IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received March 16, 2018, accepted April 29, 2018, date of publication May 17, 2018, date of current version June 26, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2837904

IAPcloud: A Cloud Control Platform for Heterogeneous Robots

SONG ZHENG[1,2], ZHICHENG LIN[1,2], QIJUN ZENG[3,4], RONG ZHENG[3,4], CHAORU LIU[3,4], AND HUAFENG XIONG[3,4]

[1]College of Electrical Engineering and Automation, Fuzhou University, Fuzhou 350116, China
[2]Key Laboratory of Industrial Automation Control Technology and Information Processing, Fuzhou University, Fuzhou 350116, China
[3]IAP(Fujian) Technology Co., Ltd., Fuzhou 350116, China
[4]Research Institute of Fujian Histron Group Co., Ltd., Fuzhou 350116, China

Corresponding author: Song Zheng (s.zheng@fzu.edu.cn)

This work was supported by the Fuzhou Science and Technology Project Foundation of China under Grant 2017-G-70.

ABSTRACT The cloud robotic technology makes multiple robots share resources and collaborate with each other more flexibly. Although this concept has been widely accepted by academia since it was put forward, it still faces many challenges in engineering. The main problems are that the functions of industrial robotic systems become more and more complex, and the programming language and computing environment of multiple vendor robots are significantly different, accordingly the cooperative control of different devices and the online testing become extremely difficult. Therefore, this paper proposes a new control platform for cloud robots, called IAPcloud, which has significant advantages in solving the problem of collaborative control among heterogeneous robots and their auxiliary devices, declining the programming difficulties of cloud robotic control systems and shortening the development and deployment cycle of the application. Finally, we verify the scientific and effectiveness of the IAPcloud platform by three cases, including the image recognition and tracking, the human–machine gobang game, and the online dynamic reconfiguration of control algorithms.

INDEX TERMS Cloud robot, collaborative control, heterogeneous robot, dynamic reconfiguration.

I. INTRODUCTION

In recent years, with the development of the network and cloud computing technology, the research of cloud robotic technology has drawn great attention from the academia. The main reason involves three aspects: i) the real-time control and control accuracy of industrial robots have reached a high level at the present stage [1]. However, due to the limitations of the hardware and software, traditional industrial robots only can perform tasks in a specific environment and handle massive amounts of data by the pre-programming method. In the unknown environment, these pre-compiled robots can't flexibly adapt to the needs of control applications, especially in the field of the collaborative control of complex systems [2], which causes complex configurations, cumbersome operations, and high costs. ii) In the future, robots will play a more critical role in the intelligent manufacturing field, which requires more data space and higher computing power. However, there are apparent deficiencies of robotic devices in these respects. iii) As the production process becomes more complex, multi-vendor automation products need to be integrated into a unified system to achieve the collaborative control, in this case, a third-party platform will be required to gather all the information for the complex data processing [3]. The cloud robotic technology is considered as a great solution to meet the above needs, of which the main idea is to combine the cloud computing with robots and upload heavy CPU tasks to remote cloud servers for processing. Each robot can accomplish its task relying on the control programs and the cloud server, and the robot itself doesn't need to store any information or have super computing power [4]. However, it is not easy to design such a complex cloud robotic system that integrates the functions of the data acquisition, network communications, cloud computing and robot motion control. Given the inherent characteristics of robots, it is almost impossible for cloud robots to apply the same architecture as traditional cloud computing platforms without any change [5]. In order to design and implement an ideal cloud robot platform, we believe that it is necessary to provide a platform-based service (PaaS) in the cloud computing environment, which can not only adapt to the demand of the flexible production, but also meet the requirements of the collaborative control of heterogeneous robots. In addition, it is essential to ensure the consistency of the computing environment in the cloud and the edge. So we propose a cloud robotic control system based on multiple virtual controllers, IAPcloud.

The rest of this article is structured as follows. In Section 2, the related works for cloud robots is reviewed. Section 3 presents an overall architecture of the control platform for IAPcloud. Section 4 details the design and development of virtual controllers. Section 5 describes the edge controller of IAPcloud and its workflow. Section 6 tests the proposed robotic control platform architecture by 3 cases of heterogeneous robots and their auxiliary equipment. In Section 7, we evaluate the features and novelty of the IAPcloud platform. Section 8 summarizes the conclusion and presents a future outlook.

II. RESEARCH STATUS OF CLOUD ROBOT

At the Humanoids international conference in 2010, the concept of ''Cloud-Robot'' was first proposed by Dr. Kuffner J (Google company), which immediately caused a wave of research in academia [6], [7]. These studies cover many technical issues including the architecture, network communications, cloud computing, heterogeneous system integration and engineering practice of cloud robots, as well as their related technical challenges [3], [4].

The cloud robotic framework is one of the most important topics in the cloud robotic research. Usually, a cloud robotic control system includes multiple robots, auxiliary devices and multi-layer networks. Therefore, it is necessary to deal with the management problems of such a complex system when designing its framework. In recent years, many cloud robotic frameworks such as RoboEarth, DAvinCi, Robot-Cloud, C2TAM, Rapyuta and CoTeSys have been proposed [8]–[13]. RoboEarth is a research project conducted by European scientists, who aim to provide robots with a cloud platform for information sharing and mutual learning [8]. Raptuya is a cloud robotic platform based on RoboEarth that devised to optimize the robotic task management, data structures, communication protocols and so on [12]. The ''Robot-Cloud'' proposed by Doriya et al. is a cloud robotic framework that designed for the integration and collaborative control of heterogeneous robots. This framework is implemented with components including the cloud controller, ROS master, storage unit, Map-reduce cluster and robot cluster, in which each robot has installed ROS (Robot Operating System) and the robotic service is published through the master node in the cloud controller [10]. Several common features can be found in these cloud robotic frameworks, i.e., (a) most of the existing platforms support to offload the compute-intensive tasks of the control system from robots to the cloud infrastructure, to take full advantage of the cloud computing resources and big data processing capabilities; (b) the service-oriented architecture (SOA) has been commonly applied to design and develop the cloud robotic architecture. However, no specialized computing pattern has been introduced to the virtual machines of cloud servers in these cloud robotic systems, thus the complexity of the multi-robot or multi-device system development will not be significantly reduced by these cloud robotic methods.

In industry, the integration and the cooperative control of heterogeneous multiple robots are the most critical issues in the cloud robotic research and also one of the most typical applications [23]. Most studies of cloud robots use the ROS platform to solve heterogeneous problems of different robots. For example, the CRCF framework combines the advantages of ROS in robotic algorithms and the hardware device management, and allows them to be separated from the cloud, resulting in more flexible and convenient management [24]. However, the existing control systems are strictly limited in the selection of robotic hardware types. Many excellent robotic products are difficult to be accepted by the cloud robotic platforms because they can't be compatible with ROS.

Since the distributed feature of cloud robotic framework, data communications have become the most important part of the cloud robotic technology. In recent years, specific IoT protocols such as XMPP, MQTT, LTE, 5G and UDP have been selectively applied to the cloud robotic control system for the connection between local robots and the cloud platform to improve the communication efficiency [14]–[20], [26]. Some scholars have also conducted in-depth research on the uncertainty of the network communication delay in cloud robotic systems. To solve the message delay problem caused by applying the gossip protocol in the mobile robot system, Hu *et al.* [21] reduced the time cost of spreading messages in the network by using the infrastructure cloud as a central node. In order to ensure the high efficiency of the data transmission, Du et al. adopted a staged transmission method to filter the low-priority data, and then transmit the high-priority data through idle channels [16].

In fact, the design of data format and communication mechanism will have a significant impact on the multi-device collaborative control in the cloud robotic system. Due to the difference of the computing environment and communication ability, the multi-robot system has great heterogeneity. It is necessary to unify the data format of heterogeneous robots to eliminate this difference through a new communication mechanism [22]. In addition, the data connection pattern between the cloud and the edge may suffer dynamically changes when the system is adjusted to meet the flexible production requirements. However, there are few studies that discuss this issue.

The concept of ''robot as a service (RaaS)'' is proposed to improve the utilization of cloud resources and the efficiency of multi-robot coordination to complete complex tasks. All the functions of different robots are encapsulated into specific services, and services are communicated through standard interfaces and protocols [25], [27]. Although many robotic vendors have provided an open library for the development of collaborative control programs, most cloud robotic control algorithms are still implemented in advanced programming languages such as java and C++ at present [27], [29], of which either the programming efficiency or the visual calculation needs further optimizations. It is worth noting that this programming method also restricts the possibility of online optimization of the cloud robotic control program to a certain extent.

Currently, the development of cloud robots is still in the early stage [17]. Although most researchers have verified the feasibility of the proposed framework through specific experimental systems, there is a long distance between the maturity of cloud robotic technologies and the requirements of their engineering applications from an overall perspective. One of the main reasons is lacking of technology standards of the multi-robot control which can bridge the gap between the cloud-based virtual environment and the edge-processing systems. The standards should aim at achieving the steady migration of robotic control programs from the cloud to the edge side and can also realize the collaborative control among heterogeneous robots. Unfortunately, we find that the existing cloud robotic frameworks can't well meet such demands. But recently Grieco et al. suggest that applying the Internet of Things (IoT) technology to the robotic control systems will help to unify the data formats of heterogeneous robots and seamlessly integrate multiple robots, which may be an effective way to achieve this goal [28].

