



Article

Opponent Hitting Behavior Prediction and Ball Location Control for a Table Tennis Robot

Yunfeng Ji ¹, Yue Mao ², Fangfei Suo ², Xiaoyi Hu ², Yunfeng Hou ^{1,*} and Ye Yuan ¹¹ Institute of Machine Intelligence, University of Shanghai for Science and Technology, Shanghai 200093, China² School of Health Science and Engineering, University of Shanghai for Science and Technology, Shanghai 200093, China

* Correspondence: yunfenghou@usst.edu.cn; Tel.: +86-1521-672-1916

Abstract: The hitting position and velocity control for table tennis robots have been investigated widely in the literature. However, most of the studies conducted do not consider the opponent's hitting behaviors, which may reduce hitting accuracy. This paper proposes a new table tennis robot framework that returns the ball based on the opponent's hitting behaviors. Specifically, we classify the opponent's hitting behaviors into four categories: forehand attacking, forehand rubbing, backhand attacking, and backhand rubbing. A tailor-made mechanical structure that consists of a robot arm and a two-dimensional slide rail is developed such that the robot can reach large workspaces. Additionally, a visual module is incorporated to enable the robot to capture opponent motion sequences. Based on the opponent's hitting behaviors and the predicted ball trajectory, smooth and stable control of the robot's hitting motion can be obtained by applying quintic polynomial trajectory planning. Moreover, a motion control strategy is devised for the robot to return the ball to the desired location. Extensive experimental results are presented to demonstrate the effectiveness of the proposed strategy.

Keywords: table tennis robot; visual module; hitting behavior prediction; ball location control



Citation: 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. <https://doi.org/10.3390/biomimetics8020229>

Academic Editors: Peng Li, Yantao Shen and Shan Luo

Received: 8 April 2023

Revised: 26 May 2023

Accepted: 28 May 2023

Published: 29 May 2023



Copyright: © 2023 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

Recent advances in science and technology have generated higher requirements for robot mobility, adaptability and survival ability [1–3]. To satisfy these requirements, robots are required to sense changing conditions occurring in the environment so that appropriate control decisions can be made [4–9]. Recently, a variety of bionic robots have been developed to complete various tasks, such as high-speed flexible movement [10], jumping [11], swimming [12], walking [13], etc. However, how to effectively react to changing environments remains a challenging task. For example, in table tennis robot systems, how can we accurately predict the trajectory of a spinning ball? One common approach is to measure the rotations of the ball via a high-speed camera by setting a mark on it [14,15]. However, these methods are conservative since they put a high requirement on the performances of the camera and the measuring algorithms. It is common sense that the players predict the rotation types of the ball according to the opponent's hitting behaviors. Inspired by this, this paper develops techniques to map the opponent's hitting behaviors to the rotation types of the ball.

Robot-arm-based table tennis robots have been studied for more than 40 years [16,17]. The authors of [18] developed a table tennis robot system that consists of a 7-degrees-of-freedom (DoF) industrial robot arm DARM-2 and two linear cameras. This robot has successfully completed two to three rounds against the wall. By using a 6-DoF PUMA260 industrial robot arm and four high-speed cameras, the table tennis robot [19] has achieved man-machine matchmaking for the first time. With the rapid development of hardware and software technology, the successful rate of returning the ball for the table tennis robot [20] has attained 58%. In [21], the table tennis robot can play with humans for up to 50 rounds. Additionally, the authors of [22] developed a self-designed lightweight robot arm, which

can achieve quick and flexible strokes. In [23], a humanoid table tennis robot was developed. The robot was designed according to the human skeleton and was able to achieve improved human-machine interaction.

Notwithstanding these advances, most of the mentioned robots have the following issues: (1) Fixed robotic arms and humanoid table tennis robots can only hit the ball within a limited workspace; (2) There is no control over the attitude of the racket; (3) Current studies ignore the perception and prediction of the opponent's hitting behavior.

To deal with these issues, we develop a bionic table tennis robot in this paper that can reach large workspaces and return the ball to the desired location based on the opponent's hitting behaviors. The developed robot shows high adaptability and stability. Specifically, we first develop the vision system of the table tennis robot for tracking the trajectory of the ball and capturing the opponent's action. Based on the opponent's behavior, procedures are developed to predict the rotation type of the ball as well as the trajectory of the ball. To return the ball precisely, a dynamic model of the robot arm is established. Then, based on the quintic polynomial trajectory planning algorithm, we realize dynamic trajectory planning even when the desired endpoint of the robot arm is changing constantly. Finally, we show how to return the ball to the desired location. Note that, in our previous work [24], we discussed how to predict the trajectory of the ball and return the ball to the desired location. This work extends our previous work by considering the opponent's stroke behaviors when returning the ball. Compared with [24], the approach developed in this paper has a higher success rate of returning the ball to the target area. In addition, benefiting from the stroke behavior prediction, the developed robot system can respond to changes in the environment more quickly compared with [24].

The rest of this paper is organized as follows: Section 2 introduces the overall structure of our table tennis robot. Section 3 introduces the visual system. Section 4 develops the robot arm control algorithm. Section 5 elaborates on the location control of the ball. Section 6 describes the experiments undertaken and provides analysis of the results. Section 7 summarizes the paper and presents proposals for our future work.

2. The Table Tennis Robot System

The developed table tennis robot system consists of a vision module, an execution module, and a control module.

As shown in Figure 1, the vision module is composed of two visual systems, denoted by visual systems 1 and 2, respectively. Each visual system contains two Baumer HXC20 high-speed monochromatic industrial cameras. Cameras are installed above the table tennis table to track the trajectory of the ping-pong ball for visual system 1. Cameras are installed above the robot arm to track the stroke motion of the player for visual system 2. Two slide rails with a length of 185 cm are installed outside the two long sides of the table, and another slide rail, with a length of 220 cm, is placed on the two slides on both sides, enabling horizontal movement of the robot with 2-DoF. The control module is used for visual processing, data communication, and manipulator motion control.

The world coordinate system of the overall system is defined as follows: the origin of the system is located at the center of the table, the positive x-axis points to the robot along the centerline on the table, the positive y-axis points to the right side of the robotic arm along the width of the table, and the positive z-axis is vertical and upward.

