International Journal Of Engineering Research And Development
e- ISSN: 2278-067X, p-ISSN: 2278-800X, www.ijerd.com
Volume 16, Issue 11 (November 2020), PP. 25-34

A Hierarchical Real-Time ControlScheme for a Life-SizeHumanoid Robot Teleoperation with Complexity-Dependent Task Decomposition

Xudong Li, Yukai Luo, Liang Gong, Pengcheng Xia,Yixiang Huang
School of Mechanical Engineering, Shanghai Jiao Tong University, Shanghai 200240, China
Corresponding Author:Liang Gong

ABSTRACT: *To improve the defects of the life-size humanoid robot, such as slow system response, operation difficulty and the lack of autonomy, a framework of distributed multi-core communication and a hierarchical control strategy are presented. First, the distributed multi-core communication framework divides the robot control system into five levels according to processing capacity. Second, based on the hierarchical control strategy, full-autonomous motions and semi-autonomous motions are decomposed and processed at different levels. Therefore, dataflow in the communication network is well planned to reduce latency and the robot can accomplish more tasks autonomously to reduceoperation difficulty. And a new human-machine interface and control scheme with complexity-dependent task decomposition are presented which enable the operator to acquire the robot's visual information and teleoperate the robot to achieve real-time visual angle transformation, real-time hand joint mapping, and complex continuous action. The hierarchical robot control scheme is proved to be effective and competent for dealing with complex action and multi-terminal interaction.*

KEYWORDS: *Hierarchical control, life-size humanoid robot,real-time teleoperation, motion planning*

Date of Submission: 18-11-2020 Date of Acceptance: 03-12-2020

I. INTRODUCTION

Teleoperation systems have been developed for a human operator to perform complex tasks in remote environments. Remote teleoperation of a robot manipulator by a human operator is often necessary for unstructured dynamic environments when human’s presence at the robot site is undesirable [1]. Humanoid robots have human-like designs and can mimic human motions [2]. The fascinating aspect of the humanoid is the possibility to interact with it: to teach, to demonstrate, even to communicate [3].

The hierarchical strategy was originally derived from the market allocation mechanism and then applied to the task allocation strategy in multi-robot collaboration such as multiple AGVs, snake robots, and target tracking. And the hierarchical strategy can be used to reduce the total system’s complexity and to simplify control [4-6]. This paper builds a distributed multi-core communication and proposes a real-time control scheme for humanoid teleoperation robots based on it to optimize the control by task decomposition. The humanoid teleoperation robot InMoov used in this paper is a life-size robot with 25 DOFs (degree of freedoms) and can be teleoperated to achieve functions including visual angle transformation, movement and complex action by using multiple HMI (human-machine interface) terminals.

The rest of the paper is organized as follows. Section II discusses the problem of the teleoperation system. Section III provides a communication framework and control strategies to solve the problem. Section IV explains the whole teleoperation robot system’s setup process. Section V tests the actual effect of the system by functional experiments. Finally, Section VI deals with the conclusions about the control scheme.

II. PROBLEM DESCRIPTION

The humanoid robot teleoperation system is a typical complex teleoperation system, which has a large number of irregular degrees of freedom based on human structure. Teleoperating such a complex system often have the problem of slow system response, poor operation, and lack of system independence.

The first problem is system teleoperation delay, which is the time interval the system requires to perform effective action after the operator gives the remote operation instructions. In some remote teleoperation systems, there may be a significant time delay between the specification of a command and its execution [2]. And it is a serious problem from the perspective of stability and performance of control systems [3].

The total system time delay depends on the HMI terminal’s polling period, wireless transmission delay, motion planning delay and wired communication delay.

Since the HMI terminal’s polling period is determined after the system was built and wired communication delay is relatively small, there are two aspects to reduce teleoperation delay. One is the delay generated by the data transmitted wirelessly. When the operator teleoperates the robot to perform the action, a large amount of control data is transmitted wirelessly. The other is the length of time used when the operating system performs complex motion planning, limited by the calculation delay of the computer's computational speed. Reducing the number of wireless intermediate levels from the HMI terminal to the local system of the robot, decreasing the amount of data in the wireless transmission and lowering IPC calculation pressure by reusing planning data is effective for cutting down the time delay.

The second problem is that the complex robot system has a lot of DOFs, which are set irregularly. It is hard for the operator to fully control each DOF of the robot and understand the movement of the robot. An HMI which is easy to handle and an intuitive 3D simulation display environment are necessary. Through the simulation environment to plan complex actions and real-time mapping to achieve simple action can improve system operability.

The third problem is the lack of system independence. The operation of complex teleoperation robot systems often relies on real-time control of remote computing terminals and human auxiliary control. Remote operating systems that include teleoperation terminals are less expandable and can be used to support different types of control terminals. The achievement of the complete control in the robot's local system and the embedded ROS system with motion planning can improve the robot system’s independence. By regulating the communication protocol between the various levels can make the system support multi-terminal control.

To solve the problem, an effective and stable communication network that contains a variety of communication protocols such as ModBus, UDP, Serial and JSMpeg and multiple forms including wired transmission and wireless transmission is necessary. And it is significant to achieve drive functions of a plurality of arithmetic processing systems and build the communication framework to achieve the link among HMI, multi-terminals and robot local embedded system. And developing a specification set of instructions which enable operators to teleoperate the system to achieve tasks with different complexity by using simple operations is also important.

