Real-time motion tracking of cognitive Baxter robot based on differential inverse kinematics

Shaobo Li¹,², Xingxing Zhang¹*, Jing Yang¹,², Qiang Bai², Jianjun Hu³, Qisong Song² and Zhiang Li²

Abstract
The tracking motion of the robot is realized based on a specific robot or relying on an expensive movement acquisition system. It has the problems of complex control procedures, lack of real-time performance, and difficulty in achieving secondary development. We propose a robot real-time tracking control method based on the control principle of differential inverse kinematics, which fuses the position and joint angle information of the robot’s actuators to realize the real-time estimation of the user’s movement during the tracking process. The motion coordinates of each joint of the robot are calculated and the coordinate conversion between man and machine is realized with the combination of the Kinect sensor and the robot operating system. We have demonstrated the robustness and accuracy of the tracking method through the real-time tracking experiment of the Baxter robot. Our research has a wide range of application value, such as automatic target recognition, demonstration teaching, and so on. It provides an important reference for the research in the field of cognitive robots.

Keywords
Cognitive robot, real-time, tracking control, inverse kinematics, gesture recognition, kinetic sensor

Date received: 28 November 2020; accepted: 17 May 2021

Introduction
The real-time tracking motion control of the robot is an effective human–computer interaction method. It is an important breakthrough in the field of intelligent robot research.¹ Since seven degrees-of-freedom (7-DOF) robot has infinite underconstrained solutions, its motion trajectory generation has always been a difficult problem in robotics, and programming it to perform specific tasks is very tedious and difficult. Therefore, it is very important to find a general and efficient remote control method for robot.

At present, the research on the target tracking of mobile robots has achieved good results.² Some target tracking methods can not only effectively implement the user tracking process in a single-user environment but also implement user tracking in a multi-user interference environment.³⁻⁵ However, these studies are limited to the robot tracking the user’s spatial movement and cannot realize real-time tracking of the user’s body movements.

¹ State Key Laboratory of Public Big Data, Guizhou University, Guiyang, China
² School of Mechanical Engineering, Guizhou University, Guiyang, China
³ University of South Carolina, Department of Computer Science and Engineering, Columbia, SC, USA

Corresponding author:
Xingxing Zhang, State Key Laboratory of Public Big Data, Guizhou University, Guiyang 550025, China.
Email: XingxingZHANG_Star@163.com
The research of robot tracking user’s body movements mainly focuses on the development of remote operation framework and the improvement of control methods. Many related researchers use Kinect sensors to obtain the position of the user’s arm and provide the robot with coordinates for tracking movement.6–8 Compared with traditional methods, the method of using remote sensing technology9,10 to directly obtain the target position reduces a lot of hardware and external sensors. However, these methods are limited to the control of Schunk lightweight manipulators,6 Nao robots,7 and low-degree-of-freedom robots,8 and cannot be extended to the remote operation of most 7-DOF robots. Currently, the real-time tracking motion control method of robots is still very complicated, the control efficiency is low, and the lack of versatility makes it difficult to achieve secondary development. To solve the above problems, we propose a real-time tracking control method. First, we use the Kinect sensor to obtain the user’s motion coordinates and publish the coordinate information of each node of the robot through robot operating system (ROS). Then, based on the control principle of differential inverse kinematics, the motion coordinates of each joint of the robot are calculated to control the Baxter robot to track the motion in real time. Finally, we developed the software and hardware platform of the Baxter robot tracking motion and carried out the robot arm tracking experiment. Experimental results show that this method can well realize the human–machine interactive control of intelligent robots.

This article is organized as follows. The second section reviews the state-of-the-art of robot tracking motion method. The third section describes the real-time control method based on differential inverse kinematics. The fourth section introduces the hardware and software platform of Baxter robot tracking motion control experience. The fifth section contains real-time tracking experiments to validate our proposed method and visualization results that support our claim. At last, the sixth section summarizes our findings and contributions.

