POLITECNICO MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE E DELL'INFORMAZIONE

Executive Summary of the Thesis

Development of an Autonomous Mobile Manipulation Robot for Industrial and Agricultural Environments

Laurea Magistrale in Computer Science Engineering - Ingegneria Informatica

Author: Simone Giampà

Advisor: Prof. Matteo Matteucci

Co-advisor: Dr. Gianluca Bardaro

Academic year: 2023-2024

1. Introduction

The rapid advancement of robotics has opened up possibilities for complex automation in various fields. While traditional applications focus on either mobile navigation or stationary manipulation, there's a growing interest in merging these capabilities into mobile manipulation systems. However, the integration of navigation and manipulation poses significant challenges due to the complexity and uncertainty of dynamic environments.

This thesis project aims to tackle these challenges by developing a mobile manipulation system capable of operating autonomously in both industrial and agricultural settings. The system leverages existing robotic platforms, a mobile robot equipped with a LiDAR sensor, and a robotic arm with a camera for perception. The project focuses on creating software components that enable the robots to perform high-level tasks, including navigation, object detection, manipulation, and task planning. The final goal is to demonstrate the system's capabilities through two real-world scenarios: interacting with a control panel in an industrial environment and picking fruits from a tree in an agricultural environment.

2. State of the Art and Literature Review

Robotic manipulator control has evolved from traditional open-loop and exteroceptive feedback methods to encompass advanced software-based solutions powered by deep learning. This evolution aims to address the growing complexity of tasks, particularly in dynamic and unstructured environments. Two major approaches dominate the field: model-based and data-driven.

Model-based approaches, exemplified by Model Predictive Control (MPC) and Inverse Kinematics (IK) solvers, rely on precise mathematical models of the robot's dynamics and kinematics. These methods excel in structured environments with well-defined tasks, offering interpretability and predictable behavior. However, their performance often degrades in complex scenarios due to the difficulty of modeling real-world uncertainties.

Data-driven approaches, on the other hand, leverage machine learning techniques, such as Deep Reinforcement Learning (DRL) [4] and imitation learning [2], to learn control policies directly from data. This allows robots to adapt to

diverse and unpredictable environments, but at the cost of reduced interpretability and potential for unsafe behavior if not properly trained.
Mobile manipulation, combining navigation and manipulation tasks, presents additional challenges. Traditional methods, often relying on heuristics and switching layers, struggle with complex coordination and dynamic environments. Deep reinforcement learning shows promise in tackling these challenges, but issues like sample efficiency, generalization, and safety remain critical concerns.
The concept of whole-body control, treating the mobile manipulator as a unified system, has gained traction for its potential to achieve dynamic and agile behaviors. Researchers have explored both model-based (MPC+IK) [3] and data-driven (DRL) approaches [1] for whole-body control, demonstrating successful manipulation of articulated objects and navigation in complex environments.
The simulation-to-reality gap is a major obstacle in deploying DRL-based solutions. Tools like Nvidia Isaac Gym and Sim-to-Real pipelines are being developed to bridge this gap by creating realistic simulations and addressing domain randomization.
Object detection and grasping are essential components of mobile manipulation. While various approaches exist, including those based on scene understanding, hand-eye coordination, template matching, and deep learning, each has its limitations. Grasping soft objects poses a particular challenge due to their deformable nature and the difficulty of modeling their behavior.
The current state of the art in robotic manipulator control is a rapidly evolving landscape, with both model-based and data-driven approaches playing crucial roles. The choice between these approaches often depends on the specific application and the trade-offs between interpretability, adaptability, and safety. The future of mobile manipulation lies in integrating the strengths of both paradigms, along with advancements in perception, planning, and control, to create more versatile and capable robotic systems.

