International Journal of Innovative
Computing, Information and Control
Volume **9**, Number **10**, October **2013**

ICIC International ©2013 ISSN 1349-4198
pp. 4181–4192

A PARALLEL FUZZY LEARNING APPROACH TO DETERMINE THE HITTING POINT FOR PING-PONG PLAYING ROBOT

Yanlong Huang, De Xu and Min Tan

State Key Laboratory of Management and Control for Complex Systems
Institute of Automation, Chinese Academy of Sciences
No. 95, Zhongguancun East Road, Beijing 100190, P. R. China
{ yanlong.huang; de.xu; min.tan }@ia.ac.cn

Received October 2012; revised February 2013

Abstract. *This paper concentrates on determining the hitting point where the racket attached to the ping-pong playing robot will intercept the incoming ball. The nearest neighbor method is used to estimate the racket velocity at the hitting point; then the racket acceleration during the hitting process is analyzed. A parallel fuzzy learning system consisting of two fuzzy subsystems is used to compute the success rate for the racket acceleration. Both these subsystems will be updated online based on the feedback. A performance function of the acceleration and the success rate is formulated to evaluate the candidate hitting points obtained by trajectory prediction; then the optimal hitting point is chosen. In comparison with the virtual hitting plane method, the proposed method shows better performance.*

Keywords: Ping-pong playing robot, Hitting point, Nearest neighbor method, Parallel fuzzy learning

1. **Introduction.** In recent years, there are many researchers who focus on various aspects of the ping-pong playing robot [1-4,6,7]. Anderson [1] studied the prototype of the ping-pong playing robot, where the ping-pong physics, vision system and real time expert system were discussed. Zhang et al. [2] analyzed the ball's flying model and rebound model. A distributed stereo vision system for ball detection was developed therein. Acosta et al. [3] built a monocular vision system to estimate the ball position and designed an expert module to define the game strategy. Matsushima et al. [4] used the locally weighed regression [5] approach to return the incoming ball to a desired position. In the framework of motor primitives, Muelling et al. [6] and Kober et al. [7] studied the motor skill learning for hitting the incoming ball.

Trajectory prediction is an important issue for the ping-pong playing robot because it is capable of guiding the robot to hit the incoming ball [2,8]. The goal of trajectory prediction is to estimate the hitting position and the hitting time at which the robot intercepts the incoming ball. Meanwhile, the ball velocity at the hitting time is important as well. As soon as the information about the hitting point is determined, the robot begins to return the ball. In general, the ball after rebound will continue to fly; thereby each point on the ball trajectory after rebound can be seen as the hitting point. Here, the hitting point includes the hitting position, the hitting time and the ball velocity. Muelling et al. [9,10] defined the hitting point as the intersection point of the ball trajectory and the virtual hitting plane. A virtual hitting plane that is parallel to the table plane was used in [2]. The virtual hitting plane is simple and easy to understand. However, it does not sufficiently concern the states of the flying ball and the capability limits of the robot.

Human players are able to predict the hitting point intuitively and improve their abilities by trial and error. As analogous to human perception and reasoning, fuzzy sets and fuzzy logic have been studied widely [11,12]. A more common framework of fuzzy logic in the practical applications is the fuzzy if-then rules [13]. The determination of the hitting point is in essence a multivariate decision-making problem. If each variable associated with the determination of the hitting point is characterized with more fuzzy sets, a large number of trials are required to train the collection of fuzzy rules. On the other hand, if each variable is characterized with less fuzzy sets, the few fuzzy rules cannot make full use of the learning information due to rough granules of variables. Inspired by the work of Karakose and Akin [14], a parallel fuzzy learning approach consisting of two fuzzy subsystems is applied to the dynamic multivariate learning process. One fuzzy subsystem is characterized with less fuzzy rules while the other one is with more fuzzy rules. When only a few trials are carried out, the subsystem with less fuzzy rules is able to roughly learn these trails. As the number of trials increases, the other subsystem with more fuzzy rules will be trained sufficiently.

In this paper, we focus on the determination of the hitting point for the ping-pong playing robot. First, we can obtain a series of candidate hitting points by trajectory prediction [2,8]. The racket velocity for each hitting point is estimated using the nearest neighbor (N-N) method [5,15]. Then, the racket acceleration for each hitting point is analyzed based on the racket velocity, the hitting position and the hitting time. The proposed parallel learning system is used to decide the success rate for the racket acceleration. Each hitting point is evaluated by a performance function whose inputs are the acceleration and the corresponding success rate. Finally, the optimal hitting point is chosen from all the candidate hitting points. The feedback regulation is introduced in the robotic system as well, which is used for updating online the parallel learning system. Since both the capability limits of the robot and the states of the flying ball are considered, a more appropriate hitting point is expected to be obtained. Moreover, the parallel fuzzy learning system is adequate for the dynamical learning process. As the experiment goes on, the parallel learning system with online parameters update will have better overall performance.

This paper is organized as follows. In Section 2, we will simply describe the coordinate system used in our robotic system. In Section 3, we explain trajectory prediction and the principle of the learning system. The analysis of racket acceleration is presented in Section 4. Then the parallel fuzzy learning approach is studied in Section 5. Experiments are carried out to demonstrate the effectiveness of the proposed method in Section 6. Finally, we will summarize our approach in Section 7.