Related work

The real-time tracking motion control of robot allows interaction with the complex task in remote and inaccessible environments.11–13 In different application fields, there are different methods to study the tracking motion of robots. For example, Jevtic et al.4 used human bone tracking technology based on 3D vision, allowing users to guide the robot by walking in front of the robot or pointing to the desired position. Pang et al.5 proposed a robust visual tracking method based on deep learning human detector, Kalman filter, and recognition module. This method solves the problem that the robot cannot follow the specific user accurately in the environment of multiuser interference. A more effective and acceptable tracking behavior is realized.

In recent years, many human–computer interaction control methods have been proposed. The Kinect sensor can capture the user’s action to realize the robot’s tracking. In the literature,14,15 a robot with human–computer interface was established to reproduce the human–computer interaction based on sensor combination. Kruse et al.16 proposed a method of using the first version of Kinect to let dual-arm industrial robots track the user’s movement. Marinho et al.17 developed a remote operation framework and proposed an intuitive robot operation control system based on a double quaternion framework. The position of the end effector is controlled by the right hand, and some commands are executed by the left hand. Xi et al.18 used the task parameterized hidden semi Markov model to learn operation skills from several human demonstrations. A shared control method to help operators complete the operation tasks is proposed to reduce the workload and improve the efficiency. Syakir et al.6 make a control system for a 4-DOF manipulator using Kinect camera. They test the accuracy of the robot’s hand motion control. Li et al.7 and Avalos et al.9 developed an online tracking system based on Kinect sensors to control the arm and head of the Nao robot. The main goal of this work is to realize that the robot can track the human user’s motion tracking in real time. Using Kinect sensor to track the human skeleton is indeed a good method to improve the motion efficiency of remote control robot, which meets the application value of real time. However, when the follower disappears suddenly, the robot will be out of control.

In fact, the realization of robot tracking motion does not need to be limited to sensors that provide visual feedback.19 Special human joint sensors can also be used to implement it. For example, in the literature,20,21 motion information is obtained from gloves designed with acquisition sensors rather than visual sensors. Scibilia et al.22 and Aragón-Martínez et al.23 proposed a novel adaptive teleoperation framework. Using the feedback of the machine force sensor, the interaction links that must be compensated by robot dynamics are identified. The above two research methods can realize the robot arm tracking human arm motion, and both have real-time performance. In conclusion, the robot real-time tracking control can greatly reduce the difficulty of robot control programming. Therefore, the research on the real-time tracking motion of the robot has very important practical significance.

Research on real-time tracking control method based on differential inverse kinematics

In this section, we describe our full control method for the Baxter robot’s real-time tracking motion. Firstly, the user’s message is transformed into transform (TF) message. TF message is coordinate conversion, including the conversion of position information and posture information in each part of the user’s body. The data are multiplied by the
rotation matrix and adjusted according to the size of the robot.

The joint angle of robot is represented by joint configuration vector \( \bar{\theta} \) as follows

\[
\bar{\theta} = (\theta_1, \theta_2, \theta_3, \ldots, \theta_n)
\]

where \( n \) is the degree of freedom of the robot.

The symbol \( x_i \) is used to represent a task, the instruction of \( x_i \) is taken as \( x_{id} \), and the error of the task is defined as

\[
e_i = x_i - x_{id}
\]

Since the task is a spatial function, \( x_i \) is represented as

\[
x_i = f_i(\theta)
\]

To obtain the relationship between the derivative of the task error and the angular velocity of the joint, the time derivative of \( e_i \) is obtained

\[
\dot{e}_i = J_i \dot{\theta} - \dot{x}_{id}
\]

where \( J_i = \partial f_i / \partial \theta \) is the Jacobian real matrix of the robot arm joint.