3. Robotic Platform for Mobile Manipulation

The mobile manipulation platform used in the project consists of an AgileX Scout 2.0 mobile robot and an Igus ReBeL 6-DoF collaborative robotic arm. The Scout 2.0, provides a robust mechanical base with ample payload capacity, while the lightweight and cost-effective ReBeL cobot offers flexibility for mounting and operation. An Intel NUC 12 computer onboard the mobile robot serves as the central control unit, running software and perception algorithms. A TP-Link router and Netgear switch facilitate remote control and monitoring, enhancing safety and troubleshooting capabilities.
The system incorporates two primary sensors: an Ouster OS1-64 LiDAR for navigation and mapping, and an Intel RealSense D435 RGB-D stereo camera mounted on the robotic arm for object detection and manipulation. The LiDAR's high resolution and scan rate enable accurate localization and obstacle avoidance, while the RGB-D camera facilitates object detection and grasping tasks using both color and depth information. The soft gripper, a pneumatic actuator from Soft Gripping with silicone rubber fingers, enables delicate object manipulation. Controlled by a pneumatic pump and an Arduino UNO microcontroller with relays, the gripper can adapt to various object shapes and sizes.
3D-printed mounts, designed using Fusion 360 and printed with PETG material, ensure secure attachment of the camera and gripper to the robotic arm. These lightweight and durable mounts maintain sensor positioning despite vibrations during operation. The initial mount design (MountV1), shown in Figure 1, was robust but limited the camera's field of view. An improved version (MountV2), shown in Figure 2, was developed to address this issue, optimizing camera placement and incorporating the soft gripper.
Power management involves DC/DC converters to power the onboard computer and sensors from the mobile robot's internal battery. External lead batteries power the robotic arm and pneumatic pump, with Molex connectors enabling easy switching between battery packs and the external power supply.
The ReBeL cobot presented challenges due to

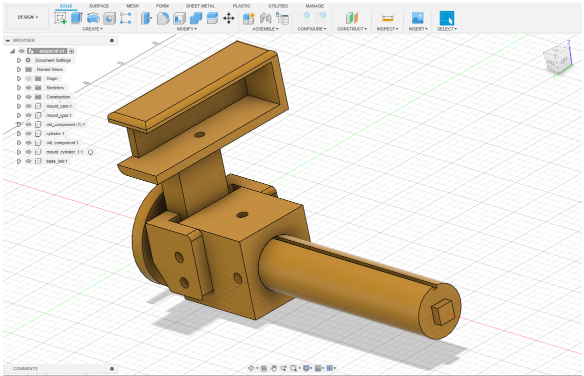

Figure 1: MountV1 with cylinder presser

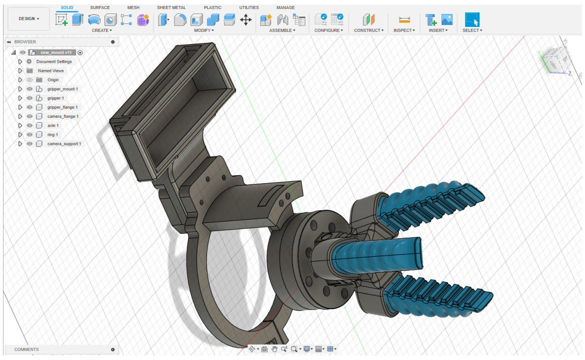

Figure 2: Mountv2 with soft gripper

Figure 3: Mobile Manipulator Robot Setup

its low-cost design. The motor encoders lacked accuracy and repeatability, and the plastic gears exhibited backlash and deformation under load. These issues were addressed with artificial error compensation and careful calibration procedures. Additionally, the Intel RealSense camera required calibration using the manufacturer's software to ensure accurate depth estimation and object localization.
Overall, this mobile manipulation platform represents a functional and adaptable solution for various applications in industrial and agricultural settings. The integration of diverse hardware and software components, combined with rigorous testing, demonstrates the system's potential for real-world automation tasks. Figure 3 shows a picture of the complete setup of the mobile manipulation robot.

4. Software Architecture and Control

The system leverages the Robot Operating System 2 (ROS2) as the middleware, providing a flexible framework for integrating diverse software components and enabling communication between nodes.

