Journal of Artificial Intelligence Practice (2024)
Clausius Scientific Press, Canada

DOI: 10.23977/jaip.2024.070211
ISSN 2371-8412 Vol. 7 Num. 2

Practical Analysis of Building Robot Operating Systems Based on Scientific Research Projects

Xiaodan Yu[1], Kun Zhang[2,a,*], Jianguo Wu[1], Suichun Qu[2], Yongjun Gu[1]

[1]*School of Electrical and Energy Engineering, Nantong Institute of Technology, Nantong, Jiangsu, China*

[2]*School of Electrical Engineering, Nantong University, Nantong, Jiangsu, China*

[a]*1966387528@qq.com*

[*]*Corresponding author*

Keywords: Robot Operating System, Modular Design, Perception and Cognitive Technology, Human Computer Interaction Design

Abstract: The operating system is the core of the robot system. It is the key to ensuring the safety, effectiveness, and intelligence of robot systems. This article takes the "Autonomous Navigation Robot" research project as the background and conducts practical research on the robot operating system. The research background focuses on the limitations of some robot operating systems, namely that current robot operating systems are not suitable for robots working in resource limited environments, and the ability to adapt to dynamic changes and unstructured environments is very important. The system adopts a modular design concept, emphasizing real-time, robustness, and scalability. This article focuses on human perception and cognitive technology, as well as the design of interaction between people. During the system development process, work in conjunction with relevant research work. A series of tests and evaluations were conducted on the independently developed autonomous navigation robot operating system, including unit testing, integration testing, and on-site testing. At the same time, a performance comparison between the Robot Operating System (ROS) and Open Robot Control Software (ORCA) systems, which are of great concern in relevant research literature, was presented. The experimental results show that the autonomous navigation robot operating system exhibits superiority in key performance indicators such as failure rate, delay time, and energy efficiency ratio, especially achieving an excellent performance of up to 408 tasks/Wh in energy efficiency ratio, significantly superior to ROS and ORCA systems. The conclusion of this study is that the autonomous navigation robot operating system not only meets the needs of current autonomous navigation robot research projects, but also has good scalability and real-time performance, providing a solid technical foundation for the development and application of future robotics technology.

1. Introduction

As the brain of robotics technology, the performance of the operating system directly affects the overall work efficiency and intelligence level of robots. However, the performance limitations of existing robot operating systems in resource limited environments and the need for adaptability to unstructured environments have become key factors restricting the further development of robot technology. In response to this situation, this study aims to rely on the "Autonomous Navigation Robot" research project to conduct in-depth practical analysis of robot operating systems, in order to break through existing technological bottlenecks and promote the development of robot operating systems towards more efficient and intelligent directions.

This article proposes a development method for robot operating systems based on modular design principles, which effectively improves the performance and adaptability of the system by emphasizing real-time and scalability. Secondly, this study employed a series of testing and validation strategies to comprehensively evaluate the performance of the self-developed autonomous navigation robot operating system compared to existing ROS and ORCA systems, providing a scientific basis for the selection and optimization of robot operating systems.

The first part of the article is the introduction, which introduces the research background, contributions, and structure of the article; The second part is related work, summarizing the research status and development trends of robot operating systems; The third part is the methodology, which provides a detailed introduction to the development methods and key technologies in the practical process of robot operating systems; The fourth part is the testing and validation strategy, which presents the testing process and results of a comprehensive evaluation of the autonomous navigation robot operating system; The fifth part is the results and discussion, which compares and analyzes the differences in key performance indicators between the autonomous navigation robot operating system and ROS and ORCA systems, and explores their significance; The last part is the conclusion, summarizing the main findings of this study and proposing prospects for future research directions.

