

1 | State of the Art and Literature Review

This chapter will present the state-of-the-art and literature review of the topics related to this project. The topics are: Robotic Manipulator Control, Deep Reinforcement Learning in Robotic Manipulation and Mobile Manipulation, Autonomous navigation, Object Detection and Grasping.

Particular focus is on the part regarding Mobile Manipulation since it is the main topic of this thesis project. In particular, the potential challenges as well as possible benefits and disadvantages of using each method will be discussed.

1.1. Robotic Manipulator Control Approaches

Currently, the control sequence of a robotic manipulator is mainly achieved by solving inverse kinematic equations to position the end effector with respect to the fixed frame of reference. Robots can be controlled in an open loop or with exteroceptive feedback. The **open-loop control** does not have external sensors or environment sensing capability but heavily relies on highly structured environments that are very sensitively calibrated. Under this strategy, the robot arm follows a series of positions stored in memory and goes through them at various times in their programming sequence. In more advanced robotic systems, **exteroceptive feedback control** (closed loop system) is employed, through the use of monitoring sensors, force sensors, even vision or depth sensors, that continually monitor the robot's axes or end-effector, and associated components for position and velocity. The feedback is then compared to information stored to update the actuator command to achieve the desired robot behavior. Either auxiliary computers or embedded microprocessors are needed to interface with these additional sensors and to perform the required computations. These two traditional control scenarios are both heavily dependent on hardware-based solutions [10].

Other control strategies may include **robotic embodiment** for **imitation learning**.

Robotic embodiment, in the context of imitation learning, is a control strategy that is based on the idea that the robotic system emulates the human body movement, to learn a task quickly, instead of relying on specific ad-hoc training and programming, as suggested in [9]. This article presents an approach to the autoprogramming of robotic assembly tasks with minimal human assistance. The approach integrates "robotic learning of assembly tasks from observation" and "robotic embodiment of learned assembly tasks in the form of skills". The aim of these skills is to let robots execute difficult tasks that involve inherent uncertainties and variations and are most useful in smart manufacturing in industrial scenarios. The robotic embodiment is associated with a dramatic reduction in the human effort required for automating robotic assembly, as well as task training.

With the advancements in modern technologies in artificial intelligence, such as deep learning, and recent developments in robotics and mechanics, both the research and industrial communities have been seeking more software-based control solutions using low-cost sensors, which have fewer requirements for the operating environment and calibration. The key is to make minimal but effective hardware choices and focus on robust algorithms and software. Instead of hard-coding directions to coordinate all the joints, the control policy could be obtained by learning and then be updated accordingly. **Deep Reinforcement Learning (DRL)** is among the most promising algorithms for this purpose because it ideally suits complex robotic manipulation and control tasks in dynamic and unstructured environments, or when the task is too complex to be explicitly programmed. A reinforcement learning approach might be trained on a dataset of experiential data, such as input data from a robotic arm experiment, with different sequences of movements, or input data from simulation models. Either type of dynamically generated experiential data can be collected and used to train a Deep Neural Network (DNN) by iteratively updating specific policy parameters of a control policy network [10].

Robotic control approaches can be broadly categorized into **model-based approaches**, such as the ones using a Model Predictive Controller (MPC) and Inverse Kinematics (IK) computation, and **model-agnostic approaches**, often characterized as **data-driven methods**, including Deep Reinforcement Learning (DRL) and other machine learning techniques.

- **Model-based approaches** rely on explicit models of the robot's dynamics or kinematics to formulate control strategies. MPC optimizes control inputs over a prediction horizon based on the system's dynamics and constraints, while IK determines joint configurations to achieve desired end-effector poses.
- **Model-agnostic approaches** learn control policies directly from data through

interactions with the environment. These data-driven methods leverage neural networks to map observations to actions, allowing robots to adapt to complex and dynamic scenarios without requiring an explicit model.

The main differences lie in the reliance on explicit models in model-based methods, providing **transparency and interpretability**, **versus** the model-free nature of data-driven methods, offering **flexibility and adaptability** to diverse and evolving environments. Integrating these approaches can harness the strengths of both paradigms, combining the precision of model-based control with the adaptability of data-driven learning for enhanced robotic control capabilities in multiple scenarios and tasks.

An issue raised by the real-world application is the safety of the system while sharing the workspace with human workers. Identifying and, more importantly, certifying methods to collaborate with humans in the workspace in a safe way are key points for bringing autonomous mobile robots to real industrial applications.

