Research on Part Grabbing System of Manipulator under Cloud Environment

Jihong Yan*, Ph.D.
Professor in Industrial Engineering
Deputy Dean of School of Mechatronics Engineering
Head of intelligent Manufacturing Scientific Research Team
Harbin Institute of Technology
Harbin, China
jyan@hit.edu.cn

Tianyi Jing, Senior student
School of Industrial Engineering
Harbin Institute of Technology
Harbin, China
18800415536@163.com

Abstract

The manipulator grasping system based on the local visual system is only be applicable to the static and already existing in the visual training model parts and can't accurately identify and grasp new or motional parts. In this paper, a part grasping system for manipulator in cloud environment is designed. By using the excellent storage performance and efficient computing capacity of cloud, the types of parts that can be grabbed by manipulator are enriched, the calculation speed of part data is improved, and the dynamic grabbing is realized. Under the structure of cloud-edge-terminal, using the part type, running speed, height, spatial position and angle information stored in the cloud, combined with the robot running speed and Rodriguez formula, the manipulator TCP position and pose of the part grabbing point is calculated. The capture strategy of parts is generated in the cloud. By the network connection between the upper computer and the manipulator, the control command generated online is transmitted to the manipulator to complete the dynamic grasping spare parts. At the same time, the status data of the manipulator is obtained and analyzed during the grasping process. The method proposed in this paper improves the adaptability of the part grabbing system, and provides a technical basis for motional part grabbing in the cloud environment.

Keywords
Industrial robots, Cloud environment, Capture strategy, Robot condition monitoring.

1. Introduction

As the rapid development of robot technology, many kinds of industrial robots are used to replace or help people to complete special work, such as welding robot and assembly robot, which greatly improve the efficiency, save the human resources and ensure the safety of worker. Industrial robots have been widely used in machining, logistics, product manufacturing and other industries, playing an important role in them. Manipulator is a cardinal part of industrial robot, whose grasping system is the symbol to measure the intelligent level of the robot system. A well designed grasping system can effectively meet the needs of the manufacturer in the production process, and save a lot of manpower, material resources and time cost.

The manipulator used in this paper is the 6-DoF UR5 robot, and its overall structure is shown in Fig. 1. The kinematic and dynamic modelling of UR5 robot is already established [1].

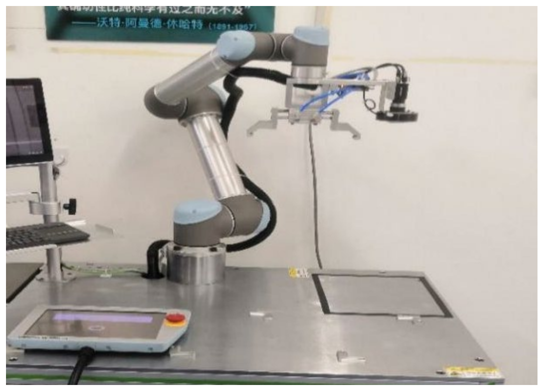

Figure 1. UR5 robot

1.1 Main Problems

At present, most of grasping systems are designed based on local machine vision. Firstly, the parts are scanned, followed by the parts properties analyzing, and then calculating the grasping points. Finally, the manipulator is controlled to grasp the parts. There is also a semi model method, which does not need to recognize the parts to be grasped completely, but only needs to collect the edge information of the parts. However, this method requires a large number of similar parts to train the algorithm, so it need a long design cycle. And it is difficult to grasp new parts effectively. In addition, there is a free model method, the appropriate grab points are directly calculated from the image of the parts, which avoids a lot of machine training. But the disadvantage of this method is the recognition speed is slow, because a new recognition calculation is needed for each part captured [2-4]. As mentioned above, the grasping system based on local machine vision is hard to complete the grasping task of the new parts that aren't included in the model. In addition, the local computing speed is limited, so it is difficult to grasp the moving parts accurately.

In order to realize the sharing of production resources by industry 4.0, the cloud manufacturing platform is rapidly developed, and many industrial production data and the intelligent algorithms are added into the platform, which makes it convenient for other users to obtain the parts information, such as dimensions, weight, and the parts grasping strategy [5]. If the parts library and the rapid identification of cloud with strong storage and computing ability, are used to design the grasping system of manipulator, the system efficiency and adaptability to new parts will be effectively improved. Many researchers have studied the cloud-edge-terminal structure. Pontes et al. [6] present a microservice architecture for DNN-based distributed MEP over heterogeneous cloud-edge environments. Wang [7] reduce the power consumption of the device and the server by IGA algorithm.

