Journal of Zhejiang University-SCIENCE C (Computers & Electronics)
ISSN 1869-1951 (Print); ISSN 1869-196X (Online)
www.zju.edu.cn/jzus; www.springerlink.com
E-mail: jzus@zju.edu.cn

# A tracking and predicting scheme for ping pong robot*

Yuan-hui ZHANG†, Wei WEI†‡, Dan YU, Cong-wei ZHONG
(*School of Electrical Engineering, Zhejiang University, Hangzhou 310027, China*)
†E-mail: zhangyh23@gmail.com; wwei@cee.zju.edu.cn

Received Aug. 25, 2009; Revision accepted Dec. 16, 2010; Crosschecked Dec. 30, 2010

**Abstract:** We describe a new tracking and predicting scheme applied to a lab-made ping pong robot. The robot has a monocular vision system comprised of a camera and a light. We propose an optimized strategy to calibrate the light center using the least square method. An ellipse fitting method is used to precisely locate the center of ball and shadow on the captured image. After the triangulation of the ball position in the world coordinates, a tracking algorithm based on a Kalman filter outputs an accurate estimation of the flight states including the ball position and velocity. Furthermore, a neural network model is constructed and trained to predict the following flight path. Experimental results show that this scheme can achieve a good predicting precision and success rate of striking an incoming ball. The robot can achieve a success rate of about 80% to return a flight ball of 5 m/s to the opposite court.

**Key words:** Ping pong robot, Calibration, Trajectory tracking, Kalman filter, Neural network
**doi:**10.1631/jzus.C0910528 **Document code:** A **CLC number:** TP242.6

## 1 Introduction

In recent years, a novel ping pong robot has gradually become a hotspot in robotics research due to its challenge in mechanical design, real-time vision, and intelligent control (Andersson, 1989; Fassler *et al*., 1990; Naghdy *et al*., 1994; Miyamoto and Kawato, 1998; Acosta *et al*., 2003; Rusdorf *et al*., 2007). The vision system of the robot, which is "equivalent to the eyes of a human being", is a top priority task among all of the key techniques. From the first ping pong robot designed by Andersson (1987) to the latest humanoid ping pong robot made by the Vietnamese company TOSY Inc. (TOSY, 2008) and exhibited in IREX 2007 (International Robot Exhibition), nearly all of the cases require a strong and robust vision system capable of tracking and predicting the flying trajectory of a ping pong ball.

In this paper, a real-time tracking and predicting scheme is presented. The main task of such a scheme is subdivided into three parts. The first part is object recognition, which focuses on locating a ball in a 3D space by a monocular vision system similar to Kim *et al*. (1998). This part includes a calibration method used to estimate the light center and to extract objects on a captured image plane. The second part is to continuously track the ball's trajectory and observe its states including its position and velocity. The third part is to predict the future ball's trajectory and select a feasible striking point for the mechanical system to perform a striking motion.

In the following sections, we first introduce our experimental robot prototype, and then provide more details on the three parts and experiment results.

## 2 Ping pong robot prototype

Our system architecture (Fig. 1) is much similar to the one designed by Acosta *et al*. (2003), plus our design has some major enhancements.

For example, Acosta's prototype restricts the

‡ Corresponding author
* Project supported by the National High-Tech Research and Development Program (863) of China (No. 2008AA042602) and the Fundamental Research Funds for the Central Universities of China (No. KYJD09035)

flying area of the ping pong ball to a narrow desk (about 0.5 m×0.5 m); in contrast, in our prototype a standard ping pong table is used, which is larger in both mechanical and vision scope. Acosta's hitting strategy ignores an incoming ball's bounce stage on the robot's court (Acosta *et al*., 2003), while our system follows the rules of a standard ping pong game.