4.1. Robotic Arm Control

A key aspect of the software development was creating a ROS2 control interface for the Igus Rebel arm, enabling high-level control and monitoring through the Ethernet interface. This interface interacts with the Joint Trajectory Controller, which receives motion plans from MoveIt2 and translates them into joint positions or velocities for the arm's motors. The implementation relies on the ROS2-Control framework and a custom hardware interface designed for the Igus Rebel arm.
MoveIt2, a powerful motion planning framework, plays a central role in planning and executing arm trajectories. Its modular architecture, consisting of components like the Planning Scene, Planning Pipeline libraries, and Kinematics Solver, facilitates seamless integration with ROS2. A custom ROS2 library simplifies the use of MoveIt2 for the Igus Rebel arm, offering high-level functions for planning and executing various types of motion.
The RViz2 visualization tool allows for real-time visualization of the robot's motion in both simulated and real environments. It aids in de-

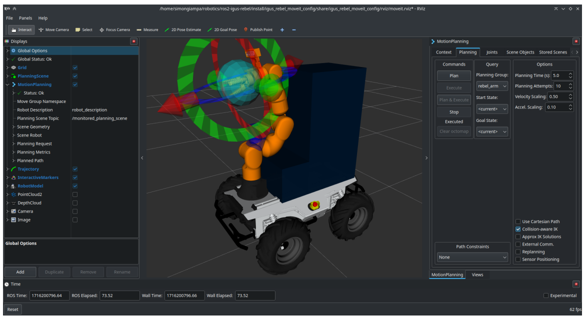

Figure 4: RViz2 visualization of MoveIt2 planning scene

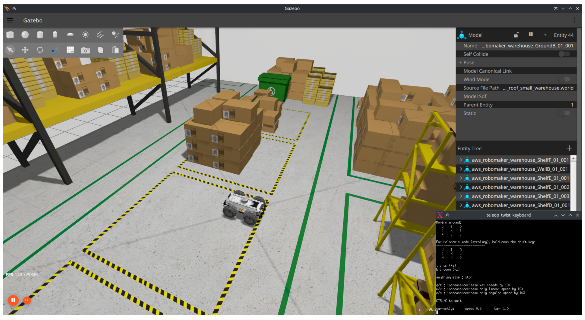

Figure 5: Ignition Gazebo Simulation Environment for the mobile robot base

bugging and validating the functionality of the software components, including the kinematic model, planning scene, and motion plans, as shown in Figure 4.

To address the challenges of collision avoidance in dynamic environments, the Octomap library is integrated with MoveIt2. Octomap generates 3D occupancy maps of the surroundings, enabling the robot to avoid obstacles. However, due to the ongoing development of the Octomap library for ROS2, limitations such as slow updates and false positives in collision checking were encountered.

Actuation of the soft gripper is achieved through a ROS2-control interface that communicates with an Arduino UNO microcontroller. This microcontroller controls the pneumatic pump responsible for opening and closing the gripper's fingers. Serial communication over the UART protocol facilitates this control, allowing for precise actuation of the soft gripper during manipulation tasks.

4.2. Mobile Robot Control

The Ignition Gazebo simulation environment serves as a crucial tool for testing and refining the robot's navigation and manipulation capabilities. A realistic warehouse environment with various obstacles was created to validate the performance of the Nav2 navigation framework and associated algorithms in a safe and controlled setting, as shown in Figure 5.

Nav2, a comprehensive software framework for autonomous navigation, is utilized for planning and executing the mobile robot's trajectories. Its modular architecture accommodates various plugins, including costmap layers, localizers, planners, and recovery behaviors. SLAM Toolbox is employed for mapping and localization, while the Hybrid A* global planner and MPPI local planner ensure efficient and collision-free navigation.

Parameter tuning in Nav2 was a significant undertaking, requiring extensive testing in both simulated and real environments to achieve optimal performance. Adjustments were made to localization algorithms, global and local planners, recovery behaviors, and costmap configurations to ensure safe and reliable navigation in diverse scenarios.

