Multi-Robot Orchestration for Item Delivery in Health Care Facilities

Sam Cole, **Ankit Shaw**, John Raiti, Haochen Zeng, Sam Wong, Yeary Yuan, Kaiwen Men, Jiawen Chen, Ravinder Gulla, Lakshita Singh, Daithy Ren. Kendra Yang, Ziqi Gao, Aayush Kumar, Louisa Shi, Haonan pengh, Jack Hatcher, Haokun Feng, Mingheng Wu, Hengjun Zhang, Xinyi Zhou, Kate Chen, Matthew Zhang.

ABSTRACT

Multi-robot orchestration in healthcare settings can significantly improve operational efficiency and patient care. This paper presents an innovative system integrating a Kinova Arm and Create 2 robot using the InOrbit operations management platform to coordinate item delivery tasks. Leveraging InOrbit's real-time monitoring and task assignment features, the system ensures seamless communication between the robots. The navigation component, built on the ROS Navigation Stack with a keepout map for obstacle avoidance, and the manipulation component, utilizing the MoveIt package, enable precise control of the Kinova Arm. A robust computer vision pipeline, incorporating the YOLO-V8 model, ensures accurate object detection. Experimental results show a 96% success rate in object placement and improved navigation accuracy. This scalable multi-robot management system enhances healthcare delivery and operational efficiency, highlighting the potential of multi-robot orchestration in medical facilities. Future work will refine algorithms to further improve performance and adaptability in real-world healthcare environments.

KEYWORDS

Multi-Robot Orchestration, Healthcare Robotics, Robot Operations Management, ROS Navigation Stack, Computer Vision, Human-Robot Interaction (HRI), Real-Time Monitoring, Task Coordination, Scalability, Operational Efficiency

ACM Reference Format:

Sam Cole, **Ankit Shaw**, John Raiti, Haochen Zeng, Sam Wong, Yeary Yuan, Kaiwen Men, Jiawen Chen, Ravinder Gulla, Lakshita Singh, Daithy Ren. Kendra Yang, Ziqi Gao, Aayush Kumar, Louisa Shi, Haonan pengh, Jack Hatcher, Haokun Feng, Mingheng Wu, Hengjun Zhang, Xinyi Zhou, Kate Chen, Matthew Zhang. . 2024. Multi-Robot Orchestration for Item Delivery in Health Care Facilities. In *Companion of the 2024 ACM/IEEE International Conference on Human-Robot Interaction (HRI '24 Companion), March 11–14, 2024, Boulder, CO, USA.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3610978.3640748

1 INTRODUCTION

Healthcare robotics is a topic of high interests in the field of Human Robot Interaction. With over 20% of the world's population experiencing different types of health problem such as Cognitive and Physical impairment [1], the inclusion of robots in healthcare settings will not only address the needs of patients' health condition [2] , but also improve the efficiency for healthcare providers in serving such population [3]. Specifically, robots have had increase adaptation in hospitals. Their use includes facilitating rehabilitation [4], monitoring patient vitals [5] and telepresence robots [6].

One application that is gaining popularity are delivery robots in hospitals. COVID-19 showed that health care workers are prone to spreading diseases without protective gear. And with delivery of items being an essential responsibility for nurses, robots offered a safe and viable alternative in delivery due to its inability in contracting viruses. In recent years, robots such as Moxi [7] and Yumi [8] have been deployed hospitals to assist healthcare workers in delivering supplies to hospital rooms and delivering samples to laboratories. These robots have been field-tested and found that it was both acceptable to nurses and doctors, as well as reducing the workload for them.

However, most robots deployed in healthcare facilities are often stand-alone that do not interact with other robots. This causes limitations in scaling the robots to meet multiple needs. For instance, multiple patients might be requesting different items at the same time, thus requires dispatching multiple robots in finishing different delivery requests at a same time. Furthermore, with the usage of different robots in healthcare settings, there requires a standardize communication protocol among such robots in order for effective communication and scalability.

Multi-robot operation management systems are often found in warehouses. For instance, systems such as Kiva [9] and TORU [10] are used to handle the flow of goods from warehouses to customers. Such systems have mostly been used in settings without much human contact hence its application outside of warehouses are left unexplored.

Hence, we propose a multi-robot orchestration system that supports item delivery across both a robotic arm and mobility robot. This system uses InOrbit, a robot operation management system to establish communication between robots. Inorbit also allows users to easily monitor and deploy robot based on their current status. Furthermore, this system incorporates a robust computer vision pipeline to perform pick and place task and a navigation system that adapts to changes in environment. We conducted experiment with the manipulation and navigation system and found that our system achieves high success rate in picking and placing socks, bottles and blankets, and high accuracy in navigating to the designated goal. Our research makes the following contributions:

- A scalable multi-robot management system where that supports interaction across multiple robots

* These authors contributed equally to the work.

