Front. Agr. Sci. Eng. 2025, 12(2): 190–207

RESEARCH ARTICLE

Design and control algorithm of a motion sensing-based fruit harvesting robot

Ziwen CHEN[1,2], Yuhang CHEN[1], Hui LI[1,2], Pei WANG (✉)[1,3,4]

1 Chongqing Key Laboratory of Agricultural Equipment for Hilly and Mountainous Regions, College of Engineering and Technology, Southwest University, Chongqing 400715, China.
2 National Citrus Engineering Research Center, Chongqing 400712, China.
3 Interdisciplinary Research Center for Agriculture Green Development in Yangtze River Basin, Southwest University, Chongqing 400715, China.
4 Institute of Urban Agriculture, Chinese Academy of Agricultural Sciences, Chengdu 610213, China.

KEYWORDS

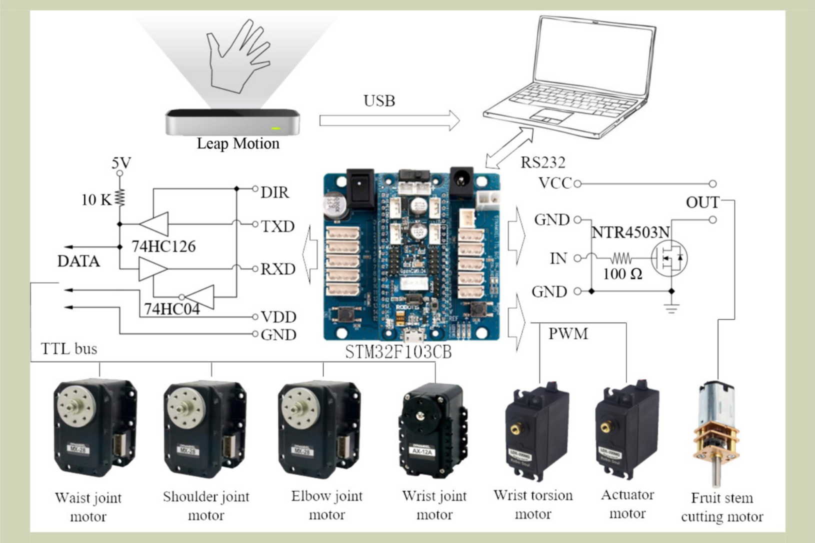
Harvesting robots, human-machine interaction, human-robot collaboration, somatosensory control, Leap Motion controller

HIGHLIGHTS

- An optimized four-step inverse kinematic solution method ensures smooth and precise motion with minimal mechanical interference.
- The robot achieves a fast response time of 74.4 ms, with an average target-picking duration reduced to 6.5 seconds after operator training.
- The system simplifies the picking process using gesture recognition.

Received March 19, 2024;
Accepted September 11, 2024.

GRAPHICAL ABSTRACT



ABSTRACT

In response to the demand of automatic fruit identification and harvesting, this paper presents a human-robot collaborative picking robot based on somatosensory interactive servo control. The robot system mainly consisted of four parts: picking execution mechanism, hand information acquisition system, human-machine interaction interface, and human-robot collaborative picking strategy. A six-degree-of-freedom robotic arm was designed as the picking execution mechanism. The D-H method is employed for both forward and inverse kinematic modeling of the robotic arm. A four-step inverse kinematic optimal solution selection method, including mechanical interference, correctness, rationality, and smoothness of motion, is proposed. The working principle and use of the Leap Motion controller for hand information acquisition were introduced. Data from three types of hand movements were collected and analyzed. Spatial mapping method between the Leap Motion interaction space and operating range of the robotic arm was proposed to achieve a direct correspondence between the cubic interaction box and the cylindrical space of the fan ring of the robotic arm. The test results demonstrated that the average response time of the double-click picking command was 332 ms. The average time consumption for somatosensory control targeting was 6.5 s. The accuracy rate of the picking gesture judgment was 96.7%.

1 Introduction

Fresh fruit harvesting and picking consume the most labor throughout the entire operation of fruit production. Since the quality of freshly harvested fruits directly affects their sales and economic benefits, the picking process is complex, with significant challenges to mechanization, and predominantly relies on manual labor. However, the use of a single harvesting machine or manual labor alone cannot adequately meet the demands for improving harvesting efficiency and reducing costs. With the development of precision agriculture and artificial intelligence, the concept of human-machine collaborative harvesting robots has gained increasing attention. These robots can effectively integrate human and machine efforts, achieving a collaborative operation between humans and machines, which holds significant importance in fresh fruit harvesting. With the decrease in labor force and a significant increase in labor costs in China, automated and intelligent fruit and vegetable harvesting has become a developmental necessity[1–5]. Scholars both domestically and internationally have conducted research on automatic harvesting technologies, mainly employing monocular or binocular cameras for fruit identification and localization. However, due to the complex and variable lighting conditions in orchards, heterogeneous tree canopy growth and various unstructured environmental factors, such as severe fruit occlusion or overlapping, visual fruit recognition and localization have become challenging. Despite extensive research by scholars on fruit recognition under overlapping and occlusion conditions[6–8], path planning and obstacle avoidance in complex canopy environments[9–11], and recognition under different lighting and shadow conditions[12–14], the operational effectiveness of on-site picking robots still fails to meet the needed standards. Further issues persist in the accuracy, real-time performance, effectiveness and applicability of machine vision recognition and localization algorithms[15,16].

Human-computer interaction is the science of studying the interaction between systems and users, which involves using a certain dialog language between humans and systems to complete specific tasks through a certain interaction method. With the rapid development of somatosensory technology, interactive technologies using somatosensory input have been applied in education, medicine, machinery, aerospace, economics, scientific research, and other fields[17–20], but have seen limited application in agriculture[21]. Xu et al.[22] designed a fruit-picking robot based on biomimetics that imitates human upper limb picking behaviors using the Kinect depth camera, with simulation experiments demonstrating an average recognition rate of 90% for picking actions. Quan et al.[23] has proposed a decision-making and planning algorithm for a three-arm greenhouse robot based on Kinect somatosensory manipulation, achieving precise control of robot arms and gestures. Wu et al.[21] developed a virtual crop somatosensory interaction system using cloud computing and somatosensory interaction technology, allowing for plant translation, rotation, growth, morphological changes, and model updates through gestures. Hu et al.[24] had used the Kinect V2 sensor to propose a non-destructive measurement method for leafy vegetable growth, with results revealing high measurement accuracy and a good linear relationship with the projection area and related measurements. Compared to Kinect, Leap Motion is equipped with an infrared filter membrane, which eliminates accuracy deviations caused by differences in light and darkness, making it more adaptable to various environments and providing higher precision in hand information collection. The effective detection range of Leap Motion is 25–600 mm, and it has a wide field of view, which further enhances the accuracy of data collection. In this paper, Leap Motion is used as the data collection tool.

Considering the requirements, characteristics, and existing problems of fruit-picking robots, and acknowledging the outstanding performance of the human visual system[25], integrating human-computer interaction systems to achieve human-machine collaboration has been proposed. This approach delegates complex cognitive tasks to humans while leaving repetitive and physically demanding tasks to be handled by machines, thereby reducing labor intensity and avoiding potential injuries in some hazardous agricultural production processes[22]. This paper proposes a human-machine collaborative picking robot that utilizes somatosensory interaction for human-machine interaction.

Operators primarily perform fruit recognition and target positioning, enabling remote, simple and real-time control of picking robots through somatosensory human-machine interaction. Compared with traditional automated picking robots that can only operate within preset paths and task ranges, and manual picking, a human-machine collaboration picking robot using a somatosensory human-computer interaction system can achieve remote, simple, real-time and safe picking operations. This provides a new approach and model for the practical application of picking robots, which can improve efficiency and reduce economic costs in small-scale complex picking environments. This allows individuals to achieve repetitive, time-consuming and labor-intensive picking tasks through simple hand movements. Additionally, for scenarios involving high tree fruit picking, such as coconut harvesting, it reduces the labor and risk associated with manual tree climbing.

2 Materials and methods

2.1 Implementation of the fruit harvesting robot

A fruit harvesting robot was equipped with the fruit-picking actuator and the hand orientation signal detection-processing system. As shown in Fig. 1, the fruit-picking actuator had six degrees of freedom which including five motion joint and a clogging joint of the end-effector. Thus, the robot had high flexibility and could catch and pick the fruit at an arbitrary posture (position and orientation) in three-dimensional motion space of the fruit-picking actuator. The joints of waist, shoulder and elbow were driven using servo motors (Dynamixel series MX28A, ROBOTIS Co., Ltd., Seoul, Korea), while the wrist joint was driven using another servo motor (Dynamixel series AX12A, ROBOTIS Co., Ltd.). Proportional-integral-derivative control mode was adopted in these motors for precise position and velocity controlling. Meanwhile, it was also installed with a feedback function for position, temperature, load and input voltage. The wrist swivel joint and gripping unit were driven with a digital servo (LDX-335MG, Lobot, Shenzhen, China). Thus, the hand of the robotic arm was able to grip the fruit and twist the stalk. Above the hand of this robotic arm, a miniature geared motor (GA12-N20, Handson Technology, Johor, Malaysia.) combined with a double-edged rotary knife was installed to cut the stalk. The reduction or gear ratio of the miniature geared motor was 210:1 with an output torque of 1.5 kg cm. The parameters of each motor are given in Table 1.

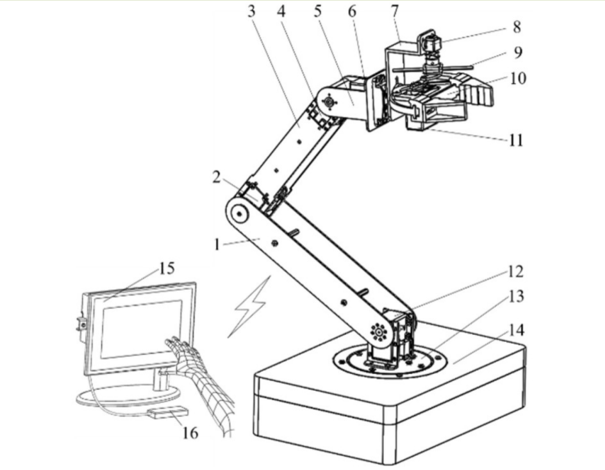


Fig. 1 Man-machine cooperation fruit-picking robot. 1, Main arm; 2, elbow motor; 3, forearm; 4, wrist motor; 5, end arm; 6, wrist-torsion joint motor; 7, bracket; 8, fruit stalks cutting motor; 9, cutting knife; 10, actuator; 11, actuator motor; 12, shoulder joint motor; 13, waist joint; 14, control box; 15, computer; and 16, Leap Motion.