A notable challenge encountered during testing was the low and unreliable frequency of the LiDAR sensor. This issue, stemming from a combination of network congestion and the default DDS middleware configuration, was resolved by reconfiguring the middleware to prioritize local communication and adopting the Cyclone DDS implementation. This optimization significantly improved data transmission and overall system stability.

A parking algorithm was designed to enable the mobile robot to autonomously position itself optimally for manipulation tasks. The algorithm considers the target location, costmap, and robot footprint to generate candidate poses and select the most suitable one based on a ranking function. Despite its effectiveness in most cases, the algorithm's limitations, primarily related to the robot arm's workspace approximation, highlight potential areas for future refinement.

4.3. Perception

ArUco markers are utilized for precise object localization and pose estimation. Detection involves identifying the marker's corners and ID using the OpenCV ArUco library, while pose es-

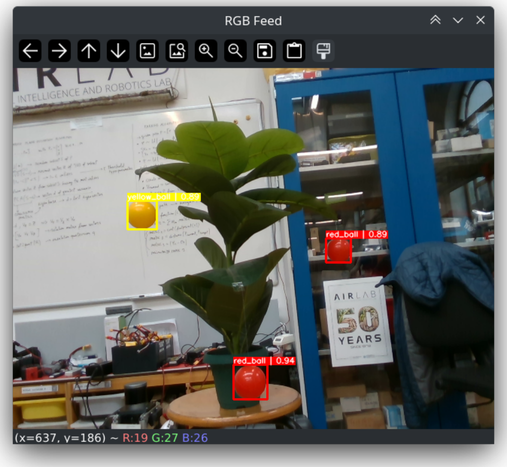

Figure 6: YOLOv8 Inference in action

timation utilizes these corners, marker size, and camera intrinsic parameters to determine the marker's position and orientation relative to the camera.
To overcome challenges in estimating the orientation of small ArUco markers, a multi-ArUco plane estimation algorithm was developed. This algorithm assumes the coplanarity of multiple markers and utilizes RANSAC, least squares estimation, PCA, and SVD to determine the plane's orientation, thus refining the individual marker poses.
The YOLOv8 object detection model, a state-of-the-art real-time algorithm, was employed for detecting colored balls and apples in the environment, as demonstrated in Figure 6. Trained on a custom dataset with augmented images, YOLOv8 achieved satisfactory performance in terms of mean Average Precision (mAP) and Intersection over Union (IoU). Challenges with high false positive rates were addressed by adjusting the confidence threshold.
The object's pose estimation process combines object detection with depth perception from the RealSense camera. A segmented point cloud is created by filtering the depth map based on the detected object's bounding box and color mask. The object's center is estimated by fitting a sphere to the segmented point cloud using RANSAC and MLESAC algorithms, enabling the calculation of an optimal grasping pose for the robotic arm.
Two algorithms, sphere center estimation and grasp pose estimation, facilitate the manipulation of objects. The former estimates the object's center from the segmented point cloud, assuming a spherical shape, while the latter generates candidate grasping poses and selects the most suitable one based on inverse kinematics feasibility and a heuristic ranking. This approach is simplified due to the compliance of the soft gripper fingers, allowing for some error in pose estimation.

4.4. Actions Architecture

The ROS2 Actions client-server architecture provides a modular and scalable framework for implementing high-level tasks. It enables efficient communication and coordination between different software components, ensuring robust and reliable execution of complex behaviors like autonomous navigation, object detection, grasping, and manipulation.
The integration of ROS2, MoveIt2, Nav2, and other components has resulted in a capable and adaptable system ready for deployment in real-world scenarios. Future work could focus on improving the integration of Octomap with MoveIt2, refining the grasping algorithms for the soft gripper, and exploring more advanced perception techniques to enhance the robot's capabilities further.