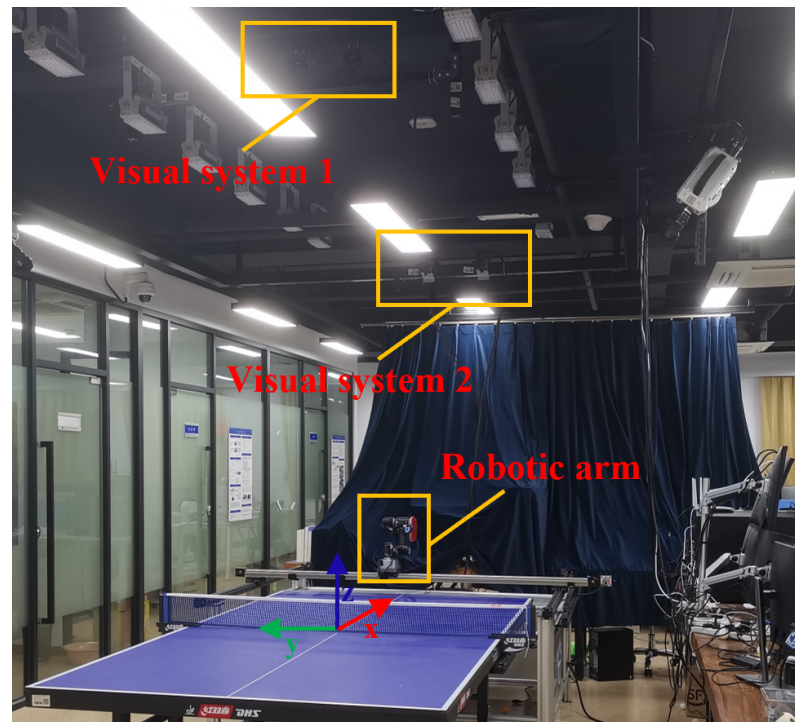


Figure 1. The developed table tennis robot system.

3. The Vision Module

3.1. Trajectory Prediction of Balls

Let us first discuss how to track the trajectory of the ball online. Specifically, we develop a moving object detection algorithm, which combines techniques of background subtraction and color segmentation. We first obtain the color threshold of the ball by comparing different images under different light conditions. It is shown that the color threshold of the ball is $[176, 255]$. Then, for each image, we recognize the ball by executing background subtraction, image binarization, morphology open operation, and corrosion expansion operation sequentially. Additionally, to improve the recognition accuracy and reduce the recognition time, we adopt the region of interest (ROI) algorithm to narrow the scan area of each image.

The internal and external parameters of the camera are obtained by Zhang's calibration method [25]. We conduct the coordinate transformation using the perspective-n-point (PnP) algorithm [26]. Then, we can obtain the three-dimensional coordinates of the ball.

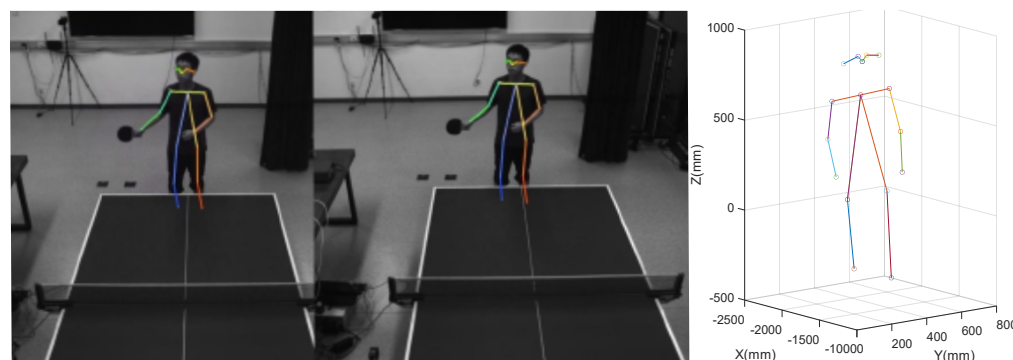
To control the robot to play table tennis with humans, it is crucial to predetermine the possible hitting point so that the robot can return the ball to the desired location in real-time. To achieve this, we need to predict the trajectory of the ball, which was solved in our previous work [24]. Roughly speaking, the process consists of three steps. First, a dynamic model of the ball is derived; second, to obtain the initial velocity of the ball, we develop an initial velocity correction method based on the feedback of the location of the ball; third, based on the dynamic model of the ball and the computed initial velocity, we calculate the hitting position of the ball. We refer the reader to [24] for more details.

3.2. Stroke Type Classification and Rotation Type Prediction

In this paper, we adopt the Fast Pose 17 human posture model [27]. The 17 key points of the human body for this model are given in Table 1. As shown in Figure 2, the binocular camera first identifies the location of the key points, and then obtains the 3D position of each key point via the 3D reconstruction. The identification error of the coordinate position of the key points is within 2 cm, which satisfies the experimental requirements.

Table 1. Key points of the Fast Pose 17 model.

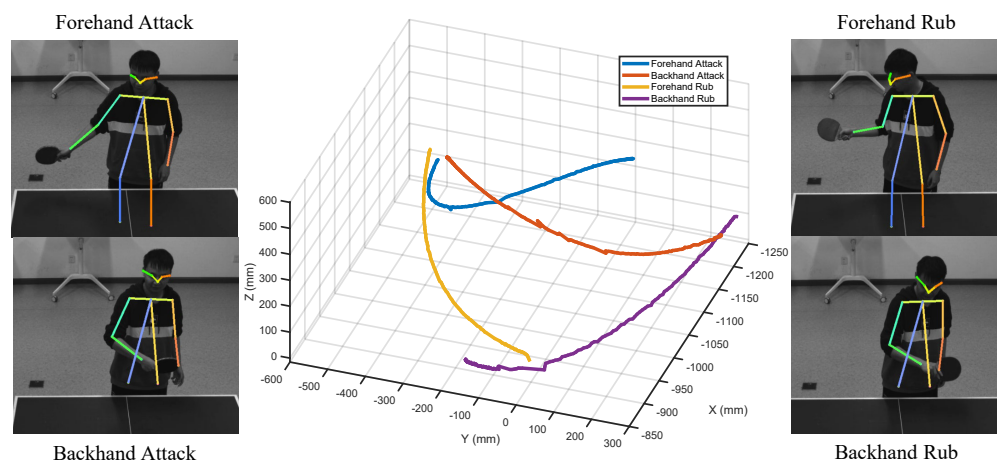
Number	Key Point	Number	Key Point
0	Nose	9	Left wrist
1	Left eye	10	Right wrist
2	Right eye	11	Left waist
3	Left ear	12	Right waist
4	Right ear	13	Left knee
5	Left shoulder	14	Right knee
6	Right shoulder	15	Left ankle
7	Left elbow	16	Right ankle
8	Right elbow		

**Figure 2.** Three-dimensional coordinates of the key points.

As shown in Table 2, each type of ball rotation corresponds to a different type of stroke, which includes forehand attacking, forehand rubbing, backhand attacking, and backhand rubbing. To better return a ball, we need to identify the stroke type of the opponent. To this end, for each stroke type, we collect 500 motion sequences of coordinate of the opponent's racket hand, as shown in Figure 3. Based on the collected data, the SVM (support vector machine) can be trained and then used to predict the stroke type of the opponent.