2. **Coordinate System.** In our robotic system, the racket is attached to a three-degree-of- freedom (DoF) mechanism that consists of three linear axes mounted on the table; a two DoFs pan-tilt unit provides the orientation for the racket [16]. If a hitting point is provided for the robot, the three DoFs mechanism moves the racket from the initial position to the hitting position, so that the racket is able to intercept the incoming ball with a certain velocity at the hitting time; meanwhile, the two DoFs units ensure the proper orientation of the racket. Since the robot performance depends mainly on the three DoFs mechanism, we will focus on the racket velocity and acceleration that reflect the movement characteristics of this mechanism. As shown in Figure 1, {R} is the reference coordinate system. The X-axis is parallel to the short side of the table; the Z-axis is perpendicular to the table plane. For the convenience of discussion, some notations are defined. The initial position of the racket is denoted as $P_0 = (x_0, y_0, z_0)$. The hitting position and the desired landing position are denoted as $P_h = (x_h, y_h, z_h)$ and $P_d = (x_d,$

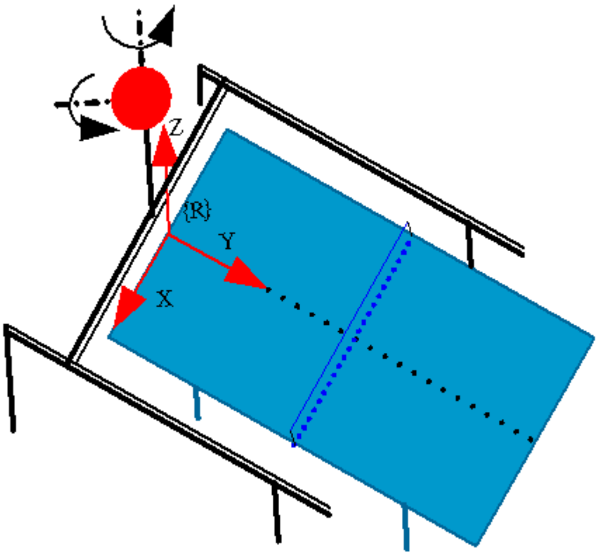


FIGURE 1. Coordinate system of the ping-pong robotic system

y_d, z_d), respectively. The hitting time is t_h. The racket velocity and acceleration are denoted as $V = (v_x, v_y, v_z)$ and $\alpha = (a_x, a_y, a_z)$, respectively. The ball velocity just before the impact of the racket is denoted as $V_i = (v_{xi}, v_{yi}, v_{zi})$. Furthermore, we use $D_d = (d_{xd}, d_{yd}, d_{zd}) = (x_d - x_h, y_d - y_h, z_d - z_h)$ to represent the desired flying distance, and use $V_h = (v_{xh}, v_{yh}, v_{zh}) = (v_x(t_h), v_y(t_h), v_z(t_h))$ to represent the hitting velocity.

3. **Description of the Learning System.** When a ping-pong ball is flying from the human opponent to the robot, we can measure its 3-dimensional (3-D) positions using the stereo vision system. After a series of 3-D positions are collected as the initial flying trajectory, we can predict its subsequent fly trajectory using the flying model [2] and the rebound model [8]. Accordingly, these points on the predicted rebound trajectory can be seen as the candidate hitting points. The information of each candidate point includes the hitting position P_h, the hitting time t_h and the ball velocity V_i. Here, the self-rotational velocity is not considered. As shown in Figure 2, the blue circles represent the initial flying trajectory and the red circles represent the candidate hitting points.

It is assumed that the desired landing position P_d is given. Then, the desired flying distance D_d for each candidate point is obtained. According to D_d and V_i for each candidate point, the hitting velocity V_h is estimated by the N-N method [5,15]. Then, we can obtain the racket's approximate acceleration $\alpha_r = (a_{xr}, a_{yr}, a_{zr})$, which is relevant to P_h, t_h and V_h. It is noted that, each component of α_r represents a different robot

