\_z} = M_z + M_{v1}$$

For the rotor torque：

$$M_z = \frac{1}{2} \cdot k_z \cdot S_1 \cdot \rho \cdot v^2$$

where:

$$k_z = f_3(p) + c_3 \cdot u_{cd}$$

For the rudder torque：

$$M_{v1} = k_v \cdot F \cdot S_3 \cdot d_2$$

where：

$$k_v = \sigma_v + \varphi_v(p) \cdot u_{av}$$

Finally, substituting back into the yaw moment formula and using the data provided in the question, we obtain the expression for the yaw moment：

$$M_{total\_z} = \frac{1}{2} \cdot k_z \cdot S_1 \cdot \rho \cdot v^2 + k_v \cdot F \cdot S_3 \cdot d_2$$

Yaw angle change model:

According to the dynamics formula relating torque to angular acceleration

$$\begin{cases} M_{total\_z} = J_z \cdot \dfrac{d^2\psi}{dt^2} \\ \psi(0) = 0, \dfrac{d\psi}{dt}(0) = 0 \end{cases}$$

Using the built-in dsolve function in MATLAB, we find that the solution to this differential equation is：

$$\psi(5) = -0.0596591$$

$$\psi(10) = -0.238628$$

$$\psi(20) = --0.954448$$

4.3 Question three

Attitude angle change model

Low speed mode

In the low-speed mode, the attitude angles are primarily controlled through the coaxial rotor and the rotor propellers. However, based on the assumption that the helicopter is moving forward at a constant speed, the rotor propellers are not active. Therefore, we only consider the coaxial rotor. We select the overall angle of the coaxial rotor u_c and the differential total angle of the coaxial rotor u_{cd} , Coaxial rotor longitudinal cycle pitch u_e, Coaxial rotor lateral cycle pitch u_a as the variables that need to be controlled, and the rest of the variables are set to zero. So we get the three directional moments as follows:

Pitch torque:

$$M_{total\_y} = \frac{1}{2} \cdot (f_2(p) + g_2(p) \cdot u_e + c_2(p) \cdot u_c + m \cdot u_{cd}) \cdot S_1 \cdot \rho \cdot v^2 + \sigma_h \cdot F \cdot S_2 \cdot d_1$$

Rolling torque:

$$M_{total\_x} = \frac{1}{2} \cdot (f_1(p) + g_1(p) \cdot u_a + c_1 \cdot u_{cd}) \cdot S_1 \cdot \rho \cdot v^2 + \sigma_v \cdot F \cdot S_3 \cdot d_3$$

Yaw torque:

$$M_{total\_z} = \frac{1}{2} \cdot (f_3(p) + c_3 \cdot u_{cd}) \cdot S_1 \cdot \rho \cdot v^2 + \sigma_v \cdot F \cdot S_3 \cdot d_2$$

As the variables we need to control, the remaining variables are set to zero. Thus, we obtain the following torque values in three directions：

Pitch Angle:

$$M_{total\_y} = J_y \cdot \frac{d^2\theta}{dt^2}$$

Roll Angle:

$$M_{total\_x} = J_x \cdot \frac{d^2\phi}{dt^2}$$

Yaw Angle:

$$M_{total\_z} = J_z \cdot \frac{d^2\psi}{dt^2}$$

High-speed mode:

In high-speed mode, the attitude angles are primarily controlled by the elevator and the rudder. We select Elevator deflection values, Rudder deflection values As control variables, the remaining variables are set to zero, and we obtain torque values in three directions as follows$u_{eh} u_{av}$:

Pitch torque:

$$M_{total\_y} = \frac{1}{2} \cdot f_2(p) \cdot S_1 \cdot \rho \cdot v^2 + (\sigma_h + \varphi_h(p) \cdot u_{eh}) \cdot F \cdot S_2 \cdot d_1$$

Rolling torque:

$$M_{total\_x} = \frac{1}{2} \cdot f_1(p) \cdot S_1 \cdot \rho \cdot v^2 + (\sigma_v + \varphi_v(p) \cdot u_{av}) \cdot F \cdot S_3 \cdot d_3$$

Yaw torque:

$$M_{total\_z} = \frac{1}{2} \cdot f_3(p) \cdot S_1 \cdot \rho \cdot v^2 + (\sigma_v + \varphi_v(p) \cdot u_{av}) \cdot F \cdot S_3 \cdot d_2$$

