Midterm Report for Carrybox with Ros Vacuum Gripper Plugin

Jiyuan Luo[1] and Siqi Ye[2] and Huiwen Xue[3]

***Abstract*— This project aims to develop an automated carry box system equipped with a ROS (Robot Operating System) controlled vacuum gripper, designed to transport boxes from a designated source location to a target destination. The system employs a vacuum-based gripping mechanism to securely lift and handle various box sizes and weights, optimizing for stability and minimal slippage during transport. By integrating ROS, the project facilitates precise control and programming flexibility, enabling adaptable path planning and collision avoidance through sensor feedback. This approach streamlines tasks in warehouse, manufacturing, and logistics environments, providing a scalable solution for automating material handling.**

I. INTRODUCTION

The motivation for this project arises from the pressing need for flexible, intelligent automation in sectors such as logistics, warehousing, and manufacturing, where efficiency, cost reduction, and operational safety are top priorities. With the rise of e-commerce and the increasingly complex demands placed on supply chain infrastructure, there is a growing demand for systems that can seamlessly adapt to dynamic workloads and diverse handling requirements. Conventional material-handling technologies often lack the necessary flexibility to handle a variety of objects without complex reconfiguration, making it challenging to maintain throughput and reliability in high-mix environments. This project, which leverages a ROS-controlled vacuum gripper, aims to overcome these challenges by designing a system capable of securely grasping, lifting, and transporting a range of box sizes and shapes with minimal setup and consistent accuracy. The vacuum gripper's adaptability addresses the need for a universal handling tool, while the ROS framework enhances control, modularity, and scalability, allowing the system to evolve as requirements change. By integrating ROS, this project not only achieves efficient and precise box handling but also enables advanced functionality, including real-time obstacle detection, path planning, and collision avoidance, making it ideal for unpredictable environments where safety is paramount. Developing this project offers the opportunity to delve into key areas of robotics and automation, such as sensory feedback, path optimization, and control systems, providing a meaningful contribution to the field and valuable practical experience. Ultimately, the system promises to improve workflow efficiency, reduce labor dependency, and pave the way for more intelligent automation solutions in industries critical to the global economy.

The project utilizes RViz, a visualization software within ROS, to display the real-time state and movements of the robot arm managed by MoveIt. MoveIt serves as a motion planning framework that allows precise control and manipulation of the robot arm. Additionally, Gazebo, a powerful simulation platform, is used to simulate real-world physics, including factors such as friction, gravity, and collisions, creating a realistic environment for testing. By integrating Gazebo with MoveIt, this setup allows for joint simulation and control, enabling MoveIt to plan and control the robot arm's movements within Gazebo. This combined approach provides a robust and reliable platform for developing, testing, and refining robot arm movements in a simulated environment before deploying them in the real world.

II. RELATED WORK

In the context of robotic automation for material handling, several projects and studies have explored the application of robotic arms, vacuum grippers, and ROS frameworks, providing valuable foundations for this work. Vacuum grippers have been widely adopted in robotic applications for their flexibility and ability to handle a range of shapes and weights, particularly in logistics and manufacturing environments where adaptability and gentle handling are essential. Studies on vacuum-based gripping mechanisms have demonstrated their reliability in picking and placing items with minimal slippage. Furthermore, the use of ROS (Robot Operating System) for robot control has become a standard in robotics research due to its open-source nature and extensive suite of tools, which facilitate modular and scalable development. Projects that leverage ROS-controlled robotic arms for path planning and real-time obstacle avoidance have shown how ROS can enhance precision and adaptability in complex workflows, making it ideal for this project's aim of achieving intelligent, responsive handling of diverse payloads.

MoveIt, as a ROS-based motion planning framework, has also been explored in various research contexts, highlighting its effectiveness in managing robotic arm trajectories and collision avoidance through advanced path-planning algorithms. MoveIt's ability to interface seamlessly with sensor feedback systems enables real-time adaptability, a critical feature for applications requiring rapid, safe adjustments in unpredictable environments. In parallel, Gazebo is frequently utilized as a simulation platform to replicate realistic physics, offering an environment where the effects of friction, gravity, and collisions can be precisely modeled. Studies that combine MoveIt with Gazebo illustrate how simulation-driven testing can improve system robustness and performance, allowing researchers to fine-tune motion planning parameters and debug issues in a safe, virtual environment before real-world implementation. Together, MoveIt and Gazebo have been successfully deployed in various robotic automation projects to achieve high levels of control accuracy and operational safety, providing a comprehensive simulation and control ecosystem well-suited to this project's goals. Building on these existing works, this project seeks to contribute further by developing a ROS-controlled robotic system with a vacuum gripper specifically tailored to the needs of warehouse and logistics automation, offering an adaptive solution that can meet the demands of dynamic and complex handling tasks.

