Design and implementation of a three-dimensional simulation system for a humanoid table tennis robot

Lu Zhou

Abstract
This paper designs and implements a simulation system that can verify the planning algorithms such as table tennis trajectory planning and robot leg movement planning. The simulation system and the humanoid robot’s control system are connected based on the RTNet real-time Ethernet protocol. The table tennis game of the humanoid robot is implemented using the simulation of the forward and inverse kinematics of the robot arm, the hitting planning, and the forward and inverse kinematics of the robot leg.

Keywords
Table tennis robot, real-time framework, three-dimensional simulation system, forward kinematics, inverse kinematics

Date received: 10 October 2019; accepted: 9 December 2019

Introduction
In recent years, research in the field of robotics has become very popular and has attracted the attention of many researchers.1–4 They have successively developed robots for various fields. Humanoid robots have the characteristics of complex structures, high manufacturing costs, and high risks in walking and other operations. Humanoid robots can be imitated by human form or behaviors,5–8 have higher levels of movement and intelligence,9–12 and reflect more the level of mechanical manufacturing and automatic control, so are widely studied. With the advancement of research on humanoid robots, many scientists have carried out research on humanoid robot table tennis. Knight and Lowery13 developed a humanoid robot SARCOS which can play a number of rounds of table tennis with a humanoid shape and is an important progress of the study of humanoid table tennis robots. Sun et al.14 completed the research on the trajectory prediction of table tennis. Some others developed the arm of table tennis15,16 with 6 or 7 degrees of freedom. In Sun et al.,17 a humanoid robot is designed for table tennis game and the table tennis player. The strategy and trajectory planning of the game are studied.18,19 Therefore, it is necessary to utilize the simulation technology for studying their principles. As for the mechanical design of the humanoid robot, it is necessary to optimize the mechanism design of the humanoid robot in order to avoid losses caused by the designer’s mistakes. When it comes to the work of a humanoid robot such as walking, it is important to first simulate its work to avoid the danger of dumping or being interfered by the surrounding environment during the work. Table tennis is a high-speed, high-definition task for humans, even more for robots, so it is very necessary to use the simulation system. Simulation technology20 can be an efficient and safe tool for the research of humanoid robots, and it is widely used in the development of the humanoid robot field.

With the development of the computer simulation technology, the robot simulation technology has improved. Microsoft’s Robotics Developer Studio21 is an integrated environment for robot control and simulation research under the Windows operating system. It provides a large number of simulation models for industrial robots and mobile robots with friendly and realistic three-dimensional (3D) simulation scenarios. Users can control simulation machine models for robotics simulation through Microsoft’s virtual programming language. MathWork’s Simulink is tightly integrated with MATLAB and is an integrated environment for dynamic modeling, simulation, and analysis.
of systems. Camelot’s ROPSIM\textsuperscript{22} is an offline programming and dynamic simulation software widely used in the industry, and its offline programming system is especially praised by the industry. However, these simulation systems are too versatile, and it is difficult to meet the needs of functional customization such as humanoid table tennis robot, especially in the real-time performance.

During the process of project design, the simulation system can optimize the mechanism design of the humanoid robot. The ball trajectory collected by high-speed stereo vision is analyzed to establish the air flight model of the ball and the collision model between the ball and the racket.\textsuperscript{23} After the design process of the robot mechanism, a realistic 3D humanoid robot model is built in the simulation system for the prediction of the ball trajectory and the motion planning of the robot arm.\textsuperscript{24–26} The virtual table tennis experiment is carried out to verify the trajectory prediction algorithm of the ball, the trajectory planning algorithm of the humanoid robot arm and the leg, and so on. After the design and algorithm verification, the simulation system can be seamlessly connected to the control system of the humanoid robot to control the real humanoid robot for table tennis game.

The remainder of this paper is as follows. Section “Humanoid robot table tennis game system” presents the structure of the humanoid robot table tennis game system. Section “Forward and inverse kinematics” introduces the proposed forward and inverse kinematics. In section “Seamless connection between the simulation system and the real system”, seamless connection between the simulation system and the real system is proposed. Experimental results and analysis are presented in section “Experimental results”. Section “Conclusion” draws the conclusion of this paper.

