A New Data Processing Architecture for Table Tennis Robot

Jianran Liu, Zaojun Fang, Kun Zhang and Min Tan

Abstract— **Playing table tennis with human is a challenging work for robot. The common architecture for table tennis robot in the past researches are to gather information at first, then predict the ball status at hitting point, and the motion mechanism moves according to the prediction at last. In order to save the time for the later two steps, the gathered information is limited. In this paper, we propose a new architecture, in which the robot would keep gathering the information and amending the prediction as the motion mechanism moves. To combine the gathered information and obtain reliable prediction of hitting point, a data processing algorithm is proposed. The algorithm is based on the kinetic analysis of the ball and the extended Kalman filter. The estimation of angular velocity measurement is also improved in this algorithm. Finally, the validity of the algorithm is confirmed by conducting experiments.**

I. INTRODUCTION

Playing table tennis with human is a challenging task for robot. The complexity of the task make table tennis robot an ideal platform for many research. Since 1980s, a number of research groups have focus on this question and quite a lot of works have been done. Most work follows a common framework: Firstly, the ball is detected with high speed vision system. Then the hitting point is predicted by model identification method. At last, the motion mechanism completes the hitting action. Therefore, high speed vision procession, model identification and motion control are the three main contents to be studied.

In most early work like [1] and [2], the spin of ball is ignored because of the difficulty both in the measurement of ball angular velocity and the complex kinetic analysis. In recent years, with the improvement of technology, many groups take the spinning of the ball into consideration. In [3], Nakashima explored a high-speed vision sensors with 900 frames per second to capture the rotation of ball, and derived analytical models of the rebound phenomenon between a ping-pong ball and the table/racket rubber. Then in 2014, with the help of a pan-tilt vision system, Zhang observed the spinning motion through recognizing the position of the brand on the ball [4]. This kind of solutions are direct and accurate. However, they must be supported by camera systems with extremely high sample speed, which make the whole system expensive and complex. From another aspect, mathematic method is proposed in [5] and [6] to solve this problem. In this kind of method, the angular velocity of the ball is estimated based on the bending trajectory of the ball. Their defect is that to guarantee the robustness, we have to capture a relative longer flying trajectory of the ball. In practical playing, a long-time detecting process will not leave the motion mechanism enough time to finish the hitting action. Therefore, they are difficult to be used in practical playing.

To overcome such conflict, we propose a solution that improves both the detecting model and the action model. In our solution, a rough initial hitting parameters is delivered to the motion mechanism once the necessary data is gathered. With these parameters, the motion mechanism begins to move at once. Meanwhile, the robot keeps tracking the ball, and amending the initial parameters when the motion mechanism is moving.

In order to realize such design, the robot shall contains two features. Firstly, the motion mechanism demands to be capable of altering its destination in real time. Second, the robot demands to

*This work was supported by the National Natural Science Foundation of China under Grant 61305024, Grant 61273337, Grant 61227804 and Grant 61333016. It was also supported by the Beijing Natural Science Foundation under Grant 3141002.

Jianran Liu, Zaojun Fang, Kun Zhang and Min Tan are with the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, P. R. China (Liu: +86-010-82544529; e-mail: liujianran2012@ia.ac.cn).

be capable to combine the ball status information gathered at realtime and the trajectory information gathered earlier.

In this paper, we introduce an algorithm which enable the robot to combine the realtime information and the history information. The work is based on physical model and the extended Kalman filter(EKF). With more abundant ball status captured, the estimation of angular velocity is also improved.

The remainder of this paper is organised as follows. First of all, section II introduces the hardware framework of our robot. Then section III gives the fundamental kinetics model for the ball. Based on this model, Section IV proposes filters both for the captured ball status and the predicted hitting points. The complete algorithm is described in V, and the experimental results are presented in section VI. Finally, in section VII, the conclusion is given.

II. System Hardware

Our work is based on a platform which is improved form the work in [7] and [8]. The system contains three parts: a stereo vision system to detect the ball, a motion mechanism to hit the ball and a control system to connect both of them. Figure 1 presents the practical robot system. In the picture, the stereo vision system is marked by the red rectangle. It contains 2 smart cameras work at 200 frames per second. Besides, the data gathered by the cameras are processed by a regular computer. The motion mechanism with 5 degrees of freedom is marked by the green rectangle. The motion controller is marked by the yellow rectangle. Its core component is a motion control card, by which the motion mechanism can change its trajectory planning at real time, which is a key feature in the proposed algorism.

