Image Processing, Electronics and Computers
L. Trajkovic et al. (Eds.)
© 2024 The Authors.
This article is published online with Open Access by IOS Press and distributed under the terms of the Creative Commons Attribution Non-Commercial License 4.0 (CC BY-NC 4.0).
doi:10.3233/ATDE240463

Prediction of Table Tennis Trajectory Based on Optimized Unscented Kalman Filter Algorithm

Kun YAN[1]
College of General Education, Harbin Huade University, Harbin, Heilongjiang, China

Abstract. This study introduces an innovative trajectory prediction algorithm for table tennis, employing an optimized Unscented Kalman Filter (UKF) combined with a Simple Physical Motion (SPM) model. The conventional UKF algorithm, while effective in real-time predictions, often encounters significant deviations in short-term forecasts, especially when dealing with abrupt changes in a table tennis ball's motion. To address this, our approach integrates UKF with SPM, effectively predicting the ball's trajectory pre- and post-collision. The method begins by using UKF to predict the ball's trajectory and landing point before collision, taking into account factors such as air resistance, gravity, and the Magnus force caused by the ball's rotation. After collision, the trajectory is forecasted using a simplified collision rebound model and a kinematic model. This dual-phase approach significantly reduces trajectory prediction errors post-collision. This algorithm's practical application is demonstrated in a constructed table tennis robot system, highlighting its superior real-time performance and accuracy, particularly in post-collision trajectory prediction. This makes it a valuable tool for advanced table tennis training and robotic interaction systems. This study contributes to the field of machine vision and robotic interaction by presenting a more efficient and accurate method for trajectory prediction, particularly in dynamic environments like table tennis. The algorithm's lower hardware requirements, combined with its robustness and simplicity, underscore its potential in broader applications where accurate real-time trajectory prediction is crucial. This development not only advances the field of sports robotics but also has implications for various industrial and research applications where precise object tracking and prediction are essential.

Keywords. Trajectory prediction, unscented kalman filter, collision rebound model, kinematic model

1. Introduction

There has been rapid development in machine vision technology both domestically and internationally [1]. Recognition and tracking of target objects are crucial in machine vision research [2]. However, solely relying on target detection and tracking cannot accurately predict the movement state and position of objects in the next moment, making trajectory prediction a focal and challenging aspect in the field of machine vision [3].

Currently, common trajectory prediction methods include: dynamic trajectory prediction based on traditional polynomial fitting [4], dynamic trajectory prediction based on Kalman filtering, and dynamic trajectory prediction based on neural networks [5].

[1] Corresponding Author: Kun YAN, meng_216@163.com.

Among these, dynamic trajectory prediction based on traditional polynomial fitting involves fitting a kth-order polynomial with k+1 parameters to approximate the function curve, minimizing the error to obtain the predicted function curve of the moving object's trajectory. This method is simple, easy to implement, and suitable for small data sets and simple trajectory predictions [6]. However, its accuracy is not high, and the system diverges with increasing order of fitting. Dynamic trajectory prediction based on Kalman filtering updates the estimation of state variables for the next moment's trajectory position by utilizing the previous moment's estimated value and the current moment's observed value, adapting well to frequently changing motion targets but unable to perform long-term trajectory prediction [7]. Meanwhile, dynamic trajectory prediction based on neural networks can effectively learn the historical motion states of moving targets, establish the mapping relationship between input and output through machine learning, achieve trajectory prediction, approximate nonlinear systems accurately, but requires a large amount of data to train the model, and has poor real-time performance [8].

In recent years, research on predicting the trajectory of ping-pong balls has primarily focused on non-spinning balls, while studies on spinning balls have been more centered on their recognition and classification, or utilizing machine learning methods for predicting the trajectory of spinning balls. Reference [9] proposed a novel physical bounce model for trajectory prediction of ping-pong balls, which depends on the friction coefficient and vertical restitution coefficient, based on a high-speed vision system composed of four high-speed 1394 firewire cameras and a powerful industrial PC. Reference [10] constructed an integrated vision system to observe the spinning motion of ping-pong balls and utilized position and spin information to predict trajectories through an extended Kalman filter (EKF). Reference [11] proposed a trajectory prediction scheme using backpropagation neural networks (BPN), where two neural networks were trained to learn the flight parabolas before and after collision, predicting trajectories based on ten points of the first parabola. Reference [12] predicted the trajectory of ping-pong balls by constructing an aerodynamics model and a rebound model, and calculated the state of hitting points based on the predicted trajectory.

The previously mentioned methods have enhanced trajectory prediction accuracy to a certain extent, but they face challenges such as high hardware requirements, complex algorithms, and neglect of rotational ball angular velocity. In this paper, a trajectory prediction algorithm based on Unscented Kalman Filter (UKF) and kinematic model under binocular vision is proposed. Firstly, the trajectory before collision is predicted using UKF; after predicting the landing point, collision models and kinematic models are utilized to predict the trajectory after collision. This method achieves more accurate prediction of ping-pong ball trajectory with lower hardware requirements for the visual system. Experimental validation shows that this method offers higher real-time performance and more accurate post-collision trajectory prediction of ping-pong balls.