Based on the dynamics formula relating torque to angular acceleration, we have three different attitude angle change models.:

Pitch Angle:

$$M_{total\_y} = J_y \cdot \frac{d^2\theta}{dt^2}$$

Roll Angle:

$$M_{total\_x} = J_x \cdot \frac{d^2\phi}{dt^2}$$

Yaw Angle:

$$M_{total\_z} = J_z \cdot \frac{d^2\psi}{dt^2}$$

Helicopter control strategy model

To achieve stable horizontal flight of the helicopter, we need to keep the helicopter's attitude angles within a very small range as much as possible. Starting from this, we consider the sum of the values of the three attitude angles at 100 different times as the objective variable. We need to adjust the control variables to minimize the objective variable and obtain the planning model：

$$\begin{cases} \min \sum_{i=1}^{100} |\theta(\mathrm{i})| + |\phi(\mathrm{i})| + |\psi(\mathrm{i})| \\ u_c \in [0, 30] \\ u_{cd} \in [-25, 25] \\ u_e \in [-25, 25] \\ u_a \in [-25, 25] \end{cases}$$

By substituting the different attitude angle change models for high-speed and low-speed scenarios, we can obtain two different planning models for high-speed and low-speed modes, respectively. Then, we utilize the Particle Swarm Optimization algorithm to search for the optimal control strategy. The steps of the Particle Swarm Optimization algorithm are as follows：

Step1: Initialize the values of the particle swarm：

Using u_c, u_e, u_{cd}, u_a Considered as the four dimensions to be optimized for each particle, where each particle is represented as $pop_i(u_c, u_e, u_{cd}, u_a)$, and initialize u_c, u_e, u_{cd}, u_a the initial change in velocity v_i^d。

Step2: Construct the objective function (fitness function) and initialize the fitness

The final fitness is：

$$\theta_{total} = \theta_{pitch}(u_c, u_e, u_{cd}) + \theta_{roll}(u_{cd}, u_a) + \theta_{yaw}(u_{cd})$$

The optimization objective is to find a suitable set of u_c, u_e, u_{cd}, u_a, such that θ_{total} (the sum of attitude angles) is minimized, achieving stable zero attitude angle flight.

Update the positions and velocities of all particles after initialization u_c, u_e, u_{cd}, u_a Using these parameters as inputs, calculate the initial fitness for all particles F_i

Step3： Velocity update

Velocity update equation: TIMEs

Position update equation:

$$v_{i+1}^d = \omega \cdot v_i^d + c_1 \cdot rand_1^d (Pbest_i^d - x_i^d) + c_2 \cdot rand_2^d \cdot (Gbest_i^d - x_i^d)$$

$$x_{i+1}^d = x_i^d + v_{i+1}^d$$

In the formula: x_i^d represents the d-th dimension value of the i-th particle; v_i^d represents the velocity of the d-th dimension of the i-th particle;

pbest_i represents the best position found by the i-th particle in the d-th dimension so far;

$Pbest_i^d$ represents the best position found by the i-th particle in the d-th dimension so far;

$Gbest_i^d$ represents the best position found by the entire population in the d-th dimension so far

The control parameters update in the low speed are:

$$u_c = 0 \quad u_e = -4.6691 \quad u_{cd} = -0.8617 \quad u_a = -1.1818$$

The control parameters update in the high speed are:

$$u_{eh} = -24.9263, \quad u_{av} = -10.2302$$

4.4 Question Four

In the consideration of the helicopter's acceleration maneuver, the forward component of the aircraft accelerates uniformly from 80 meters per second to 180 meters per second within 20 seconds, and the forward acceleration is entirely provided by the rotor propellers. Given that the mass of the aircraft is 5000 kg, it can be determined that the required thrust is at least 25000 N, which is significantly greater than the thrust that can be provided within the given range. Therefore, considering the practical situation, we contemplate the helicopter accelerating from 80 meters per second to 120 meters per second within 80 seconds.