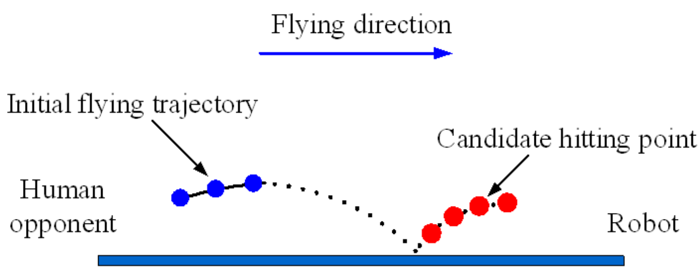


FIGURE 2. Trajectory prediction of the flying ball

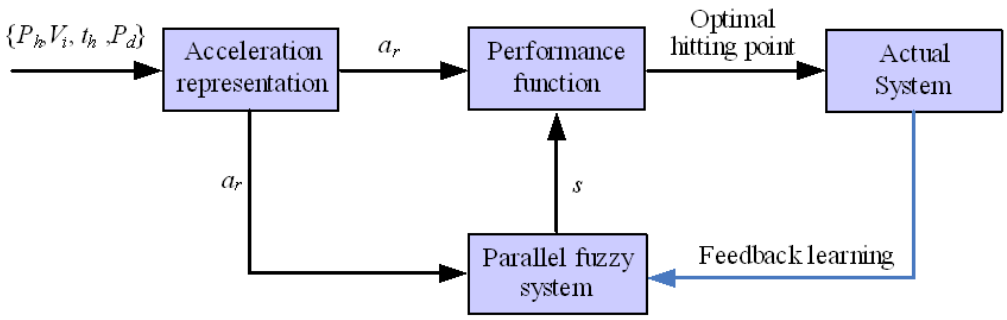


Figure 3. Architecture of the learning system

joint, and each joint has different influence on the robot performance. In other words, the importance of the variables a_{xr}, a_{yr} and a_{zr} are different for robot performance. Thus, the success rate s of returning the incoming ball is required for α_r. The rate s represents the overall performance of robot; it can be seen as a function of a_{xr}, a_{yr} and a_{zr}. The parallel fuzzy system consisting of two fuzzy subsystems determines the success rate s for α_r. To take the parameters α_r and s as the inputs, the performance function is used for evaluating the candidate points. The candidate point at which the performance function is maximal is seen as the optimal hitting point. After the optimal point is obtained, the robot begins to return the incoming ball. Based on the actual performance of the robot, the parallel fuzzy system is updated online so that the relationship between α_r and s is better represented. The architecture of the whole learning system is illustrated in Figure 3.

4. Representation of the Racket Acceleration.

4.1. **Hitting velocity.** In order to analyze the racket acceleration a, we need to obtain the hitting velocity V_h. Due to the redundant DoFs, our robot returns the incoming ball along the Y-axis at the hitting time, i.e., $v_{xh} = v_{zh} = 0$ [17]. Usually, the hitting velocity V_h is determined by D_d and V_i. Here, the racket orientation is not involved for simplicity. We use the N-N method to estimate V_h. The N-N method represents the relationship between (D_d, V_i) and V_h. The experience data are stored as shown in Figure 4. Here, n is the number of the stored data points.

1	d_{xd}^1	d_{yd}^1	d_{zd}^1	v_{xi}^1	v_{yi}^1	v_{zi}^1	v_{yh}^1
2	d_{xd}^2	d_{yd}^2	d_{zd}^2	v_{xi}^2	v_{yi}^2	v_{zi}^2	v_{yh}^2
$\vdots$				$\vdots$			
n	d_{xd}^n	d_{yd}^n	d_{zd}^n	v_{xi}^n	v_{yi}^n	v_{zi}^n	v_{yh}^n

Figure 4. Experience data (D_d, V_i) and V_h

For each candidate hitting point, we need to calculate D_d, and then combine D_d and V_i into a new vector. The N-N method is to search all the stored data points, so that the index $k \in \{1, 2, \cdots n\}$ that satisfies (1) is found. Then, v_{yh}^k is seen as the estimated hitting velocity v_{yh} for D_d and V_i:

$$\begin{aligned} & dis[(d_{xd}, d_{yd}, d_{zd}, v_{xi}, v_{yi}, v_{zi}), (d_{xd}^k, d_{yd}^k, d_{zd}^k, v_{xi}^k, v_{yi}^k, v_{zi}^k)] \\ \leq\, & dis[(d_{xd}, d_{yd}, d_{zd}, v_{xi}, v_{yi}, v_{zi}), (d_{xd}^q, d_{yd}^q, d_{zd}^q, v_{xi}^q, v_{yi}^q, v_{zi}^q)], \quad \forall q \in \{1, 2, \cdots, n\} \end{aligned} \tag{1}$$

where $dis(\cdot)$ represents the Euclidean distance.

4.2. **Representation of acceleration.** In the real game, the racket is expected to reach the hitting position P_h at the hitting time t_h, so that the incoming ball is returned. In other words, the racket velocity V should satisfy the following equations:

$$\int_0^{t_h} v_y(t)dt = y_h - y_0 \tag{2}$$

$$\int_0^{t_h} v_x(t)dt = x_h - x_0 \tag{3}$$

$$\int_0^{t_h} v_z(t)dt = z_h - z_0 \tag{4}$$

The discrete form of (2) is

$$[v_y(0) + v_y(1) + v_y(2) + \cdots + v_y(n-1)]T = y_h - y_0 \tag{5}$$

where T is the sampling time interval and satisfies $nT = t_h$.

The expanded form of (5) is

$$\begin{aligned} & [v_y(0) + (v_y(0) + a_y(0)T) + (v_y(0) + a_y(0)T + a_y(1)T) + \cdots \\ & + (v_y(0) + a_y(0)T + a_y(1)T + \cdots + a_y(n-2)T)]T \\ = & [nv_y(0) + (n-1)a_y(0)T + (n-2)a_y(1)T + \cdots + a_y(n-2)T]T = y_h - y_0 \end{aligned} \tag{6}$$

Here, $v_y(0) = 0$. Denote the minimum and maximum values of the set $\{a_y(0), a_y(1), \cdots, a_y(n-2)\}$ as $a_{y\min}$ and $a_{y\max}$, respectively. Then, we have

$$\frac{a_{y\min}(n-1)nT^2}{2} \leq y_h - y_0 \leq \frac{a_{y\max}(n-1)nT^2}{2} \tag{7}$$

