北京航空航天大學

BEIHANG UNIVERSITY

Bifang: A New Free-Flying Cubic Robot for Space Station

Galaxy Yanshi Team

Beihang University

Abstract

As astronauts conduct experiments more frequently on the space station, there is a growing trend in research on free-flying robots capable of assisting them with various tasks. This paper introduces a new class of free-flying robot named Bifang, designed to enhance astronauts' lives and improve their efficiency in space.

Bifang is a deployable cubic robot that measures 20cm×20cm×20cm when folded. Once expanded, the robot offers a range of functions, including autonomous and remote-controlled flight capabilities, the ability to grasp and carry objects, expression recognition to regulate emotions, a simple and efficient UI interaction, and health management features.

In terms of innovation, the unique variable-speed double-gimbal fans generate large thrust and make Bifang have 6 degrees of freedom, ensuring it can fly around the space station. Powerful trajectory planning algorithms enable the manipulator to grasp any object the astronauts want with constraints. Mature emotion recognition algorithms are applied on Bifang to help astronauts live healthier. Bifang has the ability of visual SLAM mapping, so that it can plan its own trajectory and work smarter.

In order to verify the feasibility of the design, we first performed the overall robot design. Then modeling the dynamics of variable-speed double-gimbal fans to theoretically verify the controllability of this structure. Theoretical research and simulation analysis on trajectory planning and obstacle avoidance algorithm of the manipulator is also carried out to ensure its functionality. Deep learning framework PyTorch is used to real-time recognize astronaut facial expressions captured by the camera and SLAM is utilized to real-time estimate the pose in changing space station environment for navigation of the Bifang in the space station. Finally, the robot parameters are checked.

A prototype of variable-speed double-gimbal fan at the initial stage of design is made and the thrust curve is given, further explained the rationality of the design.

Content

1 Introduction

Haichao Gui, the first payload expert on China's space station, hopes service robot with autonomous patrol and remote control functions will work in the space station to assist astronauts in various tasks. So we are working on a new type of free-flying robot, Bifang (毕方, shown in Fig. 1), which can fly autonomously in the space station, making their lives safer, smarter, and more comfortable.

Fig. 1 Free-flying Bifang Robot

1.1 Background of Free-Flying Robot

Free-flying robot strives to have high flexibility and maneuverability under the conditions of light weight and small volume, so as to complete as many tasks as possible. Free-flying robot can be used as microgravity research platforms for scientific experiments, as astronaut assistants, as shooting equipment, as communication devices between earth and space, as responsible for space station inspection and maintenance, reducing the workload of astronauts, and improving the efficiency and quality of work in the space station.

Free-flying robot are an important part of the in-orbit operation work of the space station, providing strong technical support for human work and life in space, and providing a broad prospect and future development direction for human development and innovation in space.

1.2 Research Status

The research and exploration of free-flying robot are paid more and more attention

by many countries in recent years, and they are developing free-flying robot with unique functions.

There are four types of classical free-flying robot which are technologically mature and have been put into use:

- **Spheres** (Synchronized Position Hold, Engage, Reorient, Experimental Satellites)[1][2], developed by the National Aeronautics and Space Administration (NASA), is small spherical robot group that can fly inside the International Space Station (ISS). Spheres are three robots, called Red, Blue and Green, which were sent to the International Space Station in 2006. They are composed of 3D printed parts, with a diameter of about 15 cm and a weight of about 4 kg. Each Sphere has some interfaces and sensors on its shell, which can connect different attachments or interact with the equipment of the space station. It is powered by carbon dioxide propulsion tanks, and requires the supervision and control of astronauts during flight. It can use its own sensors and cameras to identify the feature points of the space station, establish a three-dimensional model of the space station, achieve autonomous positioning and navigation, and have the ability to avoid obstacles.

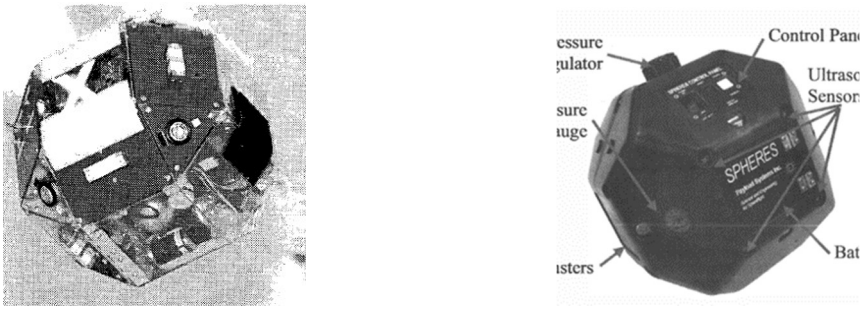


Fig. 2 Sphere

- **Int-ball** (Internal Ball Camera)[3], developed by the Japan Aerospace Exploration Agency (JAXA), is a spherical robot camera that can float inside ISS, and was sent to the International Space Station in June 2017 for preliminary testing. It is composed of 3D printed parts, with a diameter of about 15 cm and a weight of about 1 kg. It can use its built-in camera to take

pictures and videos of the inside of the space station, and perform real-time or offline image processing. It can also use its built-in microphone and speaker to communicate with astronauts by voice, test different human-machine interaction modes and strategies. Its main purpose is to reduce the time spent by astronauts on taking photos and videos, so that they can focus on more important tasks.

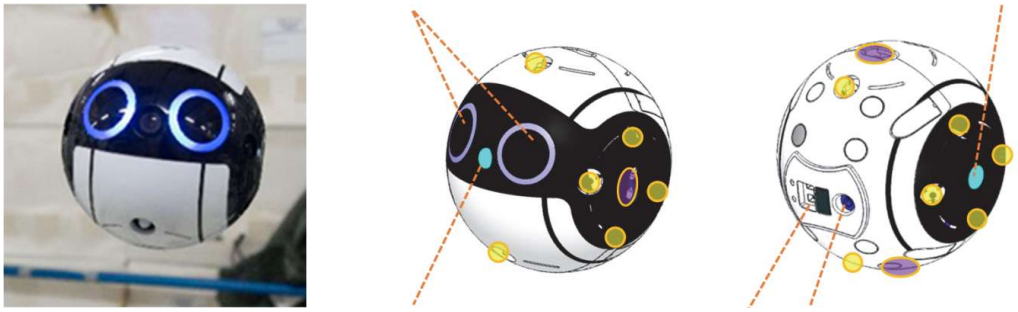


Fig. 3 Int-ball

- **Cimon** (Crew Interactive Mobile Companion), developed by Airbus on behalf of the German Aerospace Center (DLR), is a head-shaped robot that can fly inside ISS, and was sent to the International Space Station in June 2018 for preliminary testing. It is composed of 3D printed parts, roughly spherical in shape, with a diameter of 32 cm and a weight of 5 kg, with a white color overall. It has an 8-inch display screen, which is used to display Cimon's face and visual aids. It also carries multiple cameras and microphones for voice and facial recognition interaction with astronauts. It carries 12 fans that allow it to move and rotate autonomously or underground control in microgravity environments. It uses artificial intelligence technology provided by IBM Watson, which can communicate with astronauts through voice and facial recognition, understand their commands and emotions, try to understand whether astronauts are sad, angry or happy, and then display expressions and animations through the display screen. It can also act as a hands-free database, computer and camera, helping astronauts search for information, record data or take photos, as well as provide scientific knowledge and entertainment

functions for astronauts, and have conversations and interactions with them.

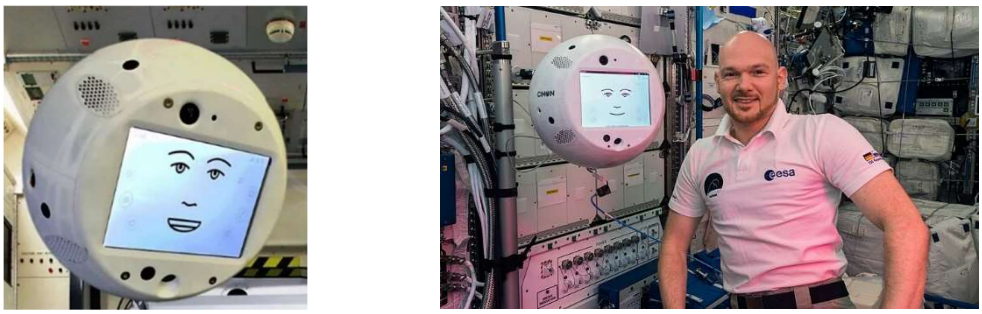


Fig. 4 Cimon

- **Astrobee**, developed by NASA[4]-[7], is a cubic robot that can fly inside ISS. Astrobee has three robots, called Honey, Bumble and Queen, which were sent to the International Space Station in April 2019 and July 2019 respectively. Its shape is a cube, with a side length of 32 cm and a weight of about 7 kg. Each corner of the cube is "wearing" a flexible protective cover. They have built-in cameras and a "docking arm" that can grab items or railings. They are powered by fans and can move autonomously or under the control of ground personnel or astronauts. It can help astronauts with daily tasks such as inventory counting, recording experiments, moving cargo, carrying scientific payloads, collecting data, detecting radiation and carbon dioxide in the space station. It can also transmit real-time images and data of the space cabin, as well as be used for maintaining the space station or providing services for future spacecraft.

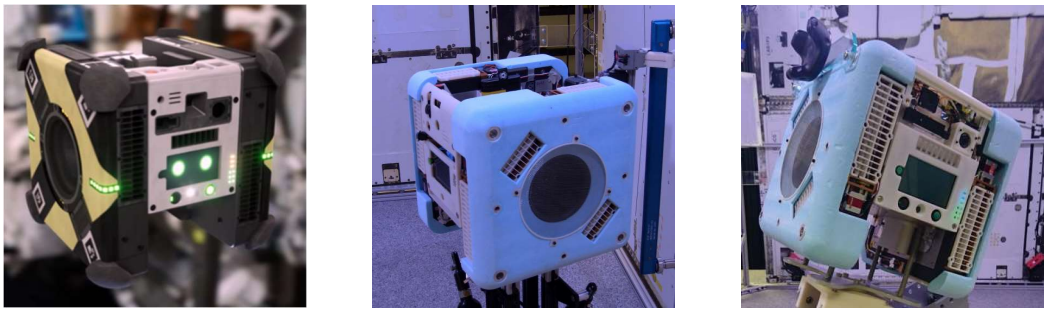


Fig. 5 Astrobee

Judging from the existing design results, free-flying robots are still in the early stages of development.

The relatively elementary Spheres robots only can fly autonomously and finish

some basic tasks.

Int-ball robot can perform simple human-computer interaction and image processing functions, replacing astronauts to complete relatively simple tasks limited by the tiny size;

Cimon robot, like the Int-ball robot, is equipped with many cameras and sensors, and can interact with Astronauts interact with facial recognition, and with the development of artificial intelligence technology, it can initially understand the emotions expressed by astronauts, but it has not solved how to respond to the different emotions of astronauts.

Compared with the different robots introduced before, the technology of Astrobee is more mature. Both the power system and the human-computer interaction system have been greatly innovated, but it ignores the interaction with the astronauts.

1.3 Free-Flying Bifang Robot

In order to provide astronauts with a comfortable and safe living space, and better help astronauts complete in-orbit experiments and tasks in the space station, we designed and developed a single-arm free-flying robot named “Bifang”.

The concept of Bifang comes from the traditional Chinese culture of the Classic of Mountains and Rivers. Bifang is described as a magical creature that has only one foot but can walk freely in the mountains and fly freely in the sky. It is also the guardian bird of Huangdi’s chariot. Bifang with one foot corresponds to the single-arm mechanical cabin assistant robot in the space station, symbolizing that the robot can have similar agility and flexibility as Bifang in the space station. It can also assist astronauts in completing a series of experiments and tasks. It is the guardian bird of astronauts’ chariot in the new era space station.

Fig. 6 Bifang is working in the space station

Bifang can not only help astronauts complete basic tasks such as material handling and object finding, but also judge the psychological activities and mental state of astronauts through facial expression recognition and react intelligently. So that astronauts can ensure their physical and mental health while performing in-orbit tasks.

Bifang is a member of the space team that completes experiments and tasks with astronauts. It is a very powerful "assistant", a trouble-solving "partner" and a trustworthy "doctor" for astronauts in the vast and boundless space. It makes the work and life of astronauts in the space station more convenient, more interesting and safer. It also makes the experiments and tasks performed by astronauts relatively simple, which can shorten the stay period in space and protect the health of astronauts.