The hand orientation signal was collected and processed via a system that includes a Leap Motion controller (Leap Motion, Inc., San Francisco, CA, USA) and a computer. With this system, the movement of a human hand in a certain space could be mapped onto the robotic arm in its working space. Then, the human-machine collaborative picking was realized through somatosensory interaction. An embedded system (OpenCm 9.04, ROBOTIS Co., Ltd.) with a 32-bit processor (STM32F103, STMicroeletronics, Geneva, Switzerland) was selected as the main controller for this robot. The system had output voltages of 12 and 7.4 V. An expansion board (Open CM485EXP, ROBOTIS Co., Ltd.) was combined with the embedded system, so that the system could provide a multiple TTL bus interface. The data exchange between the controller and the servos was realized via message communication. The digital servo was controlled using the pulse-width modulation signals with frequency of 0.5–2.5 ms, and the miniature geared motor was controlled using a field effect transistor (NTR4503N, ON Semiconductor, Scottsdale, USA). A controlling program with human-computer communication

Table 1 D-H parameters

i	θ_i (°)	d_i (mm)	a_{i-1} (mm)	α_{i-1} (°)
1	θ_1	53	0	90
2	θ_2	0	200	0
3	θ_3	0	179	0
4	θ_4	0	140	0

interface was designed using Visual Studio 2017 on the computer. With this program the Leap Motion controller can complete the data collection, analysis, selection and optimization. Based on that, the controlling patterns for the robotic arm could be obtained. The communication between the computer port and the motion main controller was established using a RS232 serial port. Figure 2 shows a schematic of the control system of this fruit harvesting robot.

2.2 Parameter calibration of D-H model of the fruit-picking actuator

The D-H model is a standard framework for the kinematics modeling of robotic arm which was introduced by Denavit and Hartenberg in 1955[26]. In this model, reference coordinate systems of the base, the actuator and each joint were established.

Figure 3 presents a schematic of the mechanism of the picking manipulator. While the actuator hand should be kept horizontally, the twist joint of wrist and the closing joint of actuator could be omitted. Meanwhile, the angle of joint 4 was correlated to angle joints 2 and 3. Figure 3(b) shows a simplified schematic of the mechanism of the picking manipulator and the coordinate system of each joint. The parameters of D-H model are presented in Table 1.

2.3 Solving for forward kinematics of the robotic arm

While the parameters of each twist joint were known, solving the forward kinematics of the robotic arm could obtain the position and orientation of the end actuator via solving the homogeneous transformation matrices. According to the D-H coordinate and the parameters in Table 1, homogeneous transformation matrices between each coordinate could be presented as T_{01}, T_{12}, T_{23}, and T_{34}. The homogeneous transformation matrix of: the manipulator base can be expressed as T_{04} by multiplying the matrices T_{01}, T_{12}, T_{23}, and T_{34}:

$$T_{01} = \begin{bmatrix} c_1 & 0 & s_1 & 0 \\ s_1 & 0 & -c_1 & 0 \\ 0 & 1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

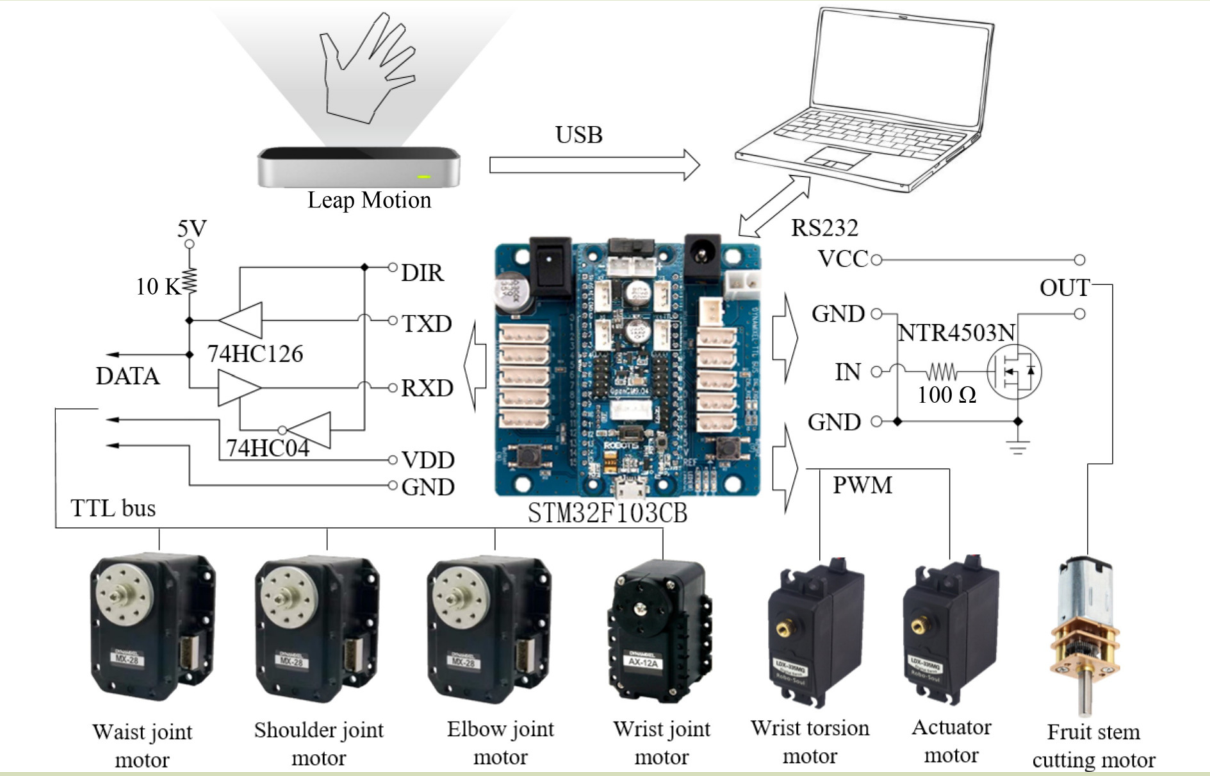


Fig. 2 Circuit system of man-machine cooperation fruit-picking robot.

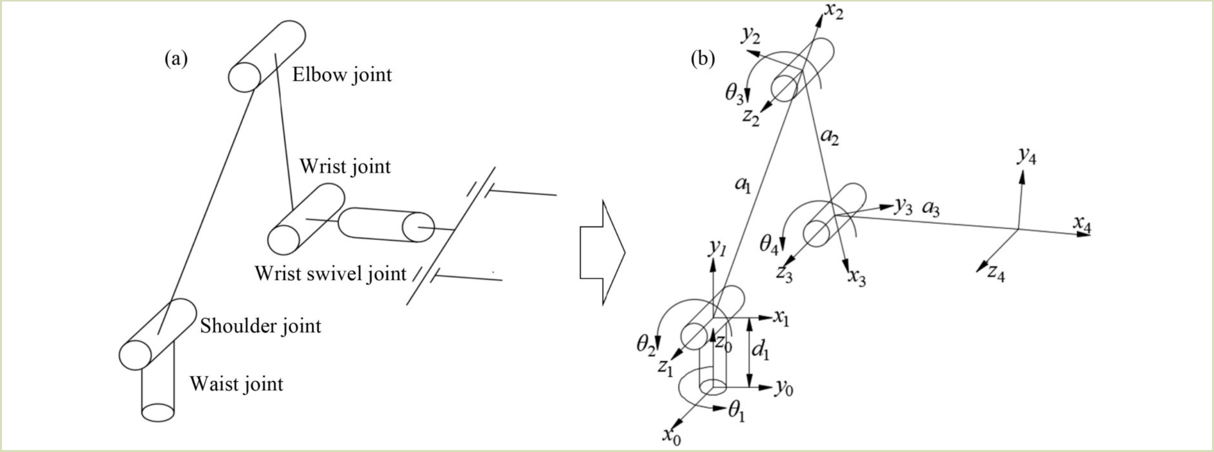


Fig. 3 Schematic diagram of manipulator. (a) Simplified diagram; (b) coordinate system construction.

$$T_{12}=\begin{bmatrix} c_2 & -s_2 & 0 & a_1c_2 \\ s_2 & c_2 & 0 & a_1s_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_{23}=\begin{bmatrix} c_3 & -s_3 & 0 & a_2c_3 \\ s_3 & c_3 & 0 & a_2s_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_{34}=\begin{bmatrix} c_4 & -s_4 & 0 & a_3c_4 \\ s_4 & c_4 & 0 & a_3s_4 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{aligned} T_{04} &= T_{01}T_{12}T_{23}T_{34} \\ &= \begin{bmatrix} c_{234}c_1 & -s_{234}c_1 & s_1 & c_1(a_3c_{234}+a_2c_{23}+a_1c_2) \\ c_{234}s_1 & -s_{234}s_1 & -c_1 & s_1(a_3c_{234}+a_2c_{23}+a_1c_2) \\ s_{234} & c_{234} & 0 & a_3s_{234}+a_2s_{23}+a_1s_2+d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned} \tag{1}$$

where, s_1 is sinθ_1, c_1 is cosθ_1, c_{23} is cos(θ_2 + θ_3), c_{234} = cos(θ_2 + θ_3 + θ_4), and so forth.

2.4 Solving for inverse kinematics of the robotic arm

Inverse kinematics calculates the configuration of each joint of the robot based on the known position and orientation of the end-effector, enabling it to reach the target pose. This is crucial for the precise control of the actions and path of a picking robot. Equation (1) can be expressed as:

$$T_{04}=\begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2}$$

where, n_x, n_y, n_z, o_x, o_y, o_z, a_x, a_y, and a_z represent the cosines of the axis of coordinate {4} to each axis of coordinate {0}, P_x, P_y, and P_z represent the coordinate position of the origin of coordinate {4} in coordinate {0}.