HRI '24 Companion, March 11–14, 2024, Boulder, CO, USA

ACM ISBN 979-8-4007-0323-2/24/03
https://doi.org/10.1145/3610978.3640748

- A novel pipeline that utilizes computer vision and YOLO-V8 model in recognizing and picking objects of different form factors with high precision.

2 RELATED WORK

The integration of robot operations management systems in multi-robot environments has been explored extensively in recent years. Several studies have demonstrated the benefits of such systems in enhancing task coordination, efficiency, and reliability. For instance, Francis and Abel [11] discuss the use of a centralized management system for coordinating multiple autonomous drones in a warehouse setting, highlighting significant improvements in task completion time and error reduction.

Similarly, Miorandi [12] explores the deployment of a multi-robot management system in a healthcare setting, where robots are used for delivering medications and supplies. The study found that the system not only improved delivery times but also reduced the workload on human staff.

The use of ROS-based systems for robot communication and control, as discussed by Mohammed Qader Kheder and Aree Ali Mohammed [13], emphasizes the importance of real-time monitoring and data analytics in optimizing robot performance. This aligns with the capabilities provided by InOrbit, which enhances the coordination between the Kinova Arm and Create 2 through its comprehensive dashboard and analytics tools.

Recent advancements in autonomous navigation for robots have significantly improved accuracy and reliability. Zhao, Liu, and Li [14] developed an autonomous navigation system using the ROS framework, focusing on SLAM for mapping and A* and DWA algorithms for path planning to ensure high precision and accuracy in navigation [14]. Additionally, the fusion of LiDAR and IMU sensors, as explored in another research, provides robust indoor localization even in dynamic environments, supporting our need for reliable navigation under varying conditions (A Robust Indoor Localization System Using LiDAR and IMU Sensors, n.d.). Furthermore, an enhanced SLAM algorithm incorporating dynamic object detection improves map accuracy and navigation stability, mirroring our further experiment's focus on parameter tuning for precision and efficiency in autonomous navigation tasks (An Enhanced SLAM Algorithm for Autonomous Mobile Robots in Dynamic Environments, n.d.). These studies collectively underscore the importance of advanced sensor integration and algorithmic refinement in achieving high-precision navigation, which is central to the success of our experimental setup.

3 METHODS

Our robot system comprise of three components: Orchestration, Navigation, and Manipulation (Figure 1).

3.1 Orchestration

In order to establish communication between the Kinova Arm and the Create 2, we utilized a InOrbit[1], a robot operation management system that allows users to initiate multi-robot actions and monitor their progress. This section discusses the integration of the InOrbit robot operations management system to coordinate delivering and pick-and-place tasks using the Kinova Arm and Create 2. InOrbit facilitates multi-robot operations, offering real-time monitoring and control through a comprehensive dashboard. Our robot system consists of 3 types of robots:

- **Kinova Arm**: An advanced robotic arm capable of precise manipulation tasks.
- **Create 2**: A mobile robotic platform designed for navigation and transportation.
- **InOrbit**: A cloud-based robot operations management system that provides real-time dashboards and analytics to monitor and control multiple robots.

To establish communication between the Kinova Arm and Create 2, InOrbit is employed as the high-level control system. InOrbit utilizes a robust communication protocol to ensure seamless data exchange between the Kinova Arm and Create 2. The system leverages ROS topics and services to facilitate command and control operations. The InOrbit Agent is installed on each of the robots in our system. This agent, a python script, establishes network communication between the robot and InOrbit. It also enables robot data to be streamed on the InOrbit Web Interface. The InOrbit republisher's main functionality is to subscribe to ROS topics from the robot and map ROS messages to key-value pairs that are recognizable by InOrbit. Furthermore, ROS messages can be directly published from the web interface to designated topics for communication from the web interface to the robot. With two-way communication established, this integration enables seamless coordination and execution of complex tasks.