In general, Bifang has the following features:

- **Unique variable-speed double-gimbal fans generate large thrust and make Bifang have 6 degrees of freedom, ensuring it can fly around the space station.**
- **Powerful trajectory planning algorithms enable the manipulator to grasp any object the astronauts want with constraints.**
- **Mature emotion recognition algorithms are applied on Bifang to help astronauts live healthier.**
- **Bifang has the ability of visual SLAM mapping, so that it can plan its own trajectory and work smarter.**

2 Description of Bifang

2.1 Appearance

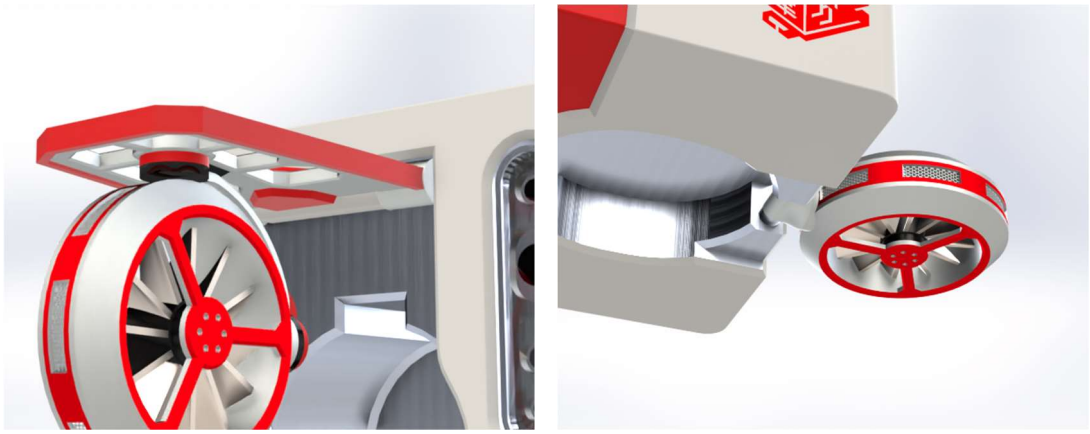
The shape of free-flying Bifang robot is a cube, with a side length of 20cm, and a weight of near 7kg. The main material of Bifang is aluminum plate, which ensures light weight and high strength. In terms of appearance, Bifang has the same design concept as the CubeSat, with a volume of 8U.

Its dynamic system is a group of double-gimbal axial flow fans , which will unfold when it needs to work(shown in Fig. 7), and when it does not need to fly, to reduce the overall volume of the robot.

(a) Folded state **(b) Unfolded state**

Fig. 7 Appearance of Bifang

It contains three fans in total, including a pair of double gimbal fans and an assisted fan (shown in Fig. 8). This unprecedented innovative structural design ensures the robot is controllable in all six degrees of freedom.



(a) Double gimbal fan **(b) Assisted fan**

Fig. 8 Dynamic system of Bifang

In terms of human-computer interaction experience, Bifang contains a display screen in the center, which is used for UI interaction by astronauts. It also carries 4 multiple cameras and microphones for voice and facial recognition interaction with astronauts.

(a) Touch Screen

(b) Multiple Camera

(b) Microphone

Fig. 9 Payload of Bifang

It also contains an independent full-fledged multi-degree-of-freedom Kuka robotic arm (shown in Fig. 10), which can be used to grab floating objects or fixed railings, in order to finish carrying work and self-inhabitation.

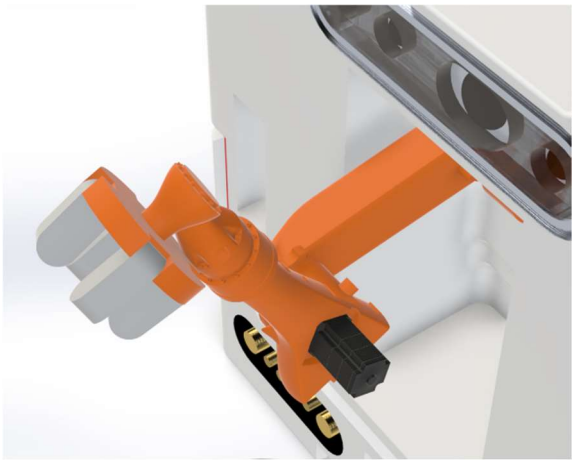


Fig. 10 Kuka robotic arm

The design inspiration of the logo comes from the legendary bird Bifang in the Classic of Mountains and Rivers (shown in Fig. 11), which has only one leg similar to the robotic arm of our designed robot. The two characters of Bifang use the ancient Chinese character writing form of the lesser seal character, to show the charm and attitude of promoting Chinese traditional culture.

Fig. 11 Bifang bird

The overall design of the logo (shown in Fig. 12) integrates the elements of ancient wisdom and advanced space technology, aiming to promote the development of space technology and become a breakthrough in the frontier field. This logo conveys respect for Chinese traditional culture and expresses the pursuit of technological innovation.

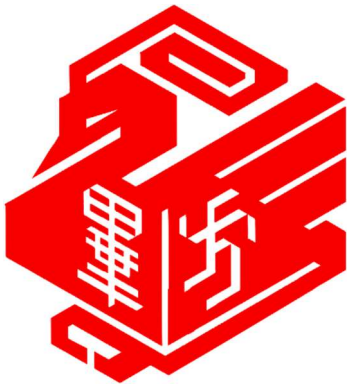


Fig. 12 Logo of Bifang

2.2 Function

As a new generation of cabin robot, in addition to the existing functions mentioned in the literature, Bifang has the following functions:

- **Autonomous and Remote-controlled Flight Capability:** Bifang can use its own sensors and cameras to identify the feature points of the space station (shown Fig. 13), establish a three-dimensional model of the space station, achieve autonomous positioning and navigation, and has the ability to avoid

obstacles by 3D mapping using its camera. It can use sensors and cameras to become an intelligent computer in the space station, helping astronauts search for information, record data or take photos.

Fig. 13 Bifang Flies in the Space Station

- **Grasp and Carry Objects Smarter:** Bifang carries a robotic arm that can grab cargo in the cabin, which can be remotely controlled to complete tasks, transport some lightweight payloads or tools, provide services or deliver items for astronauts, such as inventory counting, recording experiments, transporting cargo, etc.
- **Expression Recognition and Regulate Emotions:** Bifang provides emotional detection and health management for astronauts (shown in Fig. 14). It can use depth cameras to capture, recognize and analyze the expressions of astronauts, judge their psychological and mental state at this moment, recognize various expressions such as happiness, anger, sadness and joy and awake or sleep state shown by astronauts, and make corresponding responses. For example, when it recognizes that an astronaut has a sad or depressed expression, it will chat with the astronaut to relieve their negative emotions and make them feel accompanied. It can also adjust the light to be close to warm colors, while playing some soothing and peaceful music to make astronauts feel comfortable and relaxed.
- **Simple and Efficient UI Interactive Experience:** Bifang is able to interact

intelligently with astronauts and have UI operation functions. It carries a display screen that can be operated by touch. Astronauts can give instructions to the robot as easily as using a mobile phone when they use it. It also has basic semantic recognition capabilities, which can understand the meaning of simple words spoken by astronauts and make corresponding responses, becoming a good partner for astronauts when performing tasks.

Fig. 14 Bifang Detects the Expression

- **Health Management:** Bifang also carries an infrared sensor that can detect body temperature of astronauts. At the same time, it should inquire about the physical condition of astronauts, use relevant medical data in the database to judge their health status, achieve the four steps of “observation”, “auscultation”, “inquiry” and “pulse diagnosis”, becoming a “space doctor” in the space station.

3 Feasibility Analysis

As introduced above, Bifang is a groundbreaking free-flying robot with rich and powerful features. This chapter will introduce the overall design of Bifang, dynamic and control, machine vision applications and the selection of components for each sub-system from the perspective of feasibility.

3.1 Robot Design

According to the technical requirements of each function in the Bifang robot, the overall system framework of the robot is shown as follows:

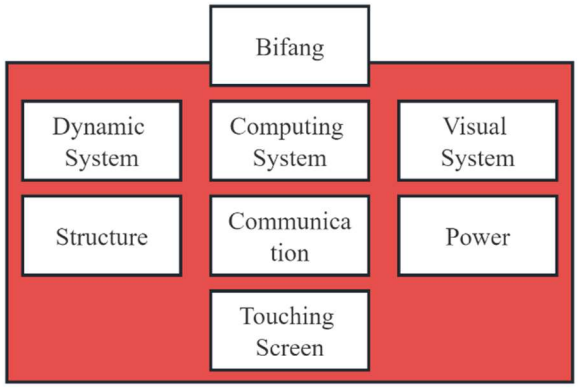


Fig. 15 System Framework of Bifang

- **Dynamic System:** Including the power fans that provides the thrust for the robot to move and the manipulator arm for grasping the item.
- **Computational System:** The processor computes and controls the screen display and dynamics system based on manipulation commands and sensor data.
- **Visual System:** Contains cameras for expression recognition and visual mapping.
- **Structure:** Provides mounting space for other components and provides physical protection..
- **Communication:** Includes a transceiver antenna for receiving signals and transmitting data
- **Power:** Contains batteries, charging devices, and power managers to provide

energy to other subsystems.

- **Touching Screen:** Interact with and provide information to the astronauts.

3.2 Control Design Based on Variable-Speed Double-Gimbal Fan

Because of the unique dynamic design of Bifang, it is necessary to rigorously analyze its controllability in theory, and to conduct simple control simulations on the computer to illustrate the reliability of the dynamic system.

3.2.1 Robot dynamic modelling

First, take the robot as the following single rigid body *B*:

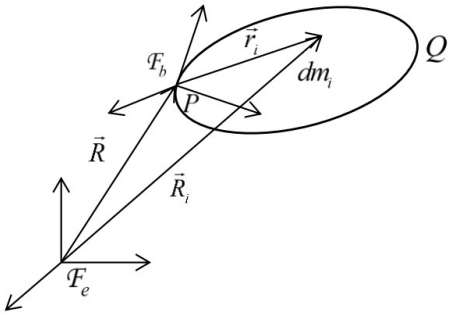


Fig. 16 Single rigid body *B*

Suppose that $\mathcal{F}_b$ is the body-fixed frame with the original point *P*, $\mathcal{F}_e$ is the inertial frame. dm_i is set as the arbitrary point in body *B*, $\vec{R}_i$ and $\vec{r}_i$ are the position of dm_i relative to the original points of frame $\mathcal{F}_e$ and $\mathcal{F}_b$, respectively. Define $\vec{R}$ as the position of point *P* relative to the original point of frame $\mathcal{F}_e$, then it can be obtained as:

$$\vec{R}_i = \vec{R} + \vec{r}_i \tag{1}$$

Take the derivative in frame $\mathcal{F}_e$

$$\vec{V}_i = \vec{V} + \vec{\Omega} \times \vec{r}_i \tag{2}$$

where $\vec{V}_i$ is the velocity of point dm_i relative to frame $\mathcal{F}_e$, $\vec{V}$ and $\vec{\Omega}$ is the relative velocity and angular velocity from frame $\mathcal{F}_e$ to frame $\mathcal{F}_b$, respectively. Define $\boldsymbol{V}$ and $\boldsymbol{\Omega}$ as the above velocity and angular velocity represented in frame $\mathcal{F}_b$, then Eq. (2) can be written as:

$$\vec{V}_i = \boldsymbol{e}_b^T \boldsymbol{V} - \boldsymbol{e}_b^T \tilde{\boldsymbol{r}}_i \boldsymbol{\Omega} \tag{3}$$

Define general velocity as $\boldsymbol{u}^T = \begin{bmatrix} \boldsymbol{V}^T & \boldsymbol{\Omega}^T \end{bmatrix}$, then the partial velocity matrix can be obtained as

$$\begin{aligned} {}^p\boldsymbol{V}_{i,1} &= \boldsymbol{e}_b^T \\ {}^p\boldsymbol{V}_{i,2} &= -\boldsymbol{e}_b^T \tilde{\boldsymbol{r}}_i \end{aligned} \tag{4}$$

Take the derivative of Eq. (3) in frame $\mathcal{F}_e$:

$$\dot{\tilde{V}}_i = \boldsymbol{e}_b^T \dot{\boldsymbol{V}} + \boldsymbol{e}_b^T \tilde{\boldsymbol{\Omega}} \boldsymbol{V} - \boldsymbol{e}_b^T \tilde{\boldsymbol{r}}_i \dot{\boldsymbol{\Omega}} - \boldsymbol{e}_b^T \tilde{\boldsymbol{\Omega}} \tilde{\boldsymbol{r}}_i \boldsymbol{\Omega} \tag{5}$$

Substituting Eqs. (4) and (5), the motion equation of single rigid body can be simplified by Kane's method[1]:

$$\begin{bmatrix} m\boldsymbol{I}_3 & -\tilde{\boldsymbol{S}} \\ -\tilde{\boldsymbol{S}}^T & \boldsymbol{J} \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{V}} \\ \dot{\boldsymbol{\Omega}} \end{bmatrix} + \begin{bmatrix} \tilde{\boldsymbol{\Omega}} \tilde{\boldsymbol{\Omega}} \boldsymbol{S} + m \tilde{\boldsymbol{\Omega}} \boldsymbol{V} \\ \tilde{\boldsymbol{\Omega}} \boldsymbol{J} \boldsymbol{\Omega} + \tilde{\boldsymbol{S}} \tilde{\boldsymbol{\Omega}} \boldsymbol{V} \end{bmatrix} = \begin{bmatrix} \boldsymbol{F}_p \\ \boldsymbol{T}_p \end{bmatrix} + \begin{bmatrix} \boldsymbol{I}_3 & \boldsymbol{0}_3 \\ \tilde{\boldsymbol{r}}_{P,Q} & \boldsymbol{I}_3 \end{bmatrix} \begin{bmatrix} \boldsymbol{F}_Q \\ \boldsymbol{T}_Q \end{bmatrix} \tag{6}$$

where $\boldsymbol{S} = \int \boldsymbol{r}_i dm_i$ and $\boldsymbol{J} = \int \tilde{\boldsymbol{r}}_i^T \tilde{\boldsymbol{r}}_i dm_i$ are the first moment and inertia matrix of body B represented in frame $\mathcal{F}_b$, respectively. $\boldsymbol{F}_P, \boldsymbol{T}_P, \boldsymbol{F}_Q, \boldsymbol{T}_Q$ are the active forces and torques applied to point P and Q, respectively, and $\boldsymbol{r}_{P,Q}$ is the vector from point P to point Q. It should be noted that in our case, the point P is set in the center of mass of body B, therefore first moment $\boldsymbol{S} = \boldsymbol{0}$. Considering that three variable-speed fans are applied, there are three points Q with constant vectors $\boldsymbol{r}_1, \boldsymbol{r}_2, \boldsymbol{r}_3$. Besides, the only inputs are the forces of variable-speed fans, i.e.,

$$\boldsymbol{F}_P = \boldsymbol{T}_P = \boldsymbol{0}, \boldsymbol{T}_{Q1} = \boldsymbol{T}_{Q2} = \boldsymbol{T}_{Q3} = \boldsymbol{0} \tag{7}$$