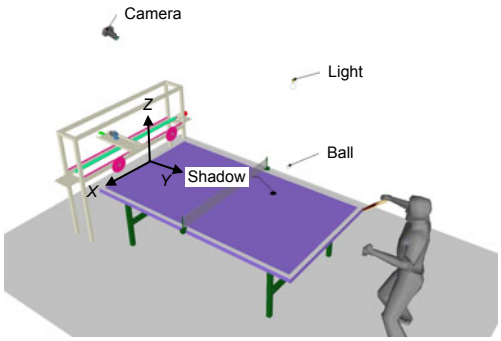


**Fig. 1 Ping pong robot prototype**
A world coordinates system is established on the edge of the table. The monocular vision system consists of a light (an electric lamp) and a digital camera fixed on the ceiling

The advantage of such a monocular vision system (Fig. 1) is that, by extracting the image coordinates of the ball and shadow, we can compute the ball position in world coordinates by means of triangulation (Reid and North, 1998; Acosta *et al*., 2003). Also, we can avoid the extra burden of building a binocular vision system with two cameras. The actual mechanical structure of our robot is shown in Fig. 2.

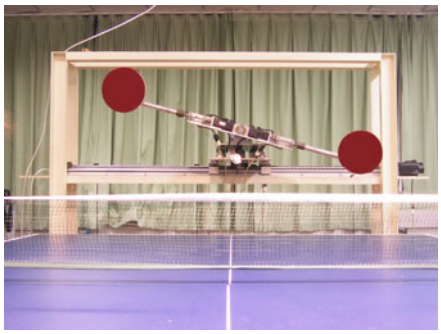

**Fig. 2 Robot mechanical structure**

## 3 Calibrating the light center

Suppose we have already performed the camera calibration process using Zhang's method (Zhang, 2000; Forsyth and Ponce, 2002), and a 3×4 camera matrix $\boldsymbol{M}$ is given. Since the position of the light center, from which we expect all the light to be emitted, is an important element of our triangulation, Acosta *et al.* (2003) did not give a method to calibrate the light center as its precision is believed to be sufficient. In our system, the scene space becomes much larger, and the light center is really hard to measure manually; thus, we propose a method to calibrate its position using a least square method in 3D spaces.

The geometry of our calibration method is demonstrated in Fig. 3.

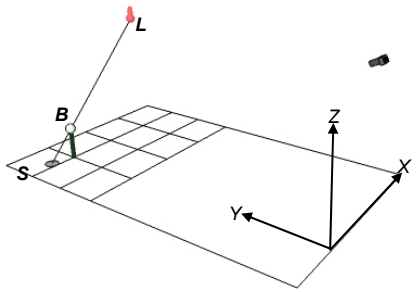


**Fig. 3 Calibration of the light center**
A stick with a ping pong ball fixed on its top end is standing perpendicular to the table plane, point $\boldsymbol{B}$ is the ping pong ball, $\boldsymbol{L}$ stands for the light center, and $\boldsymbol{S}$ denotes the shadow of the ball on the table

Three steps are used to calibrate the vision system.

Step 1: Suppose the length of the stick is known. We first draw a grid with perpendicular lines on the table (Fig. 3). As we have already established our world coordinates system on the table, the 3D position of every grid cross point is also known. Therefore, if we put the stick on each grid point, the ball position ($\boldsymbol{B}$ in Fig. 3) can also be calculated.

Step 2: Given a captured image from the camera, the 2D shadow region projected on the table can be extracted. Since the shadow ($\boldsymbol{S}$ in Fig. 3) is always lying on the $XY$ plane, its position in world coordinates can be evaluated through the calibrated camera matrix $\boldsymbol{M}$ by homography.

Step 3: The light center $\boldsymbol{L}$ lies on the ray $\boldsymbol{SB}$. If we change the stick position and perform Steps 1 and 2 repeatedly, we can obtain a series of rays constructed from different $\boldsymbol{S}_i$ and $\boldsymbol{B}_i$ (where $i$ denotes the position index of all the $n$ grid points).