The following paragraphs will describe the available methods used for robotic manipulator control.

1.1.1. Mobile Manipulation Tasks

Mobile manipulators that combine base mobility with the dexterity of an articulated manipulator have gained popularity in numerous applications ranging from manufacturing and infrastructure inspection to domestic service. Deployments span a range of interaction tasks with the operational environment from minimal interaction tasks, such as inspection, to complex interaction tasks such as logistics resupply and assembly. This flexibility, offered by the redundancy, needs to be carefully orchestrated to realize enhanced performance. Thus, advanced decision-support methodologies and frameworks are crucial for successful mobile manipulation in (semi-) autonomous and teleoperation contexts [20]. Given the enormous scope of the literature, I restrict my attention to decision-support frameworks specifically in the context of wheeled mobile manipulation.

As a quick aside, a disambiguation is necessary between the often interchangeably used "**motion planning**" and "**path planning**". Although path planning only generates a path within the configuration space, motion planning generates time-indexed motion trajectories. Instead path-following only requires spatial feasibility (e.g., obstacle avoidance), while motion planning requires compatibility with spatiotemporal constraints (engendered in the dynamics of both robot and environment). It is also noteworthy that ultimately any path planning effort requires a final time parameterization into a motion planning exercise before deployment [20].

The combined controllable degrees of freedom within the kinematic chain (from both mobile base and the articulated manipulator) presents the mobile manipulator design architecture the opportunity to address very complex tasks. However, resolving the redundancy (internal/external) is crucial to realizing this potential. As the complexity of the overall mobile manipulation process increases, a **two-stage hierarchical approach** is often pursued:

1. task planning/breakdown into a series of tractable motion planning subtasks and their sequencing
2. motion planning of the high degree-of-freedom mobile manipulator within each sequenced task

It is noteworthy that the two steps (task planning and motion planning) are closely coupled and should be solved concurrently but are addressed separately from a computational tractability perspective [20].

1.2. Traditional Control Approaches

However, a breakdown along the lines of mobile manipulator subsystems (mobile base versus manipulator versus gripper or combinations) or along the nature of the manipulation task (transportation versus grasping) feature prominently in the literature. The task-level and motion-level planning frameworks can be viewed as a form of "artificially constrained" motion planning within a higher dimensional space.

The first applications for mobile manipulators were in the field of logistics, where the mobile base was used for transportation and the manipulator for grasping and placing objects. Traditional control approaches rely on a series of heuristics to solve the problem of navigation, grasping, and placing objects. The mobile base is controlled by a multitude of algorithms, such as SLAM, AMCL and DWA for navigation, while the manipulator is controlled by lower-level motor controllers or trajectory planners. The integration between the mobile base and the manipulator is often done by a **switching layer** that determines the currently pertinent control objectives. Traditional methods often do not handle multiple and different control objectives at the same time, so the robot divides one high-level task into multiple sub-tasks executed in a sequence. This approach requires a lot of engineering effort to coordinate the arm and the base movements, and often fails in complex tasks where the decision-making process is hard.

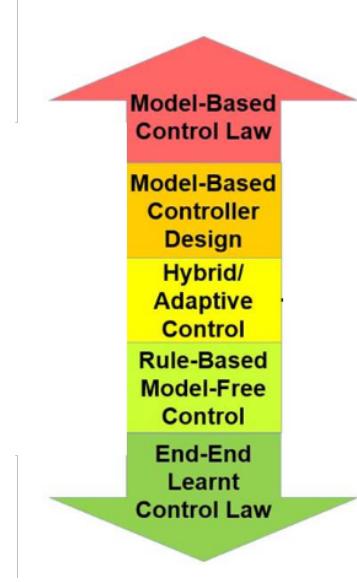


Figure 1.1: The continuum in the literature in regards to control methodology ranging from model-based to end-end data-driven control [20]

point for mobile manipulation is the work [18], presenting the winning mobile manipulation system for the *Mohamed Bin Zayed International Robotics Challenge (MBZIRC)* held in 2020. The proposed system is comprised of a mobile wheeled base performing localization and navigation in a semi-structured environment, and a 5-DoF manipulator for grasping and precise placement of bricks in a carrier. This work was among the first to demonstrate the potential of mobile manipulation in a practical scenario.

