CAMBRIDGE
UNIVERSITY PRESS

RESEARCH ARTICLE

Virtual interaction and manipulation control of a hexacopter through hand gesture recognition from a data glove

Haiming Huang[1], Di'en Wu[1], Zehao Liang[1], Fuchun Sun[1,2] and Mingjie Dong[3,*]

[1]College of Electronics and Information Engineering, Shenzhen University, Shenzhen, China, [2]Department of Computer Science and Technology, Tsinghua University, Beijing, China, and [3]Faculty of Materials and Manufacturing, Beijing University of Technology, Beijing, China
*Corresponding author. E-mail: dongmj@bjut.edu.cn

Abstract

The purpose of this study is to realize virtual interaction and manipulation control of a hexacopter based on hand gesture recognition from a designed data glove, to provide an intuitive and visual real-time simulation system for flight control algorithm verification and external control equipment testing. First, the hand gesture recognition from a designed data glove is studied, which can recognize different actions, such as mobile ready, grab, loosen, landing, take-off, and hover. Then, the design of virtual simulation system for hexacopter capture is completed, with the model design of hexacopter and manipulator, and the simulation software design with *CoppeliaSim*. Finally, virtual simulation experiment of hexacopter grasping and virtual flight control experiment based on data glove are tested, respectively, and quantitatively described. The overall recognition rate is 84.3%, indicating that the data glove produced has the ability to recognize gestures, but its recognition performance is not superior. In gesture recognition, the recognition rate of static gestures is relatively higher than that of dynamic gestures. Among the static gestures, the hover gesture has the highest recognition rate. The average correct rate of static gestures can reach 94%. The lowest recognition rate of dynamic gestures is upward movement, and the average recognition rate of dynamic gestures is 76.1%. The research can be used to remotely operate hexacopter using a data glove in the future and improve the control performance through virtual interaction and manipulation simulation before actual application.

1. Introduction

Unmanned aerial vehicles (UAVs) play an important role in the fields of aerospace, agriculture, and delivery due to their high maneuverability, flexible deployment, multi-aircraft cooperation, and easy to carry object for long-distance flight and dexterous manipulation [1]–[4]. In these fields, the flight operation of UAV equipped with manipulator has attracted more and more attention [5].

Kim et al. [6] designed a foldable arm that was carried to a UAV to perform a variety of tasks in a confined space. Lee and Jung [7] designed a shooting and rapidly retracting manipulator that could be carried on the UAV, which can bring a 30 g mass located at 0.8 m away within 600 ms. However, most of the UAVs are operated by remote controller, which results in weak human–machine interaction because the operator cannot feel the operation status of the UAV, thus making the operation of the UAV less effective. Friendly interactive control can improve the operation performance of UAV [8], so many researchers have been exploring UAV control methods, such as visual gesture control [9, 10], data glove control [11, 12], and brain wave control [13]. Among these control methods, data glove can provide better feelings for operators, because it makes the control of UAV more intuitive and can accurately recognize dynamic gestures in complex environment with less computation [14, 15]. Also, data glove-based methods cannot be affected by the occlusion problem, which is a critical issue in optic

tracking-based systems. Furthermore, data gloves are widely used in many fields, such as robot arm control [12], sign language recognition [16], and UAV control [17]. For example, ref. [18] presents an intelligent human–UAV interaction approach in real time based on machine learning using wearable gloves; it used data gloves in the Multi-Mode UAV Human-Computer Interaction System.

Algorithm is the point of gesture recognition with data gloves. At present, commonly used algorithms include sensor fusion algorithm [19], closed-form reconstruction algorithm [11], supervised learning algorithm [15], etc. Muezzinoglu and Karakose [18] compared decision tree, Nave Bayes, support vector machines, and k-nearest neighbor classification methods in the interaction between data gloves and UAV. Finally, experiments are needed to verify the control effect of the algorithm results of the data glove on the UAV. Compared with the simulation environment, using data glove to control UAV in the real environment is costly and dangerous, and the UAV experimental verification in the simulation environment has also been widely used. For example, Fernando et al. [20] used Matlab Simulink to test the flight algorithm of quadrotor UAV; Meyer et al. [21] designed a comprehensive simulation system of quadrotor UAV using ROS and Gazebo; Udvardy et al. [22] carried out obstacle avoidance simulation of UAV through CoppeliaSim; and Li et al. studied the glove-based virtual hand grasping for virtual mechanical assembly, with detailed definitions of the bending angles of grasping gestures for four types of parts [23].

In this paper, we aim to realize virtual interaction and manipulation control of a hexacopter based on hand gesture recognition from a designed data glove, to provide an intuitive and visual real-time simulation system for flight control algorithm verification and external control equipment testing. The remainder of the paper is organized as follows. Section 2 describes the hand gesture recognition from a data glove. The design of virtual simulation system for hexacopter capture is presented in Section 3. Section 4 demonstrates the simulation test verification, while conclusions and future works are given in Section 5.

2. Hand gesture recognition from a data glove

2.1. Data glove with its hardware

The designed data glove is shown in Fig. 1, which consists of a STM32F103 microcontroller and 10 MPU6050 sensors. The MPU6050 sensors are connected to the microcontroller through the IIC bus for real-time detection of acceleration and angular velocity, while the microcontroller is connected to the upper computer through RS232. The original angular velocity and acceleration can be obtained by using the gyroscope and accelerometer of the MPU6050. On this basis, the current Euler angle of the object can be obtained by filtering and integrating the data. Here, we can directly use the digital motion processor to obtain the quaternion from the MPU6050 for simplification. By the transformation of Euler angle and quaternion, the Euler angle of the detected object can be obtained. The hand gesture recognition from the data glove mainly includes the recognition of different actions based on the figure bending and palm flip judgment.
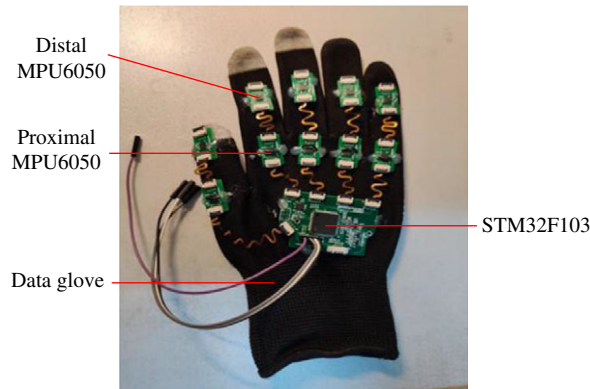
2.2. Finger bending and palm flip judgment