Step 4: Theoretically, all the rays $\boldsymbol{S}_i\boldsymbol{B}_i$ ($i$=1, 2, …, $n$) intersect at the light center $\boldsymbol{L}$. But actually, due to the measurement noise and computation error, they

may not exactly intersect at the point $\boldsymbol{L}$. They just pass through the area adjacent to the point $\boldsymbol{L}$, so we can find only a best estimated $L_{\text{opt}}$ which minimizes the sum of the square distances to each line:

$$L_{\text{opt}} = \arg\min_{\boldsymbol{L}\in\mathbb{R}^3} \sum_{i=1}^{n} d_i^2, \tag{1}$$

where $d_i$ is defined as the distance from point $\boldsymbol{L}$ to line $\boldsymbol{S}_i\boldsymbol{B}_i$ (Press *et al*., 1992). $d_i$ is evaluated as

$$d_i = \frac{\left|(\boldsymbol{L}-\boldsymbol{B}_i)\times(\boldsymbol{L}-\boldsymbol{S}_i)\right|}{\left|\boldsymbol{B}_i-\boldsymbol{S}_i\right|}. \tag{2}$$

$L_{\text{opt}}$ can be evaluated using the least square method; it is regarded as the equivalent intersection points of all the rays. Thus, the best fitted light center is obtained.

Suppose the light center $\boldsymbol{L}$ has been estimated. Fig. 4 shows a simple triangulation of the ball's 3D position. In fact, similar to the calibration of the light center, this is still a least square problem.

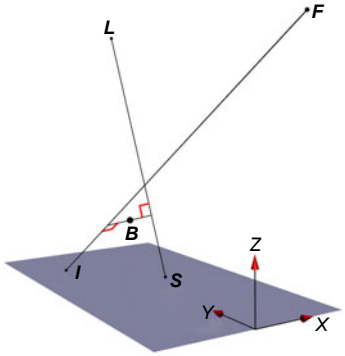


**Fig. 4 Triangulation of the ball position**

***F*** is the camera center, line ***FI*** can be established from the ball image, and line ***LS*** can be constructed from the shadow image. These two lines theoretically intersect at point ***B***. Ideally ***B*** is the ball position, but in practice, an estimated ***B*** is calculated as the midpoint of the distance between lines ***FI*** and ***LS***

## 4 Object recognition

In our system, a peripheral component interconnect (PCI) based acquisition module is used to receive the image data from a digital camera (A602fc, Basler, Germany) (Basler, 2006) through the firewire bus. Our vision system's capturing speed is 90 frames/s. Once the image data was captured and transferred to the memory buffer, some image processing algorithms were performed on that image. The camera outputs a Bayer format image, which is a raw image format. We first convert it to the HSV (hue, saturation, and value) image space (Gonzalez *et al*., 2003). To achieve better performance, we use the ROI (region of interest) to limit the search area on the image. This method rapidly reduces the computation time, since the computational complexity is determined mainly by the ROI size. For detecting the frame where the ball first appeared, a larger initial ROI is used. After detecting the ball in the first frame, the ROI size is reduced to 50×50 pixels for the next frame. The ROI is updated continuously with the previous detection result, so the object is always approximately fixed in the center of the ROI.

A background subtraction and threshold segmentation method was used to roughly extract the 'ball area' and the 'shadow area'. Tracking moving objects in a dynamic image sequence is far more complex than in a static image sequence. A common problem encountered is 'motion blurring', which is caused by scene variation in the camera sampling interval. To eliminate the motion blurring, we can manually reduce the electric shutter interval; the side effect is that we should increase the light power to compensate loss in scene illumination, which is still a workaround. Fig. 5a shows the motion blurring phenomenon of both the ball and shadow. Since the ball is believed to have a constant velocity in the exposure time, the contour of each region is considered to be fitted to an ellipse. Fig. 5b shows a fitted ellipse with its primary axes and minor axes crossed through the ball center. Also, the center of the ellipse is considered to be the target position corresponding to the mid-point of the exposure interval; thus, the time stamp associated with the image should be adjusted by subtracting a half exposure interval from the finishing time.