III. HIERARCHICAL CONTROL STRATEGY

In order to improve the robot’s response speed, autonomy, and reduce operation difficulty, the data flow needs to be optimized to reduce wireless transmission parts which produce a large delay in communication and tasks of system planning parts. Reducing the robot’s dependence on the remote operator and processing terminal and simplifying the control instructions are effective. And a hierarchical master-slave communication structure is highly required to fulfill the demand of expandability and efficiency[9].

3.1 Hierarchical framework

The whole robot system is divided into five levels from top to bottom: human, teleoperation terminal, industrial PC, InMoov core controller and drive executor. Human is responsible for making the decision and sending control instructions based on the environmental information. The teleoperation terminal level identifies the control instructions given by the person and sends it to the IPC level via the UDP protocol. IPC level depends on the ROS system and can be used as a routing node to process and transfer information. It also has the ability of motion planning. For a destination, IPC can create a plan for the movement of the robot over a period of time, and send down the timestamp and position information of each joint through ROS Serial. As the Modbus RTU protocol host, the InMoov core controller sorts, packages and sends the data to the corresponding drive executor nodes. Also, the InMoov core controller extends an SD card to store data. The InMoov core controller stores the motion data in the txt file format to SD card and names it. The drive executor level includes servo drive Nano nodes and chassis Mega control board which directly modify the driver steering to the position of the target and drive the Mecanum wheel in a specific movement speed according to the data sent from their superiors. At the same time, the data sent by the superior can contain the time, and the driving executor will accomplish the driving task at a specific time according to the time information to realize the planning action. The hierarchical framework is shown in Figure 1.

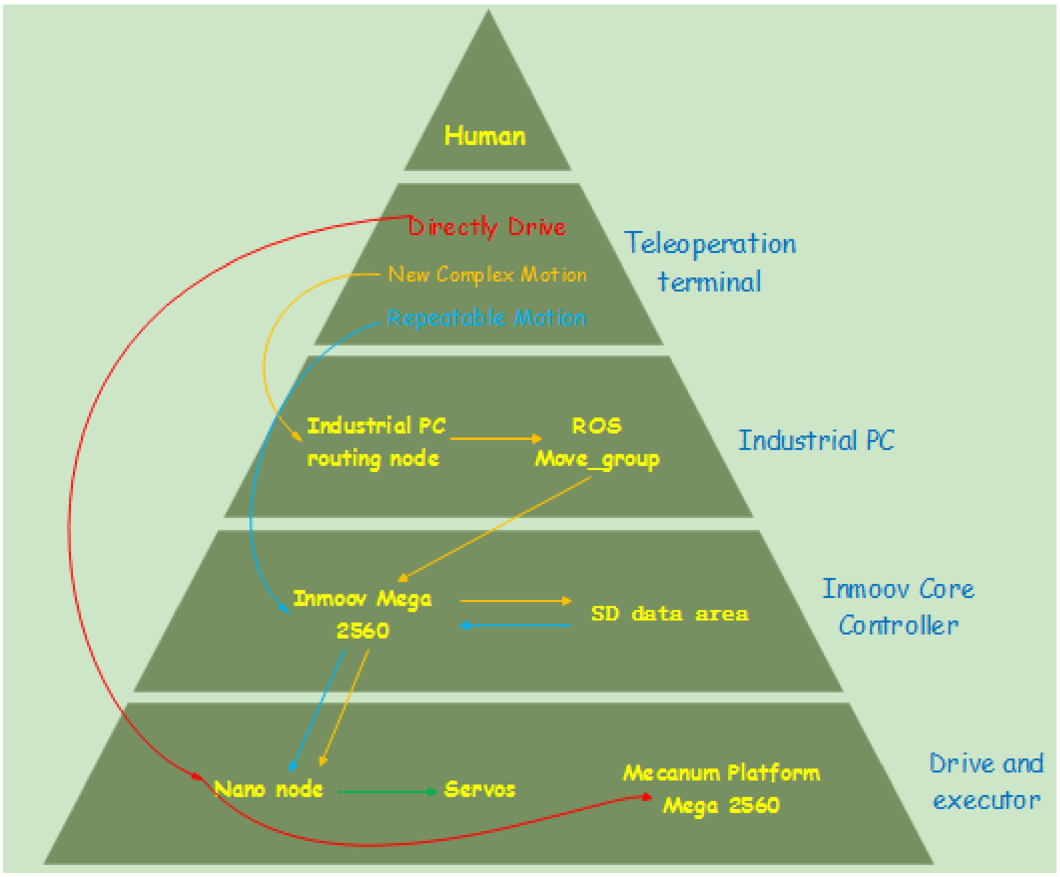


Fig 1. Hierarchical framework

All of the above levels, including the human, have the ability to process information. According to the environmental information of the teleoperation terminal video, the human makesthe decision and gives control instructions. The control instructionsare divided into three kinds of instructions which are depended on the task complexity. The first is Direct Drive which responses fast and drives directly one or more joints to a position. The second is New Complex Motion which is semi-autonomous and drives complex continuous movement for the first time in response to a particular situation under the planning of the PC operating system.The third is Repeatable Motion which is a continuous movement that has occurred or isused repeatedly. And the Repeatable Motion is a full-autonomous response action. The operator only needs to give a simple operation instruction, and the robot will perform a complete complex action autonomously. For three different instruction types, the system has different strategies.