**Humanoid robot table tennis game system**

The humanoid robot system for table tennis game is very complicated as is shown in Figure 1. There are more than 30 separately controllable joints in the whole body, which is named Huitong by the Beijing Institute of Technology. It has different sensors, including a six-dimensional force/torque sensor, a joint position sensor, a stereo vision, and so on.

It is necessary to maintain a dynamic balance for the humanoid robot’s walking and hitting. The system structure of the humanoid robot is shown in Figure 2. Since real-time control of the robot joints is necessary for the control system of the humanoid robot, the real-time Linux system based on the Xenomai real-time framework has been chosen. The control of the distributed joint controllers is implemented via RTnet-based real-time Ethernet. The feedback sensing data from the six-dimensional force/torque sensors and distributed joint sensors are transmitted to the host computer via the RTnet-based real-time Ethernet. The prediction result of the vision system is transmitted to the host computer through Wi-Fi, and the user can interact with the host computer through the mouse and the keyboard. In addition, the external control interface through the RTnet real-time Ethernet is preserved in the control system of the humanoid robot, which is the interface between our simulation system and the real control system.

**Forward and inverse kinematics**

**Forward kinematics**

There are two methods commonly used in the study of robotic forward kinematics. Denavit and Hartenberg proposed the D–H parameter method, which first establishes a coordinate system for each link of the robot through the homogeneous transformation between the coordinate systems, which finally obtains the representation of the coordinate system executed at the end in the base coordinate system.

Another method is based on the spin theory, which expresses the equation of motion as the product of the exponents of the spins. In this paper, we will use the D–H parameter method. The coordinate system established according to the above-mentioned D–H parameter method is shown in Figure 3.
According to the definition of the D–H connecting rod parameters, the D–H parameters of the humanoid robot arm are presented in Table 1.

The homogeneous transformation matrix of the coordinate system \( f_i \) relative to coordinate system \( f_{i-1} \) is a function of four link parameters

\[
T_{i-1}^i = \begin{bmatrix}
  c\theta_i & -s\theta_i & 0 & a_{i-1} \\
  s\theta_i c\alpha_{i-1} & c\theta_i c\alpha_{i-1} & -s\alpha_{i-1} & -s\alpha_{i-1} d_i \\
  s\theta_i s\alpha_{i-1} & c\theta_i s\alpha_{i-1} & c\alpha_{i-1} & c\alpha_{i-1} d_i \\
  0 & 0 & 0 & 1
\end{bmatrix}
\]

After obtaining the homogeneous transformation matrix of each link, the matrix is multiplied to obtain the final kinematic equation of the humanoid table tennis robot arm

\[
0T_7 = T_1^0 T_2^1 T_3^2 T_4^3 T_5^4 T_6^5 T_7^6
\]

### Inverse kinematics

The inverse joint kinematics calculates the joint data of the robot to be sent to the joint articulation in each joint motor. It is also one of the most critical parts of our simulation. A 7-degree-of-freedom humanoid arm has an infinite number of arm configurations corresponding to the position of the end effector in the workspace and without the limitation of the range of articulation. That is, there are infinite sets of inverse kinematic solutions.

In particular, when the third joint angle is 90°, the Z-axis of the coordinate system of the first joint is parallel to the Z-axis of the coordinate system of the fourth joint, and given the pose of an end effector, a group can be uniquely determined. The joint configuration corresponds to it. That is, the solution of inverse kinematics is unique. We define the plane consisting of the origin \( S \) of the first, second, and third joint \( f_1, f_2, f_3 \) coordinate systems, the origin \( E \) of the coordinate system of the fourth joint, and the origin \( W \) of the fifth, sixth, and seventh joint \( f_5, f_6, f_7 \) coordinate systems as the reference plane. For any joint configuration, we define \( u \) as the angle between the plane of \( SEW \) and the reference plane, as shown in Figure 4. In this way, we can solve the seven joint angles with the unknown number \( u \).

