Robot Badminton

Jianyang Tang
TU Darmstadt

Honggang Gou
TU Darmstadt

Tim Staschewski
TU Darmstadt

Abstract

In this report, we research on the task of a KUKA robot arm playing badminton under SL simulator. Due to the irregularity of shuttlecock shape, dynamic analysis and trajectory prediction of the shuttlecock is uncertain. In order to estimate the shuttlecock relatively ideally, Kalman filter is introduced. Meanwhile playing badminton requires high velocity for hitting, which makes controlling the racket with robot arm more challenging. A combined controller based on two phases controlling is introduced. Overall a simple implementation under SL simulator is delivered.

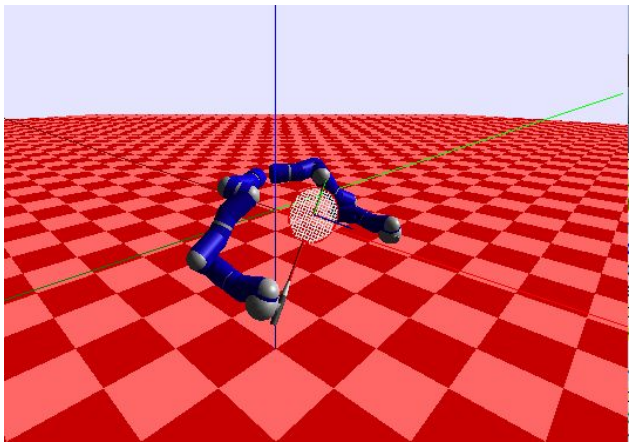

Figure 1: Robot arm with a racket

1 Introduction

To accomplish a humanoid behavior in robot scenario, a range of multidisciplinary techniques need to be implemented simultaneously. To tackle down this particular challenge we integrate diverse technologies from different areas of science.

In order to perform the desired behavior of hitting shuttlecock on a robot arm simulated under the environment of SL, this challenging task can be dissolved into several aspects of sub-issues: simulations of a badminton racket and the motion of a flying shuttlecock, simulation of kinematic features of the shuttlecock in real world, learning the trajectory of a flying shuttlecock, prediction of possible interception points and to make the robot arm end-effector move to he interception point and performing the hitting behavior.

Considering the limitations of robot arm motion speed and the short duration of shuttlecock flying motion, a prediction algorithm with fast convergence should be introduced to provide an estimated distribution of shuttlecock trajectory. With the prediction of possible interception point in advance, several controller are introduced in different stages of interception process.

2 Shuttlecock Trajectory Estimation

Shuttlecock in badminton is quite different from all those in regular ball sports, as a result there are more complicated factors to consider, which affects the prediction significantly. Self-spin of round objects is hard to detect with vision reception. As described by (Cooke, 1999) shuttlecock has relative less self-spin, but self-spin of shuttlecock causes turbulence, which easily change the orientation of shuttlecock. Another leading factor, air resistance, makes the shuttlecock kinematics even more complicated. For simplicity, we only consider air resistance and regard turbulence as model error.

Besides the vision system cannot really get accurate signals of realtime shuttlecock state, there is always error in the reception of signals.

However, the accuracy of shuttlecock trajectory prediction is very important for a good hitting performance. Following we introduce Kalman filter to smooth the real-time trajectory estimation.

Preliminary work. Under review by AISTATS 2012. Do not distribute.

2.1 Shuttlecock Kinematics

Based on some facts mentioned below, we may only consider the leading factor, air drag. Air buoyancy of a standard shuttlecock with a volume of 19 m^3 is about 0.02 gw, which is neglectable. There is no self-spin of shuttlecock, so we can also ignore the turbulence.

There are two kinds of resistance standard model, either proportional to speed or to the speed squared. By making a trade-off between complexity and execution time, this results the equation below

$$\vec{f} = -b\vec{v} \tag{1}$$

In equation (1) b is a resistance factor decided by the object itself as flowing

$$b = 6\pi\eta r \tag{2}$$

in which η is viscosity, for motions in air at temperature $15°C$ this value is $1.8 \times 10^{-5} kg/(ms)$

r is equivalent spherical diameter, for a shuttlecock with volume $V = 1.9 \times 10^{-5} m^3$ we have

$$r = \sqrt[3]{\frac{3V}{4\pi}} = 1.7 \times 10^{-2} m \tag{3}$$

Besides only gravity has influence on the shuttlecock motion. So according to Newton's law we have the shuttlecock acceleration equation.

