

Article

Design, Modeling, and Experimental Validation of a Vision-Based Table Tennis Juggling Robot

Yunfeng Ji ¹, Bangsen Zhang ¹, Yue Mao ¹, Han Wang ¹, Xiaoyi Hu ¹ and Lingling Zhang ^{2,*}

¹ School of Health Science and Engineering, University of Shanghai for Science and Technology, Shanghai 200093, China; ji_yunfeng@usst.edu.cn (Y.J.); 223332638@st.usst.edu.cn (B.Z.); 223332622@st.usst.edu.cn (Y.M.); 2235062023@st.usst.edu.cn (H.W.); 213332746@st.usst.edu.cn (X.H.)

² Physical Education Department, Shanghai University of Finance and Economics, Shanghai 200433, China

* Correspondence: zhang.lingling@mail.shufe.edu.cn

Abstract: This paper develops a new vision-based robot customized for table tennis juggling tasks. Specifically, the robot is equipped with two industrial cameras operating as a sensing system. An image-processing algorithm is proposed that allows the robot to balance a table tennis ball while controlling its bounce height. The robot adopts a parallel structure design, and the end effector employs three ball joints to increase the degree of freedom (DOF) of the parallel mechanism. In addition, we design a control scheme explicitly customized for this robotic system. Extensive real-time experiments are performed to show the effectiveness of the juggling robot at different jumping heights. Furthermore, the ability to consistently maintain a fixed preset bounce height is demonstrated. These experimental results confirm the efficacy of the developed robotic system.

Keywords: juggling robot; robot design; parallel structure; inverse kinematics

MSC: 70E60



Citation: Ji, Y.; Zhang, B.; Mao, Y.; Wang, H.; Hu, X.; Zhang, L. Design, Modeling, and Experimental Validation of a Vision-Based Table Tennis Juggling Robot. *Mathematics* **2024**, *12*, 1634. <https://doi.org/10.3390/math12111634>

Academic Editor: Daniel-Ioan Curiac

Received: 22 April 2024

Revised: 21 May 2024

Accepted: 22 May 2024

Published: 23 May 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Juggling is a classic example of a human dexterity task. There are various forms of juggling, among which table tennis juggling involves hitting a table tennis ball repeatedly against gravity. While most people can obtain a few bounces off a racket with relative ease, the unpredictability of each ball's position often results in an inability to maintain consistent contact. In addition, without sufficient experience, accurately judging the trajectory of the ball becomes a challenge. Enabling robotic systems to achieve these racket acrobatics is challenging and may require superhuman dexterity. Such systems have potential applications in areas such as ambulance, ship, and drone rescue.

In recent years, many researchers have explored robotic table tennis juggling through various robot structures and control methods. Juggling tasks using robotic arms as actuators have been implemented in [1–3], where the robotic arms are used to juggle independently and cooperatively. Specifically, an ideal model with a physical model equipped with an unscented Kalman filter has been proposed in [1], which is adopted to predict collision velocities well and to catch balls thrown by humans. In [2], a nonlinear least squares method has been designed to calculate the racket configuration for the next in-line shot to juggle between two paddles/hands. Subsequently, the optimal trajectory of the racket has been developed in SE(3). In [3], a method is proposed to predict the ball's velocity only in the z-direction, where the ball's position is always tracked in the x–y plane. Unlike methods that predict the overall trajectory, this method requires knowing fewer parameters to juggle a priori. In [4,5], quadcopters have been employed to hit balls. For this type of robot, the difficulty is that it is an under-actuated and nonlinear dynamic model and the control algorithm design is challenging. Therefore, there may be insufficient juggling times and poor juggling effects.

Inspired by the racket–ball system, the Cassie system has been adopted as a research platform to achieve juggling movements while maintaining Cassie’s balance in [6]. In addition, the stability of two controllers based on optimization and PD has been proven through Poincaré analysis. In [7], a blind juggling robot has been introduced, where vision, pressure, or other sensors are not required to achieve juggling. These robots mainly rely on the accuracy of the robot model. In [8,9], a juggling robot has been developed, which is capable of vertically bouncing a completely unconstrained ball without any sensors. The robot is powered by linear motors and features machined aluminum paddles. The curvature of the parabolic paddle prevents the ball from falling while stabilizing the ball’s peak height by slowing the paddle upon impact. Based on this form of “blind” juggling robot, the pendulum juggling robot has been designed in [10,11], where the paddles are mounted on a swinging pendulum to juggle balls. The pendulum consists of a four–bar mechanism, and, during juggling, the unconstrained ball can move horizontally up to 1 m, with a peak height of 1.1 m between impacts. The pendulum and paddle are used for robot control, which essentially form a dynamic coupling between the moving mass at the tip of the pendulum and the ball. Furthermore, optimal control is applied to compute the paddle motion that synchronizes with the ball, and feedback introduces a lookup table that maps the measured state to an appropriate paddle motion.

