S & M 3653

Fuzzy Control Design for Table Tennis Robot to Strike Spinning Balls

Chung-Hsun Sun,[1] Ying-Shu Chuang,[1] Wen-Jia Wu,[1] and Hsiang-Chieh Chen[2]*

[1]Department of Electrical Engineering, National Kaohsiung University of Science and Technology,
Kaohsiung City 807618, Taiwan
[2]Department of Mechanical Engineering, National Central University, Taoyuan City 302317, Taiwan

(Received October 20, 2023; accepted April 26, 2024)

Keywords: table tennis robot, topspinning/backspinning ball, fuzzy control

In this study, we propose a fuzzy control design for a table tennis robot to accurately strike ping-pong balls, regardless of whether they are topspinning or backspinning. The classification of topspin and backspin is determined through a comparison of ball trajectories measured by binocular vision with that of an ideal flying model derived from aerodynamics. By inputting trajectory features into the corresponding model, the striking signal, including contact position and hitting time, can be predicted. A fuzzy inference system is then utilized to derive the face angle of the racket at the hitting moment, which ultimately controls the robot to strike topspinning or backspinning balls. With an average error of approximately 3 cm for contact position prediction and less than 13.5 ms for hitting time prediction, our proposed fuzzy control design has demonstrated its effectiveness in striking spinning balls.

1. Introduction

Numerous studies have delved into the research of table tennis robots over the last few decades.[1–6] In the past, there was little discussion about robots that could strike topspinning/backspinning balls.[7] However, given the prevalence of topspinning/backspinning balls in table tennis games, in this study, we aim to introduce a fuzzy controller that enables our robot to hit such spinning balls. The accurate prediction of a spinning ball's trajectory is crucial in developing a ping-pong robot, and the key to achieving this lies in correctly classifying its spin type. Over the past years, several researchers have proposed various approaches to tackle this classification problem. Wang and Sun[8] focused on improving the neural network architecture of an extreme learning machine, successfully classifying five types of spinning balls with a high degree of accuracy. Gao *et al.*[9] employed a different method of analyzing the moving status of four lines affixed to a racket to recognize the spin type of a flying ball. Feature matching was the approach used in Ref. 10, which estimated the movement of an opponent's racket and used this information to determine the spinning direction of the ball.[10] In contrast, an inertial sensor was installed on the racket to classify the spin type on the basis of the racket's movement.[11] In addition, Furuno *et al.*[12] proposed a method that involved painting lines of five different colors on the ball and analyzing their variation to estimate the ball's spinning velocity. Several markers

[*]Corresponding author: e-mail: hcchen@cc.ncu.edu.tw

https://doi.org/10.18494/SAM4724

ISSN 0914-4935 © MYU K.K.
https://myukk.org/

were drawn on the ball to calculate the rotation speed.(13) Lastly, a method to detect the labels on a ping-pong ball and estimate the spinning speed on the basis of the change of labels was proposed in Ref. 14. However, these studies required a visual system with extremely high resolution and high computing capability, as estimating the spinning speed using a low-level camera is often difficult. As such, the Magnus effect is often ignored, resulting in trajectory predictions that differ from those measured by vision.(8) In this study, we seek to bridge this gap by adopting the difference between the trajectory predictions measured by vision and those derived from the spinning speed estimation to classify the spin type of a ball.

The vertical position and hitting timing of a flying ball are significantly affected by its topspin or backspin. To predict a robot's contact point position and hitting time, we model the shift in vertical position and hitting time in relation to the topspin/backspin. When hitting a spinning ball, the rebounding force differs owing to the direction and speed of the spin and the force from contact with the racket. For a topspinning ball, an upward force is generated, resulting in a high returning angle that may cause the ball to fly out of bounds. Tilting the racket forward and downward can reduce the rebounding angle and prevent this situation. Conversely, the racket should be tilted backward to increase the angle when hitting a backspinning ball. Obtaining the correct face angle of the racket requires the simultaneous consideration of the spinning status and contact position. To achieve this, a fuzzy controller is introduced using spinning speed and contact position as inputs. Our results showed average errors of 3 cm for the contact position and less than 13.5 ms for the hitting time. The proposed fuzzy controller is effective in striking topspinning or backspinning balls, as demonstrated by our experiments. In this study, we created a robotic arm and developed a visual system to measure and predict the trajectory of a ball. We used the prediction results to control the movement and angle of a racket to strike the spinning ball. Our demonstration showcased a cost-effective and self-made robotic arm that can hit a spinning ball. These achievements are highly relevant to the topic of this journal–applications: robotics.