$$m\vec{a} = \vec{f} + \vec{G} \tag{4}$$

For convenience of calculation, the acceleration of shuttlecock is decomposed into three dimensions. i.e., we transform formula (4) to

$$\begin{aligned} a_x &= \frac{-b}{m} v_x \\ a_y &= \frac{-b}{m} v_y \\ a_z &= \frac{-b}{m} v_z - g \end{aligned}$$

By integration operation on both side of equations above we can get a closed form solution of ball position as following.

$$\begin{aligned} x_t &= \frac{-mv_{x0}}{b}(1 - e^{\frac{-b}{m}t}) + x_0 \\ y_t &= \frac{-mv_{y0}}{b}(1 - e^{\frac{-b}{m}t}) + y_0 \\ z_t &= -(\frac{m^2 g}{b^2} + \frac{v_{z0} m}{b})e^{\frac{-b}{m}t} + \frac{mg}{b}t + \frac{m^2 g + mv_{z0} b}{b^2} + z_0 \end{aligned}$$

This closed form shuttlecock motion can also be used for prediction with least squared regression method. Take x_t as an example, regard $1 - e^{\frac{-b}{m}t}$ as x, $\frac{-mv_{x0}}{b}$ as $beta_1$, x_0 as $beta_2$, then we have

$$y = \beta_1 x + \beta_2 \tag{5}$$

After collecting a certain number of real time ball position data, let X be the list of result of observed variables, and Y be the dependent variables. By doing a least square regression we get parameter estimation

$$\hat{\beta} = (X^{\mathsf{T}} X)^{-1} X^{\mathsf{T}} Y \tag{6}$$

However the performance of least square regression in this case relies really on the shape of the closed form of shuttlecock motion. And our closed form model is based on some assumption, its accuracy has already been reduced when we use this model in practice. Therefore in next section, we introduce a better way to optimize the accuracy of shuttlecock motion prediction.

2.2 Trajectory Estimation by Kalman Filter

The following section covers our solution to smoothing the real-time estimation of the shuttlecock and the prediction of a proper hitting position.The shuttlecock approaches the set hitting point approximately one to two second after the launch, which requires the robot arm to start its movement in advance, and to make sure the measurement of the shuttlecock position is close to the true value,so we introduce a algorithm with fast convergence Kalman filter.

2.2.1 Model Statement

Estimation and prediction of the trajectory are both based on the kinematic equations of shuttlecock motion described as above. Take the air resistance into consideration, the accelerations of three axes are not constant,so we include accelerations in the state vector.

$$\hat{\mathbf{X}} = (x, y, z, \dot{x}, \dot{y}, \dot{z}, \ddot{x}, \ddot{y}, \ddot{z})^{\mathsf{T}} \tag{7}$$

2.2.2 State Space Description

The state vector considered relates to discrete-time linear systems on the form

$$\hat{\mathbf{X}}_{t+1} = A_t \hat{\mathbf{X}}_t + B_t u_t + v_t \tag{8}$$

$$z_t = H_t \mathbf{X}_t + e_t \tag{9}$$

Where $\mathbf{z}$ is the measurement of positions. The disturbance $\mathbf{v}$ and $\mathbf{e}$ are assumed to be white noise process with zero mean values. A_t denotes the state transition

Table 1: Kalman Filter Flow

Kalman Filter
Initialize: Q=const, R=const, P_0=const
For each iteration:
predict stage
update stage
Simulate: ball_predict ← state_vector
Output: ball_predict

matrix which can be derived from the kinematic equations described above. H_t denotes the measurement matrix which maps the state vector into measurement vector.In our case, we only care about the three dimensional positions of shuttlecock, then the measurement matrix will be

$$H_{t(3,9)} = \begin{bmatrix} 1 & 0 & 0\cdots & 0 \\ 0 & 1 & 0\cdots & 0 \\ 0 & 0 & 1\cdots & 0 \end{bmatrix}$$

Basically, the Kalman filter involves two stages:
Prediction stage

$$\hat{\mathbf{X}}_{t|t-1} = A_t\hat{\mathbf{X}}_{t-1|t-1} + B_t u_t \tag{10}$$

$$P_{t|t-1} = A_t P_{t-1|t-1} A_t^{\mathsf{T}} + Q_t \tag{11}$$

Update stage

$$\hat{\mathbf{X}}_{t|t} = \hat{\mathbf{X}}_{t|t-1} + K_t(z_t - H_t\hat{\mathbf{X}}_{t|t-1}) \tag{12}$$

$$P_{t|t} = P_{t|t-1} - K_t H_t Pt|t-1 \tag{13}$$

Where $\hat{\mathbf{X}}_{t|t}$denotes the posterior esimation of $\hat{\mathbf{X}}$, $\hat{\mathbf{X}}_{t|t-1}$ denotes the prior estimation of $\hat{\mathbf{X}}$, $P_{t|t}$is the covariance matrix of $\hat{\mathbf{X}}_{t|t}$, K_t denotes the Kalman gain in each iterations of Kalman filter, which is defined as below

$$K_t = P_{t|t-1}H_t^{\mathsf{T}}(H_t P_{t|t-1} H_t^{\mathsf{T}} + R_t)^{-1} \tag{14}$$