2. Related Work

The safety, efficiency, and intelligence of robot systems have become a hot research topic. Lu Jingjing aims to raise people's attention to the safety of robot systems and help them quickly understand the security solutions of current mainstream robot operating systems. He adopted in-depth analysis and classification research methods to systematically investigate and summarize the security issues of existing robot operating systems [1]. Tu Qiu proposed a method to improve the accuracy and real-time performance of image detection in traditional embedded systems, which often limits the efficiency of communication and image transmission due to limited computing resources [2]. Wu Xiaoze discussed the key technologies of industrial robot operating systems based on hybrid key systems, proposed an architecture design for robots to address the technical bottlenecks faced by robots in unstructured or semi-structured intelligent scenarios, and predicted the future evolution direction of industrial robot operating systems [3]. Hou Renluan summarized the research status of world robot operating systems from several aspects, including commercial industrial robot operating systems, open-source robot operating systems, and cloud-based integrated systems. He combines the current trend of integrating robots with artificial intelligence, 5G communication and other technologies to propose the establishment of a future cloud edge integrated collaborative intelligent robot operating system platform and the bottleneck problems that need to be solved [4]. There is a certain difference between the data accuracy of Jiang Hongmei's current navigation electronic map and the accuracy required for the actual driving process. In order to provide ultra high precision navigation maps and object recognition, he proposed the design of an autonomous driving car system based on ROS [5].

In addition, Chen L proposed a ubiquitous operating system based on parallel driving theory, named Parallel Driving Operating System. Specifically, the system architecture includes hardware layer, kernel layer, functional layer, and application layer, which can respectively achieve heterogeneous hardware support, ubiquitous resource management, algorithm optimization, and ubiquitous application development [6]. Haddadin S explored three different operating interfaces: Desk, Robot Integrated Development Environment, and Franka Control Interface. He provided guidance on how to use these interfaces for task programming, creating robot skills, application development, and integrating external sensors [7]. Jin X addresses the challenges of vascular intervention surgery by developing a tactile sensing robot assisted system. The design of the system allows surgeons to perform surgical operations under lower X-ray exposure. In order to verify the effectiveness of the system, he conducted a series of in vitro experiments to test the accuracy and safety of the system in collaborative operations [8]. Montero E E adopts deep reinforcement learning methods to improve the navigation ability of robots in crowded environments. He proposed the concept of a dynamic warning zone, which forms a circular sector around a person based on their step size and speed, in order to enhance the robot's ability to avoid collisions [9]. Chang Y has integrated a single robot front-end interface that adapts to different odometer sources and LiDAR configurations to support closed loop detection between robots and within robots in large-scale environments and multi-robot teams [10]. Although the above research has achieved significant results in the security, communication efficiency, and architecture design of robot operating systems, the performance of existing systems may be limited in resource limited environments, and there is still room for improvement in their adaptability to unstructured environments. The purpose of this study is to conduct in-depth practical analysis of robot operating systems based on scientific research projects. The research can focus on building an efficient, secure, highly adaptable, and scalable robot operating system, exploring the advantages and disadvantages of existing robot operating systems, and proposing improvement solutions for the identified problems.

3. Method

3.1. Selection and Evaluation of Scientific Research Projects

This study chooses the development of an operating system for autonomous navigation robots as the core of the research project, and conducts practical analysis of robot operating systems [11-12]. This study first clarified the specific objectives of the project, including achieving path planning in complex environments, obstacle identification and avoidance, and autonomous decision-making functions. These goals are the key basis for evaluating the compatibility of robot operating systems.

This work compares the project requirements with the functions currently provided by the robot operating system in more detail to calculate the consistency between the functional requirements and the existing functions [13-14]. In addition, the application scenarios of the project (such as the adaptability of the robot to different ground conditions during indoor and outdoor navigation) have also been fully considered to ensure that the selected operating system can meet the requirements of the given working environment. Technical feasibility analysis focuses on the technical maturity of robot operating system, including collecting and analyzing performance data of operating system, user feedback and analyzing server environment activities. At the same time, the research also describes several technical resources needed for project implementation, such as professional software tools, hardware platforms, development documents, etc., and checks whether these resources can be used to support project implementation. The study also discussed the human resources needed for the project, including the technical feasibility and expertise of the team. In addition, it also tries to find out the potential technical risks, problems and challenges that may be encountered, and formulate countermeasures against these potential risks.