When recognizing the finger bending, the back of the hand is upwards initially, and the pitch angles of all sensors are zero. The schematic diagram of a bending finger is shown in Fig. 2, with the distal joint and the proximal joint corresponding to the distal MPU6050 and the proximal MPU6050 in Fig. 1, respectively.

The pitch angles of the two fingers can be measured as θ_1 and θ_2 by the MPU6050 sensors, respectively. Then, the θ_3 can be obtained by using (1).

$$\theta\_3 = 180^\circ - (\theta\_2 - \theta\_1) \tag{1}$$

To simply the recognition process, the figure is thought to be curved when the angle between two fingers is between 60° and 120°, while the figure is considered to be in a straight state when the angle

Figure 1. *Data glove with its hardware.*
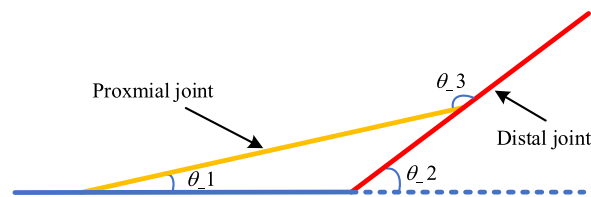
Figure 2. *Finger bending judgment diagram.*

between two fingers is between 0° and 20°. When recognizing the palm flip, the back of the hand is upwards initially, and the roll angles of all sensors are zero. The mean value of the roll angle of the remaining sensors except the thumb is taken as the flip angle of the hand gesture.

2.3. Recognition of different actions

Based on the recognition of figure bending and palm flip, and the relationship of the hand joints, we define a set of concise instruction gestures to quantify the different hand gestures in the program in order to realize the virtual simulation control of a hexacopter. The developed hand gestures are shown in Fig. 3, and the palm roll angle interval is $[-15° \sim 15°]$ with finger bend command "11111" for the mobile ready gesture (The movement of forward, backward, left, right, up, and down can be realized by modifying the parameters in the control command); the back of hand roll angle interval is $[-15° \sim 15°]$ with finger bend command "11111" for the grab gesture; the back of hand roll angle interval is $[-15° \sim 15°]$ with finger bend command "11100" for the loosen gesture; the palm roll angle interval is $[-195° \sim -165°]$ with finger bend command "11111" for the landing gesture; the palm pitch angle range interval is $[60° \sim 100°]$ with finger bend command "00111" for the takeoff gesture; and the palm roll angle interval is $[-195° \sim -165°]$ with finger bend command "11100" for the hover gesture.

When the gesture of the data glove is recognized, it is mapped to the corresponding posture of the hexacopter, and the command is sent to the simulation software through the serial port on the data glove. After the hexacopter receives the gesture instruction, it analyzes and executes the corresponding action.

3. Model design and virtual simulation development for hexacopter capture

3.1. Model design of hexacopter and manipulator

The 3D model of the hexacopter is shown in Fig. 4(a), with a size of 600 mm; the manipulator is shown in Fig. 4(b), consisting of a base, four rotary joints, and the connecting rods, and its end-effector is a clamping device.
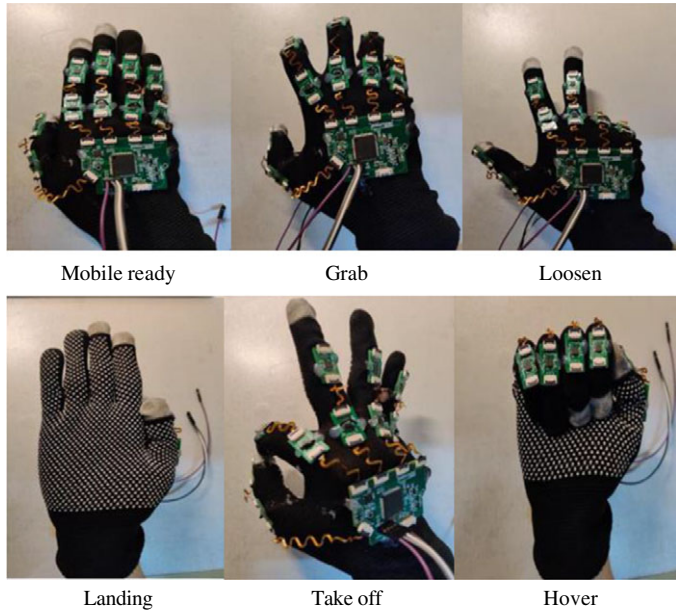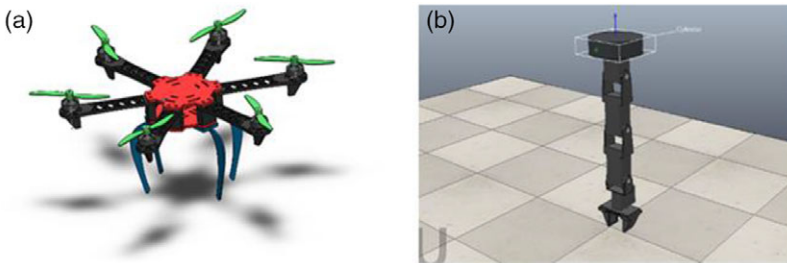
Figure 3. *Recognition of different actions.*