In recent years, our research team has proposed a universal control configuration technology that can solve the problem of communication, interoperation and collaborative control among heterogeneous devices in the Internet of Things [29]. This technique has been verified in several application scenarios such as the automatic obstacle avoidance of mobile robots [32], the trajectory planning of industrial robots [31], the heterogeneous integration of smart home devices [30] and the ship integrated control [33]. It also has the remarkable superiority in the characteristics of the efficient programming, visual calculation and dynamic reconfiguration of robots. In this paper, we try to introduce this general technology to cloud robots, and develop a unified computing environment between the cloud and the edge system to reduce the cost and difficulty of the design development and deployment of the cloud robotic collaborative control applications.

III. CLOUD ROBOTIC ARCHITECTURE

The clearly insight into the architectural requirement is extremely beneficial to design a new cloud robotic architecture. At present, two strategies are conducive to improving it. First, making full advantages of the powerful computing ability and abundant storage space of cloud servers, and offloading various complex computing tasks from the robots to cloud servers for processing. Second, introducing new architectural concepts, such as setting the robots as a part of the cloud services, that is, robot as a service (RaaS), or merging the IoT and robots to form an Internet of Robotic Things architecture (IoRT) [34]. What these studies have in common is: the computing environment of cloud servers is provided, but the software engineering methods of cloud robots have not been changed. In view of the above drawbacks of existing systems, it is necessary to design a new cloud robotic architecture that can effectively adapt to the change of the flexible production. The expected new architecture should have the following features: (a) the cloud robot has an ''edge brain'' in the local, in order to ease the network traffic congestion caused by complex calculations and avoid that robots are unable to think and make decisions caused by poor network communications; (b) the computing environment of ''cloud brain'' and ''edge brain'' must be consistent, and robotic control programs can adapt to both cloud and edge systems simultaneously; (c) it should have a good data interaction mechanism for the robotic control, which not only ensures the consistency of data between cloud and edge systems, but also supports the online reconfiguration of cloud control algorithms without disturbing the running of local robots. (d) the architecture is flexible, adjustable and supports all types of robots with open programming interfaces. The newly added robots can be integrated quickly into the cloud robotic team without affecting the original control system architecture; (e) it should avoid developing control algorithms using complex high-level programming languages, and replaced by a more flexible method with features of the computing visualization and high reusability of computing resources; (f) the cloud provides good robotic services and the computational units need to be well-designed to accommodate the assignment of complex collaborative control for multiple robots.

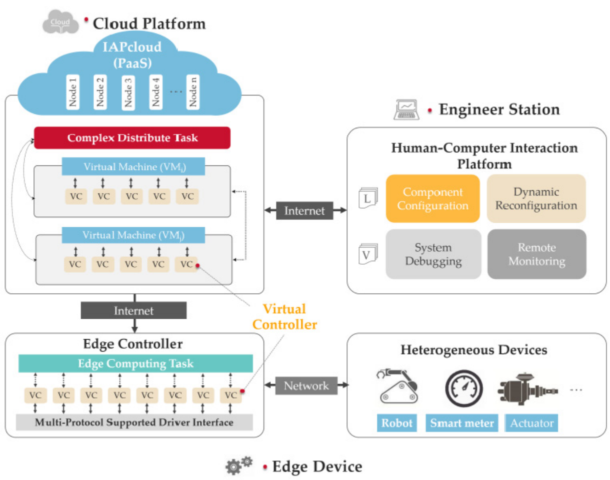


FIGURE 1. Cloud robotic architecture.

According to the above discussions, considering the factors such as scalability, interoperability, modularity and unified data interface, we design the IAPcloud architecture (see Fig. 1). In IAPcloud, cloud robots consist of three parts: the cloud (virtual controller cluster), the edge (edge controller, robots and their ancillary devices) and the engineer station (human-computer interaction platform). These three parts form an integrated system by the internet, among which the edge controller is connected to the robots and the auxiliary equipment via a Local Area Network.

The cloud platform of IAPcloud is the core part of cloud robots. It uses the PaaS architecture and is responsible for the complex task calculation and data storage including the multi-robot task scheduling and intelligent algorithms. As the most important computing infrastructure of cloud robots, in IAPcloud, a large number of virtual controllers with identical internal structures are usually deployed in the cloud. And the complex control tasks of multiple robots will be executed through the collaboration of these virtual controllers. The purpose of this design is to ensure the consistency of the cloud robotic system structure and handle the complexity issue of control tasks. IAPcloud uses the Docker technology to create a lightweight management environment for multiple virtual controllers in cloud servers. Firstly, the virtual controller is a kind of industrial micro operating system based on the data-driven technology [41], and it is also a kernel program specifically designed for handling robotic control configuration algorithms and possesses the powerful data processing capability. The cloud service of IAPcloud can be programmed in a particular environment using the graphical configuration method based on components. Second, in the cloud platform, the complex control tasks of the cloud robotic system can be decomposed into a large number of independent agents. In other words, the robotic control programs can be assigned to different virtual controllers (computing units) according to their functions. Different virtual controllers connect to each other through the data bus to form a multi-engine computing environment. In this way, different task agents can perform complex tasks cooperatively with each other through the data exchange. The advantage of this structure is that it simplifies the complex problem of robotic control to a large number of simple software agents, which is very beneficial for improving the development efficiency of the system.

The edge controller (IAPbox) embeds a computational unit consistent with the multiple virtual controller systems in the cloud, that is, the entire cloud robotic system can use a unified technical standard. Moreover, the algorithm configuration procedures and the design of human-machine interaction interface (HMI) are implemented in the engineer station by using the human-computer interaction tools. The robots and their auxiliary devices are all connected to IAPbox to form the IAPcloud edge system. IAPbox can connect multiple robots or other devices (such as smart meters or actuators). Usually, these devices are heterogeneous and use different network protocols. In IAPbox, all data communication of these heterogeneous devices can be handled separately through the embedded multi-protocol communication drivers. In order to remotely monitor or manipulate the cloud robots, virtual controllers in the cloud should be accessed by IAPcloud engineer stations. It indicates that the cloud platform must be equipped with corresponding virtual controllers for robots in IAPcloud, and all the real-time output of the robotic control instructions and their state detection can be communicated with robots through these virtual controllers.

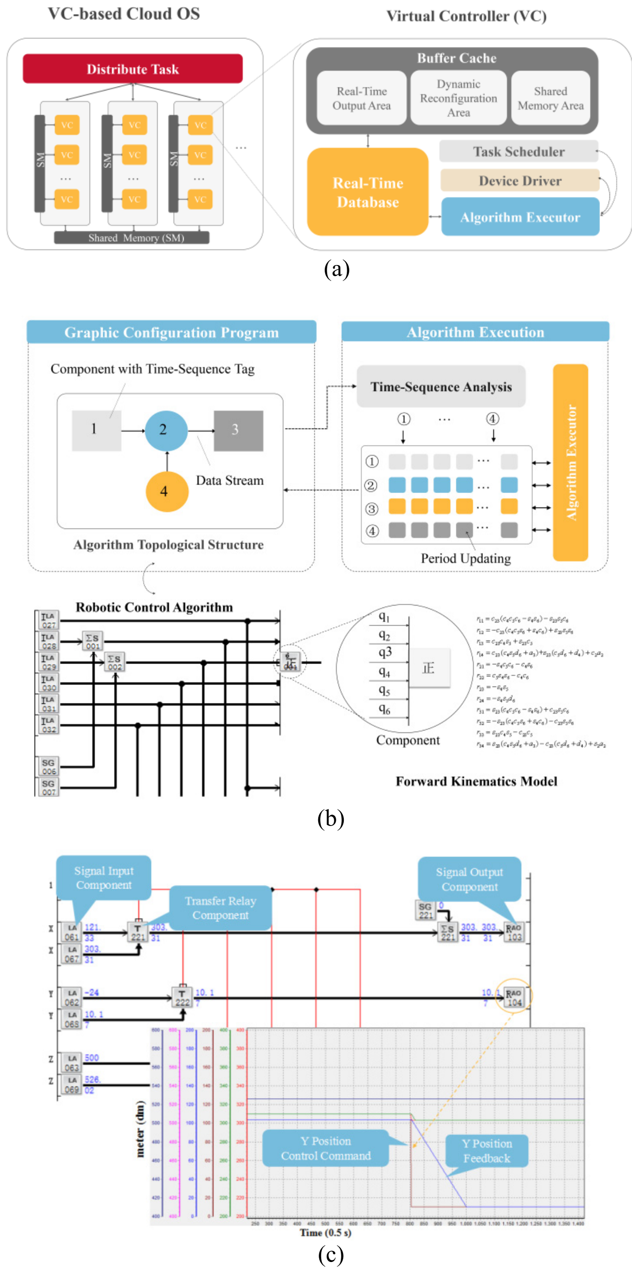


FIGURE 2. How the virtual controller works with the control configuration. (a) The principle of IAPcloud virtual controllers. (b) The equivalent relationship between control algorithms and component configurations. (c) An example of the robotic motion control configuration and monitoring curve.