2. Table Tennis Robot

Figure 1 shows the developed robotic system used in this study, which includes a robotic arm, a binocular vision system, and a computer. Additionally, a launcher machine is employed to serve a ball to verify the proposed control design.

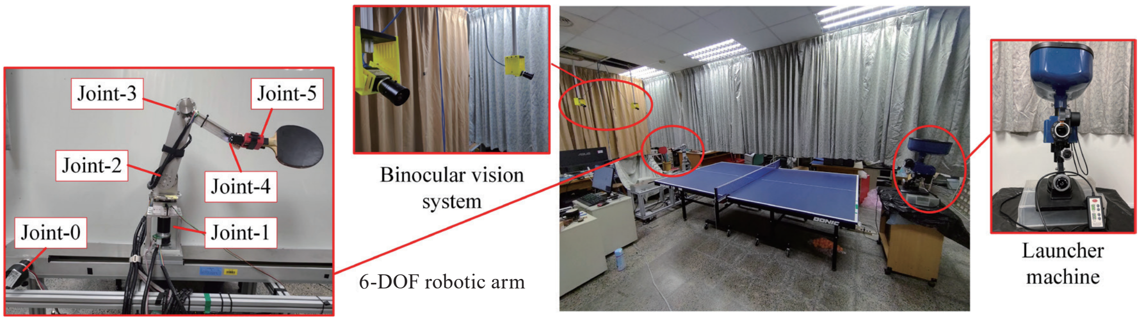


Fig. 1. (Color online) Developed robotic system.

2.1 Robotic arm

The 6-degree-of-freedom robotic arm used in this system is shown in Fig. 1. Joint-0 can drive the slide rail mechanism to move the robotic arm along the horizontal direction (*X*-coordinate). Joint-1 can derive the robotic arm swing along the *Z*-coordinate to strike the ball. Joints-2–4 form a planar link structure to move the racket on a defined hitting plane. Joint-5 can control the tilt angle of the racket for striking the topspinning/backspinning balls.

2.2 Binocular vision system

The binocular shown in Fig. 1 consists of two high-speed cameras that can capture images at the resolution of 1280 × 1024 pixels with a frame rate of 60 fps. By thresholding the captured images represented in HSV color space and the smallest enclosing circle algorithm, the ping-pong ball can be detected. Through the calibration described in Ref. 15, the intrinsic and extrinsic parameters of the binocular vision system can be obtained. The ball is then positioned through the stereo geometry of the binocular vision system.

3. Ball Trajectory Analysis

The practical ball trajectories can be estimated on the basis of the positions measured with the visual system. In this study, the type of ball spin is recognized by comparing the practical ball trajectory with the trajectory determined using the ideal flying model without the Magnus effect. Consequently, the contact point and hitting time for the table tennis robot can be predicted.

3.1 Ball spin recognition

The ideal flying model that ignored the Magnus effect is as follows:[7]

$$\begin{bmatrix} x(k) \\ y(k) \\ z(k) \\ v_x(k) \\ v_y(k) \\ v_z(k) \end{bmatrix} = \begin{bmatrix} x(k-1) \\ y(k-1) \\ z(k-1) \\ v_x(k-1) \\ v_y(k-1) \\ v_z(k-1) \end{bmatrix} + \begin{bmatrix} v_x(k-1) \\ v_y(k-1) \\ v_z(k-1) \\ -K_m \boldsymbol{v}(k-1) v_x(k-1) \\ -K_m \boldsymbol{v}(k-1) v_y(k-1) \\ -K_m \boldsymbol{v}(k-1) v_z(k-1) - g \end{bmatrix} T, \tag{1}$$

where

$$K_m = \frac{1}{2m} \rho S C_D \,. \tag{2}$$

Here, ρ is the air density, m is the mass of the ball, S is the effective cross-sectional ball area, C_D is the drag coefficient, g is the gravity acceleration, and v is the velocity of the ball. According to aerodynamics, the Magnus force caused by the spin of a ball during its flight is as follows:

$$F_M = \rho r_b S C_L \omega \times v , \tag{3}$$

where ω is the rotation velocity, r_b is the radius of the ball, and C_L is the lift coefficient.