3.2Directly Drive Instruction

The main purpose of this type of instruction is to achieve direct drive for a specified actuator in the robot system, such as head angle change and chassis movement. This type of instruction is simple and carries less information. The teleoperation terminal generates and sends a directly drive format data down when receivding such control instructions. The IPC and the InMoov core controller will not process the data and pass data to the target driver only as an adapter. The target driver executes the response immediately based on the data.

3.3New Complex Motion

This type of instruction enables humanoid special complex actions such as driving the arm to grab a particular object or switching from one pose to another. This kind of motion is usually complex and involves many degrees of freedom, so it is necessary to make motion planning with the help of ROS in IPC. The teleoperation terminal sets the target location information and transmits the information to the industrial computer terminal. The industrial computer makes the inverse solution to the target position and obtains the target pose, and do motion planningcombined with the current pose and the target pose. The industrial computer obtains a series of angle data of each node with a timestamp and sends it to the InMoov core controller. The InMoov core controller sorts data packets in a specified format and sends them to each drive executor,which will store the data to an SD card at the same time. The driver drives the corresponding actuator according to the time plan, enabling the robot to perform complex continuous actions.

3.4 Repeatable Motion

This type of action is initially also obtained by motion planning, but unlike the new complex actions, the type of action robot may be used many times such as ritual actions. After the type of action is pre-programmed, the corresponding node location data and time data are stored in the SD card. When the remote terminal receives such instructions, the IPC will no longer do motion planning but only transmit the instructions which require the core controller to read corresponding action number data from the SD card. This part of the data is the same as the planned motion data, and the after process is the same as in New Complex Motion.

The WiFi communication between the teleoperation terminal and the IPC only transmits the control instruction data. Between the IPC and the InMoov core controller, the action data is transported in large amounts only when the ROS is required for motion planning to accomplish a new complex motion. A large amount of data is often transmitted between the InMoov core control board and the drive executor.

The hierarchical control strategy allows people to simply give a control command, and the robot system autonomously processes the task by stages to realize complex robot functions. At the same time, there is always a smaller amount of control instructions in a long data transmission route and the transmission of large amounts of data occurs in the wired communication system inside the robot which ensures the stability of the communication system. Directly driving instructions and repeatable motion instructions avoid long time motion planning, which can significantly improve the system response speed. Robots have better autonomy to complete tasks to support multi-terminal and reduce difficulty in operation.

IV. SETUP OF THE TELEOPERATION ROBOT SYSTEM

The entire teleoperation robot system consists of a Mecanum Wheel Mobile Platform carrying an industrial computer, a multi-degree-of-freedom life-size humanoid robot InMoov, distributed Nano control cores, a teleoperation terminal HMI, and a complete communication framework based on the system structure. Figure 2 shows the structure of the robot.

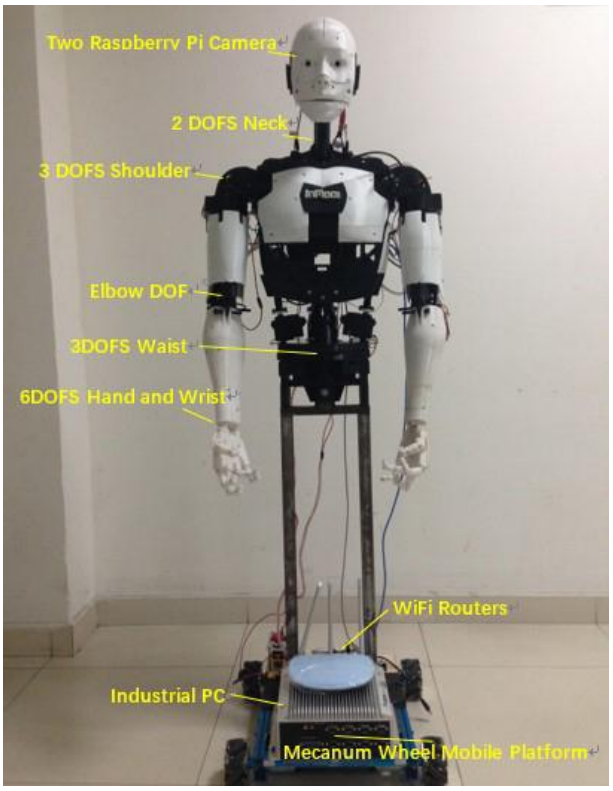


Fig 2. Structure of the robot

4.1 Mecanum wheel Mobile platform

The Mecanum wheel was invented by the Swedish engineer Bengt Ilon in 1973. Each of the Mecanum is driven by a separate motor which gives the vehicle the three degrees of freedom necessary for an omnidirectional movement on the level ground [10]. In order to achieve high motion flexibility of the robot system, we set up a mobile platform based on the Mecanum wheel to carry the whole robot.

The mobile platform contains a Mega2560 core control board, which is used to communicate with the industrial computerand carries out the PID control for each wheel according to the instruction [11]. It can drive the platform forward, backward, rotation and translational motion. The wiring diagram of the platform is shown in Figure 2. At the same time, the platform is also equipped with independent industrial computers and routers for the realization of communication with teleoperation terminal such as Android APP terminal. It also contains lithium batteries and power components to power the entire system.