Thus,

$$a_{y\min} \leq \frac{2(y_h - y_0)}{t_h^2} \leq a_{y\max} \tag{8}$$

Similarly,

$$a_{x\min} \leq \frac{2(x_h - x_0)}{t_h^2} \leq a_{x\max} \tag{9}$$

$$a_{z\min} \leq \frac{2(z_h - z_0)}{t_h^2} \leq a_{z\max} \tag{10}$$

It is noted that the hitting velocity $V_h = (0, v_{yh}, 0)$, a_y should satisfy the additional condition:

$$\int_0^{t_h} a_y(t)dt = v_{yh} \tag{11}$$

It is reasonable to assume that $a_y(i) > 0$, $i = 0, 1, 2, \cdots, \bar{n} - 1$ and $a_y(i) = 0$, $i = \bar{n}, \bar{n} + 1, \cdots, n-2$. Then, the discrete form of (11) is

$$[a_y(0) + a_y(1) + \cdots + a_y(\bar{n} - 1)]T = v_{yh} \tag{12}$$

Denote the minimum and maximum values of the set $\{a_y(0), a_y(1), \cdots, a_y(\bar{n}-1)\}$ as a_{yl} and a_{yh}. We have

$$a_{yl} \cdot \bar{n}T \leq v_{yh} \leq a_{yh} \cdot \bar{n}T \tag{13}$$

Thus,

$$a_{yl}^2 \leq \frac{v_{yh}^2}{\bar{t}_h^2} \leq a_{yh}^2 \tag{14}$$

where $\bar{t}_h = \bar{n}T$.

Let us reconsider (6); it is rewritten as

$$\begin{aligned} &[a_y(0)T + (a_y(0)T + a_y(1)T) + \cdots + (a_y(0)T + a_y(1)T + \cdots + a_y(\bar{n}-1)T)]T \\ =& [\bar{n}a_y(0)T + (\bar{n}-1)a_y(1)T + \cdots + a_y(\bar{n}-1)T]T = \Delta y \leq y_h - y_0 \end{aligned} \tag{15}$$

Then,

$$a_{yl} \leq \frac{2\Delta y}{\bar{t}_h^2} \leq a_{yh} \tag{16}$$

By combing (14) and (16), we have

$$\frac{v_{yh}^2}{2(y_h - y_0)} \leq \frac{v_{yh}^2}{2\Delta y} \leq \frac{a_{yh}^2}{a_{yl}} \tag{17}$$

Because a_y usually varies in a small range, a_{yh}/a_{yl} is roughly seen as 1. In this case, (17) is rewritten as

$$\frac{v_{yh}^2}{2(y_h - y_0)} \leq a_{yh} \tag{18}$$

According to (8)-(10) and (18), a_y, $|a_x|$ and $|a_z|$ are roughly represented by $\max\{2(y_h - y_0)/t_h^2, v_{yh}^2/(y_h - y_0)/2\}$, $2|x_h - x_0|/t_h^2$ and $2|z_h - z_0|/t_h^2$, respectively. In fact, we can obtain the racket's accurate acceleration for each candidate hitting point by the motion planning. Since each candidate point requires a new motion planning, the execution speed will slow down when there are many candidate points. Thus, we use the approximate representation of the racket acceleration for the purpose of reducing the time consuming.

5. Parallel Fuzzy Learning.

5.1. Parallel fuzzy system.

The parallel fuzzy learning system consists of the rough fuzzy subsystem (RFS) and the precise fuzzy subsystem (PFS). The RFS is characterized with less fuzzy rules while the PFS is characterized with more fuzzy rules. Denote $\max\{2(y_h - y_0)/t_h^2, v_{yh}^2/(y_h - y_0)/2\}$, $2|x_h - x_0|/t_h^2$ and $2|z_h - z_0|/t_h^2$ as a_{xr}, a_{yr} and a_{zr}, respectively. Assuming that the domains of interest for the variables a_{xr}, a_{yr} and a_{zr} are determined according to the experimental results. In the RFS, we use A_1^i, B_1^j and C_1^k to represent the fuzzy values for a_{xr}, a_{yr} and a_{zr}, respectively, where $i = 1, 2, \cdots, r_1$, $j = 1, 2, \cdots, p_1$, $k = 1, 2, \cdots, q_1$. In the PFS, we use A_2^l, B_2^m and C_2^n to represent the fuzzy values for a_{xr}, a_{yr} and a_{zr}, respectively, where $l = 1, 2, \cdots, r_2$, $m = 1, 2, \cdots, p_2$, $n = 1, 2, \cdots, q_2$. Here, $r_2 \geq r_1$, $p_2 \geq p_1$ and $q_2 \geq q_1$.

The f-th fuzzy rule in the RFS is defined as

IF a_{xr} is A_1^i and a_{yr} is B_1^j and a_{zr} is C_1^k, **Then** s is s_{1f}

where $0 \leq s_{1f} \leq 1$ is the success rate.

The f-th fuzzy rule in the PFS is defined as

IF a_{xr} is A_2^l and a_{yr} is B_2^m and a_{zr} is C_2^n, **Then** s is s_{2f}

where $0 \le s_{2f} \le 1$ is the success rate.