To solve the joint angular velocity \( \dot{\theta} \), the task weighting method is adopted. Considering \( N \) tasks, the problem can be expressed as

\[
\min_{\theta} \left\{ \sum_{i=1}^{N} w_i \| \dot{e}_i - J_i \dot{\theta} + \dot{x}_{id} \| ^2 \right\}
\]

where the \( i \)’th task has an associated weight \( w_i \). This weight is used to set the relative importance for each task.

Since the improved differential evolution (X-DE) algorithm has strong global convergence and robustness, we use the improved X-DE algorithm to optimize the arm motion trajectory of the Baxter robot. The purpose is to reduce the tracking error and achieve more accurate tracking motion. The principle of the improved X-DE algorithm is given below.

The mutation factor \( F \) and the crossover factor \( CR \) have a key influence on the convergence effect of the algorithm and the selection of the next-generation population individuals. The constants \( F \) and \( CR \) make the algorithm easy to fall into the local optimum, reduce the convergence speed and search accuracy, and it is difficult to improve the trajectory optimization efficiency of the DE algorithm. Therefore, the mutation factor \( F \) and the cross factor \( CR \) are adaptively improved to enable the X-DE algorithm to achieve better optimization results. The adaptive formulas for \( F \) and \( CR \) are shown in equations (5) and (6), respectively

\[
F = (F_1 - F_2) \frac{G - g}{G} + F_2
\]

where \( F_1 \) and \( F_2 \) are the maximum and minimum mutation factors, \( G \) is the maximum number of iterations, and \( g \) is the current number of iterations.

\[
CR = CR_2 + \frac{CR_1 - CR_2}{G} g
\]

where \( CR_1 \) and \( CR_2 \) are the maximum and minimum crossover factors, \( G \) is the maximum number of iterations, and \( g \) is the current number of iterations.

After improving the original X-DE algorithm, in the initial stage of data search, the feature diversity of the data population of the search system can be guaranteed. In the later stage of data search, the differential gradient evolution search effect with fast and uniform convergence can be achieved.

The optimal trajectory is obtained by optimizing the discrete trajectory as follows

\[
\theta_{op,i,0} = [\theta_{op,i,0}, \theta_{op,i,1}, \ldots, \theta_{op,i,2n-1}, \theta_{op,i,2n}], (i = 1, 2, \ldots)
\]

To obtain the optimal trajectory of continuous type, we use cubic spline interpolation to carry out trajectory planning. The function of cubic spline interpolation method is to obtain continuous motion trajectory by interpolating the optimal discrete trajectory.

The boundary conditions of the \( i \)’th joint interpolation of the manipulator are as follows

\[
\begin{align*}
\theta_{op,i}(0) &= \theta_{op,i,0} = \theta_0, \\
\theta_{op,i}(T_E) &= \theta_{op,i,2n} = \theta_d \\
\dot{\theta}_{op,i}(0) &= \dot{\theta}_{op,i,0} = \dot{\theta}_{op,i} \\
\dot{\theta}_{op,i}(T_E) &= \dot{\theta}_{op,i,2n} = \dot{\theta}_{d}
\end{align*}
\]

The interpolation node is

\[
\theta_{op,i}(t_j) = \theta_{op,i,j}, j = \frac{j}{2n} T_E, \quad j = 1, 2, \ldots, 2n - 1
\]

The continuous tracking function \( \theta_{op,i}(k) \) obtained by interpolation is used as the optimal trajectory for the joint angle tracking of the \( i \)’th manipulator.

The relationship between joint velocity and joint position can be obtained using the first-order Taylor expansion method. The joint velocity can be transformed into joint position by Euler integral. The position of the joint is sent directly to the robot, and the robot can realize the corresponding motion.