Despite these advances, several challenging questions remain outstanding and deserve deeper investigation. For relatively complex systems, such as the aforementioned quadcopters and bipedal robot systems, ensuring self–control stability is a difficult task, even without considering performing juggling actions. In addition, for relatively simple systems, such as blind juggling robot systems, sensors are not used as feedback to achieve better control performance. Inspired by the above observations, we develop a simple and stable juggling robot system. In this paper, a new vision–based robot customized for table tennis juggling tasks is presented. Specifically, the robot is equipped with two industrial cameras operating at 250 frames per second (FPS) as the sensing system. An image–processing algorithm is proposed that allows the robot to balance a table tennis ball while controlling the bounce height. The robot adopts a parallel structure design, and the end effector uses three ball joints to increase the DOF of the parallel mechanism. Accordingly, we propose a control scheme for this robotic system. We perform extensive real–time experiments to show the effectiveness of the juggling robot with varying bounce heights.

2. Table Tennis Juggling Robot Design

This section introduces the table tennis juggling robot system, as shown in Figure 1. The entire system consists of three main modules: the vision module, the execution module, and the control module. In particular, the vision module is a fixed binocular vision system, where two cameras are fixed directly above the robot to detect the movement trajectory of the table tennis ball. The execution module is a parallel robot, which consists of a base, a platform, and three identical parallel mechanisms. The base contains the overall framework of the robot. The juggling platform is a regular hexagon located directly above the base, and its center is vertically aligned with the center point of the base. Each parallel linkage consists of three joints, with the end joint being the ball joint and the others being rotational joints. The entire system is controlled by a computer. The cameras are connected to this computer as the sensors for the program. The computer processes input from the cameras to obtain feedback information and then sends control commands to the robot via an Ethernet–to–CAN module. In order to process image data and control robotic movements concurrently, we choose the following PC configuration: a CPU of 12th Gen Intel(R) Core(TM) i5–12400 2.50 GHz, 15.6 GB of memory, and an NVIDIA TITAN V GPU. Furthermore, our software solution is made using the C# programming language ([https://en.wikipedia.org/wiki/C_Sharp_\(programming_language\)](https://en.wikipedia.org/wiki/C_Sharp_(programming_language))) accessed on 1 January 2024) and compiled in a .NET 7.0 environment.

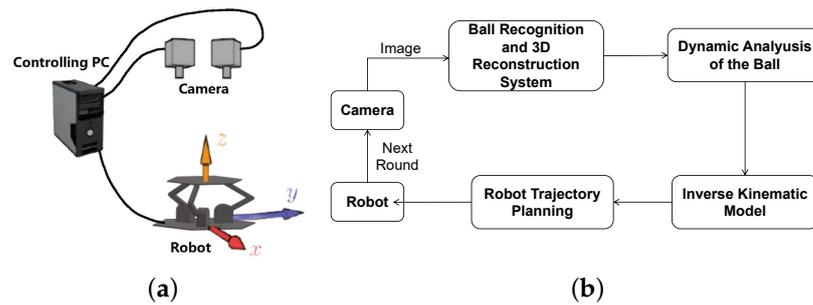


Figure 1. (a) The vision-based table tennis juggling robot. (b) Structure of the table tennis robot juggling system.

3. Image Processing

The purpose of the vision system is to collect real-time moving images of table tennis balls. To achieve stable juggling, the robot needs to complete the identification, spatial localization, and trajectory prediction of table tennis balls through the vision system. Among these tasks, the spatial localization of table tennis balls is fundamental. Since the entire robot system requires high real-time performance, an increase in the camera frame rate within the 60–300 FPS range positively impacts the consistency of table tennis trajectory recognition due to the corresponding increase in image data. However, when the frame rate exceeds 250 FPS, the enhancement in consistency becomes negligible; instead, it further affects the recognition speed. After conducting comparative experiments, we choose to use two high-speed black-and-white cameras with a resolution of 1440×1080 and a frame rate of 250 FPS to establish a binocular vision system. In addition, to obtain the accurate parameters of the table tennis ball in the Cartesian coordinate system, we correct the distortion caused by the camera and establish the relationship between the camera pixel coordinates and the three-dimensional (3D) coordinates. The binocular camera is calibrated according to the established calibration model [12]. Table tennis detection can be achieved after calibration.

In the experiment, we use white table tennis balls. Initially, the entire image is converted to grayscale, and a specific threshold is chosen to binarize the image. Background subtraction is also employed to remove interference from the image. The images, both pre- and post-processing, are depicted in Figure 2. Furthermore, appropriate circularity and contour area are selected to obtain the circular contour of the table tennis ball, and the centroid of the minimum enclosing circle of the contour is calculated as the coordinates of the table tennis ball's center in the image coordinate system. We implement a dynamic region of interest (ROI) algorithm to reduce the image size. This approach significantly decreases the processing time per image from 20 ms to 2 ms.