The humanoid robot arm end-effector coordinate system is known, and its homogeneous matrix in the base coordinate system \( \{7\} \) is

\[
0T_7 = \begin{bmatrix}
  nx & ox & ax & px \\
  ny & oy & ay & py \\
  nz & oz & az & pz \\
  0 & 0 & 0 & 1
\end{bmatrix}
\]

For the fourth joint angle, independent of \( \phi \), according to the cosine theorem

\[
\cos \theta_4 = \frac{|SE|^2 + |EW|^2 - |SW|^2}{2|SE||EW|} = \frac{L_1^2 + L_2^2 - (px^2 + py^2 + pz^2)}{2L_1L_2}
\]
where $L_1$ and $L_2$ are the lengths of the upper and lower arms. When the third joint angle is 90°, let $\theta^i R^0$ be the pose matrix of the third joint coordinate system $\{3\}$ in the base coordinate system, and the angles of the front three joints of the arm are $\theta_1, \theta_2, \theta_3$, respectively. When the angle between the arm and the reference plane is $\varphi$, the pose matrix of the third joint coordinate system $\{3\}$ in the base coordinate system is $\frac{1}{2} R^0$, and the angles of the first three joints are $\theta_1, \theta_2, \theta_3$, respectively. Then, we have the geometric relationship

$$
\begin{align*}
px &= -L_1 \cos \theta_1 s_2 \theta_2 - L_2 (\cos \theta_1 c_2 \theta_2 - \cos \theta_1 c_2 \theta_3 s_4) \\
py &= -L_1 s_2 \theta_1 s_2 \theta_2 - L_2 (\cos \theta_1 c_2 \theta_2 - \cos \theta_1 c_2 \theta_3 s_4) \\
\pz &= L_1 \cos \theta_2 + L_2 (\cos \theta_1 c_2 \theta_4 + s_2 c_2 \theta_4)
\end{align*}
\tag{3}
$$

Substituting $\theta_3 = 90°$ and equation (2) into equation (3), the following results can be derived

$$
\begin{align*}
\sin^0 \theta_1 &= \frac{-py}{\sqrt{px^2 + py^2}} \\
\cos^0 \theta_1 &= \frac{-px}{\sqrt{px^2 + py^2}} \\
\sin^0 \theta_2 &= \frac{-(\cos \theta_1 c_2 \theta_4 )}{\sqrt{px^2 + py^2}} \\
\cos^0 \theta_2 &= \frac{L_2 \sin \theta_1 + L_2 \cos \theta_1 s_4 \sin \theta_2}{\sin \theta_2 (px^2 + py^2 + pz^2)}
\end{align*}
\tag{4}
$$

By homogeneous transformation, equation (1) is available

$$
\begin{align*}
\frac{1}{2} R^0 &= \frac{1}{2} T^{01} T^{02} T^0 \\
\frac{1}{2} R^0 &= \frac{1}{2} T^{12} T^{23} T^0
\end{align*}
\tag{5}
$$

$\frac{1}{2} R^0$ can be obtained by rotating $\varphi$ around $\overline{SW}$

$$
\frac{1}{2} R^0 = R_{\overline{SW}} (\varphi)^0 R^0
\tag{6}
$$

Let the unit vector of the $\overline{SW}$ vector $\{px, py, pz\}$ be $\{k_x, k_y, k_z\}$, then $R_{\overline{SW}} (\varphi)$ is

$$
\begin{bmatrix}
k_x k_x \varphi + c \varphi & k_x k_y \varphi - k_z s \varphi & k_x k_z \varphi + k_y s \varphi \\
k_x k_y \varphi + k_z s \varphi & k_y k_y \varphi + c \varphi & k_y k_z \varphi - k_x s \varphi \\
k_x k_z \varphi - k_y s \varphi & k_y k_z \varphi + k_x s \varphi & k_z k_z \varphi + c \varphi
\end{bmatrix}
\tag{7}
$$

where $s \varphi = \sin \varphi$, $c \varphi = \cos \varphi$, and $v \varphi = 1 - \cos \varphi$, which can be solved by equations (4)–(8)

$$
\begin{align*}
\tan \varphi \theta_1 &= A_1 \sin \varphi + B_1 \cos \varphi + C_1 \\
\frac{D_1 \sin \varphi + E_1 \cos \varphi + F_1}{D_1^2 + E_1^2 + F_1^2} \\
\cos \varphi \theta_2 &= A_2 \sin \varphi + B_2 \cos \varphi + C_2 \\
\frac{D_2 \sin \varphi + E_2 \cos \varphi + F_2}{D_2^2 + E_2^2 + F_2^2} \\
\tan \varphi \theta_3 &= A_3 \sin \varphi + B_3 \cos \varphi + C_3 \\
\frac{D_3 \sin \varphi + E_3 \cos \varphi + F_3}{D_3^2 + E_3^2 + F_3^2}
\end{align*}
\tag{9}$$