Equation (1) can be transformed as:

$$T_{01}^{-1}T_{04}=T_{12}T_{23}T_{34} \tag{3}$$

Thus, the two sides of Eq. (3) can be respectively expressed as:

$$T_{01}^{-1}T_{04}=\begin{bmatrix} n_xc_1+n_ys_1 & o_xc_1+o_ys_1 & a_xc_1+a_ys_1 & p_xc_1+p_ys_1 \\ n_s & o_s & a_w & p_w-d_1 \\ n_xs_1-n_yc_1 & o_xs_1-o_yc_1 & a_xs_1-a_yc_1 & p_xs_1-p_yc_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4}$$

$$T_{12}T_{23}T_{34}=\begin{bmatrix} c_{234} & -s_{234} & 0 & a_3c_{234}+a_2c_{23}+a_1c_2 \\ s_{234} & c_{234} & 0 & a_3s_{234}+a_2s_{23}+a_1s_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5}$$

When the elements at the third row and forth column of matrices in Eqs. (4) and (5) are equal, the analytical solution of θ_1 should be:

$$\theta_1=\begin{cases} \arctan\dfrac{p_y}{p_x}+\pi i,\{(p_x,p_y)|-t_x\leqslant\dfrac{p_y}{p_x}\leqslant t_x;\text{when},\dfrac{p_y}{p_x}\geqslant 0,i=-1,0;\text{when},\dfrac{p_y}{p_x}<0,i=0,1\} \\ \dfrac{\pi}{2},\{(p_x,p_y)|\dfrac{p_y}{p_x}\geqslant t_n\} \\ -\dfrac{\pi}{2},\{(p_x,p_y)|\dfrac{p_y}{p_x}\leqslant -t_n\} \end{cases} \tag{6}$$

where, t_π is 100, which is the discrimination threshold to assign θ_1 as π/2 or −π/2; other parameters in this formulation can be solved using forward kinematics as described above. To ensure the robotic arm can move with the whole range of motion of 360°, the range θ_1 of needs to be [−π, π].

When the element in the first row and the forth column of the matrices in Eqs. (4) and (5) are equal, and the element in the second row and the forth column of two matrices is equal as well, there will be the following formulas:

$$p_x c_1 + p_v s_1 = a_3 c_{234} + a_2 c_{23} + a_1 c_2 \tag{7}$$

$$p_z - d_1 = a_3 s_{234} + a_2 s_{23} + a_1 s_2 \tag{8}$$

Thus:

$$\begin{aligned}(p_x c_1 + p_y s_1)^2 + (p_z - d_1)^2 &= (a_3 c_{234} + a_2 c_{23} + a_1 c_2)^2 \\ &+ (a_3 s_{234} + a_2 s_{23} + a_1 s_2)^2\end{aligned} \tag{9}$$

According to the trigonometric algorithm:

$$\begin{aligned}s_2 s_{23} + c_2 c_{23} &= \sin\theta_2 \sin(\theta_2 + \theta_3) + cos\theta_2 cos(\theta_2 + \theta_3) \\ &= cos[(\theta_2 + \theta_3) - \theta_2] = c_3\end{aligned} \tag{10}$$

Substitute Eq. (10) into Eq. (9), the value of cosθ_3 and sinθ_3 as well as the angle of θ_3 can be calculated as:

$$c_3 = \frac{(p_z c_1 + p_y s_1 - a_3 c_{24})^2 + (p_z - d_1 - a_3 s_{24})^2 - a_2^2 - a_1^2}{2 a_z a_1} \tag{11}$$

$$s_3 = \pm \sqrt{1 - c_3^2} \tag{12}$$

$$\theta_3 = \pm \arccos(c_3) \text{or} \arcsin(s_3) \text{or} \arctan\left(\frac{s_3}{c_3}\right) \tag{13}$$

According to the trigonometric algorithm:

$$\begin{cases} c_{23} = c_2 c_3 - s_2 s_3 \\ s_{23} = s_2 c_3 + c_2 s_3 \end{cases} \tag{14}$$

where, the values of cosθ_2 and sinθ_2 as well as the angle of θ_2 can be calculated by simultaneously solving the Eqs. (7), (8), and (14).

$$s_2 = \frac{(\alpha_2 c_3 + \alpha_1)(p_3 - d_2 - \alpha_2 s_{234}) - \alpha_2 s_3 (p_3 c_1 + p_3 s_1 - \alpha_3 c_{234})}{(\alpha_2 c_3 + \alpha_1)^2 + (\alpha_2 \varepsilon_3)^2} \tag{15}$$

$$c_2 = \frac{(a_2 c_3 + a_1)(p_x c_1 + p_x s_1 - a_3 c_{234}) + a_2 s_3 (p_x - d_1 - a_3 s_{234})}{(a_2 c_3 + a_1)^2 + (a_2 s_3)^2} \tag{16}$$

$$\theta_2 = \pm \arccos(c_2) \text{or} \arcsin(s_2) \text{or} \arctan\left(\frac{s_2}{c_2}\right) \tag{17}$$

Thus, it is necessary to calculate the value of s_{234}, which means that θ_{234} (which donates $\theta_2 + \theta_3 + \theta_4$) should be known. When both sides of the Eq. (1) are left multiplied by $T_{34}{}^{-1}T_{23}{}^{-1}T_{12}{}^{-1}$ and $T_{01}{}^{-1}$ the following can be obtained:

$$T_{34}^{-1} T_{23}^{-1} T_{12}^{-1} T_{01}^{-1} T_{04} = E \tag{18}$$

where, E is a 4 × 4 identity matrix.

When the elements of the second row and the first column in the matrices on each side of the Eq. (18) are equal, θ_{234} can be calculated after simplification with trigonometric algorithm:

$$\theta_{234} = \arctan\frac{n_z}{n_x c_1 + n_y s_1} + \pi i = 0, 1 \tag{19}$$

2.5 Optimization method for inverse solution of the robotic arm

The inverse solution of the robotic arm can have multiple results. The angles of all the joints of the robotic arm are not unique even the endpoint is fixed at a certain special position. This means that the robotic arm should be in different postures when its joint angle combinations are different. This will severely affect the control of the robotic arm. Thus, the control parameters should be obtained by selecting a reasonable inverse solution. During this, several issues should be considered, such as the motion interference, movement mode rationality and trajectory planning of the robotic arm. Meanwhile, the boundary condition constraint of each joint angle should be defined. As a result, the reasonable solution can be obtained.

In Fig. 4, a motion posture schematic of the robotic arm is presented under the plane (y_0, O, z_0) in the base coordinate system. The movement of the fruit-picking actuator in three-

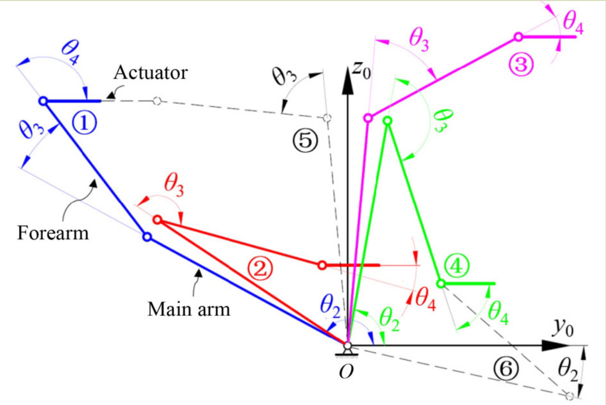


Fig. 4 Motion posture schematic of the robotic arm. The lines ①–⑥ in different colors represent different robotic arm postures of the manipulator under the plane (y_0, O, z_0) in the base coordinate system (which means the waist joint is fixed).

dimensional motion space of the manipulator could be disassembled into the movement of the waist joint and the movement of other joints in the plane of (y_0, O, z_0). In D-H model, the joint angles of the robotic arm are defined. In this fruit-picking manipulator, the plus or minus sign of the joint angle determined the position of the forearm and the main arm in the range of robot base-(waist)-body-(shoulder)-main arm-(elbow)-forearm-(wrist)-actuator. When the arm is at the left side of the extension cord of its forearm (as there was no extension cord of the robot base, the axis y_0 is selected to represent the extension cord), the joint angle between the forearm and the main arm was defined as positive, otherwise, the joint angle is defined as negative. For example, the arm postures ④ and ⑥ present two groups of inverse solutions of the robotic arm which enable the actuator to be the same special position. The angle θ_2 in arm posture ④ is positive and the angle θ_2 in arm posture ⑥ is negative. However, in practice, the robot base will limit the movement space of the manipulator. Thus, the angle of θ_2 should be set with in [0, π] to ensure that the main arm always moves above the robot base (including the waist joint and the shoulder joint), which can be expressed as:

$$\theta_2 = \begin{cases} \arccos(c_2);\ c_2 \in [-1,1] \\ \text{no solution};\ c_2 \in (-\infty,-1) \cup (1,+\infty) \end{cases} \tag{20}$$

where, θ_3 is the elbow joint angle between the main arm and the forearm. To keep the robotic arm moves smoothly and avoid intensive load for the elbow joint, θ_3 should not be changed substantially. Thus, the inverse solution of θ_3 in [0, $-\theta_{3\max}$] was always selected in this optimization method, which can be expressed as:

$$\theta_3 = \begin{cases} -\arccos(c_3);\ c_3 \in [c_{3\min},1],\ c_{3\min} = \cos(\theta_{3\max}) \\ \text{no solution};\ c_3 \in (-\infty, c_{3\min}) \bigcup (1,+\infty) \end{cases} \tag{21}$$

where, $\theta_{3\max}$ is 150°, which is the designed maximum rotation angle of the elbow joint.

The angle of θ_4 is calculated to ensure the actuator horizontally facing the fruit during the picking procedure. This can be obtained according to the geometric relationship (Fig. 4):

$$\theta_4 = \left| \ \theta_3 \ \right| - \theta_2 \tag{22}$$

Generally, the inverse solution can be optimized by the following steps:

(1) Mechanical interference judgment: According to the mechanical interference, parameters like t_π and $\theta_{3\max}$ can only be set in certain domain. That will exclude some inverse solutions which will induce the mechanical interference.

(2) Correctness judgment of inverse solution: The inverse solutions need to be substituted into the forward kinematics equations, so that wrong solutions can be excluded.

(3) Rationality judgment of robotic arm posture: The posture and mechanical properties need to be considered to keep the manipulator standing stable and the joint motors running well, so the joint angles need to be set in certain domains. Thus, unreasonable solutions can be further excluded.