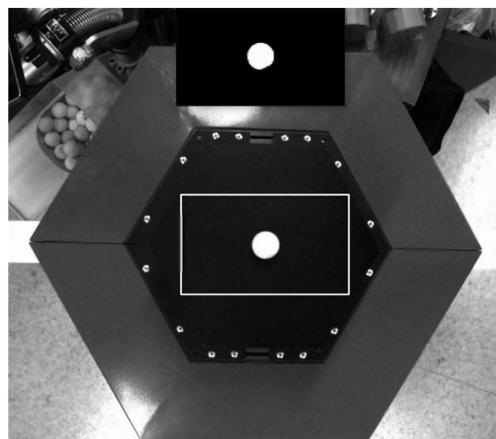


Figure 2. Image-processing results.

In a binocular camera system, 3D positioning of table tennis balls is achieved. The entire 3D positioning model is shown in Figure 3, where the image coordinates of the table tennis ball are (u_l, v_l) in camera 1 and (u_r, v_r) in camera 2. Due to the epipolar constraint, the X_c -axis coordinates of the same target in the left and right images are equal, i.e., $u_l = u_r$. The coordinate difference between the table tennis ball in camera 1 and camera 2 can be expressed as

$$d = v_l - v_r. \tag{1}$$

Since the baseline length γ and focal length f between these two cameras are known, the following equation can be obtained by the principle of similar triangles:

$$\frac{\gamma}{z_c} = \frac{(\gamma + v_r) - v_l}{z_c - f} \tag{2}$$

where z_c is the coordinate along the Z_c -axis direction. Then, the table tennis ball's coordinates in the camera 1 coordinate system can be derived as

$$z_c = \frac{\gamma f}{v_l - v_r} = \frac{\gamma f}{d}. \tag{3}$$

Similarly, the table tennis ball's coordinates in the camera 1 coordinate system can be obtained as

$$x_c = \frac{(u_l - u_o)z_c}{f} \text{ and } y_c = \frac{(v_l - v_o)z_c}{f} \tag{4}$$

where x_c and y_c are the coordinates along the X_c -axis direction and Y_c -axis direction, respectively, and (u_o, v_o) is the coordinate of the image center in the image coordinate system.

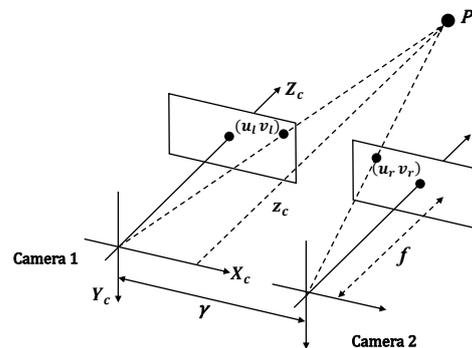


Figure 3. Binocular 3D positioning principle.

Next, the table tennis ball coordinates in the camera 1 coordinate system are transformed into the world coordinate system, where the Perspective–N–Point (PNP) method presented in [13] is applied. Finally, the corrected 3D coordinates of the table tennis ball in the world coordinate system can be obtained by the homogeneous transformation.

4. Inverse Kinematics

For parallel robots, inverse kinematics calculate the angle or displacement of each joint based on the desired position, orientation, and kinematic parameters of the robot's end effector. That is, inverse kinematics involves deriving the state of each joint from the end effector of the robot. The solution to the inverse kinematics problem is crucial for the control of parallel robots because it enables the robot system to achieve the required precise motion and operation.

The 3D model and mathematical model of the robot are illustrated in Figure 4a and 4b, respectively. The table tennis juggling robot is computed based on the world coordinate system O–XYZ. Points A_1 , A_2 , and A_3 are located on the output shafts of three motors, which are all rotating joints and form an equilateral triangle. The origin O of the O–XYZ

coordinate system is at the center of this equilateral triangle, point A_2 is on the Y-axis, and the Z-axis is perpendicular to this equilateral triangle. Points $B_1, B_2,$ and B_3 are rotating joints between the active linkage and the passive linkage. The plane formed by points $C_1, C_2,$ and C_3 is the plane formed by the three ball joints connected to the juggling paddle below the linkages. These three points are ball joints, forming another equilateral triangle, with point P located at the center of this equilateral triangle.

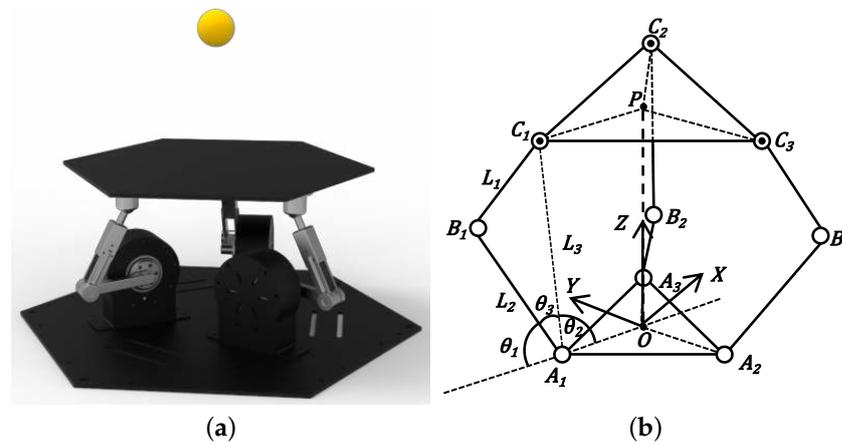


Figure 4. (a) 3D model of the table tennis juggling robot. (b) Kinematic diagram of the table tennis juggling robot.