The relevant node messages are read and published by ROS, and then, the position of the required joint angle is calculated. The positions of these joint angles are published as control elements in separate topics, which are read by the low-level controller of each arm of the Baxter robot. The target of the motion controller is defined as an operation space task based on Cartesian coordinates. Because the motion of the robot is three-dimensional, it can track the motion well.
Hardware and software design of real-time tracking motion experiment for cognitive robot

Some motion capture systems (such as depth sensors, tactile devices, etc.) can be used to obtain the relevant data of the robot’s tracking motion. We process Cartesian coordinates to get specific location coordinates, such as the user’s shoulder and elbow. The real-time tracking motion of the robot can be roughly divided into three steps: firstly, the user’s motion is obtained by Kinect; secondly, the TF coordinate of the user’s motion is transformed into the corresponding coordinate of robot; finally, the robot is controlled to complete the tracking motion based on differential inverse kinematics. The flowchart of the real-time tracking motion of Baxter robot is shown in Figure 1.

Figure 1 shows the process of robot tracking motion. The yellow box in the figure shows the debugging process before the real-time tracking experiment. First, check whether the communication between the computer and the robot is successful through the ping command, and then, execute the executable file that enables the robot. The green box is the realization process of the experiment. Kinect sensors are used to capture the user’s movements. It is an RGB-D camera that can generate color images, depth images, and 3D data from 20 joints of the human body at a rate of 30 frames/s. Then, the bone and joint data are sent to the computer via USB for processing. It is used to detect the surrounding environment and perform 3D modeling to guide the movement of the robot. The Kinect sensor device is shown at the top of the purple box in Figure 1. There are three lenses, the middle one is an RGB color camera, which is used to collect color images. The left and right lens is a 3D structured light depth sensor composed of an infrared transmitter and an infrared complementary metal oxide semiconductor camera to collect depth data. The color camera supports 1280 × 960 resolution imaging, while the infrared camera supports 640 × 480 resolution imaging.

As shown in Figure 2, the Baxter robot is a robot for tracking experiment. Kinect sensor is used as the head of the robot. Meanwhile, Figure 2 shows the eight position points to obtain the angle data of each joint. The Baxter robot has 7-DOF. We obtain the angle and position data of the joint to drive the robot to do real-time motion. Baxter provides a Software Development Kit (SDK) program running on remote computers, and the programs provide open-source ROS application programming interface (API) of ROS, Running ROS command and script through API to operate Baxter.

We use ROS as the intermediate system to connect users and robots. ROS is used under Linux system, but the default platform of Kinect sensor is Microsoft Windows. To solve this problem, we use Kinect and its default SDK under windows, and then, transfer the data to the computer running Linux. However, the official development package does not provide gesture recognition and tracking function, and does not realize the mutual alignment of RGB image/depth image but only provides the alignment of individual coordinate system. In the whole-body bone tracking, the SDK only calculates the position of the joint but not the rotation angle. Therefore, we use the unofficial combination development kit: SensorKinect + NiTE + OpenNI. SensorKinect is the driver of Kinect and NiTE is the middleware provided by PrimeSense in Israel, which can analyze the data read by Kinect, output human motions, and so on.

Figure 3 shows a frame corresponding to a user’s limb motion (called TF in ROS). From Kinect to Baxter robots, communication between all nodes is accomplished through special ROS topics. In addition to obtaining the TF
coordinates corresponding to the user’s limb actions, the point cloud data identifying the user’s motion can also be obtained through rviz, as shown in Figure 4.

Experiment and result analysis

Experimental setup

According to the research content of robot real-time tracking control method, two groups of experiments are set up. The first group: visual subjective analysis of robot real-time tracking; the second group: objective analysis of robot real-time tracking; the effectiveness of the proposed robot real-time tracking operation control method in this article has been verified. It should be noted that the second group of experimental data acquisition time is 600 ms, and the abscissa data are expressed as a 600-ms time series.

Subjective visualization analysis of real-time robot tracking

We use the experimental device described in the fourth section to test the effectiveness of the proposed method. Figure 5 shows the Baxter robot tracking the user to do different motions. As can be seen from Figure 5, the user controls the robot only by moving the arm without any additional sensors on the body.