Based on the data provided in the appendix, we can fit the data to obtain the rotor propeller's capability value as 16.7 when the acceleration is 0.5 meters per second squared, with the thrust and torque provided being 2500 N and 1505 Nm, respectively. Under this premise, we divide the aircraft's acceleration process into three stages. During different time intervals, considering the actual control situation of each control variable, we solve for the values of each control variable at each moment by minimizing the sum of the absolute values of torque in each direction as the indicator, in order to obtain the dynamic values of control inputs."

$$k_y = f_2(p) + g_2(p) \cdot u_e + c_2(p) \cdot u_c + m \cdot u_{cd}$$
$$M_y = \frac{1}{2} \cdot k_y \cdot S_1 \cdot \rho \cdot v^2$$
$$k_h = \sigma_h + \varphi_h(p) \cdot u_{eh}$$
$$M_h = k_h \cdot F \cdot S_2 \cdot d_1$$
$$M_{p\_y} = F_p \cdot d_4$$
$$M_{total\_y} = M_y + M_h + M_{p\_y}$$

$$k_x = f_1(p) + g_1(p) \cdot u_e + c_1 \cdot u_{cd}$$
$$M_x = \frac{1}{2} \cdot k_x \cdot S_1 \cdot \rho \cdot v^2$$
$$k_v = \sigma_v + \varphi_v(p) \cdot u_{av}$$
$$M_{v1} = k_v \cdot F \cdot S_3 \cdot d_2$$
$$M_{p\_y} = F_p \cdot d_4$$
$$M_{total\_x} = M_x + M_{v1} + M_{p\_y}$$

$$k_z = f_3(p) + c_3 \cdot u_{cd}$$
$$M_z = \frac{1}{2} \cdot k_z \cdot S_1 \cdot \rho \cdot v^2$$
$$k_v = \sigma_v + \varphi_v(p) \cdot u_{av}$$
$$M_{v2} = k_v \cdot F \cdot S_3 \cdot d_3$$
$$M_{total\_z} = M_z + M_{v2}$$

t: 0-10

During this time interval, speeds less than 85 meters per second are considered the low-speed phase, and control input values are only considered for u_c, u_e, u_{cd}, u_a u_t All other values are set to zero.

t: 10-40

During this time interval, speeds range from 85 meters per second to 100 meters per second, and control input values are considered for $u_c, u_e, u_{cd}, u_a, u_{eh}, u_{av}, u_t$ 。

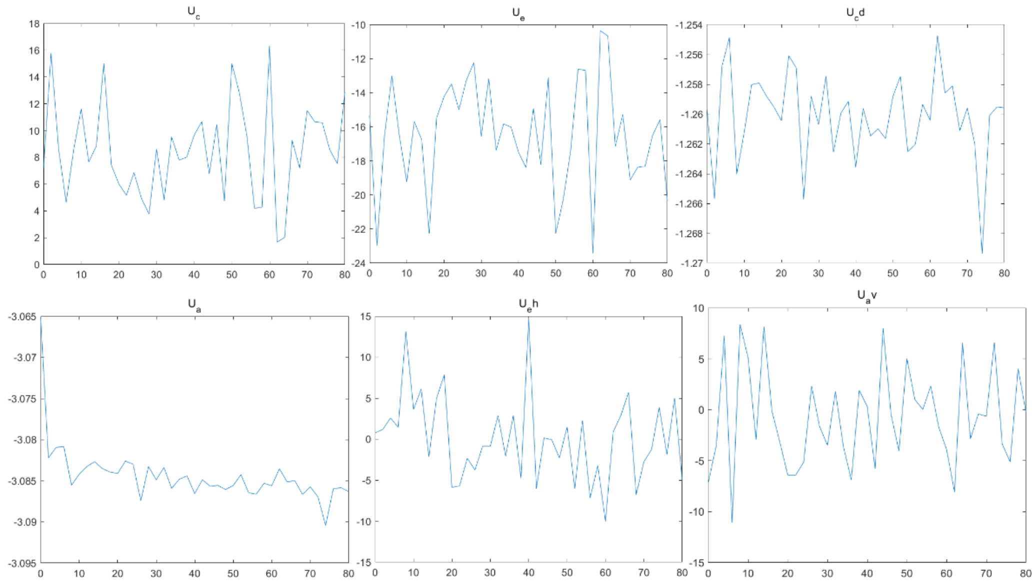
t: 40-80

During this time interval, speeds range from 100 meters per second to 120 meters per second, and control input values are considered for $u_c, u_e, u_{cd}, u_a, u_{eh}, u_{av}, u_t$

The following equation represents the objective function.

$$\min \sum \left|M_{total\_z}(\mathrm{t})\right| + \left|M_{total\_x}(\mathrm{t})\right| + \left|M_{total\_y}(\mathrm{t})\right|$$

"Similarly, utilizing the Particle Swarm Optimization algorithm within the specified range, we optimize and calculate the optimal values of various control inputs at a certain number of time points during different time intervals. Finally, we depict dynamic value graphs of the input control variables."