According to the above robot model, the process of solving inverse kinematics is as follows: in the world coordinate system $O-XYZ$, the coordinates of the three points $A_1, A_2,$ and A_3 on the three output shafts are constant, and the initial state of the robot is specified. In the initial state, the coordinates of $C_1, C_2,$ and C_3 relative to the world coordinate system are also constant and known a priori. Therefore, taking one of the parallel arms as an example, we obtain

$$\vec{OA_1} + \vec{A_1C_1} + \vec{C_1P} = \vec{OP} \tag{5}$$

where $\vec{\alpha\beta}$ denotes the vector formed between points α and β . In the inverse kinematics, the desired configuration is obtained. Thus, the homogeneous transformation matrix \mathbf{M} of the juggling paddle is

$$\mathbf{M} = \begin{bmatrix} \mathbf{R} & \mathbf{Q} \\ \mathbf{0} & 1 \end{bmatrix} \tag{6}$$

where \mathbf{R} and \mathbf{Q} are the rotation matrix and translation matrix relative to the world coordinate system, respectively, and $\mathbf{0}$ is a three-dimensional row vector with all elements equal to 0.

During each juggling process, the posture of the juggling paddle and the positions of the three ball joints constantly change. Therefore, the change in ball joint position during juggling can be described as

$$\vec{OC_1^*} = \vec{OC_1} \times \mathbf{M} \tag{7}$$

$$\vec{OP^*} = \vec{OP} \times \mathbf{M} \tag{8}$$

where $\vec{OC_1}$ represents the vector of the ball joint in the initial state and $\vec{OC_1^*}$ is the vector of the ball joint after the transformation through the pose matrix during each juggling motion; \vec{OP} and $\vec{OP^*}$ are the vectors representing the centers of the juggling paddle before and after the homogeneous transformation, respectively.

Since the coordinates of joints of the robot are known at all times, according to the mathematical model in Figure 4b, θ_2 can be determined using the cosine law:

$$\theta_2 = \arccos \frac{|\overrightarrow{OA_1}|^2 + |\overrightarrow{A_1C_1^*}|^2 - |\overrightarrow{OC_1^*}|^2}{2|\overrightarrow{OA_1}||\overrightarrow{A_1C_1^*}|} \tag{9}$$

Since the lengths of the robot linkages are constant, we calculate θ_3 using the cosine law as

$$\theta_3 = \arccos \frac{|\overrightarrow{A_1B_1}|^2 + |\overrightarrow{A_1C_1^*}|^2 - |\overrightarrow{B_1C_1^*}|^2}{2|\overrightarrow{A_1B_1}||\overrightarrow{A_1C_1^*}|} \tag{10}$$

Note that θ_2 and θ_3 theoretically have multiple solutions. According to structural restrictions, it can be easily observed that θ_2 and θ_3 are acute angles. Consequently, $\theta_2 + \theta_3$ is the result of the inverse kinematics. Then, mimicking the same argument, the angles of the other two parallel linkages can also be determined.

There are vertical distance and orientation errors between the plane formed by $C_1, C_2,$ and C_3 and the juggling paddle. Therefore, a static error equation is required to eliminate this effect. The static error matrix \mathbf{T} is

$$\mathbf{T} = \begin{bmatrix} \mathbf{r} & \mathbf{q} \\ 0 & 1 \end{bmatrix} \tag{11}$$

where \mathbf{r} is related to the position and posture of the set juggling landing position. In this experiment, the landing position is chosen to be located at the geometric center of the juggling paddle. Therefore, nine parameters of the rotation matrix are all equal to zero; \mathbf{q} is related to the distance between two planes and is chosen as $[0 \ 0 \ d_z]^T$.

5. Robot Control Method

For the control of the juggling robot, in addition to controlling the robot to perform juggling actions, it is also necessary to control the robot to balance the table tennis balls. Since the table tennis ball loses kinetic energy due to resistance and deformation during suspension and collision, it is necessary to provide energy to the table tennis ball during the control process. Thus, when controlling juggling, the robot is required to execute the command to move upward at a certain point in time. Since the juggling paddle of the robot is made of aluminum, the model of a table tennis ball bouncing off an aluminum paddle can be represented by

$$\overrightarrow{V_{out}} = \overrightarrow{K} * \overrightarrow{V_{in}} + \overrightarrow{B} \tag{12}$$

where \overrightarrow{K} is a diagonal matrix representing the velocity loss coefficient after the ball collides with the plane, \overrightarrow{B} is the velocity compensation bias after the collision, $\overrightarrow{V_{in}}$ is the incident velocity when the ball collides with the plane, and $\overrightarrow{V_{out}}$ is the exit velocity after the collision. During juggling, the direction of maximum velocity change is the Z-axis, so only the change in velocity in the Z-axis direction needs to be considered. Thus, it can be simplified to

$$\overrightarrow{V_{Z_{out}}} = k * \overrightarrow{V_{Z_{in}}} + b. \tag{13}$$

Since it is difficult to obtain the collision coefficient of a table tennis ball and an aluminum paddle, we refer to the collision and rebound model between the table tennis ball and the table tennis table. According to [14,15], the collision coefficient between the table tennis ball and the aluminum paddle is determined to be a constant. The values of the collision coefficient in this experiment are $k = -0.95$ and $b = 0$. The robot determines the movement of the juggling paddle based on this loss $\delta z = 0.037z_h$, where z_h is the fixed height set during juggling.

