Spin Observation and Trajectory Prediction of a Ping-Pong Ball

Yifeng Zhang[1], Yongsheng Zhao[1], Rong Xiong[1], Yue Wang[1], Jianguo Wang[2] and Jian Chu[1]

Abstract— **For ping-pong playing robots, observing a ball and predicting a ball's trajectory accurately in real-time is essential. However, most existing vision systems can only provide ball's position observation, and do not take into consideration the spin of the ball, which is very important in competitions. This paper proposes a way to observe and estimate ball's spin in real-time, and achieve an accurate prediction. Based on the fact that a spinning ball's motion can be separated into global movement and spinning respect to its center, we construct an integrated vision system to observe the two motions separately. With a pan-tilt vision system, the spinning motion is observed through recognizing the position of the brand on the ball and restoring the 3D pose of the ball. Then the spin state is estimated with the method of plane fitting on current and historical observations. With both position and spin information, accurate state estimation and trajectory prediction are realized via External Kalman Filter(EKF). Experimental results show the effectiveness and accuracy of the proposed method.**

I. INTRODUCTION

Ping-pong playing robots have attracted the attention of more and more researchers because it is an ideal platform for real-time hand-eye coordination experiments. Observing the ball and predicting ball's trajectory accurately in real-time mode is challenging but essential for successful playing. The main difficulties come from two points. One is that the ball travels very fast in a short distance. In professional competitions, the maximal ball speed can reach up to more than 20m/s, while the whole length of ping-pong table is only 2.73m. Another is there needs to be enough time left for the robot to react. The more time left, lower requirement on robot's mechanics, such as joint velocity and acceleration.

Various vision systems have been developed for ball observation. L. Acosta et al. developed a monocular vision system to locate the ball's 3D position based on detection of the ball and it's shadow, but the ball's flying area and velocity were highly restricted [1]. Y. Zhang et al. carried this method forward to a standard playing environment [2]. Recently, many stereo-vision [3][4][5][6] and multi-vision [7] systems have been developed and applied to ping-pong playing robots as they are more natural and can provide more accurate position information. However, current existing vision systems only observe the position of the ball, which greatly restricts the available information provided to trajectory analysis and prediction. In this paper, this type of vision system is called as position vision system.

Limited by the observation, literatures on trajectory analysis have mainly focused on position state estimation and prediction. M. Matsushima etc. estimated ball's state by fitting a polynomial and predicted following trajectory using two learned maps [3]. Z. Zhang et al. deduced the dynamic model of ping-pong ball based on forces analysis and used it in prediction, but the ball's state is estimated based on fitting a polynomial rather than using dynamic model [4]. K. Mulling et al. used EKF to get state estimation [8]. Y. Zhang et al. proposed the idea that the dynamic model is described in discrete and continuous forms for state estimation and trajectory prediction respectively and share the same parameters, and they gave a solution to learn and adapt parameters to improve the performance [5]. This method worked well and successfully supported the humanoid robots 'Wu' and 'Kong' continuously play with human players or with each other [1]. The algorithms mentioned above are good at predicting trajectory of ball without spin, but is less successful for a spinning ball.

Utilizing spin is one of the most important skills for human ping-pong players. As stated in [9], adding spin to the ball is very popular in modern ping-pong competitions, because the spin can vastly deviate the trajectory and make the trajectory hard to be estimated, even for professional players. Researchers have put much effort into analysis of how spinning will affect flight trajectory. Some experiments were carried out to measure the lift force (also known as Magnus force) and the lift coefficient for different shapes under different moving and spinning velocities [10][11][12][13]. S. Furnuno et al. tried to measure the ball's spin via a vision system configured with a super-high frequency camera. The ball was manually marked, the relationship between camera target direction and moving ball direction was constrained to be perpendicular. Moreover, image processing to recognize the marks on the ball can't catch the super-high frequency of camera (1200fps) which limits the system in real-time application [14]. A. Nakashima et al. applied a similar idea to measure the spin of a ping-pong ball that was manually marked with dots and used the results to help modelling rebound process [15]. Due to the similar approach, they have the same shortcomings. X. Chen et al. proposed a novel idea that spin can be estimated from the deviation of trajectory. They deduced the dynamic model of a spinning ball, then the flight trajectory of a spinning ball can be predicted with estimated spin state [6]. Y. Huang et al. detailed the derivation of this idea and implemented several experiments

[1] Yifeng Zhang, Yongsheng Zhao, Rong Xiong, Yue Zhao and Jian Chu are with the State Key Laboratory of Industrial Control and Technology, Zhejiang University, Hangzhou, China. Rong Xiong is the corresponding author.rxiong@iipc.zju.edu.cn

[2] Jianguo Wang is with the Center for Autonomous Systems, Faculty of Engineering and IT, University of Technology, Sydney, Australia.

[1] http://news.cnet.com/8301-17938_105-20125182-1/chinas-ping-pong-robots-got-game/

to verify the result [16]. However our experiments indicate that the spin estimation based only on position information is not robust and very sensitive to observation error, The detailed explanation is given in Section II. In summary, real-time observation and trajectory prediction for spinning ball are still open problems.