5.Strengths and Weakness

5.1 Advantages

1. PSO is an iteration-based optimization algorithm with the advantages of simplicity and easy implementation. It requires few adjustable parameters, converges quickly, and has many measures to avoid getting stuck in local optima.

6.Conclusion

To establish a model for the attitude angle changes of a compound helicopter, this paper utilizes polynomial fitting to process the subject data and conducts a force analysis on the helicopter, successfully establishing the helicopter's dynamic differential equations. The symbolic differential equations are solved using the dsolve function. Based on this, an optimization model is established with the minimization of the sum of attitude angles as the optimization goal. The optimal control strategy is solved using the Particle Swarm Optimization (PSO) algorithm. The PSO algorithm is an iterative-based optimization algorithm, whose advantages include simplicity and ease of implementation, fewer parameters to adjust, fast convergence speed, and numerous measures to avoid getting trapped in local optima.

Appendix

The code of first question

```
clc;
clear;

theta_final = [];
syms t theta(t)

F_dynamic = 1/2.*rho(t).*6404;
k_yaw = f2(sqrt(6404)/180)+g2(sqrt(6404)/180).*(-
3.4817)+m(sqrt(6404)/180).*(-2.1552);
M_yaw = k_yaw.*pi.*36.*rho(t).*1/2.*180.^2;
k_hor = -0.0001+phy_ud(sqrt(6404)/180).*(-9.0072).*0.0000001;
M_hor = k_hor.*F_dynamic.*1.*(-3);

M_total = M_yaw+M_hor;
% 定义二阶微分方程
eqn = 20000 * diff(theta, t, t) == M_total;

theta_d = diff(theta,t);
cond1 = theta(0) == 0;
cond2 = theta_d(0) == 0;

% 解微分方程
sol = simplify(dsolve(eqn, [cond1, cond2]));
% for t=0:0.1:20
%
%     theta_final=cat(1,theta_final,[t eval(simplify(subs(sol,'t',t)))]);
%
% end
theta_final= eval(simplify(subs(sol,'t',20)))
```

