The ROS-based web information center of small manipulator

Yue-Jie Wang
Graduate School of Intelligent Data Science, National Yunlin University of Science & Technology,
123 Section 3, University Road, Douliou, Yunlin 64002, Taiwan, R.O.C.

Jia-Ming Hsiao*
Bachelor Program in Intelligent Robotics, National Yunlin University of Science & Technology,
123 Section 3, University Road, Douliou, Yunlin 64002, Taiwan, R.O.C.

Shao-Yi Hsiao
Department of Intelligent Robotics, National Pingtung University ,
51 Min Sheng E. Road, Pingtung 900, Taiwan, R.O.C.
Email: M11156018@yuntech.edu.tw, hsiaojm@yuntech.edu.tw, shiao9212@gmail.com*

Abstract

A ROS-based web information center with respect to small manipulator is developed to show the feasibility of the one for factories. Small ROS-based 6-axis manipulators for teaching and research are utilized to simulate the applications of robot arms in the factories. The task of robot arm is designed to recognize the object color and then to grip the object to the desired storage area. Production operation data and status information from the robot arm are transmitted through the ROS rosbridge_suite to the web interface for display and saved in a database. A start-stop function is also implemented to remotely start and stop the operation of robot arm.

Keywords: Robot Operating System (ROS), rosbridge, manipulator

1. Introduction

With the development of Industry 4.0, smart manufacturing has become one of important issues of modern intelligent industry. Through smart manufacturing, companies from manufacturing industry can use advanced technologies such as data analysis, automation, network technology, and artificial intelligence to realize the intelligence and digitization of the production process. The three key technologies for building a smart manufacturing factory include data collection and analysis, automation and robotics, and digital manufacturing and virtual simulation [1]. There are variety of sensors and monitor facilities in the smart factory to collect manufacturing data, and this data needs to be processed and analyzed effectively. With the respect of monitoring production data, the graphical user interface (GUI) is an intuitive interface to show the real time process information in order to track and manage easily [2]. The GUI for monitoring production data can be easily implemented by dynamic web techniques.

Automation and robotics are the key technologies to reduce human errors and improve production efficiency and quality. However, the communications among different robot systems are difficult before the announcement of robot operating system (ROS). As the middleware of robot, the purpose of ROS is to solve issues of compatibility and interoperability. ROS provides system developers with modular development framework and more flexibility, and allows the combination of small modules to perform more complex tasks. Therefore, ROS simplifies the programming process, which also makes the ROS code more versatile [3]. As a package of ROS, rosbridge provides a JSON API to ROS functionality for non-ROS programs [4]. The communications between front webpages and robots can be made by rosbridge to implement GUI information center to show and monitor the process data, and even to remotely control the robot.

In the present study, we introduced a ROS-based web information center to monitor the process information of small ROS-based 6-axis manipulators. The ROS-based 6-axis manipulators for teaching and research are utilized to simulate the applications of robot arms in the factories. The task of robot arm is designed to recognize the object color and then to grip the object to the desired storage area. Production operation data and status information from the robot arm are transmitted through the ROS rosbridge package to the web interface for display and saved in a database. A start-stop function is also implemented to remotely start and stop the operation of robot arm.

2. Methodology

The proposed system consists of object identification, ROS-based manipulators, MySQL relational database management system and web station. The object identification function and ROS-based manipulators are combined to simulate a part of production process of packaging. The process data is stored in MySQL, and then can be shown in the front web page, as well as the working status of manipulators. The communications

between web and manipulators, web and camera are implemented by ROS rosbridge package.

2.1. *Object color identification*

The production object is identified by the camera image color. A ROS node is designed to subscribe the topic published by the ROS node usb_cam from the package usb_cam in order to reduce the computation load of the manipulator. The color identification function is based on HSV (Hue, Saturation, Value) color model. First the BGR (Blue, Green, Red) color model of the camera image is transferred to HSV model. Then the calculations of HSV values from BGR values are performed. The calculation equations of HSV values from BGR values are omitted and can be referred to [5]. The (x, y) pixel threshold value of image color identification is obtained by Eq. (1), where L is the lower bound and U is the upper bound. Then AND bitwise operation is used to apply the mask generated by threshold filtering to the original image, retaining only the parts that match the color range. After the object color identification is done, the manipulator is activated to grip the target object to the designated area.

$$M(x,y) = \begin{cases} 1, if\ L \le H(x,y), S(x,y), V(x,y) \le U \\ 0, otherwise \end{cases} \quad (1)$$

2.2. *ROS-based 6-axis manipulator*

In this study mecharm270 pi which is a small ROS-based 6axis manipulator for teaching and research as shown in Fig. 1 is applied to simulate the production process in the factories. A python node callback.py is launched to subscribed the topic published from the web interface to perform the start and stop of the manipulator. The communication between web interface and callback.py node is established by rosbridge package as shown in Fig. 2. Two publisher nodes are implemented in the manipulator to publish the production data as shown in Fig. 3.