2. Improved Ping-Pong Ball Trajectory Prediction Algorithm Based on UKF

Predicting the trajectory of a ping-pong ball involves forecasting the ball's trajectory based on partial three-dimensional world coordinates to determine its landing point. The requirements for ping-pong trajectory prediction mainly revolve around real-time performance and accuracy. While the traditional polynomial fitting method performs well in terms of real-time performance, it yields poor results for predicting the trajectory of spinning ping-pong balls [13]. On the other hand, neural network-based trajectory prediction methods can effectively predict the trajectory of spinning ping-pong balls but require a large amount of data for model training. This paper proposes a ping-pong ball trajectory prediction algorithm based on Unscented Kalman Filter (UKF) and kinematic model, which can simultaneously ensure real-time performance and accurate prediction of spinning ball trajectories.

2.1. Prediction of Ping-Pong Ball Trajectory before Collision Based on UKF

During the motion of a ping-pong ball, due to the force exerted by the paddle on the ball during the hitting moment, known as the Magnus force, may not necessarily pass through the center of the ball, causing the ping-pong ball to rotate during its flight. Rotating ping-pong balls have their own angular velocity. Ignoring this angular velocity and treating the ball as a point mass can result in significant errors in trajectory prediction. During flight, rotating ping-pong balls are primarily influenced by three forces: gravity, air resistance, and Magnus force. The main factor affecting the Magnus force is the angular velocity. However, directly obtaining the angular velocity requires equipping with ultra-high-speed cameras or infrared devices, which are expensive and difficult to ensure experimental conditions [14]. Therefore, this paper adopts UKF to indirectly estimate this unknown state. UKF is a nonlinear Gaussian state estimator based on the minimum variance estimation criterion, which uses Unscented Transform (UT) to approximate the posterior mean and covariance of the system state, requiring shorter estimation time and higher accuracy [15].

Discrete nonlinear systems can be represented by a process equation and an observation equation, which can be mathematically expressed as:

$$\begin{cases} X_{k+1} = F(X_k, U_k) + W_k \\ Z_k = H(X_k) + V_k \end{cases} \tag{1}$$

Here, F is the state process function, H is the observation process function, U_k is the input vector at time k, Z_k is the observation vector at time k, W_k and V_k are the process noise and measurement noise at time k respectively, both uncorrelated and following Gaussian white noise, and X_k is the motion state vector at time k.

2.1.1. Construction of system process equation

For a rotating ping-pong ball, the process equation is nonlinear while the observation equation is linear. Using a 9-dimensional vector to represent the state variables of the ping-pong ball, the specific formula is as follows:

$$X = \left[p_x, p_y, p_z, v_x, v_y, v_z, \omega_x, \omega_y, \omega_z \right]^{\mathrm{T}} \tag{2}$$

Where p_x, p_y, p_z represent the coordinates of the ping-pong ball along the x, y, z axes, respectively; v_x, v_y, v_z represent the velocities of the ping-pong ball along the x, y, z axes, respectively; ω_x, ω_y, ω_z represent the angular velocities of the ping-pong ball along the x, y, z axes, respectively.

The observation vector is represented by a three-dimensional vector, which is the three-dimensional coordinate information of the ping-pong ball obtained through binocular vision, as shown in equation (3):

$$Z = \left[p_x, p_y, p_z \right]^{\mathrm{T}} \tag{3}$$

Define $[p_{x,k}, p_{y,k}, p_{z,k}, v_{x,k}, v_{y,k}, v_{z,k}, \omega_{x,k}, \omega_{y,k}, \omega_{z,k}]^{\mathrm{T}}$ as the 9-dimensional state vector of the system at time k. Construct the system process equations for the ping-pong ball motion model, and the equations are listed as follows:

$$\begin{bmatrix} p_{x,k} \\ p_{y,k} \\ P_{z,k} \end{bmatrix} = \begin{bmatrix} P_{x,k-1} \\ p_{y,k-1} \\ P_{z,k-1} \end{bmatrix} + \begin{bmatrix} v_{x,k-1} \\ v_{y,k-1} \\ v_{z,k-1} \end{bmatrix} \Delta t + \begin{bmatrix} W_{1,k} \\ W_{2,k} \\ W_{3,k} \end{bmatrix} \tag{4}$$

$$\begin{aligned} \begin{bmatrix} v_{x,k} \\ v_{y,k} \\ v_{z,k} \end{bmatrix} = \begin{bmatrix} v_{x,k-1} \\ v_{y,k-1} \\ v_{z,k-1} \end{bmatrix} + k_1 \| V \| \begin{bmatrix} v_{x,k-1} \\ v_{y,k-1} \\ v_{z,k-1} \end{bmatrix} \Delta t + \\ k_2 \begin{bmatrix} \omega_{x,k-1} \\ \omega_{y,k-1} \\ \omega_{z,k-1} \end{bmatrix} \times \begin{bmatrix} v_{x,k-1} \\ v_{y,k-1} \\ v_{z,k-1} \end{bmatrix} \Delta t - \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} \Delta t + \begin{bmatrix} W_{4,k} \\ W_{5,k} \\ W_{6,k} \end{bmatrix} \end{aligned} \tag{5}$$