According to Newton's second law, the Magnus force applied to the ball is as follows:

$$m a_z = \rho r_b \omega S C_L v_y . \tag{4}$$

Then, we obtain

$$\omega \propto {}^{a_z}\!/_{v_y} , \tag{5}$$

where v_y is the flying velocity along the Y-coordinate and a_z is the acceleration along the Z-coordinate. The spin type and spinning strength of a ball can be estimated using Eq. (5). Note that in this study, we did not measure the ball's rotation velocity; instead, we estimated the flight velocity and acceleration, which can be measured through the binocular vision system. The differences between the practical ball trajectory and the ideal flying model that ignored the Magnus effect are illustrated in Fig. 2. Obviously, the height difference is caused by a_z.

3.2 Contact point prediction

Figure 3(a) shows the definition of the coordinate system. According to Eqs. (3) and (5), the topspinning/backspinning ball flying along the Y-coordinate only induces the Magnus effect along the Z-coordinate. Hence, the trajectory of the topspinning/backspinning ball projected on the X-Y plane is regarded as a straight line, as shown in Fig. 3(b). In this study, the predicted contact position in the X-coordinate is derived from the 2nd and 11th measured points according to Eq. (6). In Fig. 3(b), (x_{hp}, y_{hp}) represents the predicted contact position on the hitting plane, where is based on the setting of Joint-0 of the robotic arm, i.e., $y_{hp} = -42$.

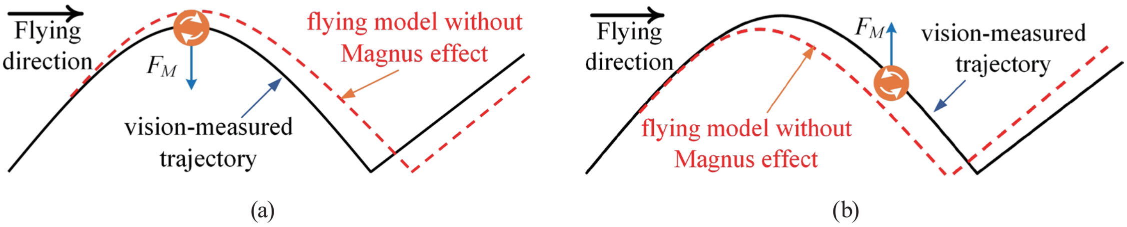
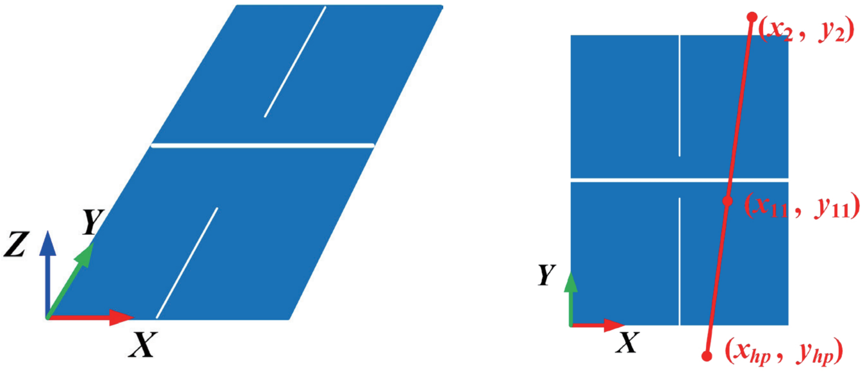


Fig. 2. (Color online) (a) Topspin and (b) backspin Magnus effects.