III. Kinetics Model of Flying Ball

The kinetics model of flying ball is important for the prediction of the ball trajectory. There have been many researches to this topic. Some groups simply use polynomials to fit the sample data [9], others use models based on simplified kinetics analysis like [4], [10]. Our work is based on the model proposed in [6], which take the spinning of ball into consideration.

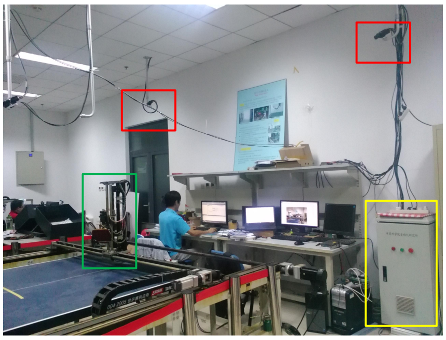

Fig. 1: The table tennis robot

In this section, we will introduce the flying and rebounding model proposed in [6].

A. Flying Model

In the model proposed in [6], there are three main forces on the flying ball, which are the Magnus Force, gravity, and air resistance. The force analysis can be expressed as follows

$$\begin{aligned} m\dot{\vec{v}} &= F_D + F_G + F_M \\ F_D &= -\frac{1}{2}\rho S C_D \|\vec{v}\|\vec{v} \\ F_M &= \frac{1}{2} C_M \rho S r_b \vec{\omega} \times \vec{v} \\ F_G &= [0\ 0\ \ -mg] \end{aligned} \tag{1}$$

where F_G is the gravity force, F_D is the air resistance, F_M is the Magnus force, ρ is the air density, S is the effective cross-sectional ball area, C_D is the drag coefficient, C_M is the Magnus coefficient, r_b is the radius of the ball, m is the mass of the ball, g is the gravity accelerator, $\vec{v} = [v_x, v_y, v_z]^T$ is the flying velocity of the ball, $\vec{\omega} = [\omega_x, \omega_y, \omega_z]^T$ is the angular velocity.

According to the force analysis, the acceleration on each direction can be calculated as

$$\begin{bmatrix} \dot{v}_x \\ \dot{v}_y \\ \dot{v}_z \end{bmatrix} = \begin{bmatrix} -k_d\|\vec{v}\|v_x + k_m\delta_1 \\ -k_d\|\vec{v}\|v_y + k_m\delta_2 \\ -k_d\|\vec{v}\|v_z - g + k_m\delta_3 \end{bmatrix} \tag{2}$$

where $k_d = \frac{1}{2m}\rho S C_D$, $k_m = \frac{1}{2m} C_M \rho S r_b$, $\delta_1 = w_y v_z - w_z v_y$, $\delta_2 = w_z v_x - w_z v_z$, $\delta_3 = w_x v_y - w_z v_x$.

Assume that in the flying process, the angular velocity $\vec{w}$ keeps unchanged. With the calculated acceleration, the motion of the ball can be represented by a state-transition equation as

$$S[n] = AS[n-1] + BU[n] \quad (3)$$

where the vector $S = [x, y, z, v_x, v_y, v_z]$ denotes the status of ball, and the other parameters are explained as follows.

$$A = \begin{bmatrix} I & t_c I \\ 0 & P \end{bmatrix}$$
$$B = I$$
$$P = \begin{bmatrix} a_d & -k_m w_z t_c & k_m w_y t_c \\ k_m w_z t_c & a_d & -k_m w_x t_c \\ -k_m w_y t_c & k_m w_x t_c & a_d \end{bmatrix} \quad (4)$$

where t_c denotes the sample time, I is a 3×3 identity matrix , $a_d = 1 - k_d \|\vec{v}\| t_c$, $U[n] = [0\ 0\ 0\ 0\ 0\ -gt_c]$.

In the model, P varies with $S[n]$. Therefore, the ball status is a nonlinear system.