Mechanical errors may occur during juggling, leading to uncertainty in where the table tennis ball lands. If the robot is controlled only to perform juggling actions without relying on sensor feedback, the table tennis ball will exceed the range of the juggling paddle after bouncing a few times. Therefore, the landing position of the table tennis ball should be controlled during juggling. To achieve stable juggling, a proportional–derivative (PD) controller is applied to correct the landing position of the ball. First, the target position is set as the geometric center of the juggling paddle, and the error is defined as the distance between the real–time position of the table tennis ball and the target position. The output of the controller is the tilt angle of the juggling paddle around the X–axis and Y–axis, ensuring that the actual landing position gradually approaches the target position during each continuous juggling. Unlike balancing ball systems, the tilt angle of the juggling paddle in the juggling robot is limited, which implies that the entire output of the PD controller needs to be limited. During the entire control process, the trajectory of the robot’s motion needs to be planned to ensure that the joint velocities of the robot are continuous. In velocity curve planning, T–shaped curves and S–shaped curves are classic planning methods. In this work, we use a T–shaped curve to optimize the velocity of the joints. In general, a T–shaped curve consists of three stages: acceleration, constant speed, and deceleration. For the predetermined motion time of the robot, δz becomes a key factor in planning. Short δz will result in a short constant speed phase, while long δz will lead to a long constant speed phase. This implies that the paddle speed is related to δz and time. Throughout the juggling motion, to ensure that the robot remains mainly in the constant–speed phase, after testing, $\delta z = 0.037z_h$ is a suitable height. Additionally, due to the high real–time requirements of the juggling robot, all three joints use high–precision and high–performance servo motors. The entire control framework of the robot is shown in Figure 5.

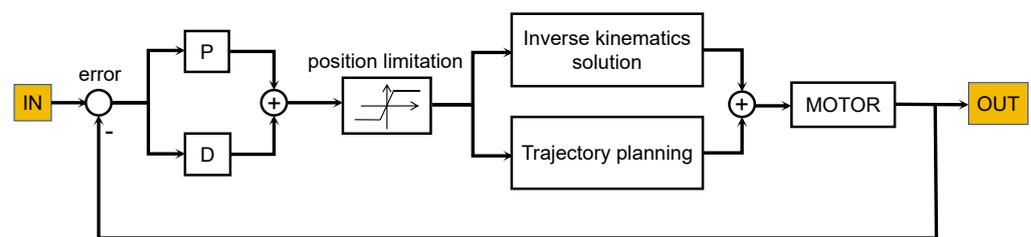


Figure 5. The control framework of the table tennis juggling robot.

The detailed algorithms employed in our robotic systems are summarized as follows. First, the stereo camera needs to be calibrated, and multiple images under different lighting conditions are statistically analyzed to obtain the color threshold of the white table tennis ball in the image; then, the image is binarized. In order to eliminate the interference in the image, a background–subtraction method is proposed to eliminate large–area pixel interference; at the same time, multiple morphological opening operations and erosion and dilation operations are used to eliminate individual pixel interference in the binarized image that has not been processed. Dynamic ROI is used for performance optimization, and the target detection time is controlled within 2ms to meet the real–time requirements. Then, the spatial position of the ball is calculated using the stereo three–dimensional positioning algorithm, and the coordinates of the ball in the camera coordinate system are converted to the world coordinate system through the PNP solution; the error in coordinate conversion is calculated through experiments and corrected through homogeneous transformation. The visual system provides the position information of the table tennis ball as a feedback signal, so the error between the actual position and the expected position is calculated. The total control output is obtained using the PD controller to adjust the posture of the robot, and the output part is the tilt angle of the baffle. The inverse motion of the robot is calculated using the geometric method, and, according to the known tilt angle of the baffle, the angle of motion required by each motor can be obtained. For the motion situation of

the robot, the T-type curve planning method is used. Combining these two algorithms can control the robot to reach a specified position with a specified posture and speed.

6. Physical Experiments

In this section, we assess the performance of the robot system through the following two experiments: (1) We verify the landing position of the table tennis ball at different heights for each juggle and compare the patterns of table tennis ball landing maps at different heights; (2) We verify the error between the preset target juggling height and the actual juggling height for each juggle.