(a) (b)

Fig. 3. (Color online) (a) Coordinate system of robotic system and (b) ball trajectory on *X*–*Y* plane.

$$x_{hp} = x_{11} + \frac{x_{11} - x_2}{y_{11} - y_2}\left(y_{hp} - y_{11}\right) \tag{6}$$

As shown in Fig. 2 and Eq. (5), the contact position in the *Z*-coordinate is generally affected by factors such as the ball's flight velocity and acceleration. Therefore, the predicted contact position in the *Z*-coordinate is determined by multivariate regression as

$$z_{hp} = b + w_1 z_{11} + w_2 y_{11} + w_3 v_{z_{11}} + w_4 v_{y_{11}} + w_5 \frac{a_{z_{11}}}{v_{y_{11}}}. \tag{7}$$

Here, b is the bias and w_i for i = 1, 2, ..., 5 is the weight of the corresponding variable. The subscript 11 indicates the value of the corresponding variable at the 11th timestamp.

3.3 Hitting time prediction

For the table tennis robot to successfully strike the topspinning/backspinning ball, it should predict correctly not only the contact point, but also the hitting time. Likewise, the above multivariate regression is also used to predict the hitting time as

$$T_h = c + w_6 z_{11} + w_7 y_{11} + w_8 v_{z_{11}} + w_9 v_{y_{11}} + w_{10} \frac{a_{z_{11}}}{v_{y_{11}}}. \tag{8}$$

Here, c is the bias and w_i for i = 6, 7, ..., 10, is the weight. T_h is the time required for the ball to fly from the position at the 11th timestamp to the predicted contact point.

4. Control Design for Robotic Arm

In this study, the control design for the robotic arm is divided into three parts, namely, posture control, racket tilt angle control, and swing control.

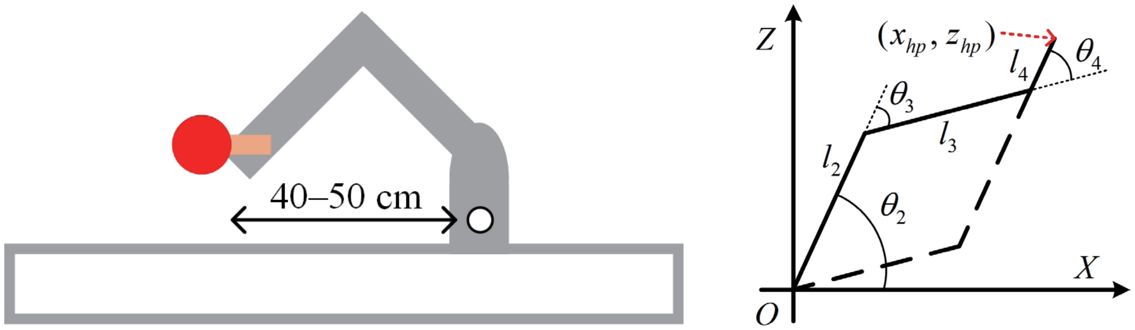
4.1 Posture control for robotic arm

The posture control of the robotic arm is mainly realized by Joint-0 and Joints-2–4. Joint-0 can ensure that the robotic arm has an appropriate swing radius when striking the ball. If the swing radius is very large, the motor should drive more torque, which also causes the robotic arm to swing slower. If the swing radius is very small, the force of the strike will be too small to return the ball over the net. According to our experimental results, 40–50 cm would be an appropriate swing radius, as shown in Fig. 4(a).

Then, the posture of the robotic arm on the hitting plane can be solved through inverse kinematics. Figure 4(b) illustrates the geometric relationship between the posture of the robotic arm and the predicted contact point (x_{hp}, z_{hp}) on the hitting plane. In Fig. 4(b), the origin is located at the center of Joint-2. l_2, l_3, and l_4 are link lengths of Joints-2–4, respectively. θ_2, θ_3, and θ_4 are the corresponding angles of Joints-2–4, respectively.