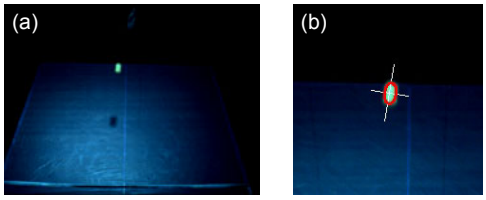


**Fig. 5 Ping pong ball image after motion blurring (a) and using the elliptical fitting method (b)**

Given the ball and shadow location in the image frame, the ball position in world coordinates can be estimated by a simple triangulation between the four elements including the light, camera, ball image, and shadow image as described in the previous section.

## 5 High-speed moving ball tracking

As the camera captures images successively, the ball positions ($X$, $Y$, and $Z$ components in world coordinates with an attached time stamp) are generated continuously. We must track the ball's trajectory precisely because the initial condition of the predicting procedure is greatly imposed on the ball's current states. Andersson (1989) used a curve fitting method to analyze trajectory, where a quadratic polynomial was fitted to ball motion on each axis; in contrast, Miyazaki *et al*. (2006) chose a least square method to estimate the current motion states of the ball. In this work, we choose a Kalman filter (Kalman and Bucy, 1961; Haykin, 2001) to estimate the motion on each axis. For example, consider the ball motion on the $X$ axis (Fig. 1). A discrete state transition equation and a discrete observation equation are established in the state space model (Ogata, 2001):

$$\boldsymbol{x}_k = \boldsymbol{F}_k \boldsymbol{x}_{k-1} + \boldsymbol{B}_k \boldsymbol{u}_{k-1} + \boldsymbol{w}_{k-1}, \tag{3}$$

$$\boldsymbol{z}_k = \boldsymbol{H}_k \boldsymbol{x}_k + \boldsymbol{v}_k. \tag{4}$$

In Eq. (3), $\boldsymbol{x}_k$ is a variable vector at time step $k$, $\boldsymbol{F}_k$ is the transition matrix describing state $\boldsymbol{x}_{k-1}$ to $\boldsymbol{x}_k$, $\boldsymbol{B}_k$ is the input matrix, $\boldsymbol{u}_{k-1}$ is defined as the control input, and $\boldsymbol{w}_{k-1}$ is a white noise term which has a Gaussian distribution with zero mean and covariance $\boldsymbol{Q}_{k-1}$. In Eq. (4), $\boldsymbol{z}_k$ is the observation state vector at time step $k$, $\boldsymbol{H}_k$ is a measurement matrix, and $\boldsymbol{v}_k$ is the normally distributed measurement noise with covariance $\boldsymbol{R}_k$. Here the state vector is defined as $\boldsymbol{x}_k = [x \ \dot{x}]^{\mathrm{T}}$, which is the ball position and velocity on the $X$ axis. Thus, the transition equation is rewritten as

$$\begin{bmatrix} \boldsymbol{x}_k \\ \boldsymbol{v}_k \end{bmatrix} = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_{k-1} \\ \boldsymbol{v}_{k-1} \end{bmatrix} + \begin{bmatrix} 0.5T^2 \\ T \end{bmatrix} \boldsymbol{a}_{k-1}, \tag{5}$$

where $T$ is the time interval between two successively captured images, but is not a constant. The matrices containing $T$ should be updated when a new image arrives. $\boldsymbol{a}_{k-1}$ is the noise term and $\boldsymbol{u}_{k-1}=\boldsymbol{0}$ on the $X$ axis (while on the $Z$ axis, $\boldsymbol{u}_{k-1}$ is the gravity acceleration). The Kalman filter output compared with a first-order derivative of the measurement position is illustrated in Fig. 6. The solid line (Kalman filter output) gives smoother and more accurate estimation of the velocity on the $X$ axis.