B. Rebounding Model

In [6], model to describe the ball rebounding at the table is also proposed. The model is presented as follow:

$$\begin{bmatrix} v_{x2} \\ v_{y2} \\ v_{z2} \end{bmatrix} = A_r \begin{bmatrix} v_{x1} \\ v_{y1} \\ v_{z1} \end{bmatrix} + \begin{bmatrix} a_3 w_{y1} \\ a_4 w_{x1} \\ 0 \end{bmatrix} \quad (5)$$

$$\begin{bmatrix} w_{x2} \\ w_{y2} \\ w_{z2} \end{bmatrix} = B_r \begin{bmatrix} w_{x1} \\ w_{y1} \\ w_{z1} \end{bmatrix} + \begin{bmatrix} b_3 v_{y1} \\ b_4 v_{x1} \\ 0 \end{bmatrix} \quad (6)$$

where $[v_{x1}, v_{y1}, v_{z1}]^T$ and $[w_{x1}, w_{y1}, w_{z1}]^T$ denotes the velocity and the angular velocity before the rebound, while $[v_{x2}, v_{y2}, v_{z2}]^T$ and $[w_{x2}, w_{y2}, w_{z2}]^T$ denotes the one after rebound. $A_r = diag(a_1, a_2, -e_z)$, $B_r = diag(b_1, b_2, e_w)$. a_i, b_i, e_z and e_w are parameters in the rebound model, and can be estimated by experiment.

C. The calculation of angle velocity

In both [6] and [4], similar models for the angle velocity calculation were proposed. The basic equation is introduced as

$$\begin{bmatrix} \Delta_1 \\ \Delta_2 \\ \Delta_3 \end{bmatrix} = M_w \begin{bmatrix} w_x \\ w_y \\ w_z \end{bmatrix} \quad (7)$$

The parameters in (7) are explained as follows:

$$M_w(n) = \begin{bmatrix} 0 & v_z(n) & -v_y(n) \\ -v_z(n) & 0 & v_x(n) \\ v_y(n) & -v_x(n) & 0 \end{bmatrix}$$

$$\Delta_1 = \frac{1}{k_m t_c} \left[v_x(n+1) - v_x(n) + k_d \|\overrightarrow{v}(n)\| v_x(n) t_c \right]$$

$$\Delta_2 = \frac{1}{k_m t_c} \left[v_y(n+1) - v_y(n) + k_d \|\overrightarrow{v}(n)\| v_y(n) t_c \right]$$

$$\Delta_3 = \frac{1}{k_m t_c} \left[v_z(n+1) - v_z(n) + k_d \|\overrightarrow{v}(n)\| v_z(n) t_c \right] \quad (8)$$

In (7), $M_w(n)$ is a singular matrix, thus the angle velocity can not be solved directly. However, given that our method keeps tracking the ball in the whole bout, ball statues can be captured for many times in one trajectory. Then (7) can be solved by least square method(LSM).

IV. Filters on Data

Based on the model described above, the gathered information demands to be processed to generate reliable prediction of hitting point. In this section, two filters are designed for both the captured ball status and the predicted hitting points.

A. Analysis on the ball status prediction

Theoretically, once the initial state is given, the subsequent states can be computed by numerical integration, and the hitting points can be predicted. However, the error both in the modeling of the process and in the detection of the ball is unavoidable, and would gather with the count of iteration. In this section, the model error caused by noise is analyzed, and its covariance matrix is given.

Considering a predict process from t_1 to t_2, and n times of iterations are needed($n = [(t_1 - t_2)/t_c]$). Let $S[i]$ denotes the ball status at the time $t = t_1 + it_c$, thus $S[0] = S_{t1}$ and $S[n] = S_{t2}$. Besides, let $w_c[i]$ denotes the model error in one iteration.

If there is no rebound process between t_1 and t_2, the prediction process can be represented in (9).

$$S[n] = (\prod_{i=0}^{n\ \ 1} A[i]) S[0] + \lambda[n] BU + W_c[n] \quad (9)$$

where

$$\begin{aligned}\lambda[n] &= \sum_{i=1}^{n-1}(\prod_{k=i}^{n-1}A[k]) + I \\ W_c[n] &= \sum_{i=1}^{n-1}(\prod_{k=i}^{n-1}A[k])w_c[i] + w_c[n]\end{aligned} \tag{10}$$

If the ball is just lower than the table in the kth iterations($0 < k < n$), then rebound model shall be applied. In this situation, $A[k]$ in (9) is replaced as $\hat{A}[k] = A_r A[k]$.