Where $W_{i,k}$ ($i = 1, 2, 3, \ldots, 9$) are the components of the 9-dimensional zero-mean process noise vector at time k; k_1 is the coefficient for air resistance; k_2 is the coefficient for Magnus force; $\|V\|$ represents the magnitude of the velocity vector in three directions at time k-1.

Furthermore, it is derived that the mathematical formula for k_1 can be written as:

$$k_1 = -\frac{1}{2m} C_D \rho A \tag{6}$$

Where C_D depends on the surface roughness of the ping-pong ball; ρ is the air density; A is the maximum cross-sectional area of the ping-pong ball.

Simultaneously, it is derived that the mathematical formula for k_2 can be written as:

$$k_2 = \frac{C_L \rho D^3}{2\pi m} \tag{7}$$

Where C_L represents the lift coefficient; D represents the diameter of the ping-pong ball; m represents the mass of the ping-pong ball.

In conclusion, the angular velocity of the ping-pong ball in the x, y, and z coordinate axes at time k can be calculated using the following formula:

$$\begin{bmatrix} \omega_{x,k} \\ \omega_{y,k} \\ \omega_{z,k} \end{bmatrix} = \begin{bmatrix} \omega_{x,k-1} \\ \omega_{y,k-1} \\ \omega_{z,k-1} \end{bmatrix} + \begin{bmatrix} W_{7,k} \\ W_{8,k} \\ W_{9,k} \end{bmatrix} \tag{8}$$

Hence, the construction of the system process equations for the ping-pong ball motion model is completed.

2.1.2. System observation equation research

The system observation equation for the ping-pong ball motion model is represented as:

$$\begin{bmatrix} z_{x,k} \\ z_{y,k} \\ z_{z,k} \end{bmatrix} = \begin{bmatrix} \rho_{x,k} \\ \rho_{y,k} \\ \rho_{z,k} \end{bmatrix} + \begin{bmatrix} \boldsymbol{V}_{1,k} \\ \boldsymbol{V}_{2,k} \\ \boldsymbol{V}_{3,k} \end{bmatrix} \tag{9}$$

Where $V_{i,k}$ (i = 1, 2, 3) is the three-dimensional zero-mean observation noise vector at time k.

W_k is linearly independent of V_k, and satisfies equation (10):

$$\begin{cases} E\{W_k^T W_j^T\} = \delta_{ij} R_k, \forall k, j \\ E\{V_k^T V_j^T\} = \delta_{ij} Q_k, \forall k, j \end{cases} \tag{10}$$

The UT transformation is based on weighted statistical linear regression to calculate the posterior distribution of random variables. A certain number of points are selected according to the prior distribution of random variables ($\tilde{\boldsymbol{\sigma}}$, P_σ), and the values of these points after nonlinear transformation are calculated; using weighted linear regression to linearize nonlinear functions of random variables.

This article adopts a symmetric sampling strategy. In this strategy, the number of σ points sampled is L=2×N+1, where N is the dimension of the state variables, and the proportion factor θ of the mean σ and the distance between σ points is $\theta=\alpha^2(9+\lambda)-9$. Here, α represents the distribution distance of σ points, and λ is the proportion coefficient. The mathematical formulas required for the symmetric sampling strategy mentioned in the article are:

$$\begin{cases} \sigma_0 = \tilde{\sigma} \\ \sigma_{i-1} = \left[\tilde{\sigma}_{i-1}, \tilde{\sigma}_{i-1} \pm \sqrt{\theta+9} \left(\sqrt{P_{i-1}} \right) \right] \end{cases} \tag{11}$$

Where $\sqrt{\boldsymbol{P_{i-1}}}$ is the square root of the covariance.

Next, the Cholesky decomposition method will be used to calculate $\sqrt{\boldsymbol{P_{i-1}}}$. When $P_{i-1} = A^T A$, A represents the lower triangular matrix with positive diagonal elements, and the i-th row of A is taken; when $P_{i-1} = AA^T$, the i-th column of A is taken.