Based on the fact that a spinning ball's motion can be separated into global movement and spinning respect to its center, this paper proposes a new vision system that combines a traditional position vision system with a pan-tilt vision system to observe the two motions separately. We call the new vision system as position and spin vision system, in which the the position vision system observes the ball's position state using the method of [5], while the pan-tilt vision system restores ball's 3D pose by recognizing the brand of the ball and estimates ball's spin state with the method of plane fitting on current and historical observations. With both position and spin information, we can estimate ball's state and predict the trajectory accurately via EKF. Compared with [14][15], the proposed method dose not require manual marks, or super-high frequency camera, and there are no limitation on the camera's installation. Our method gives an innovative solution to observe spin directly and in real-time, with which the prediction accuracy is well improved.

The remainder of this paper is arranged as follows. In Section II, the motivation of this work is discussed and the framework is introduced. Then 3D pose restoring, spinn state estimation and trajectory prediction are described in Section III, IV and V respectively. Experiments that verify the effectiveness of the proposed method are conducted in Section VI. Finally, conclusion is given.

II. MOTIVATION STATEMENT AND FRAMEWORK

A. MOTIVATION

There are three main forces that impact on a flying ball: gravity F_g, air drag F_d, and Magnus force F_m. F_d is opposite to flying direction, and F_m is perpendicular to spin axis and flying direction, as shown in Fig.1. Their values can be calculated as follows.

$$F_g = \begin{pmatrix} 0 & 0 & -mg \end{pmatrix}^T \tag{1}$$

$$F_d = -\frac{1}{2} C_D \rho_a A \|v\| v \tag{2}$$

$$F_m = \frac{1}{2} C_M \rho_a A r (\omega \times v) \tag{3}$$

where m is the mass of ball, g is the acceleration of gravity, ρ_a is the air density, C_D is the drag coefficient, C_M is the lift coefficient, r is the radius of the ball, A is the cross-sectional area of the ball, $v = [v_x, v_y, v_z]^T$ is the global velocity, and $\omega = [\omega_x, \omega_y, \omega_z]^T$ is the spin velocity of the ball.

Then the kinematical model is given as

$$\begin{pmatrix} \dot{v}_x \\ \dot{v}_y \\ \dot{v}_z \end{pmatrix} = \begin{pmatrix} -k_D \|v\| v_x + k_M(\omega_y v_z - \omega_z v_y) \\ -k_D \|v\| v_y + k_M(\omega_z v_x - \omega_x v_z) \\ -g - k_D \|v\| v_z + k_M(\omega_x v_y - \omega_y v_x) \end{pmatrix} \tag{4}$$

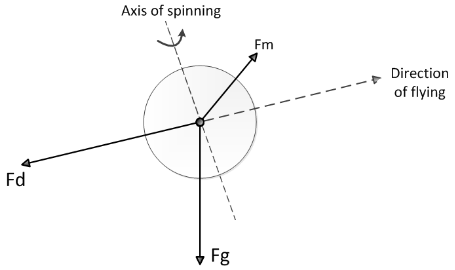

Fig. 1. Forces that impact on a flying ball.

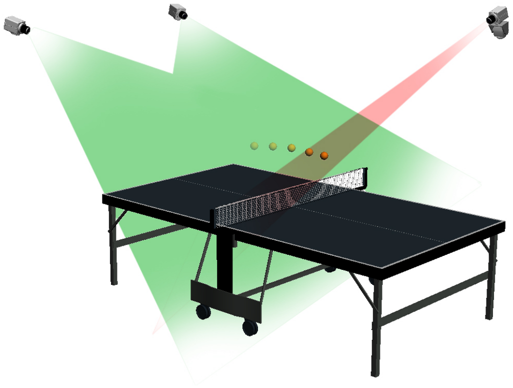

Fig. 2. 3D virtual scene of position and spin vision system.

where $k_D = \frac{1}{2m} C_D \rho_a A$, $k_M = \frac{1}{2m} C_M \rho_a r A$. (4) shows the fact that velocities along different axes are coupled, so are the positions. When a ball is moving with spinning, the trajectory analysis based on the assumption that position and velocity can be separated into three directions and be solved independently, will cause unexpected error.

For position vision systems, only ball positions on each cycle can be obtained. If we want to resolve the spin velocity ω based on (4), the estimation error would be high due to first derivative and second derivative of the observation noise, as well as the nonlinearity between v and ω. However, this error can be reduced if the spin is directly observed, which inspires the strategy of our work.

B. FRAMEWORK

Fig.2 shows the idea of combining position vision system with pan-tilt vision system. The position vision system has two or more cameras with short lens to get a wide view covering the table, with which the image of ball is very small and blurred, so only ball's position information can be observed. The pan-tilt vision system equips long lens and tracks the ball when ball moves. Thus it can get large and clear image of ball, with which it is practical to recognize the brand on it and then realize the spin estimation.