For an input (a_{xr}, a_{yr}, a_{zr}), the rule in the RFS is chosen as follows:
Find $i^* \in \{1, 2, \cdots, r_1\}$, $j^* \in \{1, 2, \cdots, p_1\}$ and $k^* \in \{1, 2, \cdots, q_1\}$ satisfying

$$\mu_{A_1^{i^*}}(a_{xr}) \ge \mu_{A_1^i}(a_{xr}), \ \ \forall i \in \{1, 2, \cdots, r_1\} \tag{19}$$

$$\mu_{B_1^{j^*}}(a_{yr}) \ge \mu_{B_1^j}(a_{yr}), \ \ \forall j \in \{1, 2, \cdots, p_1\} \tag{20}$$

$$\mu_{C_1^{k^*}}(a_{zr}) \ge \mu_{C_1^k}(a_{zr}), \ \ \forall k \in \{1, 2, \cdots, q_1\} \tag{21}$$

Then, the address f_1 of the selected rule in RFS is

$$f_1 = (i^* - 1)p_1 q_1 + (j^* - 1)q_1 + k^* \tag{22}$$

Similarly, $l^* \in \{1, 2, \cdots, r_2\}$, $m^* \in \{1, 2, \cdots, p_2\}$ and $n^* \in \{1, 2, \cdots, q_2\}$ are found. The address f_2 of the selected rule in PFS is

$$f_2 = (l^* - 1)p_2 q_2 + (m^* - 1)q_2 + n^* \tag{23}$$

After the rules in both the subsystems are chosen, the output s for the input (a_{xr}, a_{yr}, a_{zr}) is

$$s(a_{xr}, a_{yr}, a_{zr}) = \frac{m_1 s_{1f_1} + m_2 s_{2f_2}}{m_1 + m_2} \tag{24}$$

where

$$m_1 = \min\{\mu_{A_1^{i^*}}(a_{xr}), \mu_{B_1^{j^*}}(a_{yr}), \mu_{C_1^{k^*}}(a_{zr})\} \tag{25}$$

$$m_2 = \min\{\mu_{A_2^{l^*}}(a_{xr}), \mu_{B_2^{m^*}}(a_{yr}), \mu_{C_2^{n^*}}(a_{zr})\} \tag{26}$$

If the success rate s of the approximate acceleration α_r is determined, the performance function used for evaluating the candidate hitting point is given as

$$J = s(a_{xr}, a_{yr}, a_{zr})/(a_{xr} + a_{yr} + a_{zr}) \tag{27}$$

5.2. **Feedback regulation.** The candidate hitting points are obtained by trajectory prediction, and then the optimal point is chosen according to the evaluation function (27). Once the final hitting point is determined, the robot will plan to intercept the ball. The success rates s_{1f1} in RFS and s_{2f2} in PFS are updated online as the experiment goes on. If the robot returns the incoming ball so that the ball lands on a satisfactory region, the corresponding success rates in RFS and PFS will increase. Otherwise, the success rates will decrease.

The feedback regulation is formulated as

$$s_{1f_1} = s_{1f_1} + m_1 \cdot \alpha \cdot s_n \tag{28}$$

$$s_{2f_2} = s_{2f_2} + m_2 \cdot \beta \cdot s_n \tag{29}$$

where $\alpha > 0$ and $\beta > 0$ are constants. $s_n = 1$ or -1 indicates whether the ball lands on the desired region. The parameters s_{1f1} and s_{2f2} represent the overall success rates for the input (a_{xr}, a_{yr}, a_{zr}). It is noted that the racket's approximate acceleration (a_{xr}, a_{yr}, a_{zr}) is different from its actual acceleration (a_x, a_y, a_z). The success rate may not precisely represent the relationship between the robot performance and the actual acceleration, but it can be seen as an approximate measure of this relationship.

6. Experiments and Results.

6.1. **Experimental system.** A stereo vision system developed in [2] was used to detect the ball position, where two high speed cameras (250fps) were used. The computer used for predicting the hitting point had a 2.3 GHZ CPU and a 2 GB RAM. The physical structure of the ping-pong playing robot is shown in Figure 5.

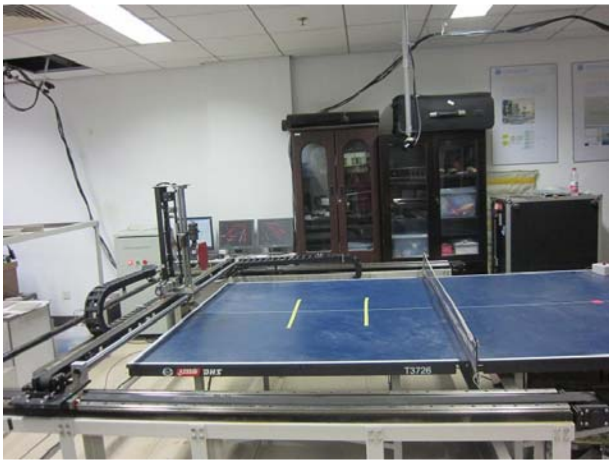


FIGURE 5. Robot physical structure

TABLE 1. The average performances of different methods

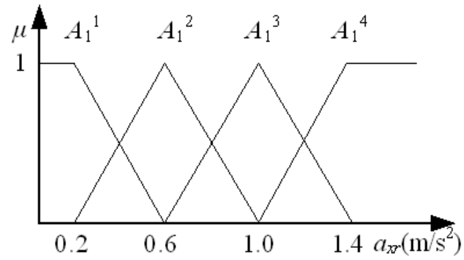
Proposed method	Method in [10]	Method in [2]
0.20	0.17	0.08

6.2. **Definitions of fuzzy subspaces.** The fuzzy sets A_1^i, B_1^j and C_1^k for a_{xr}, a_{yr} and a_{zr} in RFS are defined as shown in Figures 6(a)-6(c). A_2^l, B_2^m and C_2^n for a_{xr}, a_{yr} and a_{zr} in PFS are shown in Figures 6(d)-6(f).