When the user starts to move, the robot will not move immediately, and there will be a delay of about 1 s before it starts to track. The reason is that the robot has inertia, but the short delay will not cause the loss of motion information, and the impact is very small for the actual robot demonstration. There are scaling factors between the motion of the user and the robot. The reason is that the arm size of human and robot is different.

We combine the point cloud image with TF coordinates to show the relationship between the user’s motions and their corresponding TF coordinates. The results are shown in Figure 6.

As shown in Figure 6, TF coordinates almost coincide with user motions, which shows that TF coordinates obtained by ROS are very accurate. It makes the information received by the robot more accurate to achieve the corresponding motion accurately.

Analysis of objective results of robot real-time tracking experiment

In the experiment of the last section, the motion position data of each joint of the manipulator arm, as shown in Figure 2, are obtained. Due to the influence of the environment or light, the motion data may have a mutation. Therefore, we preprocess the experimental data. Firstly, the abnormal data are processed as missing values; then, it is deleted by deletion method. The reason for the direct use of the deletion method is that the proportion of missing values is small. Deleting these abrupt values can ensure the integrity of the data, which has little impact on the experimental results.

We validated our real-time robot tracking method in the above experiment. When the user raises his hands above his head (as shown in Figure 3), the trajectory trends of the left and right arms of the Baxter robot are almost the same, as shown in Figure 7.

Figure 7 shows the trajectory of Baxter robot’s elbow tracking the user’s movement. The duration of the trajectory is 10 s, and each tick mark in the figure represents 1 s. It can be seen from Figure 7 that when the user’s left and right arm are doing the same motion, the movement trend
of the left and right arm of Baxter robot is the same. It shows that the real-time control requirements of the human body to robot can be met.

Figure 8 shows the trajectory of shoulder 1 in the process of Baxter robot tracking. By comparing Figures 7 and 8, we find that elbow tracking is better than shoulder tracking, and the trend is smoother. There is a slight deviation in the tracking trajectory of the robot shoulder. The reason is that the user’s shoulder motion range is small.
and the data change is small, which is easy to cause position data to overlap. However, the overall motion trajectory is consistent, which does not affect the robot’s real-time follow-up motion.

As shown in Figures 9 and 10, when the user’s arm speeds up to do a certain motion (the time sequence is between 805 and 1345, that is, the time is 266–445 ms), the robot cannot track the motion at a similar speed due to the limitation of torque. But it tries to track users as much as possible without affecting its own institutions.

To show the tracking effect of the robot more intuitively, we compared the position data of the user and the robot, as shown in Figures 11 and 12. Firstly, Figure 11 shows a comparison between the position of the user’s right wrist and the Baxter robot’s right elbow 1. Secondly, Figure 12 shows a comparison between the position of the user’s left shoulder and the Baxter robot’s left shoulder 1.

As can be seen from Figures 11 and 12, there is an offset between the motion of the user and the robot. The reason for the offset is that the coordinate origin of the human and the robot is different. The origin of the coordinates of the Baxter robot is the center of the head, and the origin of the user’s coordinates is the center of the Kinect sensor.

Experimental result shows that the tracking motion of the robot has a delay of 0–3 ms. It is worth noting that in robotics, nonreal-time motion usually refers to offline processing within minutes or even hours after acquiring user motion, and then, allowing the robot to reproduce the motion. Since the delay of 0–3 ms is very small in practical applications, we can consider our experimental process to be real time. In addition, the main purpose of the research is to make the robot have perception ability. Such a small delay will not affect the robot’s perception of the user’s actions.
Conclusion

We propose a remote control method for the robot to follow the user’s movement, and we verify the effectiveness of the method through the tracking experiment of the Baxter robot. We found that when the user performs symmetrical arm actions, the trajectories of the left and right arms of the Baxter robot are almost the same in the same time period. When the user makes random actions, the trend changes of the Baxter robot’s arm motion trajectory and the user’s motion trajectory are almost the same in the same time period. The visualization experiment results show that the proposed method can well realize the real-time tracking movement of the robot.