III. Method

A. How to obtain suction cup properties

Modeling suction-based gripping in robotic manipulation simulations introduces specific technical challenges, particularly in achieving realistic interaction dynamics. In this work, we detail the implementation of a suction gripper within the Gazebo simulation environment, utilizing the *gazebo ros vacuum gripper debugger* plugin, a robust, open-source solution that replicates essential suction-based gripping behaviors. This plugin was selected due to its capability to simulate key gripping characteristics, such as configurable force application and attachment behavior based on proximity—features critical for our robotic manipulation tasks.

The vacuum gripper plugin allows customization of essential parameters, including maximum gripping force, operational range, and attachment mechanics. These adjustable settings enable us to fine-tune the gripper's behavior, providing a more lifelike simulation of suction-based interactions that can be tailored to different manipulation scenarios. Integrating this plugin requires embedding it within a ROS (Robot Operating System) workspace, followed by compiling its source code to produce binary shared object (.so) files. These files enable Gazebo's plugin system to interface directly with the gripper functions, thus ensuring accurate simulation of the physical properties and behavior of the suction gripper.

This implementation provides a flexible and adaptable foundation for further optimization, allowing us to modify parameters to suit specific project needs. By leveraging this open-source tool, we create a scalable framework that supports detailed testing, future enhancements, and adjustments tailored to various robotic manipulation requirements. This approach improves both the realism and versatility of suction-based gripping simulations, facilitating more effective robotic manipulation research and development.

B. Suction Gripper Design

In the design of a suction gripper for simulation, our focus was on balancing functional fidelity with computational efficiency. While a detailed, physically precise model was achievable, such complexity would have introduced significant development overhead without proportional improvements in the simulation's operational accuracy.

Our approach employs a streamlined cylindrical geometry to represent the suction gripper, purposefully aligning with the UR10 robotic arm's cylindrical tool controller interface. This design choice facilitates a geometrically consistent connection with the UR10's end-effector while offering a sufficient contact area for effective gripping simulation. The cylindrical configuration provides several notable benefits:

1) Compatibility with the UR10's tool controller interface geometry
2) Simplified modeling of collisions and contact points
3) Reduced computational load within the physics engine
4) Reliable functional fidelity in gripping tasks

This optimized approach balances realism with implementation efficiency, allowing us to maintain the necessary gripping capabilities while minimizing the computational resources required. Such a design strategy reflects a practical trade-off, focusing on core manipulation functionality while ensuring that simulation performance remains robust.

The combination of this streamlined gripper geometry and the previously implemented vacuum gripper plugin yields a highly efficient framework for simulating suction-based manipulation. This configuration achieves essential functional accuracy, optimizes simulation performance, and conserves development resources. The effectiveness of this design choice will be further validated in the experimental results section, where we demonstrate its performance in real-world manipulation scenarios.

C. Design of Simulation Environment

In response to feedback from our project proposal, we recognized the need to construct a custom simulation environment. To streamline development, we opted for an initial design—*World Version 1*—comprised of three basic box structures. These boxes are positioned to represent key locations relevant to our manipulation tasks, allowing us to simulate essential interactions and validate our robotic functions effectively.

This simplified environment serves as a practical starting point, reducing initial setup complexity while providing a flexible foundation for further adjustments. We will demonstrate the implementation and performance of *World Version 1* in the experimental section, where its design will support the validation of our gripping and manipulation approaches in various task scenarios.

IV. Experiments

A. Connecting the Cylindrical Suction Gripper to the Robotic Arm

Initially, our setup did not include an offset, resulting in a visible gap between the suction gripper and the robotic arm, as shown in Figure 1. Through further research, we determined that the suction gripper's coordinate frame is located at the center of the cylindrical structure. Given our gripper length of 0.05m, the coordinate frame effectively bisects the cylinder, indicating that we needed an offset of -0.025m along the Z-axis to align the gripper correctly.

After adjusting the offset accordingly, the gripper and arm aligned seamlessly. The chosen gripper length of 0.05m was derived experimentally; our initial length was overly large, causing the gripper to clip through the arm. For brevity, we have omitted images of these initial overlapping issues.

This fine-tuned alignment allows for a stable connection between the gripper and robotic arm, facilitating accurate simulations in subsequent experiments.