Table 2. Strokes and the corresponding types of the rotation.

Type	Stroke	Rotation
1	Forehand attacking	Topspin, left sidespin
2	Backhand attacking	Topspin, right sidespin
3	Forehand rubbing	Backspin, left sidespin
4	Backhand rubbing	Backspin, right sidespin

**Figure 3.** Four types of strokes.

Since the stroke types of the opponent and the motion sequence of the wrist are highly correlated, we select the position of the wrist as the feature point of the SVM algorithm. Specifically, we use the Fast Pose 17 human posture model to locate the position of a human's wrist. Then, the current velocity of the wrist is predicted by the Kalman filter algorithm [28]. Each feature vector of the SVM consists of velocities of the wrist in the past 20 samplings. For each stroke type, different control strategies to the end attitude of the robot arm are performed. The details of the control are discussed in the following sections.

By predicting the opponent's hitting behavior and trajectory, we can simulate the flight path and the rotation direction of the ball. The overall working diagram for the visual module is depicted in Figure 4.

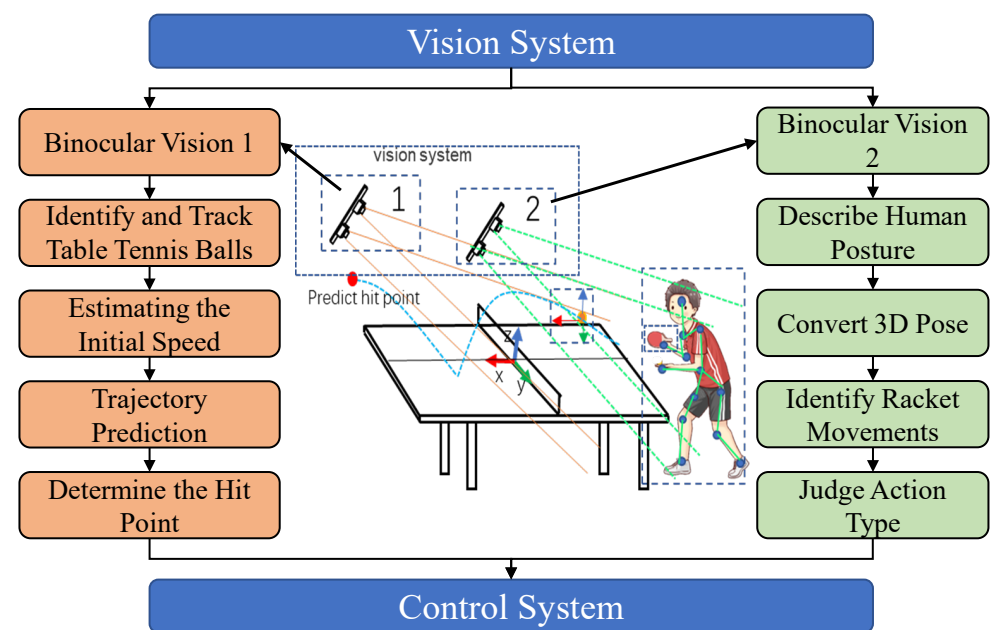


Figure 4. Flowchart of the visual system.

4. Humanoid Robotic Arm Control

To return the ball, we design a humanoid 4-DoF robotic arm. Figure 5 shows how the humanoid robotic arm simulates the forehand and backhand attacking or rubbing of a player. Specifically, Figure 5a–d show that the player has completed a forehand stroke; in Figure 5e–h, the robot imitates the forehand stroke behaviors shown in Figure 5a–d, respectively. Similarly, Figure 5i–l illustrate that the player has completed a backhand stroke, and in Figure 5m–p, the robot imitates the backhand stroke behaviors shown in Figure 5i–l, respectively.

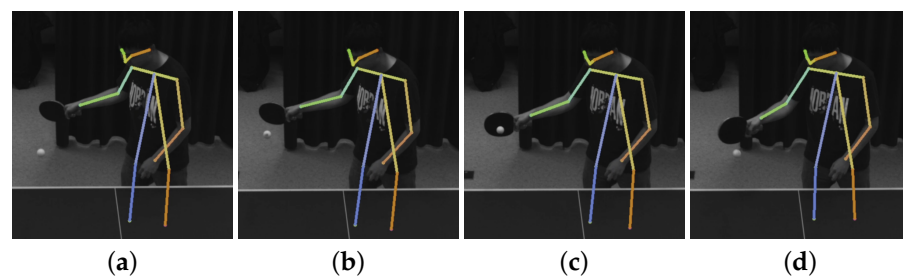


Figure 5. Cont.