Calculate the mean weight $\boldsymbol{W}_{(\boldsymbol{i})}^{(\boldsymbol{m})}$ and the variance weight $\boldsymbol{W}_{(\boldsymbol{i})}^{(\boldsymbol{c})}$ of the σ points. When $i \neq 0$, they are equal, which can be represented as:

$$W_{(i)}^{(m)} = W_{(i)}^{(c)} = \frac{1}{2(\theta+9)}, i \neq 0 \tag{12}$$

In this section, the computation of initial values can be described as follows:

(1) The initial value of the mean weight $\boldsymbol{W}_{(\boldsymbol{0})}^{(\boldsymbol{m})}$ is calculated using the following formula:

$$W_{(0)}^{(m)} = \frac{\theta}{\theta+9} \tag{13}$$

(2) The initial value of the variance weight $\boldsymbol{W}_{(\boldsymbol{0})}^{(\boldsymbol{c})}$ is calculated using the following formula:

$$W_{(0)}^{(c)} = \frac{\theta}{\theta^2+9} + (1-\alpha^2+\beta) \tag{14}$$

Where β represents the distribution information of the sampled points.

According to the σ-point sampling strategy, σ points $\sigma_{k\text{-}1}$ (k=1, 2,.., L) are computed based on the mean $\tilde{\boldsymbol{\sigma}}(k\text{-}1|k\text{-}1)$ and estimated covariance $P(k\text{-}1|k\text{-}1)$ at time step k-1, propagated through the state function F to obtain $\boldsymbol{\sigma}_{(\boldsymbol{k}|\boldsymbol{k}-\mathbf{1})}$. The mathematical expressions can be written as follows:

$$\sigma_{(k|k-1)} = F(\sigma_{k-1}, U_{k-1}) \tag{15}$$

The formula for computing the next state prediction value $\sigma_{(k|k\text{-}1)}$ from $\tilde{\boldsymbol{\sigma}}(k\text{-}1|k\text{-}1)$ is derived as follows:

$$\tilde{\sigma}(k \mid k-1) = \sum_{i=0}^{18} W_i^m \sigma_{(k|k-1)} \tag{16}$$

And the next step error covariance $P(k\text{-}1|k\text{-}1)$ is computed according to the mathematical definition:

$$\begin{aligned} P(k \mid k-1) = &\sum_{i=0}^{18} W_i^c \left(\sigma_{(k|k-1)} - \tilde{\sigma}(k \mid k-1)\right) \\ &\times \left(\sigma_{(k|k-1)} - \tilde{\sigma}(k \mid k-1)\right)^T + Q_{k-1} \end{aligned} \tag{17}$$

Similarly, using $\tilde{\boldsymbol{\sigma}}(k\text{-}1|k\text{-}1)$ and $P(k|k\text{-}1)$ according to the sampling strategy, compute $\boldsymbol{\sigma}_{(k|k-1)}$, propagate through the measurement function H to obtain $\boldsymbol{z}_{(k|k-1)}$), then calculate the weighted measurement prediction value $\tilde{\boldsymbol{z}}_{(k|k-1)}$, which can be represented as:

$$z_{(k|k-1)} = H \mid (\sigma_{(k|k-1)}) \tag{18}$$

Compute the error covariance $\boldsymbol{P}_{\tilde{z}\tilde{z}}(k|k\text{-}1)$, which can be written as follows:

$$\begin{aligned} P_{\tilde{z}\tilde{z}}\left(k \mid k-1\right) = \sum_{i=0}^{18} W_i^c \left(z_{(k\|-1)} - \tilde{z}\left(k \mid k-1\right)\right) \\ \times\left(z_{(k|k-1)} - \tilde{z}\left(k \mid k-1\right)\right)^{\mathrm{T}} + R_{k-1} \end{aligned} \tag{19}$$

Compute the cross-covariance $\boldsymbol{P}_{\tilde{\sigma}\tilde{z}}(k|k\text{-}1)$, which can be determined by the following equation:

$$\begin{aligned} P_{\tilde{\sigma}\tilde{z}}\left(k \mid k-1\right) = \sum_{i=0}^{18} W_i^c \left(\sigma_{(k|k-1)} - \tilde{\sigma}\left(k \mid k-1\right)\right) \\ \times\left(z_{(k|k-1)} - \tilde{z}\left(k \mid k-1\right)\right)^{\mathrm{T}} \end{aligned} \tag{20}$$

After obtaining the new measurement value z_k, compute the Kalman gain $K(k)$, and update the state value $\tilde{\boldsymbol{\sigma}}(k|k)$ and error covariance $P(k|k)$ for the next time step, the derived mathematical formulas are as follows:

$$\begin{cases} K\left(k\right) = P_{\tilde{\sigma}\tilde{z}}\left(k \mid k-1\right) \\ \quad \times P_{\tilde{z}\tilde{z}}^{-1}\left(k \mid k-1\right) \\ \tilde{\sigma}\left(k \mid k\right) = \tilde{\sigma}\left(k \mid k-1\right) + K\left(k\right)\times\left(z_k - \tilde{z}\left(k \mid k-1\right)\right) \\ P\left(k \mid k\right) = P\left(k \mid k-1\right) \\ \quad -K\left(k\right)\times P_{\tilde{z}\tilde{z}}\left(k \mid k-1\right)\times K^{\mathrm{T}}\left(k+1\right) \end{cases} \tag{21}$$