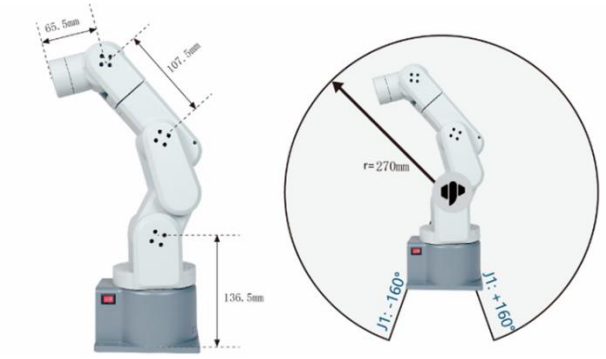

Fig. 1. mecharm270 pi

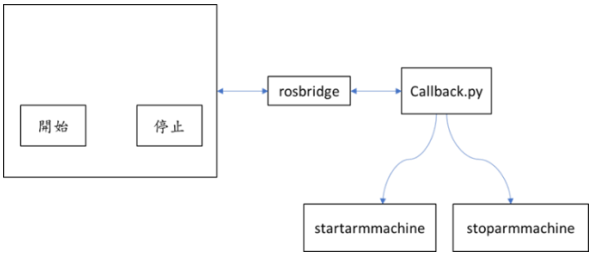

Fig. 2. Communication structure of web interface start-stop function and node callback.py

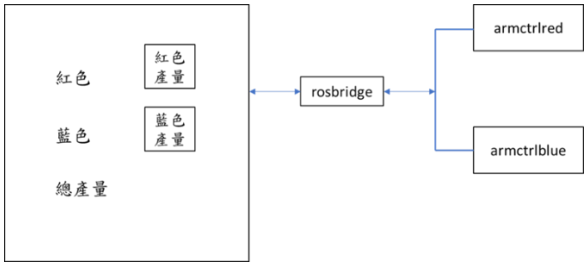

Fig. 3. Production data and publisher nodes

2.3. *Web interface and database*

The web interface is constructed by node.js and a MySQL database server runs for data storage. The layout for one status and data is arranged as Fig. 4. The left column is designed for tool icons and the right hand side is kept to show the manipulator information. The start and stop buttons function to remotely start and stop the manipulator movement. The process data is shown in the object amount numbers. The real time camera image is also monitored in the interface. It is shown in Fig. 5 that the communication between the web user interface which is a non-ROS client and the manipulator as ROS server is made by ROS rosbridge package.

The production process data is accessed in the backend of the web interface. The object amount data is stored in MySQL in order to track the process information in the future.

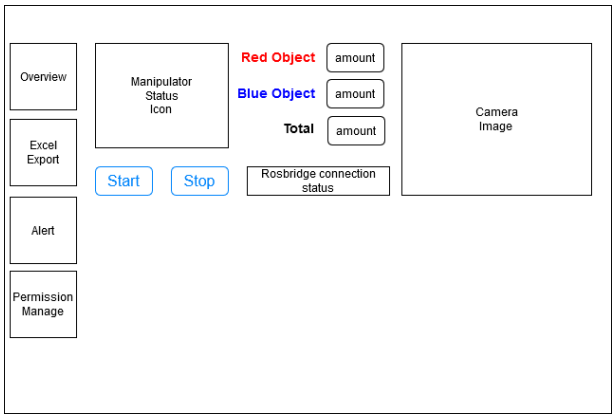

Fig. 4. Web interface layout.

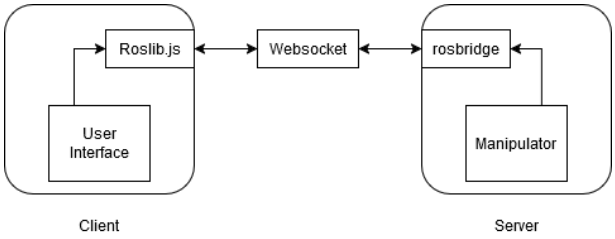

Fig. 5. Rosbridge communication.