2.2.3 Implementation in SL

For the accomplishment of the estimation of a proper hitting position and also a distribution of the trajectory closer to the real distribution, the results of the previous sections are required. With the estimation of trajectory, the robot arm now just needs to adjusts its joints to the desired joints configuration by using a controller which we are going to discuss later.

As described in the Kalman Filter Flow table, the process of Kalman filter is executed in the order of initialization of relevant parameters, iterations in real-time servo and output.

First we have to state the initial conditions.The initial covariance propagation P_0 is given by the following formula:

$$P_0 = E[(X_0 - \hat{X_0})(X_0 - \hat{X_0})^{\mathsf{T}}] \tag{15}$$

Since there is no information about the initial state of the robot arm, we resort to trial and error methods, and this is what we specified about the magnitude of initial state covariance: 0.04.

With the input of a desired threshold of the hitting plane and real-time iterations executed, we can get a fusion of two probability distributions,the measurement distribution simulated by adding zero-mean gaussian noise into current actual state and the prediction distribution. Then we can get a relatively more accurate estimation of the shuttlecock trajectory.

Here is one thing which should not be neglected when implementing Kalman filter. As described by (Man,2009) , the difference between the magnitudes of process covariance matrix Q and measurement covariance matrix R has influence on the estimated states. Q, the process covariance . contributes to the overall uncertainty. When Q is large, the Kalman Filter tracks large changes in the data more closely than for smaller Q. R determines how much information from the measurement is used. If R is relatively high, the Kalman Filter considers the measurements as not very accurate. For smaller R, it will follow the measurements more closely.

3 Racket Controller

In this section, we work on controlling the robot arm to hit the shuttlecock. Firstly our target object is moving all the time, tracking a moving point is far more complicated than a fixed point. Another problem is that at the hitting moment, as by observation on other badminton playing robot or on real human-to-human playing, requires instant acceleration on racket, while bringing the racket to a desired position with a desired velocity.

3.1 Two Phases Controller

Before the shuttlecock comes close to the racket, badminton players doesn't move a lot. Instead they try to figure out, which point the shuttlecock shall fall and then try to locate himself as precisely as possible to a preparation point. Short time after this,

the badminton player does a precise analysis to find a point, for which there remains enough time to bring the racket to target.

Inspired by these observations we introduce a two phases racket controlling strategy as follows.

Preparation phase In this phase, we use shuttlecock motion prediction by Kalman filter to predict a position with a certain lower height, at where the shuttlecock will fall. Then we use transpose Jacobian controller to bring the racket to this preparation position.

Striking phase Our robot has to firstly estimate a target position, to which the racket and shuttlecock will arrive at the same time. However it's very difficult to find such a position along the shuttlecock trajectory. A better controller for this part is also more complicated than that in preparation stage, as not only position matters, but also the velocity. In order to get a steady controlling of robot arm end-effector, we use minimum jerk method to plan a smooth tracking trajectory of racket, other words to say the robot arm end-effector, while in our SL simulator there is a fixed projection between robot arm end-effector and badminton racket. By using minimum jerk controller we can also define a time duration for this tracking period, which solves the striking point estimation problem.

In Figure 2 a loop flow diagram for controller is given, this procedure will be called in every task servo loop.