4.2 Racket tilt angle control

When the spinning ball hits the racket, the spin creates additional force F_s. Owing to the combination of rebound force F_b and rotation-induced force F_s, the actual rebound direction of the returned ball is different from the direction of the rebound force. Figures 5(a) and 5(b) illustrate the topspinning and backspinning cases, respectively. Therefore, when striking a topspinning ball without tilting the racket face, it is easy for the returned ball to fly out of bounds. Consequently, when striking a topspinning ball, tilting the racket face forward and downward appropriately can prevent the returned ball from flying out of bounds, as illustrated in Fig. 5(c). Similarly, tilting the racket face backward and upward appropriately can avoid



(a) (b)

Fig. 4. (Color online) (a) Appropriate swing radius and (b) robotic structure definition.

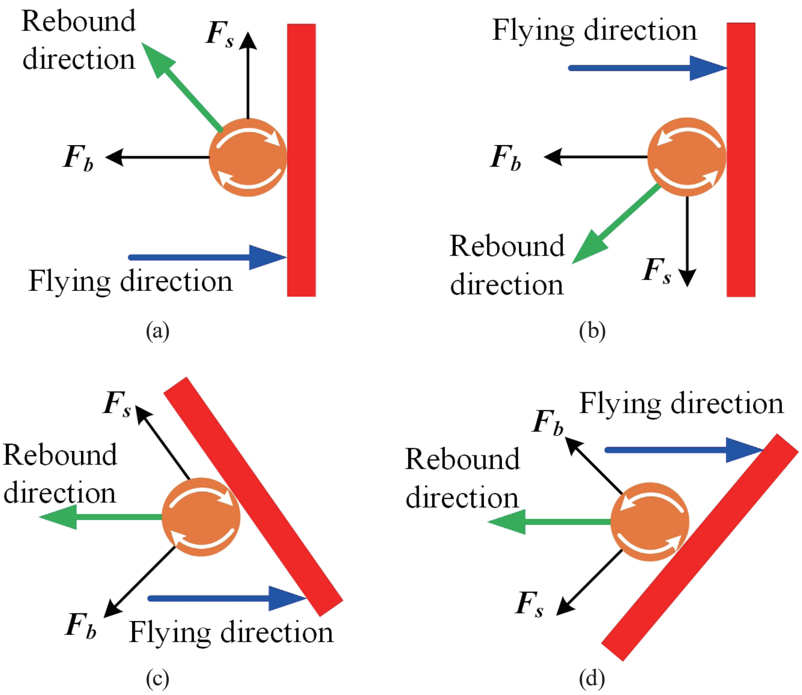


Fig. 5. (Color online) (a) Striking back directions with topspin, (b) striking back directions with backspin, (c) tilting the racket for hitting the topspinning ball back, and (d) tilting the racket for hitting the backspinning ball back.

returning the ball to the net, as illustrated in Fig. 5(d). According to Eq. (5), the slower the ball flies, which implies the greater spinning strength, the more tilt the racket should have. In addition to tilting the racket face, the height of the contact point should also be considered when striking the spinning balls. When striking a backspinning ball, the higher the contact point, the less tilt the racket should have. Conversely, when striking a topspinning ball, the higher the contact point, the greater the tilt required for the racket. Accordingly, the fuzzy controller for controlling the tilt of the racket is designed on the basis of the contact position z_{hp} and a_z/v_y in Eq. (5). The fuzzy sets of these two input (antecedent) variables are shown in Figs. 6(a) and 6(b), respectively. The fuzzy sets of the output (consequent) variable, which indicate the tilt angle of the racket through Joint-5 for striking topspinning or backspinning balls, are shown in Figs. 6(c) and 6(d), respectively.

The rule table of the fuzzy controller design for striking the topspinning balls is shown in Table 1. The fuzzy rule table for striking the backspinning balls is shown in Table 2. Finally, the center of gravity defuzzification is used to obtain the fuzzy control final output.