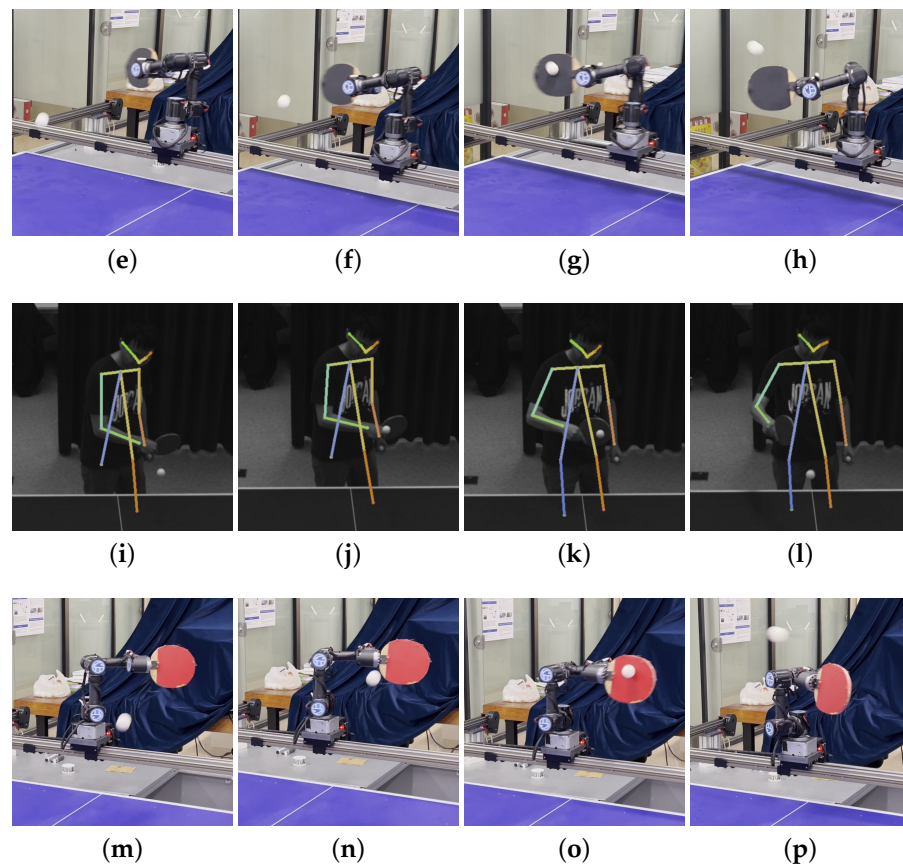


Figure 5. Forehand and backhand stroke simulations. (a) Forehand stroke: stage 1; (b) Forehand stroke: stage 2; (c) Forehand stroke: stage 3; (d) Forehand stroke: stage 4; (e) Forehand stroke imitation: stage 1; (f) Forehand stroke imitation: stage 2; (g) Forehand stroke imitation: stage 3; (h) Forehand stroke imitation: stage 4; (i) Backhand stroke: stage 1; (j) Backhand stroke: stage 2; (k) Backhand stroke: stage 3; (l) Backhand stroke: stage 4; (m) Backhand stroke imitation: stage 1; (n) Backhand stroke imitation: stage 2; (o) Backhand stroke imitation: stage 3; (p) Backhand stroke imitation: stage 4.

As shown in Figure 6, the execution module of the robot is comprised of the 4-DoF robotic arm and the two-dimensional slide rail. The robot is anthropomorphic in the sense that: (i) a 2-DOF rail (Rails 1 and 2) is used to imitate the movement of the human; (ii) joints 3 and 4 are used to imitate the rotations of the human waist and elbow, respectively; (iii) joints 5 and 6 function as a universal joint to imitate the human wrist. Benefiting from the anthropomorphic design of the robot arm, it can improve the accuracy of returning the ball.

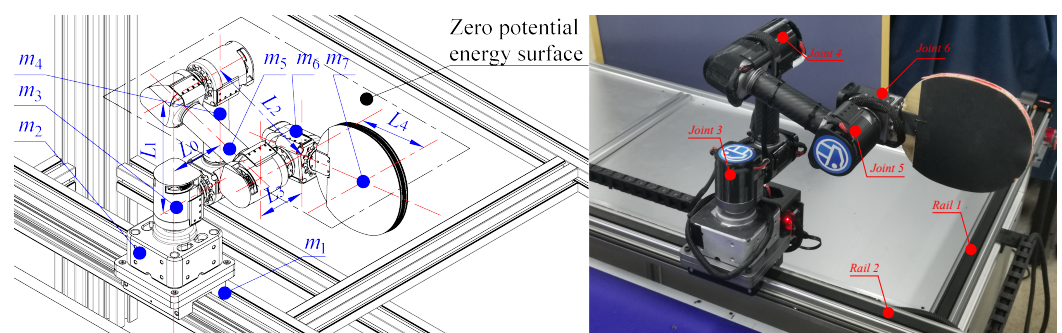


Figure 6. The structure of the robot arm.

To track the designed acceleration and velocity of the robot in real time, we need to analyze the robot's dynamic model. To this end, we first analyze the kinematics of the robot arm. As shown in Figure 6, we establish the coordinate system of the robot. The D-H parameter method is used to obtain the position and attitude of the racket. The D-H parameters of the table tennis robot system can be found in our previous work [24].

4.1. Joint Position Calculation

Here, i denotes the i th joint, d_i denotes the offset at the joint i axis, θ_i is the angle of joint i , and L_1, \dots, L_4 are illustrated in Figure 6. The homogeneous transformation matrix between two adjacent coordinate systems can be derived as ${}^i_{i-1}T$. The transformation matrix from the base to the terminal of the robot arm can be calculated as follows:

$${}^0_6T = {}^0_1T {}^1_2T {}^2_3T {}^3_4T {}^4_5T {}^5_6T = \begin{bmatrix} {}^0_6R & {}^0_6P \\ 0 & 1 \end{bmatrix} \quad (1)$$

$${}^0_6R = \begin{bmatrix} {}^0_6R_1 & {}^0_6R_2 & {}^0_6R_3 \end{bmatrix} \quad (2)$$

$${}^0_6R_1 = \begin{bmatrix} c(\theta_4)c(\theta_5)s(\theta_6)s(\theta_3) - c(\theta_3)s(\theta_6) - c(\theta_6)s(\theta_3)s(\theta_4)s(\theta_5) \\ c(\theta_3)c(\theta_6)s(\theta_4)s(\theta_5) - c(\theta_3)c(\theta_4)c(\theta_5)c(\theta_6) - s(\theta_3)s(\theta_6) \\ s(\theta_4 + \theta_5)c(\theta_6) \end{bmatrix} \quad (3)$$

$${}^0_6R_2 = \begin{bmatrix} s(\theta_3)s(\theta_4)s(\theta_5)s(\theta_6) - c(\theta_4)c(\theta_5)s(\theta_3)s(\theta_6) - c(\theta_3)c(\theta_6) \\ c(\theta_3)c(\theta_4)c(\theta_5)s(\theta_6) - c(\theta_6)s(\theta_3) - c(\theta_3)s(\theta_4)s(\theta_5)s(\theta_6) \\ -s(\theta_4 + \theta_5)s(\theta_6) \end{bmatrix} \quad (4)$$

$${}^0_6R_3 = \begin{bmatrix} s(\theta_4 + \theta_5)s(\theta_3) \\ -s(\theta_4 + \theta_5)c(\theta_3) \\ -c(\theta_4 + \theta_5) \end{bmatrix} \quad (5)$$