B. Creation of the Simulation Environment

Initially, we constructed our simulation environment by directly manipulating rectangular prisms within the simulation software through drag-and-drop operations, allowing us to stretch and rotate the objects as needed. This foundational setup enabled us to establish a basic world structure.

Building upon this, we modified the world file to assign different colors to the objects, providing clearer visual distinctions and enhancing the overall representation. This approach also allowed us to gain more precise control over the attributes of each object, which is not possible through the standard drag-and-drop interface alone.

In our first version of the simulation environment, we defined the following components:

- Red Box: Serves as the destination for transporting the boxes.
- Blue Box (Box0): Acts as the base for the robotic arm.
- Green Box (Box1): Represents the conveyor belt where the boxes are initially placed.

This carefully crafted environment sets the stage for further experimentation and testing of our robotic manipulation capabilities.

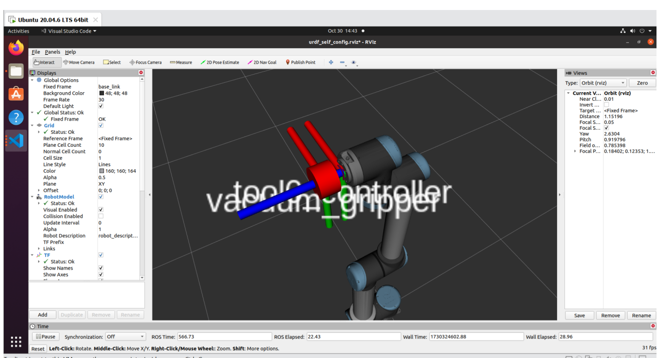

Fig. 1.

V. Next Steps

The next steps for this project focus on enhancing the simulation environment and implementing object-grasping functionality. Currently, the simulated world is quite basic, so optimizing and refining it is a priority. We plan to create realistic box models and adjust the size and properties of the existing three boxes to better reflect their intended roles. Additionally, we will ensure that the robotic arm is securely mounted on its base.

Another critical task is to develop motion planning for the robotic arm, potentially utilizing MoveIt to facilitate this process. Our goal is to enable the robotic arm to detect, approach, and securely grasp target objects, allowing it to

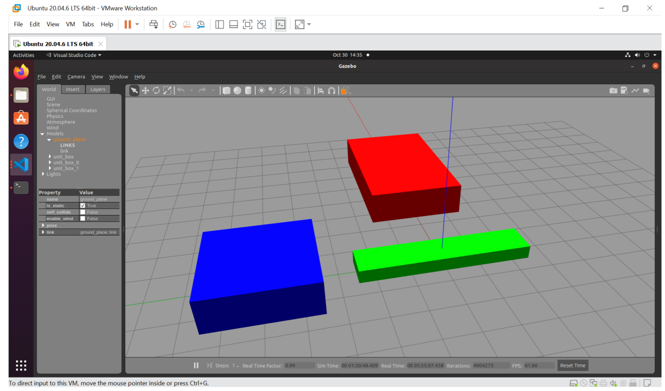

Fig. 2.

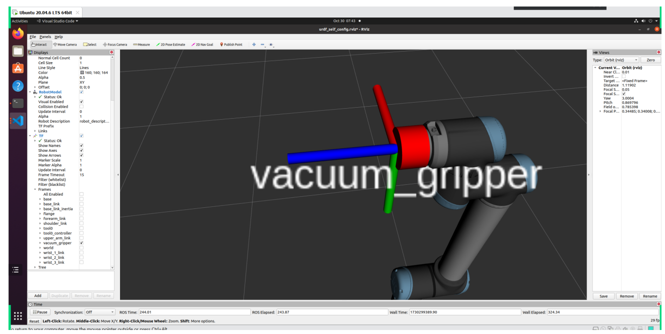

Fig. 3.

pick up boxes from one location and place them at a designated destination. This will require further development and fine-tuning of the suction gripper's parameters and control algorithms to ensure stable and reliable manipulation.

By addressing these enhancements, we aim to create a more realistic and comprehensive testing ground for evaluating the robotic arm's performance. These improvements will bring the system closer to a practical solution for automated material handling applications in warehouse and logistics environments.

References

[1] Gazebo ROS Vacuum Gripper Debugger GitHub repository. Available: `https://github.com/tatsuya-s/gazebo_ros_vacuum_gripper_debugger`. Accessed: Nov. 1, 2024.

[2] ROS-Industrial Universal Robot GitHub repository. Available: `https://github.com/lihuang3/ur5_ROS-Gazebo`. Accessed: Nov. 1, 2024.