5. Experimental Setups and Demonstrations

Experimental setups and demonstrations were designed to showcase the capabilities of the mobile manipulation robot in industrial and agricultural scenarios. Three distinct demonstrations were conducted: the ArUco Follower, the Button Presser, and the Object Picking demos.
The experimental setups involved both simulated and real-world scenarios. A simulated warehouse environment in Ignition Gazebo was used to test and validate the navigation and obstacle avoidance algorithms. Additionally, artificial plants with colored balls and apples served as realistic environments for testing the object picking capabilities of the mobile manipulator.
The ArUco Follower demo served as a preliminary test for the robotic arm's autonomous control software. Utilizing the MoveIt2 motion planning framework and ArUco marker detection and pose estimation algorithms, the robot

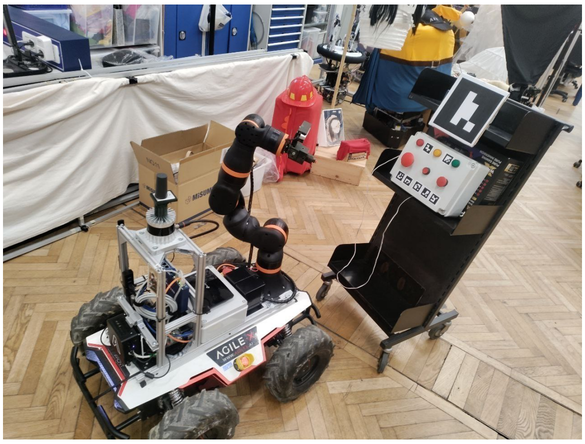

Figure 7: Mobile Button Presser Demo

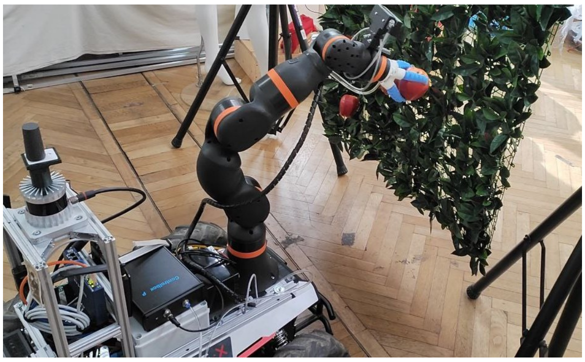

Figure 8: Mobile Fruit Picking Demo: Object Picking Version 2 while picking apples from an artificial plant

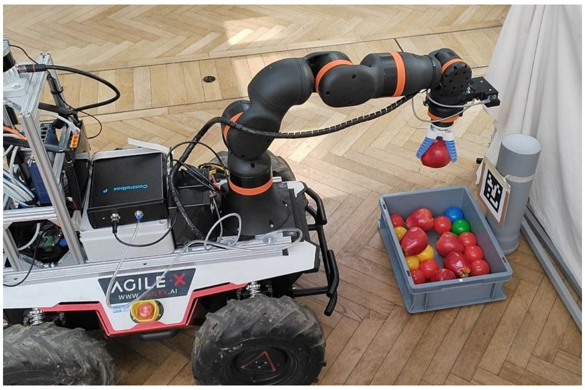

Figure 9: Mobile Fruit Picking Demo: Object Picking Version 2 while dropping a picked apple

arm successfully tracked and followed a moving ArUco marker with its end effector. However, attempts to integrate the MoveIt2 Servo node for real-time control proved challenging due to stability issues and the need for precise PID tuning.

The Button Presser demo highlighted the robot's potential in industrial settings by demonstrating its ability to autonomously press buttons on a custom-designed control panel. Two iterations of the end-effector mount were employed, with MountV2 incorporating a vacuum suction cap to enhance button-pressing reliability. The demo involved a sequence of steps, including navigating to the control panel, detecting ArUco markers with the multi-ArUco plane estimation algorithm to locate the buttons, and executing planned trajectories to press them in a predefined order. Figure 7 shows the robot while interacting with the control panel.