IV. VIRTUAL CONTROLLER DESIGN AND DEVELOPMENT

A. DESIGN PRINCIPLE

In this research, the virtual controller [36] structure and their runtime environment are specially designed. Fig. 2 (a) and (b) show the principle of the virtual controllers in IAPcloud. Fig. 2 (a) presents the connection of multi-virtual controllers in the cloud and the internal structure of each controller. In general, the robotic complex control tasks are processed in the virtual machine of the cloud server. Each virtual machine contains a group of distributed computing clusters that composed of multiple virtual controllers, and each virtual controller just needs to complete a relatively simple control task. The virtual controller consists of several parts including the memory database, the task scheduling module, the algorithm executor and the buffer area. Among them, the complex control data generated by the robots and auxiliary devices are stored in the real-time database. The algorithm executor is a core part which specifically designed to handle the configuration of control algorithms.

Fig. 2 (b) shows the equivalent relationship between control algorithms and component configurations. Instead of being programmed by the high-level programming language, cloud robotic control algorithms are implemented by a component-based configuration method that is independent of the control objects. Each type of components is encapsulated with a specific function. The topological connection of components represents the recursive relationship of the control states, the data transfer flow and the encoding process of the robotic control algorithms. The purpose of this design is to enhance the readability of the control algorithms and create the conditions for the visual computation so that the design and development of the robotic control algorithm can be completed intuitively and the algorithm debugging process can also be simplified. Although virtual controllers cannot accept the graphic files of the component configuration as their executable programs, it can receive the configuration data which are transformed from the component configuration. These data inside the component mainly include the state of the component variables, the algorithm information, the component parameters and the results of the last calculation cycle. Once these data are transmitted to the real-time database, a periodic data update will be triggered in the virtual controller. As shown in Fig.2 (b), there is a specified sequential relationship among the components, and the virtual controller will conduct the computation of each component according to its time-sequence tag.

Taking the Mitsubishi robot as an example, Fig. 2 (c) shows the graphical configuration logic and the trajectory of the manipulator endpoint when it moves from point A to point B at the speed of 5 mm/s, the robotic control program can be realized simply and quickly by using a few types of components, such as signal input components, signal switching components and signal output components, and setting the parameters and the connection relationship.

In addition, virtual controllers support the mixed computation of real numbers and Boolean data. And the computation cycle should be set short enough to meet the requirements of the real-time control of cloud robots. The virtual controller provides the data cache space to support the communication with the outside world, which includes three essential parts: one part is used for storing the real-time output of the components during their computing process to achieve the visual computation; one part is used for storing the dynamic reconfiguration data to perform the undisturbed update of control algorithms without stopping the virtual controllers; another part is used for the data sharing among different virtual controllers. Therefore, we can well accomplish the collaborative control tasks of cloud robots in the computing environment of multiple virtual controllers.

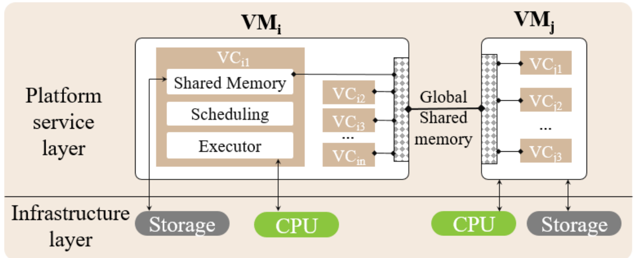


FIGURE 3. The resource management of virtual machines.

B. RUNTIME ENVIRONMENT

In IAPcloud, we also design a specific resource management mechanism while introducing the virtual controller technology. As shown in Fig 3, IAPcloud contains the infrastructure layer and the platform services layer. In the platform service layer, a large number of virtual controllers are deployed in the virtual machines to form multiple virtual control nodes. These control nodes process the cloud robotic control instructions by reading and writing the memory data. When the virtual machine runs, each virtual controller will automatically divide a data area from the memory which is applied for storing and updating the real-time data of control algorithms.

The algorithm executor and the task scheduling module of virtual controllers play a key role in the resource management of virtual machines. Within a virtual machine, the algorithm executor can directly invoke CPU resources, and the algorithm executors of multiple virtual controllers share the same physical CPU, which only responds to the invocation request of these executors. The task scheduling module adds the specified components to the task queue according to the algorithm configuration, and then the algorithm executor reads the component internal data to complete the algorithm calculation. Furthermore, the virtual controllers share data by the data bus to achieve data synchronization, and these shared data are periodically updated. More specifically, each virtual controller has a local shared memory area, which is used to synchronize data with other virtual controller members in the same virtual machine, and the data interaction of different virtual machines can be realized through their global shared memory. This mechanism is also beneficial for the stable data processing of robotic collaborative control. It should be noted that when dealing with the complex collaborative control tasks, the virtual controllers need to map the component topological model to the data model, allocate the memory address and update the real-time data.

C. RECONFIGURABLE CLOUD SERVICES

IAPcloud provides cloud computing services for multi-robotic control tasks. Different robots or control subsystems can cooperatively complete specific control tasks by cloud services. Because of the complexity of multi-robot control tasks, it is an optimal way to introduce the dynamic reconfiguration technology to cloud services, which can ensure the reliability, security and diversity of the robotic system when modifying cloud service algorithms online. Dynamic reconfiguration is the technology that dynamically updates the control algorithm when the cloud control system is running, and it is closely related to the control configuration model and its computing environment [35]. It uses a type of data unit with a fixed structure format as the control algorithm model, and its modification does not affect the normal work of the related software. In IAPcloud, we take the component as the control algorithm model and realize the dynamic reconfiguration of the cloud service algorithms by five kinds of operations, including adding, deleting and replacing components, adjusting their topological relations, and updating the control configurations.

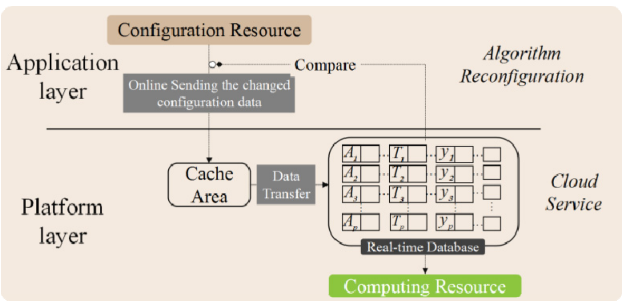


FIGURE 4. **Reconfigurable cloud service.**

Robots connected to the cloud platform can apply for cloud services using reusable component resources. When we submit a new cloud service application to the system by a configuration program, the related component resource will be reset in the command queue of the virtual controller according to the new data relationship. The algorithm executor is responsible for the data analysis and calculation, the update of the real-time database and the device driver management. The software process of the algorithm executor is only relevant to the content of the data units (components). Therefore, the process has the characteristics of reusability. In such a computing environment, the control process is essentially transformed into the updating process of the real-time database. Because the virtual controller only accepts the configuration data that converted from the robotic control algorithm, it just needs to update the specified configuration data in the memory area within one scan cycle. The principle of the algorithm dynamic reconfiguration is shown in Fig. 4. When performing the algorithm reconfiguration of the cloud service, the system will compare the differences between the new configuration data with the real-time database. Only the changed configuration data will be sent to the cache area of the virtual controller. When the virtual machine receives the reconfiguration request, it will stop the computing services of all the current components and only update the changed data to the real-time database. Either the software process of virtual controllers or the structure and location of algorithm variables in the memory will not change. For example, A represents the absolute address of components in the memory area of the virtual controller, T represents the topological arrangement of components, and y represents the target of components (i.e. the output of the component which is used as an external drive of the next component). Assuming that the external drive of the component P comes from the components 1, 2 and 3, it is clear that A and y will not be changed with the topological relationship. Therefore, this process will not greatly interfere with the normal running of robots. This design also makes the outputs of the robotic system keep consistent during the execution process of the algorithm reconfiguration.

V. CLOUD ROBOTICS EDGE CONTROLLER

A. DESIGN INTENTION

The network communication is a major part of cloud robotic system. Usually, a robot or device can be designed as a network node, and the difference of their communication method will have a great impact on the system integration and the data exchange. There are many types of industrial robots and their auxiliary equipment, and the supported network communication protocols are also different which makes their communication hard to be compatible with each other. The existing cloud robotic system architecture mainly connects local robots and cloud service platforms via the internet, and most of them are designed for robots (such as mobile robots) that only support certain communication protocols. And there are few studies focus on the industrial robots with various network protocols. Besides, the main reason why most cloud robotic architecture over rely on cloud servers is lacking the edge computing ability. To solve the network communication problem, it is necessary to introduce a gateway device that is compatible with a variety of robotic communication protocols, when heterogeneous industrial robots access to the cloud control platform. Thus, this study uses IAPbox as the edge controller of cloud robots to deal with the communication problem of heterogeneous industrial robots and the auxiliary equipment, and the data processing issues. Because IAPbox edge controllers are deployed near the data sources of robots and less susceptible to network delays, they have the ability to respond quickly and help to handle the multi-robot local control tasks efficiently.