The framework of position and spin vision system is shown in Fig.3. The position vision system outputs ball's global position and the pan-tilt vision system estimates the spin state of ball, then these two observations are combined together to do state estimation and trajectory prediction of ball's flight. Brand recognition and ball tracking are

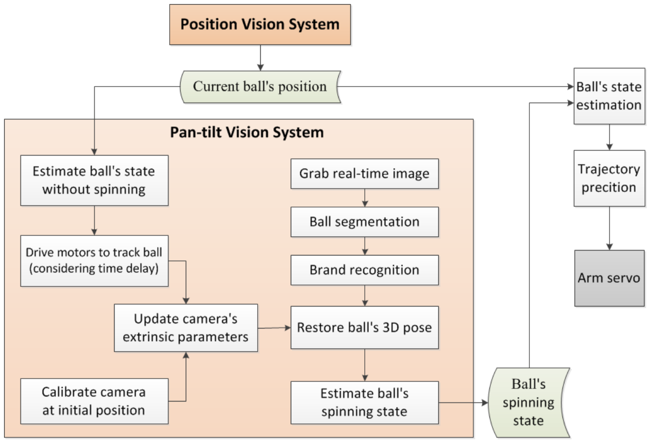

Fig. 3. Framework of position and spin vision system.

implemented in multi-thread way. Once current position of ball is updated by position vision system, the motors of pan-tilt vision system are driven to track the ball according to geometric relationship. The time delay of whole system should be taken into account when tracking. In the meantime, the pan-tilt camera works independently with the cameras in position vision system. When a new frame is captured, the brand on the ball is recognized based on frame difference algorithm. Then ball's 3D pose is restored and the spin state is estimated with current and historical observations. One thing that needs to be pointed out is that two or more pan-tilt vision systems can work together in our framework to get more observations and do more precise estimation of ball's spin state. As each pan-tilt camera can only observe one side of the ball, two such cameras are equipped on both sides of the table in our experiment.

III. RESTORING BALL'S 3D POSE

Restoring the ball's 3D pose at each frame is one of the key techniques of our method, which can be realized through three main steps: brand recognition, 3D pose restoration in camera coordinates, and 3D pose calculation in ball coordinates.

A. DEFINITION OF BALL COORDINATES

The spinning motion can be split with ball's global movement, and analyzed in a dynamic coordinates respect to ball's center. Fig.4 shows the definition of ball coordinates, in which the origin is the ball's center and the axes are parallel to axes of global coordinates. Then ping-pong ball's 3D pose can be represented as the vector pointing from ball's center to brand's center, in other words the position of brand's center in ball coordinates.

B. BRAND RECOGNITION

The main challenge of brand recognition lies in the uneven illumination in different areas of environment. In most cases, both shadows and reflection exist in ball's image. Based on the illumination model of environment and the fact that ball's positions are very close between two adjacent frames, we can assume that illuminations are the same in two adjacent frames. Thus we proposed to use frame difference technique between the ball's background image and the ball's image to recognize the brand, where the ball's background image is updated frame by frame to adapt the uneven illumination. Several ball's background images in different areas are learned off-line which can be used as the initial background image according to ball's global position.

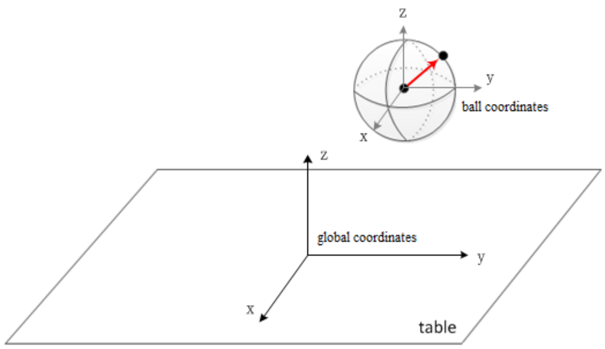

Fig. 4. Definition of ball coordinates and representation of ball's 3D pose.

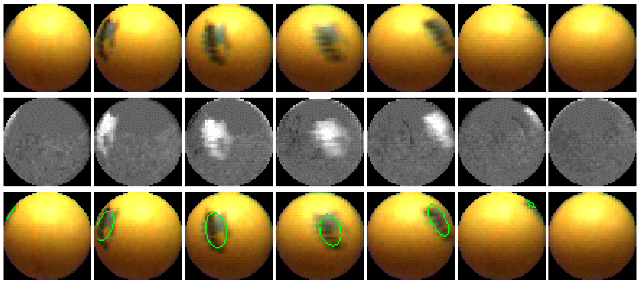

Fig. 5. Demonstration of brand recognition process. Images in first row are ball's raw images in successive frames after size normalization. Images in second row are the difference results between ball's current and background images. Images in third row show the brand recognition results.

Fig.5 demonstrates the brand recognition process. Firstly, all of the ball's images in different frames are unified into same size, $61 * 61$ pixels in this paper, as shown in first row in Fig.5. Secondly, difference between current frame image and the updated background image is obtained, as illustrated in second row. After that, the brand is segmented using a threshold and verified by a contour fitting method, the result is shown in third row. The difference and segmentation steps can be described by

$$I_{bd}(x,y) = \begin{cases} 1 & if \quad I_{bk}(x,y) - I_{bl}(x,y) > T_r(x,y) \\ 0 & else \end{cases} \tag{5}$$

where I_{bd} is the segmentation result, which is a binary image. I_{bk} and I_{bl} are ball's background and current images respectively. T_r is the threshold with a positive value and varying at different pixel points.