where $A_1, \ldots, F_i (i = 1, 2, 3)$ are constant coefficients. In the same way, we obtain $\tan \varphi \theta_5$, $\cos \varphi \theta_6$, and $\tan \varphi \theta_7$ as

$$
\begin{align*}
\tan \varphi \theta_5 &= A_5 \sin \varphi + B_5 \cos \varphi + C_5 \\
\frac{D_5 \sin \varphi + E_5 \cos \varphi + F_5}{D_5^2 + E_5^2 + F_5^2} \\
\cos \varphi \theta_6 &= A_6 \sin \varphi + B_6 \cos \varphi + C_6 \\
\frac{D_6 \sin \varphi + E_6 \cos \varphi + F_6}{D_6^2 + E_6^2 + F_6^2} \\
\tan \varphi \theta_7 &= A_7 \sin \varphi + B_7 \cos \varphi + C_7 \\
\frac{D_7 \sin \varphi + E_7 \cos \varphi + F_7}{D_7^2 + E_7^2 + F_7^2}
\end{align*}
\tag{10}
$$

where $A_i, \ldots, F_i (i = 1, 2, 3)$ are constant coefficients too. Therefore, we get the expression of $\theta_1, \ldots, \theta_7$ with $\varphi$ as the parameter. For a given $\varphi$, from equations (2), (9)–(14), combined with the range of motion of the joint, a unique set of seven joint angles can be obtained.

### Seamless connection between the simulation system and the real system

The external interface of the distributed joint controller based on RTnet real-time Ethernet is preserved in the control system of humanoid table tennis robot. We can directly transmit the simulation control signal of the table tennis robot to the actual control system of the humanoid robot through this interface, making the simulation system capable of controlling the real humanoid robot for table tennis game. The system structure of the seamless connection between the simulation system and the real system of the humanoid robot is shown in Figure 5.

The simulation system also uses the Linux system based on Xenomai’s real-time framework. The trajectory prediction of the ball, the leg trajectory planning, and the arm trajectory planning are all implemented within this real-time system. The 3D graphics simulation runs on the non-real-time Linux operating system. The real-time system communicates with the non-real-time system to change the simulation scenario. The
real-time system of the simulation system communicates with the preserved interface in the humanoid robot control system through RTnet-based real-time Ethernet.

**Xenomai-based real-time system**

Xenomai is a real-time framework based on the Linux kernel, which can provide user-space-based programs with comprehensive interface-independent real-time and seamless integration into the GNU/Linux environment. The structure of the adopted Xenomai is shown in Figure 6.

The bottom layer of Xenomai is Adeos, which directly interacts with the hardware. Adeos can abstract the upper kernel software as domains. The real-time kernel is the Xenomai domain, and the non-real-time Linux part is the Linux domain. Adeos provides a unified abstract interface to the domain by hardware abstraction layer. The Xenomai domain provides different real-time interfaces to the user space with different real-time system specifications through different skins. The simulation system of the humanoid table tennis robot is based on the Xenomai real-time framework. The simulation system is divided into real-time and non-real-time parts. The real-time part is executed in the Xenomai domain, mainly for online table tennis trajectory prediction. If the humanoid robot enters the range within which it can hit, the trajectory planning modules of the arm and the leg can calculate the joint motion of the humanoid robot, respectively. Then, the signals are sent to the control system of the humanoid robot by real-time Ethernet cycle. The non-real-time part in the Linux domain is mainly responsible for the 3D scene rendering and implementing the table tennis simulation. The real-time Xenomai domain and the non-real-time Linux domain exchange data through shared memory and pipeline. The real-time and non-real-time processes are shown in Figure 7.

**Real-time communication based on RTnet**

RTnet is an open-source real-time Ethernet communication protocol stack in Xenomai, which utilizes standard Ethernet cards and provides driver support for the most common Ethernet cards. RTnet implements User Datagram Protocol (UDP)/Internet Protocol (IP), Transmission Control Protocol (TCP)/IP, Internet Control Message Protocol (ICMP), and Address Resolution Protocol (ARP) and provides POSIX standard Socket function interfaces for user space and kernel space.