B. STRUCTURE AND PRINCIPLE

When designing an edge controller, we need to consider the communication framework of the control system and different robots or device products. Presently, industrial robotic products usually have the powerful network communication capacity, which provides a very favorable condition for the development of cloud robotic control systems. As shown in Fig. 5, IAPbox has designed a multi-protocol driver interface which can be embedded in various communication

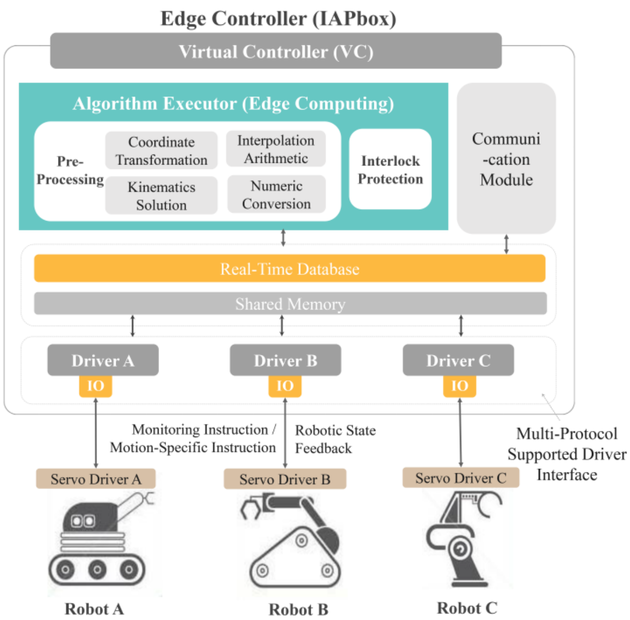


FIGURE 5. The principle of the edge controller.

protocols such as RS232 / RS485, CAN-BUS, Modbus and Ethernet to adapt to the communication specifications of different vendors [30]. The role of IAPbox is to collect the state data of robots and related devices, especially the internal state data of robots, including 3D coordinates, angular velocity, angular acceleration etc., and transmit the motion control instructions to the robotic servo drivers.

For example, the communication system between the Mitsubishi manipulator and IAPbox adopts the C/S framework model (Client/Server), in which IAPbox is used as the client, and the manipulator is used as the server. Each of them uses a specified data protocol to pack the sent data and parse the received data. The communication cycle of receiving the service request by the manipulator is 7.1ms. Therefore, the IAPbox must send the real-time monitoring instructions or motion control commands and receive the corresponding state feedback data of the manipulator at the same time interval. In such a short time interval, IAPbox continuously transmits the coordinate data of the manipulator endpoint to achieve the approximate effect of continuously driving the manipulator. After IAPbox sends the real-time monitoring instructions, it can receive 70 feedbacks including the current coordinates and angle of the manipulator endpoint, the angle of current joint, the pulse number of current servo motor driver, the communication status, the acceleration state, the temperature of the servo driver card, the motor electricity, the motor speed, the joint load, the encoder temperature, the force sensors and other data. These data are first pre-processed by the communication driver of IAPbox and then sent to the corresponding input components of the virtual controller for processing at the edge. When IAPbox transmits a motion control instruction to the robot, 28 outputs of the components in the virtual controller will be packaged and sent by the communication driver in specific format. These data include the communication data type, the target coordinate and angle of the endpoint, the target joint angle, the number of specified motor pulses, and so on. Thus, when the cloud platform sent the robotic motion instructions to the edge, the control algorithms need to be processed in IAPbox for gaining the appropriate data combination of control instructions.

And in general, the cloud robotic system based on IAPcloud contains many field devices, and the data collected by the edge controller needs to be preprocessed locally, and then the valuable data will be uploaded to the cloud for complex calculations, so as to reduce the pressure of bandwidth consumption and avoid the core network congestion. Meanwhile, as the "local brain" of cloud robots, virtual controllers must be deployed in IAPbox for the standardization of the edge computing environment. As shown in Fig. 5 and Fig. 6, in the edge system, the real-time data of each robot or device can be dynamically collected into IAPbox through the relevant drive interface and stored in a designated memory address in the virtual controller. And the robotic state data will be mapped to the specified input components of the edge control algorithms for preprocessing, and finally uploaded to the virtual control nodes in the cloud. Similarly, these data will also be read to some input components in the cloud for more complex data processing.

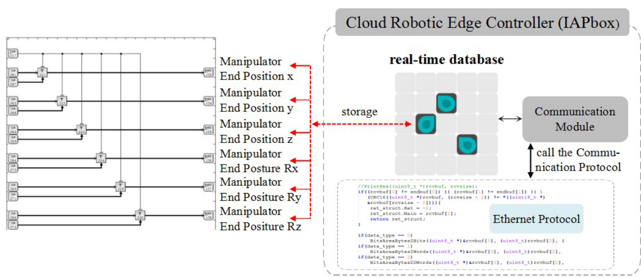


FIGURE 6. An example of Mitsubishi manipulator control.

In addition, how to guarantee the reliability and security of cooperative tasks of the multi-robot community is another problem we have to consider. Therefore, some important interlock protection functions also need to be set in the edge controller. IAPbox can protect the critical data or variables by locked and unlock operations to avoid suffering from safety risks caused by incorrect operation. At the same time, a local protection mechanism is also designed in IAPbox. When there is a network interruption between the cloud and the robots, IAPbox will send an emergency stop command to the robotic servo driver quickly to stop the robots.

VI. CASE STUDY AND VERIFICATION

A. VERIFICATION SYSTEM DESIGN

In this paper, we design a cloud robotic verification system, of which the hardware consists of the cloud servers (FusionServer RH2288 V3), edge controller, engineer station devices, three different types of robots and their servo

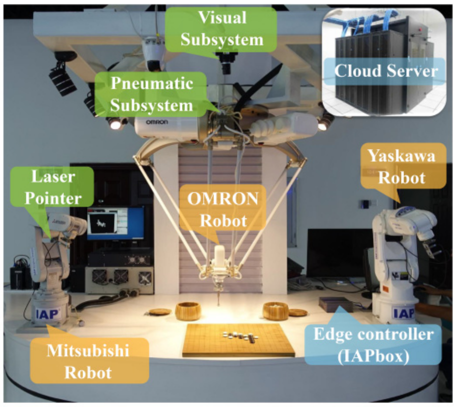


FIGURE 7. Cloud robotic verification system.

drives (including Mitsubishi MH3F manipulator, Yaskawa R6Y3 manipulator and OMRON Delta R6Y3 parallel robot), as well as the visual camera, laser pointer, suction cup, solenoid valve, pressure switch, vacuum pump, network device and other auxiliary devices. As shown in Fig. 7, three different types of robots and their actuators in the verification system form the motion control subsystem of heterogeneous robots.

Each robot and its servo driver are connected to the edge controller via LAN (Local Area Network), the edge controller controls them to work and monitor their state by calling internal specific communication protocols. The edge controller and the cloud platform communicate with each other through the internet. The endpoint of the Mitsubishi manipulator and Yaskawa manipulator are respectively fixed with a laser pointer to show the target position of the stone on the board, the visual camera is used to capture the board image, and the endpoint of OMRON parallel robot is fixed with a suction cup. The vacuum pump, trachea, solenoid valve, pressure switch and suction cup are connected in series to form a controllable gas passage for sucking up the stone. The solenoid valve can be controlled by the driver of the OMRON parallel robot in the edge controller. Since the verification system subtly covers several typical application subsystems in the cloud manufacturing field, including the heterogeneous manipulator motion control subsystem, the visual subsystem, the pneumatic subsystem, as well as other universal third-party control devices like the edge controller and cloud platform. Therefore, it can well meet the cloud robotic technologies and their future research needs.

B. CASE STUDIES

In this study, three cases based on the verification system are designed to verify the feasibility of the proposed cloud robotic architecture, the cooperative control performance and the human-machine interaction capability for various heterogeneous robots and auxiliary devices.

Case 1 (Collaborative Control Between the Robotic Motion Control Subsystem and the Visual Subsystem): This case is aimed to validate the cooperative control ability between the robotic motion control subsystem and the visual subsystem in IAPcloud. Taking Mitsubishi robotic manipulator as a control object, the spatial location of the moving target object is found through the image recognition (OMRON FH series industrial vision system). With the help of the image subsystem and the motion control subsystem of the manipulator, the laser pen of the manipulator endpoint will be guided to track the moving target.

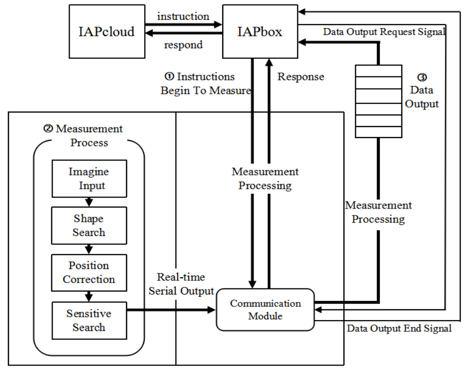


FIGURE 8. The image recognition workflow.

1) IMAGE RECOGNITION

The workflow of the image recognition is shown in Fig. 8 When IAPcloud sends an "image position measurement" control instruction to IAPbox, IAPbox will trigger the vision subsystem to control the camera to capture the image information. Then the visual subsystem matches the 2D graphics in the image with the user-preset graphic template by a shape-based image retrieval algorithm [39] to calculate the spatial position of the target image. Both the position correction and sensitivity search methods are used to improve the position accuracy of the target image, which can be used as the basis of the motion control of the manipulator endpoint. To ensure the data transmission efficiency and avoid the communication blockage, IAPbox and the visual subsystem use the User Datagram Protocol (UDP) protocol to transfer the image coordinates.