The ball's background image is updated for each frame as

$$I_{bk}(x,y) = \begin{cases} I_{bk}(x,y) & if \quad I_{bd}(x,y) = 1 \\ (1-\alpha)I_{bk}(x,y) + \alpha I_{bl}(x,y) & else \end{cases} \tag{6}$$

where α is the update rate in [0, 1]. The value of α is inverted with the frequency of vision system. The lower the frequency is, the larger the value. According to our experiments, $\alpha = 0.5$ works well at the frequency of 120fps.

Fig.5 also indicates that when brand locates closer to the edge area of ball in the image, its location error becomes larger due to occlusion and noise. So we assign the recognition result a confidence. The closer the brand location is to the ball's center, the higher the confidence it has, and vise verse.

C. 3D POSE RESTORATION IN CAMERA COORDINATES

The camera calibration result gives the relationships between image coordinates and camera coordinates. After getting the positions of ball's center and brand's center in image coordinates, their positions in camera coordinates are easy to be calculated.

$$\begin{pmatrix} Xc_b \\ Yc_b \\ Zc_b \end{pmatrix} = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix}^{-1} \begin{pmatrix} s_b u_b \\ s_b v_b \\ s_b \end{pmatrix} \tag{7}$$

$$\begin{pmatrix} Xc_m \\ Yc_m \\ Zc_m \end{pmatrix} = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix}^{-1} \begin{pmatrix} s_m u_m \\ s_m v_m \\ s_m \end{pmatrix} \tag{8}$$

where f_x,f_y,c_x,c_y are the intrinsic parameters of camera, s_b and s_m are the scale factors, (u_b, v_b) and (u_m, v_m) are positions of ball's center and brand's center in image coordinates, (Xc_b, Yc_b, Zc_b) and (Xc_m, Yc_m, Zc_m) are positions of ball's center and brand's center in camera coordinates.

Camera calibration also gives the initial position of camera P_c, while the ball's position P_b at each cycle is provided by position vision system. The distance between ball and camera can be approximately calculated as the ball to the initial position of camera since the pan-tilt movement will not change the camera's position much. Denote $D = \|P_b - P_c\|$, then

$$Xc_b^2 + Yc_b^2 + Zc_b^2 = D^2 \tag{9}$$

From (7) and (9), the ball's position in camera coordinates is gotten.

$$\begin{cases} Xc_b = \frac{u_b - c_x}{f_x} Zc_b \\ Yc_b = \frac{v_b - c_y}{f_y} Zc_b \\ Zc_b = \sqrt{D^2 - Xc_b^2 - Yc_b^2} \end{cases} \tag{10}$$

The ball's sphere surface in camera coordinates can be described as

$$(X - Xc_b)^2 + (Y - Yc_b)^2 + (Z - Zc_b)^2 = r^2 \tag{11}$$

where r is the radius of ping-pong ball. And (8) can be simplified as

$$\begin{cases} Xc_m = \frac{u_m - c_x}{f_x} Zc_m \\ Yc_m = \frac{v_m - c_y}{f_y} Zc_m \\ Zc_m = s_m \end{cases} \tag{12}$$

Combining (11) and (12), we get

$$a{s_m}^2 + b s_m + c = 0 \tag{13}$$

where

$$\begin{cases} a = (\frac{u_m - c_x}{f_x})^2 + (\frac{v_m - c_y}{f_y})^2 + 1 \\ b = -2(\frac{u_m - c_x}{f_x} Xc_b + \frac{v_m - c_y}{f_y} Yc_b + Zc_b) \\ c = (Xc_b)^2 + (Yc_b)^2 + (Zc_b)^2 - r^2 \end{cases}$$

(13) has two solutions of s_m, which indicates that two points on the sphere can project to the pixel (u_m, v_m). The smaller one corresponds to the real position of brand's center.

After getting the positions of ball's center and brand's center, the ball's 3D pose in the camera coordinates can be determined as the vector pointing from ball's center to brand's center.

$$P_c = [Zc_m - Zc_b \quad Zc_m - Zc_b \quad Zc_m - Zc_b]^T \tag{14}$$

D. 3D POSE RESTORATION IN BALL COORDINATES

The camera coordinates is changing with the pan-tilt movement. To restore ball's 3D pose in ball coordinates, the camera's pose in global coordinates has to be updated at each cycle. According to Rodrigues' rotation formula, the rotation matrix caused by pan-tilt movement can be solved as

$$\begin{aligned} R_r(\beta) = & \cos(\beta) * I + (1 - \cos(\beta)) * r * r^T \\ & + \sin(\beta) * \begin{pmatrix} 0 & -r_z & r_y \\ r_z & 0 & -r_x \\ -r_y & r_x & 0 \end{pmatrix} \end{aligned} \tag{15}$$

$$R_x(\alpha) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha) & -\sin(\alpha) \\ 0 & \sin(\alpha) & \cos(\alpha) \end{pmatrix} \tag{16}$$

where α and β are the angles that camera pans and tilts from initial position, $r = [r_x, r_y, r_z]^T$ is the axis of pan movement in initial camera coordinates. Then the extrinsic parameters of camera at each cycle can be updated as follow.