The code of two question

```
clc;
clear;

theta_p_final=[];
theta_r_final =[];
theta_y_final=[];
```

```
syms t theta_p(t) theta_r(t) theta_y(t)

F_dynamic = 1/2.*rho(t).*6400.04;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% 俯仰
k_pitch = f2(sqrt(6400.04)/180)+g2(sqrt(6400.04)/180).*(-
3.4817)+m(sqrt(6400.04)/180).*(-2.1552);
M_pitch = k_pitch.*pi.*36.*rho(t).*1/2.*180.^2;
k_hor = -0.0001+phy_ud(sqrt(6400.04)/180).*(-9.0772).*(-0.0000001);
M_hor = k_hor.*F_dynamic.*(-3);
M_y_total = M_pitch+M_hor;

m(sqrt(6400.04)/180)
eval(simplify(subs(M_pitch,'t',20)))


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% 滚动
k_roll = f1(sqrt(6400.04)/180)+g1(sqrt(6400.04)/180).*(-
2.0743)+c1(sqrt(6400.04)/180).*(-2.1552);
M_roll = k_roll.*pi.*36.*rho(t).*1/2.*180.^2;
k_ver_r =
sigma_v(sqrt(6400.04)/180)+phy_dir(sqrt(6400.04)/180).*(4.1869).*0.000000
1;
M_ver_r = k_ver_r.*F_dynamic.*0.5.*(0.2);

M_x_total = M_roll+M_ver_r;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% 偏航
k_yaw = f3(sqrt(6400.04)/180)+c3(sqrt(6400.04)/180).*(-2.1552);
M_yaw = k_yaw.*pi.*36.*rho(t).*1/2.*180.^2;
k_ver_y =
sigma_v(sqrt(6400.04)/180)+phy_dir(sqrt(6400.04)/180).*(4.1869).*0.000000
1;
M_ver_y = k_ver_y.*F_dynamic.*0.5.*(-3);

M_z_total = M_yaw+M_ver_y;

%% 求俯仰角度
eqn = 20000 * diff(theta_p, t, t) == M_y_total;

theta_p_d = diff(theta_p,t);
cond1 = theta_p(0) == 0;
cond2 = theta_p_d(0) == 0;
```

```
% 解微分方程
sol = dsolve(eqn, [cond1, cond2]);
for i = 0:0.1:20
    i
    theta_p_final=cat(1,theta_p_final,[i
eval(simplify(subs(sol,'t',i)))]);
end
%% 求滚动角度
eqn = 8000 * diff(theta_r, t, t) == M_x_total;

theta_r_d = diff(theta_r,t);
cond1 = theta_r(0) == 0;
cond2 = theta_r_d(0) == 0;

% 解微分方程
sol = dsolve(eqn, [cond1, cond2]);

for i = 0:0.1:20
    i
    theta_r_final=cat(1,theta_r_final,[i
eval(simplify(subs(sol,'t',i)))]);
end
%% 求偏航角度
eqn = 25000 * diff(theta_y, t, t) == M_z_total;

theta_y_d = diff(theta_y,t);
cond1 = theta_y(0) == 0;
cond2 = theta_y_d(0) == 0;

% 解微分方程
sol = dsolve(eqn, [cond1, cond2]);

for i = 0:0.1:20
    i
    theta_y_final=cat(1,theta_y_final,[i
eval(simplify(subs(sol,'t',i)))]);
end

figure
plot(theta_p_final(:,1),theta_p_final(:,2),'LineWidth',2)
xlabel('t /s',FontSize=18)%设置横坐标轴
ylabel('theta of pitch /deg',FontSize=18)
figure
```

```
plot(theta_r_final(:,1),theta_r_final(:,2),'LineWidth',2)
xlabel('t /s',FontSize=18)%设置横坐标轴
ylabel('theta of roll /deg',FontSize=18)
figure
plot(theta_y_final(:,1),theta_y_final(:,2),'LineWidth',2)
xlabel('t /s',FontSize=18)%设置横坐标轴
ylabel('theta of yaw /deg',FontSize=18)
```