2) MOTION CONTROL OF THE MANIPULATOR

In order to track the moving target, the cloud robotic system needs to compute the relationship between the position and angle of the laser pointer and the spatial position of the moving target. We can use (1) to achieve the image coordinate transformation from the visual subsystem to the robotic reference coordinate system.

$$\begin{bmatrix} xR \\ yR \\ zR \end{bmatrix} = R_V^R \begin{bmatrix} x_V \\ y_V \\ z_V \end{bmatrix} + P_V^R \tag{1}$$

Where xR, yR, zR represents the coordinate of the laser pointer based on the robotic reference coordinate, xV, yV, zV is the target image coordinate recognized by the visual subsystem., R_V^R and P_V^R are the rotation matrix and the translation

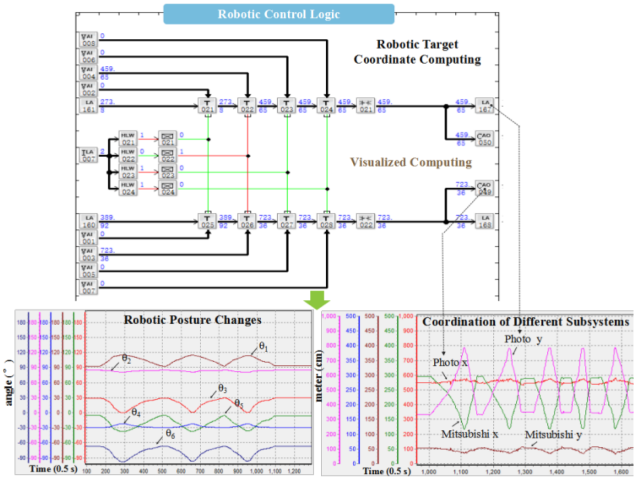


FIGURE 9. **The control configuration of robotic image tracking.**

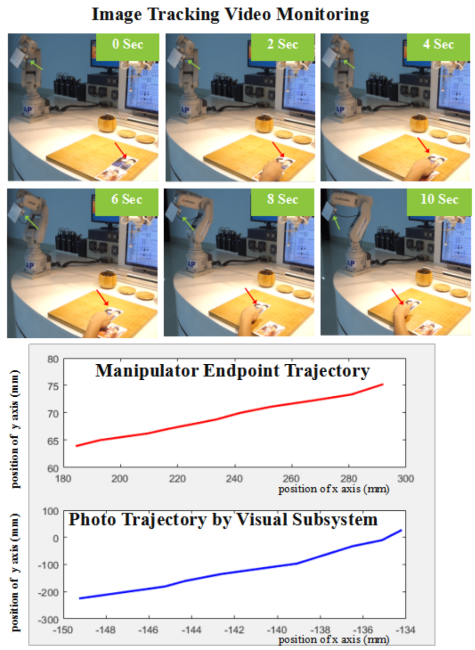


FIGURE 10. **The dynamic process and trajectory of robotic image tracking.**

matrix from the reference coordinates of the visual subsystem to the robotic motion subsystem. To achieve the precise tracking of the manipulator and the image, we gather a set of coordinates of the manipulator endpoint y and the image position based on the visual subsystem $f(t) = 2\sin\left(\frac{2\pi}{250} \times t\right)$, then perform the coordinate transformation using (1) to get the new coordinate of the image xR. After that, there will be a linear spatial correlation between the manipulator endpoint coordinate xR and the transformed coordinate of the image xR, which can be represented by (2).

$$\frac{x - x0}{x2 - x0} = \frac{y - y0}{y2 - y0} = \frac{z - z0}{z2 - z0} \tag{2}$$

The above computations including the image coordinate transformation, the forward and inverse kinematics of Mitsubishi robot, the trajectory planning and other algorithms are completed by the graphical components. Fig. 9 shows the calculation logic of the target position of the manipulator endpoint. It can be observed that the computing process of the components in the configuration program is visualized, and all the running state of the device can be dynamically captured and monitored by these components. These state data not only include the real-time angles of the six joints and the endpoints coordinates of the manipulator, but also include the real-time position of the target image. Fig. 10 further shows the dynamic process of the robot-image tracking experiment. From the motion trajectory of the manipulator endpoint and the image, we can see that the Mitsubishi manipulator can well track the movement of the photo image in real time with a smaller error. The results also show that the cloud robots and their related smart devices, including the image recognition subsystem and the robotic motion subsystem, can be well coordinated with each other, and the related applications can be easily achieved by the configuration tools and virtual controllers of IAPcloud.

Case 2 (Motion Control of Human-Robot Gobang Game Subsystem): The purpose of this case is to verify the cooperative control ability of heterogeneous multiple robots and their control subsystems in IAPcloud and the human-machine interaction capacity. A robot-assisted human-AI (Artificial Intelligence) gobang game system is designed in this case.

3) THE WORKFLOW OF THE MAN-MACHINE GAME

Human can remotely observe the game change and send the instruction of placing a stone by the human-computer interaction system. The interpretation of human minds and the operations of placing a stone are completed by the cloud platform, the robots and the auxiliary devices. The AI (artificial intelligence) algorithm can recognize the game with the help of the vision subsystem, and then analyze the situation and make decisions in the cloud. That is, the AI procedure will search downward from the initial position (the root) to form a game tree that involves all play strategies, use the maximum minimum search algorithm [38] to find the best legal position, and use the alpha-beta pruning strategy [39] to simplify the search process.

The play-stone operation of AI is also completed by robots and their auxiliary subsystems. When the game starts, according to the target position of the stone determined by human's operation on the HMI (Human-Machine Interface) or the optimal position of the stone that judged by AI in the cloud server, the HMI system cooperates with the OMRON robotic motion control subsystem to simulate the mode of manually grabbing the stone. During this process, the Mitsubishi manipulator and the Yaskawa manipulator will move cooperatively to identify the target position of the stone by their laser pointers. The OMRON robot will make the decision that whether to move the robotic endpoint suction cup to adsorb a stone according to the judgment of the vacuum degree of the suction cup. And an interval-stopping approach and the coordinate

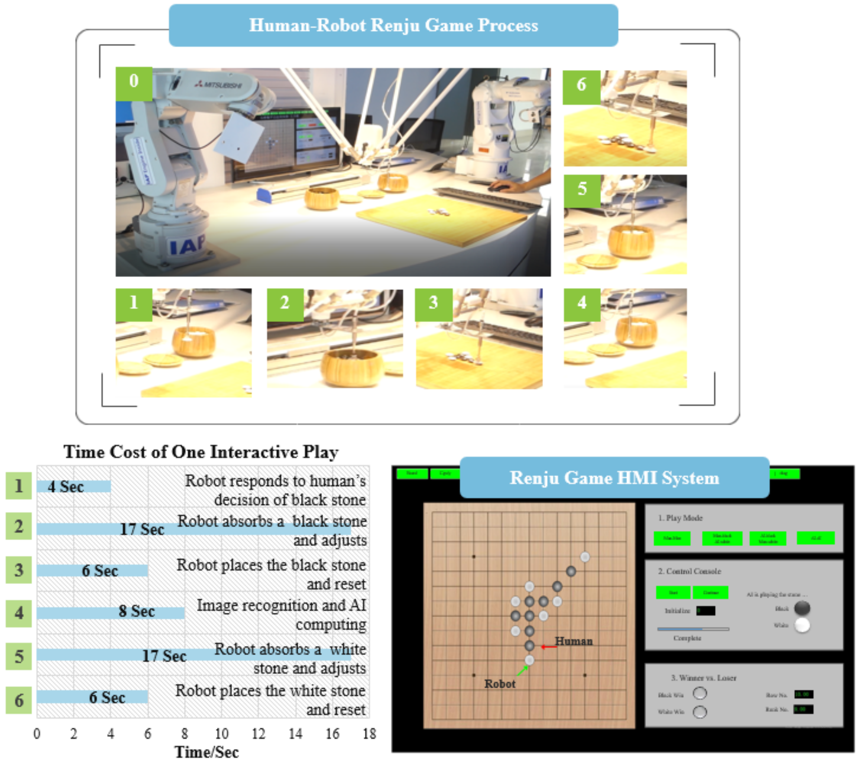


FIGURE 11. The process of Man-machine gobang game, the time cost and the monitoring interface.

adjustment method of the visual subsystem are used to deal with the skew problem during the stone adsorbing process, so as to accurately place the stone from the stone basket to the specified position on the board. In the meantime, the cloud server also needs to carry out some necessary computations like the gobang model iteration, the winner or loser judgment, the forbidden rule judgment and so on.

4) BOARD RECOGNITION

In this case, the board ranks are evenly divided into 9 regions to keep the overall pose of the Mitsubishi and Yaskawa manipulators in each region changed within a small degree. The vision subsystem can read the stone positions on the board, and then transform them into the row and column coordinates. Two main equations are used in this control process. First, the endpoint of the Mitsubishi and Yaskawa manipulators should maintain a linear spatial relationship with the board, which can be represented by (3).

$$\begin{cases} x_m = (m-a)*k_0 + b_0 \\ y_m = (n-b)*k_1 + b_1 \\ z = c \end{cases} \tag{3}$$

Where x_m, y_m, z_m is the coordinate of the manipulator endpoint, m, n are the row and column positions of the board, respectively, a, b represent the start row and column number corresponding to each region of the board, k_0, k_1, b_0, b_1, c are constant.