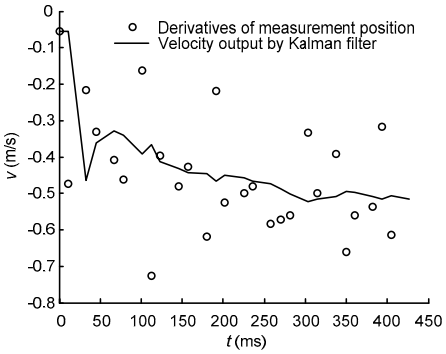


**Fig. 6 Kalman filter output of velocity on the $X$ axis**

## 6 Predicting the following trajectory

After the current ball states are estimated, the robot needs to predict the following path of the trajectory. The mechanical system needs this information to plan and perform a striking motion. In general, there are three stages of prediction: (1) the trajectory before the ball bounces on the table; (2) the collision stage at which the ball is bouncing on the table; (3) the trajectory after the bounce.

There are two kinds of prediction schemes existing in the previous ping pong robots. One was using an explicit physical model, and the other was using machine learning strategies which focus on the historic data and experiences but ignore the explicit physical formulas. For the former scheme, Acosta's prototype employs a physical model of a parabolic throwing formula (Acosta *et al*., 2003), and Andersson used a curve extrapolation model and a collision model (Andersson, 1989; Modi *et al*., 2005) to predict the future paths.

Due to the non-linear terms of these physical models (Resnick *et al*., 2002; White, 2002; Goodwill *et al*., 2004), the machine learning schemes have more advantages. Miyazaki's robot uses a locally-weighted regression (LWR) learning scheme to predict the target position (Matsushima *et al*., 2003; 2005). The LWR method is a memory-based model,

in which the training data should be stored in memory; this will increase the computational load. Our approach implements a multi-layer neural network (NN) where only the parameters of the NN can be used. Though it requires more training time, it runs much faster once the training is completed. The chosen NN has three layers including an input layer, a middle layer, and an output layer, which have 16, 22, and 6 cells respectively. A nonlinear input/output relationship is shown below:

$$[\boldsymbol{P}_1, t_1, \boldsymbol{P}_2, t_2, \boldsymbol{P}_3, t_3, \boldsymbol{P}_4, T]\rightarrow[\boldsymbol{P}_o, \boldsymbol{V}_o], \qquad (6)$$

where the input vectors describe the current states of the incoming ball, and the output vectors represent the estimated states after the bounce. $\boldsymbol{P}_i$ and $\boldsymbol{V}_i$ are the ball position vector and velocity vector, respectively, both containing three elements of $X$, $Y$, and $Z$. $\boldsymbol{P}_1$, $\boldsymbol{P}_2$, $\boldsymbol{P}_3$, and $\boldsymbol{P}_4$ are four successive ball positions in world coordinates given by the Kalman filter output. The time interval between $\boldsymbol{P}_i$ and $\boldsymbol{P}_{i+1}$ is defined as $t_i$ ($i$=1, 2, 3). Input scalar $T$ specifies the time interval between the predicted output location $\boldsymbol{P}_o$ and input point $\boldsymbol{P}_4$. These relationships are simply shown in Fig. 7.

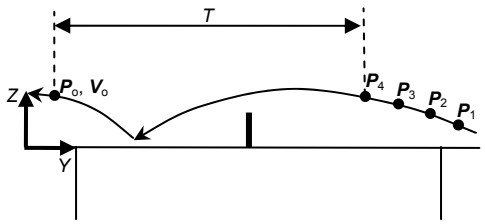


**Fig. 7 Input and output vectors of the neural network model**

The NN model is trained using the ball trajectories data sent from a ping pong pitching machine, so both the input and output vectors can be measured with the vision system. We used 100 trajectories as the training data, and each trajectory has 20 input vectors and 20 output vectors, so the total amount of the training data is 20×20×100=40 000. Finally we chose 36 trajectories to execute prediction trials.

Figs. 8a–8c show the prediction results of one trajectory using the NN model on $X$, $Y$, and $Z$ axes, respectively. Comparison of the predicted trajectory and the actual trajectory showed that the maximum predicted errors were 30, 40, and 30 mm on $X$, $Y$, and $Z$ axes, respectively.