The overall experimental environment of the robot is shown in Figure 6. First, we conduct experiments on a hexagonal juggling paddle using a standard 4 cm diameter white table tennis ball. The purpose of designing this shaped juggling paddle is to match the positions of the three parallel linkages, making it easy to identify key points in the vision system. The length of each active linkage is 7 cm, while the length of the passive linkage is 9.4 cm. Additionally, the vertical distance between the plane formed by the ball joint and the plane of the juggling paddle is 3 cm. We use servo joint motors with a rated torque of 4 Nm and a peak torque of 12 Nm. The servo motor is equipped with a gear reduction ratio of 7.75, at which the rated speed is 240 revolutions per minute (rpm). Two 250 FPS industrial cameras are mounted 1 m above the robot to ensure that the cameras' combined field of view covers the entire robot.

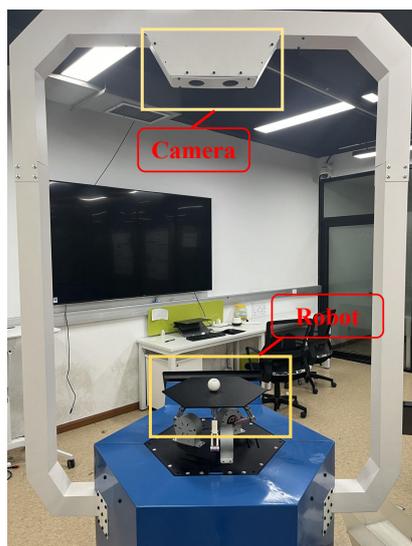


Figure 6. Robot experimental environment and equipment.

In this experiment, the target point is set at the center of the juggling paddle, and three height values of 10 cm, 30 cm, and 50 cm are set. The number of juggles is fixed at 100 for each experiment. In the robot model, when the robot is in a stationary state, the distance between the juggling paddle and the XOY plane of the world coordinate system is 18.1 cm. Thus, the entire juggling height curve also needs to be offset by 18.1 cm. The experimental results and the corresponding process are demonstrated in Figures 7 and 8, respectively. For the selection of each juggling landing point, we employ the following method. The image information captured by two high-speed cameras undergoes processing to obtain continuous and dense discrete data. The selection of landing points involves analyzing the trajectory of the table tennis ball. Within a set of continuous discrete data, we identify the local minima and consider the corresponding two-dimensional coordinates as the landing points. These results indicate that the effectiveness of juggling can be ensured by controlling the posture of the juggling paddle. However, as the set juggling height increases, the distribution of table tennis landing positions becomes more dispersed. In the three error maps, the maximum errors corresponding to height values of 10 cm, 30 cm,

and 50 cm are 250 mm², 2500 mm², and 6000 mm², respectively. It can be observed that the deviation between the average landing position and the set target position increases as the height increases.

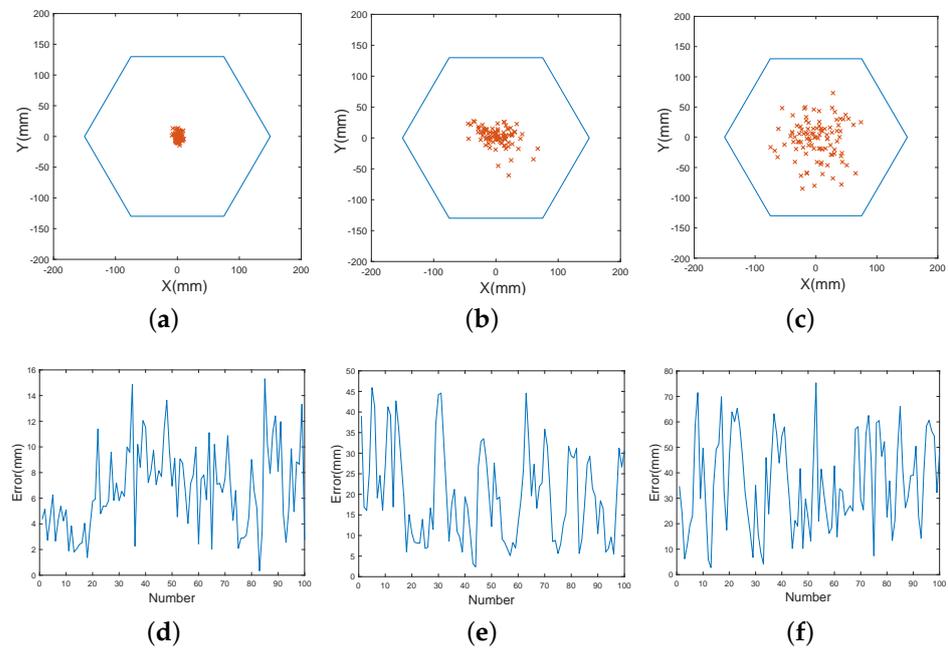


Figure 7. Landing positions and position errors at different heights. (a) The landing positions at the height of 10 cm. (b) The landing positions at the height of 30 cm. (c) The landing positions at the height of 50 cm. (d) The errors at the height of 10 cm. (e) The errors at the height of 30 cm. (f) The errors at the height of 50 cm.