2.2. Table Tennis Ball Trajectory Prediction after Collision Based on Kinematics

Swiftly acquiring the trajectory of the table tennis ball post-collision is paramount to ensuring that the mechanical arm has ample time to adjust its posture and position for striking the ball back. The altered motion state of the table tennis ball after collision leads to longer convergence times and increasing prediction errors in traditional Unscented Kalman Filter (UKF) trajectory predictions [16]. This paper employs a simplified post-bounce physical motion model (SPM) to predict the trajectory of the table tennis ball after collision.

Given the marginal impact of the Magnus force on the physical motion model of the table tennis ball and its complexity in construction, we omit it when establishing the post-collision physical model. The force analysis during the motion of the table tennis ball is depicted in Figure 1.

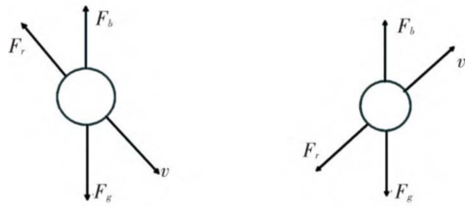


Figure 1: Force analysis diagram during table tennis movement.

In Figure 1, F_g represents the gravitational force acting on the table tennis ball; F_r denotes air resistance; C_d is the air resistance coefficient; F_b stands for the buoyant force due to air; v represents the velocity of the table tennis ball.

Due to the negligible magnitude of the buoyant force, it is not included in the subsequent model derivation. The mathematical formulas for the forces acting on the table tennis ball are specified as follows:

$$\begin{cases} F_g = mg \\ F_r = \frac{1}{2} C_d \rho A v^2 \\ F_b = \frac{1}{6} \rho g \pi D^3 \end{cases} \tag{22}$$

Where C_d denotes the air resistance coefficient; ρ denotes air density; A denotes the maximum cross-sectional area of the table tennis ball; D denotes the diameter of the table tennis ball.

Since the buoyant force is negligible, it is not included in the subsequent model derivation.

For the convenience of force analysis and kinematic model derivation along the x, y, and z axes, let $\boldsymbol{k = \frac{1}{2} C_d \rho A}$. Based on the force analysis of the table tennis ball on the x-axis and Newton's second law, equation (23) can be obtained:

$$m \frac{\mathrm{d}v_x}{\mathrm{d}t} = -k v_x^2 \tag{23}$$

Integrating equation (23), we derive the mathematical formula for the x-axis velocity v_x of the table tennis ball at time t:

$$v_x = \frac{v_{x0} m}{k v_{x0} t + m} \tag{24}$$

Where v_{x0} represents the initial velocity of the table tennis ball along the x-axis at time t=0.

The formulas for the y-axis velocity v_y and the y-axis coordinate p_y of the table tennis ball at time t are as follows:

$$\begin{cases} v_y = \dfrac{v_{y0}m}{kv_{y0}t+m} \\ p_y = \dfrac{m}{k}\ln\left(1+\dfrac{kv_{y0}t}{m}\right)+y_0 \end{cases} \tag{25}$$

Where v_{y0} represents the initial velocity of the table tennis ball along the y-axis at time $t = 0$, and y_0 represents the initial position coordinate of the table tennis ball along the y-axis at time t=0.

In the z-axis direction, the table tennis ball experiences air resistance and gravity. When v_z>0, based on force analysis and Newton's second law for the table tennis ball, we have equation (26):

$$m\frac{\mathrm{d}v_z}{\mathrm{d}t} = -mg - kv_z^2 \tag{26}$$

Integrating equation (26), we obtain the mathematical formula for the z-axis velocity v_z of the table tennis ball at time t:

$$v_z = \sqrt{\frac{mg}{k}}\tan\left[-\sqrt{\frac{mg}{k}}+\arctan\left(v_{z0}\sqrt{\frac{mg}{k}}\right)\right] \tag{27}$$

Where v_{z0} represents the initial velocity of the table tennis ball along the z-axis at time t=0.

Integrating the velocity v_z in equation (27) from 0 to t, we can calculate the z-axis coordinate p_z of the table tennis ball at time t using the following equation:

$$p_z = \frac{m}{k}\ln(|\cos(\sqrt{\frac{mg}{k}}t)+v_{z0}\sqrt{\frac{mg}{k}}\sin(\sqrt{\frac{mg}{k}}t)|) \\ +z_0 \tag{28}$$

Where z_0 represents the initial position coordinate of the table tennis ball on the z-axis at time $t = 0$.