$$[R \quad T] = (R_r(\beta) R_x(\alpha))^{-1} [R_0 \quad T_0] \tag{17}$$

where R_0 and T_0 are the camera extrinsic parameters at initial position. Since the axes of ball coordinates is parallel to that of global coordinates, ball's 3D pose in ball coordinates can be easily calculated by

$$P_b = R^{-1} P_c \tag{18}$$

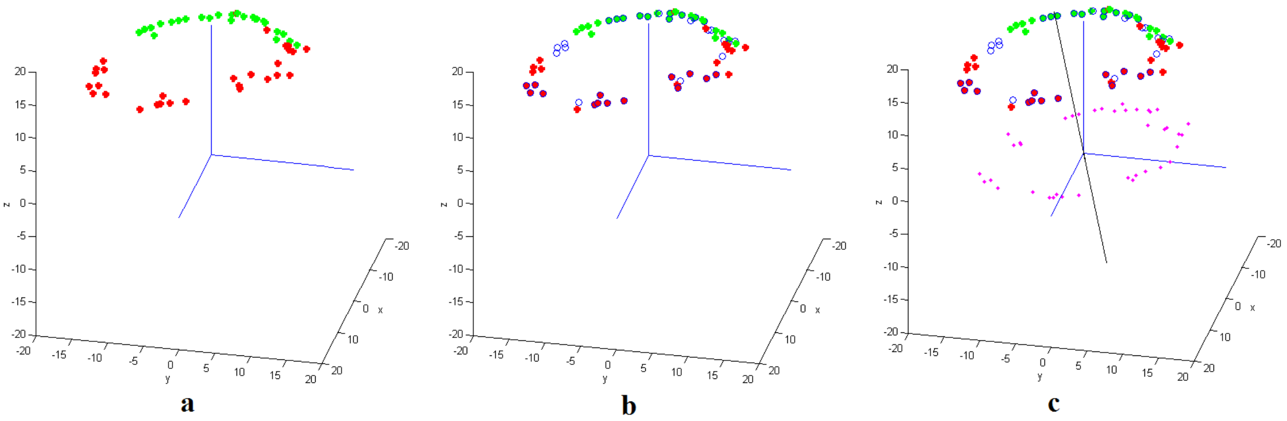

Fig. 6. **a**. The observations of brand positions in frames of one flying trajectory in ball coordinates, the ball has spun 6 cycles. The red points are the observations by one pan-tilt camera and the green points are by the other. **b**. In some frames both pan-tilt cameras observed the brand, then weighted average result will be used as new observation for that frame, as shown by blue circles. **c**. The axis of spin pass through ball's center and is perpendicular to the plane, shown as the black line. The magenta points are the projection of observation points to the cross section of ball perpendicular to axis, which will be used to calculate the angular velocity.

IV. SPIN STATE ESTIMATION

Spin state is estimated based on several frames of the ball's 3D pose restored. All of the ball's 3D poses are described in the ball coordinates.

Fig.6a illustrates the distribution of brand positions in different frames of one flying trajectory, in which two colors, red and blue, correspond to the observations of two pan-tilt cameras. In some frames, both pan-tilt cameras observe the brand, then a weighted average result is calculated based on the confidences of brand recognition and used as the observation of that frame as shown in Fig.6b.

It is seen the brand positions lie almost in one plane, which means the axis of spin is not changing during flying. Based on the forces analysis in Section II, main forces that impact on ball all pass through the ball's center, which means they have no effect on the angular velocity of spin. Hence we can infer: **the axis and the angular velocity of spin do not change during flying**.

Because the normal vector of the plane is parallel to the axis of spin, we can transform the problem of spin axis calculation to that of normal vector calculation. Fitting the plane of all observation points with RANSAC method, the normal vector is solved for. The axis of spin can be determined by fixing the normal vector passing through ball's center, as shown in Fig.6c.

Fig.7 demonstrates how to calculate the angular velocity of spin. The error variance of brand observation is quite different in each frame according to various situations of illumination and ball's pose in camera coordinates, thus it's hard to be modeled. So we use the average filter rather than other filter algorithms, the average angular velocity is calculated as

$$\phi = \frac{m \times 2\pi + \theta_n}{(n-1)T} \tag{19}$$

where n is current frame count, m means how many rotations the ball has spun, and θ_n is the angle between current point and the first point. Observation error is only occurred in θ_n and independent to time. Thus as time goes by, the estimation of ϕ will be more and more accurate.

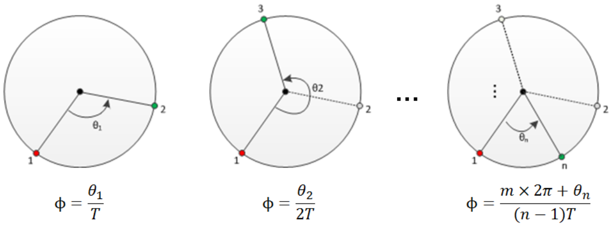
1
2
θ_1
$\phi = \frac{\theta_1}{T}$
1
2
3
$\theta 2$
$\phi = \frac{\theta_2}{2T}$
...
1
2
3
n
θ_n
$\phi = \frac{m \times 2\pi + \theta_n}{(n-1)T}$

Fig. 7. Calculate angular velocity at different frames.