3.2. Development Methods for Robot Operating Systems

Based on the principle of modular design, this paper constructs a robot operating system. The system is divided into several independent functional modules, such as sensor input module, path planning module, navigation control module and so on. This design method makes the system clear, easy to maintain, and convenient for team cooperation, so that each team member can concentrate on developing the module he is responsible for [15-16]. Moreover, modular design is also beneficial to the stability of the system. Each functional module is relatively independent, and can be updated and optimized independently according to its own needs, without affecting the stability of other modules. Real-time is an important feature of autonomous navigation robot operating system. The system can respond to the sensor input and complete the navigation task within a predetermined time as required. Therefore, the real-time operating system is selected to ensure the real-time performance of the system. In the system design, the real-time requirements of the system, such as task mode, priority scheduling and interrupt handling, are fully considered. From the perspective of scalability, the system aims to be compatible with future hardware updates and functional expansion. For example, reserved interfaces are reserved for new sensors and actuators, and flexible data structures and communication protocols are designed according to the specific navigation environment and task requirements. It adopts standardized module interface and open source plug-in architecture design, which can effectively avoid a lot of investment in new module development, and can simply integrate new functional modules to adapt to the changes of autonomous navigation technology.

3.3. Key Technologies in the Practice Process

Perceptual cognitive technology is the robot's ability to perceive the environment. In the decision-making process, the robot uses sensor data to determine the position of obstacles, measure the distance between the robot and obstacles, and identify landmarks [17-18]. In this study, the combination technology of machine vision, laser radar and ultrasonic sensor is used to realize high-precision environmental perception. The mathematical model of data fusion for multi-sensor fusion is expressed as:

$$p_{combined} = f\,(p_{vision}, p_{LiDAR}, p_{ultrasonic}) \tag{1}$$

$p_{combined}$ is the fused perception data, p_{vision}, p_{LiDAR}, and $p_{ultrasonic}$ are perception data from machine vision, LiDAR, and ultrasonic sensors, respectively. f is a fusion function used to integrate these data.

Cognitive technology is a robot that processes and recognizes perceptual data, such as scene analysis, decision-making and behavior prediction. In this study, deep learning algorithm is used to improve the cognitive level of robots, so that robots can make reasonable decisions in complex environments. Secondly, the research pays attention to the intuition and friendliness of the interactive interface to ensure that ordinary people can interact with the robot easily, and adjust the design according to different habits and hobbies of different people. The research also integrates natural language processing (NLP) technology to enable robots to understand and execute natural language instructions, which is also a step towards natural interaction. At the same time, in the design process, considering the user's skills and cognitive limitations, the design process can be reflected in the following formal iterative model:

$$D_{n+1} = h(D_n, F_n, U_n) \tag{2}$$

D_{n+1} is the $n+1$ th iteration, D_n is the interaction design after the n th iteration, and F_n is the user feedback collected after the n th iteration. U_n is the user satisfaction after the n th iteration, and h is an update function used to improve design based on user feedback and satisfaction.

3.4. Testing and Verification Strategy

The testing and validation strategy is a crucial step in ensuring system reliability and performance meets expectations, and unit testing is aimed at testing each independent module in the robot operating system [19-20].

Unit testing uses assertions to verify the correctness of code, ensuring that each function or method produces the expected output under a given input. For unit testing of path planning modules, it can be represented as:

$$T_{planning} = \{test\_case | output = planning\_function(input, parameters) \wedge verify(output)\} \tag{3}$$

$T_{planning}$ is the set of test cases for the path planning module, $input$ is a single test case, which is the test input, and $output$ is the output. $parameters$ is the module parameter, $test\_case$ $planning\_function$ is the path planning function, $verify$ is the validation function, which is used to check whether the output meets expectations.