When $v_z \leqslant 0$, according to the force analysis of the table tennis ball and Newton's second law, equation (29) is obtained:

$$m\frac{\mathrm{d}v_z}{\mathrm{d}t} = -mg + kv_z^2 \tag{29}$$

Integrating equation (29) yields:

$$v_z = \sqrt{\frac{mg}{k}} \frac{1 - e^{2t\sqrt{\frac{mg}{k}}}}{1 + e^{2t\sqrt{\frac{mg}{k}}}} + v_{z0} \tag{30}$$

Integrating the velocity v_z in equation (30) from 0 to t gives the coordinate p_z of the table tennis ball on the z-axis at time t, with the computational formula shown in equation (31):

$$p_z = -\frac{m}{2k} \ln \frac{(1 + e^{2t\sqrt{\frac{kg}{m}}})}{4e^{2t\sqrt{\frac{kg}{m}}}} + z_0 + v_{z0} \tag{31}$$

When the table tennis ball collides with the tabletop, the coefficients of friction and attenuation in the horizontal and vertical directions differ due to differences in ball speed and motion. Constructing a rebound model for the table tennis ball, the mathematical formulas for calculating the velocities v_{xout}, v_{yout}, and v_{zout} of the table tennis ball after rebounding on the x, y, and z axes, respectively, are derived as follows:

$$\begin{cases} v_{xout} = k_x v_{xin} + b_x \\ v_{yout} = k_y v_{yin} + b_y \\ v_{zout} = k_z v_{zin} \end{cases} \tag{32}$$

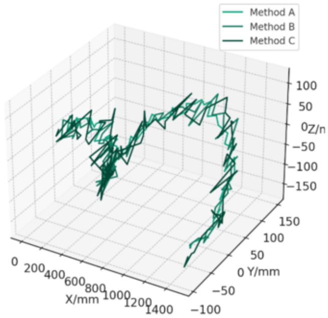
Where k_x, k_y, k_z, b_x, and b_y represent the fitting parameters for rebound, and v_{xin}, v_{yin}, and v_{zin} represent the velocities of the table tennis ball before rebounding along the x, y, and z axes, respectively.

By fitting the trajectories of table tennis ball collision rebounds, sampling multiple trajectories of table tennis ball rebounds, recording coordinates before and after rebounds, and using the least squares method for fitting to determine the rebound fitting parameters.

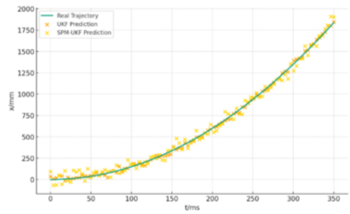
3. Comparative Analysis of Experimental Results

The effectiveness of the algorithm proposed in this study is validated through the construction of a high-speed table tennis system. The image acquisition device in the table tennis robot system is a USB industrial camera with a frame rate of 120 fps. The data sampling interval for trajectory prediction is 20 ms. The comparison of trajectory prediction results between the traditional UKF method and the SPMUKF method is illustrated in Figure 2.

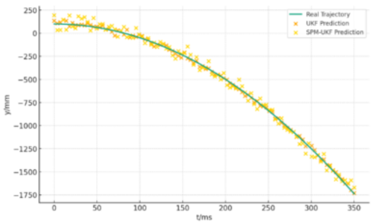
As depicted in Figure 2, the error of the traditional UKF prediction method does not exceed 10 mm on the x-axis, and remains within 20 mm on the y and z axes. Due to the continuous correction of state estimation quantities required by the UKF algorithm to improve the prediction of the next point's position, significant errors may arise in trajectory prediction when only partial data are input over time. However, the refined UKF table tennis trajectory prediction algorithm, by modeling the physical model, reduces the overall error in trajectory prediction after the table tennis ball rebounds. The errors in the *x* and *z* axes are kept within 10 mm, and within 15 mm in the *y*-axis direction. Compared to the traditional UKF prediction method, there is a reduction in error to a certain extent.



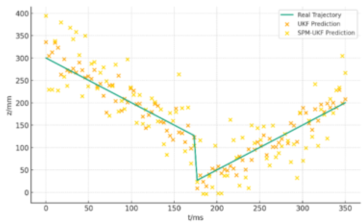
(a) Three-dimensional trajectory prediction comparison chart



(b) x-axis trajectory prediction comparison chart



(c) *y*-axis trajectory prediction comparison chart



(d) *z*-axis trajectory prediction comparison chart

Figure 2. Illustrates the comparison of results from different algorithms.

Through simulation results, it can be observed that during the first turning maneuver, both UKF and EKF algorithms demonstrate good tracking performance, with no significant differences observed. However, during the second turning maneuver, noticeable deviations in tracking occur, with the UKF algorithm outperforming the EKF algorithm, albeit insignificantly. Subsequent to the third turning maneuver, significant differences emerge, with the EKF algorithm's tracking trajectory gradually diverging from the true trajectory, leading to a notable decrease in accuracy. To comprehensively assess the tracking accuracy of the UKF algorithm, 100 Monte Carlo simulations were conducted, yielding the error curve depicted in Figure 3.