Finally, the motion equation of the robot can be simplified as

$$\begin{bmatrix} m\boldsymbol{I}_3 & \\ & \boldsymbol{J} \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{V}} \\ \dot{\boldsymbol{\Omega}} \end{bmatrix} + \begin{bmatrix} m \tilde{\boldsymbol{\Omega}} \boldsymbol{V} \\ \tilde{\boldsymbol{\Omega}} \boldsymbol{J} \boldsymbol{\Omega} \end{bmatrix} = \begin{bmatrix} \boldsymbol{F}_1 + \boldsymbol{F}_2 + \boldsymbol{F}_3 \\ \tilde{\boldsymbol{r}}_1 \boldsymbol{F}_1 + \tilde{\boldsymbol{r}}_2 \boldsymbol{F}_2 + \tilde{\boldsymbol{r}}_3 \boldsymbol{F}_3 \end{bmatrix} \tag{8}$$

Define $\boldsymbol{M} = \begin{bmatrix} m\boldsymbol{I}_3 & \\ & \boldsymbol{J} \end{bmatrix}$, $\boldsymbol{F}^I = \begin{bmatrix} m \tilde{\boldsymbol{\Omega}} \boldsymbol{V} \\ \tilde{\boldsymbol{\Omega}} \boldsymbol{J} \boldsymbol{\Omega} \end{bmatrix}$, and $\boldsymbol{q} = \begin{bmatrix} \boldsymbol{F}_1 + \boldsymbol{F}_2 + \boldsymbol{F}_3 \\ \tilde{\boldsymbol{r}}_1 \boldsymbol{F}_1 + \tilde{\boldsymbol{r}}_2 \boldsymbol{F}_2 + \tilde{\boldsymbol{r}}_3 \boldsymbol{F}_3 \end{bmatrix}$, then Eq. (8) can also be expressed as:

$$\boldsymbol{M} \dot{\boldsymbol{u}} + \boldsymbol{F}^I = \boldsymbol{q} \tag{9}$$

3.2.2 Robot kinematic modelling

The attitude of robot can be represented with Euler angle. Define the roll, pitch and yaw angles from the frame $\mathcal{F}_b$ to the frame $\mathcal{F}_e$ as φ, θ, and ψ, with rotational order 3-1-2. Then the attitude relationship can be written as

$$\boldsymbol{\Omega} = \begin{bmatrix} \Omega_x \\ \Omega_y \\ \Omega_z \end{bmatrix} = \begin{bmatrix} \dot{\phi}\cos\theta - \dot{\psi}\cos\phi\sin\theta \\ \dot{\theta} + \dot{\psi}\sin\phi \\ \dot{\phi}\sin\theta + \dot{\psi}\cos\phi\cos\theta \end{bmatrix} \tag{10}$$

Define $\boldsymbol{\Theta} = \begin{bmatrix} \phi & \theta & \psi \end{bmatrix}^T$, Eq. (10) can be modified as

$$\boldsymbol{\Omega} = \boldsymbol{T}(\boldsymbol{\Theta})\dot{\boldsymbol{\Theta}} \tag{11}$$

where $\boldsymbol{T}(\boldsymbol{\Theta}) = \begin{bmatrix} \cos\theta & 0 & -\cos\phi\sin\theta \\ 0 & 1 & \sin\phi \\ \sin\theta & 0 & \cos\phi\cos\theta \end{bmatrix}$ is the coefficient matrix. The derivative of Eq. (11) can be represented as

$$\dot{\boldsymbol{\Omega}} = \boldsymbol{T}(\boldsymbol{\Theta})\ddot{\boldsymbol{\Theta}} + \boldsymbol{G}(\boldsymbol{\Theta})\dot{\boldsymbol{\Theta}}_{cp} \tag{12}$$

where

$$\boldsymbol{G}(\boldsymbol{\Theta}) = \begin{bmatrix} -\sin\theta & -\cos\phi\cos\theta & \sin\phi\sin\theta \\ 0 & 0 & \cos\phi \\ \cos\theta & -\cos\phi\sin\theta & -\sin\phi\cos\theta \end{bmatrix}, \qquad \dot{\boldsymbol{\Theta}}_{cp} = \begin{bmatrix} \dot{\phi}\dot{\theta} \\ \dot{\theta}\dot{\psi} \\ \dot{\phi}\dot{\psi} \end{bmatrix} \tag{13}$$

It should be noted that the rotation transform matrix $\boldsymbol{A}_{be}$ from the frame $\mathcal{F}_e$ to the frame $\mathcal{F}_b$ can also be represented by Euler angles as

$$\boldsymbol{A}_{be} = \begin{bmatrix} \mathrm{c}\theta\mathrm{c}\psi - \mathrm{s}\phi\mathrm{s}\theta\mathrm{s}\psi & \mathrm{c}\theta\mathrm{s}\psi + \mathrm{s}\phi\mathrm{s}\theta\mathrm{c}\psi & -\mathrm{s}\theta\mathrm{c}\phi \\ -\mathrm{s}\psi\mathrm{c}\phi & \mathrm{c}\phi\mathrm{c}\psi & \mathrm{s}\phi \\ \mathrm{s}\theta\mathrm{c}\psi + \mathrm{s}\phi\mathrm{s}\psi\mathrm{c}\theta & \mathrm{s}\theta\mathrm{s}\psi - \mathrm{s}\phi\mathrm{c}\theta\mathrm{c}\psi & \mathrm{c}\phi\mathrm{c}\theta \end{bmatrix} \tag{14}$$

where symbol “c” means function “cos”, and symbol “s” means function “sin”.

3.2.3 Control design

PID control[9] is one of the classic controls, which adjusts the actual motion through negative feedback to converge to the desired motion. Although PID control is theoretically only suitable for linear systems, it can also be applied to nonlinear systems

in engineering after some slight modifications.

The designed PID control can be expressed as:

$$\boldsymbol{q} = \boldsymbol{F}^{I} + \boldsymbol{M}\left(\dot{\boldsymbol{u}}_d - k_p\tilde{\boldsymbol{u}} - k_i\int\tilde{\boldsymbol{u}}\right) \tag{15}$$

where $\dot{\boldsymbol{u}}_d$ is the desired acceleration, $\tilde{\boldsymbol{u}} = \boldsymbol{u} - \boldsymbol{u}_d$ is the tracking error of joint angle, k_p and k_i are the control parameters, respectively. It should be noted that the desired angle, desired angle velocity, and desired angle acceleration are all be calculated by the trajectory planning module.

Substituting Eq. (15) into Eq. (9), it can be obtained that:

$$\dot{\tilde{\boldsymbol{u}}} + k_p\tilde{\boldsymbol{u}} + k_i\int\tilde{\boldsymbol{u}} = \boldsymbol{0} \tag{16}$$

Or it can be expressed as:

$$\dot{\tilde{u}}_i + k_p\tilde{u}_i + k_i\int\tilde{u}_i = 0 \quad (i = 1, 2, \cdots, n) \tag{17}$$

Since each equation in Eq. (17) is the second-order control model, the tracking error will be asymptotically converged to zero by selecting appropriate control parameters k_p and k_i. Therefore, the designed control can force the robot to tracking the desired trajectory and meet the desired performance requirements.

3.2.4 Allocation of rotational angle and angle velocity for three fans

The structure of the variable-speed double-gimbal[10] fan is shown as follows:

Fig. 17 Structure of variable-speed double-gimbal fan

It can be seen that the force generated by the fan is controlled by two rotation angles α and β. Suppose that the force F can be expressed as:

$$F = k\omega^2 \tag{18}$$

where ω is the angle velocity of the fan. Therefore, the control forces of two fans can be obtained in the frame $\mathcal{F}_b$ as:

$$\boldsymbol{F}_1 = \boldsymbol{L}_z(\alpha_1)\boldsymbol{L}_y(\beta_1)\begin{bmatrix} k_1\omega_1^2 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} \cos\alpha_1\cos\beta_1 \\ -\sin\alpha_1\cos\beta_1 \\ \sin\beta_1 \end{bmatrix} k_1\omega_1^2 \tag{19}$$

$$\boldsymbol{F}_2 = \boldsymbol{L}_z(\alpha_2)\boldsymbol{L}_y(\beta_2)\begin{bmatrix} k_2\omega_2^2 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} \cos\alpha_2\cos\beta_2 \\ -\sin\alpha_2\cos\beta_2 \\ \sin\beta_2 \end{bmatrix} k_2\omega_2^2 \tag{20}$$

where $\boldsymbol{L}_z(\alpha) = \begin{bmatrix} \cos\alpha & \sin\alpha & \\ -\sin\alpha & \cos\alpha & \\ & & 1 \end{bmatrix}$, and $\boldsymbol{L}_y(\beta) = \begin{bmatrix} \cos\beta & & -\sin\beta \\ & 1 & \\ \sin\beta & & \cos\beta \end{bmatrix}$ are the unit rotation matrix around *z*-axis and *y*-axis, respectively.

The vector $\boldsymbol{r}$ of two variable-speed double-gimbal fans are designed as:

$$\boldsymbol{r}_1 = \begin{bmatrix} x & y & 0 \end{bmatrix}^T, \quad \boldsymbol{r}_2 = \begin{bmatrix} x & -y & 0 \end{bmatrix}^T \tag{21}$$

Then the control torques can be expressed in the frame $\mathcal{F}_b$ as

$$\boldsymbol{T}_1 = \tilde{\boldsymbol{r}}_1\boldsymbol{F}_1 = \begin{bmatrix} y\sin\beta_1 \\ -x\sin\beta_1 \\ -y\cos\alpha_1\cos\beta_1 - x\sin\alpha_1\cos\beta_1 \end{bmatrix} k_1\omega_1^2 \tag{22}$$

$$\boldsymbol{T}_2 = \tilde{\boldsymbol{r}}_2\boldsymbol{F}_2 = \begin{bmatrix} -y\sin\beta_2 \\ -x\sin\beta_2 \\ y\cos\alpha_2\cos\beta_2 - x\sin\alpha_2\cos\beta_2 \end{bmatrix} k_2\omega_2^2 \tag{23}$$

where $\tilde{\boldsymbol{r}} = \begin{bmatrix} & -z & y \\ z & & -x \\ -y & x & \end{bmatrix}$ is the cross product antisymmetric matrix.

The third fan is fixed without rotation angle. Suppose the position $\boldsymbol{r}_3 = \begin{bmatrix} z & 0 & z \end{bmatrix}^T$, then the force and torque can be expressed in the frame $\mathcal{F}_b$ as:

$$\boldsymbol{F}_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} k_3\omega_3^2, \quad \boldsymbol{T}_3 = \tilde{\boldsymbol{r}}_3\boldsymbol{F}_3 = -\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} zk_3\omega_3^2 \tag{24}$$

In conclusion, the total control forces and torques generated by the fans can be summarized as:

$$\boldsymbol{q} = \begin{bmatrix} \cos\alpha_1\cos\beta_1 k_1\omega_1^2 + \cos\alpha_2\cos\beta_2 k_2\omega_2^2 \\ -\sin\alpha_1\cos\beta_1 k_1\omega_1^2 - \sin\alpha_2\cos\beta_2 k_2\omega_2^2 \\ \sin\beta_1 k_1\omega_1^2 + \sin\beta_2 k_2\omega_2^2 + k_3\omega_3^2 \\ y\sin\beta_1 k_1\omega_1^2 - y\sin\beta_2 k_2\omega_2^2 \\ -x\sin\beta_1 k_1\omega_1^2 - x\sin\beta_2 k_2\omega_2^2 - zk_3\omega_3^2 \\ -\left(y\cos\alpha_1\cos\beta_1 + x\sin\alpha_1\cos\beta_1\right)k_1\omega_1^2 + \left(y\cos\alpha_2\cos\beta_2 - x\sin\alpha_2\cos\beta_2\right)k_2\omega_2^2 \end{bmatrix} \tag{25}$$

Define the controlled variables $\boldsymbol{\theta}$ as:

$$\boldsymbol{\theta} = \begin{bmatrix} \cos\alpha_1\cos\beta_1 k_1\omega_1^2 \\ \sin\alpha_1\cos\beta_1 k_1\omega_1^2 \\ \sin\beta_1 k_1\omega_1^2 \\ \cos\alpha_2\cos\beta_2 k_2\omega_2^2 \\ \sin\alpha_2\cos\beta_2 k_2\omega_2^2 \\ \sin\beta_2 k_2\omega_2^2 \\ k_3\omega_3^2 \end{bmatrix} \tag{26}$$

Then Eq.(25) can be modified as:

$$\boldsymbol{q} = \boldsymbol{G}\boldsymbol{\theta} \tag{27}$$

Where $\boldsymbol{G} = \begin{bmatrix} 1 & & & 1 & & & \\ & -1 & & & -1 & & \\ & & 1 & & & 1 & 1 \\ & & y & & & -y & \\ & & -x & & & -x & -z \\ -y & -x & & y & -x & & \end{bmatrix}$ is the coefficient matrix.