Based on (9), the mapping from S_{t1} to S_{t2} can be defined as

$$S_{t_2} = F_n(S_{t_1}) = \prod_{i=0}^{n-1} A[i]S_{t_1} + \lambda[n]BU \tag{11}$$

As discussed above, $F_n(\cdot)$ is a nonlinear function. It's Jacobian matrix is $\tilde{A} = \prod_{i=0}^{n-1} A[i]$. Besides, $W[n]$ denotes the gathered noise in the process of iteration. It can be computed by a recurrence formula

$$W_c[n] = A[n-1]W_c[n-1] + w_c[n] \tag{12}$$

Assuming that $w_c[n]$ is subject to Gaussian distribution with covariance matrix c_c, and the covariance matrix of $W_c[n]$ is denoted as $C_c[n]$, then we have

$$C_c[n] = A[n-1]C_c[n-1]A^T[n-1] + c_c \tag{13}$$

B. EKF on the ball status sequence

When the ball is flying across the table, a sequence of ball status is captured by the stereo vision system. Let S_{t_k} denotes the kth captured ball status, t_k denotes the capture time. To predict S_{t_k} from $S_{t_{k-1}}$, we shall run n_{k-1} iterations. According to (11) and (13), $S_k = F_{n_{k-1}}(S_{k-1})$, and the corresponding covariance of predict error is $C_{n_k} = C_c[n_k]$. With these information, EKF can be applied to estimate the status of the ball.

In the remaining of this paper, we use $\{\bar{S}_{t_k}\}$ to denote the filtered sequence.

C. Filter on the hitting point

With a relative stable estimation of ball status $\{\bar{S}_{t_k}\}$, the hitting points can be predicted.

In our robot, the hitting points are fixed on a virtual plane, as shown in figure **??**. Assuming that we need run n_h steps of iterations to translate ball status $\bar{S}_k$ to the virtual hitting plane, and the final predicted hitting point is H_k, we have

$$H_k = F_{n_h}(\bar{S}_k) \tag{14}$$

The covariance corresponding matrix is $C_H = C[n_h]$, which can be obtained by (13).

From each ball status in $\{\bar{S}_{t_k}\}$, corresponding sequence of landing point $\{H_k\}$ and its error covariance matrix $\{c_h\}$ can be obtained according to (13) and (14).

To combine the information in $\{H_k\}$ and $\{c_h\}$, a filter is designed as a refresh equation:

$$\begin{aligned}C_s &= C_h[k-1] + c_h \\ K_1 &= C_h[k-1]C_s^{-1} \\ K_2 &= c_h C_s^{-1} \\ C_h[k] &= (I - K_1)C_h[k-1] \\ \bar{H}[k] &= K_1\bar{H}[k-1] + K_2 H_k\end{aligned} \tag{15}$$

where K_1 and K_2 are respectively the weight matrix of the new predicted H_k and the last estimated $\bar{H}[k-1]$, $C_h[k]$ is the error covariance matrix of newly estimated $\bar{H}[k]$, which would be used in process of next captured data.

V. Algorithm Architecture

Based on the discussion in the section IV, a complete arithmetic is designed to generate accurate and stable sequence of $\bar{H}$. The flow chart for the algorithm is shown in figure 2.

As shown in the figure, the arithmetic is divided into two part. Part A is the a filter to smooth the captured ball status. In this part, the captured ball status at t_k serves as the observed value for the EKF, while the predicted ball status at t_k based on $\bar{S}_{t_{k-1}}$ serves as the predicted value. Furthermore, during the computation of prediction, the Jacobian matrix of F_{n_k} and the covariance matrix of the error($\tilde{A}_k$ and C_{t_k}), are obtained. All these information are delivered into the EKF filter. The outcome of the filter is $\bar{S}_{t_k}$, which is a more reliable estimation of ball status compared to S_{t_k}.

Then in the second part, a hitting point H_k is predicted based on $\bar{S}_{t_k}$. Its error covariance matrix c_h is also calculated during the iteration process according to (13). These information are processed by filter B, which is described in (15).