Integration testing is conducted after unit testing to verify the interfaces and interactions between different modules. This study adopted a step-by-step integration strategy, which involves integrating one or several modules at a time to gradually build a complete system. At each integration stage, integration testing is performed to ensure that newly added modules do not compromise the functionality of existing modules:

$$T_{\text{integration}} = \{(M_i, M_j, test\_\text{case}) | M_i \wedge M_j \wedge \text{interaction\_test}(M_i, M_j, test\_\text{case})\} \tag{4}$$

$T_{\text{integration}}$ is the set of integration test cases, M_i and M_j are the modules involved in integration, and interaction_test is the testing function used to test the interaction between modules.

On site testing is conducted in practical application environments to verify the performance and stability of the robot operating system under real working conditions. The results of on-site testing are used for the final adjustment and optimization of the system. The mathematical model for on-site testing can be expressed as:

$$T_{field} = \{scenario | performance\_metrics(scenario) \geq threshold\} \tag{5}$$

T_{field} is a set of on-site testing scenarios, $scenario$ is a single testing scenario, $performance\_metrics$ is a performance metric function, and $threshold$ is a performance threshold.

Through these three stages of testing and verification, this study can ensure that the robot operating system meets the requirements of scientific research projects in terms of functionality, stability, and performance, and provide solid technical support for the practical application of autonomous navigation robots. As shown in Table 1, the system test data is as follows:

The table lists the key performance indicators of the robot operating system during unit testing, integration testing, and on-site testing stages. These data collectively constitute a comprehensive perspective for evaluating the performance of robot operating systems, helping developers identify the advantages and improvement points of the system, and providing a basis for subsequent product optimization. At the same time, the project management team can also use this data to monitor project progress and ensure that the final delivered product quality meets established standards.

Table 1: System test data

Test Phase	Test Cases	Passed Cases	Failed Cases	Avg. Response Time (ms)	System Stability (hrs)	Defects Found	Remarks
Unit Testing	150	147	3	25	-	5	High coverage testing
Integration Test	80	72	8	40	-	8	Inter-module interaction test
Field Testing	30	27	3	35	48	3	Real-world environment simulation

4. Results and Discussion

The purpose of this comparative experiment is to evaluate and compare the differences in key performance indicators between the self-developed robot operating system (autonomous navigation robot operating system) and the existing two mainstream systems - ROS and ORCA. Specifically, this study can focus on three core comparative indicators: failure rate, delay time, and energy efficiency ratio.

At the beginning of the experiment, a unified testing environment was established for the autonomous navigation robot operating system, ROS, and ORCA systems to ensure that all systems were tested under the same hardware and software conditions. Then, ROS and ORCA systems were installed and configured. At the same time, a self-developed autonomous navigation robot operating system can be deployed, and 25 sets of test cases can be developed for each comparative indicator, including stability testing, real-time testing, and energy consumption testing. Stability testing records the number of failures and running time of each system by running test cases for a long time, while real-time testing tests the delay time of the system through specific real-time tasks. Energy consumption testing monitors and records the energy consumption data of each system during the execution of the same task, collects data generated during the testing process using automated tools, and ensures the accuracy and completeness of the data. Subsequently, statistical analysis can be conducted on the collected data to determine the performance of each system on various indicators.

4.1. Failure Rate

The failure rate, as an important indicator of system stability, can reflect the frequency of problems and failures encountered by various operating systems during long-term operation. The comparison results are shown in Figure 1:

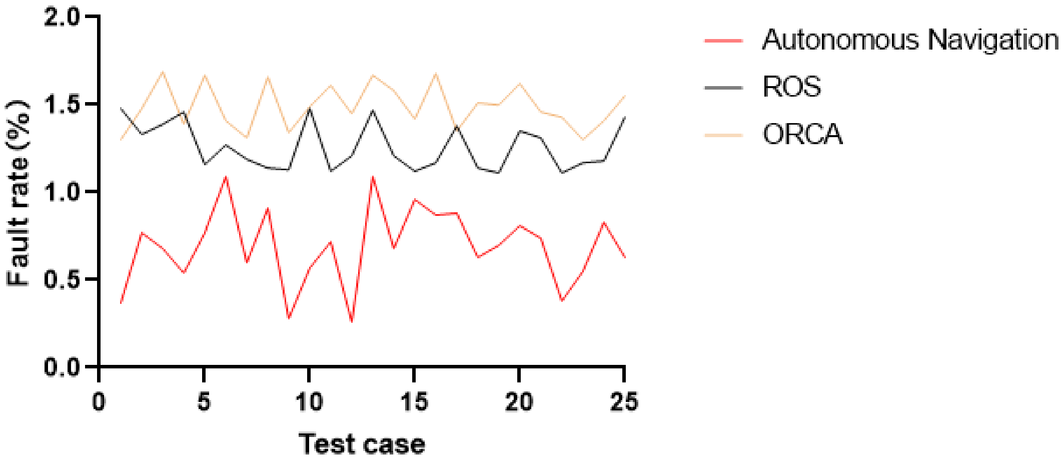

Figure 1: Comparison of failure rates

According to the data in the graph, it can be observed that the failure rates of the three robot operating systems are all maintained at a relatively low level. This indicates that the performance of autonomous navigation robot operating systems, ROS, and ORCA systems is relatively good. Specifically, the autonomous navigation robot operating system studied in this article has a lower failure rate compared to ROS and ORCA. The highest failure rate of the autonomous navigation robot operating system is only 1.09%, but ROS and ORCA reach 1.48% and 1.69% respectively. This result indicates that autonomous navigation robot operating systems have greater advantages in terms of stability. In practical applications, this means that autonomous navigation robot operating systems may require less manual intervention and less downtime for maintenance and repair, which is of great significance for improving production efficiency and reducing operating costs.

4.2. Delay Time

The delay time defines the time interval between the system receiving the input signal and generating the corresponding output response, as shown in Figure 2:

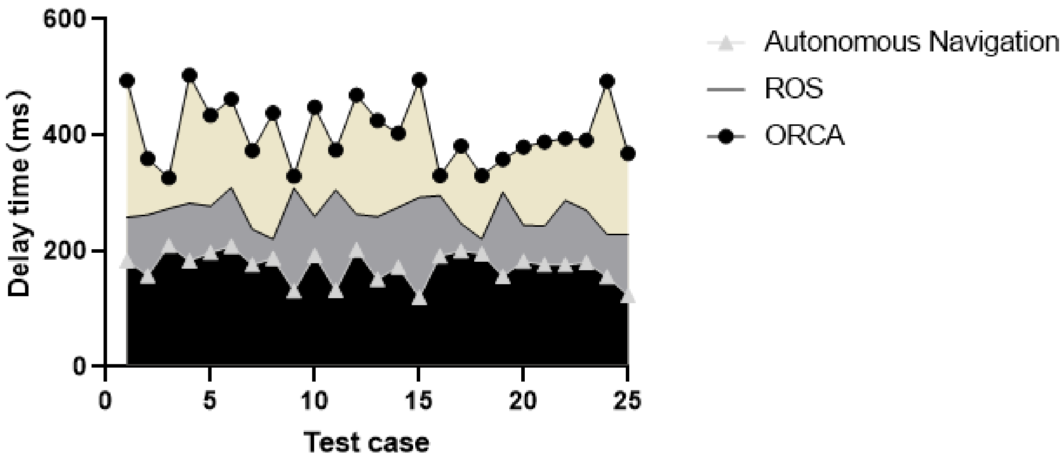

Figure 2: Delay time comparison

As shown in the figure, the latency of the autonomous navigation robot operating system is the lowest, reaching 120ms, but the ROS and ORCA are the lowest, reaching 220ms and 326ms, respectively. The highest values even reached 309ms and 503ms, indicating that autonomous navigation robot operating systems can process sensor inputs and make decisions faster, thereby improving the robot's reaction speed and overall performance.