For appropriate selection of x, y, and z, $\boldsymbol{G}$ is invertible. Therefore:

$$\begin{bmatrix} \cos\alpha_1 \cos\beta_1 k_1\omega_1^2 \\ \sin\alpha_1 \cos\beta_1 k_1\omega_1^2 \\ \sin\beta_1 k_1\omega_1^2 \\ \cos\alpha_2 \cos\beta_2 k_2\omega_2^2 \\ \sin\alpha_2 \cos\beta_2 k_2\omega_2^2 \\ \sin\beta_2 k_2\omega_2^2 \\ k_3\omega_3^2 \end{bmatrix} = \boldsymbol{\theta} = \boldsymbol{G}^{+}\boldsymbol{q} \triangleq \begin{bmatrix} F_{x1} \\ F_{y1} \\ F_{z1} \\ F_{x2} \\ F_{y2} \\ F_{z2} \\ F_{z3} \end{bmatrix} \tag{28}$$

where $\boldsymbol{G}^{+}$ denotes the pseudo-inverse of $\boldsymbol{G}$:

$$\boldsymbol{G}^{+} = \boldsymbol{G}^{T}\left(\boldsymbol{G}\boldsymbol{G}^{T}\right)^{-1} \tag{29}$$

Solve Eq.(28), it can be obtained that:

$$\begin{cases} k_1\omega_1^2 = \sqrt{F_{x1}^2 + F_{y1}^2 + F_{z1}^2} \\ k_2\omega_2^2 = \sqrt{F_{x2}^2 + F_{y2}^2 + F_{z2}^2} \\ k_3\omega_3^2 = F_{z3} \end{cases} \tag{30}$$

$$\begin{cases} \alpha_1 = \text{atan2}\left(F_{y1}, F_{x1}\right) \\ \beta_1 = \text{atan2}\left(F_{z1}, F_{y1}/\sin\alpha_1\right) \end{cases} \tag{31}$$

$$\begin{cases} \alpha_2 = \text{atan2}\left(F_{y2}, F_{x2}\right) \\ \beta_2 = \text{atan2}\left(F_{z2}, F_{y2}/\sin\alpha_2\right) \end{cases} \tag{32}$$

Eqs.(30), (31) and (32) together form the allocation of rotational angle and angle velocity for three fans.

3.2.5 Numerical simulation and analysis

Take Fig. 17 as the model of robot and variable-speed fans, the according parameters are set as follows:

$$m = 10\,,\ \ \boldsymbol{I} = \text{diag}\left\{\frac{1}{15}, \frac{1}{10}, \frac{1}{12}\right\},\ \ x = 0.04\,,\ \ y = 0.17\,,\ \ z = -0.1 \tag{33}$$

The desired trajectory is designed as follows:

$$\boldsymbol{r}_d = \begin{bmatrix} R\cos\omega t - R & R\sin\omega t & 0.01t \end{bmatrix}^{T},\ \ \varphi_d = \theta_d = \psi_d = 0 \tag{34}$$

where $R = 5$ denotes the radius, and $\omega = 1$ denotes the frequency.

The entire algorithm is demonstrated in the Simulink platform:

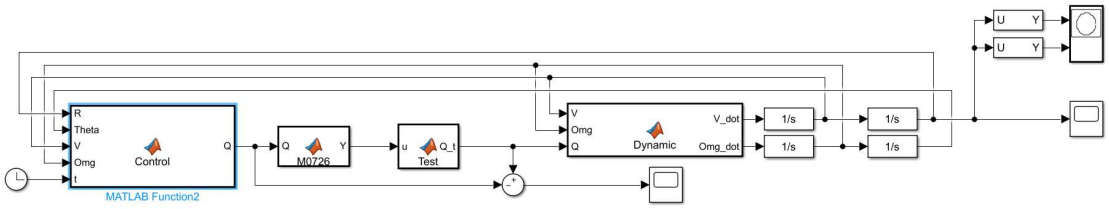


Fig. 18 Algorithm designed in the Simulink

where the fourth Runge-Kutta formula is used with simulation step $t = 0.01\text{s}$ and total simulation time $t = 100\text{s}$.

The control parameters in Section 3.2.3 is selected as:

$$k_p = 2, \quad k_i = 1 \tag{35}$$

and the whole system is set as zero-initial status, which means for the initial time $t_0 = 0$, the system is:

$$k_1\omega_1^2 = k_2\omega_2^2 = k_3\omega_3^2 = \alpha_1 = \alpha_2 = \beta_1 = \beta_2 = 0 \tag{36}$$

$$\dot{\omega}_1 = \dot{\omega}_2 = \dot{\omega}_3 = \dot{\alpha}_1 = \dot{\alpha}_2 = \dot{\beta}_1 = \dot{\beta}_2 = 0 \tag{37}$$

The simulation results are shown as follows:

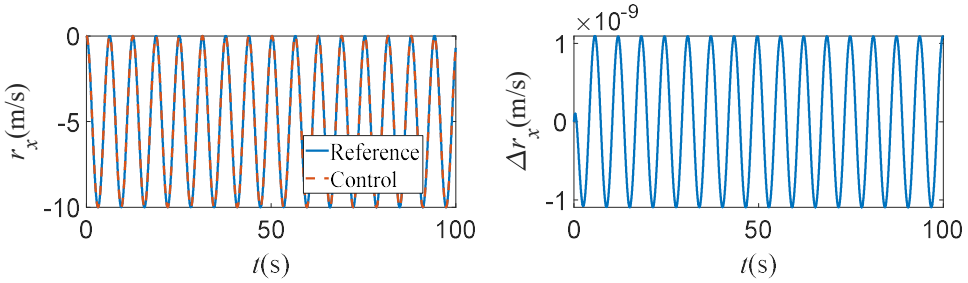


Fig. 19 Reference and control trajectory of r_x

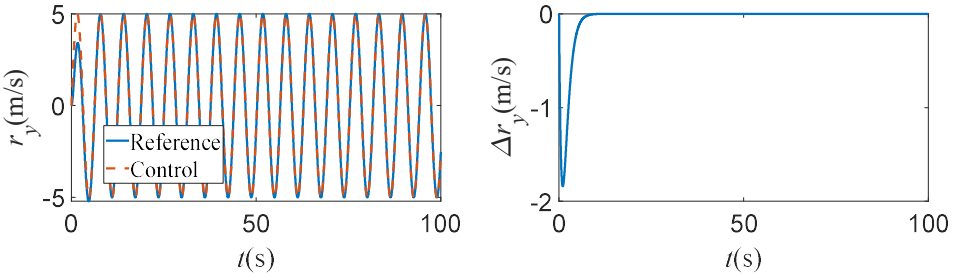


Fig. 20 Reference and control trajectory of r_y

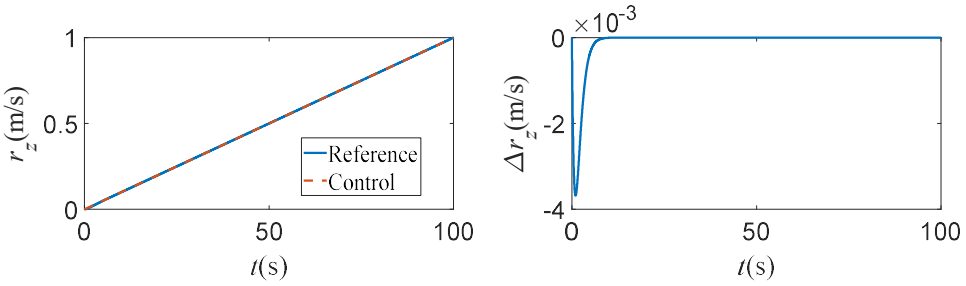


Fig. 21 Reference and control trajectory of r_z

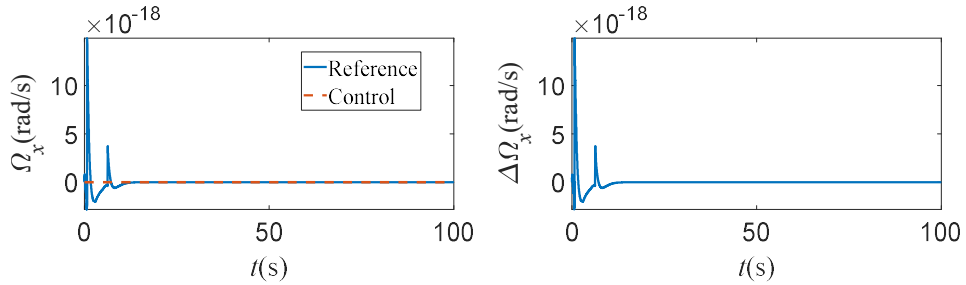


Fig. 22 Reference and control trajectory of Ω_x

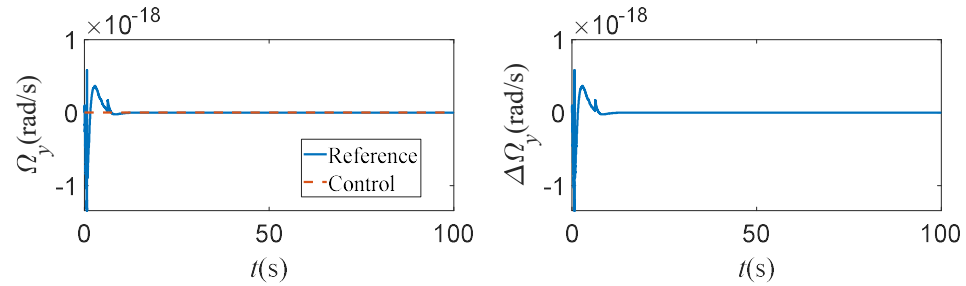


Fig. 23 Reference and control trajectory of Ω_y

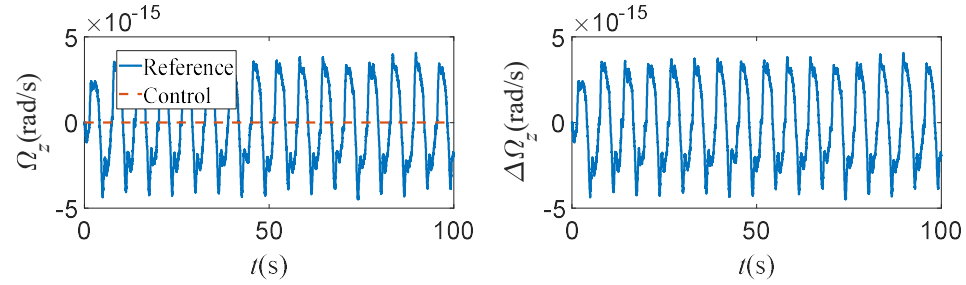


Fig. 24 Reference and control trajectory of Ω_y

From Fig. 19 to Fig. 24, it can be seen that the control trajectory tracks the reference trajectory with error smaller than 10^{-3}. The translation in control trajectory can be plotted in 3D form as:

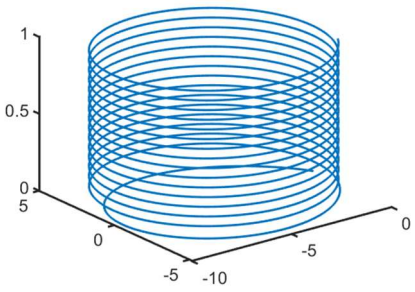


Fig. 25 Translation in control trajectory

where the translation trajectory can be regarded as the spiral upward curve, which demonstrates that the effectiveness of designed controller.

Moreover, the control forces and torques are actually generated by three variable-speed fans with some rotation angles. During the simulation, the parameter $\boldsymbol{\theta}$ in previous section are shown as follows:

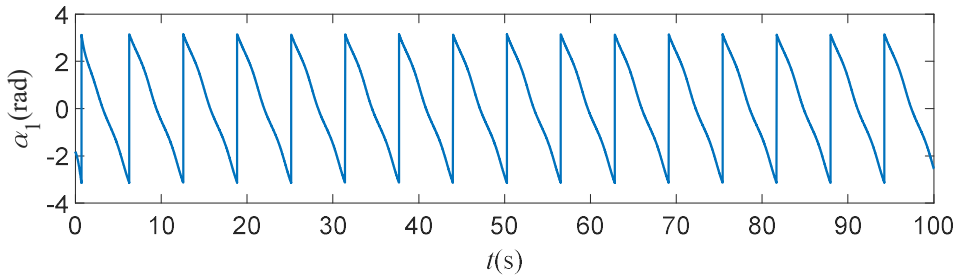


Fig. 26 Motion of rotation angle α_1

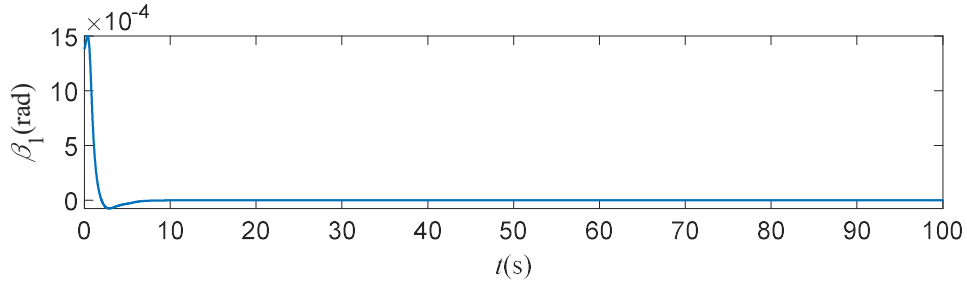


Fig. 27 Motion of rotation angle β_1

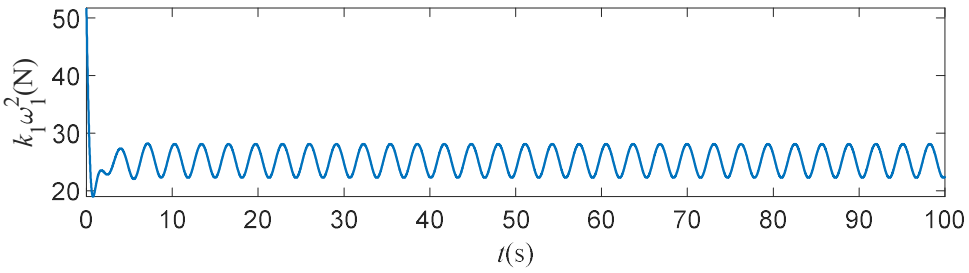


Fig. 28 Motion of first fan $k_1\omega_1^2$

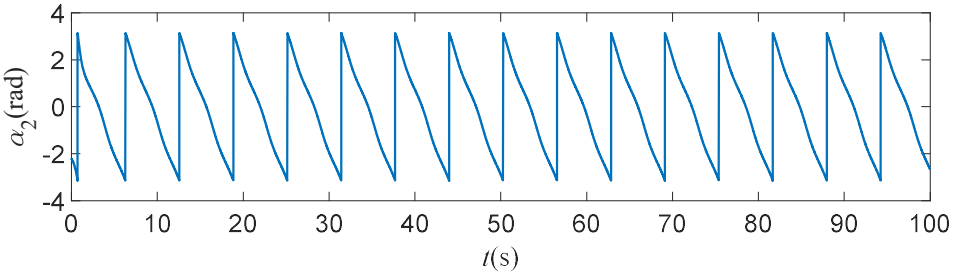


Fig. 29 Motion of rotation angle α_2

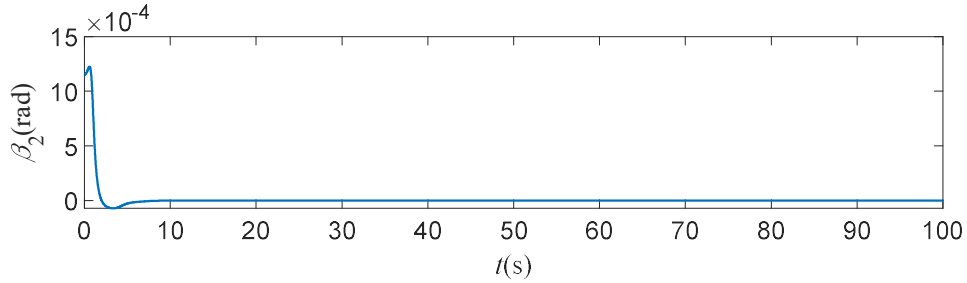


Fig. 30 Motion of rotation angle β_2

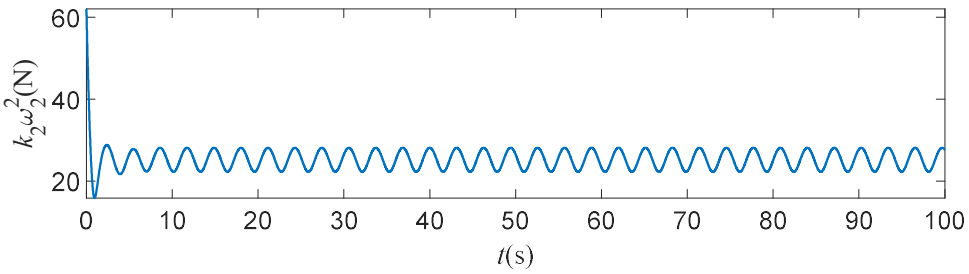


Fig. 31 Motion of second fan $k_2\omega_2^2$

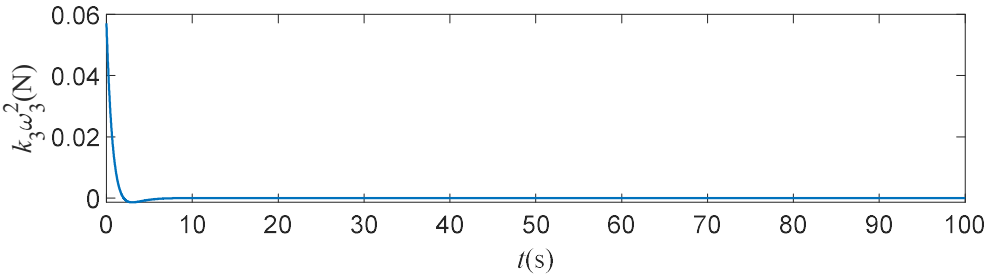


Fig. 32 Motion of third fan $k_3\omega_3^2$

To demonstrate the designed allocation algorithm in previous section, the reference control force and the force calculated from the allocated angle are compared as follows:

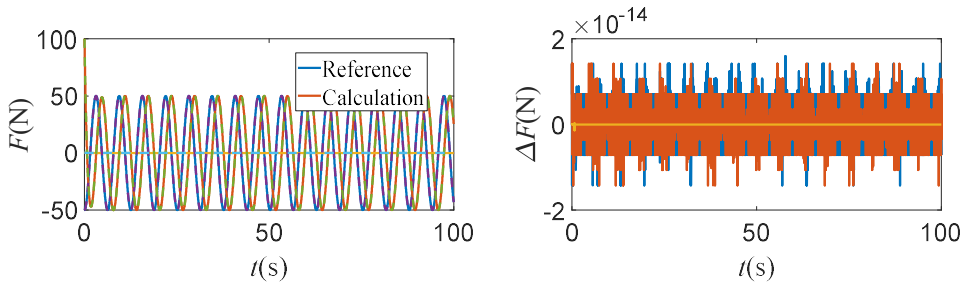


Fig. 33 Comparison of reference force and calculation forces

It can be seen that the reference forces and the calculation forces are almost the same, with the error small than 10^{-14}. Similarly, the allocation torques are compared with reference control torques, which are shown as follows:

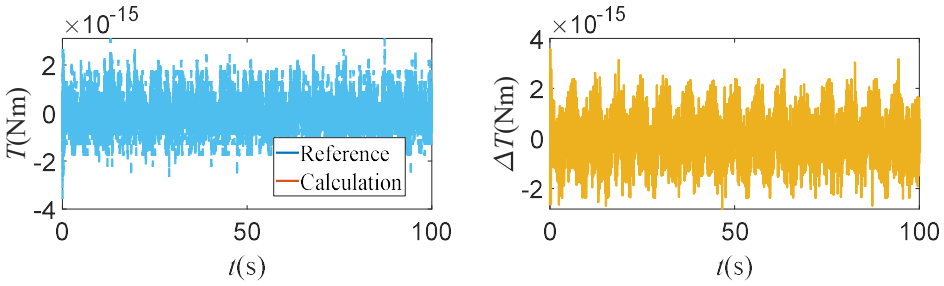


Fig. 34 Comparison of reference force and calculation torques

Since the desired motion of rotation is zeros, the control torques maintain zero, together with allocation torques. Therefore, it can be concluded the correctness of the designed allocation algorithm can be demonstrated.

3.3 Trajectory Planning of Robot Manipulator

The manipulator is an important part of Bifang and is used to carry out many works.

Therefore, we have carried out theoretical research and simulation analysis on trajectory planning and obstacle avoidance algorithm of the manipulator to ensure its functionality.

3.3.1 Kinematic modelling of manipulator

The establishment of the kinematics of the manipulator involves the transformation between several coordinate frames. These frames are one base frame, body-fixed frame of each joint, and the Cartesian coordinate frame of the end of the manipulator. Firstly, the coordinate frames are established by D-H method[11]:

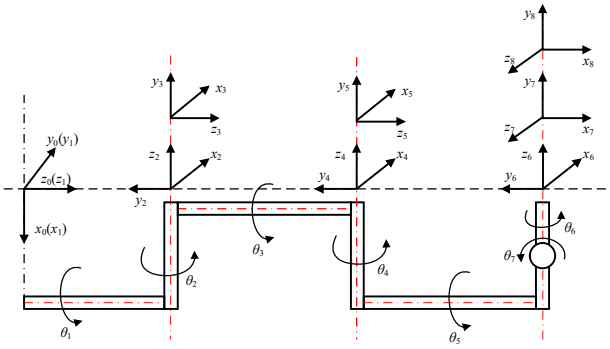


Fig. 35 Reference frame of manipulator

Specifically, D-H method can used to describe the relationship of the motion between manipulator arms. For *i*-th arm, the length, the rotational angle, and the offset of each arm are needed.

3.3.1.1 Forward kinematics analysis

Forward kinematic is applied to solve the pose of the end of the manipulator in the frame of the base, with the known rotational angle of each arm. Define the rotation matrix from frame *i* to frame *i*-1 as ${}^{i-1}\boldsymbol{T}_i$, the coordinate transform can be performed by four steps, which are shown as follows:

1. Rotate with axis x_{i-1} by angle α_{i-1}.
2. Transform along axis x_{i-1} by distance a_{i-1}.
3. Rotate with axis z_i by angle θ_i.
4. Transform along axis z_i by distance d_i.

Follow the above transform step, then the transform matrix ${}^{i-1}\boldsymbol{T}_i$ can be expressed as:

$$ {}^{i-1}\boldsymbol{T}_i = \mathbf{Rot}(x,\alpha_{i-1})\mathbf{Trans}(x,a_{i-1})\mathbf{Rot}(z,\theta_i)\mathbf{Trans}(z,d_i) \tag{38} $$

where **Rot** and **Trans** denote the attitude and translation transform matrix, respectively. Substituting the detailed parameters into Eq. (38) results:

$$ {}^{i-1}\boldsymbol{T}_i = \begin{bmatrix} \mathrm{c}\theta_i & -\mathrm{s}\theta_i & 0 & a_{i-1} \\ \mathrm{s}\theta_i\mathrm{c}\alpha_{i-1} & \mathrm{c}\theta_i\mathrm{c}\alpha_{i-1} & -\mathrm{s}\alpha_{i-1} & -d_i\mathrm{s}\alpha_{i-1} \\ \mathrm{s}\theta_i\mathrm{s}\alpha_{i-1} & \mathrm{c}\theta_i\mathrm{s}\alpha_{i-1} & \mathrm{c}\alpha_{i-1} & d_i\mathrm{c}\alpha_{i-1} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{39} $$

where $\mathrm{s}\theta$ and $\mathrm{c}\theta$ denote the function $\sin\theta$ and $\cos\theta$, respectively. Multiplying each transform matrix leads to the entire transform matrix from the end frame to the base frame, which is the base of the forward kinematic analysis.

3.3.1.2 Inverse kinematic analysis

Inverse kinematic analysis is used to solve the angle of each arm based on the known pose transform matrix ${}^{B}_{T}\boldsymbol{T}$. For non-redundant manipulator, Jacobian matrix can project the velocity and the angle acceleration of the end of the manipulator into the joint space, and then each joint angle can be solved by integration. Specifically,

$$ \dot{\boldsymbol{q}} = \boldsymbol{J}^{-1}\begin{bmatrix} \dot{\boldsymbol{p}} \\ \boldsymbol{\omega} \end{bmatrix} \tag{40} $$

where $\dot{\boldsymbol{q}}$ denotes the angle velocity of each joint. $\boldsymbol{J}$ means the Jacobian matrix, which is always invertible at the nonsingular case. $\dot{\boldsymbol{p}}$ and $\boldsymbol{\omega}$ are the velocity and the angular velocity of the end of the manipulator, respectively. After that, the rotation angle of each joint can be solved by the integration of rotation angle acceleration.

For redundant manipulator, since the number of joint angle is larger than the number of the motion constraints, its inverse kinematic analysis becomes the uncertain problem. The common way of inverse kinematic solution is gradient projection method[12], which is expressed as:

$$\dot{\boldsymbol{q}} = \boldsymbol{J}^{+}\begin{bmatrix}\dot{\boldsymbol{p}}\\ \boldsymbol{\omega}\end{bmatrix} + k(\boldsymbol{I} - \boldsymbol{J}^{+}\boldsymbol{J})\nabla \boldsymbol{H}_{\text{Opt}}(\boldsymbol{q}) \tag{41}$$

where $\boldsymbol{J}^{+}[\dot{\boldsymbol{p}} \quad \boldsymbol{\omega}]^{\mathrm{T}}$ is the minimum-norm solution of joint angular velocity, and $k(\boldsymbol{I} - \boldsymbol{J}^{+}\boldsymbol{J})\nabla \boldsymbol{H}_{\text{Opt}}(\boldsymbol{q})$ is the corresponding homogeneous solution of equation $\boldsymbol{J}\dot{\boldsymbol{q}} = \boldsymbol{0}$. $\boldsymbol{I}$ denotes the unit matrix, ∇ the gradient operator, and $\boldsymbol{J}^{+}$ the pseudo inverse of Jacobi matrix, which can be shown as

$$\nabla \boldsymbol{H}_{\text{Opt}}(\boldsymbol{q}) = \begin{bmatrix}\dfrac{\partial \boldsymbol{H}_{\text{Opt}}}{\partial \theta_1} & \dfrac{\partial \boldsymbol{H}_{\text{Opt}}}{\partial \theta_2} & \cdots & \dfrac{\partial \boldsymbol{H}_{\text{Opt}}}{\partial \theta_7}\end{bmatrix} \tag{42}$$

$$\boldsymbol{J}^{+} = \boldsymbol{J}^{\mathrm{T}}(\boldsymbol{J}\boldsymbol{J}^{\mathrm{T}})^{-1} \tag{43}$$

Since the minimum norm solution is unique, there always exists the only optimal joint angles to minimize the index function and to reach desired pose simultaneously. It should be noted that the angle by gradient projection method is approximate, cumulative error cannot be ignored when the trajectory is long.