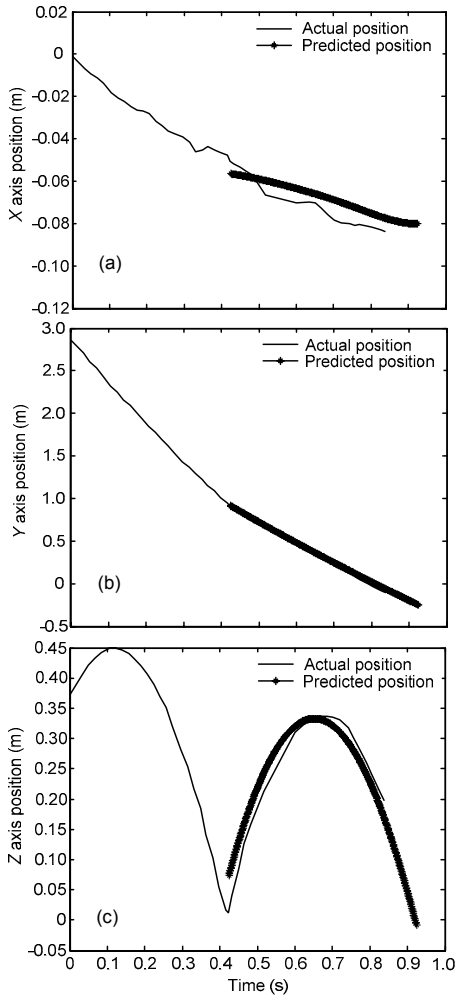


**Fig. 8 Predicted results on $X$ (a), $Y$ (b), and $Z$ (c) axes versus time using the neural network mode**

## 7 Conclusions

This paper presents a new tracking and predicting scheme applied in the vision system of a ping pong robot. A calibration method is developed to estimate the light center to enable a more precise triangulation. To eliminate the errors caused by 'motion blurring', an image processing procedure including an ellipse fitting is used to calculate the centroid of the target position in the image plane. To dynamically track the ball motion, the discrete Kalman filter is used to eliminate the measurement noise and give a smooth estimate result. Furthermore, for predicting the following path, we trained a neural network model to map the relationship from the current states to future states of the ball. Currently, the

robot can achieve a success rate of about 80% to return a flight ball of 5 m/s to the opposite court. Current progress and videos can be seen from the first author's websites (Zhang, 2009). We are still trying to improve the performance and intelligence of the robot. One important feature of the vision system is that all the methods are confirmed to be stable and fast. It is suitable for use in a real-time object tracking environment.

## References


Acosta, L., Rodrigo, J.J., Mendez, J.A., Marichal, G.N., Sigut, M., 2003. Ping-pong player prototype. *IEEE Robot. Autom. Mag.*, **10**(4):44-52. [doi:10.1109/MRA.2003.1256297]

Andersson, R.L., 1987. A Robot Ping-Pong Player: Experiment in Real-Time Intelligent Control. MIT Press, London, England.

Andersson, R.L., 1989. Dynamic sensing in a ping-pong playing robot. *IEEE Trans. Robot. Autom.*, **5**(6):728-739. [doi:10.1109/70.88095]

Basler, 2006. Camera Manual A600 Series Data Sheet. Available from http://www.graftek.com/pdf/Brochures/basler/A600fmanualNEW.pdf [Accessed on Dec. 1, 2007].

Fassler, H., Beyer, H., Wen, J., 1990. A robot ping-pong player: optimized mechanics, high performance 3D vision, and intelligent sensor control. *Robotersysteme*, **6**:161-170.

Forsyth, D.A., Ponce, J., 2002. Computer Vision: a Modern Approach. Prentice Hall, New Jersey, USA.

Gonzalez, R., Woods, R., Eddins, S., 2003. Digital Image Processing Using MATLAB. Prentice-Hall, New Jersey, USA.