$${}^0_6P = \begin{bmatrix} d_2 - L_2c(\theta_3) + L_3c(\theta_4)s(\theta_3) + L_4c(\theta_4)s(\theta_3)s(\theta_5) + L_4c(\theta_5)s(\theta_3)s(\theta_4) \\ -d_1 - L_2s(\theta_3) - L_3c(\theta_3)c(\theta_4) - L_4c(\theta_3)c(\theta_4)s(\theta_5) - L_4c(\theta_3)c(\theta_5)s(\theta_4) \\ L_1 - L_4 - c(\theta_4 + \theta_5) + L_3s(\theta_4) \end{bmatrix} \quad (6)$$

where 0_6R and 0_6P represent the rotation matrix and the translation matrix between the base coordinate system and the terminal coordinate system, respectively; $c(\theta)$ represents $\cos(\theta)$; $s(\theta)$ represents $\sin(\theta)$. To precisely control the posture and position of the racket, an algebraic approach is applied to obtain the inverse kinematic model. The elements are transformed from Euler angles or quaternions as:

$${}^0_6T_m = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (7)$$

Based on the D-H parameters, we obtain all the required variables by substituting into the transformation matrix ${}^i_{i-1}T$.

By Algorithm 1, there are 16 possible solutions of the robot inverse kinematics, of which 2–8 solutions are correct, and the remaining solutions are wrong. In practical applications, it is necessary to estimate the correctness of the solutions in real-time and then choose the most appropriate solution according to the optimal control principle of the shortest displacement.

4.2. Joint Velocity Calculation and Trajectory Planning

To return the ball successfully, we need to control the robot to reach the final hitting point quickly and accurately while preserving the continuous acceleration and jerk of the motors and avoiding vibrations and shocks on the joints. To this end, we execute the trajectory planning for all joints of the robot arm to ensure its accuracy and stability and to keep it as continuous as possible.

Algorithm 1: Solving parameters of 0T_m

Solving $\theta_4 + \theta_5 = \pm a \cos(a_z)$

Solving $\theta_6 = \pm a \cos(n_z / \sin(\theta_4 + \theta_5))$

Solving $\theta_3 = \pm a \cos(n_y \cos(\theta_6) - o_y \sin(\theta_6) / a_z)$

Solving $\theta_4 = \pm a \sin((p_z - L_4 a_z - L_1) / L_3)$ or

$$\theta_4 = \begin{cases} \pi - a \sin((p_z - L_4 a_z - L_1) / L_3) & \text{if } 0 \leq a \sin((p_z - L_4 a_z - L_1) / L_3) \leq \pi/2 \\ -\pi - a \sin((p_z - L_4 a_z - L_1) / L_3) & \text{if } -\pi/2 \leq a \sin((p_z - L_4 a_z - L_1) / L_3) \leq 0 \end{cases} \quad (8)$$

Solving $\theta_5 = (\theta_4 + \theta_5) - \theta_4$

Solving

$$d_1 = -p_y - L_2 s(\theta_3) - L_3 c(\theta_3) c(\theta_4) - L_4 c(\theta_3) c(\theta_4) s(\theta_5) - L_4 c(\theta_3) c(\theta_5) s(\theta_4)$$

Solving $d_2 = p_x + L_2 c(\theta_3) - L_3 s(\theta_4) s(\theta_3) - L_4 c(\theta_4) s(\theta_3) s(\theta_5) - L_4 c(\theta_5) s(\theta_3) s(\theta_4)$

First, we need to obtain the angular velocity and the angle of each joint when the racket reaches the final hitting point and meets the expected hitting velocity. The angle of rotation of each joint can be obtained by inverse kinematics, and the angular velocity of each joint can be obtained by

$$[v_{d1} \ v_{d2} \ v_{\theta3} \ v_{\theta4} \ v_{\theta5} \ v_{\theta6}]^T = J^{-1} [v_x \ v_y \ v_z \ \omega_x \ \omega_y \ \omega_z]^T \quad (9)$$

where v_{d1} and v_{d2} are the linear velocities of the two slide rails, $v_{\theta3} \sim v_{\theta6}$ are the angular velocities of the joints of the robot arm, v_x, v_y, v_z are the linear velocities of the racket in the x, y , and z directions, respectively, $\omega_x, \omega_y, \omega_z$ are the angular velocities of the racket in the x, y , and z directions, respectively, and J^{-1} is the inverse matrix of the Jacobi matrix of the table tennis robot.

Next, we perform trajectory planning for each joint of the robot using quintic polynomials as follows.

$$\theta_i(t) = c_{i0} + c_{i1}t + c_{i2}t^2 + c_{i3}t^3 + c_{i4}t^4 + c_{i5}t^5 \quad (10)$$

Substituting the computed $c_{i0} \sim c_{i5}$ into (10), we obtain the position that the racket will reach in the next moment t_s . The above process is repeated after each t_s . Note that $\theta_i(t_s)$ and $\dot{\theta}_i(t_s)$ are given by the stroke decision returned by the vision module and are reset after each stroke. In addition, note that $\ddot{\theta}_i(t_s) = 0$, i.e., the angular acceleration of joint i is zero when hitting the ball. In this way, we can drive the racket to the desired position with the desired velocity. The trajectory planning for this stroke is complete.

5. Ball Location Control

In this section, we first establish the collision rebound model. A square area of 60 cm in length and width is set as the target area to return the ball. Then, based on the established collision rebound model, we can calculate the stroke speed and angle for the robot to return the ball to the target area. When the ball collides with the table, its velocity will decrease in both the horizontal and vertical directions. The collision rebound model can be expressed by:

$$\vec{V}_{out} = K_t \vec{V}_{in} + \vec{B}_t, \quad (11)$$

where K_t is a diagonal matrix representing the velocity loss coefficient after the ball collides with the table, \vec{B}_t is the velocity compensation bias after the collision, \vec{V}_{in} is the incident velocity of the collision between the ball and the table, and \vec{V}_{out} is the exit velocity after the collision between the ball and the table. The coordinates of \vec{V}_{in} and \vec{V}_{out} should be relatively static with respect to the collision plane.

The axes of these coordinates are perpendicular to the collision plane and point in the ball's incoming direction.

The flight trajectory of the ball after the collision can be obtained through the prediction model according to the calculated exit velocity \vec{V}_{out} and the coordinates of the collision point.