3.3.2 Trajectory planning method

3.3.2.1 Interpolation algorithm

The initial and final pose of the trajectory planning are given in the Cartesian space, while the manipulator moves in the joint space. Therefore, it is necessary to transform the pose of the end of the manipulator into the angles of each joint with the smooth interpolation function.

One of the most effective and widely used interpolation methods is polynomial interpolation method[13]. During the manipulator movement, each joint is constrained by 6 constraints in total, i.e., the desired joint angle, velocity and acceleration, a total of 6 constraints. Therefore, the quintic polynomial interpolation is usually selected. Suppose that the function of the joint angle is:

$$\theta(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5 \tag{44}$$

where the constraints are expressed as:

$$\begin{cases} \theta(0) = a_0 \\ \theta(t_f) = a_0 + a_1 t_f + a_2 t_f^2 + a_3 t_f^3 + a_4 t_f^4 + a_5 t_f^5 \\ \dot{\theta}(0) = a_1 \\ \dot{\theta}\left(t_f\right) = a_1 + 2a_2 t_f + 3a_3 t_f^2 + 4a_4 t_f^3 + 5a_5 t_f^4 \\ \ddot{\theta}(0) = 2a_2 \\ \ddot{\theta}\left(t_f\right) = 2a_2 + 6a_3 t_f + 12a_4 t_f^2 + 20a_5 t_f^3 \end{cases} \tag{45}$$

In order to force the end of the manipulator to move under the desired trajectory, the pose trajectory in the Cartesian space should be transformed into the pose sequence in the joint space by the interpolation algorithm, for the further inversion solution. Common interpolation methods are linear interpolation, arc interpolation, polynomial interpolation, and spline curve interpolation. Among these interpolations, the spline curve interpolation is applied since it can fit complex curves more effectively.

Spline curve interpolation uses polygonal value points to represent the continuous curve, and the Bezier curve equation is

$$\begin{aligned} P(u) &= C_n^0 (1-u)^n P_0 + C_n^1 (1-u)^{n-1} u P_1 + \ldots + C_n^j (1-u)^{n-j} u^j P_j + \ldots C_n^n u^n P_n \\ &= \sum_{j=0}^{n} C_n^i (1-u)^{n-j} u_j P_j = \sum_{j=0}^{n} B_{j,n}(u) P_j \end{aligned} \tag{46}$$

where P_0, P_n are the initial point and terminal point, respectively. P_1, P_{n-1} are the control points, which can control the density distribution of curve interpolation. In other words, the selection of the control point can decide the shape of the Bezier curve. The common way is to use cubic B-spline interpolation, which is shown as:

$$\theta(u) = \left[u_3, u_2, u_1, 1\right] \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \end{bmatrix} \tag{47}$$

where $P_i\left(i = 0,1,2,3\right)$ is the 4 Bezier-type points, which are the key to fit the desired trajectory. Generally, based on several known points, the points are calculated to plan the desired curve, together with the curve error, until the curve expressions are obtained.

It should be noted that interpolation function is one relatively simple method for

motion planning of manipulator, which is suitable for general cases. However, when there are several obstacles in the environment, such motion planning may lead to collisions, resulting in work failure. Therefore, it is also necessary to consider the motion planning method with obstacle avoidance.

3.3.2.2 Sample-based planning algorithm

For the manipulator, the path planning with obstacle avoidance needs to plan one feasible trajectory under the satisfaction of initial and desired configurations. Furthermore, certain index function, such as shortest path and least energy, is used to plan the optimal path.

According to the perception of manipulator around the environment, path planning is divided into path planning with known environment and path planning with unknown environment. Here the offline planning scheme with known environment is proposed.

Traditional obstacle avoidance methods rely on the accurate modeling of obstacles in the configuration space. With the increase of DOF of the manipulator, and the complexity of the environment, the calculation of traditional methods increases significantly, leading to the terrible efficiency. In practical applications, the trajectory planned by traditional methods can only be applied into the manipulator with 5 DOF. While most redundant manipulator has at least 7 DOF. Therefore, traditional methods may not be suitable for the obstacle avoidance planning of redundant manipulator.

The sample-based obstacle avoidance path planning constructs a feasible map to search the feasible trajectory from the initial point to the desired point. First, the feasible information of sample points is obtained through the collision detection module. Then, a series of points sampled from free space are connected to finish the feasible map. Since sample-based method needs no accurate modeling of the obstacles in the configuration space, the calculation is greatly reduced, leading high efficiency. Among sample-based methods, fast extended random tree (RRT)[14] and its improved methods are widely used, which is also applied in our design.

3.3.3 Numerical simulation and analysis

The simulation results of the manipulator motion planning are shown as follows. It can be concluded that the motion planning meets the task requirements.

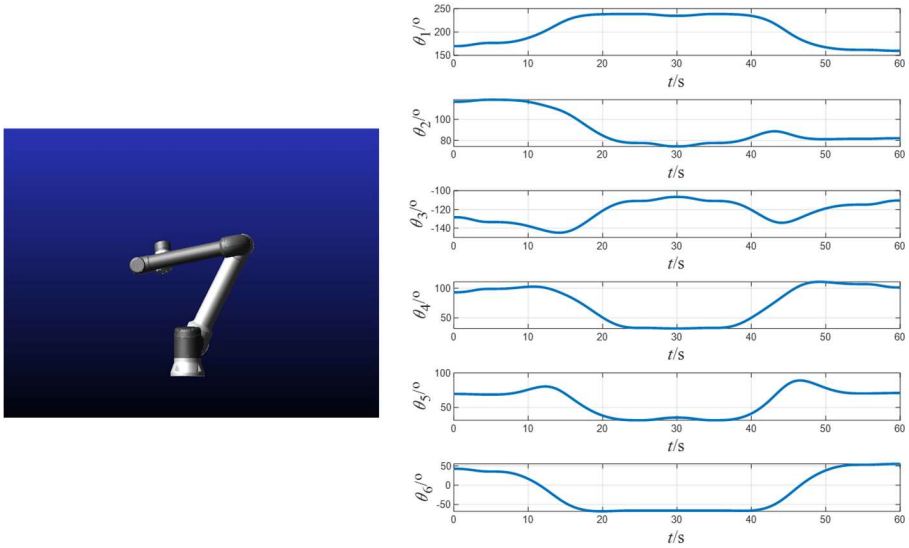


Fig. 36 Simulation results of the manipulator motion planning

In order to demonstrate the effectiveness of the sample-based algorithm, a 2-dimensional C space with narrow channel is designed. The sample-based planning algorithm is used on this space, which is shown in Fig. 37. It can be seen that the high sampling density at the narrow channel improves the connectivity and the shape description of the free space well, leading to the successful collision-free path planning.

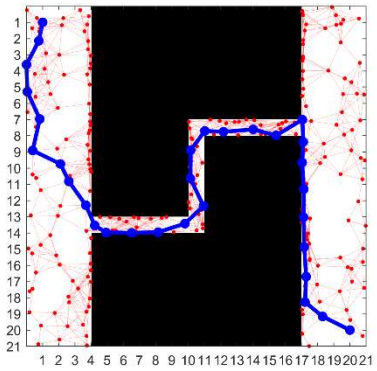


Fig. 37 PRM with improved sampling strategy

In order to demonstrate the effectiveness of the sample-based algorithm on the manipulator, manipulator UR10 is selected for simulation. The working environment is designed in Fig. 22. Considering the narrow channels, the path planning is relative difficult, and the simulation results verifies that sample-based algorithm is suitable for

the complex environment.

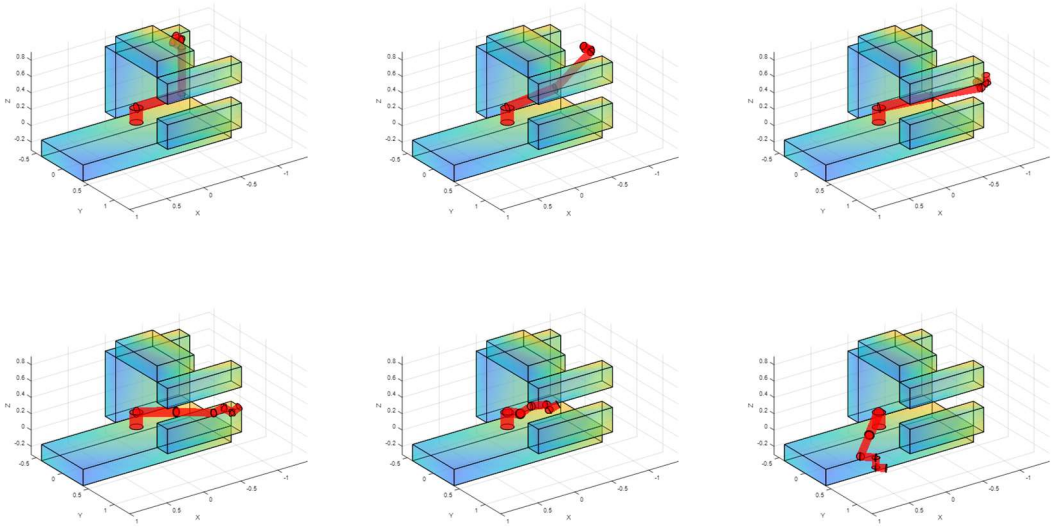


Fig. 38 Sample-based Obstacle Avoidance Path Planning of Manipulator

3.4 Emotion Recognition

Deep learning framework PyTorch is used to real-time recognize astronaut facial expressions captured by the camera. The facial expressions recognition model is trained on fer2013. Although the module is constructed based on Python, it is transferred by C++ so that the result of recognition can be utilized by other process conveniently.

3.4.1 Construction of model

In recent years, feed-forward neural network (FNN) and convolutional neural network (CNN) have also been used to extract expression features. A new recognition framework based on convolutional neural network (CNN) has achieved remarkable results in FER. Multiple convolutional and pooling layers in CNN can extract higher and multi-level features of the entire face or local regions, and have good classification performance of facial expression image features. Empirically, CNNs are better at image recognition than other types of neural networks.

Resnet18 is utilized to extract facial expression features and classify the facial expression. Resnet18 means that the network structure is Resnet and there are eighteen weight layers.

Residual network consists of series of residual blocks. The structure of residual

blocks is illustrated by Fig. 39. The input is convolved repeatedly and add to output.

The whole Resnet18 contain 17 convolutional layer and one full connection layer, which is illustrated by

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2 $\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times23$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Fig. 39 Structure of Resnet18

3.4.2 Training of model

The model is trained on fer2013 and cross entropy is utilized to design the loss function. Cross entropy is a concept in information entropy theory, which was originally used to estimate the average code length. In deep learning, it can be seen as the difficulty of representing the probability distribution $p(x)$ through the probability distribution $q(x)$, which is defined as follows

$$H(p,q)=\sum_{i=1}^{n}p(x_i)\log\frac{1}{q(x_i)}=-\sum_{i=1}^{n}p(x_i)\log q(x_i) \tag{48}$$

Cross-entropy describes the distance between two probability distributions, that is to say, the smaller the cross-entropy value (the smaller the relative entropy value), the closer the two probability distributions are.

After the fully connected layer, the model obtains the output probability of each category, but at this time the probability has not been normalized. A softmax layer is utilized to normalize the probability to 1, which is easier for data processing.

The FER2013 dataset consists of 28709 training images, 3589 public testing images and 3589 private testing images. Each image is a grayscale image with a pixel size of 48*48. There are 7 expressions in the FER2013 database: Anger, Disgust, Fear,

Happy, Sad, Surprised and Neutral. This database is the data of the Kaggle competition in 2013. Since this database is mostly downloaded from web crawlers, there are certain errors. Human accuracy on this database is $65 \pm 5\%$.

The effect of using the trained model to recognize faces is as follows:

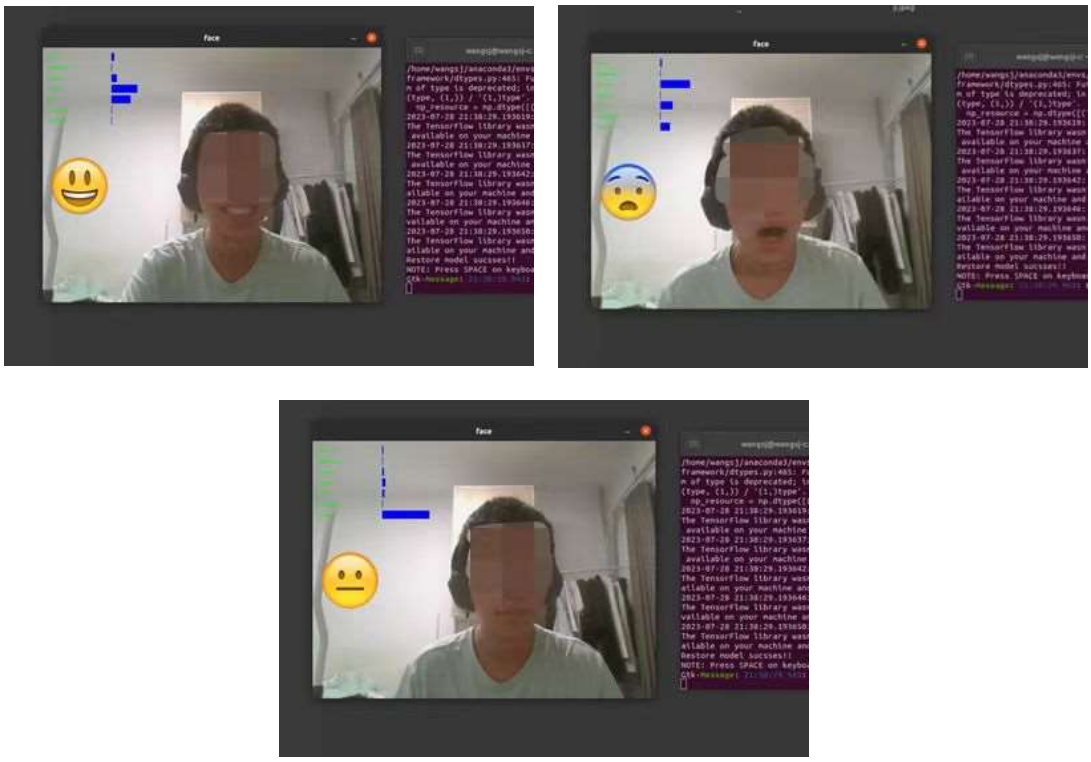


Fig. 40 Expression Recognition Test

3.4.3 C++ API

The deep learning frameworks utilized by facial expression recognition are generally implemented based on Python, and building, training, saving and calling models can be easily done under Python. But sometimes, we may need to use other programming languages when actually applying these models to system. This section will import TensorFlow pre-trained models by directly calling TensorFlow's C/C++ API.