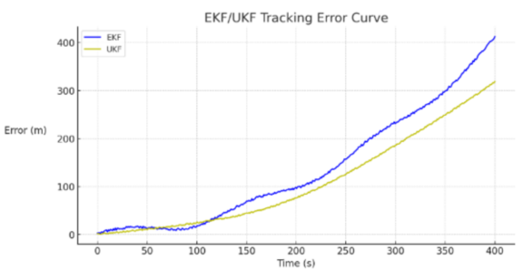


Figure 3. Error tracking curves for EKF/UKF.

Figure 3 indicates that during the first turning maneuver, the tracking errors of both methods are nearly synchronous. However, as the second turning commences, the errors begin to diverge and gradually widen. It is evident that the tracking accuracy of the UKF algorithm exceeds that of the EKF algorithm.

Figure 4 illustrates the computation time for 100 Monte Carlo simulations.

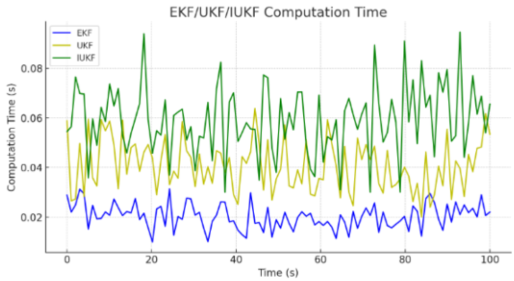


Figure 4. Computation time for EKF/UKF/IUKF.

Regarding the issue of high computational complexity associated with the UKF algorithm, enhancements were implemented. To maintain filtering precision, an investigation was conducted into the optimal sampling point-to-center distance for the system, alongside parameter tuning for proportionate correction denoted by α. By varying α values, the full-course average tracking error was computed, with results detailed in Table 1.

Table 1. The influence of parameter α on tracking accuracy.

Parameter α	<0.3	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
RMSE/m	NA	520	786	983	1288	1460	995	848	199
Parameter α	0.91	0.92	0.93	0.94	0.95	0.96	0.97	0.98	0.99
RMSE/m	806	756	665	485	352	180	127	106	112

Table 1 reveals that when $10^{-4} \leq \alpha < 0.3$, the covariance matrix becomes non-positive definite, rendering it unsuitable for the system. Conversely, when$0.3 \leq \alpha \leq 1.0$, the algorithm operates normally, with the peak average error occurring around 0.7, sharply decreasing at 0.9, and ultimately plateauing in the range (0.97, 0.99) to reach a minimum. The average error fluctuates insignificantly around 0.98, leading to the selection of the system parameter α=0.98. A comparative analysis of the tracking performance of the enhanced UKF (IUKF) algorithm is depicted in Figure 5.

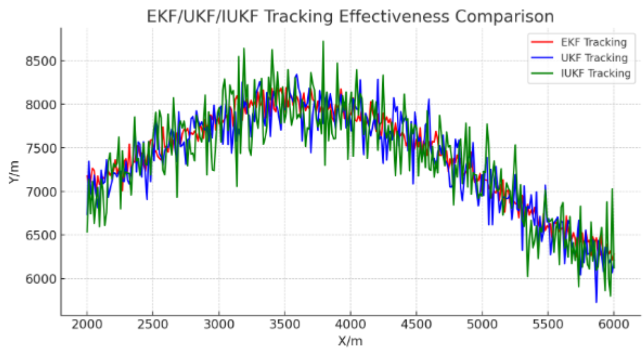


Figure 5. Comparison of tracking performance of EKF/UKF/IUKF.

Figure 5 illustrates that although the precision of the IUKF algorithm experiences a slight decline, it remains consistently superior to the EKF algorithm overall. Furthermore, Figure 6 presents the error comparison of EKF, UKF, and IUKF algorithms across 100 Monte Carlo simulations.

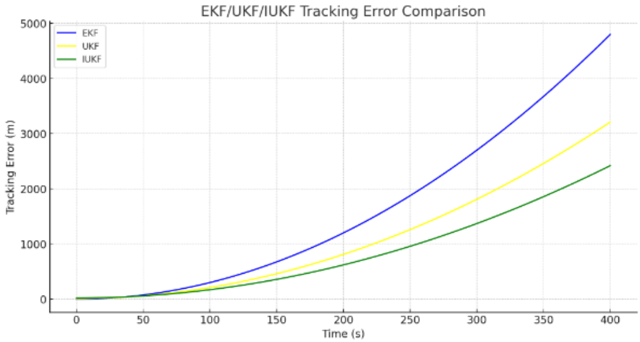


Figure 6. Tracking error curves of EKF/UKF/IUKF.