4.3. Energy Efficiency Ratio

Energy efficiency ratio is an important indicator for measuring the energy efficiency performance of a system, which reflects the efficiency of energy utilization during the execution of tasks. The high or low energy efficiency ratio directly affects the endurance and operating cost of robots, especially in energy-saving situations such as mobile robots or remotely operated robots. The specific experimental data is shown in Figure 3:

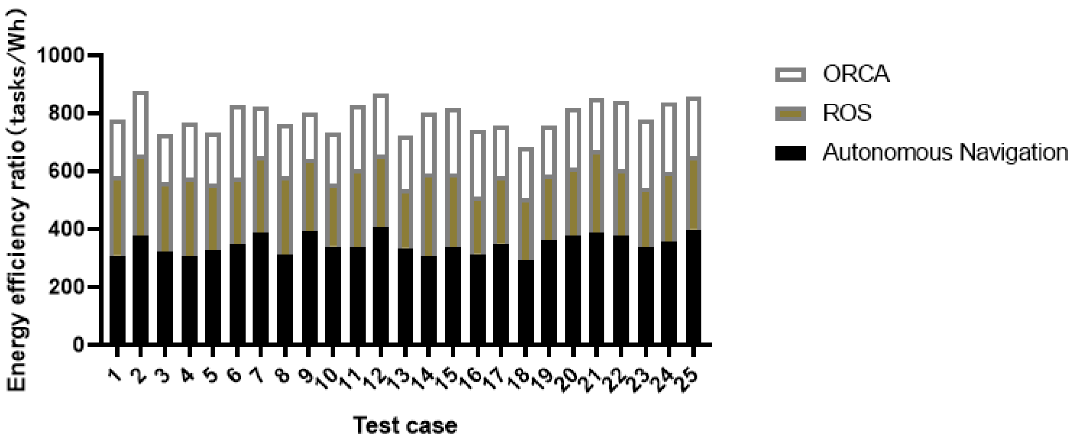

Figure 3: Energy efficiency comparison

As shown in the figure, in each test case, the energy efficiency ratio of the autonomous navigation robot operating system is significantly higher than ROS and ORCA, with a maximum energy efficiency ratio of 408 tasks/Wh. In the same situation, the highest ROS is only 287 tasks/Wh, and the ORCA is only 289 tasks/Wh. This result indicates that, under the same amount of task execution, the autonomous navigation robot operating system can complete more tasks with less energy consumption or with the same amount of energy consumption.

5. Conclusions

This study conducted a comprehensive practical analysis and performance evaluation of the robot operating system based on the "Autonomous Navigation Robot" research project. By adopting modular design principles, considering real-time and scalability, and applying key technologies, this study successfully developed and tested an independently developed autonomous navigation robot operating system. In comparative experiments with existing ROS and ORCA systems, the autonomous navigation robot operating system has shown significant advantages in key performance indicators such as failure rate, delay time, and energy efficiency ratio.

In addition, the stability test results of the system further demonstrate the reliability of the autonomous navigation robot operating system, which is particularly important for long-term robot tasks. Real-time testing reveals the system's ability to respond quickly, which is crucial for application scenarios that require immediate feedback. The results of this study not only provide new perspectives and methods for the development of robot operating systems, but also lay a solid foundation for the innovation and application of future robot technology.

Acknowledgement

Nantong Basic Science Research and Social Livelihood Science and Technology Plan Project "Design and Research of Automatic Sorting System Based on Industrial Cameras and Collaborative Robots" (Project Number: JC1202022049) Achievements.

References

[1] Lu Jingjing, Qin Yunchuan, Liu Zhizhong, Tang Zhuo, Zhang Yongjun, Li Kenli.Review of research on the security of robot operating system ROS. Journal of Software, 2024, 35(2):1010-1027.