3.1.1 *InOrbit Web Interface.*

InOrbit offers a suite of features essential for robot operations management on their web interface:

- **Real-Time Dashboard**: Displays the robots' status, including their ROS (Robot Operating System) nodes and messages. InOrbit's real-time dashboard is crucial for monitoring the status of the robots. Users can view:
  - **ROS Node Status**: Active/inactive status of each ROS node.
  - **Task Progress**: Current state of each assigned task.
  - **Robot Health**: Battery levels, connectivity status, and any error messages.
  - **Environmental Feedback**: Sensor data from the robots' surroundings.
- **Multi-Robot Coordination**: Allows initiation and monitoring of actions across multiple robots.
- **Alerts and Notifications**: Provides real-time alerts for any anomalies or completion of tasks. InOrbit includes robust error handling mechanisms. If a robot encounters an obstacle or fails to complete a task, InOrbit alerts the user and can automatically trigger predefined recovery actions, such as re-routing the Create 2 or reattempting the pick operation with the Kinova Arm.
- **Analytics and Reporting**: Generates detailed reports on robot performance and task completion.

In this system, we utilized real-time dashboard in monitoring the multi-robot communication and the progress of the robot. The dashboard displays the real time location and orientation of the robot (Figure 2.1), and the current action the robot is performing

[1]InOrbit https://www.inorbit.ai

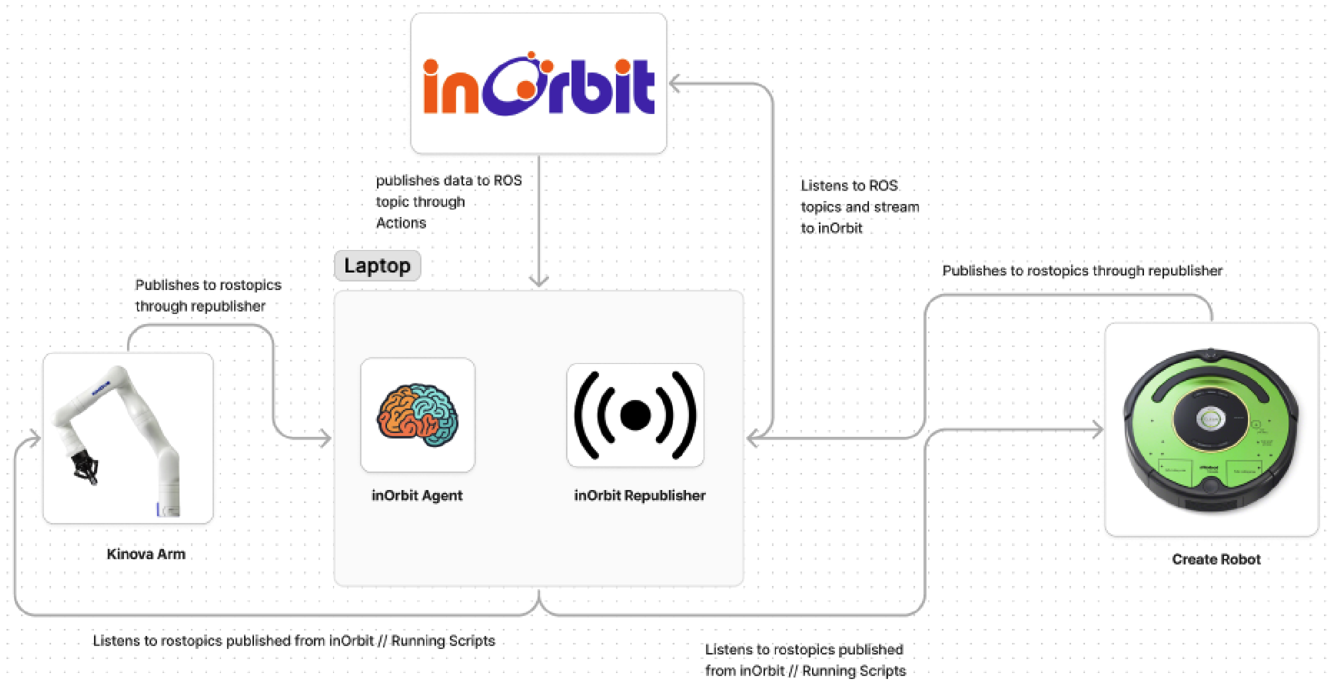

Figure 1: Software architecture of our our robot management system, consisting of orchestration of Kinova Arm and Create 2 robot through the InOrbit agent, which has the ability to subscribe to ROS topics from robots and publish messages to robots.

(Figure 2.2). Any anomalies that occurred during the task execution will be reported as a SEV alert. These interface feature enables users to request items for delivery and see real time information of where the delivery robot is.