To ensure the delay of network transmission, RTnet needs to use a dedicated Ethernet segment. The devices on the dedicated Ethernet segment control the access of the communication channel through RTmac. Both the humanoid robot control system and simulation system in this paper use the RTnet protocol stack and connect with each other directly through twisted pair. The connection structure is shown in Figure 8.

It can be seen from Figure 8 that RTnet implements real-time communication by replacing its traditional Ethernet Media Access Control (MAC) layer protocol based on collision detection and collision avoidance, and it has its own MAC layer protocol RTmac. It can...
provide interfaces for configuring and analyzing real-time communication through RTcfg and RTcap.

The simulation system and the humanoid robot control system can communicate according to the UDP protocol. The control system of the humanoid robot is the server and the simulation system is the client. The format definition of a UDP data packet is shown in Figure 9. The UDP data contain the UDP packet header part and the joint data. The joint data part contains the number of joints, the check information of the joint data, and the data of each joint. The data of each joint include joint number, joint angle, and so on.

The communication process of the simulation system and the humanoid robot is shown in Figure 10.

First, the humanoid robot waits and receives the data at the designated port. After getting the data sent by the simulation system, the legality of the data will be checked. If the data are correct, then the humanoid robot control system sends the data to the distributed driver and informs the simulation system that the data have been correctly received. If the received data are incorrect, the simulation system will be notified to resend it. Although a reliable transmission strategy for UDP transmission is added to ensure the reliability of real-time Ethernet transmission, it may affect the real-time performance of the network. However, in our experiments with hundreds of thousands of data transmission and reception, the packet loss or packet error never occurs, and the transmission delay of the 1000-byte packet is below 0.1 ms.

**Experimental results**

**Layout network**

First, in the 3D simulation system based on Coin3d, the game table of humanoid robot table tennis is constructed as is shown in Figure 11.

It can verify the forward and backward kinematics of the arm, the stroke planning, and the motion planning. Then, through the Linux-based Xenomai real-time system framework and RTnet real-time Ethernet communication protocol network architecture, the seamless connection between the simulation system and the humanoid robot control system is realized, and the simulation system drives the real humanoid robot to play the table tennis game.

All experiments have been recorded with the arm hitting the table tennis ball. In Figure 12, there are 20 sets of data of robot deviation in the $xyz$ position. If the robot fails to hit the ball, or the hitting effect is obviously incorrect, then there is no record.

**Data acquisition and analysis**

It can be found from the experimental data that the deviation between the predicted point and the observed hitting point under the control of the simulation system is consistent.
There is a clear consistency in the hitting time offset, that is, the hitting point is basically earlier than the predicted time. In the $X$ direction, the real velocity is basically larger than the predicted value (Table 2).

By analyzing 20 sets of data, the deviation between the hitting time and the predicted time is 10 ms on average, and the maximum is 30 ms. At present, the error between the simulation system and the actual control system is in the allowable range and is related to the calibration error of the vision system and the parameter error of the ball model.

**Conclusion**

In this paper, we have proposed the implementation of the seamless connection between the simulation system and the humanoid robot control system. It allows the simulation system to drive two real humanoid robots to play the multi-round table tennis game. First, this paper briefly introduces the real-time system adopted by the humanoid robot control system. Then, according to the demands of the humanoid robot control interface, Xenomai is chosen as the real-time part of the simulation system. RTnet is used for the simulation system and humanoid robot control. The system includes the interface of real-time Ethernet communication and introduces the definition of specific implementation and communication format. Finally, our simulation system can drive the real humanoid robot to play multi-round table tennis game, and the proposed method is verified by our experiments.