Before each stroke, the flight trajectory of the incoming ball can be predicted, and the hitting position, velocity, and time can also be calculated. To return the ball to the desired location, we can compute the initial velocity of the ball based on feedback on the location of the ball. Specifically, according to the hitting position P_0 and the central point of the target area P_n , we can obtain the desired velocity of the ball after hitting. By using the collision rebound model, the transformation matrix of the ball's velocity between the world coordinate system and the racket coordinate system can be derived as follows:

$$R = \begin{bmatrix} -\cos(\phi)\sin(\psi) & \cos(\phi)\cos(\psi) & \sin(\phi) \\ -\sin(\psi)\sin(\phi) & \cos(\psi)\sin(\phi) & -\cos(\phi) \\ -\cos(\psi) & -\sin(\psi) & 0 \end{bmatrix} \quad (12)$$

where ϕ and ψ are the *pitch* and *psi* in the Tait–Bryan angles.

In addition to the collision rebound model between the ball and the table, there is a similar collision rebound model between the ball and the racket. We have

$$R(\vec{V}_r^w - \vec{V}_O) = K_r R(\vec{V}_I - \vec{V}_O) + \vec{B}_r, \quad (13)$$

where $\vec{V}_I = [v_{Ix}, v_{Iy}, v_{Iz}]$, $\vec{V}_O = [v_{Ox}, v_{Oy}, v_{Oz}]$, and $\vec{V}_r^w = [v_{rx}, v_{ry}, v_{rz}]$ are the velocity of the ball before the collision, the velocity of the ball after the collision, and the velocity of the racket with respect to the world coordinate system. Similarly, K_r is a diagonal matrix representing the velocity loss coefficient between the ball and the racket, k_x, k_y, k_z are the elements on the diagonal of the K_r , and \vec{B}_r is the velocity compensation bias after the collision. Let the velocities of the ball in the directions of Z and Y be 0. Then, we can obtain the following nonlinear equation system:

$$v_{Oz}S_\phi - b_x - v_{Iz}k_xS_\phi + v_{Oy}C_\phi C_\psi + C_\phi S_\psi(v_{rx}^w - v_{Ox}) - v_{Iy}k_xC_\phi C_\psi - k_xC_\phi S_\psi(v_{rx}^w - v_{Ix}) = 0 \quad (14)$$

$$v_{Oy}C_\psi S_\phi - v_{Oz}C_\phi - b_y + S_\phi S_\psi(v_{rx} - v_{Ox}) + v_{Iz}k_yC_\phi - k_yS_\phi S_\psi(v_{rx} - v_{Ix}) - v_{Iy}k_yC_\psi S_\phi = 0 \quad (15)$$

$$C_\psi(v_{rx} - v_{Ox}) - v_{Oy}S_\psi - b_z + v_{Iy}k_zS_\psi - k_zC_\psi(v_{rx} - v_{Ix}) = 0, \quad (16)$$

where $S_\phi = \sin(\phi)$, $C_\phi = \cos(\phi)$, $S_\psi = \sin(\psi)$, and $C_\psi = \cos(\psi)$.

By solving these equations, we can obtain the racket speed and attitude for returning the ball to the desired area. Because of the complexity of the nonlinear equations, it is difficult to directly calculate the analytical solution. To deal with this issue, we propose Algorithm 2 to approach the solutions from the initial value by using the Newton method.

The reason we use a linear model to model the ping-pong ball collision is that we consider the collision to be inelastic. Therefore, the problem of modeling the collision of ping-pong balls can be solved by determining the K_t , K_r , \vec{B}_t and \vec{B}_r . By trial and error, we get

$$K_t = \begin{bmatrix} 0.50259 & 0 & 0 \\ 0 & 0.75 & 0 \\ 0 & 0 & -0.9 \end{bmatrix} \vec{B}_t = \begin{bmatrix} 0.50652 \\ 0 \\ 0 \end{bmatrix} K_r = \begin{bmatrix} 0.82 & 0 & 0 \\ 0 & -0.88 & 0 \\ 0 & 0 & -0.82 \end{bmatrix} \vec{B}_r = \begin{bmatrix} 0.1 \\ 0 \\ 0.1 \end{bmatrix} \quad (17)$$

Using the inverse kinematics, the desired joint position can be calculated. Then, by applying the trajectory planning in the joint space, the angle, angular velocity, and angular acceleration of each joint of the robot can be uniquely determined over time.

Algorithm 2: Solutions of the nonlinear equations

Step 1: Let Equation (14) be f_1 , Equation (15) be f_2 , and Equation (16) be f_3 .

Let $\phi_0 = \pi/2$, $v_{rx0} = -2m/s$, $\psi_0 = \pi/2$ if $Y > 0$, and $\psi_0 = -\pi/2$ if $Y \leq 0$.

Let the jacobian matrix F' be

$$F'(\phi, \psi, v_{rx}) = \begin{bmatrix} \frac{\partial f_1}{\partial \phi} & \frac{\partial f_1}{\partial \psi} & \frac{\partial f_1}{\partial v_{rx}} \\ \frac{\partial f_2}{\partial \phi} & \frac{\partial f_2}{\partial \psi} & \frac{\partial f_2}{\partial v_{rx}} \\ \frac{\partial f_3}{\partial \phi} & \frac{\partial f_3}{\partial \psi} & \frac{\partial f_3}{\partial v_{rx}} \end{bmatrix}$$

Step 2: Solving elements of F' : $\frac{\partial f_1}{\partial \phi}$, $\frac{\partial f_1}{\partial \psi}$, $\frac{\partial f_1}{\partial v_{rx}}$, $\frac{\partial f_2}{\partial \phi}$, $\frac{\partial f_2}{\partial \psi}$, $\frac{\partial f_2}{\partial v_{rx}}$, $\frac{\partial f_3}{\partial \phi}$, $\frac{\partial f_3}{\partial \psi}$, $\frac{\partial f_3}{\partial v_{rx}}$

Step 3: Let $S_0 = \begin{bmatrix} \phi_0 \\ \psi_0 \\ v_{rx0} \end{bmatrix}$.

Step 4: Calculate $S_1 = S_0 - F'(S_0)^{-1} \begin{bmatrix} f_1(S_0) \\ f_2(S_0) \\ f_3(S_0) \end{bmatrix}$.