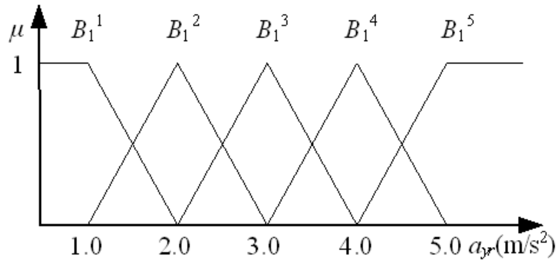
6.3. **Performance evaluation.** The initial position of the racket was (0, 0, 300) mm. The N-N method was previously trained using 2000 trails. The ball trajectory was predicted using the flying model [2] and the rebound model [8], where the self-rotational velocity was neglected. On the rebound trajectory we had predicted, a series of points with time interval 10 ms were chosen as the candidate hitting points. The initial success rates in RFS and PFS were set to 1. The parameters in feedback regulation were $\alpha = \beta = 0.1$. The desired landing position was $(0, 2250)$ mm. A region whose x-coordinate lay in $[-400, 400]$ mm and y-coordinate lay in [1850, 2650] mm was seen as the satisfactory region. The racket parameters were obtained by the locally weighed regression method in [16].

We had 350 trials to train the parallel fuzzy system. Then, 80 trails were carried out to evaluate our method using (27). The virtual hitting plane methods proposed in [2,10] were also used for comparison. In our experiment, the virtual plane in [10] was defined as $y_h = 450$ mm and the virtual plane in [2] was defined as $z_h = 220$ mm. Figure 7 shows the predicted hitting positions using the three methods. The results of performance evaluation are shown in Figure 8. It can be seen that the proposed method has the best performance overall. Table 1 shows the average performances of the three methods. Since the proposed method considers both the ball states and the robot capability, the predicted hitting point is much more flexible. Thus, the robot is able to return the incoming ball to a desired region with a higher probability.

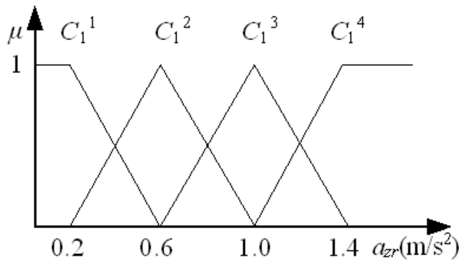
6.4. **Determination of the hitting point.** In this experiment, we had 450 trials to train the parallel fuzzy system. Then, another trail was carried out. x_h, y_h, z_h, v_{xi}, v_{yi}, v_{zi}, t_h, a_{xr}, a_{yr}, a_{zr}, s and J for each candidate hitting point were stored. Figure 9 shows the curves of these parameters. The optimal point is the one at which the performance function is the largest. As shown in Figure 9(b), the optimal point selected from the



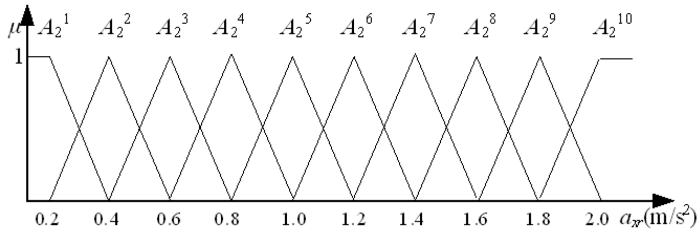
(a) RFS: Fuzzy values of a_{xr}



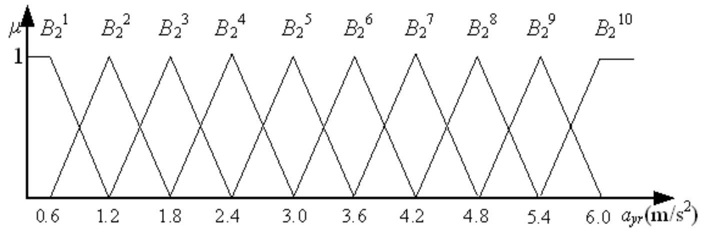
(b) RFS: Fuzzy values of a_{yr}



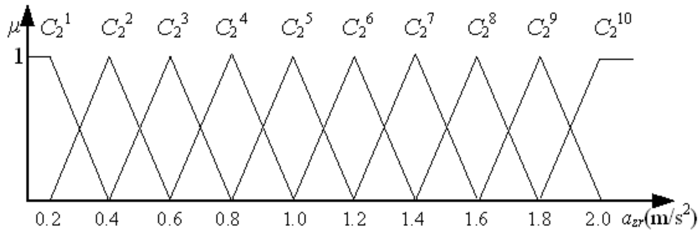
(c) RFS: Fuzzy values of a_{zr}



(d) PFS: Fuzzy values of a_{xr}



(e) PFS: Fuzzy values of a_{yr}



(f) PFS: Fuzzy values of a_{zr}

FIGURE 6. The fuzzy sets for a_{xr}, a_{yr} and a_{zr} in RFS and PFS, respectively