[2] Tu Qiu, Zhao Lei, Wang Feng, Zhou Zifa.Research on the improvement of image transmission efficiency of robot operating system (ROS) multi-machine communication in embedded systems. Journal of Shangrao Normal University, 2023, 43(6):20-27.

[3] Wu Xiaoze, Li Qingfeng, Niu Jianwei.Industrial robot operating system based on hybrid critical system. Single chip Microcomputer and Embedded system Application, 2023, 23(12):4-7.

[4] Hou Renluan, Tong Kun, Han Bing, Yu Xiaolong, Yang Fan.Review of research on intelligent robot operating system. Hangzhou Science and Technology, 2023, 54(3):58-64.

[5] Jiang Hongmei, Huang Peng, Tang Jun. Research on autonomous vehicle system based on robot operating system. Technological innovation and application, 2023, 13(17):46-49.

[6] Chen L, Zhang Y, Tian B, et al. Parallel driving OS: A ubiquitous operating system for autonomous driving in CPSS. IEEE Transactions on Intelligent Vehicles, 2022, 7(4): 886-895.

[7] Haddadin S, Parusel S, Johannsmeier L, et al. The franka emika robot: A reference platform for robotics research and education. IEEE Robotics & Automation Magazine, 2022, 29(2): 46-64.

[8] Jin X, Guo S, Guo J, et al. Development of a tactile sensing robot-assisted system for vascular interventional surgery. IEEE Sensors Journal, 2021, 21(10): 12284-12294.

[9] Montero E E, Mutahira H, Pico N, et al. Dynamic warning zone and a short-distance goal for autonomous robot navigation using deep reinforcement learning. Complex & Intelligent Systems, 2024, 10(1): 1149-1166.

[10] Chang Y, Ebadi K, Denniston C E, et al. LAMP 2.0: A robust multi-robot SLAM system for operation in challenging large-scale underground environments. IEEE Robotics and Automation Letters, 2022, 7(4): 9175-9182.

[11] Kroemer O, Niekum S, Konidaris G. A review of robot learning for manipulation: Challenges, representations, and algorithms. Journal of machine learning research, 2021, 22(30): 1-82.

[12] Huang M H, Rust R T. Engaged to a robot? The role of AI in service. Journal of Service Research, 2021, 24(1): 30-41.

[13] McLeay F, Osburg V S, Yoganathan V, et al. Replaced by a Robot: Service Implications in the Age of the Machine. Journal of Service Research, 2021, 24(1): 104-121.

[14] Koch M, Manuylov I, Smolka M. Robots and firms. The Economic Journal, 2021, 131(638): 2553-2584.

[15] Dixon J, Hong B, Wu L. The robot revolution: Managerial and employment consequences for firms. Management Science, 2021, 67(9): 5586-5605.

[16] Chang Y, Ebadi K, Denniston C E, et al. LAMP 2.0: A robust multi-robot SLAM system for operation in challenging large-scale underground environments. IEEE Robotics and Automation Letters, 2022, 7(4): 9175-9182.

[17] Wang G, Li W, Jiang C, et al. Simultaneous calibration of multicoordinates for a dual-robot system by solving the AXB= YCZ problem. IEEE Transactions on Robotics, 2021, 37(4): 1172-1185.

[18] Bravi C A, Paciotti M, Sarchi L, et al. Robot-assisted radical prostatectomy with the novel Hugo robotic system: initial experience and optimal surgical set-up at a tertiary referral robotic center. European Urology, 2022, 82(2): 233-237.

[19] Yu D, Chen C L P, Xu H. Fuzzy swarm control based on sliding-mode strategy with self-organized omnidirectional mobile robots system. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2021, 52(4): 2262-2274.

[20] Kumar M R S. Design and developmeent of automatic robotic system for vertical hydroponic farming using iot and big data analysis. Turkish Journal of Computer and Mathematics Education (TURCOMAT), 2021, 12(11): 1597-1607.