A PyTorch model's journey from Python to C++ is enabled by Torch Script, a representation of a PyTorch model that can be understood, compiled and serialized by the Torch Script compiler. As a result, torch script environment needs to be downloaded and configured firstly. Then the trained model is converted to torch script in python. There exist two ways of converting a PyTorch model to Torch Script. The first is known

as tracing, and the second approach is to add explicit annotations to your model that inform the Torch Script compiler that it may directly parse and compile the model code, subject to the constraints imposed by the Torch Script language. In this sction, we converting the model to Torch Script via Annotation.

Secondly, the script module is serialized to a file. Then, the module can be loaded from the file in C++ and execute it without any dependency on Python.

Finally, scrip module is loaded in C++ application, which must be depend on the PyTorch C++ API known as LibTorch. LibTorch distribution encompasses a collection of shared libraries, header files and CMake build configuration files. While CMake is not a requirement for depending on LibTorch, it is the recommended approach and will be well supported into the future. A Mimimal C++ Application is shown as follows.

3.5 Visual Mapping Using SLAM

For navigation of Bifang in the space station, SLAM is utilized to real-time estimate the pose in changing space station environment. Depth camera is chosen as visual sensor for SLAM and the module runs on ROS.

3.5.1 ORB_SLAM3

ORB-SLAM3 is a SLAM system that supports vision, vision plus inertial navigation, and hybrid maps. It can run on monocular, binocular, and RGB-D cameras using pinhole or fisheye models.

ORB-SLAM3 is the first tightly coupled feature-based VIO system that only relies on maximum a posteriori estimation (including the IMU at initialization). The effect of such a system is: whether it is in a large scene or a small scene, indoors or outdoors, it can run robustly in real time, and its accuracy has increased by 2 to 5 times compared with the previous version. The second innovation of ORB_SLAM3 is the hybrid map built based on the new relocation module that improves recall, because this module allows ORB-SLAM3 to run for a long time in scenes with poor features: when the odometer fails, The system will rebuild the map and align this map with the originally constructed map. Compared to odometry, which only utilizes the latest few frames of

data, ORB-SLAM3 is the first system capable of reusing all previous information at all algorithm stages. Such a mechanism can use key frames with a common view relationship during BA, even if the two frames are far apart in time, or come from the original mapping process. This system achieved an average accuracy of 3.6cm on the EuRoC dataset, and 9mm accuracy on the TUM-VI dataset (AR/VR scene), which uses fast-moving handheld devices.

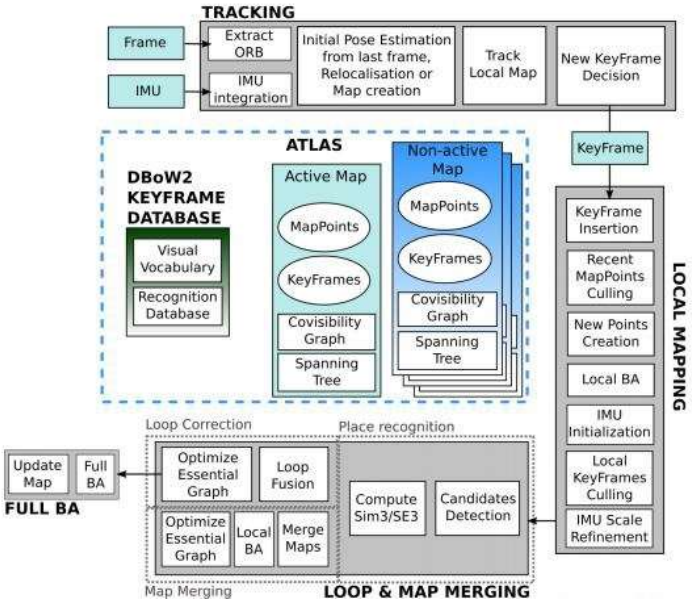


Fig. 41 ORB-SLAM3[15]

ORB-SLAM3 is a system based on ORB-SLAM2 and ORB-SLAM-VI, which can run robustly in pure vision or visual inertial navigation systems (monocular, binocular and RGB-D use pinhole or fisheye model).

3.5.2 Running SLAM on ROS with Depth Camera

Firstly, add the ROS path of SLAM to the environment variable and compile the ROS version of ORB_SLAM3. Then, open ROS Master, modify the topic related to /cam0/image_raw and /cam1/image_raw in the source code and start the stereo node. The results of running the example in the ROS environment is shown as follows.

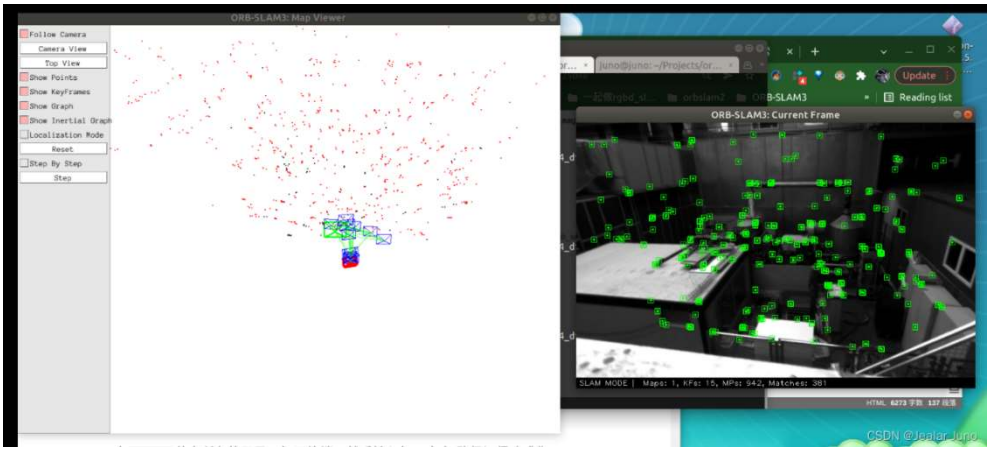


Fig. 42 Example in the ROS

For RealSense D435i, install RealSense sdk and ROS wrapper and tested.

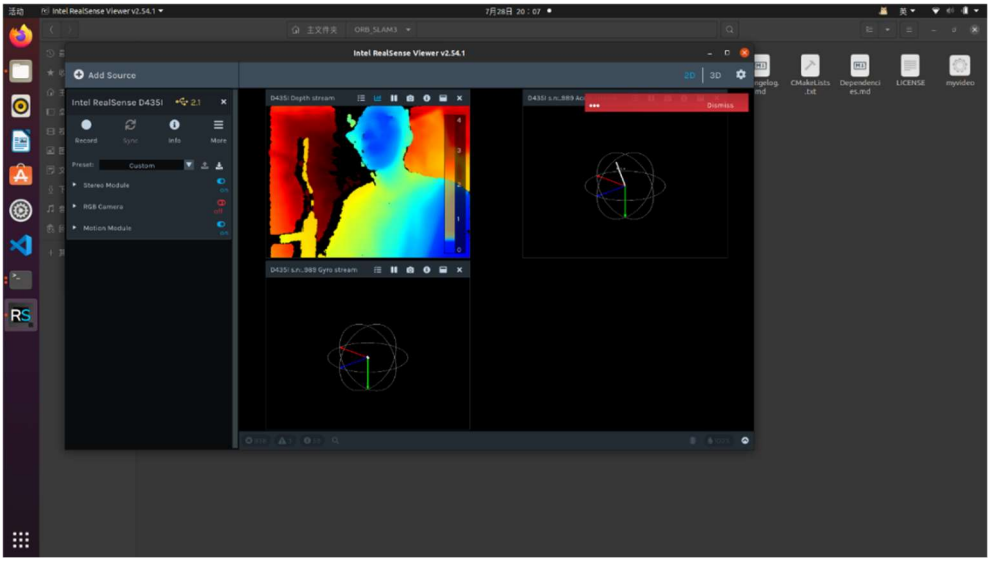


Fig. 43 Sdk and ROS Wrapper Test

Finally, start the ROS node of RealSense. The ROS interface of realsense can be understood as a node that reads data and publishes topics. When writing ROS programs, we can subscribe to the topics sent by this node. Calculate the camera parameters according to the official manual and modify the parameters in the yaml file according to the parameters of your own camera, basically only need to modify the internal parameters of the camera and the width and height of the image. Change the topic name in the source code to the camera topic and run.

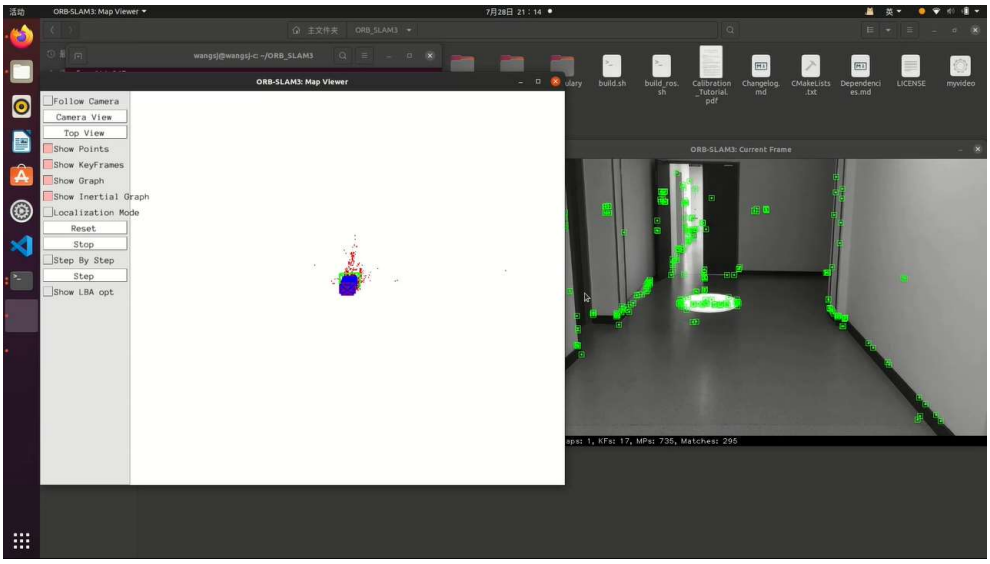


Fig. 44 Real-time Mapping in ROS

3.6 Device Selection and Parameter Checking

3.6.1 Device selection

According to the previous analysis, we have a certain understanding of the performance index of the required components. Here are some of the components we initially selected for each subsystem.

3.6.1.1 Dynamic system

The motor on the quadrotor UAV has the characteristics of small size and large power, therefore, the SUNNYSKY brand X2204 series motor is selected, and the primary KV value is 1480. The matching propeller blade is the standard model 5045 with a diameter of 127mm. For the joint part of the Double-Gimbal Fan, GDW RS0708 servo is selected, and its torque is about $7.5\text{kg}\cdot\text{cm}$. Skywalker-15A is selected for electrical adjustment.

(a) SUNNYSKY X2204 Brushless Motor

(b) Blade of Propeller 5045

(c) Steering Gear GDW RS0708 **(d) Electric Modulation Skywalker-15A**

Fig. 45 Device of Dynamic System

3.6.1.2 Computational system & Communication

Drawing on the design of some mature high performance computing drones, this part of the 13th generation Intel NUC Arena Canyon. Equipped with i7 processor, it supports up to 64GB dual-channel 3200MHz memory. Dual PCIe x4 Gen 4 NVMe SSD is supported with one M.2 slot reserved. Wi-Fi 6E (Intel solution) and Bluetooth 5.3 are supported. Equipped with Intel i226 wired network card, 2.5 Gbps rate. Equipped with dual HDMI and dual Thunderbolt 4, support four 4K@60Hz external connection. Three USB 3.2 and one USB 2.0.

Fig. 46 Intel NUC Arena Canyon

3.6.1.3 Visual system

Intel RealSense Depth Camera D435i can be regarded as an upgraded version of D435, the difference from D435 is that there is an extra IMU. The D435i has almost the same specifications as the previous generation. It is characterized by not only providing a depth sensor module, but also equipped with an IMU unit (inertial measurement unit, Bosch BMI055 used). With the built-in IMU unit, combined with visual data, 6DoF

tracking function can be realized. Among them, the IMU combines the data of various linear accelerometers and gyroscopes to detect the rotation and translation of the X, Y, and Z axes, as well as actions such as pitch and roll. The D435i's 20-megapixel RGB camera and 3D sensor can deliver resolutions up to 1280 x 720 at 30 fps, or a lower resolution of 848 x 480 at 90 fps. The camera has a global shutter to handle fast moving objects and can be operated both indoors and outdoors. The depth distance is between 0.1 m and 10 m, and the field of view angle is 85 × 58 degrees.