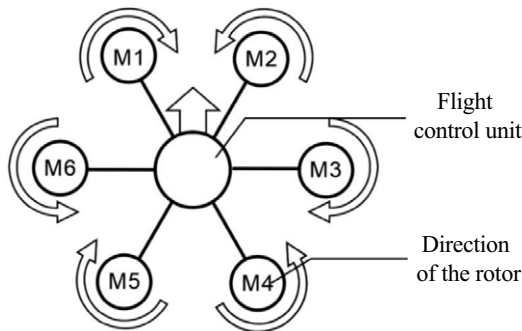**Figure 4.** *(a) 3D models of hexacopter; (b) 3D models of manipulator.*

Figure 5. *Simplified model of a hexacopter.*

3.1.1. Model design of the hexacopter

The simplified model of a hexacopter is shown in Fig. 5. It includes a central flight control unit, three pairs of rigid arms, rotors, and motors. Each pair of rotors are evenly distributed at a distance of 60°, and at the same height, with the same distance from the center. The rotors 1, 3, and 5 rotate clockwise,
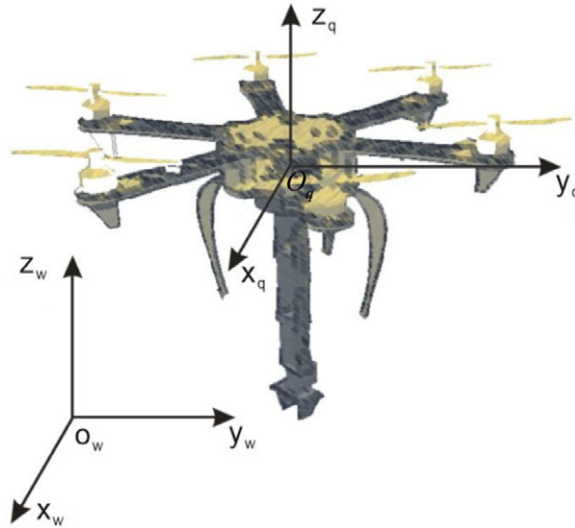
Figure 6. *Schematic diagram of the coordinate system.*

while rotors 2, 4, and 6 rotate counterclockwise. The reason for this setting is to counteract the torque effect and gyro effect and maintain the conservation of torque to ensure flight stability.

To establish the kinematics model of the hexacopter, we first need to assume the followings:

- The whole body of the hexacopter is rigid body and will not undergo elastic deformation.
- The surrounding environment is close to the ideal state, ignoring various external disturbances.
- The established body coordinate system coincides with the center of mass of the hexacopter.

For the hexacopter, we establish two coordinate systems, the body coordinate system $O_q - X_q Y_q Z_q$ and the world coordinate system $O_w - X_w Y_w Z_w$, as shown in Fig. 6.

For the body coordinate system $O_q - X_q Y_q Z_q$, each rotor of the hexacopter can generate lift force F_i and moment M_i, and their relationships with the rotation speed of the motors are given in (2), where k_b indicates the lift coefficient, k_d represents the torque coefficient, and i indicates the *ith* motor.

$$\begin{cases} F_i = k_b r_i^2, k_b \geq 0 \\ M_i = k_d r_i^2, k_d \geq 0 \end{cases} \tag{2}$$

Here, we define the Euler angle of the hexacopter as $\eta = \begin{bmatrix} \phi & \theta & \psi \end{bmatrix}$, where ϕ is the roll angle, θ is the pitch angle, and ψ is the yaw angle. The angular velocity is $\Omega = \begin{bmatrix} p & q & r \end{bmatrix}$, so the transformation matrix R from the body coordinate system $O_q - X_q Y_q Z_q$ to the world coordinate system $O_w - X_w Y_w Z_w$ can be established, as given in (3).

$$R = \begin{bmatrix} \cos\theta \cos\psi & \sin\theta \cos\psi \sin\phi - \sin\psi \cos\phi & \sin\theta \cos\psi \cos\phi + \sin\psi \sin\phi \\ \cos\theta \cos\psi & \sin\theta \sin\psi \sin\phi + \cos\psi \cos\phi & \sin\theta \sin\psi \cos\phi - \cos\psi \sin\phi \\ -\sin\theta & \cos\theta \sin\phi & \cos\theta \cos\phi \end{bmatrix} \tag{3}$$

Ignoring the air resistance, the total lift of the hexacopter F_q in the body coordinate system is given in (4).

$$F_q = \begin{bmatrix} F_x & F_y & F_z \end{bmatrix}^T = \begin{bmatrix} 0 & 0 & \sum_{i=1}^{6} F_i \end{bmatrix}^T \tag{4}$$

Then, the total lift of the hexacopter F_w converted from the body coordinate system $O_q - X_q Y_q Z_q$ to the world coordinate system $O_w - X_w Y_w Z_w$ is given in (5).

$$F_w = RF_q = \begin{bmatrix} \sin\theta\cos\psi\cos\phi + \sin\psi\sin\phi \\ \sin\theta\sin\psi\cos\phi - \cos\psi\sin\phi \\ \cos\theta\cos\phi \end{bmatrix} \sum_{i=1}^{6} k_b r_i^2 \tag{5}$$

The force of the hexacopter F_g in the world coordinate system $O_w - X_w Y_w Z_w$ is given in (6).

$$F_g = F_w - mgE_z \tag{6}$$

where m represents the mass of the hexacopter, g represents the acceleration of gravity, and E_z represents the z-axis unit vector of the world coordinate system. According to Newton's second law, the linear displacement equation can be obtained by combining (3), (5), and (6), as given in (7).

$$\begin{cases} \ddot{x} = \dfrac{\sum\limits_{i=1}^{6} F_i \left(\sin\theta\cos\psi\cos\phi + \sin\psi\sin\phi\right)}{m} \\[3ex] \ddot{y} = \dfrac{\sum\limits_{i=1}^{6} F_i \left(\sin\theta\sin\psi\cos\phi - \cos\psi\sin\phi\right)}{m} \\[3ex] \ddot{z} = \dfrac{\sum\limits_{i=1}^{6} F_i \left(\cos\theta\cos\phi - g\right)}{m} \end{cases} \tag{7}$$

Define the vector M_q in the body coordinate system $O_q - X_q Y_q Z_q$, as given in (8).

$$M_q = J_b \dot{\Omega} + \Omega \times J_b \Omega \tag{8}$$

where J_b is the moment of inertia matrix of the hexacopter, and the angular velocities corresponding to the three anglers in the body coordinate system $O_q - X_q Y_q Z_q$ are p, q, and r. We can get (9), where the *sec* is the abbreviation of secant.