Step 5: If $S_1 - S_0 \leq 10^{-6}$, return S_1 , and otherwise, set $S_0 \leftarrow S_1$ and go to Step 4.

6. Experiment

As shown in Figures 7 and 8, a target square area is prespecified and $(-900, 200)$ is set as the target point. In Figure 7, the squares composed of the dotted lines and solid lines correspond to the outer square and the inner square, respectively, in Figure 8. Moreover, to clearly show the performance of our method, we use “+” to mark the target point.

In this experiment, the robot plays against a player who returns the ball at a random velocity and angle. Specifically, the robot plays a total of 20,648 rounds to approximate the performance of the robot precisely. The stroke type prediction accuracy and success rate of returning the ball to the target area are shown in Tables 3 and 4. In Table 3, the prediction accuracy of the model for forehand attack, backhand attack, forehand rub and backhand rub is 95.52%, 94.8%, 93.17% and 93.41% respectively. It can be seen that the rub stroke has a worse prediction accuracy. This is mostly because our model has a higher propensity to identify the rub as an attack. The detail of the experimental results are shown in Figure 7.

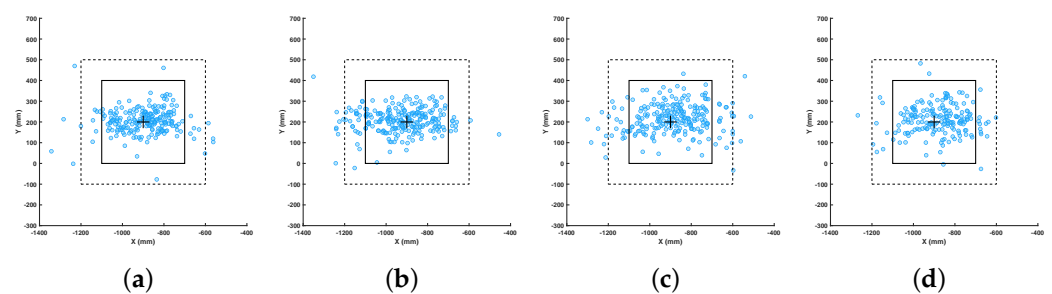


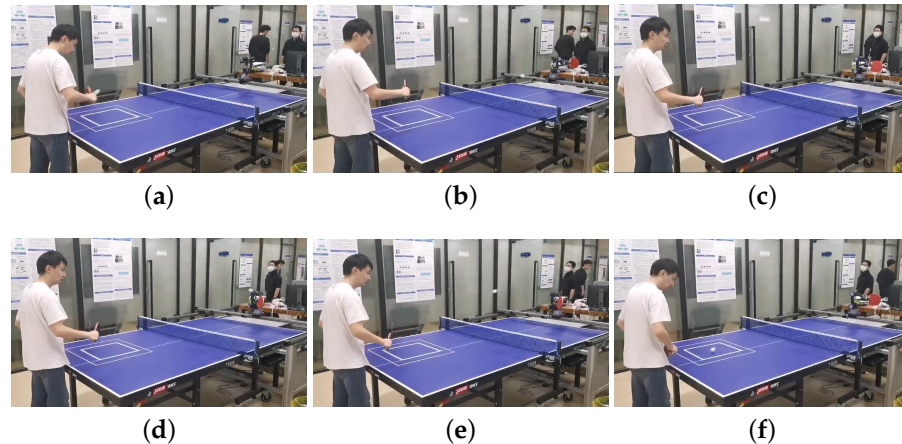
Figure 7. Actual landing points and target area (4 experiments with same target point). (a) Landing points of the returned ball: forehand attack; (b) Landing points of the returned ball: backhand attack; (c) Landing points of the returned ball: forehand rub; (d) Landing points of the returned ball: backhand rub.

Table 3. Stroke type prediction accuracy.

Stroke Type	Accuracy
Forehand attack	95.52%
Backhand attack	94.8 %
Forehand rub	93.17%
Backhand rub	93.41%

Table 4. Success rate of returning the ball to the target area.

Rounds	Hitting Success Rate	Within the Inner Target Area	Within the Outlier Target Area	Mean Placement of the Ball
20,648	98.35%	82.52%	95.16%	(−875.13, 178.27)

**Figure 8.** Experimental process. (a) Backhand attack: touching; (b) Backhand attack: flying; (c) Backhand attack: landing; (d) Backhand attack return: touching; (e) Backhand attack return: flying; (f) Backhand attack return: landing.

Following further investigation, we discovered that misidentification was the primary cause of several balls failing to strike the inner target area. Our identification system may mistake the player's white cuff for a ball, thus supplying the robot with the incorrect position of the ball. Although the influence of this error can be gradually eliminated in trajectory prediction, if the recognition error is too large, the robot's performance will suffer and the ball will finally travel out of bounds.

7. Conclusions

This paper proposed a bionic table tennis robot system. A novel mechanical structure and a robust algorithm were designed. Extensive experimental results demonstrated that the algorithm can effectively improve the stability and accuracy of the robot's performance. However, due to the limitations of the mechanical structure, the robot failed in most of the cases where the returned ball had a high speed exceeding 10 m/s as well as strong rotation. In the future, both the structure and algorithm of the table tennis robot will be further refined to enable the robot to return the ball with even greater speed and spin.

Author Contributions: Conceptualization, Y.J. and Y.H.; methodology, Y.J., X.H. and Y.M.; software, Y.J., X.H., Y.Y. and Y.M.; validation, Y.J., Y.M. and F.S.; formal analysis, Y.J., Y.Y. and F.S.; investigation, Y.J.; resources, Y.J. and Y.H.; data curation, F.S.; writing—original draft preparation, Y.J. and Y.M.; writing—review and editing, Y.J., Y.M., X.H. and Y.H.; visualization, Y.J., X.H. and Y.M.; supervision, Y.J. and Y.H.; project administration, Y.J. and Y.H.; funding acquisition, Y.J., Y.Y. and Y.H. All authors have read and agreed to the published version of the manuscript.

Funding: This study is supported by the Young Scientists Fund of the National Natural Science Foundation of China (Grant No. 62206175).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The data are not publicly available due to privacy restrictions.