(4) Motion smoothness judgment: When the end actuator moves a short distance in the working space, the joint angles need to be varied as little as possible to ensure the manipulator moves smoothly.

The optimized inverse solution is then selected after the four-step judgment as described above.

2.6 Simulation of the inverse kinematics using MATLAB Robotic Toolbox

The optimized inverse solution was verified via simulation using MATLAB Robotic Toolbox. Five points marked as A to E were selected as the terminal position of the end actuator in the simulation space. The coordinate of the five points were listed in Table 2. The inverse solutions were calculated and optimized

Table 2 Manipulator angle solution of inverse kinematics

Point	Terminal position (m)			Randomly selected solution of joint angle (°)				Optimized solution of joint angle (°)			
	p_x	p_y	p_z	θ_1	θ_2	θ_3	θ_4	θ_1	θ_2	θ_3	θ_4
A	−0.45	0.01	0.05	178.7	−33.5	70.3	−36.8	178.7	32.4	−70.3	−37.9
B	−0.35	0.23	0.25	146.7	59.5	−51.6	−7.9	146.7	59.5	−51.6	−7.9
C	0.01	0.37	0.35	88.5	45.1	15.1	−60.2	88.5	59.4	−15.1	−44.3
D	0.28	0.28	0.25	45.0	7.9	63.2	−71.1	45.0	67.2	−63.2	−4.0
E	0.25	0.01	0.15	2.3	−18.5	134.8	116.4	2.3	101.2	−134.8	33.7

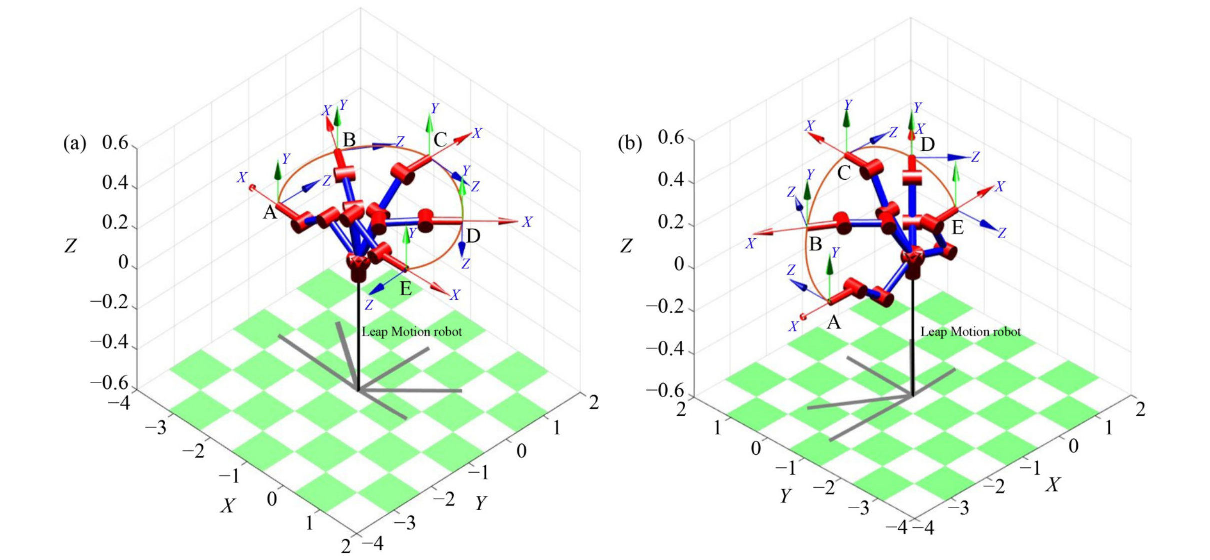


Fig. 5 Manipulator simulation of inverse kinematics. (a) Optimized solution; (b) randomly selected solution.

due to the four-step judgment as described above. The optimized solution and a group of other solution randomly selected from all the inverse solution groups are given in Table 2. Figure 5 presents the postures of the picking manipulator reaching points A to E. It shows that the optimized inverse solution enabled the robotic arm to rotated with smaller joint angles. For example, if the end actuator is to move from point A to point B, the elbow joint angle θ_3 will need to rotate just 18.7° using the optimized inverse solution while 121.9° using the randomly selected inverse solution.

2.7 Motion information collection and optimization using Leap Motion

The Leap Motion controller (LMC) is a consumer grade motion gesture sensor that was launched in 2013. It allows users to interact with the computer via hand gesture recognition. The sensor provides users a natural user interface and a human-computer interaction platform to precisely track the hand and finger gestures with the precision of 0.01 mm. LMC is combined with a vision-based system and a proximity-based system. The vision system contains two CCD camera which can obtain the hand and figure gestures. The proximity system contains three infrared LEDs which can measure the distance of the hand or other objects above the Leap Motion controller[27,28]. The right-handed Cartesian coordinate system was applied while LMC acquiring the data. As shown in Fig. 6, the upper surface center of LMC is set as the origin of the right-handed Cartesian coordinate system. The X-axis is parallel to the long edge of the sensor surface toward right side, the Y-axis is perpendicular to the sensor surface toward upside and the Z-axis is perpendicular to the sensor surface toward the user. The hands should be 25–600 mm and no more than 150° above the sensor.

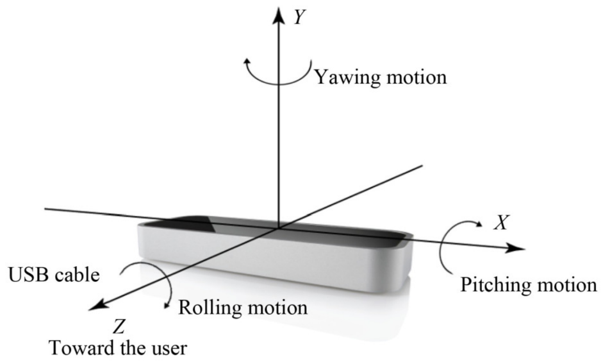


Fig. 6 The right-handed Cartesian coordinate system applied in Leap Motion controller.

The hand motion information collection program was developed using Visual Studio 2017 with the Leap Motion software development kit (UltraLeap, Bristol, UK). With this program, the gesture information could be collected, including the normal vector, the special position, the direction and the posture of the palm. Palm motion direction was predefined in the program, such as pitching for rotation around the X-axis,

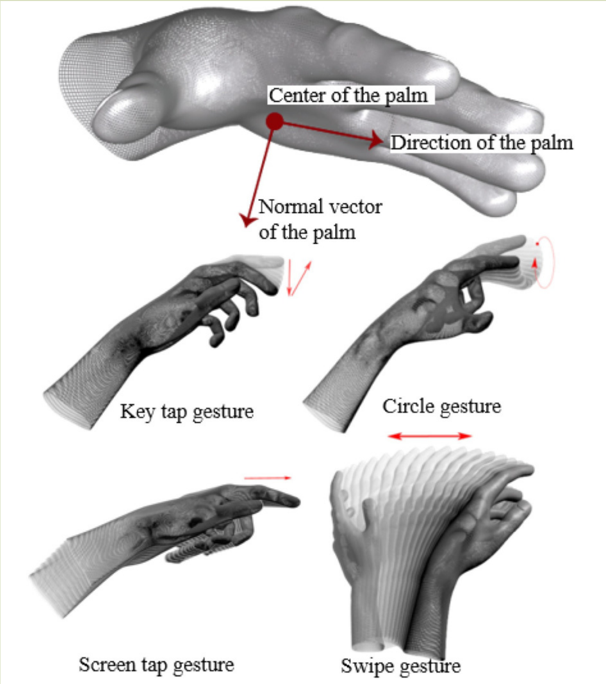


Fig. 7 Palm information and gesture.

yawing for rotation around the *Y*-axis and rolling for rotation around the *Z*-axis. Leap Motion provides some gestures of specific motion modes including key tap, screen tap, circle and swipe (Fig. 7).

To reduce the impact of data fluctuation during the static and dynamic palm gesture information collection, data filtering methods were created by testing the position information collection while the palm kept statically, and moved under straight line or broken line. Figure 8 shows the palm position. For static operation, the end actuator should keep static, which means the slight movement of the user's hand should be filtered. Six different positions were selected in the detection region of LMC to test the combined fluctuation effect of the operator's hand and the sensor. As given in Table 3, the average fluctuation of the static position sensing is in a 2.0 mm × 1.4 mm × 1.9 mm range. Thus, in further programming, palm position movement within the range of 2.0 mm × 1.4 mm × 1.9 mm will be ignored and the end actuator will not move.

For dynamic operation, when a hand is moved smoothly, the end actuator is designed to also move smoothly. However, due to the fluctuation of the LMC sensing information, significant outliers may occur as shown in Fig. 8(b). To avoid an outlier effect on actuator control, the locally weighted linear regression (LWLR) was used in this study to filter the data collected by the LMC. With LWLR, a segment of data before and after the center point was applied for weighted linear regression using the weight function *w*. The output $\hat{y}$ is the value of the center point on the regression model. The procedure can be expressed as:

$$J(\theta) = \min\left\{\sum_i w^{(i)}\left(y^{(i)} - \theta^T x^{(i)}\right)^2\right\} \tag{23}$$

$$w^{(i)} = \exp\left(-\frac{(x^{(i)} - x)^2}{2\tau^2}\right) \tag{24}$$

where, *J(θ)* is the loss function, $w^{(i)}$ is the weight function, $y^{(i)}$ is the dependent variable, $x^{(i)}$ is the independent variable, θ^T is a parameter unrelated to $x^{(i)}$, *i* is the number of the collected data, *x* is the predicted value of the sample point, and *τ* is the parameter controlling the weight changing rate. The weight fits to Gaussian kernel function:

$$\begin{cases} \left|x^{(i)} - x\right| \approx 0, w^{(i)} \approx 1; \\ \left|x^{(i)} - x\right| \approx +\infty, w^{(i)} \approx 0; \end{cases} \tag{25}$$

Thus, the sample data near the predicted point has greater weights. Then, the predicted value can be calculated as:

$$\widehat{y} = \theta^T x^{(i)} \tag{26}$$

Figure 8(b) presents the comparison of original and predicted data via smoothing filtering. The data processing successfully replaced the fluctuated data with predicted data to ensure the end actuator can move smoothly. The global average values of sample position on *Y*-axis before and after smoothing filtering were –24.9 and –24.8 mm, respectively, and the global variance were 9.76 and 9.65 mm^2, respectively. While filtering using the LWLR, the local average values of sample position on *Y*-axis before and after smoothing filtering were –15.8 and –15.4 mm, respectively, and the local variance were 4.13 and 2.39 mm^2, respectively. This shows that the LWLR significantly excluded the fluctuated data while the palm moved generally in a tendency.