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin\theta\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi\sec\theta & \cos\phi\sec\theta \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{9}$$

The three components of the vector M_q represent the torque of the three coordinate axes in the body coordinate system $O_q - X_q Y_q Z_q$, respectively, as given in (10).

$$M_q = \begin{bmatrix} M_x & M_y & M_z \end{bmatrix}^T = A\begin{bmatrix} r_1^2 & r_2^2 & r_3^2 & r_4^2 & r_5^2 & r_6^2 \end{bmatrix}^T \tag{10}$$

And, the A is given in (11).

$$A = \begin{bmatrix} k_b & \frac{1}{2}k_b l & -\frac{1}{2}k_b l & -k_b l & -\frac{1}{2}k_b l & \frac{1}{2}k_b l \\ 0 & -\frac{\sqrt{3}}{2}k_b l & -\frac{\sqrt{3}}{2}k_b l & 0 & \frac{\sqrt{3}}{2}k_b l & \frac{\sqrt{3}}{2}k_b l \\ k_d & -k_d & k_d & -k_d & k_d & -k_d \end{bmatrix} \tag{11}$$

3.1.2. Dynamics modeling of the manipulator

The manipulator mounted under the hexacopter will cause the center of gravity of the entire system to change during the movement, and the mass of the object grabbed by the manipulator will also affect the attitude of the hexacopter. In general, the fetching operation occurs when the hexacopter is hovering, so the motion analysis of the manipulator can be simplified to the motion analysis of its center of mass.

Define the position of the center of mass of the hexacopter in the world coordinate system $O_w - X_w Y_w Z_w$ as $^w P_u$, the position of the center of mass of the manipulator in the body coordinate system

$O_q - X_q Y_q Z_q$ as ${}^q P_c = \begin{bmatrix} x_{\text{com}} & y_{\text{com}} & z_{\text{com}} \end{bmatrix}^T$ which can be converted to the world coordinate system $O_w - X_w Y_w Z_w$, as given in (12).

$$ {}^w P_c = {}^w P_u + R^q P_c \tag{12} $$

In the body coordinate system $O_q - X_q Y_q Z_q$, the absolute velocity of the center of mass of the manipulator is given in (13), where the q_v is the three-axis speed in the body coordinate system.

$$ {}^q v_c = q_v + {}^q P_c + \Omega \times {}^q P_c \tag{13} $$

Convert the gravity of the manipulator to the body coordinate system $O_q - X_q Y_q Z_q$, as given in (14).

$$ m_c g \begin{bmatrix} -\sin\theta \\ \sin\phi\cos\theta \\ \cos\phi\cos\theta \end{bmatrix} - m_c \left({}^q v_c + \Omega \times {}^q v_c \right) = F_c \tag{14} $$

where F_c represents the force of the manipulator on the joint of the hexacopter in the body coordinate system $O_q - X_q Y_q Z_q$. From the moment balance, we can get (15), where ${}^q P_L$ is the position vector of the joint in the body coordinate system $O_q - X_q Y_q Z_q$.

$$ \sum_{i=1}^{6} {}^q P_L \times \left[F_i E_z \right] + \left({}^q P_c - {}^q P_L \right) \times F_c = 0 \tag{15} $$

3.2. Virtual simulation development

The simulation software we use is *CoppeliaSim*, and the physics engine used in this simulation experiment is the *Bullet* 2.78 physics library, which can simulate structures including rigid bodies, soft bodies, and elastic bodies, in terms of dynamics, and also has a 3D boundary collision function.

For the flight control simulation in the CoppeliaSim software, the first step is to obtain the handle of each object through *sim.getObjectHandle*(), which can be a model, sensor or joint, etc., so that various parameters can be passed to it to achieve the corresponding model object control. Then, the kinematics information of the hexacopter can be obtained through the Application Programming Interface function library encapsulated in the software. For example, the position information of the hexacopter and the target in the inertial coordinate system can be obtained by *sim.getObjectPosition*(). The velocity information of the hexacopter and the target can be obtained by *sim.getObjectVelocity*(). The attitude matrix of the hexacopter in the inertial coordinate system can be obtained through *sim.getObjectOrientation*(). The transformation matrix from body coordinate system to inertial coordinate system can be obtained through *sim.getObjectMatrix*(). This information is used as the input to perform vertical control, horizontal control, and rotation control, respectively, so as to solve the lift and torque that each motor of the hexacopter should have. Finally, it is assigned to the six rotor motors through *sim.addForceAndTorque*(), which realizes the flight control of the hexacopter in the simulation software.