1.2 Objectives

This paper adopts the structure of cloud-edge-terminal, using PC as the edge node and combining the grasping system of manipulator under the cloud environment. The whole design is divided into four parts.

a. Establishment and reading of cloud database. The cloud database is established, which contains the data needed by the manipulator to complete the grasping task, including: part location, speed of part, part height, angle of part and so on. By obtaining the parts data contained in the cloud in real time, it provides conditions for transferring the part grasping strategy to the control commands of manipulator.

b. Realization of the remote control of manipulator. The remote control of manipulator is realized by network connection. PC is used as edge node to connect the cloud database and manipulator, so as to ensure the feasibility of the structure of cloud-edge-terminal.

c. Design of part grasping strategy. In order to achieve the dynamic grasping of parts and ensure the accuracy of that, it is necessary to design the part grasping strategy for the manipulator, which needs to combine the type of parts, current position, speed of part, height, placement angle and other information of part to calculate the tool center position (TCP) of manipulator. And the TCP pose of manipulator is also needed to be calculated according to the information of the placement angle of the part.

d. Robot status monitoring. During the robot movement, the status data of the robot is monitored and uploaded to cloud database.

2. System Structure

This paper adopts the structure of cloud-edge-terminal, which integrates the cloud platform, edge computing and terminal control into a unified and coherent system. Under this structure, the cloud platform stores a large amount of parts information, and has various types of cloud computing services, such as big data processing, artificial intelligence algorithm, etc. Moreover, due to its strong hardware support and distributed computing, the cloud platform has very efficient data computing and information processing capabilities. The emergence and development of cloud realizes real-time data sharing, and balances the computing load of edge and terminal [8]. The edge is the computing processing layer closest to the terminal. Although the network transmission speed is developed, there is still the delay between the cloud and terminal because of the huge amount of data stored in the cloud, so the edge computing is needed to finish the data processing. By deploying computing resources to the edge, the response speed of the system is improved, which realizes the real-time data sharing and transmission between cloud and terminal, and reduces the pressure of terminal computing [9,10]. The terminal is the executive mechanism to complete the task in the industrial field, including manipulator, AGV, production line, etc. With the development of industrial technology and basic theory, the computing power and data processing capacity of the terminal have been enhanced correspondingly. Relying on the structure of cloud-edge-terminal to complete collaborative computing and data processing will greatly improve the execution ability of the terminal.

In this structure, the manipulator relies on all kinds of industrial data transmission protocols to connect with the edge and transmit the information to the edge in real time. After data processing is completed in the edge, it is uploaded to the cloud. The cloud is equipped with the cloud manufacturing platform database, which stores a number of manufacturing information, including the related information of the manipulator and various parts. In addition, there are intelligent algorithms such as machine vision, big data processing, deep learning and device diagnosis. According to the actual situation, many kinds of algorithms are used to calculate data in the cloud. By accessing the interface of cloud manufacturing platform, the edge can obtain the data results generated by cloud computing, process the data results and transmit them to the robot through network transmission protocol to complete the field task. The cooperation among cloud, edge and terminal breaks the limitation of traditional terminal computing power and directly improves the intelligent level of robot. The structure of cloud-edge-terminal is shown in Fig. 2.