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

References

- Li, Y.; Li, B.; Ruan, J.; Rong, X. Research of mammal bionic quadruped robots: A review. In Proceedings of the 5th IEEE Conference on Robotics, Automation and Mechatronics (RAM), Qingdao, China, 17–19 September 2011; pp. 166–171.
- Fan, X.; Sayers, W.; Zhang, S.; Han, Z.; Chizari, H. Review and Classification of Bio-inspired Algorithms and Their Applications. *J. Bionic Eng.* **2020**, *17*, 611–631. [\[CrossRef\]](#)
- Mukherjee, D.; Gupta, K.; Li, H.C.; Najjaran, H. A Survey of Robot Learning Strategies for Human-Robot Collaboration in Industrial Settings. *Robot. Comput.-Integr. Manuf.* **2022**, *73*, 102231. [\[CrossRef\]](#)
- Arents, J.; Greitans, M. Smart industrial robot control trends, challenges and opportunities within manufacturing. *Appl. Sci.* **2022**, *12*, 937. [\[CrossRef\]](#)
- Billard, A.; Kragic, D. Trends and challenges in robot manipulation. *Science* **2019**, *364*, eaat8414. [\[CrossRef\]](#) [\[PubMed\]](#)
- Tang, C.; Sun, L.; Zhou, G.; Shu, X.; Tang, H.; Wu, H. Gait Generation Method of Snake Robot Based on Main Characteristic Curve Fitting. *Biomimetics* **2023**, *8*, 105. [\[CrossRef\]](#) [\[PubMed\]](#)
- Gu, S.; Meng, F.; Liu, B.; Zhang, Z.; Sun, N.; Wang, M. Stability Control of Quadruped Robot Based on Active State Adjustment. *Biomimetics* **2023**, *8*, 112. [\[CrossRef\]](#) [\[PubMed\]](#)
- Guo, X.; Ma, S.; Li, B.; Fang, Y. A Novel Serpentine Gait Generation Method for Snakelike Robots Based on Geometry Mechanics. *IEEE/ASME Trans. Mechatron.* **2018**, *23*, 1249–1258. [\[CrossRef\]](#)
- Zhang, Y.; Wu, Z.; Wang, J.; Tan, M. Design and Analysis of a Bionic Gliding Robotic Dolphin. *Biomimetics* **2023**, *8*, 151. [\[CrossRef\]](#) [\[PubMed\]](#)
- Seok, S.; Wang, A.; Meng, Y.C.; Otten, D.; Lang, J.; Kim, S. Design principles for highly efficient quadrupeds and implementation on the MIT Cheetah robot. In Proceedings of the 2013 IEEE International Conference on Robotics and Automation, Karlsruhe, Germany, 6–10 May 2013.
- Liu, G.H.; Lin, H.Y.; Lin, H.Y.; Chen, S.T.; Lin, P.C. A Bio-Inspired Hopping Kangaroo Robot with an Active Tail. *J. Bionic Eng.* **2014**, *11*, 541–555. [\[CrossRef\]](#)
- Duraisamy, P.; Sidharthan, R.K.; Santhanakrishnan, M.N. Design, Modeling, and Control of Biomimetic Fish Robot: A Review. *J. Bionic Eng.* **2019**, *16*, 967–993.
- Westervelt, E.R.; Grizzle, J.W.; Chevallereau, C.; Choi, J.H.; Morris, B. *Feedback Control of Dynamic Bipedal Robot Locomotion*; CRC Press: Boca Raton, FL, USA, 2018.
- Tebbe, J.; Klamt, L.; Gao, Y.; Zell, A. Spin detection in robotic table tennis. In Proceedings of the 2020 IEEE International Conference on Robotics and Automation (ICRA), Paris, France, 31 May–31 August 2020; pp. 9694–9700.
- Gomez-Gonzalez, S.; Prokudin, S.; Schölkopf, B.; Peters, J. Real time trajectory prediction using deep conditional generative models. *IEEE Robot. Autom. Lett.* **2020**, *5*, 970–976. [\[CrossRef\]](#)
- 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. [\[CrossRef\]](#)
- Mulling, K.; Kober, J.; Peters, J. A biomimetic approach to robot table tennis. *Adapt. Behav.* **2010**, *19*, 359–376. [\[CrossRef\]](#)
- Hashimoto, H.; Ozaki, F.; Osuka, K. *Development Of A Pingpong Robot System Using 7 Degrees Of Freedom Direct Drive Arm*; Industrial Applications of Robotics and Machine Vision; SPIE: Bellingham, WA, USA, 1987.
- Andersson, R.L. *A Robot Ping-Pong Player: Experiment in Real-Time Intelligent Control*; MIT Press: Cambridge, MA, USA, 1988.
- Nakashima, A.; Kobayashi, Y.; Hayakawa, Y. Paddle Juggling of one Ball by Robot Manipulator with Visual Servo. In Proceedings of the International Conference on Control, Singapore, 5–8 December 2006; pp. 5347–5352.
- Tebbe, J.; Gao, Y.; Sastre-Rienietz, M.; Zell, A. A Table Tennis Robot System Using an Industrial KUKA Robot Arm. In Proceedings of the German Conference on Pattern Recognition, Stuttgart, Germany, 9–12 October 2018.
- Satoshi, Y. Table Tennis Robot “Forpheus”. *J. Robot. Soc. Jpn.* **2011**, *38*, 19–25.
- Sun, Y.; Xiong, R.; Zhu, Q.; Wu, J.; Chu, J. Balance motion generation for a humanoid robot playing table tennis. In Proceedings of the IEEE-RAS International Conference on Humanoid Robots, Bled, Slovenia, 26–28 October 2011; pp. 19–25.
- Ji, Y.; Hu, X.; Chen, Y.; Mao, Y.; Zhang, J. Model-Based Trajectory Prediction and Hitting Velocity Control for a New Table Tennis Robot. In Proceedings of the 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Prague, Czech Republic, 27 September–1 October 2021.
- Zhang, Z.; Zhang, P.; Chen, G.; Chen, Z. Camera Calibration Algorithm for Long Distance Binocular Measurement. In Proceedings of the 2022 IEEE International Conference on Mechatronics and Automation (ICMA), Guilin, China, 7–10 August 2022; pp. 1051–1056.
- Lu, X.X. A review of solutions for perspective-n-point problem in camera pose estimation. *J. Phys. Conf. Ser.* **2018**, *1087*, 052009. [\[CrossRef\]](#)

27. Zhang, F.; Zhu, X.; Ye, M. Fast human pose estimation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 3517–3526.
28. Welch, G.F. Kalman filter. In *Computer Vision: A Reference Guide*; Springer: Cham, Switzerland, 2020; pp. 1–3.

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.