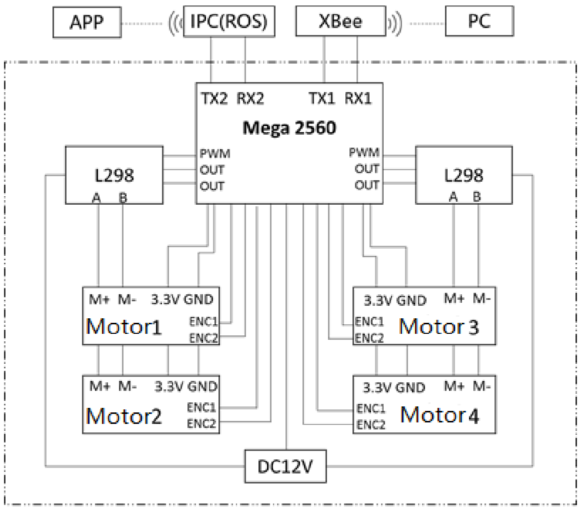

APP
IPC(ROS)
XBee
PC
TX2 RX2
TX1 RX1
Mega 2560
PWM OUT OUT
PWM OUT OUT
L298 A B
L298 A B
M+ M- 3.3V GND Motor1 ENC1 ENC2
3.3V GND M+ M- ENC1 ENC2 Motor 3
M+ M- 3.3V GND Motor2 ENC1 ENC2
3.3V GND M+ M- ENC1 ENC2 Motor4
DC12V

Fig 3. Wiring diagram of the platform

4.23D print robot InMoov

With the rapid development of 3D printing technology, the cost of 3D printing is getting lower and lower, but the printing element is more precise and complex. A 3D-printed humanoid robot like Flobi, iCub has been designed in recent years. InMoov is a full 3D print humanoid robot [12]. It was first developed by French sculptor Gail Lang, who used 3D printers to print a complete humanoid robot for low-cost humanoid robot development. It has 25 degrees of freedom for performing humanoid motions, including grasping, swinging the arm, shaking the head, etc. Plus, it has 4 facial degrees of freedom, such as the opening of the mouth or eye movement. Each degree of freedom is driven by the servo, and the motion range is similar to that of the man. In Figure 5, the upper half of the system shows the overall structure of the robot. There are two raspberry pies inside the head of the robot and each one drives a raspberry pie IP camera. The raspberry pie is connected with the router of the chassis through the network cable, which transfers the robot visual video information through the video stream of the JSMpeg web page. The video information can be transmitted from WiFi to the remote APP interface or teleoperation PC for the operator to obtain the environmental information of the robot.

4.3 Servo drive Nano node

A small Arduino Nano control core board is designed as a communication and control node which can drive 6 servos for local area modular control. The whole robot is distributed with 4 Nano nodes, which are responsible for steering control of the left hand, left shoulder and neck, right hand, right shoulder and lumbar region. Each node has a data processing and communication function, which can generate PWM wave to drive a servo. All nodes are communicated with the robot mega2560 master board via 485 Hub based on the Modbus RTU protocol. As a protocol slave, the Nano node can follow the instructions immediately, and can also store the data and drive the servos to the desired location in accordance with the time data.

4.4 Teleoperation terminal HMI

Teleoperation terminals are devices that can acquire robot visual video and give control instructions remotely. A teleoperation Android tablet PC APP HMI is built to teleoperate the robot, as is shown in Figure 4. After the network information was set up, the APP can obtain the robot visual video stream through the WiFi router and send the control instructions to the industrial computer through the UDP protocol. The type of instruction includes the rotation of the robot head, the three degrees of freedom motion of the robot chassis, the execution of complex continuous movements, and the use of Leapmotion for hand joint mapping. The operator can remotely control the robot to perform the above functions by using the APP. The other terminals such as PC which have the same protocol can perform the same function.

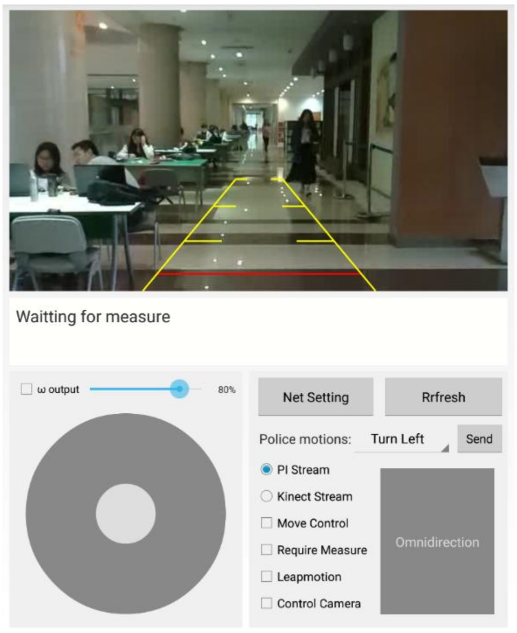


Fig 4. Teleoperation APP

The communication framework of the whole system is shown in Figure 5. The framework is a multi-core distributed structure that integrates the various parts of the robot system into the communication network. For this communication network, we formulate a set of instructions and embed them into teleoperation terminals. The operator only needs to operate the teleoperation terminal to realize the remote operation of the whole or local joint of the robot system.