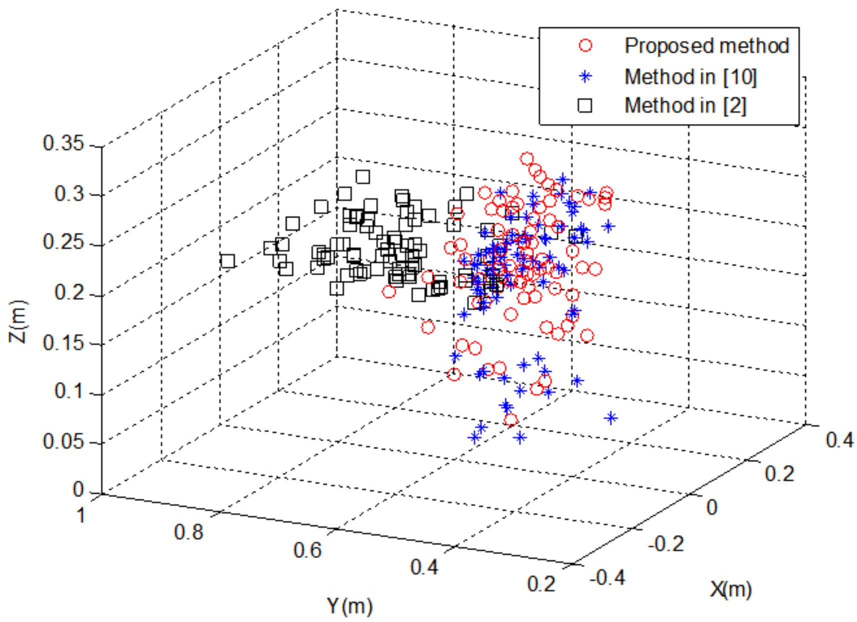


FIGURE 7. The predicted hitting points using different methods

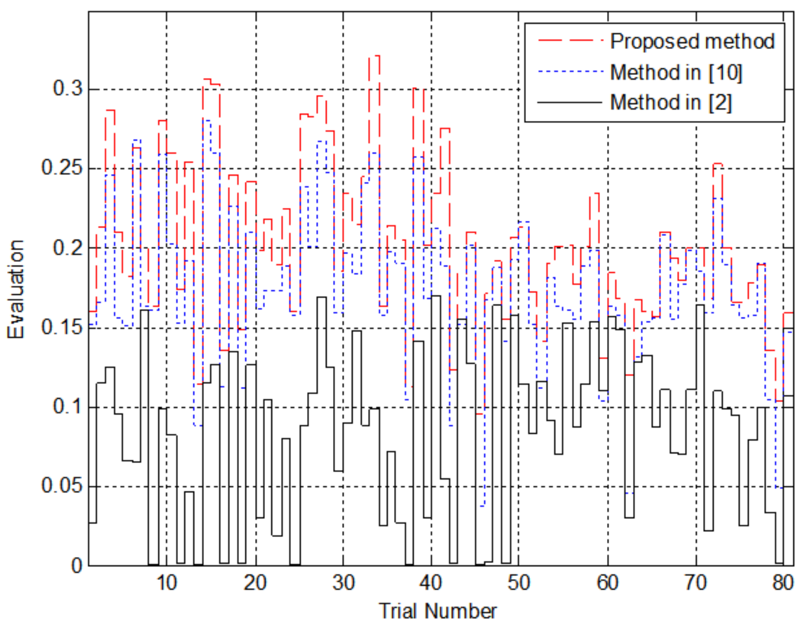
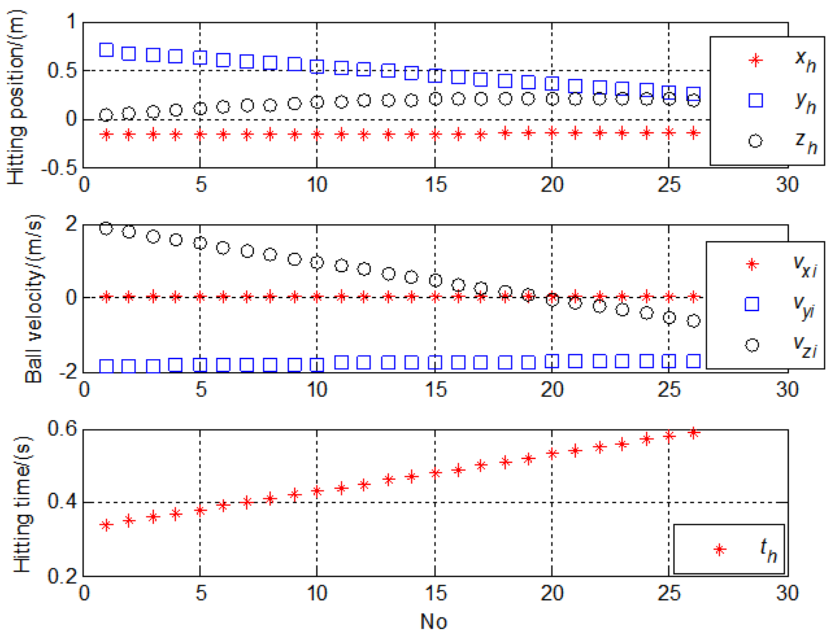