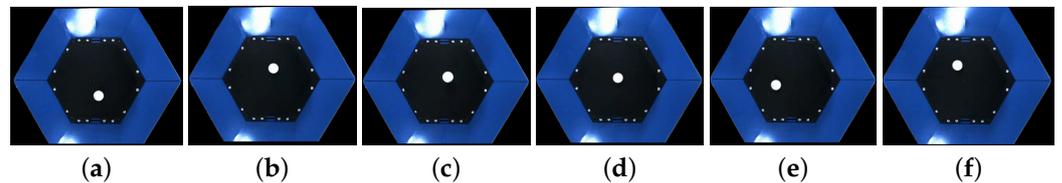


Figure 8. The six landing points recorded when the robot continuously juggles the ball. (a) Landing position 1. (b) Landing position 2. (c) Landing position 3. (d) Landing position 4. (e) Landing position 5. (f) Landing position 6.

In the second experiment, we also set the target position at the center of the juggling paddle and set three height values of 10 cm, 30 cm, and 50 cm. Each experiment lasts 30 s. The experimental results and the corresponding process are shown in Figures 9 and 10, respectively. Similarly, the selection of the maximum height also involves extraction from discrete data, where the selected local maxima represent the maximum height value. These results illustrate that the juggling robot can control each juggle within an error range of 10% from the set height while ensuring that the ball does not leave the paddle. Table tennis balls are constantly subjected to air resistance during movement and lose kinetic energy during the collision with the juggling paddle. These result in the free-falling table tennis ball eventually coming to rest. Therefore, each juggle requires the juggling paddle to exert a reverse collision impulse on the table tennis ball to compensate for its loss of energy and stabilize it near the set height value (as mentioned above, $\delta z = 0.037z_h$).

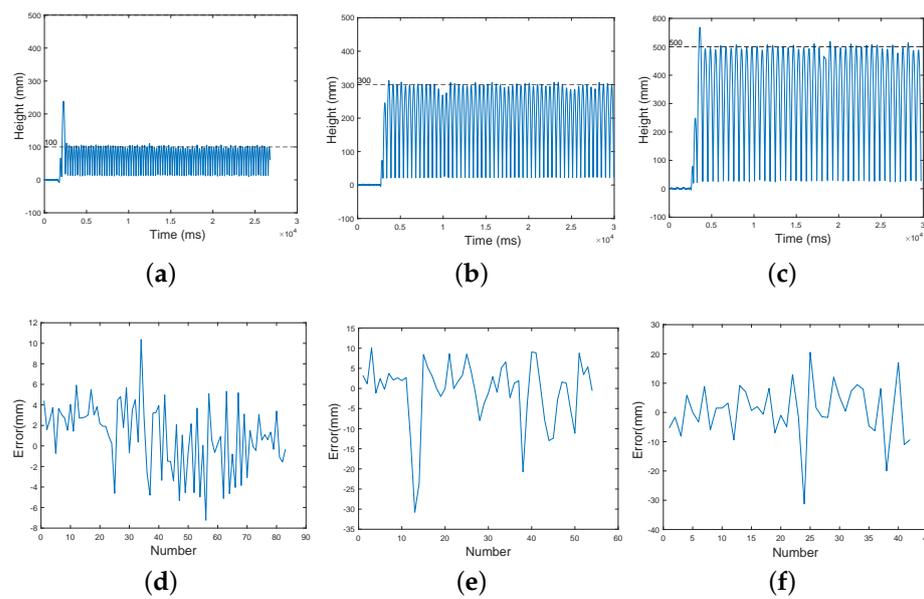


Figure 9. Height curves and height errors at different heights. (a) Height curve of juggling height at 10 cm. (b) Height curve of juggling height at 30 cm. (c) Height curve of juggling height at 50 cm. (d) The height error of juggling height at 10 cm. (e) The height error of juggling height at 30 cm. (f) The height error of juggling height at 50 cm.

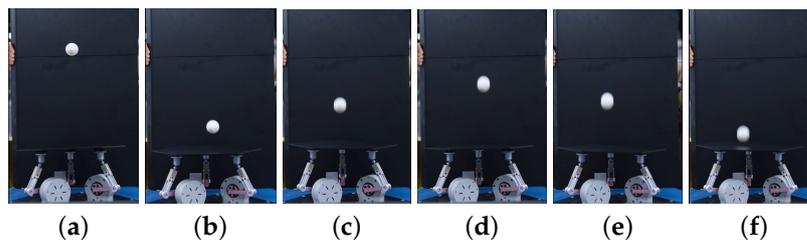


Figure 10. The height recorded during continuous ball juggling by the robot. (a) Stage 1. (b) Stage 2. (c) Stage 3. (d) Stage 4. (e) Stage 5. (f) Stage 6.