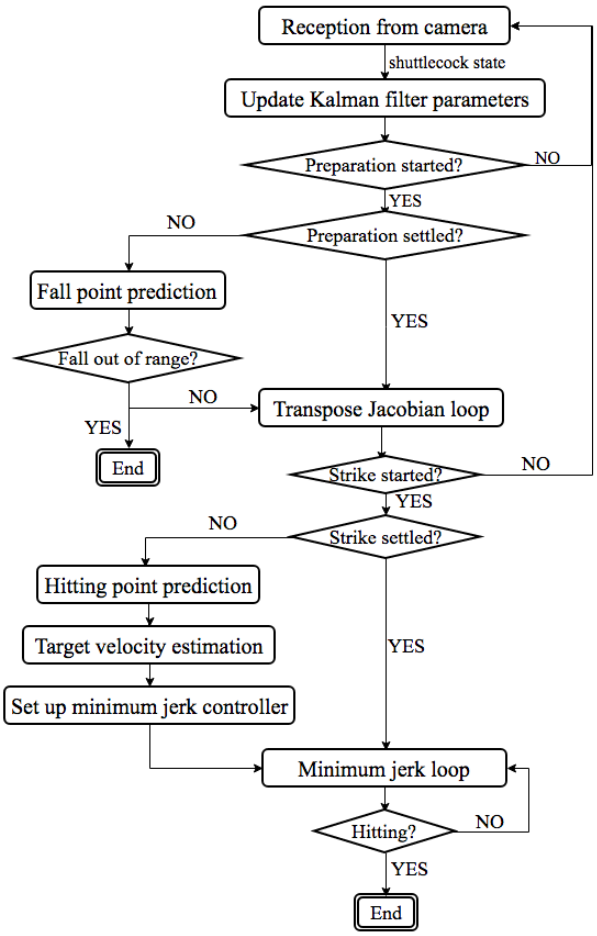

Figure 2: Controller loop flow disgram

3.2 Preparation with transpose Jacobian controller

As described by J. J.Craig (2005), the Jacobian matrix relates the joint space q with the end-effector position and orientation space X. Formally, a Jacobian is a set of partial differential equations:

$$J = \frac{\partial x}{\partial q} \tag{16}$$

$$\dot{x} = J\dot{q} \tag{17}$$

The basic idea of transpose Jacobian is very simple, we use the transpose of J instead of the inverse of J. Although transpose of Jacobian is not really the inverse, the correctness of using transpose Jacobian has already been proved in many papers. With transpose of J we have an update formula for joint space controller as below.

$$\dot{q} = \alpha J^T \dot{x} \tag{18}$$

where α is the learning rate.

Transpose Jacobian is sensible to learning rate, but it works still good in our case. As we don't have so much time constraints, we have relatively more time to bring the end-effector to the preparation position. In order to make controlling more stable, a relatively smaller gain will be enough and safe.

3.3 Striking phase with minimum jerk controller

3.3.1 Minimum jerk trajectories

The main idea of minimum jerk controller is to plan a trajectory from start joint configuration to target joint configuration with minimum jerk. After estimation of target racket state, we transfer this target state to joint state by inverse kinematic model. With start and target joint state we initialize the minimum jerk controller. Here we use a sixth order polynomial, which is safe for robot arm. The constants in the trajectory will be decided as follows.

Our minimum jerk trajectory has this form

$$x_t = a_0 + a_1\tau + a_2\tau^2 + a_3\tau^3 + a_4\tau^4 + a_5\tau^5 \qquad (19)$$

Note that $\frac{d\tau}{dt} = \frac{1}{T}$, where T is the total time duration. With (19) we have the first and second order derivation of x_t as follows.

$$\dot{x}_t = \frac{a_1}{T} + \frac{2a_2}{T}\tau + \frac{3a_3}{T}\tau^2 + \frac{4a_4}{T}\tau^3 + \frac{5a_5}{T}\tau^4 \qquad (20)$$

$$\ddot{x}_t = \frac{2a_2}{T^2} + \frac{6a_3}{T^2}\tau + \frac{12a_4}{T^2}\tau^2 + \frac{20a_5}{T^2}\tau^3 \qquad (21)$$

With boundary values x_0, $\dot{x}_0$, $\ddot{x}_0$, x_T, $\dot{x}_T$, $\ddot{x}_T$ we can get the constants as follows.

$$\begin{aligned}
a_0 &= x_0 \\
a_1 &= T\dot{x}_0 \\
a_2 &= \frac{T\ddot{x}_0}{2} \\
a_3 &= -\frac{3T^2}{2}\ddot{x}_0 - 6T\dot{x}_0 + 10(x_T - x_0) \\
a_4 &= \frac{3T^2}{2}\ddot{x}_0 + 8T\dot{x}_0 - 15(x_T - x_0) \\
a_5 &= -\frac{T^2}{2}\ddot{x}_0 - 3T\dot{x}_0 + 6(x_T - x_0)
\end{aligned}$$

3.3.2 Deciding target racket state

The task of playing badminton is not only to hit the ball at a certain position with a certain velocity, but also to make a effective strike, which means the player has to make the ball return in a way, such that the adversary will not easily hit it back.

Meanwhile, this impact point should also be easier for robot arm to reach. This can be achieved by our prediction model by Kalman filter in capital 2.