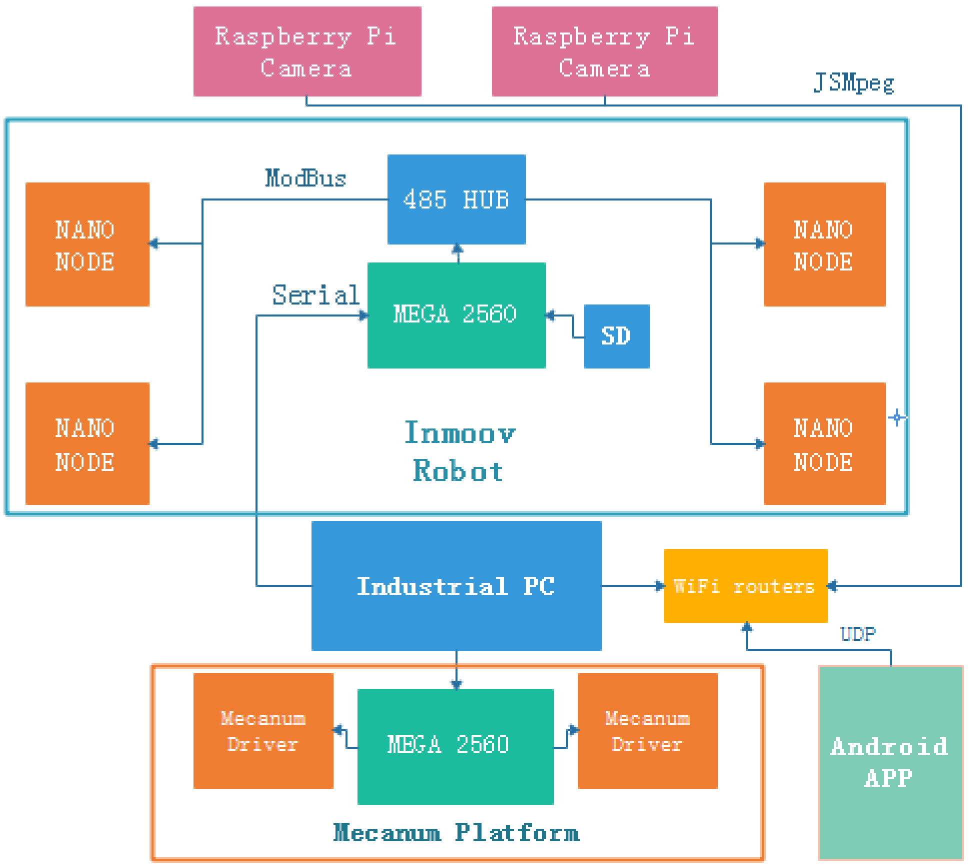


Fig 5. Communication framework of the system

V. EXPERIMENT

5.1 data flow density

The amount of data processed at each level is determined and can be calculated when an instruction is sent down. The complex action is a 20-frame action as an example. And the direct drive instruction drives one part of the robot's body. Figure 6 shows the data density flow through each level with different control strategies.

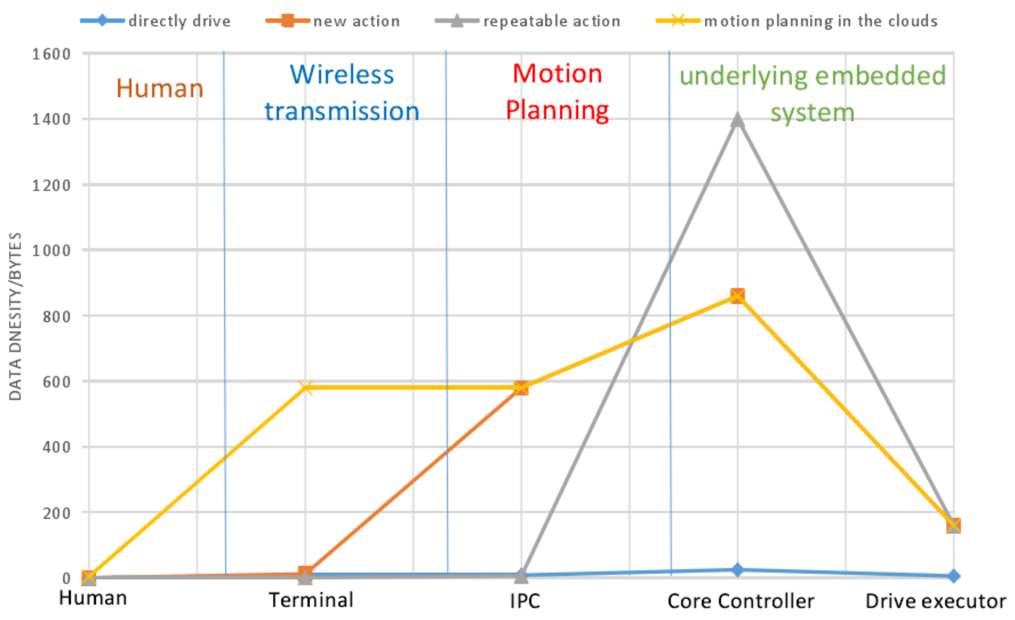


Fig 6. Density of data flows through each level

It is shown that direct drive has very little data transfered to ensure the response speed and the amount of data transmitted wirelessly is reduced compared with the conventional use of the cloud computing for motion planning. Repeatable action strategy avoids IPC motion planning calculations when a complex action is recalled. Based on the hierarchical strategy mentioned above, the robot teleoperation system can respond quickly to some control instructions to realize the real-time action and perform scheduled complex action quickly.