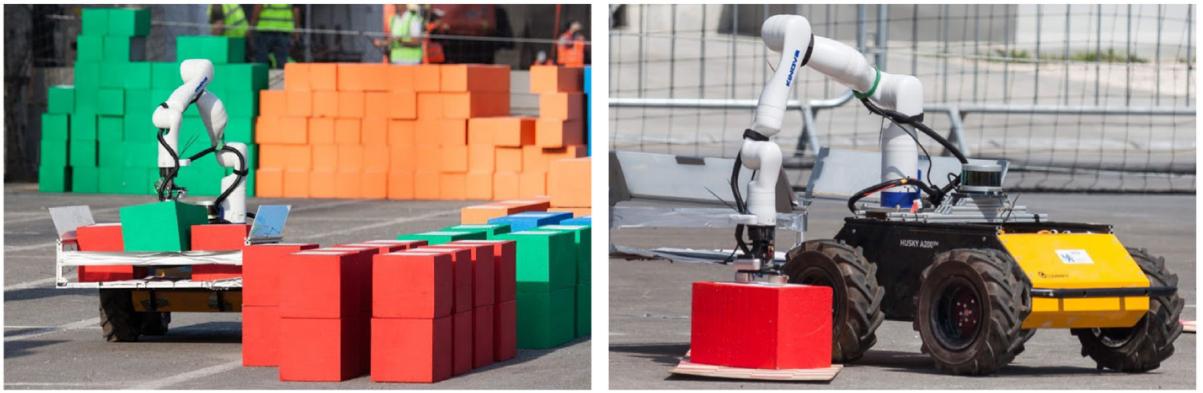


Figure 1.2: The described system loading and placing building material during the MBZIRC 2020 contest. [18]

However, their approach is based on many simplifying assumptions which may be suitable as a first one-of-a-kind robotic pick-and-place system in a real-world scenario and

application but is not robust enough for more complex tasks. For example, the grasping pipeline is trained to handle only bricks, i.e. solid parallelepiped objects, for which the grasping pose is straightforward to compute. Furthermore, the robot is not able to autonomously decide where to place the brick, but it is only able to place it in a predefined position. Also, the arm controller is quite primitive, since it doesn't handle collisions with the mobile base appropriately, and the arm is not able to avoid obstacles in its workspace.

Go Fetch: Mobile Manipulation in Unstructured Environments This work [3] presents a mobile manipulation system that combines perception, localization, navigation, motion planning and grasping skills into one common workflow for "fetch-and-carry" applications in unstructured indoor environments. The integration across the various modules is experimentally demonstrated in the video [2], showing the task of finding a commonly available object in an office environment, grasping it, and delivering it to a desired drop-off location.

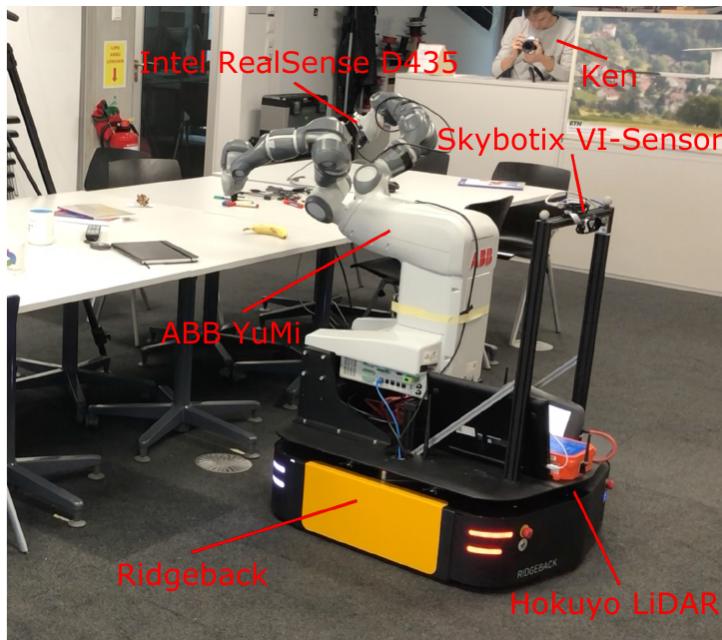


Figure 1.3: A picture of RoyalYumi in action. It features a two-arm ABB Yumi, a Clearpath Ridgeback mobile base, two Hokuyo 2D LiDARs, an Intel RealSense D435 and a Skybotix VI-Sensor. [3]

The research [3] is an example of a mobile manipulation system for pick-and-place tasks in an indoor environment that adopted many simplifications in order to achieve the desired results. The system uses a series of heuristics to approach the object and the grasping phase, as can be seen from the demonstration video [2]. The navigation uses a feature-based map and localization modules, while the arm controller is handled by MoveIt! [15]

solver coupled with the ROS framework [16]. This system is not very well integrated as each phase of the task is carried out by a different module, and shows how slow and inefficient the robot is in picking the banana. The grasping phase uses multiple views of the object in order to compute a better grasp pose, which works well but seems to be overly complex given the predefined task. Overall, the system can be regarded as a starting point for mobile manipulation and one of the first works in dual-arm manipulability.