The code of three question

```
clc;
clear;
%% 粒子群优化
u_c = 0:30;
u_e = -25:25;
u_cd = -25:25;
u_a = -25:25;


c1 = 1.49445;
c2 = 1.49445;

maxgen=200;   % 进化次数
sizepop=1000;   %种群规模

Vmax=2;
Vmin=-2;
pop_1_max=30;
pop_1_min=0;

pop_2_max=25;
pop_2_min=-25;


for i=1:sizepop
    i
    %随机产生一个种群
    pop1(i,:)=[randi([0,30])];
    pop2(i,:) = [randi([-25,25]),randi([-25,25]),randi([-25,25])];%初始种
群
    pop = cat(2,pop1,pop2);
    V(i,:)=2*rands(1,4);  %初始化速度
    %计算适应度
    fitness(i)=theta_total(pop(i,:));
```

```
%     fitness_r(i)=theta_r(pop(i,4),pop(i,2));
%     fitness_y(i)=theta_y(pop(i,2));
    %染色体的适应度
end

[bestfitness bestindex]=max(fitness);
zbest=pop(bestindex,:);   %全局最佳
gbest=pop;    %个体最佳
fitnessgbest=fitness;   %个体最佳适应度值
fitnesszbest=bestfitness;   %全局最佳适应度值

%% 迭代寻优
for i=1:maxgen
    i
    for j=1:sizepop

        %速度更新
        V(j,:) = V(j,:) + c1*rand*(gbest(j,:) - pop(j,:)) + c2*rand*(zbest
- pop(j,:));
        V(j,find(V(j,:)>Vmax))=Vmax;
        V(j,find(V(j,:)<Vmin))=Vmin;

        %种群更新
        pop1(j,:)=pop1(j,:)+V(j,1);
        pop2(j,:) = pop2(j,:)+V(j,2:4);
        pop1(j,find(pop1(j,:)>pop_1_max))=pop_1_max;
        pop1(j,find(pop1(j,:)<pop_1_min))=pop_1_min;
        pop2(j,find(pop2(j,:)>pop_2_max))=pop_2_max;
        pop2(j,find(pop2(j,:)<pop_2_min))=pop_2_min;
        pop = cat(2,pop1,pop2);

        %适应度值
        fitness(j)=theta_total(pop(j,:));

    end

    for j=1:sizepop

        %个体最优更新
        if fitness(j) < fitnessgbest(j)
            gbest(j,:) = pop(j,:);
            fitnessgbest(j) = fitness(j);
        end
```

```
        %群体最优更新
        if fitness(j) < fitnesszbest
            zbest = pop(j,:);
            fitnesszbest = fitness(j);
        end
    end
    yy(i)=fitnesszbest;

    zbest
    fitnesszbest

end

zbest
fitnesszbest


function theta_final = theta_total(u)

    syms t theta_p(t) theta_r(t) theta_y(t)

    F_dynamic = 1/2.*rho(t).*6400.04;
%  俯仰角度
    k_pitch =
c2(sqrt(6400.04)/180).*u(1)+f2(sqrt(6400.04)/180)+g2(sqrt(6400.04)/180).*u(
2)+m(sqrt(6400.04)/180).*u(3);
    M_pitch = k_pitch.*pi.*36.*rho(t).*1/2.*180.^2;
    k_hor = -0.0001;
    M_hor = k_hor.*F_dynamic.*(-3);
    M_p_total = M_pitch+M_hor;

    eqn = 20000 * diff(theta_p, t, t) == M_p_total;

    theta_p_d = diff(theta_p,t);
    cond1 = theta_p(0) == 0;
    cond2 = theta_p_d(0) == 0;

    % 解微分方程
    sol = dsolve(eqn, [cond1, cond2]);


    theta_p_final = abs(eval(simplify(subs(sol,'t',20))));

% 滚转角度
```

```
    k_roll =
f1(sqrt(6400.04)/180)+g1(sqrt(6400.04)/180).*u(4)+c1(sqrt(6400.04)/180).*u(
3);
    M_roll = k_roll.*pi.*36.*rho(t).*1/2.*180.^2;
    k_ver_r = sigma_v(sqrt(6400.04)/180);
    M_ver_r = k_ver_r.*F_dynamic.*0.5.*(0.2);

    M_r_total = M_roll+M_ver_r;

    eqn = 8000 * diff(theta_r, t, t) == M_r_total;

    theta_r_d = diff(theta_r,t);
    cond1 = theta_r(0) == 0;
    cond2 = theta_r_d(0) == 0;

    % 解微分方程
    sol = dsolve(eqn, [cond1, cond2]);
    theta_r_final = abs(eval(simplify(subs(sol,'t',20))));

% 偏航角度
    k_yaw = f3(sqrt(6400.04)/180)+c3(sqrt(6400.04)/180).*u(3);
    M_yaw = k_yaw.*pi.*36.*rho(t).*1/2.*180.^2;
    k_ver_y = sigma_v(sqrt(6400.04)/180);
    M_ver_y = k_ver_y.*F_dynamic.*0.5.*(-3);

    M_y_total = M_yaw+M_ver_y;

    eqn = 25000 * diff(theta_y, t, t) == M_y_total;

    theta_y_d = diff(theta_y,t);
    cond1 = theta_y(0) == 0;
    cond2 = theta_y_d(0) == 0;

    % 解微分方程
    sol = dsolve(eqn, [cond1, cond2]);

    theta_y_final = abs(eval(simplify(subs(sol,'t',20))));

% 总角度
    theta_final = theta_p_final+theta_r_final+theta_y_final;
end
```