5.2Real-time visual angle transformation

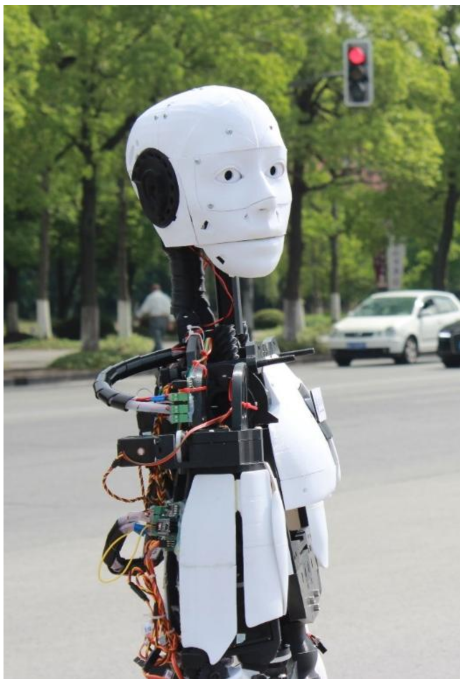
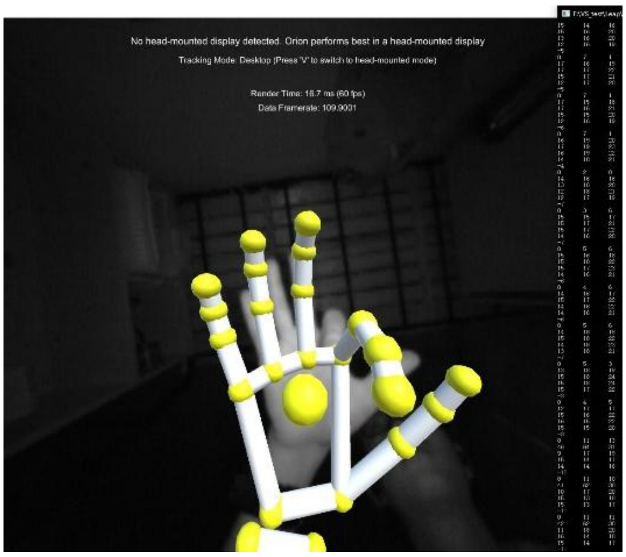


Fig 7. Head view angle turning

In order to obtain better information about the location of the robot, the flexibility of the head angle turning is very important. The response time of the system will affect the flexibility and operability of the head view angle turning. After opening the head angle control mode in the Android APP, the circular virtual control handle in the APP establishes the mapping relation with the two movement degrees of the head. The circular virtual handle can be operated to drive the head of the robot to rotate. The actual rotation of the robot head is shown in Figure 7. APP continuously detects the corresponding data value of the circular virtual handle and sends it to the IPC. The rotation angle of the head can respond to the operation of the operator in real-time.

5.3 Real-time hand joint mapping

Each finger of the robot system is a pull-type single degree of freedom structure. Limited by the mechanical structure of the robot, the success rate of the robot's autonomous grasping is low. As a teleoperation system, the operator can use Leapmotion to realize hand joint real-time mapping. It is a suitable scheme to improve the grasp success rated by human-robot cooperation. Leapmotion is a virtual reality sensor of the American company of the same name, which captures hand joint details. The details include the bending angles of the finger joints and the angle between the wrist and the horizontal plane, which are shown in Figure 8.



```
No head-mounted display detected. Orion performs best in a head-mounted display
Tracking Mode: Desktop (Press 'V' to switch to head-mounted mode)

Render Time: 16.7 ms (60 fps)
Data Framerate: 109.9001
```

Fig 8. Details of the virtual hands

The hand detail obtained by Leapmotion is actually the angle value of each joint. Teleoperation terminal maps the obtained hand detail data directly to the robot system after linear processing to realize the synchronous movement of the robot hand and the human hand. Figure 9 shows a manually assisted grasp of a cup. Hand remote real-time mapping of hand by using Leapmotion can quickly respond to details of hand movements, including finger opening and closing and wrist twisting.

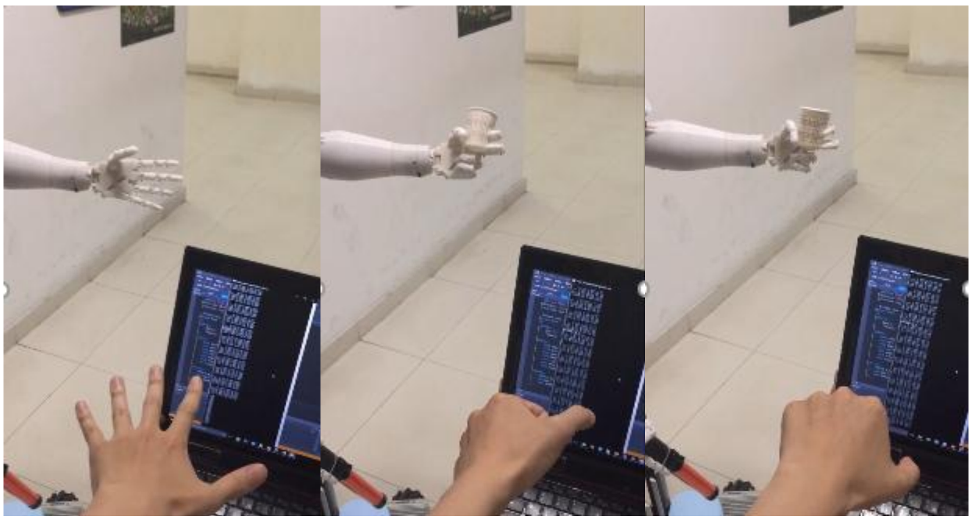


Fig 9. Manually assisted grasp

5.4 Complex motion planning and recalling

The system can use the existing hierarchical framework to complete the planning, execution, storage and recalling of complex actions. ROS is an open-source Robot Operating System [13]. And the motion planning process of complex motions under ROS is as follows.