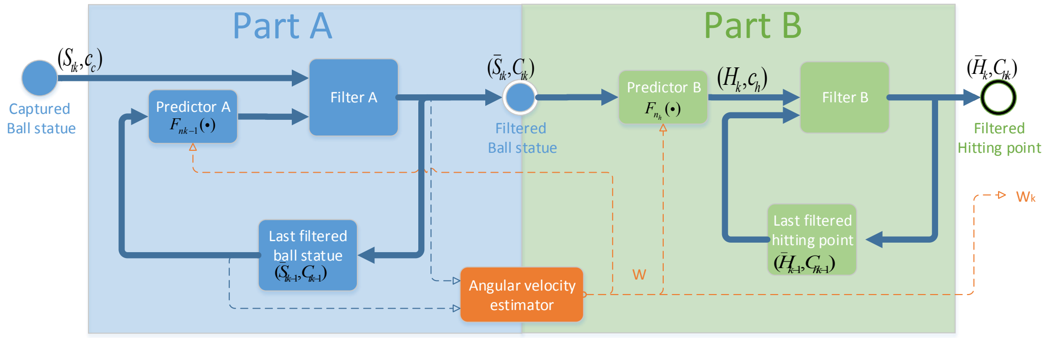

Fig. 2: Algorithm Architecture.

Finally, we get the filtered hitting point $\bar{H}_k$, and its error covariance matrix C_{h_k}.

Besides, with $\bar{S}_{t_k}$ and $\bar{S}_{t_{k-1}}$ keep refreshing, a sequence of $M_w[n]$ in (7) can be gathered. To save the calculating time, the recursive least squares method is explored to refresh the estimated angle velocity W. Then the refreshed angle velocity is updated to both the predictor A and predictor B, in preparation of the next captured data.

VI. EXPERIMENTS AND RESULTS

To testify the proposed method, a groups of trajectories were measured. The gathered trajectories contained different flying direction and initial status, and can present the situation in practical playing.

A. Filter on the ball's position

First of all, the proposed trajectory filter was explored to process the captured ball status by the stereo camera system. The effect is presented in figure 3. It shall be noticed that the scales of different axis are different, and the variation in y-direction is only 4 cm. Therefore, the error in y-direction is quite acceptable for practical playing.

It's clear that the filtered positions were more accordant with the physical law that a flying object obeys. Thus the filter is effective to restrain the noise in the captured data.

B. Filter on the hitting points

To verify the effectiveness of the proposed method, two further experiments were conducted. In the first experiment, with the filtered initial points $\{\vec{S}_{t_k}\}$, the hitting points were predicted.

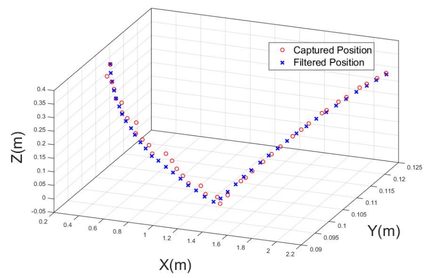

Fig. 3: Captured ball positions and filtered positions

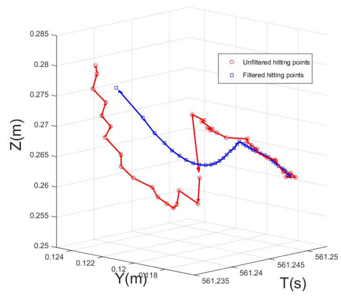

Fig. 4: Predicted hitting points with and without filtered.

The results were recorded with out filtered. In second experiment, the predicted hitting points were filtered by the proposed filter. Figure 4 represent the vibration of the predicted hitting points position in one typical trajectory.