Goodwill, S.R., Chin, S.B., Haake, S.J., 2004. Aerodynamics of spinning and non-spinning tennis balls. *J. Wind Eng. Ind. Aerodyn.*, **92**(11):935-958. [doi:10.1016/j.jweia.2004.05.004]

Haykin, S., 2001. Kalman Filtering and Neural Networks. Wiley, Chichester, UK.

Kalman, R.E., Bucy, R.S., 1961. New results in linear filtering and prediction theory. *Trans. ASME Ser. D: J. Basic Eng.*, **83**:95-107.

Kim, T., Seo, Y., Hong, K., 1998. Physics-Based 3D Position Analysis of a Soccer Ball from Monocular Image Sequences. 6th Int. Conf. on Computer Vision, p.721-726. [doi:10.1109/ICCV.1998.710797]

Matsushima, M., Hashimoto, T., Miyazaki, F., 2003. Learning to the Robot Table Tennis Task-Ball Control & Rally with a Human. IEEE Int. Conf. on Systems, Man and Cybernetics, p.2962-2969. [doi:10.1109/ICSMC.2003.1244342]

Matsushima, M., Hashimoto, T., Takeuchi, M., Miyazaki, F., 2005. A learning approach to robotic table tennis. *IEEE Trans. Robot.*, **21**(4):767-771. [doi:10.1109/TRO.2005.844689]

Miyamoto, H., Kawato, M., 1998. A tennis serve and upswing learning robot based on bi-directional theory. *Neur. Networks*, **11**(7-8):1331-1344. [doi:10.1016/S0893-6080(98)00062-8]

Miyazaki, F., Matsushima, M., Takeuchi, M., 2006. Learning to Dynamically Manipulate: a Table Tennis Robot Controls a Ball and Rallies with a Human Being. *In*: Advances in Robot Control. Springer Berlin Heidelberg, p.317-341. [doi:10.1007/978-3-540-37347-6_15]

Modi, K.P., Sahin, F., Saber, E., 2005. An Application of Human Robot Interaction: Development of a Ping-Pong Playing Robotic Arm. IEEE Int. Conf. on Systems, Man and Cybernetics, p.1831-1836. [doi:10.1109/ICSMC.2005.1571413]

Naghdy, F., Wyatt, J., Tran, S., 1994. A Transputer-Based Architecture for Control of a Robot Ping-Pong Player. *In*: Parallel Computing and Transputers. IOS Press, New York, p.311-317.

Ogata, K., 2001. Modern Control Engineering. Prentice Hall, New Jersey, USA, p.100-140.

Press, W., Teukolsky, S., Vetterling, W., Flannery, B., 1992. Numerical Recipes in C: the Art of Scientific Computing. Cambridge University Press, Cambridge.

Reid, I., North, A., 1998. 3D Trajectories from a Single Viewpoint Using Shadows. The British Machine Vision Conf., p.863-872.

Resnick, R., Halliday, D., Krane, K.S., 2002. Physics. John Wiley & Sons, Singapore.

Rusdorf, S., Brunnett, G., Lorenz, M., Winkler, T., 2007. Real-time interaction with a humanoid avatar in an immersive table tennis simulation. *IEEE Trans. Visual. Comput. Graph.*, **13**(1):15-25. [doi:10.1109/TVCG.2007.18]

TOSY, 2008. Citing Electronic Sources of Information. TOSY Robotics JSC. Available from http://www.tosy.com/ [Accessed on Dec. 12, 2009].

White, F.M., 2002. Fluid Mechanics. McGraw-Hill, New York, USA.

Zhang, Y.H., 2009. Citing Electronic Sources of Information. Personal Website of Ping-Pong Robot. Available from https://sites.google.com/site/pprobot/home [Accessed on Dec. 1, 2009].

Zhang, Z., 2000. A flexible new technique for camera calibration. *IEEE Trans. Pattern Anal. Mach. Intell.*, **22**(11):1330-1334. [doi:10.1109/34.888718]