7. Conclusions

In this paper, we have developed a vision-based robot customized for table tennis juggling tasks based on a parallel structure. The hardware configuration includes a binocular vision system with two cameras, a computer, and a parallel robot. We have detailed the image-processing techniques tailored to the task requirements, used geometric methods to solve the inverse kinematics problem of the juggling robot, and introduced a continuous juggling algorithm based on PD control. To evaluate the performance of the entire system, we conducted two experiments. The results demonstrated that, when the preset height value is constant, the robot's juggling height can be regulated close to the target value, maintaining the average error within an acceptable range. However, we observed some variations in the juggling performance at different heights. As the set height increases, the average error between the juggling landing point and both the maximum height value and the set threshold also increases. This can largely be attributed to the accumulation of various errors, including mechanical errors, vision system errors, and other hardware errors. Furthermore, we have not performed an in-depth study on the physical model of table tennis, and it might be beneficial to consider integrating robot dynamics for better robot control. In our future work, we will build a physical model of the table tennis ball to predict its trajectory and develop a new collision model to ensure high robustness so that the landing position remains close to the set target position. Juggling robots are complex systems composed of multiple subsystems, and current research results are limited. These

systems are studied for more than just juggling functions; the control algorithms developed for them can also be applied in other fields, including the fields of drones, humanoid robots, and medical devices. In addition, these systems can also be applied in table tennis robot systems, laying the foundation for further research on table tennis robot systems.

Author Contributions: Methodology, Y.J.; Software, H.W.; Validation, Y.M.; Formal analysis, X.H.; Investigation, B.Z.; Writing—original draft, B.Z.; Writing—review & editing, X.H.; Supervision, L.Z.; Project administration, Y.J. and L.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data sharing is not applicable.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Rapp, H.H. A ping-pong ball catching and juggling robot: A real-time framework for vision guided acting of an industrial robot arm. In Proceedings of the 5th International Conference on Automation, Robotics and Applications, Wellington, New Zealand, 6–8 December 2011; pp. 430–435.
2. Serra, D.; Ruggiero, F.; Lippiello, V.; Siciliano, B. A nonlinear least squares approach for nonprehensile dual-hand robotic ball juggling. In Proceedings of the 20th IFAC World Congress, Toulouse, France, 9–14 July 2017; pp. 11485–11490.
3. Nakashima, A.; Sugiyama, Y.; Hayakawa, Y. Paddle juggling of one ball by robot manipulator with visual servo. In Proceedings of the 2006 9th International Conference on Control, Automation, Robotics and Vision, Singapore, 5–8 December 2006; pp. 1–6.
4. Müller, M.; Lupashin, S.; D’Andrea, R. Quadrocopter ball juggling. In Proceedings of the 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems, San Francisco, CA, USA, 25–30 September 2011; pp. 5113–5120.
5. Dong, W.; Gu, G.Y.; Ding, Y.; Zhu, X.; Ding, H. Ball juggling with an under-actuated flying robot. In Proceedings of the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, Germany, 28 September–2 October 2015; pp. 68–73.
6. Poggensee, K.L.; Li, A.H.; Sotsaichich, D.; Zhang, B.; Kotaru, P.; Mueller, M.; Sreenath, K. Ball juggling on the bipedal robot cassie. In Proceedings of the 2020 European Control Conference (ECC), Petersburg, Russia, 12–15 May 2020; pp. 875–880.
7. Ronsse, R.; Lefevre, P.; Sepulchre, R. Rhythmic feedback control of a blind planar juggler. *IEEE Trans. Robot.* **2007**, *23*, 790–802. [[CrossRef](#)]
8. Reist, P.; D’Andrea, R. Bouncing an unconstrained ball in three dimensions with a blind juggling robot. In Proceedings of the 2009 IEEE International Conference on Robotics and Automation, Kobe, Japan, 12–17 May 2009; pp. 1774–1781.
9. Reist, P.; D’Andrea, R. Design and analysis of a blind juggling robot. *IEEE Trans. Robot.* **2012**, *28*, 1228–1243. [[CrossRef](#)]
10. Reist, P.; D’Andrea, R. Design of the pendulum juggler. In Proceedings of the 2011 IEEE International Conference on Robotics and Automation (ICRA), Shanghai, China, 9–13 May 2011; pp. 5154–5159.
11. Fontana, F.; Reist, P.; D’Andrea, R. Control of a swinging juggling robot. In Proceedings of the 2013 European Control Conference (ECC), Zurich, Switzerland, 17–19 July 2013; IEEE: New York, NY, USA, 2013; pp. 2317–2322.
12. Zhang, Z. A flexible new technique for camera calibration. *IEEE Trans. Pattern Anal. Mach. Intell.* **2000**, *22*, 1330–1334. [[CrossRef](#)]
13. Li, S.; Xu, C.; Xie, M. A robust O (n) solution to the perspective-n-point problem. *IEEE Trans. Pattern Anal. Mach. Intell.* **2012**, *34*, 1444–1450. [[CrossRef](#)]
14. Zhang, Z.; Xu, D.; Tan, M. Visual measurement and prediction of ball trajectory for table tennis robot. *IEEE Trans. Instrum. Meas.* **2010**, *59*, 3195–3205. [[CrossRef](#)]
15. Ji, Y.; Mao, Y.; Suo, F.; Hu, X.; Hou, Y.; Yuan, Y. Opponent hitting behavior prediction and ball location control for a table tennis robot. *Biomimetics* **2023**, *8*, 229. [[CrossRef](#)] [[PubMed](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.