In addition, this method has been widely used. For example, it can not only realize the rapid response of robots in emergency situations but also can be used for robot teaching. The user only needs to teach the robot a few times or even once to control the robot to repeat the corresponding actions. This provides an important reference for future research in the direction of robot teaching.

To give full play to the characteristics of the Baxter robot, future work will involve more body movements to further improve the robot’s flexibility and accuracy. In addition, we will consider adding functions to store user’s complex actions. These functions can be called directly or controlled by voice when necessary. At the same time, we will consider how to apply this method to industrial robots and service robots in the future.

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: This work was supported in part by the National Science and Technology Innovation 2030 Major Program of China under grant no. 91746116, and in part by the National Natural Science Foundation of China under grant no. 2018AAA0101803, in part by the National Natural Science Foundation of China under grant no. 91746116, and in part by the Science and Technology Major Special Project of Guizhou Province under grant no. [2017]3001.

ORCID iD

Xingxing Zhang https://orcid.org/0000-0001-6368-5762

References

1. Lin C, Chen PC, Pan YC, et al. The manipulation of real-time Kinect-based robotic arm using double-hand gestures. J Sensors 2020; 2020: 1–9.
2. Bai Q, Li S, Yang J, et al., Object detection recognition and robot grasping based on machine learning: a survey. IEEE Access 2020; 8: 181855–181879.
3. Wang M, Liu Y, Su D, et al. Accurate and real-time 3-D tracking for the following robots by fusing vision and ultrasonic information. IEEE/ASME Trans Mechatron 2018; 23(3): 997–1006.
4. Jevtic A, Doisy G, Parmet Y, et al. Comparison of interaction modalities for mobile indoor robot guidance: direct physical interaction, person following, and pointing control. IEEE Trans Hum-Mach Syst 2015; 45(6): 653–663.
5. Pang L, Cao Z, Yu J, et al. A robust visual person-following approach for mobile robots in disturbing environments. IEEE Syst J 2020; 14(2): 2965–2968.
6. Syakir M, Ningram ES, and Adjji Sulistijono I. Teleoperation robot arm using depth sensor. In: International electronics symposium (IES). Surabaya, Indonesia, 27–28 September 2019, pp. 394–399. IEEE.
7. Li C, Yang C, Liang P, et al. Development of Kinect based teleoperation of Nao robot. In: International conference on advanced robotics and mechatronics (iCARM). Macau, China, 18–20 August 2016. DOI: 10.1109/iCARM.2016.7606908.
8. Avalos J, Cortez S, Vasquez K, et al. Telepresence using the Kinect sensor and the NAO robot. In: IEEE latin american symposium on circuits & systems. Florianopolis, Brazil, 28 February–2 March 2016.
9. Elias Miranda-Vega J, Flores-Fuentes W, Sergiyenko O, et al. Optical cyber-physical system embedded on an FPGA for 3D measurement in structural health monitoring tasks. Microprocess Microsyst 2018; 56: 121–133.
10. Flores-Fuentes W, Elias Miranda-Vega J, Rivas-Lopez M, et al. Comparison between different types of sensors used in the real operational environment based on optical scanning system. Sensors 2018; 18(6): 1864–1879.
11. Xiao X, Cizmeci B, Schuwerk C, et al. Model-mediated teleoperation: toward stable and transparent teleoperation systems. IEEE Access 2017; 4: 425–449.
12. Xiao X and Steinbach E. Towards real-time modeling and haptic rendering of deformable objects for point cloud-based model-mediated teleoperation. In: IEEE international conference on multimedia and expo workshops (ICMEW). Chengdu, China, 14–18 July 2014. DOI: 10.1109/ICMEW.2014.6890578.
13. Li Z, Xia Y, Wang D, et al. Neural network-based control of networked trilateral teleoperation with geometrically unknown constraints. IEEE Trans Cybern 2016; 46(5): 1051–1064.
14. Xiao Y, Zhang Z, Beck A, et al. Human-robot interaction by understanding upper body gestures. Presence-Teleop Virtual Environ 2014; 23(2): 133–154.
15. Lin TC, Krishnan AU, and Li Z. Physical fatigue analysis of assistive robot teleoperation via whole-body motion mapping. In: International conference on intelligent robots and systems (IROS). Macau, China, 3–8 November 2019. DOI: 10.1109/iCARM.2016.7606908.
16. Kruse D, Wen JT, and Radke RJ. A sensor-based dual-arm tele-robotic system. IEEE Trans Autom Sci Eng 2015; 12(1): 4–18.
17. Marinho MM, Geraldes AA, Bó AP, et al. Manipulator control based on the dual quaternion framework for intuitive
teleoperation using Kinect. In: Brazilian robotics symposium & Latin American robotics symposium. Fortaleza, Brazil, 16–19 October 2012. DOI: 10.1109/SBR-LARS.2012.59.