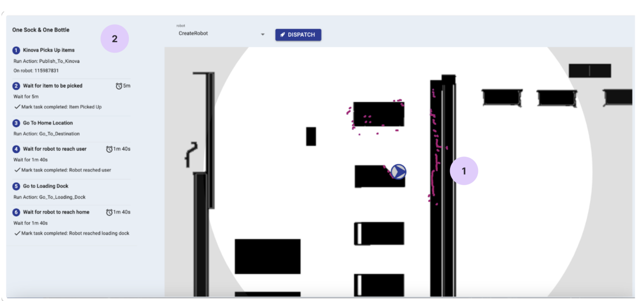

Figure 2: Web Interface of Inoribt. 1) A map displaying the location and the orientation of the robot, together with point clouds. 2) The sequence of actions to be performed by the robot.

3.1.2 Task Definition. The delivering and pick-and-place tasks are defined as a series of coordinated actions between the Kinova Arm and Create 2. For instance:

(1) The Create 2 navigates to a specified location.
(2) The Kinova Arm performs a pick operation at the location.
(3) The Create 2 transports the item to a designated delivery point.
(4) The Kinova Arm performs a place operation at the delivery point.

3.1.3 Workflow.

(1) **Initialization**: The InOrbit system initializes the robots and ensures all ROS nodes are active.
(2) **Task Assignment**: InOrbit assigns specific tasks to the Kinova Arm and Create 2.
(3) **Execution**: The Create 2 navigates to the pick location while the Kinova Arm prepares for the pick operation.
(4) **Coordination**: InOrbit synchronizes the actions, ensuring the Kinova Arm picks the item once Create 2 is in position.
(5) **Transport**: Create 2 moves to the delivery location with the item.
(6) **Placement**: The Kinova Arm places the item at the designated location.
(7) **Monitoring**: Throughout the process, InOrbit's dashboard provides real-time updates on the status and progress of each task.

3.2 Navigation

The navigation of a robot within an environment involves several critical components: localization, mapping, and path planning. Each component utilizes specific algorithms and methods to achieve robust and reliable navigation.

3.2.1 Localization. Localization is the process by which a robot determines its position within a map of the environment. Adaptive Monte Carlo Localization (AMCL) is one of the primary methods employed for localization in our system. AMCL uses a particle filter to represent the distribution of potential robot states, adjusting the set of hypotheses in response to sensory inputs. This method is well-suited for environments with significant levels of noise and

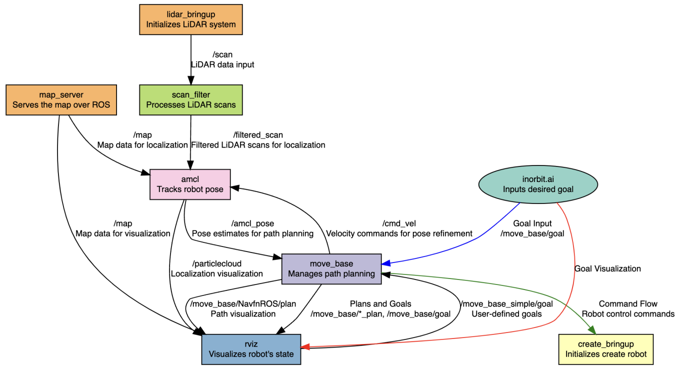

Figure 3: Communication flow within the ROS navigation stack. This diagram illustrates the topics and interactions among key components including `map_server`, `amcl`, `move_base`, `rviz`, `scan_filter`, `lidar_bringup`, and `inorbit.ai`. The edges represent the topic messages exchanged to facilitate localization, mapping, and path planning, ensuring seamless robot navigation.

provides high accuracy by maintaining multiple hypotheses of the robot's position.

3.2.2 Mapping. Mapping involves the creation of a map of the environment which the robot can use to navigate and perform tasks. Our approach uses a 2D occupancy grid map, which represents the environment as a grid of cells, each indicating the presence or absence of an obstacle. The map is dynamically updated as the robot navigates through the environment, using data from LiDAR sensors and the AMCL algorithm to adjust and refine the map.

3.2.3 Path Planning. Path planning is the process of determining a viable path from the robot's current position to a desired goal location, taking into account the map and current localization estimate. We utilize the costmap-based approach in the `move_base` package, which involves two layers of planning: a global planner and a local planner. The global planner creates a high-level route using the entire known map, while the local planner adjusts the path in real-time to avoid dynamic obstacles detected by the robot's sensors. This dual-layer approach allows for flexible, efficient navigation that adapts to changes in the environment.

These methods collectively form the backbone of our robot's ability to autonomously navigate and perform tasks within complex environments.