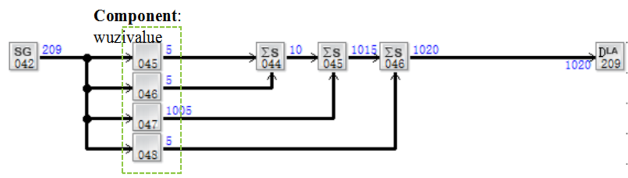


FIGURE 12. The logic of valuation calculation.

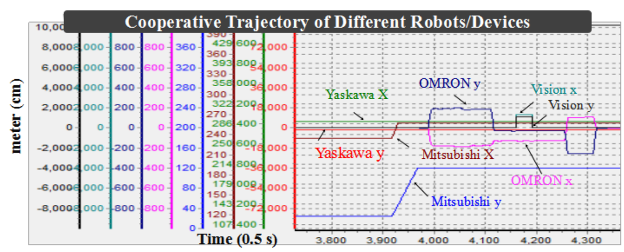


FIGURE 13. The cooperative trajectory of different robots/devices.

Second, (4) can be used to calculate the coordinates of the OMRON robot endpoint.

$$\begin{cases} x_0 = k_2(m-a_1) + b_2 \\ y_0 = k_3(n-a_2) + b_3 \end{cases} \tag{4}$$

Where x_0, y_0 is the coordinate of the robotic endpoint, m, n are the row and column positions of the board, $k_2, k_3, b_2, b_3, a_1, a_2$ are constant.

The results are shown in Fig. 11-14. In these figures, the process of one-interactive play is detailed, including the

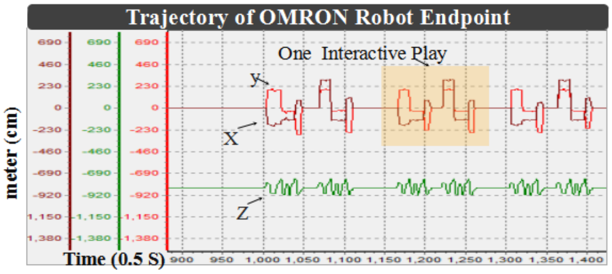


FIGURE 14. **The trajectory of OMRON robotic endpoint.**

field control process, the HMI interface as well as the motion trajectory of the robots and auxiliary devices. From the HMI and the field monitoring figures (Fig. 11), we can see that man (black) is blocked twice by AI (white) in double-four forks, and the OMRON robot can assist human and the artificial intelligence to operate the field stones accurately. From the time cost data (Fig. 11), we can find that compared with the robotic motion control, the image recognition and AI judgment process can be completed in the cloud servers within a shorter and acceptable time.

In this case, we use approximately 130,000 configuration components in total to achieve the system major function, including the AI algorithm, the board coordinate transformation, the robotic motion control, etc. Because the cloud server undertakes the computations of complex tasks such as the intelligent algorithm, the number of the components processed in the cloud accounts for 74%, and the calculations of the remaining components are performed by the edge controller IAPbox. Fig. 12 shows an example of the valuation algorithm application procedure for the gobang game. It is implemented by using the specific configuration component WuziValue. In the figure, SG is the target position of the stone, and the whole procedure uses four WuziValue components to respectively calculate the scores for four directions, including the upper, the lower, the left and the right. The total of all these scores is used for the final evaluation of the target position. As shown in Fig. 13, we can find that during the experiment, the XYZ coordinates of OMRON robot change in real time and have certain regularity. These changes correspond to different motions such as the robotic movement, sucking up a stone, interval-stopping motion and placing the stone. Fig. 14 shows the dynamic relation between the endpoint trajectory of all the robots and the real-time coordinates of the target position.

From this case, we can find that the man, the cloud and the robots or devices are highly coordinated in IAPcloud control system, and all of them can not only comply with the same rules but also respond to the control requirements caused by various random changes.

Case 3: The Dynamic Reconfiguration of the Robotic Control Algorithm In order to verify the feasibility of applying the dynamic reconfiguration technology to cloud robotic controls, this case takes the Mitsubishi manipulator and its driver as the test subjects and uses cloud virtual controllers to test and observe the reconfigurable behavior and state of robotic algorithms. When the experiment starts, we make the Mitsubishi robot perform the initial control function firstly, then modify and test the control configuration program to adjust the robotic control mode, and send the procedure to the virtual controllers in the cloud. After analyzing and calculating the new algorithm, the virtual controller sends control data to IAPbox, and then drives the manipulator to perform the new control action and maintain the current motion state. Due to the inherent characteristics of the configuration components and the mechanism of virtual controller, the algorithm reconfiguration does not address recompilation issues, and it can be completed without stopping the robot running.

In this case, the robotic state can be classified into four types, which can be described as (5).

$$f(t) = \begin{cases} 0, & closing \\ 1, & opening \\ f_1(t), & worked \\ f_2(t), & s\tan dby \end{cases} \tag{5}$$

The dynamic reconfiguration requires that the execution of the control configuration must be continuous and consistent when the original algorithm gets replaced by the new algorithm [35]. After reconfiguration, the new algorithm $g(t)$ can generate the corresponding control effect C when the specific external condition D_k is satisfied, which can be described as (6).

$$\begin{cases} C(state) = g(t) \cdot f(t) \\ D_k \begin{cases} |t_i - t_{i-1}| \leq \delta \\ |t_{i+1} - t_i| \leq \delta \end{cases} \end{cases} \tag{6}$$

Fig. 15 shows the results of this case. From the motion trend of the manipulator, it can be seen that in the initial state, the Mitsubishi robot keeps the x coordinates of the manipulator endpoint unchanged and makes a periodic up and down movement with the amplitude of ±5dm along the y axis. At 3.2 seconds, the robotic endpoint performs the periodic up and down movement with the amplitude of +3dm and −5dm along the y axis, after online adjusts the amplitude (plus y direction) to +3dm by modifying the component parameter, and then keep the state. At 3.4s, the manipulator endpoint moves follow a sine wave pattern with the amplitude of 2dm and the period of 125ms, after we online modify the robotic control mode again. From the figure, we can find that when the algorithm is modified using the dynamically reconfiguration method, the control system can timely and accurately adjust the trajectory of the manipulator endpoint and keep the motion state continuous.

VII. EVALUATION AND DISCUSSION

Compared with other existing cloud robotic framework, IAPcloud is a novel control platform for cloud robots, in our view, it may become the future development trend of the

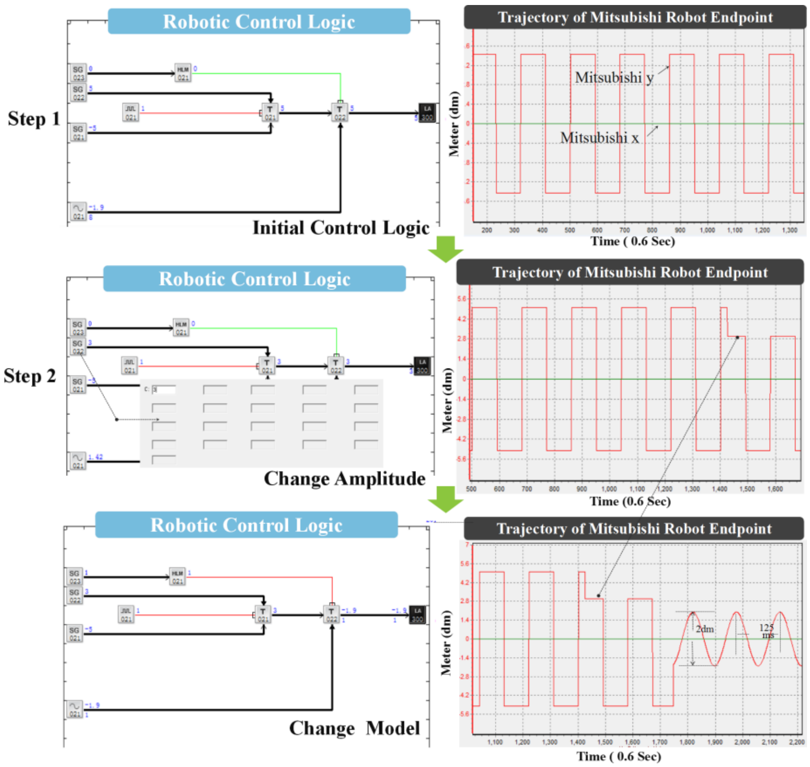


FIGURE 15. Dynamic reconfiguration of the Mitsubishi manipulator control algorithm.

cloud robotic control technology for its significant advantages in many aspects.

First, IAPcloud improves the deployment efficiency of the cloud platform. It establishes a consistent computing environment for the robotic control algorithm in both the cloud and the edge system to avoid tedious operations such as code transformations. In other words, this platform can migrate the complex robotic control tasks from the edge controller to the cloud virtual controller (including the third-party cloud platform that deploys virtual controllers) using a unified technical standard, which effectively reduces the difficulty of the cloud platform application.