The Object Picking demo showcased the robot's adaptability in agricultural scenarios, specifically focusing on picking colored balls and apples from artificial plants. The demo utilized the YOLOv8 object detection model, trained on a custom dataset, to identify and locate objects. Two versions of the Object Picking demo were implemented. In the first version, the robot picked objects and placed them in a basket mounted on the mobile base. The second version involved navigating to a separate dropping location, requiring the robot to utilize its navigation and obstacle avoidance capabilities. Both versions successfully demonstrated the robot's ability to autonomously pick and place objects in a simulated agricultural environment. Figure 8 shows the robot picking apples from an artificial plant, while Figure 9 depicts the robot dropping a picked apple in the designated location.

The object picking routine, common to both versions, involved generating a list of waypoints for the robot to follow, detecting objects at each waypoint, computing priority scores for detected objects based on distance and confidence, estimating object centers using a RANSAC sphere fitting algorithm, and calculating grasping poses. The robot then executed trajectories to approach, grasp, and transport the object to the designated dropping location.

The implementation of these demos relied heavily on the ROS2 framework, leveraging its action client-server architecture for high-level task coordination. The system architecture encompassed various components, including MoveIt2 servers for planning and execution, Nav2 servers for navigation, a client node for orchestration, a neural network node for object detection, and an ArUco pose estimation node.

Throughout the development and testing of these demos, several challenges were encountered. These included dealing with the infrared reflectivity of objects, the slow inference speed of the object detection network, and the limitations of the Octomap library for collision avoidance. Workarounds and optimizations were implemented to mitigate these issues, demonstrating the resilience and adaptability of the robotic system.
Overall, these experimental setups and demonstrations showcase the potential of the mobile manipulation robot for autonomous operation in both industrial and agricultural settings. They highlight the successful integration of diverse software components, including perception, motion planning, grasping, and navigation, into a unified system capable of performing complex tasks. The results of these experiments provide valuable insights for future research and development in mobile manipulation robotics, paving the way for more sophisticated and versatile autonomous systems.

6. Conclusions

This thesis has successfully demonstrated the feasibility of developing a cost-effective, autonomous mobile manipulation robot for industrial and agricultural applications. By integrating a mobile platform with a robotic arm and utilizing ROS2, Nav2, and MoveIt2 frameworks, the system achieved autonomous navigation, object manipulation, and task execution in dynamic environments. A key contribution was the design and implementation of a pneumatic soft gripper, showcasing its potential for delicate object handling.
Through rigorous testing in simulated and real-world scenarios, the project highlighted the importance of robust perception, localization, and sensor fusion in achieving reliable performance. The modular software architecture and comprehensive documentation facilitate future research and development, enabling further refinement and extension of the system's capabilities.
This research lays the groundwork for advancing mobile manipulation robotics. Future endeavors could focus on enhancing perception algorithms, exploring deep learning techniques for object recognition and grasping, and integrating ROS2 behavior trees for more complex task planning. Ultimately, this project contributes to the vision of creating adaptable and intelligent robotic systems that can revolutionize industries by automating complex tasks and improving efficiency and safety.

References

[1] Zipeng Fu, Xuxin Cheng, and Deepak Pathak. Deep whole-body control: Learning a unified policy for manipulation and locomotion. In *Conference on Robot Learning (CoRL)*, 2022.

[2] Zipeng Fu, Tony Z. Zhao, and Chelsea Finn. Mobile aloha: Learning bimanual mobile manipulation with low-cost whole-body teleoperation. 2024.

[3] Mayank Mittal, David Hoeller, Farbod Farshidian, Marco Hutter, and Animesh Garg. Articulated object interaction in unknown scenes with whole-body mobile manipulation. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1647–1654, 2022.

[4] Cong Wang, Qifeng Zhang, Xiaohui Wang, Shida Xu, Yvan Petillot, and Sen Wang. Multi-task reinforcement learning based mobile manipulation control for dynamic object tracking and grasping. In *2022 7th Asia-Pacific Conference on Intelligent Robot Systems (ACIRS)*, pages 34–40, 07 2022.