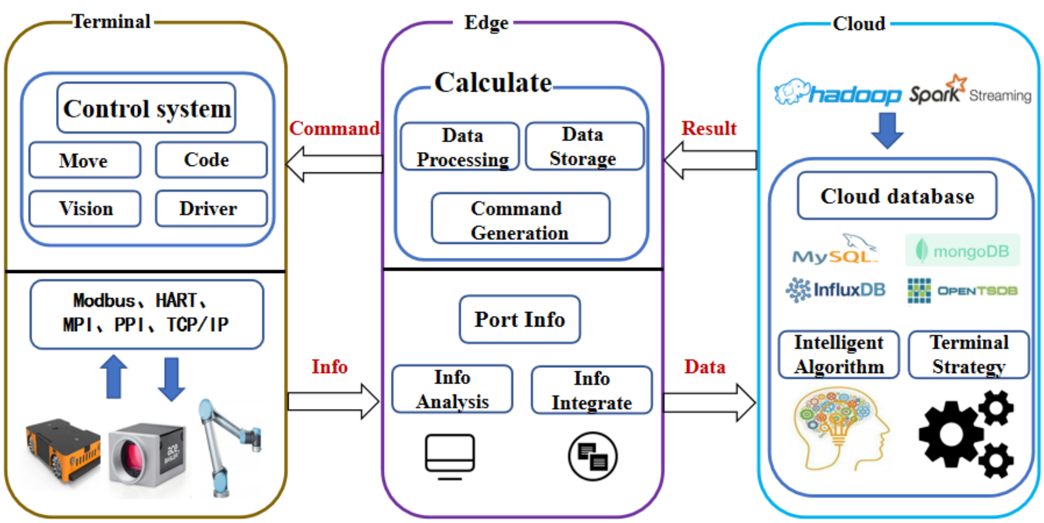

Figure 2. Structure of cloud-edge-terminal

This paper uses TCP/IP network protocol to build information transmission channel. The cloud manufacturing environment database is established in the cloud to store the relevant information of parts and UR5 status. According to the information of the part, the grasping strategy of the part is generated in the cloud and transmitted to the edge by Java and cloud database communication technology, and it is converted into UR5 control commands online. Through the network connection between the upper computer and UR5 robot, the control commands are transmitted to the manipulator control system in real time to realize the dynamic grasping of the parts, and the running state of the manipulator is monitored during the working period. The process is shown in Fig. 3.

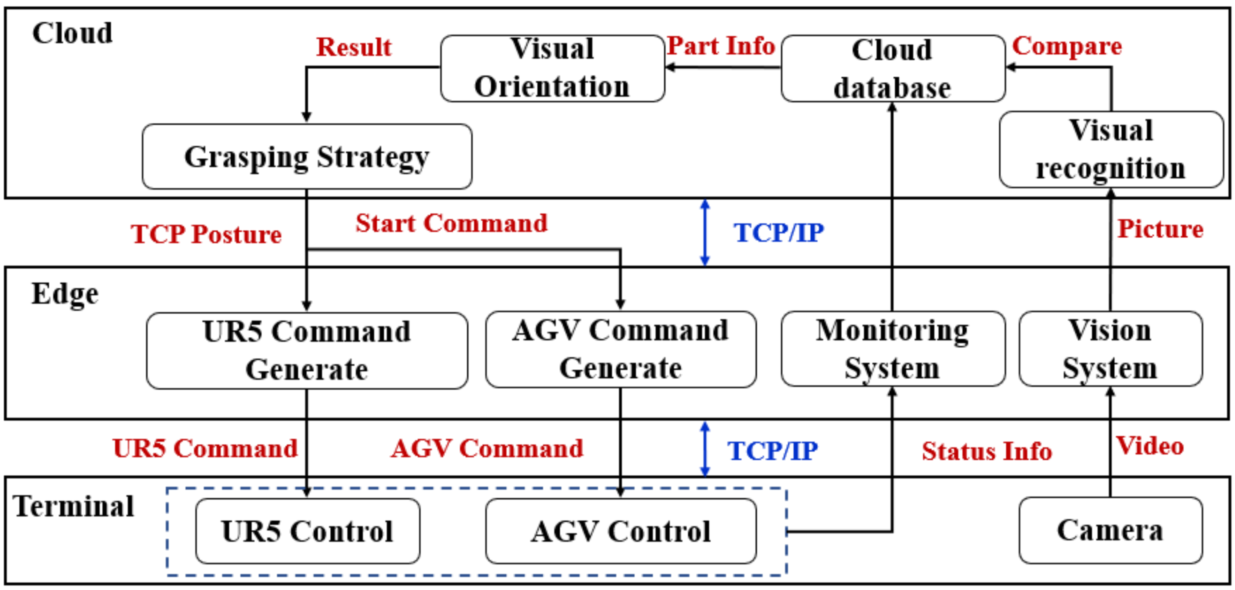

Figure 3. System Process

3. Methods

3.1 Grasping Strategy

The part grasping strategy of manipulator is the robot TCP coordinates and trajectory of the part grasping point. The part grasping strategy design includes two aspects: calculating the TCP spatial position and TCP pose when the part moves to the grasping point. Firstly, the initial TCP coordinates of UR5 robot are set according to the actual situation. The initial position of the robot [*x, y, z, Rx, Ry, Rz*] = [400mm, 645mm, 50mm, 3.092rad, -0.553rad, -0.04rad] is set by using UR5 robot teach pendant.

The position calculation of TCP includes the x-axis and z-axis position when the manipulator grasps part, as well as the movement time and waiting time in the process of operation. This design uses the part position prediction method to calculate the TCP position. The braking time of the robot is very short, which can be ignored for the calculation of the overall TCP spatial position.