Although traditional methods have led to promising mobile manipulation skills in some specific tasks, mobile manipulation tasks require the explicit programming of **hard-to-engineer behaviors** and often fail in more complex tasks where the decision-making process is hard. In addition, such solutions are generally very inflexible and error-prone due to the impossibility of modeling all the uncertainty of dynamic industrial environments when those are programmed.

1.3. Deep Reinforcement Learning: a Data-Driven Approach

Explicit programming is often needed in practice to account for uncertainties in the environment and sensors used, as well as to solve highly variable problems efficiently. Explicit behavior programming is therefore often tedious and impractical, and more flexible solutions are needed in environments where the robot must be adaptable. Alternatively, data-driven approaches address the main limitations of traditional methods and propose to learn robotic behaviors from real experience, thus alleviating the cost of modeling complex behaviors. This approach allows them to use deep neural networks to model the uncertainties of the environment, which leads to a more robust controller compared to traditional ones. Unlike deep learning (DL), the reinforcement learning (RL) paradigm allows to automatically obtain the experience needed to learn robotic skills through trial-and-error and allows to **learn complex decision-making policies**.

With RL, the explicit modeling of the problem is no longer required since the learned models are grounded in real experience. Recently, the combination of DL and RL, also known as Deep Reinforcement Learning (DRL), has made it possible to tackle complex decision-making problems that were previously unfeasible. It combines the ability of DL to model very high dimensional data with the ability of RL to model decision-making agents through trial and error. DRL has proven to be the state-of-the-art technology for learning complex robotic behaviors through the interaction with the environment and the training solely guided by a reward signal [10].

While ML-based methods are generally used for offline forecasting, DRL is generally used online in sequential decision-making problems. In fact, DRL allows one to autonomously learn complex control policies through trial and error and only guided by a reward signal. In the case of robotics, the most common use case is to use such algorithms to model agents capable of performing continuous control of robots.

DRL has been successfully applied in a wide variety of areas such as **robotics, computer vision and video games**. Taking into account the difficulty of modeling complex decision-making robotic skills, DRL offers a promising way to take advantage of the experience gathered by interacting with the environment to autonomously learn complex robotic behaviors. In particular, the field of DRL applied to robotics has recently gained popularity due to the remarkable performance obtained in applications with high decision-making and control complexity. Applications range from manipulation to autonomous navigation and locomotion. [8]

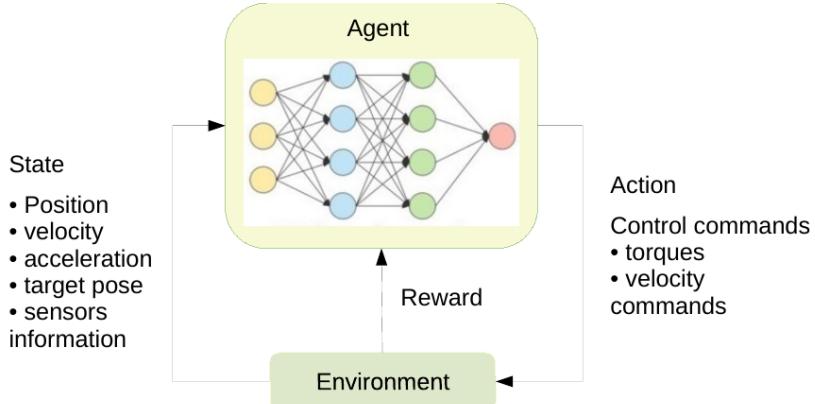


Figure 1.4: A schematic diagram example for robotic manipulation control using a data-driven approach such as DRL [10]

Pick and Place Operations in Logistics Using a Mobile Manipulator Controlled by Deep Reinforcement Learning The work [7] presents one of the first and pioneering approaches to DRL-based methods for pick and place tasks. Their work focused on the positioning problem, consisting of a local navigation problem where the robot must move to a desired position moving by small distances in a confined environment to reach the target object. They relied on DRL for controlling the mobile wheeled base robot, while the arm controller was handled by MoveIt! framework [15]. This can be regarded as a first step towards more complex tasks, as it shows the foundations of DRL-based methods for mobile manipulation. The method had some flaws, like the imprecise navigation due to errors in localization and odometry, which the network was not able to overcome since it was not fed video data stream. However, it paved the way for a lot of other works,

many of which are mentioned in this chapter.