The findings suggest that the error of the IUKF algorithm is marginally higher than that of the UKF algorithm, although the disparity is not pronounced. However, the computational burden of the IUKF algorithm notably diminishes, with computation time slashed by over 50%, yielding the anticipated outcome. The computation time for 100 Monte Carlo simulations of the IUKF algorithm is depicted in Figure 4.

4. Conclusions

This paper tackles the problem of escalating prediction errors in table tennis ball trajectory after collision when using the traditional UKF algorithm. An enhanced UKF table tennis ball trajectory prediction algorithm is introduced. It employs the UKF algorithm to predict the trajectory before collision and utilizes a simple physical motion model to forecast the trajectory after collision. Experimental findings confirm that the improved UKF algorithm can enhance trajectory prediction accuracy to a certain degree, thereby ensuring the smooth progress of subsequent table tennis robot hitting tasks.

The current enhancement primarily addresses the pre- and post-collision phases separately. Integrating these phases more seamlessly may present a challenge due to the increase in model complexity. The simple physical motion model might not account for all the nuances of the ball's interaction with the environment, such as air resistance and spin effects. While the improved UKF algorithm shows better performance, its adaptability to different playing styles and conditions has not been extensively tested.

Future work could explore more sophisticated integration methods that combine the UKF predictions with the physical motion model in a more unified framework. Developing more complex models that capture additional physical aspects of the ball's trajectory, such as spin-dependent aerodynamics, could improve prediction accuracy further. Incorporating machine learning techniques to learn from a variety of playing styles may enhance the generalization of the trajectory prediction. It's crucial to optimize the computational efficiency to ensure the algorithm can operate in real-time, a necessity for table tennis robots.

References

[1] Li H, Ali S G, Zhang J, et al. Video-based table tennis tracking and trajectory prediction using convolutional neural networks. Fractals, 2022, 14(3): 15-18.
[2] Bai S, Zhang J. Research on Stage Automatic Light-following System Based on Trajectory Prediction.2021 International Conference on Culture-oriented Science & Technology (ICCST), 2021: 543-546.
[3] Yin J. The method of table tennis players' posture recognition based on a genetic algorithm. International Journal of Biometrics, 2021, 30(1): 61-72.
[4] Pan N, Wang Y, Tong J. Analysis and Improvement of Tennis Motion Recognition Algorithm Based on Human Body Sensor Network. Mobile information systems, 2022, 169(1): 8188571.
[5] Tang H. Parabolic Detection Algorithm of Tennis Serve Based on Video Image Analysis Technology. Security and Communication Networks, 2021, 43(7): 930-936.
[6] Zhifei X I, An X U, Yingxin K, et al. Target maneuver trajectory prediction based on RBF neural network optimized by hybrid algorithm. Systems Engineering and Electronics Technology: English Edition, 2021, 32(2): 19.
[7] Yu X. Evaluation of training efficiency of table tennis players based on computer video processing technology. Optik, 2023, 19(1): 90-94.
[8] Duan S, Meng L, Ma D, et al. Application Research on Roller Skater Detection, Tracking, and Trajectory Prediction Based on Video Stream. Sci. Program. 2021: 2702272.
[9] Wang Y, Wen M. "Simulation of tennis match scene classification algorithm based on adaptive Gaussian mixture model parameter estimation." Complexity 2021 (2021): 1-12.
[10] Omkar L J B, Koteswara R S, Kavitha L M. A Recognition Algorithm of Player Motion of Table Tennis Based on Improved 3D Convolution. Computer Science and Application, 2021, 11(6): 1791-1801.
[11] Sakile R, Sinha U K. Target Tracking Algorithm in Football Match Video Based on Deep Learning. Discrete Dynamics in Nature and Society, 2022, (3): 5.
[12] Jaganathan T, Panneerselvam A, Kumaraswamy S K. Object detection and multi-object tracking based on optimized deep convolutional neural network and unscented Kalman filtering. Concurrency and computation: practice and experience, 2022: 142.
[13] Dahal P, Mentasti S, Arrigoni S, et al. Fuzzy Dynamic Obstacle Avoidance Algorithm for Basketball Robot Based on Multi-Sensor Data Fusion Technology. International Journal of Foundations of Computer Science, 2022, 14(3): 750.
[14] Icallo I, Lopez-Iturri P, Astrain J J, et al.Research on Basketball Goal Recognition Based on Image Processing and Improved Algorithm. Secur. Commun. Networks, 2021 (1): 73.
[15] Divya G N, Rao S K. Target Tracking Algorithm of Basketball Video Based on Improved Grey Neural Network. Scientific programming, 2021(1): 15.
[16] Choi D, Yim J, Baek M, et al. Using Camshift and Kalman Algorithm to Trajectory Characteristic Matching of Basketball Players. Complexity, 2021, 2021(9): 1-11.