V. STATE ESTIMATION AND TRAJECTORY PREDICTION

A. STATE ESTIMATION

To apply the kinematical model to trajectory analysis of a flying ball, the discrete form of dynamic model is needed due to discrete observations. Denote the time interval between frames as ΔT, and $k_d = k_D \Delta T$, $k_m = k_M \Delta T$, the discrete dynamic model is given as follows.

$$\begin{pmatrix} x(k+1) \\ y(k+1) \\ z(k+1) \\ v_x(k+1) \\ v_y(k+1) \\ v_z(k+1) \end{pmatrix} = F(k) \begin{pmatrix} x(k) \\ y(k) \\ z(k) \\ v_x(k) \\ v_y(k) \\ v_z(k) \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -g\Delta T \end{pmatrix} \tag{20}$$

where $(x(k), y(k), z(k))$ and $(v_x(k), v_y(k), v_z(k))$ are ball's position and velocity in k^{th} frame, and

$$F(k) = \begin{pmatrix} 1 & 0 & 0 & \Delta T & 0 & 0 \\ 0 & 1 & 0 & 0 & \Delta T & 0 \\ 0 & 0 & 1 & 0 & 0 & \Delta T \\ 0 & 0 & 0 & 1 - k_d\|v(k)\| & -k_m\omega_z & k_m\omega_y \\ 0 & 0 & 0 & k_m\omega_z & 1 - k_d\|v(k)\| & -k_m\omega_x \\ 0 & 0 & 0 & -k_m\omega_y & k_m\omega_x & 1 - k_d\|v(k)\| \end{pmatrix}$$

Then EKF can be applied to do state estimation, the state variable is $[x(k), y(k), z(k), v_x(k), v_y(k), v_z(k)]^T$, the predict model is given by (20). The measurement model is given as

$$\begin{pmatrix} x(k) \\ y(k) \\ z(k) \end{pmatrix} = \begin{pmatrix} I_{3\times 3} & 0_{3\times 3} \end{pmatrix} \begin{pmatrix} x(k) \\ y(k) \\ z(k) \\ v_x(k) \\ v_y(k) \\ v_z(k) \end{pmatrix} \tag{21}$$

The spin velocity ω is used as a constant value in EKF. As discussed in Section IV, ω is estimated based on all historical observations, and the estimation tends to be more precise with more observations. So in each frame, ω will be re-estimated, and the EKF will also have to restart over again for all observations of this flying trajectory with new ω. For a ping-pong game, this process will not cost much time but can provide more precise estimation. For other continuous running applications, we can just refilter the latest several observations.

B. *TRAJECTORY PREDICTION*

When playing a ping-pong game, the flight trajectory can be split into three parts: flying before collision, collision with table, and flying after collision. The two flying trajectories satisfy the dynamic model given above.

Considering the physical properties of rebound and referring to experiments result, the rebound model for collision between ball and table is described as

$$\begin{cases} v_{xout} = & \begin{bmatrix} v_{xin} & \omega_{yin} \end{bmatrix} b_1 \\ v_{yout} = & \begin{bmatrix} v_{yin} & \omega_{xin} \end{bmatrix} b_2 \\ v_{zout} = & v_{zin} b_3 \\ \omega_{xout} = & \begin{bmatrix} v_{yin} & \omega_{xin} \end{bmatrix} b_4 \\ \omega_{yout} = & \begin{bmatrix} v_{xin} & \omega_{yin} \end{bmatrix} b_5 \\ \omega_{zout} = & \omega_{zin} b_6 \end{cases} \tag{22}$$

where $b_1, b_2, b_4, b_5 \in R^2$ and $R_3, R_6 \in R$, and the values are estimated using LSM(Least Square Method).

Once the current state of ball is estimated, its following trajectory can be easily calculated.

VI. EXPERIMENTS

Experiments are conducted to verify the proposed method on observation and trajectory prediction of spinning ping-pong ball. One position vision system catches global position of ball, the maximum localization error $< 0.9cm$. Two pan-tilt vision systems are used to estimate spin state of ball. All cameras are working in 120fps with the resolution of 640×480 pixels. The time cost of the proposed method for each frame is less than 5ms on a PC equips Core 2 Duo 2.67GHz processor and 2GB RAM.

Parameters used in experiments are as follows. $T = 1/120$, $r = 0.02$, $m = 0.00275$, $g = 9.82$, $\rho_a = 1.29$, $C_D = 0.405$, $C_M = 0.62$, $b_1 = [0.75, 0.0015]^T$, $b_2 = [0.75, 0.0015]^T$, $b_3 = -0.97$, $b_4 = [-26, 0.53]^T$, $b_5 = [25, 0.6]^T$, $b_6 = 0.9$.