2.8 Somatosensory servo control strategy

2.8.1 Space coordinate system mapping

Both the LMC coordinate and the motion space coordinate of the manipulator have edge limitations. To realize the correspondence between the positions of two coordinates, an algorithm needs to be defined to describe the mapping relationship of these coordinate systems. When the user's hand is near or beyond the detection edge of LMC, the signal of palms will lack fidelity. To ensure the signal accuracy of the palm position, LMC defined a virtual interaction box as shown in Fig. 9. It is a 235 mm × 147 mm × 235 mm cube with the

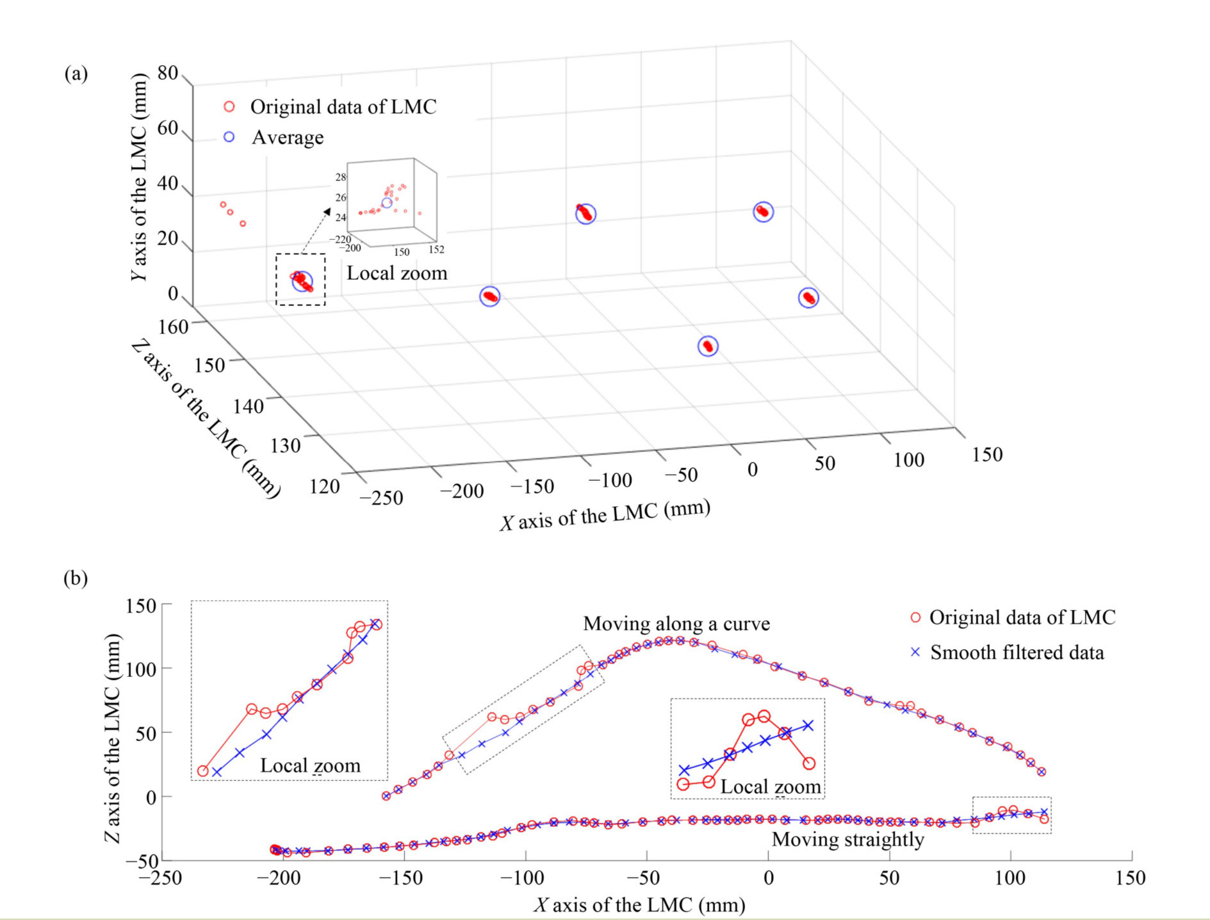


Fig. 8 Data acquisition and smoothing of Leap Motion. (a) Data acquisition of static palm; (b) data acquisition of moving palm.

Table 3 The range of data fluctuation in Cartesian coordinates

Sampling point	*X*-axis	*Z*-axis	*Y*-axis
1	3.1250	3.2914	4.0960
2	2.2770	1.1648	1.3810
3	2.2770	1.1648	2.8670
4	1.6297	1.1404	0.9140
5	1.2998	0.7421	1.3270
6	2.6260	1.1946	1.0130
Average	2.0528	1.4385	1.9330

center at (0, 200, 0) mm in the LMC coordinate.

Figure 10 presents the motion region of the end actuator under the limitation of each joint parameters. In practice, the manipulator and the operator are usually facing the same side. Thus, the waist joint angle was set within [0, π] as the top view demonstrated in Fig. 10(a). This shows that the edge of the motion region was hemispherical. In Fig. 10(b), a cross section via the waist Joint revolute pair ($x_0 = 0$) was displayed. It shows that the inner part of the globular cover is hollow. According to the operation position of the manipulator, the working region was set inside the red frame in Fig. 10(b), in which $y \in [0.24\ \text{m}, 0.44\ \text{m}]$ and $z \in [-0.09\ \text{m}, 0.27\ \text{m}]$. Further, the working region of the manipulator was set in a sector with angle of 120° as shown in Fig. 10(a). Therefore, the actuator of the manipulator will work in a sector ring cylinder.

To realize the mapping between the interaction box of LMC and the operation sector ring cylinder of the actuator, a position p_l (x_l, y_l, z_l) in LMC box was mapping to the position p_a in the operation region. p_a was expressed as (θ, $-r$, z_a) using a combined coordinate system of polar coordinates and Cartesian coordinates. The mapping relationship between each parameter is:

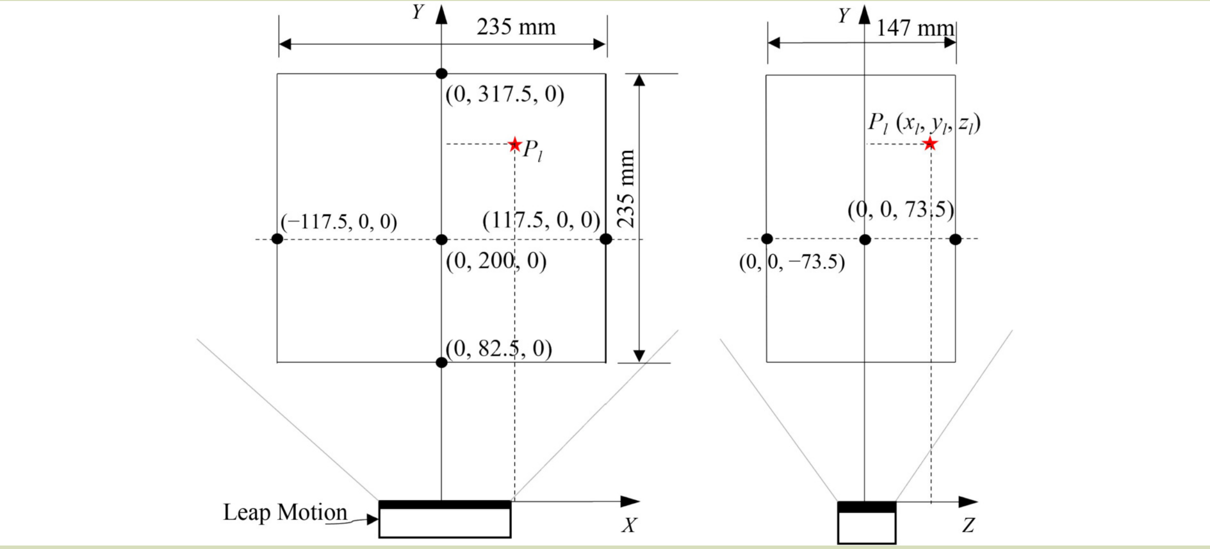


Fig. 9 The virtual interaction box of LMC.

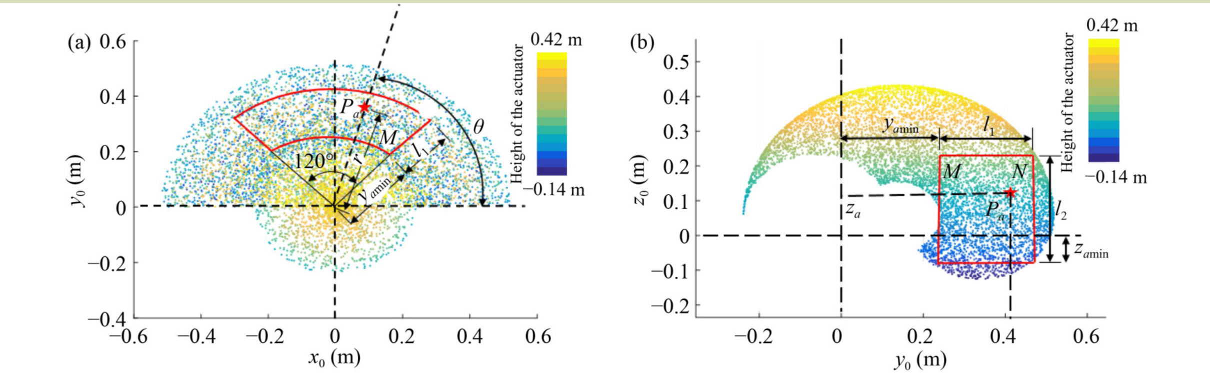


Fig. 10 Working region of the end actuator. (a) Top view; (b) cross section via the waist joint revolute pair (x_0 = 0).

$$\begin{cases} x_l \rightarrow \theta \\ y_l \rightarrow z_a \\ z_l \rightarrow -r \end{cases} \tag{27}$$

During the coordinate parameter transformation, the LMC coordinates were normalized at first. Then, these normalized coordinates were transformed into the combined coordinates of the actuator working space as a second step. The transforming algorithm can be expressed as:

$$\begin{cases} \theta = \theta_{\text{end}} - \dfrac{x_{l\text{end}} - x_l}{x_{l\text{end}} - x_{l\text{start}}}(\theta_{\text{end}} - \theta_{\text{start}}) \\ r = l_1 + y_{amin} - \dfrac{z_{l\text{end}} + z_l}{z_{l\text{end}} - z_{l\text{start}}} l_1 \\ z_a = l_2 + z_{amin} - \dfrac{y_{l\text{end}} - y_l}{y_{l\text{end}} - y_{l\text{start}}} l_2 \end{cases} \tag{28}$$

where, the subscript “start” is the starting position, the subscript “end” is the ending position (e.g., when the polar angle of robotic arm (θ) is the rotary angle of waist joint, θ_{start} is the starting position which is 150° and θ_{end} is the ending position which is 30°, then θ varies in a range of [0°, 120°]); l_1 is the polar radius of robotic arm as y_{amin} is its minimum length; and l_2 is the length of robotic arm under Z-axis as z_{amin} is its minimum length.