At first, writing a URDF (universal robot description format) format Robot Description file which can be identified by MoveIt according to the model STL file of InMoov robot. And setting left and right arms as the main motion planning groups of robots by using the moveit_setup_assisstant package. Setting collision matrix to prevent robot collision and setting kinematic constraints, such as joint constraints. The above configuration is eventually saved to the SRDF file. The robot system we use is the third part robot. We also have to generate the corresponding group IKFast kinematics plug-in.

After that, we establish the InMoov robot simulation model in the visual chemistry tool Rviz. To establish the mapping relationship between the simulation model and the real robot, we do a series of practical robot motion measurement experiments. The practical experimental measurements determine the relationship between the motion of the virtual simulation robot model and the steering angle of the real robot, which will be used as a mathematical mapping between the simulation model and the real robot.

The GUI in Rviz can directly configure the joint angle values of the simulation robot. A series of joint angle values will determine a robot pose and save it. Figure10 shows the robot simulation model change of a continuous action. RVIZ will plan a moving path based on each node's speed configuration after selecting the target pose. The motion path is actually a continuous array of data containing timestamps data, and 25 degrees of motion freedom position, velocity, and acceleration data. Since the robot is servo drive, IPC will discard the velocity and acceleration data, and do the conversion for position data according to the mathematical mapping relation. The timestamps data and converted position data are repackaged and sent down through ROS Serial. The repackaged data includes the action number in the first bit，two bits of timestamp data,25 bits of position data of different degrees of freedom, and two bits of CRC16 check code.

The first action number data will determine the storage space in the SD card to store the data of the motion. The timestamp data determines the time at which the frame motion data is executed. The position data determine the bending angle of the robot in order for each degree of freedom. The CRC16 check code is used to check the accuracy of data transmission.

The planning data for the same action number will be stored in the same space in SD and executed sequentially to achieve complex continuous actions. The new planning actions will be performed after the system completes the motion planning and stores the data. The data stored has been mathematically processed in the industrial computer and can be used directly by the InMoov core board. For planned repeatable motions, the operator may choose to perform the action of the different action numbers through the teleoperation terminal. The InMoov control core board will quickly read motion data from the SD. The motion data will be sorted and packaged first in the core board’s memory and sent to the corresponding Nano node respectively through the ModBus RTU protocol. Figure 11 shows the robot do continuous action as planned. New planning actions need to wait for more than ten seconds before execution. But for repeated actions, the robot will respond within one second.