3.3 Manipulation

In the manipulation component, the focus is on developing a robust manipulation node for operating the Kinova robotic arm across various tasks. The MoveIt package, a motion planning framework, is utilized to implement key functionalities. The developed manipulation node is pivotal for enabling the Kinova robot's manipulation capabilities, facilitating precise control of the robot arm and interfacing with MoveIt for intuitive and efficient arm movement commands. The project employs advanced features in MoveIt, such as the Open Motion Planning Library (OMPL), which consists of state-of-the-art sampling-based motion planning algorithms. Specifically, the Probabilistic Roadmap Method (PRM) is used, a sampling-based algorithm that attempts to connect states to a fixed number of neighbors. PRM gradually increases the number of connection attempts as the roadmap grows, providing convergence to the optimal path. This ensures efficient and safe arm movements in complex manipulation tasks.

Furthermore, the node integrates constraints for the gripper to enhance performance. Using MoveItGoalBuilder's add-path-orientation-constraint method, orientation constraints are added to keep the arm upright. This is particularly important for tasks like picking up a bottle, where maintaining an upright orientation prevents the bottle from flipping over and causing placing failures. In addition to trajectory planning, the project incorporates advanced features like obstacle avoidance. The system uses an Intel D435 Depth Camera to generate an Octomap as in Figure 4, which is then utilized by MoveIt for collision detection and avoidance. This feature enhances the robot's versatility and capability in unstructured environments, aligning with real-world application demands.

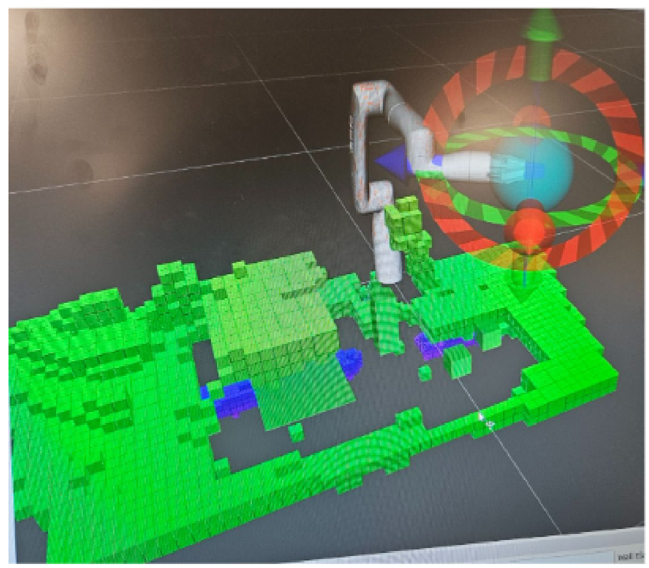

Figure 4: Octomap's 3D Grid for Collision Avoidance

The project also focuses on object detection and classification as in Figure 5, using a trained YOLO-V8 model (yolov8l-seg) with an impressive accuracy of approximately 95%. This model not only detects objects but also provides segmentation masks, bounding boxes, and centroids, which are crucial for accurate object manipulation. To precisely determine object positions, a transformation pipeline is implemented. It involves fixed positioning of an ArUco marker relative to the Kinova arm's base link, using static_transform_publisher to broadcast the transformation. Additionally, the system converts pixel coordinates from YOLO's object detection results into camera coordinates, considering depth information. This meticulous approach ensures accurate object localization for successful pick-and-place operations.

The effectiveness of these techniques is demonstrated in pick-and-place evaluations. With ground truth positions, the system achieves a 100% success rate for picking and placing socks, bottles, and blankets. However, errors in positional estimation by vision recognition can sometimes lead to failed "Pick" actions, highlighting areas for future improvement.