Second, IAPcloud improves the development efficiency of the robot applications. In IAPcloud, a unified control configuration method is applied to realize the collaborative control of heterogeneous cloud robotics, which has significant characteristics, such as graphical configuration, visual calculation and dynamically reconfiguration. Compared with other off-line programming methods based on high-level languages, this approach has the advantages of simple programming, and support to track the robotic control process by the human-computer interaction tools. In addition, it also can well adapt to the change of the flexible production process through the online optimization approach, so as to effectively reduce the development and deployment cost of cloud robotic applications.

Third, the edge gateway devices can integrate a large number of heterogeneous robots and devices, which not only improves the interoperability between heterogeneous systems, but also creates advantageous conditions to form a vendor-independent resource pool of robots/devices, and achieve multi-robot collaborative control and the platform-based services. Although our research cases focus on several types of manipulators and auxiliary devices, it should be noted that the way virtual controllers tackle the heterogeneity of devices and their collaborative control is not limited to these robots or devices. In IAPcloud, the development of different cloud robotic systems can be implemented by the same technical standards and methods, including the device integration, the system control, the cloud service invocation and so on. The main difference is the resource usage of the cloud platform, such as the number of the requested virtual controllers, the type and number of components, and so on. Table 1 lists several technical indicators of IAPcloud. As shown in the Table, we can see that IAPcloud provides more than 110 reusable components for different robotic control applications. Each virtual controller can process the computing tasks of 16,000 components at the same time and supports the communication for at least 300 analog components and 999 digital components. The communication cycle of each virtual controller is not more than 1s. During the running period of virtual controllers, the robotic algorithms can be reconfigured online. And different cloud robotic control applications can flexibly invoke these service resources according to their task requirements.

Finally, IAPcloud supports for complex control tasks. The cloud platform of IAPcloud can deploy numerous virtual controllers. It supports the complex computation of various

TABLE 1. **IAPcloud technical indicators.**

Indicator type	IAPcloud platform
Number of components	<= 16000 / controller
Type of components	> 110
Communication cycle of virtual controllers	< 1s
Communication data between virtual controllers	> = 300 analog outputs, 999 digital outputs
Support dynamic reconfiguration of robotic algorithms	Yes
Support different network protocols	Yes, such as FINS, RS232 / RS485 and other third-party network protocols

components and has good stability, which can well meet the control needs of the complex industrial manufacturing.

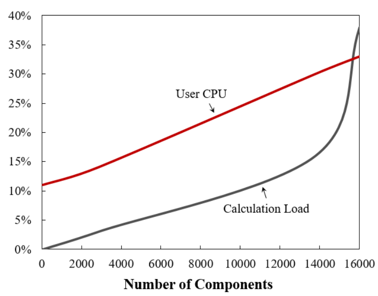


FIGURE 16. **The relation between the computing resource utilization and the computing load of the cloud platform.**

The various combinations of these components can not only form the kinematic control algorithm of manipulators and the search algorithm of the gobang game mentioned above, but also implement some new intelligent algorithms like deep learning and big data processing technology. In IAPcloud, all complex algorithms functions are mapped into the computing units with the multiple inputs and single output feature for calculation, so as to realize the integration and application of the intelligent algorithm in IAPcloud. Fig. 16 shows the relation between the computing resource utilization of the cloud platform and the computing load. From the figure, we can find that though the system computing load increases with the increase of components when the virtual controller processes control logics, the overall growth is stable (when the number of configuration components doesn't exceed 14000), and the CPU occupancy rate can be controlled below 50%. The computing load is calculated by using the following formula:

$$Q = \frac{T_s}{T_c} \times 100\% \tag{7}$$

Where Q is the computing load, T_s is the scan period of the virtual controller, and T_c is the pre-set calculation period. To further verify the stability of the IAPcloud platform, we test the system average load in the case of 50 virtual controllers simultaneously process the complex calculation of 16,000 components. As shown in Fig. 17, the result indicates that the system can keep a good stability when the virtual controllers deal with complex control tasks.

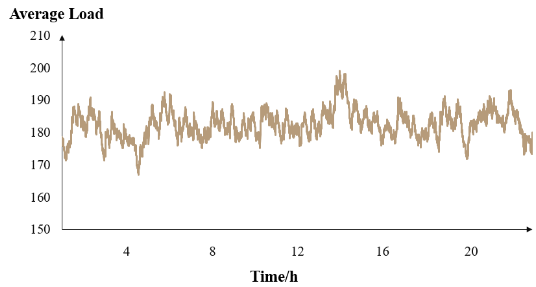


FIGURE 17. **The stability test of the cloud platform.**

VIII. CONCLUSION AND FUTURE WORK

The method and technology of cloud robots are feasible in engineering, and it can also promote the traditional network control technology in theory. However, the lack of unified technical standards that are vendor-independent and platform-agnostic to improve the interoperability among heterogeneous systems is a critical issue in the current cloud robotic research. It is also one of the most important reasons why the existing cloud robotic platforms can't be well used in practice or engineering. Fortunately, the distance between the current cloud robotic technical status and its engineering applications can be greatly shortened by IAPcloud, which is presented as a novel control platform of cloud robots and may become their future development trend in our view. Compared with other existing cloud robotic framework, this platform shows significant advantages in many aspects, such as establishing the technical standard of robotic cloud controls, integrating heterogeneous multiple robots or devices, and implementing the collaborative control in a graphical, visual, and reconfigurable way in order to adapt to the flexible manufacturing needs. The main contribution of this paper is to propose the IAPcloud architecture and its detailed design. The results indicate that it is scientific and feasible to apply the virtual controller technology to solve the collaborative control problem among heterogeneous cloud robots or devices.

In the future, applying platform and service approaches to the control system is the development trend of cloud robots. IAPcloud will face new challenges. Attacks may occur at different levels of the cloud robot system, either at the physical level or the clouds, therefore the information security risk of the industrial production process must be addressed more carefully [40]. Moreover, there is another issue worth studying continuously, that is, how to design the granularity of components to meet the reused need of the robotic control algorithms, while improving the efficiency of the control configuration. In the near future, the platform-based service mode of IAPcloud will be further enhanced, including improving the runtime efficiency of the virtual controller and the resource consumption, promoting the quality of the cloud platform service, designing a flexible collaboration approach and making full use of the robotic resources to tackle more challenging tasks from cloud robotic users.

REFERENCES

[1] V. Nagrath *et al.*, "Dynamic electronic institutions in agent oriented cloud robotic systems," *Springerplus*, vol. 4, no. 1, p. 103, Dec. 2015.

[2] J. M. Quintas, "Cloud robotics: Towards context aware robotic networks," in *Proc. Int. Conf. Robot.*, Pittsburgh, PA, USA, 2011, pp. 87–101.

[3] I. Osunmakinde and R. Vikash, "Development of a survivable cloud multi-robot framework for heterogeneous environments," in *Proc. Int. J. Adv. Robot. Syst.*, vol. 11, no. 10, pp. 1657–1667, Jul. 2014.

[4] W. Li, C. Zhu, L. T. Yang, L. Shu, E. C.-H. Ngai, and Y. Ma, "Subtask scheduling for distributed robots in cloud manufacturing," *IEEE Syst. J.*, vol. 11, no. 2, pp. 941–950, Jun. 2017.

[5] R. M. A. Mateo, "Scalable adaptive group communication for collaboration framework of cloud-enabled robots," in *Proc. 17th Int. Conf. Knowl. Based Intell. Inf. Eng. Syst. (KES)*, Sep. 2013, vol. 22, no. 22, pp. 1239–1248.

[6] B. Kehoe, A. Matsukawa, S. Candido, J. Kuffner, and K. Goldberg, "Cloud-based robot grasping with the Google object recognition engine," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, Karlsruhe, Germany, 2013, pp. 4263–4270.

[7] J. Kuffner, "Cloud-enabled humanoid robots," in *Proc. IEEE-RAS Int. Conf. Humanoid Robot.*, Nashville, TN, USA, Dec. 2010, pp. 176–181.

[8] M. Waibel *et al.*, "RoboEarth," *IEEE Robot. Autom. Mag.*, vol. 18, no. 2, pp. 69–82, Jun. 2011.

[9] R. Arumugam *et al.*, "DAvinCi: A cloud computing framework for service robots," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, Anchorage, AK, USA, 2010, pp. 3084–3089.

[10] R. Doriya, P. Chakraborty, and G. C. Nandi, "'Robot-cloud': A framework to assist heterogeneous low cost robots," in *Proc. Int. Conf. Commun. Inf. Comput. Technol. (ICC)*, Mumbai, India, 2012, pp. 1–5.

[11] L. Riazuelo, J. Civera, and J. M. M. Montiel, "C^2TAM: A Cloud framework for cooperative tracking and mapping," *Robot. Auton. Syst.*, vol. 62, no. 4, pp. 401–413, Jan. 2014.

[12] D. Hunziker, M. Gajamohan, M. Waibel, and R. D'Andrea, "Rapyuta: The RoboEarth cloud engine," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, Karlsruhe, Germany, 2013, pp. 438–444.