In the figure, axis T denotes the predicted landing time and hitting time. It can be seen that the unfiltered prediction changed with no regular,

TABLE I: Standard deviations of the predicted hitting points (m)

	x		y		z		t	
NO.	Unfiltered	Filtered	Unfiltered	Filtered	Unfiltered	Filtered	Unfiltered	Filtered
1	0.0093	0.0066	0.0027	0.0017	0.0004	0.0002	0.0010	0.0008
2	0.0079	0.0058	0.0021	0.0015	0.0004	0.0001	0.0011	0.0008
3	0.0111	0.0085	0.0030	0.0021	0.0004	0.0001	0.0018	0.0014
4	0.0119	0.0074	0.0014	0.0010	0.0004	0.0001	0.0026	0.0018
5	0.0105	0.0090	0.0012	0.0009	0.0004	0.0006	0.0030	0.0022
6	0.0191	0.0133	0.0027	0.0019	0.0005	0.0002	0.0019	0.0014
7	0.0107	0.0074	0.0018	0.0012	0.0004	0.0001	0.0023	0.0016
8	0.0236	0.0145	0.0032	0.0019	0.0004	0.0002	0.0009	0.0004
9	0.0122	0.0085	0.0030	0.0018	0.0004	0.0001	0.0023	0.0017
10	0.0119	0.0094	0.0024	0.0016	0.0003	0.0001	0.0021	0.0017
11	0.0056	0.0042	0.0017	0.0010	0.0004	0.0001	0.0019	0.0014

which would lead to the vibration of the servo motion mechanism. Meanwhile, the variation of the filtered prediction is smooth and compact, which denotes that the prediction is stable.

The proposed algorithm is applied on the different trajectories, and the captured points at the hitting plane are considered as the ground truth points. The standard deviations of the predicted hitting points are recorded in the table I. It can be seen that the standard deviations of filtered prediction is much smaller than the unfiltered one in all the trajectories, which indicates that the algorithm is effective and robust.

VII. CONCLUSIONS

In this paper, a new data processing algorithm for table tennis robot is proposed. Firstly, we introduce the kinetic analysis model for the ball. Based on the model, filters for both the captured ball status and the predicted hitting points are designed to fuse the original captured data. Furthermore, we improve the estimation method for the ball's angular velocity. The experiments confirmed the effectiveness and the robustness of the proposed algorithm, and demonstrated that it is sufficiently strong to generate stable and accurate prediction of the hitting point status.

REFERENCES

[1] Russell L Anderson. *A robot ping-pong player: experiment in real-time intelligent control*. MIT press, 1988.

[2] Zhengtao Zhang, De Xu, and Min Tan. Visual measurement and prediction of ball trajectory for table tennis robot. *IEEE Transactions on Instrumentation and Measurement*, 59(12):3195–3205, 2010.

[3] Akira Nakashima, Yuki Ogawa, Yosuke Kobayashi, and Yoshikazu Hayakawa. Modeling of rebound phenomenon of a rigid ball with friction and elastic effects. In *American Control Conference (ACC), 2010*, pages 1410–1415. IEEE, 2010.

[4] Yifeng Zhang, Yongsheng Zhao, Rong Xiong, Yue Wang, Jianguo Wang, and Jian Chu. Spin observation and trajectory prediction of a ping-pong ball. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 4108–4114. IEEE, 2014.

[5] Yanlong Huang, De Xu, Min Tan, and Hu Su. Trajectory prediction of spinning ball for ping-pong player robot. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3434–3439. IEEE, 2011.

[6] Hu Su, Zaojun Fang, De Xu, and Min Tan. Trajectory prediction of spinning ball based on fuzzy filtering and local modeling for robotic ping–pong player. *IEEE Transactions on Instrumentation and Measurement*, 62(11):2890–2900, 2013.

[7] Ping Yang and De Xu. Control system design of inserting machine and its high precision alignment. *International Jouranal of innovative computing information and control*, 6(4):1595–1604, 2010.

[8] Jianran Liu, Zaojun Fang, Kun Zhang, and Min Tan. Improved high-speed vision system for table tennis robot. In *IEEE International Conference on Mechatronics and Automation (ICMA)*, pages 652–657, Aug 2014.

[9] Hailing Li, Haiyan Wu, Lei Lou, Kolja Kuhnlenz, and Ole Ravn. Ping-pong robotics with high-speed vision system. In *12th International Conference on Control Automation Robotics & Vision (ICARCV)*, pages 106–111. IEEE, 2012.

[10] Yifeng Zhang, Rong Xiong, Yongsheng Zhao, and Jian Chu. An adaptive trajectory prediction method for ping-pong robots. In *Intelligent Robotics and Applications*, pages 448–459. Springer, 2012.