For the control simulation of the manipulator, the first is to obtain the handle of each rotation joint through *sim.getObjectHandle*() and then to set the initial position of each joint and the range of the rotation value. After that, using *sim.rmlMoveToJointPosition*() to assign all the set values to each joint through, so that each joint can be driven to move at the same time. The grabbing of the clamping device can be controlled by the *sim.setintegerSignal*(), and the grabbing device is closed when the acquired signal is 1, while the grabbing device is released when the acquired signal is 0. As shown in Fig. 7, the manipulator can be controlled to perform different actions by setting the angle value of each rotary joint separately.
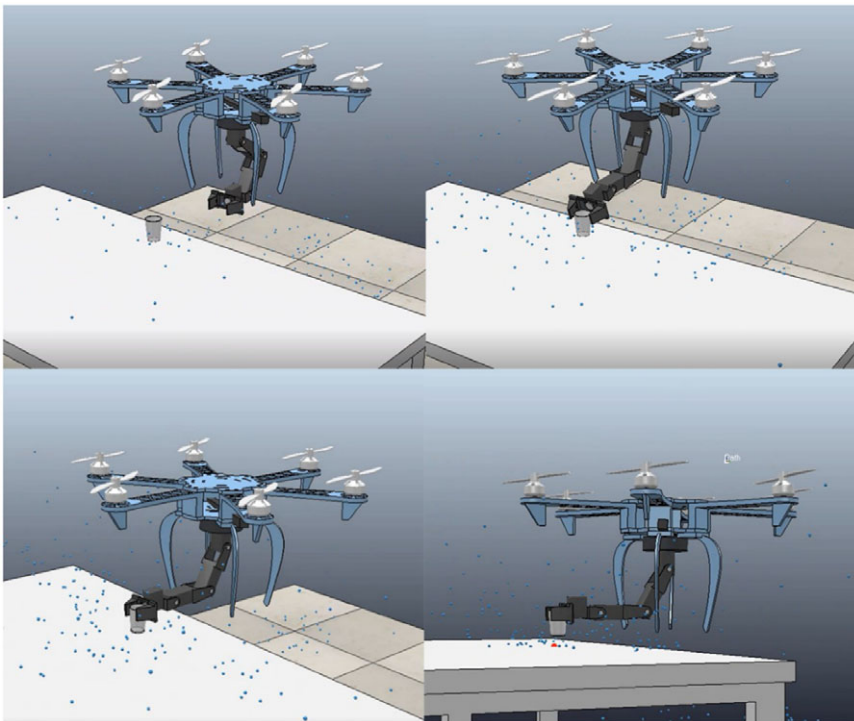
4. Simulation test verification

4.1. Virtual simulation experiment of hexacopter grasping

Before the virtual simulation control of a hexacopter based on hand gesture recognition from a data glove, we simulated the virtual experiment of hexacopter grasping in *CoppeliaSim*. Due to the size limitation
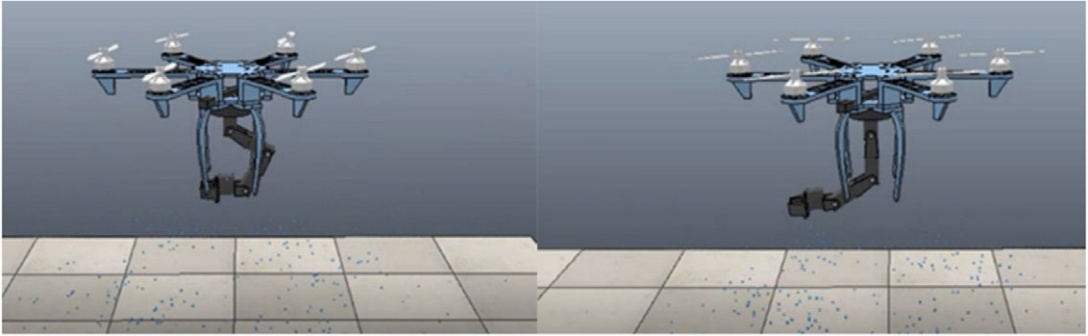
Figure 7. *Motion control of the manipulator in CoppeliaSim.*

Figure 8. *Simulation of cup grabbing process in CoppeliaSim.*

of the gripping device of the manipulator, a cup was used as the grasping targets of this experiment, and we use the simulation camera under the hexacopter to determine the relative position of the target and determine the timing of grasping during the grasping process. Figure 8 shows the entire process of grasping simulation, with the simulation time of about 10 s. The first three sub-figures are the 3*rd*, 7*th*, and 10*th* s, and the last one is the observation picture under different angles. Through Fig. 8, we can intuitively see that the entire grabbing process is smooth and fast. Of course, the moment of grabbing will also have a certain impact on the flying attitude of the hexacopter along the variations of the weight of the grabbed items. During the simulation, it was observed that the manipulator will shake to a certain extent due to the flying attitude of the hexacopter during the grasping process. @@In order to more accurately observe the grasping accuracy of the manipulator, a dummy point is added to the endpoint of the manipulator. And a chart is added to record the position and angle value of the endpoint of the manipulator.
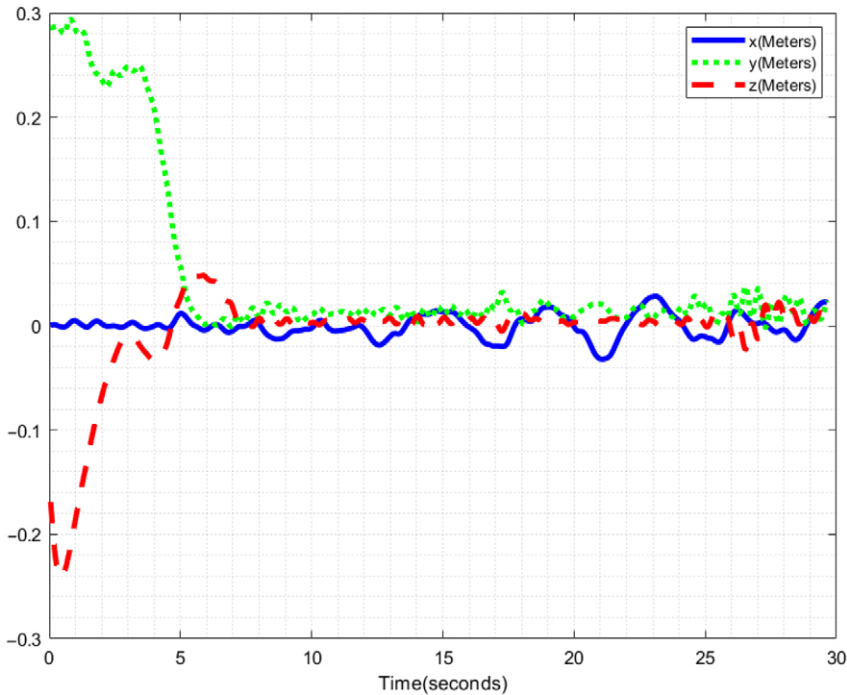
Figure 9. *Three-axis position of the end point of the manipulator.*

Figure 9 shows the three-axis position of the endpoint of the manipulator during grasping simulation.