2.8.2 Human-machine collaborated picking mode

To simplify the interaction algorithm between the human and the manipulator, the picking algorithm of the robot was designed via targeting the fruit manually and picking the fruit with one click. Three fixed modes were set for the manipulator, which were zero mode, picking ready mode and fruit dropping mode. Under zero mode, the arm of the manipulator was straight upwards, while all the joint angles were 0°. This was the mode for the communication test of the robot and standby rest. Under picking ready mode, the human-machine collaborated picking mode would be switched on. The picking actuator would be initialed to the position which corresponding to the center of the Leap Motion coordinate system. The manipulator would be able to be controlled by the

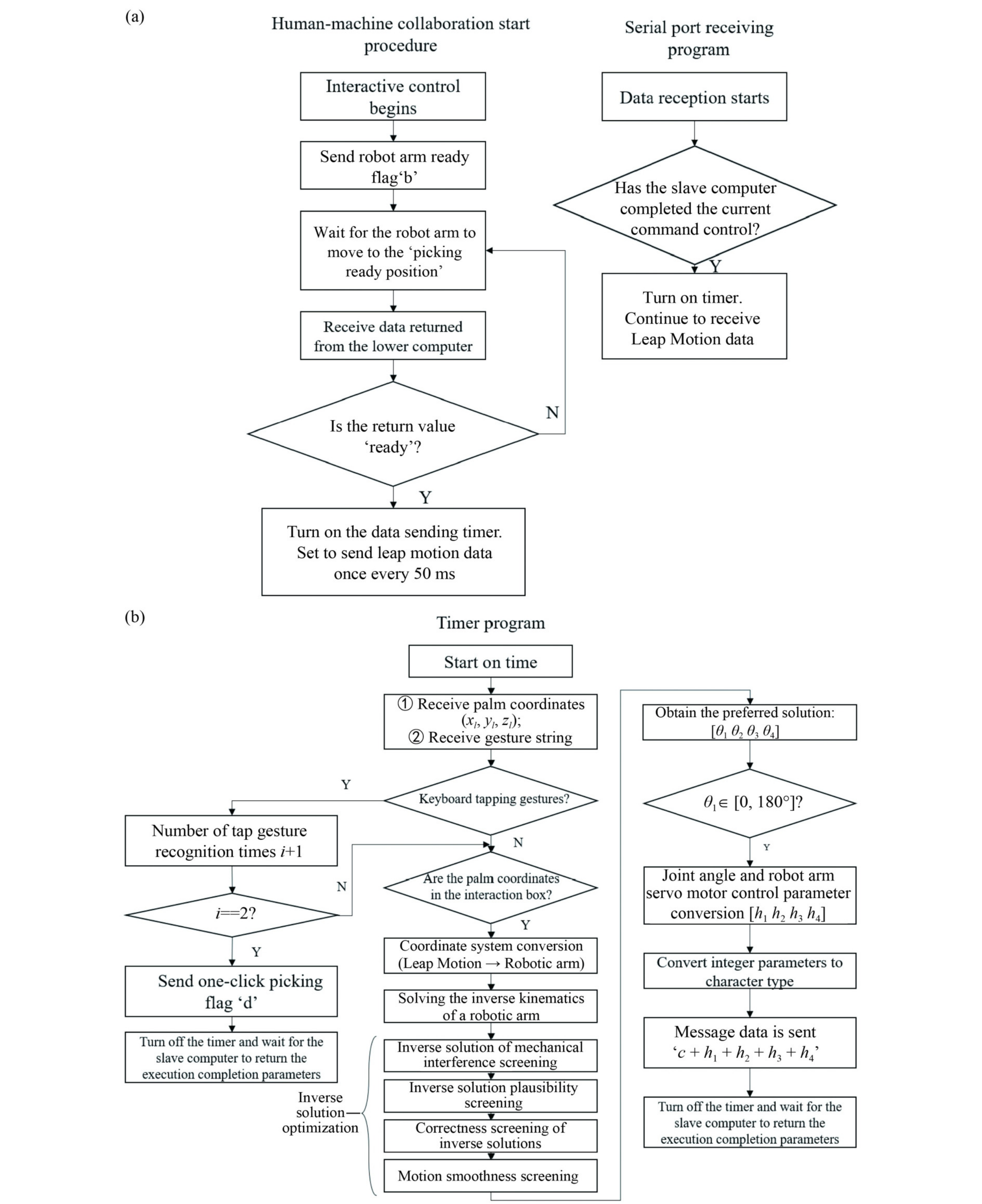

(a)
Human-machine collaboration start procedure
Interactive control begins
Send robot arm ready flag‘b’
Wait for the robot arm to move to the ‘picking ready position’
Receive data returned from the lower computer
Is the return value ‘ready’?
N
Y
Turn on the data sending timer. Set to send leap motion data once every 50 ms
Serial port receiving program
Data reception starts
Has the slave computer completed the current command control?
Y
Turn on timer. Continue to receive Leap Motion data
(b)
Timer program
Start on time
① Receive palm coordinates (x_l, y_l, z_l); ② Receive gesture string
Keyboard tapping gestures?
Y
N
Number of tap gesture recognition times i+1
i==2?
N
Y
Send one-click picking flag ‘d’
Turn off the timer and wait for the slave computer to return the execution completion parameters
Are the palm coordinates in the interaction box?
Y
Coordinate system conversion (Leap Motion → Robotic arm)
Solving the inverse kinematics of a robotic arm
Inverse solution optimization
Inverse solution of mechanical interference screening
Inverse solution plausibility screening
Correctness screening of inverse solutions
Motion smoothness screening
Obtain the preferred solution: $[\theta_1\ \theta_2\ \theta_3\ \theta_4]$
$\theta_1 \in$ [0, 180°]?
Y
Joint angle and robot arm servo motor control parameter conversion $[h_1\ h_2\ h_3\ h_4]$
Convert integer parameters to character type
Message data is sent ‘$c + h_1 + h_2 + h_3 + h_4$’
Turn off the timer and wait for the slave computer to return the execution completion parameters

Fig. 11 Control flow chart of host computer. (a) Main program and serial port receiving program; (b) timer program.

Leap Motion controller. Under fruit dropping mode, the picked fruit would be dropped into the assigned basket.

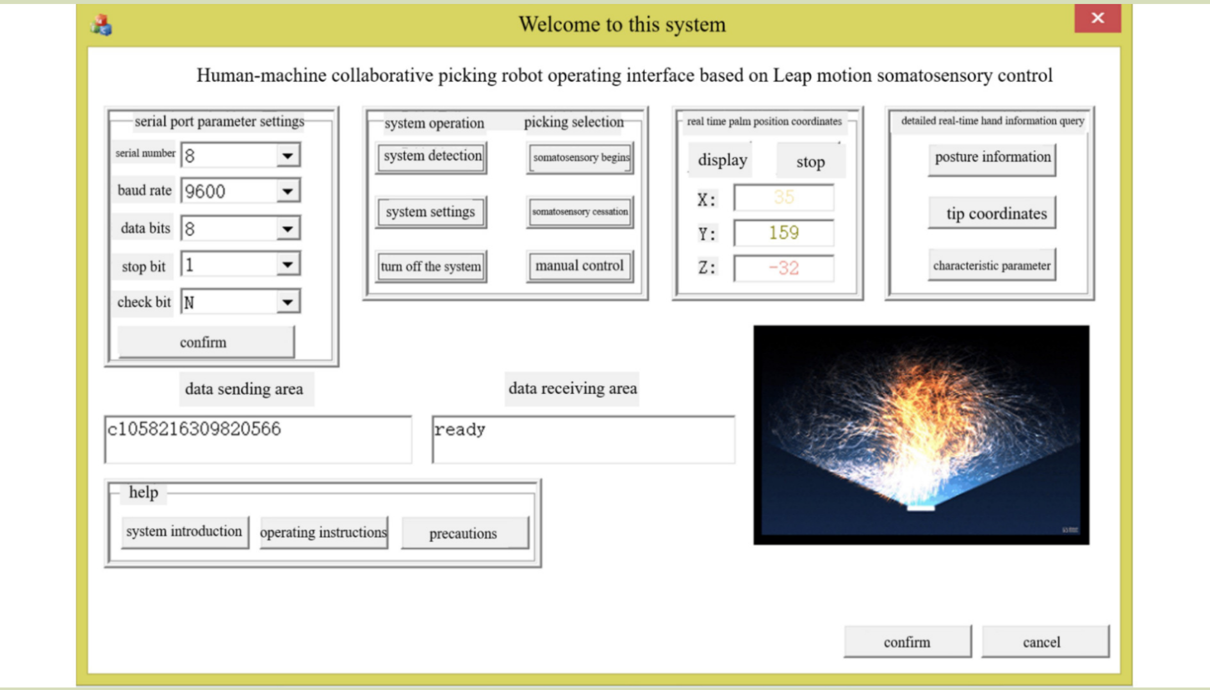
Figure 11 shows the control flow of the host computer human-machine collaborated picking mode. When the “Switch On” button of collaborating control was clicked, the manipulator would be set to the picking ready mode. After the initialization was finished, the Leap Motion would start to collect data regularly, including the palm center coordinate and the gestures. The data were processed using a timer program. The palm center coordinates would be used to calculate the rotary angles of each joint via the coordinate transform, inverse kinematics solution and inverse kinematics optimization. To ensure that the manipulator moved inside the working region in Fig. 10, the inverse kinematics solution of waist joint angle between 0° and 180° was selected. Then, the rotary angle was translated to the control parameter for each servo motor. The control parameters were sent to the lower computer as string messages, with the initial letter “c” representing motion interaction controlling mode. The timer would be switched off when the message was sent. The lower computer would send feedback to the host computer when the control operation was finished. Then the host computer would start the timer and run the data collection, processing and sending operation again. Thus, the system could realize the real-time tracking of the hand.