4.3 Swing control for robotic arm

Table tennis players turn their shoulders and swing their arms to strike the balls. In this study, the robotic arm swing is realized by controlling Joint-1 in trapezoidal point-to-point

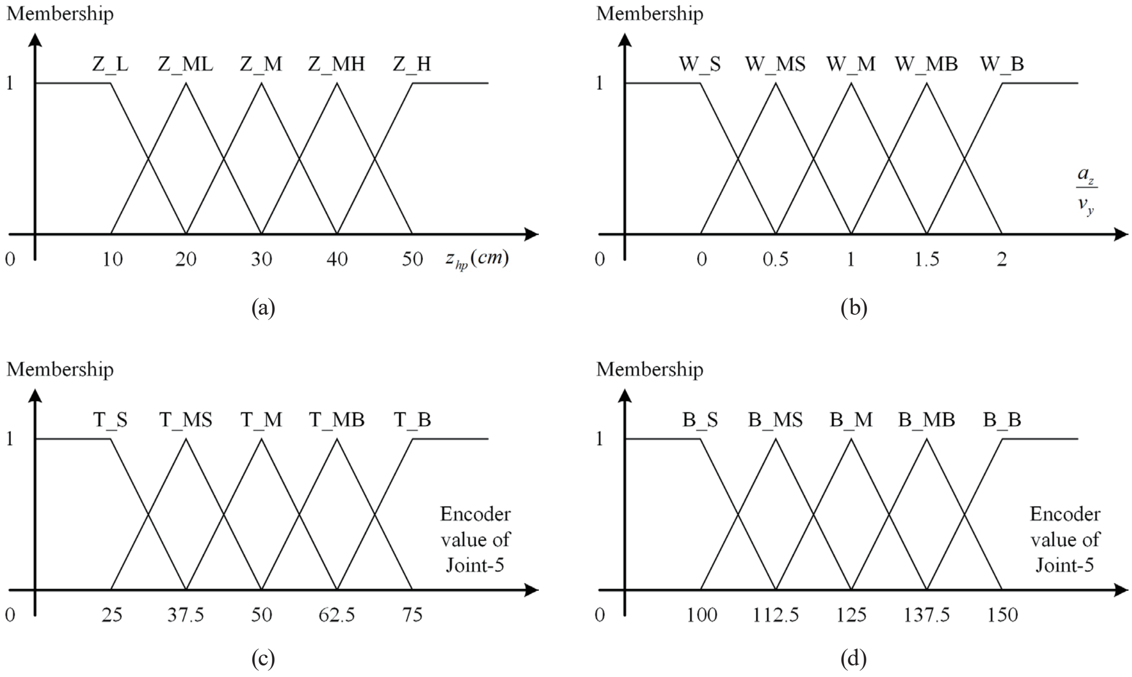


Fig. 6. (Color online) (a) Fuzzy sets of input variable 1: z_{hp}; (b) fuzzy sets of input variable 2: a_z/v_y; (c) fuzzy sets of output variable: for topspin; (d) fuzzy sets of output variable: for backspin.

Table 1
Fuzzy rule table for striking topspinning balls.

z_{hp}	a_z/v_y				
	W_S	**W_MS**	**W_M**	**W_MB**	**W_B**
Z_L	T_S	T_S	T_MS	T_MS	T_M
Z_ML	T_S	T_MS	T_MS	T_M	T_MB
Z_M	T_MS	T_M	T_M	T_M	T_MB
Z_MH	T_M	T_M	T_M	T_MB	T_MB
Z_H	T_M	T_MB	T_MB	T_MB	T_B

Table 2
Fuzzy rule table for striking backspinning balls.

z_{hp}	a_z/v_y				
	W_S	**W_MS**	**W_M**	**W_MB**	**W_B**
Z_L	B_B	B_MB	B_MB	B_MB	B_M
Z_ML	B_MB	B_MB	B_M	B_M	B_M
Z_M	B_MB	B_MB	B_M	B_M	B_MS
Z_MH	B_MB	B_M	B_M	B_MS	B_S
Z_H	B_M	B_MS	B_MS	B_S	B_S

control mode. The trapezoidal point-to-point control mode includes acceleration, constant speed, and deceleration phases. The total area of the trapezoid is the angle through which Joint-1 swings. In this study, when preparing to strike the ball, the robotic arm is set at an angle of 30° behind the striking plane, and the total angle of the robotic arm swing is 120°.

5. Experimental Results

In this section, we showcase the effectiveness of the proposed method through three distinct experiments. These experiments aim to verify the accuracy of z_{hp} for topspinning/backspinning balls, the accuracy of hitting time prediction, and the performance of striking the ball. It should be emphasized that spin type classification is a necessary prerequisite.