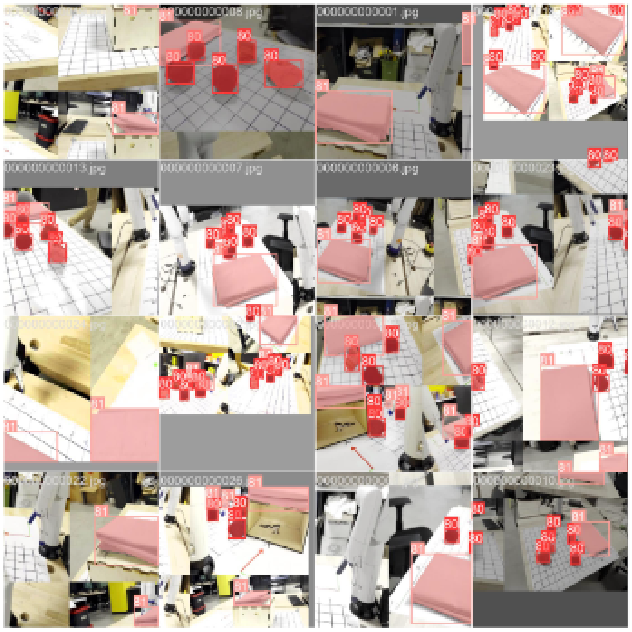

Figure 5: YOLO detection and classification

4 EXPERIMENT, RESULT, AND DISCUSSION

4.1 Experiment Setup and Procedure

Environment: The experiments were conducted in a simulated heathcare environment mapped from architectural plans and measurements of a physical space. This environment was filled with typical makerspace obstacles such as tables, chairs and various workstations, creating realistic navigational challenges. The space was expansive, crafted to simulate an active makerspace with dynamic elements like pedestrians intermittently blocking routes and other environmental variables, adding layers of complexity to the robot's navigation and task execution. The Kinova arm was setup at a desk by a hallway, with empty space next to the table as a loading dock.

Equipment Used: The setup included:

- **Kinova Arm:** Controlled by a server PC for precise manipulation tasks.
- **Create 2 Robot:** Connected to a Raspberry Pi 4 for navigation and transportation.
- **LiDAR Sensor:** Attached to the Raspberry Pi 4 for SLAM navigation.
- **Server PC:** Manages the Kinova Arm.
- **Laptop:** Controls the Create 2 via the Raspberry Pi 4.

Procedure: Robots were initialized and communicated through the InOrbit platform. The Create 2 used LiDAR for navigation, while the Kinova Arm performed pick-and-place tasks. Real-time monitoring via the InOrbit dashboard tracked task progress, with performance metrics recorded and analyzed.

4.2 Navigation Experiments

Navigation Experiment Process: The experimental procedure begins with the calibration of the robot's starting point on the map. The initial position is set at the coordinates (x = 16.07, y = -32.44), representing the location of the Kinova Arm. This ensures that the robot's navigation system has a reliable starting reference. Following the calibration, the first navigation task involves sending the robot to a fixed coordinate, specifically Destination 1, located at (x = 25.57, y = -25.95), representing a classroom. The robot is programmed to navigate autonomously to these coordinates. Its progress is monitored throughout the journey to confirm successful arrival at the target point. Once the robot successfully reaches Destination 1, the second navigation task commences. The fixed coordinates for Destination 2, which are the same as the starting point (x = 16.07, y = -32.44), are sent to the robot. The robot then navigates autonomously back to this location. Upon reaching each destination, the final error is determined by measuring the displacement from the fixed coordinates. The success rate is assessed based on the accuracy with which the robot reaches the designated points. This evaluation provides insight into the robot's autonomous navigation capabilities and precision.

Performance Metrics: Success Rate, Time Spent, Accuracy of Target Location

Parameters Tuning: In the experiment, the goal of Create 2 is to accurately reach a fixed target point every time, and at the position of the Kinova Arm, Create 2 needs to be as close as possible to the robotic arm. To achieve the above goals and improve the accuracy and success rate of Create 2 navigation, we need to experiment with different parameters to achieve the best navigation performance:

- xy_goal_tolerance: This parameter defines the tolerance of the robot's navigation process towards the target position. Specifically, it determines the allowable error range for the robot to reach the target position during navigation tasks. The smaller the parameter, the more accurate the relative target position of the robot after reaching the destination.
- yaw_goal_tolerance: yaw_goal_tolerance defines the allowable range of orientation error (in radians) for a robot to reach its target position. The smaller the parameter, the more precise the orientation will be when reaching the target position. However, counterintuitively, if this parameter is smaller, the robot will take a long time to adjust after reaching its destination, so this parameter should be adjusted larger.
- inflation_radius: infliction_radius is a key parameter used to set the radius of the obstacle expansion layer. The smaller this parameter, the closer the robot can approach the boundary of the obstacle.
- cost_scaling_factor: cost_scaling_factor is used in robotic navigation to regulate the growth rate of Cost Value in Inflation Layer.It defines how the value of the expansion area is attenuated from the edge of the obstacle.The lower Cost_Scaling_factor will increase the value of the generation, and the robot may choose the path closer to the obstacle.