3. Results and Discussion

The result web interface is shown as Fig. 6. There are 2 manipulators process data shown in the interface. The object image is obtained by the ROS usb_cam node. The topic /image_raw is published by usb_cam node and the node image_view and web interface subscribe it. The image_view output the image to the display window as shown in Fig. 7. The object color identification is performed by utilizing OpenCV python library with HSV model. The thresholding upper and lower bound of color red is selected as [10, 255, 255] and [0, 100, 100], respectively. The thresholding upper and lower bound of color blue is selected as [130, 255, 255] and [110, 50, 50], respectively. It is shown in Fig. 8 that identification results of red and blue objects. The initial image is transformed from ROS image format into OpenCV image format by our combined_color_recognition node and CvBridge package. After the target object is detected, the manipulator is activated to grip the target object.

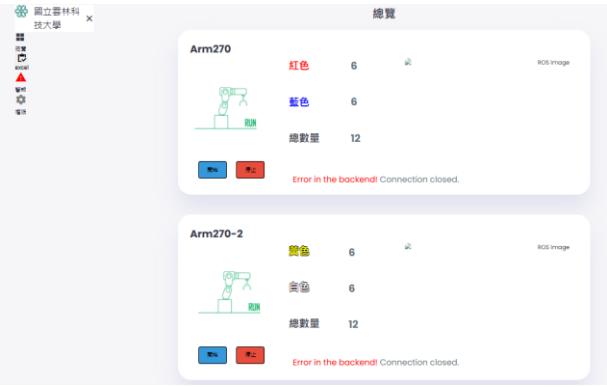

Fig. 6. Result web interface.

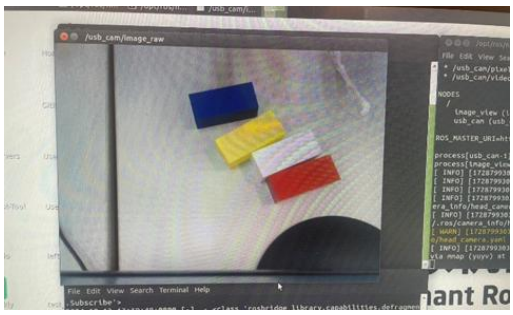

Fig. 7. The object image.

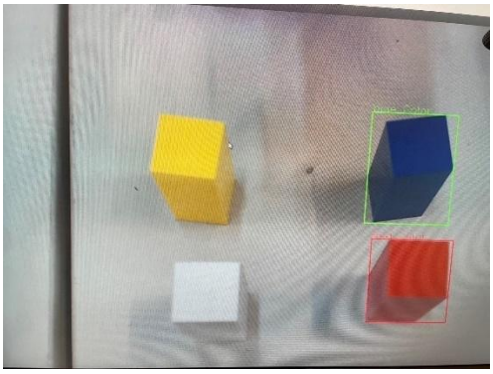

Fig. 8. identification results of red and blue objects.

After the gripping activation is done, the manipulator publishes a message of target color object amount. The web interface subscribes this message and calls self-defined function updateDatabase() to insert the process object data into database table with JSON format. Fig. 9 is a sample result for the database records.

id	red	blue	redtotal	bluetotal	total	timestamp_column
1	0	1	0	1	1	2024-09-30 16:23:25
2	1	0	1	1	2	2024-09-30 16:23:52
3	1	0	2	1	3	2024-09-30 16:24:26

Fig. 9. Process data records.

The process data is queried from database and shown in web interface as Fig. 6. This data can be exported to excel file by clicking the “excel” tool icon in the left column of web interface. The exportation result is shown in Fig. 10.

ID	red	blue	redtotal	bluetotal	total	Timestamp
1	0	1	0	1	1	2024-09-30 16:23:25
2	1	0	1	1	2	2024-09-30 16:23:52
3	1	0	2	1	3	2024-09-30 16:24:26
4	1	0	3	1	4	2024-09-30 16:32:50
5	0	1	3	2	5	2024-09-30 22:18:14
6	1	0	4	2	6	2024-09-30 22:20:14
7	1	0	5	2	7	2024-09-30 22:21:57
8	1	0	6	2	8	2024-09-30 22:22:28
9	1	0	7	2	9	2024-09-30 22:23:06
10	0	1	7	3	10	2024-09-30 22:23:37
11	0	1	7	4	11	2024-09-30 22:25:02
12	1	0	8	4	12	2024-09-30 22:25:34
13	1	0	9	4	13	2024-09-30 22:32:46
14	1	0	10	4	14	2024-10-01 13:25:11
15	1	0	10	4	14	2024-10-01 13:25:11
16	0	1	10	5	15	2024-10-13 16:17:08
17	0	1	10	6	16	2024-11-03 15:24:30
18	0	1	10	7	17	2024-11-03 15:25:59
19	0	1	10	8	18	2024-11-03 15:27:55
20	0	1	10	9	19	2024-11-03 15:29:29
21	0	1	10	10	8	2024-11-04 15:46:54
22	0	1	10	11	9	2024-11-04 15:48:30
23	0	1	10	9	19	2024-11-04 15:52:43
24	0	1	10	10	8	2024-11-04 15:54:10
25	0	1	10	11	8	2024-11-04 16:00:34
26	0	1	10	9	19	2024-11-04 16:02:50
27	0	1	0	1	1	2024-11-05 16:03:24

Production Data | Alert Data | Production2 Data | Alert2 Data

Fig. 10. Process data exported to excel file.