It can be seen that although the fluctuation range of the three-axis position is within ±0.05 m, the details are different; the position deviation of the *x*-axis is the largest, it is judged to be caused by the influence of the yaw angle of the hexacopter, the deviation of the *y*-axis is moderate, and the deviation of the *z*-axis is the smallest, because it is related to the height control of the hexacopter. Therefore, the grasping accuracy of the manipulator is greatly affected by the flight control attitude of the hexacopter. From this experiment, we can know that the deviation range of the three axes is within ±0.05 m, which is within the acceptable range.

4.2. Virtual flight control experiment of hexacopter based on data glove

To test the effectiveness of the designed virtual flight control performance of hexacopter, we connected the data glove to the personal computer, as shown in Fig. 10.

As we mentioned above, the MPU6050 sensors deployed on the data glove were used to obtain the attitude and acceleration data of the hands and to express different gestures with different characteristics. A total of 10 experimenters were invited to participate in the data glove test, and in the experiment, the experimenter worn the data glove and controlled 10 gestures including: flight commands for take-off, landing, and hover; forward, backward, left, right, up, and down movement commands; grasp and release gripper control order to perform the test, each person repeated the test 10 times on the same gesture, for a total of 110 tests per person. The data were uploaded to the upper computer through the serial port; the instructions from the data glove were also sent to *CoppeliaSim* through the serial port. The snapshot during the experiment to illustrate the relationship using the data glove to control the manipulator is shown in Fig. 11.

One trajectory of the hexacopter flying in the *CoppeliaSim* experiment by using data glove is shown in Fig. 12. What needs illustration is that, for the data glove, we need to place the palm flat for initialization before each use, and calibration is not required before each use.

For the manipulator control using the data glove, as shown in Fig. 13, the grasping and loosening operations can be realized.
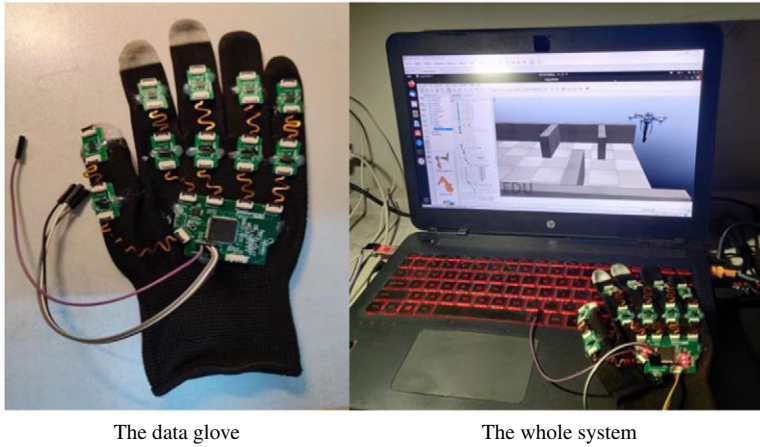
The data glove   The whole system

Figure 10. *System of virtual flight control experiment.*

Grasp   Release

Hover   Landing   Take-off

Forward   Backward   Left
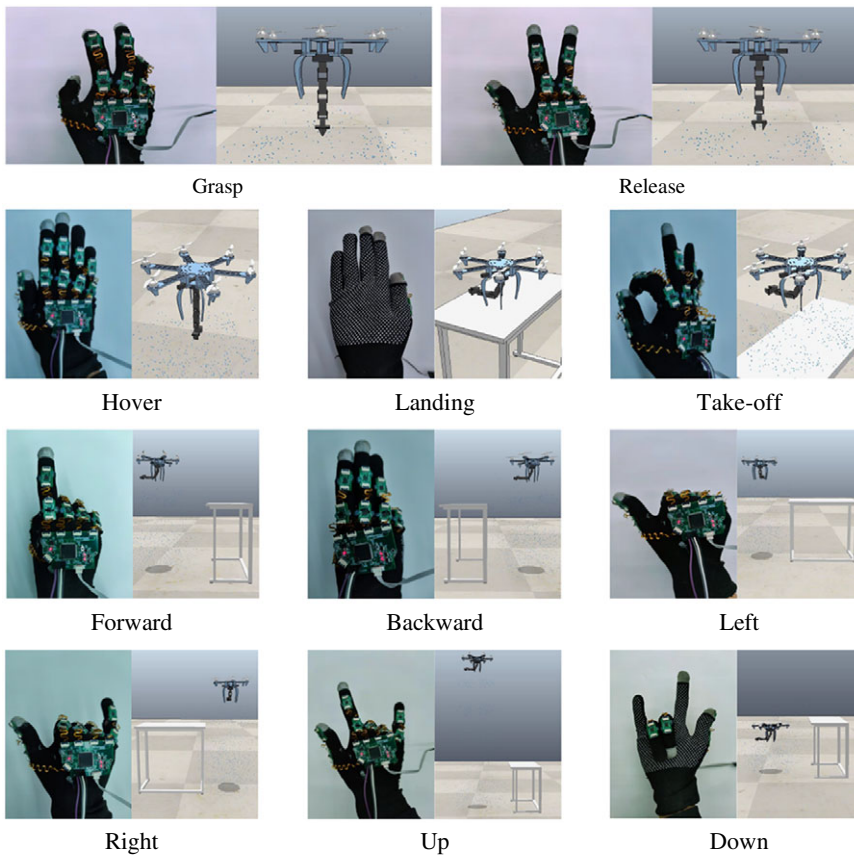
Right   Up   Down