Navigation Experiments Results: After multiple experiments, we have summarized the best performing parameter combinations as follows:

- costmap_common_params_burger.yaml: inflation_radius=0.2, cost_scaling_factor=4.0
- dwa_local_planner_params_burger.yaml: xy_goal_tolerance=0.1, yaw_goal_tolerance=0.2
- base_local_planner_params.yaml: xy_goal_tolerance=0.1, yaw_goal_tolerance=0.2

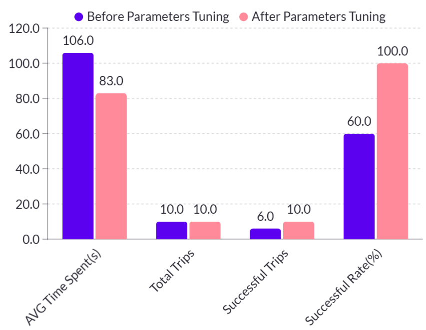

Figure 6: Comparison Experiment To Classroom Teaching Station(25.57, -25.95)

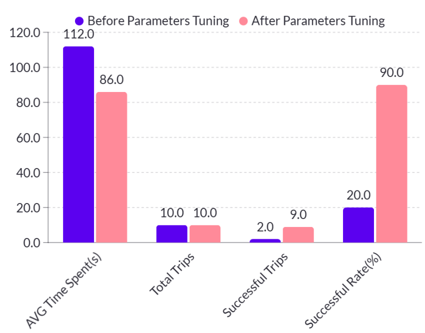

Figure 7: Comparison Experiment To Kinova Arm(16.07, -32.44)

It can be judged from Figure 6 and Figure 7 that the success rate of Create 2 navigation and the accuracy of reaching the destination have been significantly improved after parameter tuning. Especially, the success rate of going to Kinova Arm has increased from 20% before parameter tuning to 90%. At the same time, the time consumption of the entire trip has also been greatly reduced.

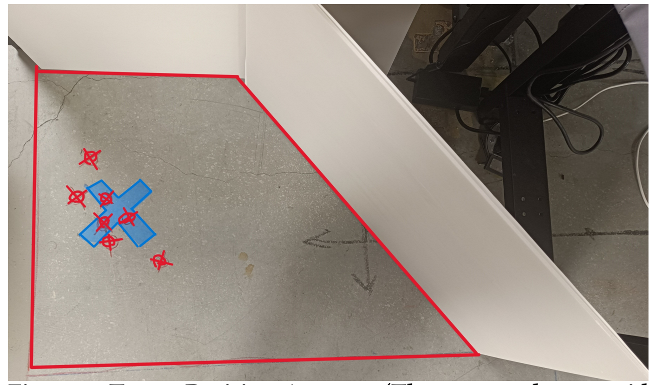

Figure 8: Target Position Accuracy(The outer red trapezoid represents the effective range of the robot's target position, the blue cross represents the accurate position of the destination, and the red cross represents the final position reached by Create 2 navigation in the complete travel test.)

According to Figure 8, the accuracy of Create 2 navigation was significantly improved during the full journey testing. After parameter tuning, the error of Create 2 in reaching the target position decreased to below 10cm, which is within an acceptable error range, and the arrival position of Create 2 is 100% within the effective range. This indicates that parameter tuning has a significant impact on improving the success rate and accuracy of navigation.

4.3 Manipulation Experiments

The experiment involved executing specific commands to evaluate the performance of the robotic system. These commands were structured as follows: <object1_name> <Quantity 1> <object2_name> <Quantity 2> ....

For this evaluation, three pairs of socks and three bottles were included within a single command to test the system's capability to handle multiple objects simultaneously. Additionally, the experiment included six instances of blankets to assess the system's performance with repeated actions. The success rate for the tasks is displayed in Table 1.

During the experiment, several challenges were encountered. One significant issue was the error in positional estimation caused by vision recognition. This error primarily affected the accuracy of the "Pick" action, leading to failures in object retrieval. Another challenge stemmed from center bias resulting from tilted bounding boxes. This bias introduced inaccuracies in the system's perception of object positions, affecting the overall performance. Moreover, instances where the blanket holder overlapped with the octomap posed a challenge for pick planning, leading to further failures in object manipulation.

4.4 Result and Discussion

Performance Metrics: The average task completion time for the multi-robot orchestration system was 6 minutes. The success rate for placing objects was 96%, with the system recording 4 errors primarily related to grip misalignment and navigation challenges.
Object Detection and Manipulation: The YOLO-V8 model used for object detection achieved a precision higher than 66%. The manipulation tasks demonstrated high accuracy, with the Kinova

Item	Pick Success Rate (%)	Pick Success Rate with GT position (%)	Place Success Rate (%)	Place Success Rate with GT position (%)
Socks	66	100	100	100
Bottle	100	100	100	100
Blanket	66	100	66	100

Table 1: Success Rates for Different Items

Arm successfully picking and placing socks, bottles, and blankets. However, positional estimation errors due to vision recognition occasionally led to failed "Pick" actions. This highlights the need for further refinement in the vision algorithms to improve reliability.