The gesture judgment algorithm was also programmed in the timer. A double-click gesture was assigned as the operation signal. This could reduce the misjudgment ratio than using the single-click gesture. When a correct gesture was identified, a function sign “d” of one-button harvest would be sent to the lower computer. Then, the lower computer would control the end actuator to cut the stem, rotate the fruit, unload the fruit into the basket and reposition automatically. An operation-finished command would be sent to the host computer as the end of the control flow.

2.8.3 Human-machine collaboration interface

Human-machine collaboration interfaces were designed using the MFC package in Visual Studio 2017. The main interface (Fig. 12) complied five functions including, the parameter setting for data communication, the system operation, the harvest mode selection, the real-time monitoring of gesture parameters and the display of communication data. In addition, several operation and observation functions were designed under modeless dialog windows (Fig. 13).

Parameters of communication ports between host and lower computers can be set on the left side of the interface. The



Human-machine collaborative picking robot operating interface based on Leap motion somatosensory control

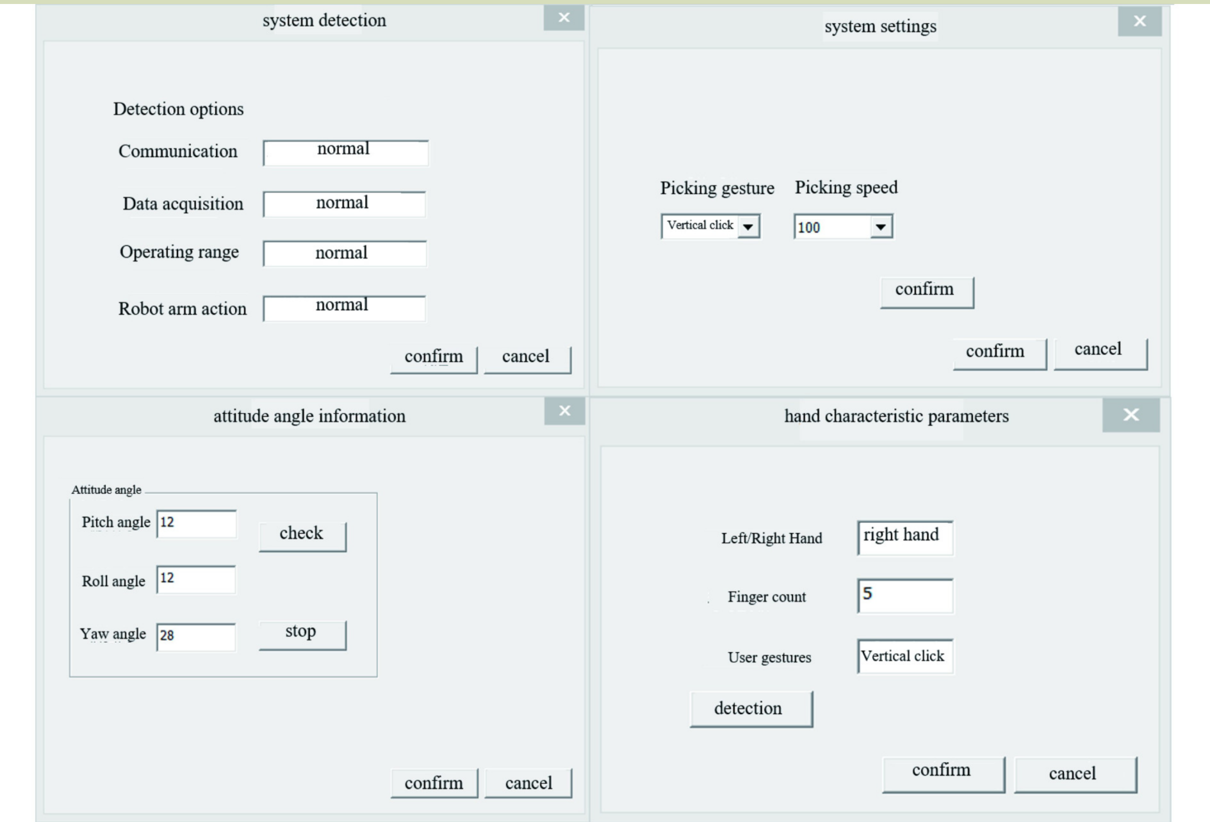
serial number 8
baud rate 9600
data bits 8
stop bit 1
check bit N

X: 35
Y: 159
Z: -32

data sending area
c1058216309820566

data receiving area
ready

Fig. 12 Main interface of human-computer interaction.



system detection

Detection options

Communication: normal

Data acquisition: normal

Operating range: normal

Robot arm action: normal

system settings

Picking gesture: Vertical click

Picking speed: 100

attitude angle information

Attitude angle

Pitch angle: 12

Roll angle: 12

Yaw angle: 28

hand characteristic parameters

Left/Right Hand: right hand

Finger count: 5

User gestures: Vertical click

Fig. 13 Modeless dialog windows for operations and observations.

system operation included two functions of system self-check and system setting. Both were programmed under modeless dialog windows as shown in Fig. 13. The area of hand parameter real-time monitoring displays information of the palm center coordinates, the hand stance, the left/right hand identification, the finger number, the gesture and the fingertip coordinates during the operation process. The controlling messages are displayed at the areas of data sending and data receiving.

The system self-check window was used to check the motion status of each unit of the robot which integrated the functions of communication module checking, Leap Motion data collection module checking, operation range checking, manipulator motion status checking. The system setting windows enables operators manually setting the harvest gesture and the moving speed of the manipulator.

3 Operational test

To validate and quantify the working efficiency of the harvesting robot, a laboratory -based harvesting experiment of the prototype robot was conducted in September 2021 at South-west University, Chongqing, China. The actual prototype and test scene are shown in Fig. 14. The prototype robot contained four parts including, the host computer, the Leap Motion sensor, the manipulator and the control box of the manipulator. In the control box, it contained the lower computer for manipulator control, the TTL bus module, the field effect transistor module, the communication module, and the power conversion module. A citrus tree was used as the operation target and was put inside the motion range of the picking actuator. Figure 15 presented the picking procedure of the robot during the test.

As picking and unloading were done automatically when the robot received the double-click command, only response time of manipulator was recorded, which included data receiving, processing, sending in the host computer and operation control processing in the lower computer. Additionally, the response time of double-click command and the time consumption of manipulator movement were also recorded. 20 fruit samples on the citrus tree were picked during the experiment. The results were listed in Table 4.

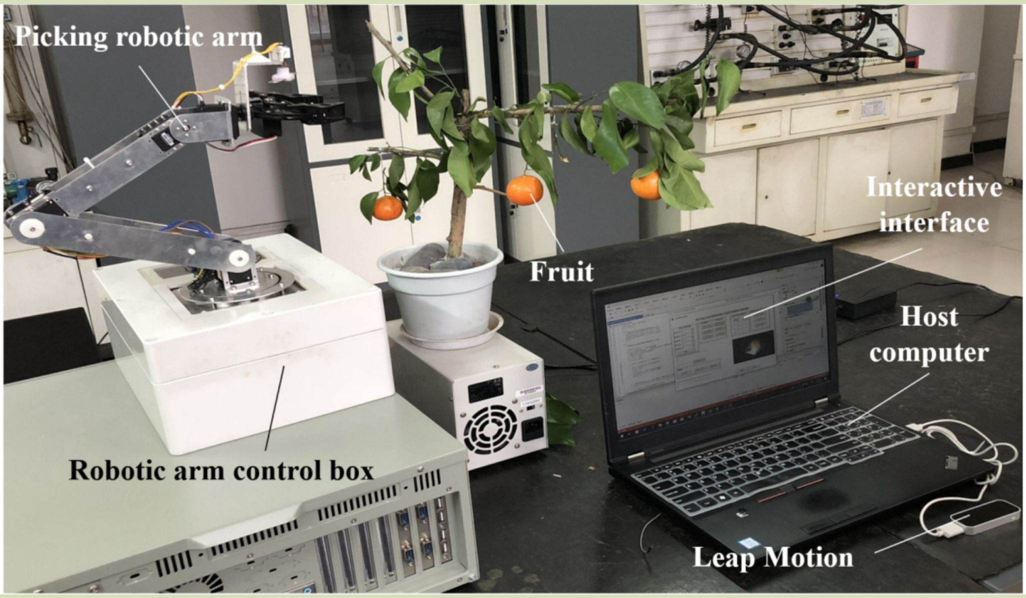


Fig. 14 Picking robot prototype and experimental scenarios.