Height of TCP is calculated by,

$$z_f = z_0 + z_1 + z_2 + z_3 - z' \tag{1}$$

where:

z_f : Height of TCP(m).

z_0 : Height of part(m).

z_1 : Height of AGV(m).

z_2 : Distance between TCP and the gripper(m).

z_3 : Distance between gripper and the bottom of part(m).

z' : Distance between UR5 base and ground(m).

The x-axis position of TCP is calculated by,

$$x_1 = x_0 + L + v_1\left(\frac{v_2}{a_1} + t_1 + \frac{z_f}{v_3}\right) \tag{2}$$

where:

x_1 : TCP x-axis position of grasping point (m).

x_0 : Distance between part initial position and UR5 base(m).

L : Distance between initial position of TCP and x_0 (m).

v_1 : Speed of part(m/s).

v_2 : Speed of TCP(m/s).

a_1 : The first acceleration of TCP(m/s^2).

t_1 : The uniform motion time of TCP(s).

z_f : Grasping point height of TCP(m).

v_3 : Speed of TCP downward movement(m/s).

Suppose that the manipulator gripper is closed to complete the grasping task after time,

$$t_2 = \frac{v_3(v_2^2 - 2v_1v_2) + 2a_1[(v_2 - v_1)v_3t_1 - v_1z_f]}{2a_1v_1v_3} \tag{3}$$

The manipulator TCP position calculation is completed.

When the grasping axis of the part is at a certain angle with the initial closing axis of the manipulator gripper, the gripper needs to be rotated at the angle to complete the grasping of the part. The rotation angle is the acute angle between the grasping axis of the part and the initial closing axis of the gripper. When the angle of the part is clockwise around the positive direction of the *z* axis, the symbol is positive.

The TCP pose control of UR5 robot is represented by rotation vector method, which is difficult to directly calculate, so Rodriguez formula (formula 3) is used to transform rotation vector and rotation matrix,

$$R = I + \sin(\theta)K + [1 - \cos(\theta)]K^2 \tag{4}$$

where:

R : Rotation matrix.

I : Identity matrix.

θ : Rotation angle.

K : Unit vector of rotation axis.

The unit vector is set as $K = [kx, ky, kz]$, and the initial rotation vector is set as $T = [Tx, Ty, Tz]$. And then the θ and K can be express as,

$$\theta = \sqrt{Tx^2 + Ty^2 + Tz^2} \tag{5}$$

$$kx = \frac{Tx}{\theta}, ky = \frac{Ty}{\theta}, kz = \frac{Tz}{\theta} \tag{6}$$

After the rotation vector is transformed into a rotation matrix, the matrix is multiplied by the angle rotation matrix (formula 6) to obtain a new rotation matrix, which is then transformed into a rotation vector,

$$R(z,\psi) = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{7}$$

where ψ is the angle of part.

Finally, the target rotation vector is calculated by the inverse solution of Eq. 4.

3.2 Establish and Operation of Cloud Database

Firstly, the parts data storage table and UR5 status information data storage table in MySQL are created to establishing cloud manufacturing environment. The data table content and structure are shown in Table 1 and Table 2.

Table 1. Parts data storage table

Name	Storage Format
Type	Varchar (20)
Height	Double
Position	Varchar (30)
Angle	Double
Speed	Double
ID	Int
Size	Varchar (30)
Time	Timestamp

Table 2. UR5 status data storage table

Name	Storage Format
ID	Int
Time	Timestamp
UR5_Angle	String
A	Double
V	Double
Speed	Double
Temperatures	Double
Force	Double

The distributed MySQL database is used in this paper. After the data tables created, the JDBC driver of java is used to establish the connection between the manipulator upper computer and the cloud database. During the writing and reading of database, the control of thread frequency is needed to ensure the real-time requirements of the system. The procedure is shown in Table 3.

Table 3. Cloud database connection, writing and reading

Database Connection & Writing & Reading
Input: Database Name, Password, URL, vision information **Output**: part grasping strategy
Set database parameters (name, password, URL, data format, etc.)
Get Connection (name, password, URL, JDBC)
Define writing (Connection, UR5 status information, 20Hz) function
Get UR5 information
SQL: INSERT INTO
Calculate the intermediate time of each save
END the writing
Define read (60 data per second)
Get part grasping strategy
Return part grasping strategy

3.3 Connection Between UR5 and Upper Computer

This paper adopts TCP / IP transmission protocol, which can provide high data transmission speed, to realize the connection between the upper computer and UR5. Firstly, the upper computer and UR5 robot need to be connected to the same LAN, with UR5 robot as the server and PC as the client. The PC is connected to the UR5 robot network by using the socket interface provided by TCP / IP protocol. Static IP address of UR5 is set to further realize control and status monitoring of UR5, using the functions of write and flush in socket to complete the transmission of UR5 control commands. And while loop is used to get UR5 status data. The procedure is shown in Table 4.