Figure 11. *Snapshot to illustrate the relationship using the data glove to control the manipulator.*

4.3. Discussion

In order to be able to quantitatively describe the performance of virtual control of the hexacopter and the manipulator by using data gloves, we conducted a statistical analysis of the test results, as given in Table I, from which we can see that the overall recognition rate is 84.3%, indicating that the data glove produced has the ability to recognize gestures, but its recognition performance is not superior. In gesture
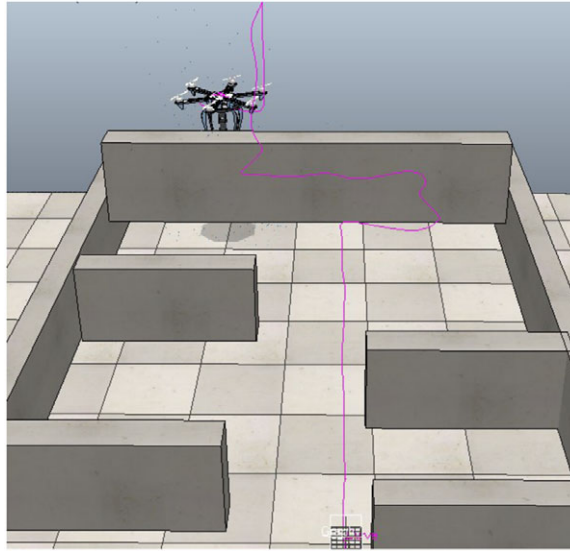
Figure 12. *Flying trajectory in CoppeliaSim.*
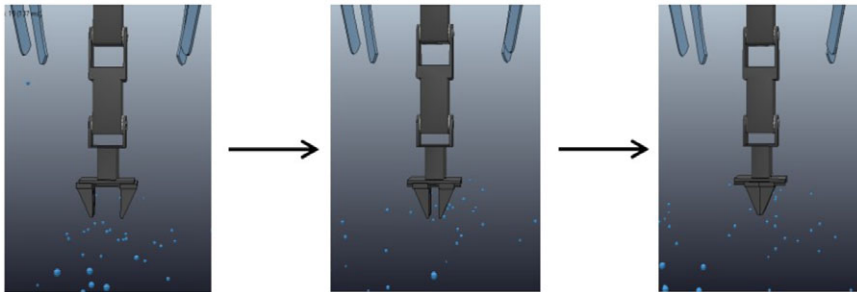
Figure 13. *Manipulator control in CoppeliaSim using data glove.*

recognition, the recognition rate of static gestures is relatively higher than that of dynamic gestures. Among the static gestures which is the top five gestures in Table I, the hover gesture has the highest recognition rate. The average correct rate of static gestures can reach 94%. And the lowest recognition rate of dynamic gestures which is the last six gestures in Table I is upward movement, and the average recognition rate of dynamic gestures is only 76.1%.

Although the results are satisfactory, the accuracy of gesture recognition is not very high, which may be caused by the following reasons.

(1) The accuracy of the three attitude angles measured by the sensor is not high, and there will be a large deviation from the true value, especially in the process of hand movement and turning, the MPU6050 measurement data will have a large deviation because the fingers and the back of the hand are irregular.

(2) There is a certain error between the data measured by the sensor fixed on the glove and the actual finger posture angle.

(3) The test shows that the accuracy of dynamic gestures is low. The reason is that the hand movement will be accompanied by interference such as jitter, which causes the measured acceleration data to contain both the real motion direction data and the uncertain jitter direction data.

Table I. *Statistical results of the experiments (the number of successful tests out of 10).*

Tester	1	2	3	4	5	6	7	8	9	10	Total	Ratio
Take-off	10	9	9	10	9	10	10	10	8	10	95	95%
Landing	10	10	9	10	9	10	9	10	10	10	97	97%
Hover	9	10	10	10	9	10	9	10	9	10	96	96%
Grasp	9	8	10	10	10	10	8	9	10	10	94	94%
Release	9	8	8	10	8	10	9	9	8	9	88	88%
Up	8	7	6	8	8	7	9	7	8	8	76	76%
Down	7	8	6	7	8	6	9	8	6	8	73	73%
Forward	8	6	8	8	6	9	7	8	6	9	75	75%
Backward	6	5	7	6	7	8	8	9	7	9	72	72%
Left	7	8	7	8	9	9	6	7	8	8	77	77%
Right	9	8	7	8	10	8	9	8	8	9	84	84%
Total	97	87	87	95	93	97	93	95	88	100	927	84%

(4) The main clock frequency of STM32F103 is $72M$; it is still difficult to read and process 11 sensor data, the frame information sent per unit time is not enough, and there is a large delay in motion gesture recognition.

(5) Different gestures have some different recognition rate; the reason for this is that for static gestures recognition, the wearing position of the MPU6050 for different people is not exactly the same because the size of the palm is different. The ideal situation is that the sensor on the glove is parallel to the fingers of the straight palm, while the sensor will be offset at different angles due to the different size of the hand. Although the program has a certain error tolerance for the sensor offset, it fails to cover all possible situations, resulting in inaccurate judgment of the final result. For dynamic gesture recognition, the sensitivity of the MPU6050 used in the glove is inconsistent with the up, down, left, and right, front, and back. For example, the up and down jitter occurs during the left and right movement, and the data changes in the up and down direction will be particularly large if the sensitivity to the up and down movement is high, which can lead to large deviations compared with the actual situation. Although this problem can be improved by optimizing the program, it still exists.

5. Conclusions and future works

To provide an intuitive and visual real-time simulation for flight control algorithm verification and external control equipment testing, we studied the virtual interaction and manipulation control of a hexacopter based on hand gesture recognition from a data glove. The hand gesture recognition is studied, and the virtual simulation experiment of hexacopter grasping and virtual flight control experiment were conducted, respectively. The results are satisfactory, although the accuracy of gesture recognition is not very high.