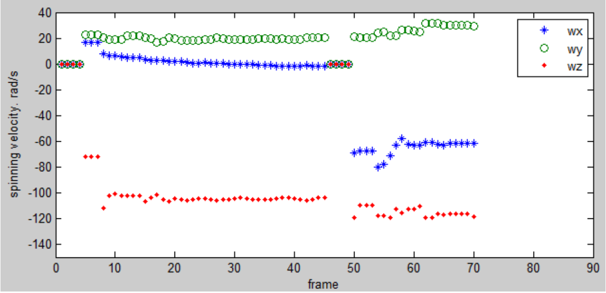

Fig. 8. Estimation of spin state $\omega = (\omega_x, \omega_y, \omega_z)^T$, in each frame the estimation is based on observations before this frame.

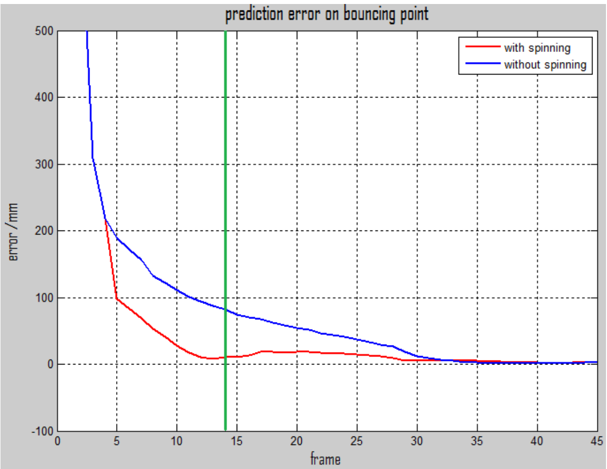

Fig. 9. Prediction result on bouncing point. The green line indicates the time when ball is flying through 1/4 length of table

Fig.8 shows the spin estimation of each frame in one flight trajectory. The estimation is based on the current and historical observations, at least three observations is needed to estimate the spin state. So the vision system has to wait a few frames to get a estimation result, the waiting time is only related to images of brand in pan-tilt cameras and independent to ball's spin velocity. According our experiments, In most cases our vision system can give a precise spin estimation in less than 10 frames. For the example in Fig.8, the vision system gave accurate spin estimation at 8th frame for flying before collision with table. After collision between frame 45 and 46, the spin estimating process restarted and gave accurate result 4 frames later. Fig.8 also proves the inference that spin state does not change during flying is reasonable.

Fig.9 shows the prediction result on the bouncing point of the same data using the spin estimation result in Fig.8. The pan-tilt vision system had to wait 5 frames to get a rough estimation of spin state, so the prediction results were the same in first 4 frames. At the 5th frame, although the estimated spin state was not accurate, its spin direction was approximately correct and can still increase the prediction accuracy if this information is used. The spin estimation became more precise since 8th frame and the prediction was greatly improved.

In the application of ping-pong robots, the vision system

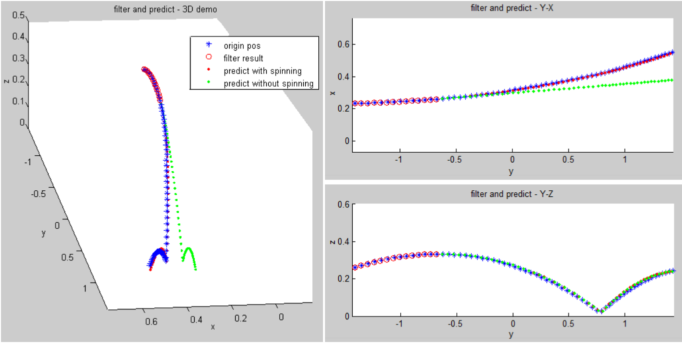

Fig. 10. Trajectory prediction when ball has flied 1/4 length of the table. The blue points give the truth trajectory. The red circle at the head of the trajectory is the filter result based on observations before each frame, and the last red circle indicate current frame. The red points and the green points are the prediction result with and without considering spin.

should give precise predictions in early stage to leave enough time for robots to react. When ball is moving with a velocity of $[6-10]m/s$, the robot should begin to move at the time about ball passing through 1/4 length of the table, shown as the green line in Fig.9. Fig.10 illuminates the prediction result when ball was flying through 1/4 length of the table in different views.

In addition, the main component of ball's velocity was along positive y axis, and Fig.8 shows the spin was mostly in negative z direction. According to dynamic model, the Magnus force was mainly in positive x direction, so the trajectory would be deviated to positive x direction, which is consistent with the result that the prediction error is mostly in x axis if not considering spin in Fig.10. Furthermore, the rebound model indicates that ω_{xout} would decrease after rebounding which also was supported by the result in Fig.8.

The proposed method was tested by 15 trajectory data with various spin velocities from $60rad/s$ to $200rad/s$, and compared with the method not considering spin and the method estimate spin from trajectory deviation. Tab.I lists the error of trajectory prediction on bouncing points and hitting points, the prediction is done at the last frame before ball flying through 1/4 length of the table. Method1 uses only position information and didn't consider spin. Method2 uses only position information, but it considers spin which is estimated based on trajectory deviation. Method3 is the proposed method that uses both position and spin information to do state estimation and trajectory prediction. The result shows the effectiveness and advancement of the proposed method.