The manipulator status is also displayed in the interface with 3 icon images. The 3 status icons include Run, Idle and Alarm as shown in Fig. 11. The blue “Start” button is used to remotely activate the manipulator to work, and

the red "stop" button is used to remotely stop the manipulator.

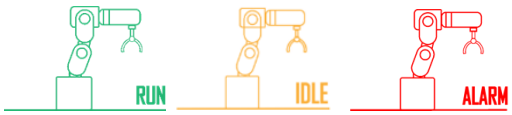

Fig. 11. Manipulator status icons.

4. Conclusion

Graphical user interfaces are an integral part of smart factories, but ROS is not friendly to novice developers. Developers need a deeper understanding of ROS, and there is a lack of specific suppliers to provide GUI options. This study proposes a customized web visual monitoring interface based on the ROS rosbridge package, which is applied to collect the equipment status, color classification and production data of ROS equipment operating in the factory. Rosbridge is used to connect ROS-based robot arms and non-ROS visual monitoring interfaces for bidirectional communication. In this way, non-ROS devices or systems can also be integrated into ROS systems. It will be convenient to develop the hardware and software integration under ROS framework.

Table. 1. Comparison of protocols

Item \ Protocol	OPC	OPC UA	Rosbridge
Platform supported	MS Windows	MS Windows, Linux, iOS	MS Windows, Linux, iOS, ROS, Web
Security	High	High	Medium
Expansibility	Low	High	High
Development difficulty	Medium	High	Low
Typical application scenarios	Local Monitoring	Cross-system data integration, industrial Internet of Things	Robot control, Web monitoring, Data visualization

The comparison of open platform communications (OPC), OPC unified architecture (OPC UA) and Rosbridge is listed in Table. 1. Rosbridge has lower development difficulty and higher expansibility, but Rosbridge also has some problems that need to be optimized. For example, the security design of industrial applications is relatively simple, and for high standardization of industrial security, security is low. As the number of ROS users who develop robots grows year by year, the need to integrate ROS and non-ROS applications will be an issue worthy of attention.

References

1. Osterrieder, P., Budde, L., & Friedli, T., The smart factory as a key construct of Industry 4.0: A systematic literature review, International Journal of Production Economics, Vol. 221, 2020, 107476.
2. Kumar, N., & Lee, S. C., Human-machine interface in smart factory: A systematic literature review, Technological Forecasting and Social Change, Vol. 174, January 2022, 121284.
3. Hax, V. A., Duarte Filho, N. L., Da Costa Botelho, S. S., & Mendizabal, O. M., ROS as a middleware to Internet of Things, Journal of Applied Computing Research, Vol. 2, 2012, pp.91–97.
4. ROS wiki – rosbridge_suite, http://wiki.ros.org/rosbridge_suite
5. Loesdau, M., Chabrier, S., & Gabillon, A., Hue and saturation in the RGB color space, In Lecture Notes in Computer Science, 2014, pp.120–128.

Authors Introduction

Mr. Yue-Jia Wang

He received his Bachelor's degree in Management in 2022 from the Department of Industrial & Management, National Chin-Yi University of technology in Taiwan, R.O.C.. He is currently a master student in National Yunlin University of Science & Technology, Yunlin, Taiwan, R.O.C..

Prof. Jia-Ming Hsiao

He received the B.S. and Ph.D. degree in Electrical Engineering from National Sun Yat-Sen University, Kaohsiung, Taiwan in 1994 and 2002, respectively. He is an Assistant Professor in the Bachelor Program in Intelligent Robotics at National Yunlin University of Science & Technology, Yunlin, Taiwan, R.O.C.. He is also the 2022-2025 youth group vice presiding judge of mobile robotics at national skills competition in Taiwan. His current research interests include system and control theory, internet of things, mobile robotics and artificial intelligence.

Mr. Shao-Yi Hsiao

He is currently an undergraduate student in Department of Intelligent Robotics, National Pingtung University in Taiwan, R.O.C.. His current research interests include internet of things and mobile robotics.