The code of four question

```
clc;
clear;


u_t = [4    8   12  16  20  24  28  32  36];
F_thr = [200    1000    1900    2400    3200    4000    5800    8200
    11000];
M_roll_pro = [200   700   1100  1400   1900   2500   3700   4500   7000];

%%
a = polyfit(u_t,F_thr,3);
b = polyfit(u_t,M_roll_pro,3);

%%
% syms ut

% ut_solu= vpasolve(a(1).*ut.^3+ a(2).*ut.^2 + a(3).*ut + a(4)==2500,ut)
ut = 16.6182413689754323935866072417l8;
pro_roll = b(1).*ut.^3+ b(2).*ut.^2 + b(3).*ut + b(4);


u_c = [];
u_e = [];
u_cd = [];
u_a = [];
u_eh = [];
u_av = [];

fitnessrecord = [];

for t = 10:5:80

    c1 = 1.49445;
    c2 = 1.49445;

    maxgen=200;   % 进化次数
    sizepop=1000;   %种群规模

    Vmax=1.5;
    Vmin=-1.5;
    pop_1_max=30;
    pop_1_min=0;

    pop_2_max=25;
```

```
    pop_2_min=-25;


    for i=1:sizepop
        i
        %随机产生一个种群
        pop1(i,:)=[randi([0,30])];
        pop2(i,:) = [randi([-25,25]),randi([-25,25]),randi([-
25,25]),randi([-25,25]),randi([-25,25])];%初始种群
        pop = cat(2,pop1,pop2);
        V(i,:)=2*rands(1,6);  %初始化速度
        %计算适应度
        fitness(i)=four_stage2(pop(i,:),t);
    %     fitness_r(i)=theta_r(pop(i,4),pop(i,2));
    %     fitness_y(i)=theta_y(pop(i,2));
        %染色体的适应度
    end

    [bestfitness bestindex]=max(fitness);
    zbest=pop(bestindex,:);   %全局最佳
    gbest=pop;    %个体最佳
    fitnessgbest=fitness;   %个体最佳适应度值
    fitnesszbest=bestfitness;   %全局最佳适应度值

    %% 迭代寻优
    for i=1:maxgen
        i
        for j=1:sizepop

            %速度更新
            V(j,:) = V(j,:) + c1*rand*(gbest(j,:) - pop(j,:)) +
c2*rand*(zbest - pop(j,:));
            V(j,find(V(j,:)>Vmax))=Vmax;
            V(j,find(V(j,:)<Vmin))=Vmin;

            %种群更新
            pop1(j,:)=pop1(j,:)+V(j,1);
            pop2(j,:) = pop2(j,:)+V(j,2:6);
            pop1(j,find(pop1(j,:)>pop_1_max))=pop_1_max;
            pop1(j,find(pop1(j,:)<pop_1_min))=pop_1_min;
            pop2(j,find(pop2(j,:)>pop_2_max))=pop_2_max;
            pop2(j,find(pop2(j,:)<pop_2_min))=pop_2_min;
            pop = cat(2,pop1,pop2);
```

```
            %适应度值
            fitness(j)=four_stage2(pop(j,:),t);

        end

        for j=1:sizepop

            %个体最优更新
            if fitness(j) < fitnessgbest(j)
                gbest(j,:) = pop(j,:);
                fitnessgbest(j) = fitness(j);
            end

            %群体最优更新
            if fitness(j) < fitnesszbest
                zbest = pop(j,:);
                fitnesszbest = fitness(j);
            end
        end
        yy(i)=fitnesszbest;

        zbest
        fitnesszbest


    end

    fitnessrecord = cat(1,fitnessrecord,fitnesszbest);
    u_c = cat(1,u_c,[t zbest(1)]);
    u_e = cat(1,u_e,[t zbest(2)]);
    u_cd = cat(1,u_cd,[t zbest(3)]);
    u_a = cat(1,u_a,[t zbest(4)]);
    u_eh = cat(1,u_eh,[t zbest(5)]);
    u_av = cat(1,u_av,[t zbest(6)]);
end
fitnessrecord


% uc = cat(1,load('u_c.mat').u_c,u_c);
% ue = cat(1,load('u_e.mat').u_e,u_e);
% ucd = cat(1,load('u_cd.mat').u_cd,u_cd);
% ua = cat(1,load('u_a.mat').u_a,u_a);
% ueh = cat(1,load('u_eh.mat').u_eh,u_eh);
% uav = cat(1,load('u_av.mat').u_av,u_av);
```

```
figure


subplot(2,3,1)
plot(u_c(:,1),u_c(:,2))
subtitle('U_c')

subplot(2,3,2)
plot(u_e(:,1),u_e(:,2))
subtitle('U_e')

subplot(2,3,3)
plot(u_cd(:,1),u_cd(:,2))
subtitle('U_cd')

subplot(2,3,4)
plot(u_a(:,1),u_a(:,2))
subtitle('U_a')

subplot(2,3,5)
plot(u_eh(:,1),u_eh(:,2))
subtitle('U_eh')

subplot(2,3,6)
plot(u_av(:,1),u_av(:,2))
subtitle('U_av')
```