**Declaration of conflicting interests**

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

**Funding**

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: Research was funded by the National 863 Program of China (2012AA041403) and the National Natural Science Foundation of China (61305107).
References

1. Gülhan MM and Erbatur K. Kinematic arrangement optimization of a quadruped robot with genetic algorithms. Meas Control 2018; 51: 406–416.
2. Luka BL, Collieb AA, Cookec DS, et al. Walking and climbing service robots for safety inspection of nuclear reactor pressure vessels. Meas Control 2006; 39: 43–47.
3. Sun N, Liang D, Wu Y, et al. Adaptive control for pneumatic artificial muscle systems with parametric uncertainties and unidirectional input constraints. IEEE T Ind Inform. Epub ahead of print 19 June 2019. DOI: 10.1109/TII.2019.2923715.
4. Önkol M and Kasnakoğlu C. Adaptive model predictive control of a two-wheeled robot manipulator with varying mass. Meas Control 2018; 51: 38–56.
5. Andersson RL. *A Robot ping-pong player*. Cambridge, MA: The MIT Press, 1988.
6. Prat GA. Legged robots at MIT: what’s new since Rairbert. IEEE Robot Autom Mag 2000; 7: 159–151.
7. Dickinson M, Farley C, Full R, et al. How animals move: an integrative view. Science 2000; 288: 100–106.
8. Huang Q, Yokoi K and Kajita S. Planning walking patterns for a biped robot. IEEE T Robot Autom 2001; 17: 280–289.
9. Li Z, Huang Q, Li K, et al. Stability criterion and pattern planning for humanoid running. In: Proceedings of the IEEE international conference on robotics and automation, ICRA ’04, New Orleans, LA, 26 April–1 May 2004.
10. Kajita S, Kanehiro F and Kaneko K. A real time pattern generator for biped walking. In: Proceedings of the IEEE international conference on robotics and automation, Washington, DC, 11–15 May 2002.
11. Garcia E, Estremem J and Santos G. A classification of stability margins for walking robots. Robotica 2002; 20: 595–606.
12. Kajita S, Nagasaki T and Yokoi K. Running pattern generation for a humanoid robot. In: Proceedings of the IEEE international conference on robotics and automation, Washington, DC, 11–15 May 2002.
13. Knight J and Lowery D. Pingpong-playing robot controller by a microcomputer. Microprocess Microsys 1986; 10: 332–335.
14. Sun L, Liu J, Wang Y, et al. Ball’s flight trajectory prediction for table-tennis game by humanoid robot. In: Proceedings of the IEEE international conference on robotics and biomimetics (ROBIO), Guilin, China, 19–23 December 2009.
15. Yu Z, Liu Y, Huang Q, et al. Design of a humanoid ping-pong player robot with redundant joints. In: Proceedings of the IEEE international conference on robotics and biomimetics (ROBIO), Shenzhen, China, 12–14 December 2013.
16. Zhang Z, Xu D and Tan M. Visual measurement and prediction of ball trajectory for table tennis robot. IEEE T Instrum Meas 2010; 59: 3195–3205.
17. Sun Y, Xiong R, Zhu Q, et al. Balance motion generation for a humanoid robot playing table tennis. In: Proceedings of the 11th IEEE-RAS international conference on humanoid robots, Bled, 26–28 October 2011.
18. Biswas K, Kundu A and Kar I. Real-time energy-optimal moving target tracking by holonomic vehicle. In: Proceedings of the IEEE annual India conference, Bangalore, India, 16–18 December 2016.
19. Conceição AGS, Dórea CET, Martinez L, et al. Design and implementation of model-predictive control with friction compensation on an omnidirectional mobile robot. IEEE/ASME T Mech 2013; 19: 467–476.
20. Brogliato B. Some perspectives on the analysis and control of complementarity systems. IEEE T Autom Control 2003; 48: 918–935.
21. http://www.microsoft.com/robotics/#About
22. http://www.camelot.dk/
23. Wang Y, Sun L, Liu J, et al. A novel trajectory prediction approach for table-tennis robot based on nonlinear output feedback observer. In: Proceedings of the IEEE international conference on robotics and biomimetics, Tianjin, China, 14–18 December 2010.
24. Miah S, Shaik F and Chaoui H. Universal dynamic tracking control law for mobile robot trajectory tracking. In: Proceedings of the IEEE international conference on industrial technology (ICIT), Toronto, ON, Canada, 22–25 March 2017.
25. Wang Y, Sun L, Zhou L, et al. Online minimum-acceleration trajectory planning with the kinematic constraints. Acta Automatca Sinica 2014; 40: 1328–1338.
26. Lai C and Tsay TIJ. Self-learning for a humanoid robotic ping-pong player. Adv Robotics 2011; 25: 1183–1208.