5.1 Accuracy of contact position prediction

Following the classification process, we utilized 364 data points for regression analysis training while reserving 110 data points for testing. The trained prediction model of the predicted contact point in the *Z*-coordinate for topspinning balls is denoted by Eq. (9).

$$z_{hp}^{t} = 34.3083 + 0.256 z_{11} - 0.016 y_{11} + 0.028 v_{z_{11}} - 0.007 v_{y_{11}} - 0.379 \frac{a_{z_{11}}}{v_{y_{11}}} \tag{9}$$

Moving on to the analysis of backspinning balls, we utilized a total of 369 data points for the regression analysis training and 132 data points for testing. The trained prediction model of the predicted contact point in the *Z*-coordinate for backspinning balls is denoted by Eq. (10).

$$z_{hp}^{b} = 20.423 + 0.457 z_{11} - 0.066 y_{11} + 0.058 v_{z_{11}} - 0.003 v_{y_{11}} - 0.26 \frac{a_{z_{11}}}{v_{y_{11}}} \tag{10}$$

In Table 3, we have compiled an overall summary of the average errors and standard deviation for our model's training and testing phases.

5.2 Accuracy of hitting time prediction

In the context of the topspin analysis, 347 data points were utilized for regression analysis training, whereas 129 data points were set aside for testing. The resulting formula for the prediction of hitting time is presented below.

$$T_{h}^{t} = 134.079 + 0.261 z_{11} + 2.148 y_{11} + 0.006 v_{z_{11}} - 0.162 v_{y_{11}} - 0.334 \frac{a_{z_{11}}}{v_{y_{11}}} \tag{11}$$

Table 3
Accuracy of contact point for topspinning/backspinning balls.

	Phase	Average error (cm)	Standard deviation (cm)
Topspinning ball	Training	2.6079	3.2872
	Testing	2.7724	3.2875
Backspinning ball	Training	2.5160	3.2102
	Testing	3.0552	3.6970

In this study, we employed 363 data points for training and 136 for testing the backspin case in the context of regression analysis. The model was subsequently derived and formulated as in Eq. (12), and numerical results are tabulated in Table 4.

$$T_h^b = 203.368 - 1.082 z_{11} + 2.923 y_{11} + 0.06 v_{z_{11}} - 0.286 v_{y_{11}} + 0.124 \frac{a_{z_{11}}}{v_{y_{11}}}. \tag{12}$$

5.3 Experiments of striking performance

On the basis of the experimental results collected, it was observed that the contact point position had an average error of approximately 3 cm, whereas the average error in predicting the hitting time was less than 13.5 ms. To validate the robustness of returning balls, we conducted experiments using a ball launcher to launch topspinning/backspinning balls at three different angles. Results for the topspinning and backspinning balls are shown in Table 5. A successful return is defined as the robot being able to strike the ball to the opposite side of the table, while failed attempts include swinging the ball in the air, returning it to the net, or going out of bounds.

6. Discussion

To fully control a table tennis robot, an accurate visual system is required to track and predict the ball trajectory, along with a well-designed strategy for striking the ball back. Our experimentation has yielded successful results in programming the robotic arm to counterattack the topspinning or backspinning balls. What differentiates our study from others is the implementation of binocular vision to sense the necessary parameters for estimating the ball's

Table 4
Accuracy of hitting time for topspinning/backspinning balls.

	Phase	Average error (ms)	Standard deviation (ms)
Topspinning ball	Training	5.8810	7.5929
	Testing	6.432	8.2375
Backspinning ball	Training	10.5436	13.2606
	Testing	13.4743	15.6570

Table 5
Striking performance for topspinning/backspinning balls.

	Angle (°)	Successful return	Failed counterattack	Success rate (%)
Topspinning ball	11	39	11	78
	14.2	36	14	72
	17.4	44	6	88
	Total	119	31	79.3
Backspinning ball	8	19	31	38
	9.6	26	24	52
	11.2	22	28	44
	Total	67	83	44.7

rotation degree, which is directly proportional to its rotation velocity, by utilizing Eq. (5). Subsequently, we designed a fuzzy controller to adjust the racket's facing angle.