18. Xi B, Wang S, Ye X, et al. A robotic shared control teleoperation method based on learning from demonstrations. Int J Adv Robot Syst 2019; 16(4): 172988141985742.

19. Zhang F, Cully A, and Demiris Y. Probabilistic real-time user posture tracking for personalized robot-assisted dressing. IEEE Trans Robot 2019; 35(4): 873–888.

20. Fang B, Sun F, Liu H, et al. A novel data glove using inertial and magnetic sensors for motion capture and robotic arm-hand teleoperation. Ind Robot 2017; 44(2): 155–165.

21. Fang B, Guo D, Sun F, et al. A robotic hand-arm teleoperation system using human arm/hand with a novel data glove. In: IEEE international conference on robotics & biomimetics. Zhuhai, China, 6–9 December 2015. DOI: 10.1109/ROBIO.2015.7419712.

22. Scibilia A, Laghi M, De Momi E, et al. A self-adaptive robot control framework for improved tracking and interaction performances in low-stiffness teleoperation. In: 18th international conference on humanoid robots (humanoids). Beijing, China, 6–9 November 2018. DOI: 10.1109/HUMANOIDS.2018.8625062.

23. Aragón-Martínez A, Arias-Montiel M, Lugo-González E, et al. Two-finger exoskeleton with force feedback for a mobile robot teleoperation. Int J Adv Robot Syst 2020; 16(6): 1–18.

24. Zhang J, Wang S, Zhou P, et al. Novel prescribed performance-tangent barrier Lyapunov function for neural adaptive control of the chaotic PMSM system by backstepping. Int J Electr Power Energy Syst 2020; 121: 105991.

25. Junxing Z, Shilong W, Shaobo L, et al. Adaptive neural dynamic surface control for the chaotic PMSM system with external disturbances and constrained output. Recent Adv Electr Electron Eng 2020; 13: 894–905.

26. Avalos J and Ramos OE. Flexible visually-driven object classification using the Baxter robot. In: Electrical engineering and computing (INTERCON), Cusco, Peru, 15–18 August 2017. DOI: 10.1109/INTERCON.2017.8079677.

27. Yang J, Li S, Wang Z, et al. Real-time tiny part defect detection system in manufacturing using deep learning. IEEE Access 2019; 7: 89278–89291.

28. Yang J, Li S, Gao Z, et al. Real-time recognition method for 0.8 cm darning needles and KR22 bearings based on convolution neural networks and data increase. Appl Sci 2018; 8(10): 1857–1867.

29. Avalos J and Ramos OE. Real-time teleoperation with the Baxter robot and the Kinect sensor. In: IEEE 3rd colombian conference on automatic control (CCAC). Cartagena, Colombia, 18–20 October 2017. DOI: 10.1109/CCAC.2017.8320346.