An effective estimation of the target racket state, more precisely to say the velocity and orientation of racket, will be quite helpful. In order to make the shuttlecock fall in a certain area in the adversary side, meanwhile the shuttlecock must also fly over the net, we have to estimate the desired outgoing velocity of shuttlecock.

Assume we know where on the adversary area the shuttlecock should fall, other words to say, at time T, the shuttlecock is at position x_T, y_T, z_T. With closed form shuttlecock trajectory in capital 2 we have now three equations.

$$x_T = \frac{-mv_{x0}}{b}(1 - e^{\frac{-b}{m}T}) + x_0 \qquad (22)$$

$$y_T = \frac{-mv_{y0}}{b}(1 - e^{\frac{-b}{m}T}) + y_0 \qquad (23)$$

$$z_T = -(\frac{m^2g}{b^2} + \frac{v_{z0}m}{b})e^{\frac{-b}{m}T} + \frac{mg}{b}T + \frac{m^2g + mv_{z0}b}{b^2} + z_0 \qquad (24)$$

In order to make sure that the shuttlecock flies over the net, we consider at time N (net), x_N the coordinate of shuttlecock is same as the x axis coordinate of net. And we only have to make sure that $z_N > net\ height$. Let net height be h_N, let s_N be a given safe distance from the shuttlecock to the net, we have two equations as below.

$$x_N = \frac{-mv_{x0}}{b}(1 - e^{\frac{-b}{m}N}) + x_0 \qquad (25)$$

$$\begin{aligned}
z_N = h_N + s_N = -(\frac{m^2g}{b^2} + \frac{v_{z0}m}{b})e^{\frac{-b}{m}N} \\
+ \frac{mg}{b}N + \frac{m^2g + mv_{z0}b}{b^2} + z_0
\end{aligned} \qquad (26)$$

Now we have five equations (22)-(26) with five unknown variables v_{x0}, v_{y0}, v_{z0}, T, N. Solve the equations we get desired outgoing velocity of shuttlecock.

Now we have desired outgoing velocity of shuttlecock, we only have to transfer it into desired racket velocity. To achieve this, we have to do an inverse impact analysis. The forward impact kinetics is modeled as follows.

In ideal case, shuttlecock goes back in a reflexive direction with same speed, due to air drag there is a

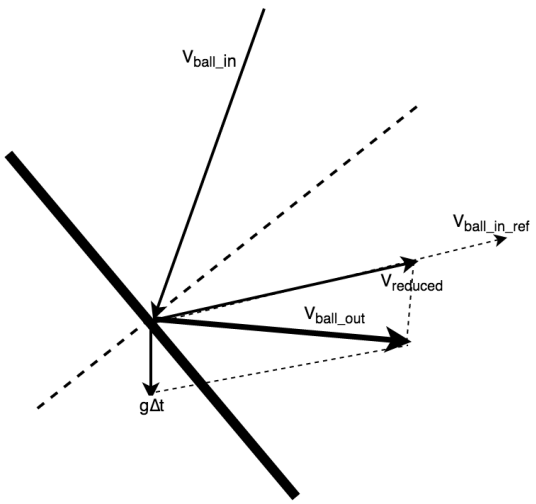

Figure 3: Forward impact kinetics

reduction on the speed.

$$\vec{v}_{red} = \vec{v}_{ref} - b\vec{v}_{ref}\Delta t \tag{27}$$

Adding gravity influence into (24) we get the final model of forward kinetic of impaction.

$$\vec{v}_{out} = \vec{v}_{ref} - b\vec{v}_{ref}\Delta t + g\Delta t \tag{28}$$

The inverse impact kinetics can be done in a easy way according to the equation above.

4 Summary

In this paper we mainly concentrate on two things, one is to fix reception error and model error by applying Kalman filter. The other is to control the racket effectively in order to hit the shuttlecock. Our badminton playing robot has not yet good adversary playing functionality, thus our next step will concentrate on effective adversary playing strategy.

References

S. Careelmont (2013). *Badminton shot classification in compressed video with baseline angled camera*, Kongens lygby, Denmark.

Y. Man, L. Zhao, J. Hu (2009).Proceedings of the 2009 IEEE International Conference on Robotics and Biomimetics. *Application of Kalman Filter in Track Prediction of Shuttlecock*, Guilin, China

J. J.Craig (2005). *Introduction to Robotics: Mechanics ,nd Control*, U.S.: Pearson/Prentice Hall

A. J.Cooke (1999).Shuttlecock Aerodynamics.*Sports Engineering* **2** ,85-96. Trumpinton, Cambridge.: Blackwell Science

K. Mlling, J. Kober, J. Peters (2011). *A Biomimetic Approach to Robot Table Tennis*, Tbingen, Germany