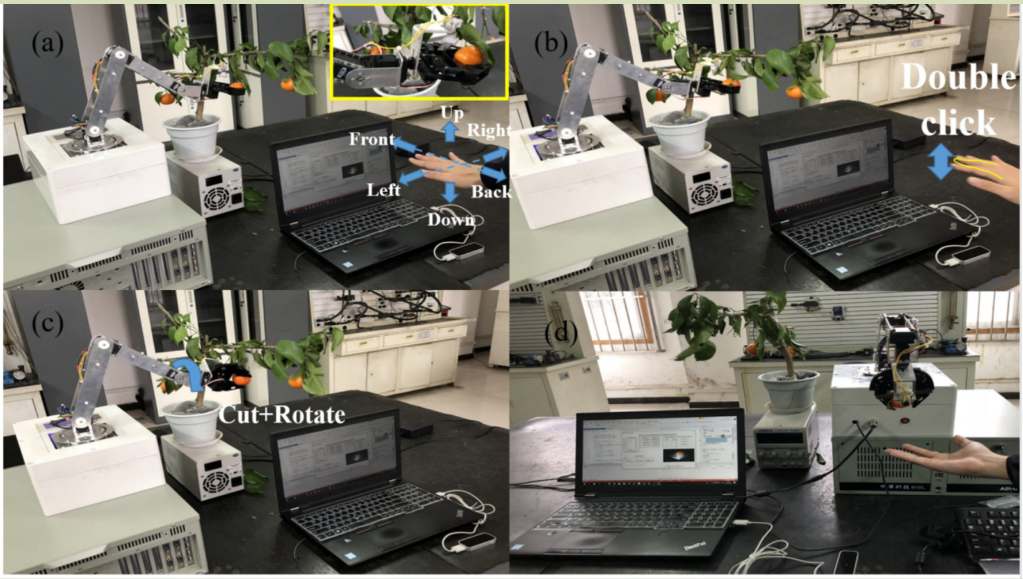


Fig. 15 Picking process of the robot. (a) Toward-target by motion-sensing; (b) picking by double click gesture; (c) automatic picking; (d) automatic unloading.

The average response time of the picking manipulator was 74.4 ms. This was sufficient for real-time control, as the one-click picking operation was actually initiated with a double-click gesture which would take the average duration of 332 ms and this prototype robot took an average duration of 8.3 s for the manipulator to finish the targeting and picking operation using the motion sensing technology. This duration closely corresponded to the operator’s angle of view, operation position, proficiency and the size of the robotic arm as well as maximum moving speed. The operation test was repeated twice to pick another 40 fruits with 20 in each test group. The average picking duration achieved was 7.1 and 6.5 s,

Table 4 Response time of the manipulator, double-click command and manipulator movement

No.	Average response time of manipulator (ms)	Response time of double-click command (ms)	Time consumption of manipulator movement (s)
1	78.4	347	8.4
2	75.6	312	8.7
3	69.8	319	6.4
4	72.3	355	5.3
5	76.4	322	12.2
6	65.7	326	9.1
7	67.2	319	6.8
8	74.1	340	10.9
9	74.5	338	6.1
10	73.6	342	10.6
11	75.8	348	7.9
12	71.4	367	7.0
13	80.2	314	6.2
14	75.2	328	8.2
15	76.7	325	7.1
16	74.4	314	11.4
17	75.6	325	9.7
18	74.3	316	6.9
19	78.2	349	8.3
20	78.3	337	8.7
Average	74.4	332	8.3

respectively. Therefore, it was clear that the operators could be trained to be more familiar with the robot and finish the operation more quickly. The average accuracy of the 60 fruits picking was 96.7%.

4 Conclusions

In this article has presented the design and control algorithm of a prototype motion sensing-based fruit harvesting robot. The system was tested on a citrus tree with the following conclusions.

(1) A gesture-based human-robot-collaborated picking system is proposed. This realized the wide motion space of the manipulator mapping from the tight motion space of the hands. Meanwhile, with the use of the one-click picking mode, the working intensity of the fruit-picking operation was significantly improved.

(2) The inverse solution of the robotic arm was optimized with mechanical interference judgment, correctness judgment, rationality judgment and motion smoothness judgment. Verification results using MATLAB Robotic Toolbox show that the optimized inverse solution was right.

(3) The average response duration is 74.4 ms. Picking operation duration of the robot can be reduce from 8.3 to 6.5 s after a short period of operator training.

Acknowledgements

This research was supported by the Key R&D Projects in the Artificial Intelligence Pilot Area of Chongqing, China (cstc2021jscx-gksbX0067), the Fundamental Research Funds for the Central Universities (SWU-KT22024), and the Local Financial of National Agricultural Science & Technology Center, Chengdu (NASC2021KR02).

Compliance with ethics guidelines

Ziwen Chen, Yuhang Chen, Hui Li, and Pei Wang declare that they have no conflicts of interest or financial conflicts to disclose. This article does not contain any studies with human or animal subjects performed by any of the authors.

REFERENCES

1. Liu J. Research progress analysis of robotic harvesting technologies in green house. *Transactions of the Chinese Society for Agricultural Machinery*, 2017, **48**(12): 1–18 (in Chinese)
2. Song J, Zhang T, Xu L, Tang X. Research actuality and prospect of picking robot for fruits and vegetables. *Transactions of the Chinese Society for Agricultural Machinery*, 2006, **37**(5): 158–162 (in Chinese)
3. Hayashi S, Shigematsu K, Yamamoto S, Kobayashi K, Kohno Y, Kamata J, Kurita M. Evaluation of a strawberry-harvesting robot in a field test. *Biosystems Engineering*, 2010, **105**(2): 160–171
4. Xu L, Zhang T. Present situation of fruit and vegetable harvesting robot and its key problems and measures in application. *Transactions of the Chinese Society of Agricultural Engineering*, 2004, **20**(5): 38–42 (in Chinese)
5. Fang J. Present situation and development of mobile harvesting robot. *Transactions of the Chinese Society of Agricultural Engineering*, 2004, **20**(2): 273–278 (in Chinese)
6. Si Y, Qiao J, Liu G, Liu Z, Gao D. Recognition and shape features extraction of apples based on machine vision. *Transactions of the Chinese Society for Agricultural Machinery*, 2009, **40**(8): 161–165 (in Chinese)
7. Feng J, Wang S, Liu G, Zeng L. A separating method of adjacent apples based on machine vision and chain code information. In: Li D, Chen Y, eds. Computer and Computing Technologies in Agriculture V. CCTA 2011. IFIP Advances in Information and Communication Technology, vol 368. Berlin, Heidelberg: *Springer*, 2012, 258–267
8. Miao Z, Shen Y, Wang X, Zhou X, Liu C. Imaged recognition algorithm and experiment of overlapped fruits in natural environment. *Transactions of the Chinese Society for Agricultural Machinery*, 2016, **47**(6): 21–26 (in Chinese)
9. Van Henten E J, Schenk E J, van Willigenburg L G, Meuleman J, Barreiro P. Collision-free inverse kinematics of the redundant seven-link manipulator used in a cucumber picking robot. *Biosystems Engineering*, 2010, **106**(2): 112–124
10. Bac C W, Roorga T, Reshef R, Berman S, Hemming J, van Henten E J. Analysis of a motion planning problem for sweet pepper harvesting in a dense obstacle environment. *Biosystems Engineering*, 2016, **146**: 85–97
11. Barth R, Hemming J, van Henten E J. Design of eye-in-hand sensing and servo control framework for harvesting robotics in dense vegetation. *Biosystems Engineering*, 2016, **146**: 71–84
12. Tu J, Liu C, Li Y, Zhou J, Yuan J. Apple recognition method based on illumination invariant graph. *Transactions of the Chinese Society of Agricultural Engineering*, 2010, **26**(S2): 26–31 (in Chinese)
13. Song H, Qu W, Wang D, Yu X, He D. Shadow removal method of apples based on illumination invariant image. *Transactions of the Chinese Society of Agricultural Engineering*, 2014, **30**(24): 168–176 (in Chinese)
14. Huang L W, He D J. Apple recognition in natural tree canopy based on fuzzy 2-partition entropy. *International Journal of Digital Content Technology and Its Applications*, 2013, **7**(1): 107–115
15. Wang D, Xu Y, Song H, He D, Zhang H. Fusion of *K*-means and Ncut algorithm to realize segmentation and reconstruction of two overlapped apples without blocking by branches and leaves. *Transactions of the Chinese Society of Agricultural Engineering*, 2015, **31**(10): 227–234 (in Chinese)
16. Wang D, Song H, He D. Research advance on vision system of apple picking robot. *Transactions of the Chinese Society of Agricultural Engineering*, 2017, **33**(10): 59–69 (in Chinese)
17. Li X. The Design and Implementation of Virtual Human Body Dissection Teaching System Based on Kinect Gesture Recognition. Beijing: *Beijing University of Technology*, 2014 (in Chinese)
18. Lu H, Zhang Y, Niu M, Shui W, Zhou M. Three-dimensional virtual reassembling for broken artifact fragments based on Leap Motion. *Journal of System Simulation*, 2015, **27**(12): 3006–3011 (in Chinese)
19. Liu C, Li Q. Leap Motion somatosensory controller and its application in aircraft structure display system. *Computer Applications and Software*, 2016, **33**(4): 227–229 (in Chinese)
20. Liu J, Jia S, Wang P, Huo D. Study of 3D product display technology based on Leap Motion. *Journal of Dalian Jiaotong University*, 2016, **37**(4): 110–113 (in Chinese)
21. Wu F, Ding Y, Ding W, Xie T. Design of human computer interaction system of virtual crops based on Leap Motion. *Transactions of the Chinese Society of Agricultural Engineering*, 2016, **32**(23): 144–151 (in Chinese)
22. Xu C, Wang Q, Chen H, Mei S, Du L. Design and simulation of artificial limb picking robot based on somatosensory interaction. *Transactions of the Chinese Society of Agricultural Engineering*, 2017, **33**(S1): 49–55 (in Chinese)
23. Quan L, Li C, Feng Z, Liu J. Algorithm of works' decision for three arms robot in greenhouse based on control with motion sensing technology. *Transactions of the Chinese Society for Agricultural Machinery*, 2017, **48**(3): 14–23 (in Chinese)
24. Fu D, Xu L, Li D, Xin L. Automatic Detection and Segmentation of Stems of Potted Tomato Plant Using Kinect. In: Proceedings Volume 9159, Proceedings of the Sixth International Conference on Digital Image Processing

(ICDIP), Athens, Greece, 5–6 April 2014. *Society of Photo-Optical Instrumentation Engineers (SPIE)*, 2014, 915905

25. Pérez-Ruíz M, Slaughter D C, Fathallah F A, Gliever C J, Miller B J. Co-robotic intra-row weed control system. *Biosystems Engineering*, 2014, **126**: 45–55

26. Denavit J, Hartenberg R S. A kinematic notation for lower pair mechanisms based on matrices. *Journal of Applied Mechanics*, 1955, **22**(2): 215–221

27. Wang W. Leap Motion Human–Computer Interaction Application Development. Xi’an: *Xidian University Press*, 2015 (in Chinese)

28. Weichert F, Bachmann D, Rudak B, Fisseler D. Analysis of the accuracy and robustness of the Leap Motion Controller. *Sensors*, 2013, **13**(5): 6380–6393