Table 4. UR5 control and parsing of status information

UR5 Connection & Control & Data Parsing
Input: IP of UR5, commands of UR5 **Output**: Status information of UR5
Get Connection socket (UR5_IP, 30003 port)
Define command_send (commands of UR5) function
Get part grasping strategy
Transform strategy into control command of UR5
Send the command to UR5
END the command_send
Define Data_parsing (information from UR5)
Transform byte information into Hex_string type
Transform Hex_string type into double type
Return status information of UR5

4. Conclusion

In this paper, based on the cloud-edge-terminal structure as the overall framework, a set of parts dynamic grasping system for UR5 manipulator is designed. By the network protocol, a stable information transmission channel is established. Based on the type of part information stored in the cloud manufacturing platform and the actual grasping method, a dynamic grasping algorithm for parts is proposed. According to the results of cloud computing, the online generation of manipulator control commands is realized. The control commands are transmitted to the manipulator to complete the grasping task, and the real-time monitoring of the robot status is completed. Through cloud database creation, connection and read-write technology, cloud communication is realized.

Acknowledgements

This work was supported in part by the JCKY Project of China (#JCKY2019603C016).

References

Kebria, Parham M., AlWais, Saba, Abdi, Hamid, Nahavandi, Saeid, Kinematic and dynamic modelling of UR5 manipulator, *2016 IEEE International Conference on Systems, Man, and Cybernetics, SMC 2016-Conference Proceedings*, pp. 4229-4234, February 6, 2017.

Jiang, Rong, Wang, Zhipeng, He, Bin, Di, Zhou, Vision-Based Deep Reinforcement Learning for UR5 Robot Motion Control, *2021 IEEE International Conference on Consumer Electronics and Computer Engineering*, pp. 246-250, January 15, 2021.

Joshi, Keyur D., Chauhan, Vedang, Surgenor, Brian, A flexible machine vision system for small part inspection based on a hybrid SVM/ANN approach, Journal of Intelligent Manufacturing, vol. 31, no. 1, pp. 103-125, January 1, 2020.

Lin, Chieh-Chun, Gonzalez, Pablo, Cheng, Ming-Yang, Luo, GuorYieh, Kao, Tzu-Yang, Vision based object grasping of industrial manipulator, *2016 International Conference on Advanced Robotics and Intelligent Systems*, ARIS 2016, March 24, 2017.

Walter, Christoph, Scholle, JB., Elkmann, N., Catering to Real-Time Requirements of Cloud-Connected Mobile Manipulators, *KUNSTLICHE INTELLIGENZ*,vol. 33, no. 2, pp. 143-150, January, 2019.

Pontes, Felipe Arruda, Curry, Edward, Cloud-Edge Microservice Architecture for DNN-based Distributed Multimedia Event Processing, *Communications in Computer and Information Science*, vol. 1360, pp. 65-72, 2021.

Wang, Dequan, Xiong, Ao, Liao, Boxian, Yang, Chao, et al., Improved Genetic Algorithm for Computation Offloading in Cloud-Edge-Terminal Collaboration Networks, *Advances in Intelligent Systems and Computing*, vol. 1274

AISC, pp. 1493-1499, 2021.

Dong, Yunmeng, Xu, Gaochao, Zhang, Meng; Meng, Xiangyu, A high-efficient joint 'Cloud-Edge' aware strategy for task deployment and load balancing, *IEEE Access*, vol. 9, pp. 12791-12802, 2021.

De Souza, Felipe Rodrigo, Dias De Assuncao, Marcos, et al., An optimal model for optimizing the placement and parallelism of data stream processing applications on cloud-edge computing, *Proceedings - Symposium on Computer Architecture and High Performance Computing*, vol. 2020-September, pp. 59-66, September 2020.

Ou, Qinghai, Wang, Yanru, Song, Wei, Zhang, Ningchi; Zhang, Jie; Liu, Hui, Research on network performance optimization technology based on cloud-edge collaborative architecture, *2021 IEEE 2nd International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering, ICBAIE 2021*, pp. 274-278, March 26, 2021.