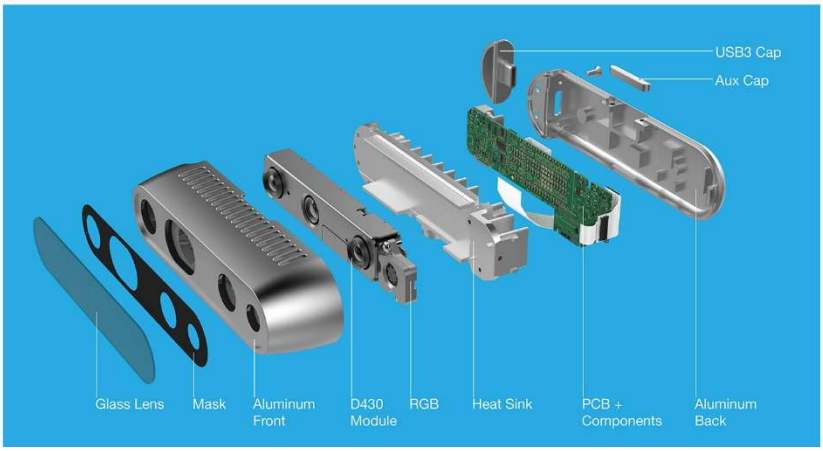


Fig. 47 Intel RealSense Depth Camera D435i

3.6.1.4 Touching screen

Tablet computers on the market have good operating experience and fluency, so Bifang designs a Huawei tablet as its interactive screen, and connects other functional hardware to the tablet through API.

Fig. 48 HUAWEI MatePad SE

3.6.1.5 Power

Rough estimation of the above requirements, the initial selection of the model

plane battery battery on the market, its capacity of 5300mAh with 4S.

Fig. 49 Battery

3.6.2 Parameter checking

The quality, power and cost of each component required by Bifang can be initially accounted for. As shown in the table:

Table 1 Parameter checking of Bifang

Name	Mass(g)	Power(W)	Cost(￥)	Number
Brushless Motor	180	5	95	2
Blade	30	0	5	2
Steering Gear	100	5	180	4
Electric Modulation	20	10	100	2
Intel NUC	800	50	3000	1
Depth Camera	500	100	3000	4
HUAWEI MatePad SE	300	100	2000	1
Robotic Arm	1000	50	7000	1
Battery	660	5300mAh, 6S	500	4
Structure	About 1000	0	2000	1
Total	About 7000	Peak Power Use Time: 1h Cruise Use Time:1.5h	About 30000	

3.7 Conclusion

This section first introduces the overall design of the Bifang robot, then demonstrates the feasibility of the Bifang design from the perspective of technical implementation, and finally checks the relevant parameters.

4 Status and Prospect

4.1 Current Research Progress

In addition to the attempts on face recognition and SLAM mapping in the previous chapter, we also made a prototype model of the variable-speed double-gimbal fan and carried out experiments.

In order to verify the feasibility of the propulsion scheme in this design and obtain basic dynamic characteristic data, we made a prototype of the power device in the robot at the initial stage of design and measured the thrust curve.

The design of the prototype is basically the same as that of the complete design. In order to facilitate the production, some simplifications have been made in the structure, omitting part of the shell and supporting structure, and retaining the key configuration of the three-degree-of-freedom rotation. The power unit is mainly composed of wing, cantilever, rotor frame, servo, brushless motor, brushless ESC, and rotor. Among them, the wing, cantilever, and rotor frame are made by 3D printing, and the structure is relatively light. The servo uses the RS0708 produced by GDW, and the controllable range is 0-350°. The brushless motor uses Langyu X2204-KV1800 motor. The brushless ESC uses Hobbywing Skywalker 12A ESC. The rotor model is EP5045.

Fig. 50 Model of Prototype

Using two servos, the omnidirectional rotation of the fan can be realized. Through reasonable wiring, the wires are fixed on the cantilever, leaving a torsion margin so that it will not interfere with the omnidirectional rotation of the fan. Use a brushless ESC to control the motor, and by adjusting the duty cycle of the input PWM signal, you can

control the thrust of the fan.

In this experiment, an Arduino is used to control the servos and brushless motors. The development board used is Arduino UNO R3, with compiler version 2.0.3. Read the voltage of the potentiometer knob through the analog interface to adjust the steering angle. By sending a signal in the serial monitor, the PWM duty cycle is changed to control the speed of the brushless motor. PWM frequency is set to 500Hz, and effective duty cycle is 50%-100%. Fix the power unit on the electronic scale, and the direction of the thrust is upward. By calculating the difference between the readings of the electronic scale before and after the thrust is applied, the magnitude of the thrust can be calculated.

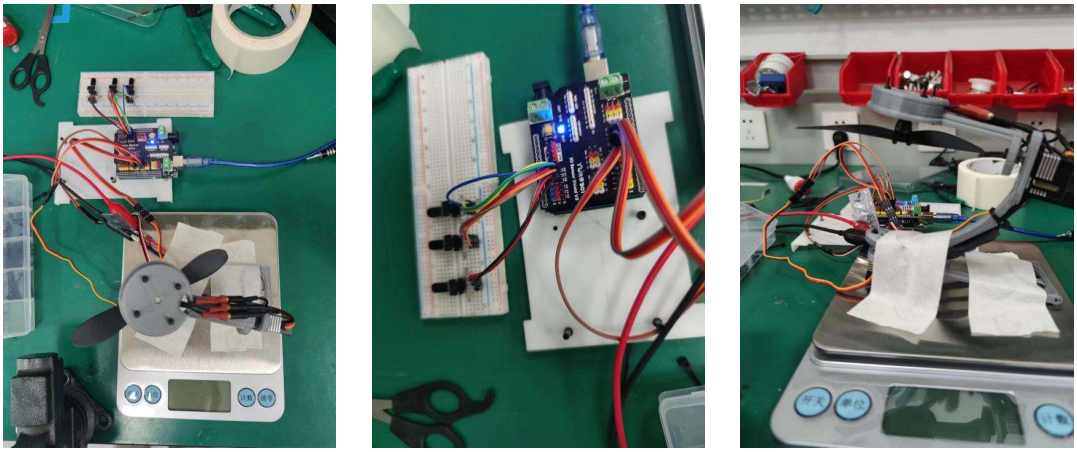
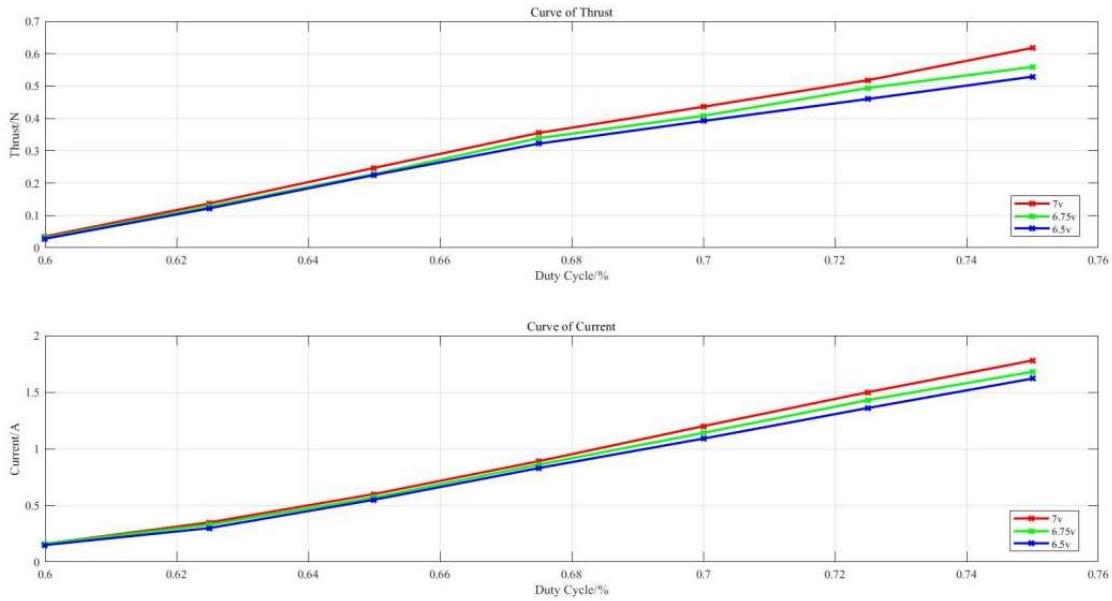


Fig. 51 Experiment Environment

In the experiment, we use a constant voltage power supply, fix the input voltage, adjust the PWM duty cycle, measure the thrust and the current of the power supply, and the results are as follows:



改

When the power supply voltage is 7V, the maximum thrust measured by a single fan is about 0.7N. The complete robot uses two fans, and the maximum total thrust can reach 1.4N, which is more than twice that of Astrobee, which means that our design has better athletic ability.

4.2 Future Work and Prospects

The thrust test of the fan in the whole state will be completed in the future to explore the interference of the fuselage to the fan.

In addition, we will try to deploy the vision algorithm to a removable device to further explore the feasibility of our design scheme.

We truly hope that Bifang will come true soon and bring some convenience to the astronauts working in the space station.

5 Conclusion

This paper describes a novel free-flying robot working in the space station, Bifang, which has novel dynamics design and powerful human-robot interaction. The feasibility of Bifang is verified from the technical principle and the related prototype test is carried out to ensure the reliability of the design.

In conclusion, Bifang represents a promising advancement in the field of free-flying robots for space station operations. Its versatile capabilities, ranging from object manipulation to emotion recognition, hold the potential to significantly assist astronauts and improve their overall experience during missions. The thorough research and verification conducted during the design process add credibility to the robot's functionality and feasibility in a real space station setting.

References

[1] A. S. Otero, A. Chen, D. W. Miller and M. Hilstad, "SPHERES: Development of an ISS Laboratory for formation flight and docking research," Proceedings, IEEE Aerospace Conference, Big Sky, MT, USA, 2002, pp. 1-1, doi: 10.1109/AERO.2002.1036828.

[2] A. Saenz-Otero, J. Katz and A. T. Mwijuka, "The Zero Robotics SPHERES challenge 2010," 2011 Aerospace Conference, Big Sky, MT, USA, 2011, pp. 1-13, doi: 10.1109/AERO.2011.5747666.

[3] S. Mitani et al., "Int-Ball: Crew-Supportive Autonomous Mobile Camera Robot on ISS/JEM," 2019 IEEE Aerospace Conference, Big Sky, MT, USA, 2019, pp. 1-15, doi: 10.1109/AERO.2019.8741689.

[4] Bualat M, Barlow J, Fong T, et al. Astrobee: Developing a free-flying robot for the international space station[C]//AIAA SPACE 2015 conference and exposition. 2015: 4643.

[5] Smith T, Barlow J, Bualat M, et al. Astrobee: A new platform for free-flying robotics on the international space station[C]//International Symposium on Artificial Intelligence, Robotics, and Automation in Space (i-SAIRAS). 2016 (ARC-E-DAA-TN31584).

[6] Park I W, Smith T, Sanchez H, et al. Developing a 3-DOF compliant perching arm for a free-flying robot on the International Space Station[C]//2017 IEEE International Conference on Advanced Intelligent Mechatronics (AIM). IEEE, 2017: 1135-1141.

[7] Bualat M G, Smith T, Smith E E, et al. Astrobee: A new tool for ISS operations[C]//2018 SpaceOps Conference. 2018: 2517.

[8] Z. Hussain and N. Z. Azlan, "KANE's method for dynamic modeling," 2016 IEEE International Conference on Automatic Control and Intelligent Systems (I2CACIS), Selangor, Malaysia, 2016, pp. 174-179, doi: 10.1109/I2CACIS.2016.7885310.

[9] Q. Guo and D. Jiang, "Moving Process PID Control in Robots' Field," 2012

International Conference on Control Engineering and Communication Technology, Shenyang, China, 2012, pp. 386-389, doi: 10.1109/ICCECT.2012.108.

[10] Daan Stevenson and Hanspeter Schaub, Nonlinear Control Analysis of a Double-Gimbal Variable-Speed Control Moment Gyroscope. Journal of Guidance, Control, and Dynamics 2012 35:3, 787-793

[11] Singh, A., Singla, A. Kinematic Modeling of Robotic Manipulators. Proc. Natl. Acad. Sci., India, Sect. A Phys. Sci. 87, 303–319 (2017). https://doi.org/10.1007/s40010-016-0285-x

[12] Dressel, L. and Kochenderfer, M.J. (2019), Tutorial on the generation of ergodic trajectories with projection-based gradient descent. IET Cyber-Physical Systems: Theory & Applications, 4: 89-100. https://doi.org/10.1049/iet-cps.2018.5032

[13] Kathrin Glau & Mirco Mahlstedt (2019) Improved error bound for multivariate Chebyshev polynomial interpolation, International Journal of Computer Mathematics, 96:11, 2302-2314, DOI: 10.1080/00207160.2019.1599364

[14] M. Elbanhawi and M. Simic, "Sampling-Based Robot Motion Planning: A Review," in IEEE Access, vol. 2, pp. 56-77, 2014, doi: 10.1109/ACCESS.2014.2302442

[15] Campos C, Elvira R, Rodríguez J J G, et al. Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam[J]. IEEE Transactions on Robotics, 2021, 37(6): 1874-1890.