Navigation and Task Coordination: The ROS Navigation Stack, enhanced with a keepout map and parameter tuning, significantly improved the Create 2 robot's navigation accuracy and efficiency. The success rate for reaching target positions increased from 20% to 90% after parameter tuning, and the average error in reaching target positions decreased to below 10cm. This underscores the importance of parameter optimization in achieving reliable autonomous navigation.

5 CONCLUSION

The system demonstrates the potential of multi-robot orchestration in healthcare, enhancing operational efficiency and patient care. By integrating the Kinova Arm and Create 2 robot using the InOrbit operations management platform, we successfully coordinated item delivery tasks. InOrbit's real-time monitoring and task assignment capabilities ensured seamless communication between the robots. The ROS Navigation Stack, enhanced with a keepout map, provided safe and accurate navigation, while the MoveIt package enabled precise control of the Kinova Arm.

Our computer vision pipeline, using the YOLO-V8 model, achieved high accuracy in object detection, contributing to a 96% success rate in object placement tasks. Parameter tuning significantly improved navigation accuracy and task completion times.

Future work will focus on refining navigation and manipulation algorithms to further enhance the system's performance and adaptability in real-world healthcare environments. This includes integrating more sophisticated sensors and advanced machine learning techniques to improve object detection, obstacle avoidance, and overall system robustness. Additionally, exploring the standardization of communication protocols among different robots will be crucial for achieving seamless multi-robot interactions and scalability. Enhancing user interfaces and developing intuitive control mechanisms will also be key areas to address for improving human-robot interaction and overall system usability.

This scalable multi-robot management system highlights the potential for coordinating complex tasks across different robotic platforms in healthcare settings. It improves healthcare delivery efficiency and reduces the workload on providers, allowing them to focus on patient care.

REFERENCES

[1] Laurel D Riek. Healthcare robotics. *Communications of the ACM*, 60(11):68–78, 2017.
[2] George Mois and Jenay M Beer. The role of healthcare robotics in providing support to older adults: a socio-ecological perspective. *Current Geriatrics Reports*, 9:82–89, 2020.
[3] Feni Betriana, Ryuichi Tanioka, Joko Gunawan, and Rozzano C Locsin. Healthcare robots and human generations: Consequences for nursing and healthcare. *Collegian*, 29(5):767–773, 2022.
[4] Zhiqin Qian and Zhuming Bi. Recent development of rehabilitation robots. *Advances in Mechanical Engineering*, 7(2):563062, 2015.
[5] Caridad Mireles, Misael Sanchez, David Cruz-Ortiz, Iván Salgado, and Isaac Chairez. Home-care nursing controlled mobile robot with vital signal monitoring. *Medical & Biological Engineering & Computing*, 61(2):399–420, 2023.
[6] Katherine M Tsui, Munjal Desai, Holly A Yanco, and Chris Uhlik. Exploring use cases for telepresence robots. In *Proceedings of the 6th international conference on Human-robot interaction*, pages 11–18, 2011.
[7] Diligent Robotics. Moxi robot.
[8] ABB Robotics. Yumi robot.
[9] Dev Bahadur Poudel. Coordinating hundreds of cooperative, autonomous robots in a warehouse. *Jan*, 27(1-13):26, 2013.
[10] Sergey Trefilov and Yury Nikitin. Automatic warehouses with transport robots of increased reliability. *Acta Logistica*, 5(3):19–23, 2018.
[11] Abel Williams Gin Francis E. Chinda. Autonomous drones network with ai-enhanced coordination for disaster management. *ResearchGate*, 2023.
[12] F. De Pellegrini D. Miorandi, S. Sicari and I. Chlamtac. The internet of things: A survey. *Computer Networks*, 54(15):2787–2805, 2012.
[13] Mohammed Qader Kheder and Aree Ali Mohammed. Real-time traffic monitoring system using iot-aided robotics and deep learning techniques. *Kuwait Journal of Science*, 51(1):100153, 2024.
[14] Y. Zhao, H. Liu, and Z. Li. Research and implementation of autonomous navigation for mobile robots based on slam algorithm under ros. *Sensors*, 22(11):4172, 2022.