Future research work will mainly focus on how to improve the recognition rate and the response speed of the system.

References

[1] R. Jiao, W. S. Chou, Y. F. Rong and M. J. Dong, "Anti-disturbance attitude control for quadrotor unmanned aerial vehicle manipulator via fuzzy adaptive sigmoid generalized super-twisting sliding mode observer," *J. Vib. Control* **28**(11–12), 1251–1266 (2021).

[2] H. Xiang and L. Tian, "Development of a low-cost agricultural remote sensing system based on an autonomous unmanned aerial vehicle (UAV)," *Biosyst. Eng.* **108**(2), 174–190 (2011).

[3] B. D. Song, K. Park and J. Kim, "Persistent UAV delivery logistics: MILP formulation and efficient heuristic," *Comput. Ind. Eng.* **120**, 418–428 (2018).

[4] Q. Wu, Y. Zeng and R. Zhang, "Joint trajectory and communication design for multi-UAV enabled wireless networks," *IEEE. Trans. Wirel. Commun.* **17**(3), 2109–2121 (2018).

[5] R. Jiao, Z. W. Wang, R. H. Chu, M. J. Dong, Y. F. Rong and W. S. Chou, "An intuitive end-to-end human-UAV interaction system for field exploration," *Front. Neurorobot.* **13**, 1–17.

[6] S.-J. Kim, D.-Y. Lee, G.-P. Jung and K.-J. Cho, "An origami-inspired, self-locking robotic arm that can be folded flat," *Sci. Robot.* **3**(16), eaar2915 (2018).

[7] D. J. Lee and G. P. Jung, "Snatcher: A highly mobile chameleon-inspired shooting and rapidly retracting manipulator," *IEEE Robot. Autom. Lett.* **5**(4), 6097–6104 (2020).

[8] J. Liang, J. Cao and L. Wang, "Design of Multi-Mode UAV Human-Computer Interaction System," *2017 IEEE International Conference on Unmanned Systems (ICUS)*, Beijing, China (2017) pp. 353–357.

[9] A. G. Perera, Y. W. Law and J. Chahl, "UAV-GESTURE: A Dataset for UAV Control and Gesture Recognition," **In:** *Computer Vision - ECCV, 2018 Workshops* (Springer, Cham, 2019) pp. 117–128.

[10] C. Liu and T. Sziranyi, "Real-time human detection and gesture recognition for on-board UAV rescue," *Sensors* **21**(6), 1–21 (2021).

[11] Y. Han, "A low-cost visual motion data glove as an input device to interpret human hand gestures," *IEEE Trans. Consum. Electron.* **56**(2), 501–509 (2010).

[12] B. Fang, F. C. Sun, H. P. Liu and D. Guo, "A novel data glove using inertial and magnetic sensors for motion capture and robotic arm-hand teleoperation," *Ind. Rob. Int. J.* **44**(2), 155–165 (2017).

[13] T. Shi, H. Wang and C. Zhang, "Brain Computer Interface system based on indoor semi-autonomous navigation and motor imagery for Unmanned Aerial Vehicle control," *Expert Syst. Appl.* **42**(9), 4196–4206 (2015).

[14] P. Kumar, J. Verma and S. Prasad, "Hand data glove: A wearable real-time device for human-computer interaction," *Int. J. Adv. Sci. Technol.* **43**, 15–26 (2012).

[15] G. Luzhnica, J. Simon, E. Lex and V. Pammer, "A Sliding Window Approach to Natural Hand Gesture Recognition Using a Custom Data Glove," **In:** *2016 IEEE Symposium on* 3d *User Interfaces*, Greenville, SC, USA (2016) pp. 81–90.

[16] M. J. Dong, B. Fang, J. F. Li, F. C. Sun and H. P. Liu, "Wearable sensing devices for upper limbs: A systematic review," *Proc. Inst. Mech. Eng. H. J. Eng. Med.* **235**(1), 117–130 (2020).

[17] T. Mezzinolu and M. Karakse, "Wearable Glove Based Approach for Human-UAV Interaction," **In:** *2020 IEEE International Symposium on Systems Engineering (ISSE)*, Vienna, Austria (2020) pp. 1–6.

[18] T. Muezzinoglu and M. Karakose, "An intelligent Human-Unmanned Aerial Vehicle interaction approach in real time based on machine learning using wearable gloves," *Sensors* **21**(5), 1–24 (2021).

[19] B. S. Lin, I. J. Lee, S. Y. Yang, Y. C. Lo, J. Lee and J. L. Chen, "Design of an inertial-sensor-based data glove for hand function evaluation," *Sensors* **18**(5), 1–17 (2018).

[20] H. C. T. E. Fernando, A. T. A. De Silvia, M. D. C. De Zoysa, K. A. D. C. Dilshan and S. R. Munasinghe, "Modelling, Simulation and Implementation of a Quadrotor UAV," **In:** *2013 8th IEEE International Conference on Industrial and Information Systems*, Peradeniya, Sri Lanka (2013) pp. 207–212.

[21] J. Meyer, A. Sendobry, S. Kohlbrecher, U. Klingauf and O. von Stryk, "Comprehensive Simulation of Quadrotor UAVs Using ROS and Gazebo," **In:** *SIMPAR 2012: Simulation, Modeling, and Programming for Autonomous Robots*, Berlin, Heidelberg (2012) pp. 400–411.

[22] P. Udvardy, B. Beszedes, B. Toth, A. Foldi and A. Botos, "Simulation of Obstacle Avoidance of an UAV," **In:** *2020 New Trends in Aviation Development (NTAD)*, Stary Smokovec, Slovakia (2020) pp. 245–249.

[23] J. Li, Y. Xu, J. Ni and Q. Wang, "Glove-based virtual hand grasping for virtual mechanical assembly," *Assembly Autom.* **36**(5), 349–361 (2016).