[13] E. Guizzo, "Robots with their heads in the clouds," *IEEE Spectr.*, vol. 48, no. 3, pp. 16–18, Mar. 2011.

[14] R. Chaari *et al.*, "Cyber-physical systems clouds: A survey," *Comput. Netw.*, vol. 108, pp. 260–278, Oct. 2016.

[15] L. Zhao *et al.*, "Research on realizing robot cloud-operation platform," *J. Huazhong Univ. Sci. Technol. (Natural Sci. Ed.)*, vol. 74, no. s1, pp. 161–164, Dec. 2012.

[16] Z. Du, L. He, Y. Chen, Y. Xiao, P. Gao, and T. Wang, "Robot Cloud: Bridging the power of robotics and cloud computing," *Future Gener. Comput. Syst.*, vol. 74, no. 4, pp. 337–348, Sep. 2017.

[17] B. Kehoe, S. Patil, P. Abbeel, and K. Goldberg, "A survey of research on cloud robotics and automation," *IEEE Trans. Autom. Sci. Eng.*, vol. 12, no. 2, pp. 398–409, Apr. 2015.

[18] S. Ahmed, A. Topalov, and N. Shakev, "A robotized wireless sensor network based on MQTT cloud computing," in *Proc. IEEE ECMSM*, Donostia-San Sebastian, Spain, May 2017, pp. 1–6.

[19] S. A. Miratabzadeh *et al.*, "Cloud robotics: A software architecture: For heterogeneous large-scale autonomous robots," in *Proc. World Autom. Congr. (WAC)*, Rio Grande, Puerto Rico, 2016, pp. 1–6.

[20] B. V. S. Krishna, J. Oviya, S. Gowri, and M. Varshini, "Cloud robotics in industry using Raspberry Pi," in *Proc. Int. Conf. Sci. Technol. Eng. Manage. (ICONSTEM)*, Chennai, India, 2016, pp. 543–547.

[21] G. Hu, W. P. Tay, and Y. Wen, "Cloud robotics: Architecture, challenges and applications," *IEEE Netw.*, vol. 26, no. 3, pp. 21–28, May 2012.

[22] Y. Duan and X. Yu, "Multi-robot system based on cloud platform," in *Proc. CGNCC*, Nanjing, China, 2017, pp. 614–617.

[23] R. Wirz, R. Marin, and P. J. Sanz, "Remote programming over multiple heterogeneous robots: A case study on distributed multirobot architecture," *Ind. Robot, Int. J.*, vol. 33, no. 6, pp. 431–442, Nov. 2006.

[24] J. X. Zhang and Y. J. Wu, "Cloud collaborative computing framework for a service robot based on ROS," *Comput. Syst. Appl.*, vol. 25, no. 9, pp. 85–91, Feb. 2016.

[25] H. E. Schaffer, "X as a service, cloud computing, and the need for good judgment," *IEEE IT Prof.*, vol. 11, no. 5, pp. 4–5, Sep./Oct. 2009.

[26] M. Puleri, R. Sabella, and A. Osseiran, "Cloud robotics: 5G paves the way for mass-market automation," *Ericsson Tech. Rev.*, vol. 93, no. 6, pp. 2–13, Jun. 2016.

[27] Y. Chen, Z. Du, and M. García-Acosta, "Robot as a service in cloud computing," in *Proc. 5th IEEE Int. Symp. Service Oriented Syst.*, Nanjing, China, Jun. 2010, pp. 151–158.

[28] L. A. Grieco *et al.*, "IoT-aided robotics applications: Technological implications, target domains and open issues," *Comput. Commun.*, vol. 54, pp. 32–47, Dec. 2014.

[29] P. P. Ray, "Internet of robotic things: Concept, technologies, and challenges," *IEEE Access*, vol. 4, pp. 9489–9500, Jan. 2017.

[30] S. Zheng, Q. Zhang, R. Zheng, B.-Q. Huang, Y.-L. Song, and X.-C. Chen, "Combining a multi-agent system and communication middleware for smart home control: A universal control platform architecture," *Sensors*, vol. 17, no. 9, pp. 2135–2158, Sep. 2017.

[31] Z. Lu and S. Zheng, "Inverse kinematics algorithm of 6-DOF robots based on geometric method and Screw theory," *J. Mech. Transmiss.*, vol. 17, no. 9, pp. 111–114, Jun. 2017.

[32] Z. Song, Z. Wang, L. Chaoru, and G. Caimeng, "Research on the configuration method of mobile robot and its realization," in *Proc. IEEE Chin. Control Decis. Conf. (CCDC)*, Guiyang, China, May 2013, pp. 2877–2883.

[33] S. Zheng, "Application of platform integration control technology in ship," in *Proc. Chin. Conf. Commun. Control*, Beijing, China, May 2014, pp. 77–82.

[34] S. Zheng, Y. Pan, and L. Wang, "Application of data engine technology to complex control systems," *Elect. Age*, vol. 6, no. 1, pp. 20–21, Aug. 2005.

[35] S. Zheng and W. Ni, "Research and implementation of dynamic reconfiguration technology in distributed control system," *Energy Sci. Technol.*, vol. 43, no. 1, pp. 724–729, Oct. 2009.

[36] S. Du, Z. Hou, D. Ge, and C. Wang, "Research on virtual controller," *Metall. Ind. Autom.*, vol. 28, no. s1, pp. 972–974, Apr. 2004.

[37] D. E. Knuth and R. W. Moore, "An analysis of alpha-beta pruning," *Artif. Intell.*, vol. 6, no. 4, pp. 293–326, Dec. 1975.

[38] E. C. Shannon, "Programming a computer for playing chess," *Philos. Mag.*, vol. 41, no. 314, pp. 256–275, Mar. 1950.

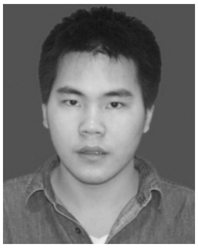
[39] D. Zhang and G. Lu, "Shape-based image retrieval using generic Fourier descriptor," *Signal Process., Image Commun.*, vol. 17, no. 10, pp. 825–848, Nov. 2002.

[40] K. Ren, C. Wang, and Q. Wang, "Security challenges for the public cloud," *IEEE Internet Comput.*, vol. 16, no. 1, pp. 69–73, Jan. 2012.

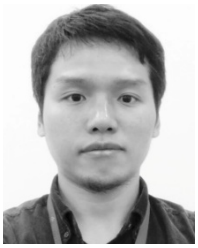
[41] S. Zheng and W. Ni, "General control station based on data engine technology," *Elect. Age*, vol. 6, no. 9, pp. 122–125, 2008.

SONG ZHENG received the Ph.D. degree from the Department of Thermal Engineering and Thermal Physics, Tsinghua University. He is currently a Researcher with the Institute of Electrical Engineering and Automation, Fuzhou University, and the Chairman of the Fuzhou University Advanced Control Technology Research Center. He is also the Chairman of the Key Laboratory of Industrial Automation Control Technology and Information Processing, Fuzhou University, the Dean of Histron Research, and the President of the Fujian Association of Automation.

He has authored or co-authored over 80 papers and holds patents in the research areas of industrial automation. His research interests include Industrial Internet, complex system, Internet of Things, cloud control, heterogeneous system, and industrial information security. He is the High-Level Entrepreneurial Innovation Introduced Talent of Fujian Province and the Entrepreneurial Innovation Talent of Fujian 100-Talent Program. He received the second prize of the Fujian Scientific and Technological Advancement Award in 2012 and the second prize of Fuzhou Scientific and Technological Advancement Award in 2012 and 2013, respectively.



ZHICHENG LIN received the B.Eng. degree in electrical engineering and automation from Jinan University in 2016. He is currently pursuing the master's degree in pattern recognition and intelligent system with Fuzhou University. His current research interests include new configuration technology of robot programming, image recognition and processing, and complex system control.



QIJUN ZENG received the B.S. degree in aircraft design and engineering and the M.S. degree in aircraft design from the Harbin Institute of Technology, Harbin, China, in 2007 and 2009, respectively. He is currently the Director of the Intelligent Manufacturing Laboratory, Research Institute of Fujian Histron Group Co., Ltd., and IAP(Fujian) Technology Co., Ltd. His research interests include general technology of industrial automation and human–computer interaction.

RONG ZHENG received the B.S. degree in bioengineering and the M.S. degree in biochemistry and molecular biology from Fuzhou University, Fuzhou, China, in 2009 and 2012, respectively. She is currently an Assistant Researcher with Research Institute of Fujian Histron Group Co., Ltd., and IAP(Fujian) Technology Co., Ltd. Her research interests include smart life, robot, complex systems, and bioinformatics.

CHAORU LIU received the B.Eng. degree in software engineering from Fuzhou University in 2007. He is currently the Head Engineer with the Research Institute of Fujian Histron group Co., Ltd., and IAP(Fujian) Technology Co., Ltd. His current research interests include complex network communication technology and control platform software.

HUAFENG XIONG received the B.Eng. degree in electronic information engineering from Hangzhou Normal University in 2013. He is currently an Engineer with the Network Information Laboratory, Research Institute of Fujian Histron Group Co., Ltd., and IAP(Fujian) Technology Co., Ltd. His research interests include cloud computing, cloud control platform, and big data technology.