The following discussion was centered on two fundamental performance indicators: the prediction error of the contact position and the batting success rate. Our analysis revealed a prediction error of approximately 3 cm for the contact position, which is a slight improvement from Ref. 2's result of 3.6 cm. In terms of batting success rate, our performance was similar to that shown in Ref. 3, but the proposed method can hit back the topspinning/backspinning balls.

7. Conclusions

Through our research, we developed a fuzzy controller that allows a robotic arm to strike topspinning/backspinning balls. Our proposed method was found to be effective in striking the ball back, with small average errors in contact point position and hitting time prediction. However, we encountered certain difficulties in hitting the backspinning ball, with a success rate of only 44.7%, while the topspinning ball had a success rate of 79%. To address this issue, we conducted a thorough analysis of the reasons for the counterattack failure and found that improving the accuracy of prediction errors remains a key challenge. To enhance the performance, it is essential to utilize a camera with a higher capturing rate as this would be beneficial in reducing the error in predicting the contact point position and hitting time. In addition, controlling the face angle of the racket is also crucial in improving success rates.

Acknowledgments

This research was funded by the National Science and Technology Council, Taiwan, grant number NSTC 112-2221-E-992-064-MY2.

References

1 R. L. Anderson: A Robot Ping-Pong Player: Experiment in Real-Time Intelligent Control (MIT Press, Cambridge, MA, 1988) 1st ed.

2 H. I. Lin, Z. Yu, and Y. C. Huang: Sensors **20** (2020) 333. https://doi.org/10.3390/s20020333

3 L. Acosta, J. J. Rodrigo, J. A. Mendez, G. N. Marichal, and M. Sigut: IEEE Rob. Autom. Mag. **10** (2003) 44. https://doi.org/10.1109/MRA.2003.1256297

4 M. Matsushima, T. Hashimoto, M. Takeuchi, and F. Miyazaki: IEEE Trans. Rob. **21** (2005) 767. https://doi.org/10.1109/TRO.2005.844689

5 K. P. Modi, F. Sahin, and E. Saber: Proc. 2005 IEEE Int. Conf. Systems, Man and Cybernetics (IEEE, 2005) 1831.

6 Z. Yu, Y. Liu, Q. Huang, X. Chen, W. Zhang, J. Li, G. Ma, L. Meng, T. Li, and W. Zhang: Proc. 2013 IEEE Int. Conf. Robotics and Biomimetics (IEEE, 2013) 911.

7 Z. T. Zhang, D. Xu, and M. Tan: IEEE Trans. Instrum. Meas. **59** (2010) 3195. https://doi.org/10.1109/TIM.2010.2047128

8 Q. Z. Wang and Z. Y. Sun: Proc. 2015 IEEE Cyber Technology in Automation, Control, and Intelligent Systems Conf. (IEEE, 2015) 551.

9 Y. P. Gao, J. Tebbe, J. Krisme, and A. Zell: Proc. 2019 Third IEEE Int. Conf. Robotic Computing (IEEE, 2019) 189.

10 P. Blank, J. Hoßbach, D. Schuldhaus, and B. M. Eskofier: Proc. 2015 ACM Int. Symp. Wearable Computers (ACM, 2015) 93.

11 G. D. Chen, D. Xu, Z. J. Fang, Z. M. Jiang, and M. Tan: IEEE Trans. Instrum. Meas. **62** (2013) 2901. https://doi.org/10.1109/TIM.2013.2265471

12 S. Furuno, K. Kobayashi, T. Okubo, and Y. Kurihara: Proc. 2009 IEEE ICROS-SICE Int. Joint Conf. (ICROS-SICE, 2009) 3439.

13 C. F. Liu, Y. Hayakawa, and A. Nakashima: Proc. 2011 IEEE Robotics and Biomimetics Conf. (IEEE, 2011) 2270.

14 Y. F. Zhang, R. Xiong, Y. S. Zhao, and J. G. Wang: IEEE Trans. Instrum. Meas. **64** (2015) 2280. https://doi.org/10.1109/TIM.2014.2385173

15 Camera Calibration Toolbox for Matlab: https://data.caltech.edu/records/jx9cx-fdh55 (accessed July 2023).