FIGURE 8. Performance evaluation of different methods

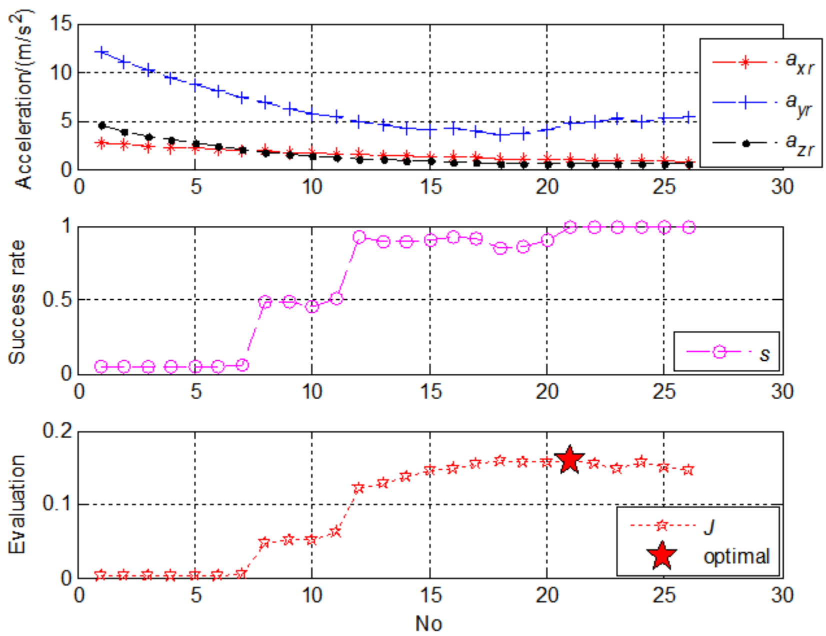
candidates hitting points has comparatively small acceleration and big success rate. Thus, the predicted hitting point obtained by the proposed method is more appropriate.

7. Conclusions.

A parallel fuzzy learning approach was proposed so as to determine the hitting point for the ping-pong playing robot. The parallel fuzzy system consisted of the rough and precise fuzzy subsystems. The rough fuzzy subsystem was characterized with less fuzzy rules and the precise fuzzy subsystem was with more fuzzy rules. Both the subsystems were updated online by the feedback learning. The parallel system is suitable



(a) Information of the candidate hitting points



(b) Approximate accelerations and success rates

FIGURE 9. The parameters used for determining the hitting point

for the dynamic learning process. The proposed method was able to sufficiently consider the ball states and the capabilities of the robot, and accordingly a more appropriate hitting point is obtained. The experimental results showed that the proposed method had better performance than the virtual hitting plane method.

Acknowledgment. This work was supported in part by the National Natural Science Foundation of China under Grant 61075035 and Grant 61273337.

REFERENCES

[1] R. L. Anderson, Dynamic sensing in a ping-pong playing robot, *IEEE Trans. on Robotics and Automation*, vol.5, no.6, pp.728-739, 1989.
[2] Z. Zhang, D. Xu and M. Tan, Visual measurement and prediction of ball trajectory for table tennis robot, *IEEE Trans. on Instrumentation and Measurement*, vol.59, no.12, pp.3195-3205, 2010.
[3] L. Acosta, J. J. Rodrigo, J. A. Méndez, G. N. Marichal and M. Sigut, Ping-pong player prototype, *IEEE Robotics & Automation Magazine*, vol.10, pp.44-52, 2003.
[4] M. Matsushima, T. Hashimoto, M. Takeuchi and F. Miyazaki, A learning approach to robotic table tennis, *IEEE Trans. on Robotics*, vol.21, no.4, pp.767-771, 2005.
[5] C. G. Atkeson, A. W. Moore and S. Schaal, Locally weighted learning, *Artificial Intelligence Review*, vol.11, pp.11-73, 1997.
[6] K. Muelling, J. Kober and J. Peters, Learning table tennis with a mixture of motor primitives, *Proc. of IEEE-RAS International Conference on Humanoid Robots*, Nashville, TN, USA, pp.411-416, 2010.
[7] J. Kober, E. Oztop and J. Peters, Reinforcement learning to adjust robot movements to new situations, *Robotics: Science and Systems*, pp.33-40, 2010.
[8] A. Nakashima, Y. Ogawa, Y. Kobayashi and Y. Hayakawa, Modeling of rebound phenomenon of a rigid ball with friction and elastic effects, *Proc of American Control Conference*, Baltimore, MD, USA, pp.1410-1415, 2010.
[9] K. Muelling and J. Peters, A computational model of human table tennis for robot application, *Autonomy Mobile System*, pp.57-64, 2009.
[10] K. Muelling, J. Kober and J. Peters, A biomimetic approach to robot table tennis, *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*, Taipei, pp.1921-1926, 2010.
[11] C. Z. Janikow, Fuzzy decision trees: Issues and methods, *IEEE Trans. on System, Man, and Cybernetics, Part B: Cybernetics*, vol.28, no.1, pp.1-14, 1998.
[12] L. Wang, The WM method completed: A flexible fuzzy system approach to data mining, *IEEE Trans. on Fuzzy System*, vol.11, no.6, pp.768-782, 2003.
[13] L. A. Zadeh, Toward human level machine intelligence – Is it achievable? The need for a paradigm shift, *IEEE Computational Intelligence Magazine*, pp.11-22, 2008.
[14] M. Karakose and E. Akin, Block based fuzzy controllers, *International Journal of Research and Reviews in Applied Sciences*, vol.3, no.1, pp.100-110, 2010.
[15] D. R. Wilson and T. R. Martinez, An integrated instance-based learning algorithm, *Computational Intelligence*, vol.16, no.1, pp.1-28, 2000.
[16] Y. Huang, D. Xu, M. Tan and H. Su, Adding active learning to LWR for ping-pong playing robot, *IEEE Trans. on Control Systems Technology*, 2012.
[17] P. Yang, D. Xu, H. Wang and Z. Zhang, Control system design for a 5-DOF table tennis robot, *Proc. of International Conference on Control, Automation, Robotics and Vision*, Singapore, pp.1731-1735, 2010.