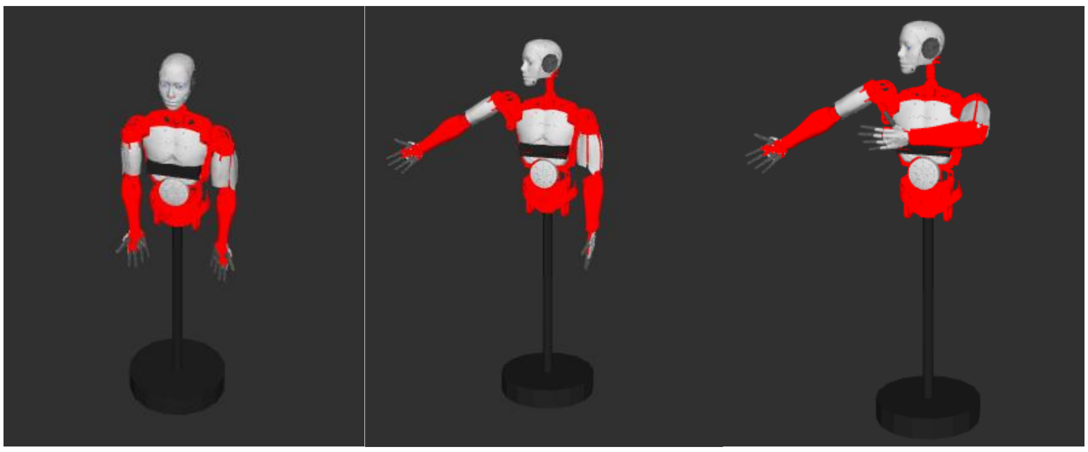


Fig 10. Robot’s simulation model change

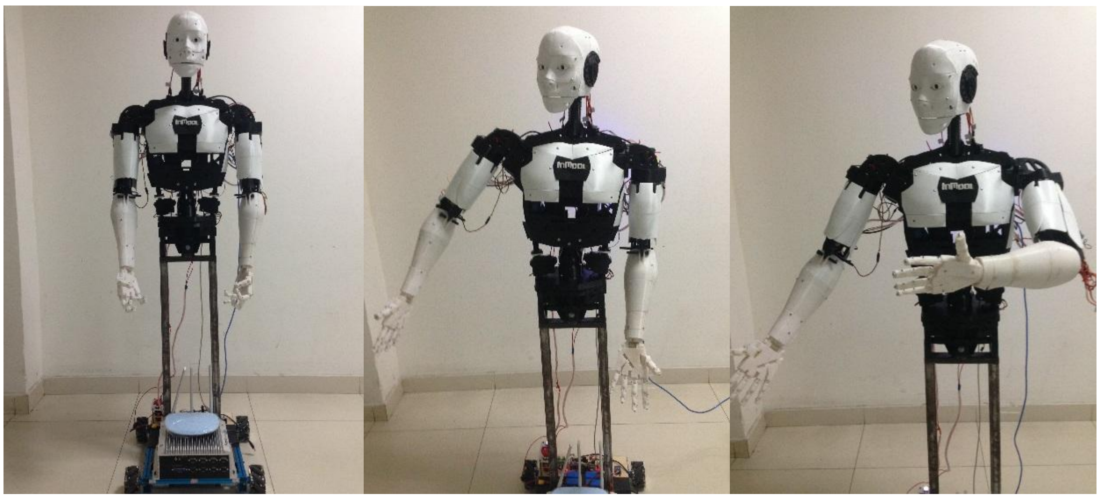


Fig 11. Robot's continuous action

Table 1 shows the delay measured of each function above. New complex motion delay is measured by taking 20-frame action as an example.

Table 1. Delay of each function

	Visual angle transformation	hand joint mapping	New complex motion	Motion Recalling
Time required	Less than 0.1s	Less than 0.1s	Nearly 5s	Less than 0.5s

VI. CONCLUSION

This paper constructs a hierarchical real-time control scheme with complexity-dependent task decomposition for humanoid teleoperation robot, which is effective for improving system response. The operator can easily teleoperate the robot to achieve real-time visual angle transformation, real-time hand joint mapping, and complex continuous action by using HMI such as Android tablet PC APP and Simulation Environment. The system has good expandability and can support multiple different terminals.

ACKNOWLEDGMENT

This work was supported in part by the National Natural Science Foundation of China under Grant 51775333, in part by the Project of Science and Technology Commission of Shanghai Municipality under Grant 18391901000.

REFERENCES

[1]. J Kofman, X Wu, S Verma. "Teleoperation of a robot manipulator using a vision-based human-robot interface," IEEE Transactions on Industrial Electronics, 2005, 52 (5) :1206-1219.

[2]. N. N. Rodriguez, G. Carbone, M. Ceccarelli, "Antropomorphic design and operation of a new low-cost humanoid robot," in Biomedical Robotics and Biomechatronics, 2006. BioRob 2006. The First IEEE/RAS-EMBS International Conference on, 2006, pp. 933-938.

[3]. G Mett, L Natale, F Nori, G Sandini, D Vernon. "The iCub humanoid robot: an open-systems platform for research in cognitive development," Neural Networks the Official Journal of the International Neural Network Society, 2010, 23 (8-9) :1125.

[4]. V Digani, L Sabattini, C Secchi, C Fantuzzi, "Towards Decentralized Coordination of Multi Robot Systems in Industrial Environments: a Hierarchical Traffic Control Strategy," IEEE International Conference on Intelligent Computer Communication & Processing 2013 :209-215.

[5]. Alireza Mohammadi, Ehsan Rezapour, Manfredi Maggiore, Kristin Y. Pettersen, Maneuvering Control of Planar Snake Robots Using Virtual Holonomic Constraints, IEEE TRANSACTIONS ON CONTROL SYSTEMS TECHNOLOGY, VOL. 24, NO. 3, MAY 2016.

[6]. Xusheng Yang, Wen-An Zhang, Member, IEEE, Li Yu, Member, IEEE, and Kexin Xing, Multi-Rate Distributed Fusion Estimation for Sensor Network-Based Target Tracking, IEEE SENSORS JOURNAL, VOL. 16, NO. 5, MARCH 1, 2016.

[7]. VJ Lumelsky, E Cheung, "Real-time collision avoidance in teleoperated whole-sensitive robot arm manipulators," IEEE Transactions on Systems Man & Cybernetics, 1993, 23 (1) :194-203.

[8]. K Natori, T Tsuji, K Ohnishi, A Hace, K Jezernik, "Time-Delay Compensation by Communication Disturbance Observer for Bilateral Teleoperation Under Time-Varying Delay," IEEE Transactions on Industrial Electronics, 2010, 57(3):1050-1062.

[9]. H Kim, JH Lee, HK Bo, KH Ha, Development of Multi-scale Cooperative Robot, International Conference on Ict Convergence, 2013:787-788.

[10]. A Gfrerrer. Geometry and kinematics of the Mecanum wheel, Computer Aided Geometric Design, 2008, 25 (9) :784-791.

[11]. KJ Åström, T Hägglund, The future of PID control, Control Engineering Practice,2001,9 (11) :1163-1175

[12]. G. Langevin, "InMoov," URL http://www. InMoov. fr/project, 2014.

[13]. M Quigley, K Conley, BP Gerkey, J Faust, T Foote, ROS: an open-source Robot Operating System[J], Icra Workshop on Open Source Software, 2009, 3.