VII. CONCLUSIONS

This paper proposed an innovative position and spin vision system, which can observe both position and spin information of ping-pong ball directly and in real-time. Applying these information in a dynamic model based on forces analysis with EKF method, we can get more precise state estimation and trajectory prediction results for a spinning ball. Experimental results show the effectiveness and accuracy of the proposed method.

TABLE I

ERROR ANALYSIS OF TRAJECTORY PREDICTION RESULT.

		prediction error on hitting point(x-y)		prediction error on bouncing point(x-z)	
/mm		average	standard deviation	average	standard deviation
Method1	x	65.9	71.9	141.7	156.8
	y/z	78.6	103.7	29.4	53.8
Method2	x	35.6	38.0	46.6	51.3
	y/z	36.4	48.0	39.1	45.6
Method3	x	13.2	16.1	18.6	21.3
	y/z	21.6	24.9	20.6	23.5

Although the proposed method may not work well in the particular case that the brand on the ball lies at the spin axis. One solution is using the brand recognition result as an ellipse rather than just a point, then axis direction of ellipse can be used to estimate spin state. This solution is relied on more accurate and robust brand recognition, which we will focus on in our next work. Also the method of estimating spin state from trajectory deviation can be used as a supplementary of the proposed method, such as providing initial value in first several frames and handling above special situations.

Another future piece of work will concentrate on modeling the rebounding process between ball and racquet, which can help the robot to hit spinning balls back to a target area on the opponent's court.

ACKNOWLEDGMENT

This work was supported by the Scholarship Award for Excellent Doctoral Student granted by Ministry of Education and the Joint Center for Robotics Research (JCRR) between Zhejiang University and the University of Technology, Sydney.

REFERENCES

[1] L. Acosta, J. Rodrigo, J. A. Mendez, G. Marichal, and M. Sigut, "Ping-pong player prototype," *Robotics & Automation Magazine, IEEE*, vol. 10, no. 4, pp. 44–52, 2003.

[2] Y.-h. Zhang, W. Wei, D. Yu, and C.-w. Zhong, "A tracking and predicting scheme for ping pong robot," *Journal of Zhejiang University SCIENCE C*, vol. 12, no. 2, pp. 110–115, 2011.

[3] M. Matsushima, T. Hashimoto, M. Takeuchi, and F. Miyazaki, "A learning approach to robotic table tennis," *Robotics, IEEE Transactions on*, vol. 21, no. 4, pp. 767–771, 2005.

[4] Z. Zhang, D. Xu, and M. Tan, "Visual measurement and prediction of ball trajectory for table tennis robot," *Instrumentation and Measurement, IEEE Transactions on*, vol. 59, no. 12, pp. 3195–3205, 2010.

[5] Y. Zhang, R. Xiong, Y. Zhao, and J. Chu, "An adaptive trajectory prediction method for ping-pong robots," in *Intelligent Robotics and Applications*. Springer, 2012, pp. 448–459.

[6] X. Chen, Y. Tian, Q. Huang, W. Zhang, and Z. Yu, "Dynamic model based ball trajectory prediction for a robot ping-pong player," in *Robotics and Biomimetics (ROBIO), 2010 IEEE International Conference on*. IEEE, 2010, pp. 603–608.

[7] C. H. Lampert and J. Peters, "Real-time detection of colored objects in multiple camera streams with off-the-shelf hardware components," *Journal of Real-Time Image Processing*, vol. 7, no. 1, pp. 31–41, 2012.

[8] K. Mülling, J. Kober, and J. Peters, "A biomimetic approach to robot table tennis," *Adaptive Behavior*, vol. 19, no. 5, pp. 359–376, 2011.

[9] Y. Drianovski and G. Otcheva, "Survey of games styles of some of the best asian players at the 12th world university table tennis championships (sofía, 1998)," *Table Tennis Sciences*, vol. 4, pp. 3–9, 2002.

[10] B. Robins and W. Curtis, *New principles of gunnery*. Richmond Publishing Company Limited, 1972.

[11] G. Magnus, “Ueber die abweichung der geschosse, und: Ueber eine auffallende erscheinung bei rotirenden körpern,” *Annalen der physik*, vol. 164, no. 1, pp. 1–29, 1853.

[12] H. Barkla and L. Auchterlonie, “The magnus or robins effect on rotating spheres,” *J. Fluid Mech*, vol. 47, no. part 3, pp. 437–447, 1971.

[13] C. T. Crowe, D. F. Elger, J. A. Roberson, C. T. Crowe, and C. T. Crowe, *Engineering fluid mechanics*. Wiley Hoboken, 2005, vol. 7.

[14] S. Furuno, K. Kobayashi, T. Okubo, and Y. Kurihara, “A study on spin-rate measurement using a uniquely marked moving ball,” in *ICCAS-SICE, 2009*. IEEE, 2009, pp. 3439–3442.

[15] A. Nakashima, Y. Ogawa, Y. Kobayashi, and Y. Hayakawa, “Modeling of rebound phenomenon of a rigid ball with friction and elastic effects,” in *American Control Conference (ACC), 2010*. IEEE, 2010, pp. 1410–1415.

[16] Y. Huang, D. Xu, M. Tan, and H. Su, “Trajectory prediction of spinning ball for ping-pong player robot,” in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*. IEEE, 2011, pp. 3434–3439.