Fully Autonomous Real-World Reinforcement Learning with Applications to Mobile Manipulation A work from Berkeley AI research [19] show *ReLMM*, a model that can learn continuously on a real-world platform without any environmental instrumentation, without human intervention, and access to privileged information, such as maps, objects positions, or a global view of the environment. Their method employs a modularized policy with components for manipulation and navigation, where manipulation policy uncertainty drives exploration for the navigation controller and the manipulation module provides rewards for navigation. They trained the policy on a room cleanup task, where the robot must navigate to and pick up items scattered on the floor. The robot **learns entirely from its sensors** in a real-world environment, without any simulation and minimal human intervention. Furthermore, the entire learning process is efficient enough for real-world training. On top of this, the robot can continually gather data at scale and improve its performance over time, with the auto-reset functionality.

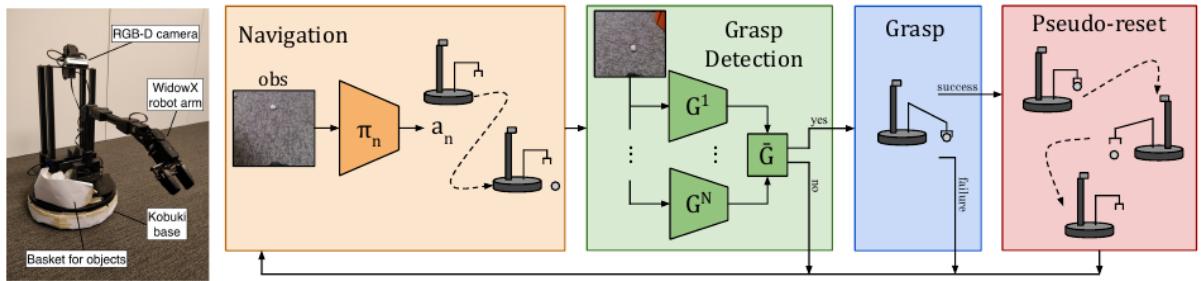


Figure 1.5: ReLMM partitions the mobile manipulator into a navigation policy and grasping policy. Both policies are rewarded when an object is grasped [19]

Although the results shown in [19] demonstrate efficient performance in the pick and place task, there are many issues circumvented by simplifying the problem. For example, the robot is very small, a modified version of Turtlebot, running in a small and contained environment. They didn't address any safety and collision issues since the platform mounted bumping sensors, and any collision would not harm the robot at all. Doing so enabled them to train the policy online without any prior simulation. Furthermore, the kinematics of the mobile base and the robotic arm are very simple, and having the stereo camera mounted on top of the robotic platform allowed them to easily detect the objects on the floor. This work paves the way for more complex systems and tasks, but it is still far from being a general solution for mobile manipulation.

1.3.1. Challenges in Data-Driven approaches

Two of the most important challenges here concern **sample efficiency and generalization**. The goal of DRL in the context of robotic manipulation control is to train a deep policy neural network, to detect the optimal sequence of commands for accomplishing the task. The current state of the algorithm can include the angles of joints of the manipulator, the position of the end effector, and their derivative information, like velocity and acceleration. The output of this policy network is an action indicating control commands to be implemented to each actuator, such as torques or velocity commands. When the robotic manipulator accomplishes a task, a positive reward will be generated. With these delayed and weak signals, the algorithm is expected to find out the most successful control strategy for the robotic manipulation [10].

The challenges of learning robust and versatile manipulation skills for robots with DRL are still far from being resolved in real-world applications. Currently, robotic manipulation control with DRL may be suited to fault-tolerant tasks, like picking up and placing objects, where failure will not cause significant damage. It is quite attractive in situations, where there are too many variables that make explicit modeling algorithms not work effectively [10].

However, even in this kind of application, DRL-based methods are not widely used in real-world robotic manipulation. The reasons are multiple, including sample efficiency and generation, where more progress is still required, as both the gathering experiences by interacting with the environment and the **collection of expert demonstrations** for imitation learning are expensive procedures, especially in situations where robots are heavy, rigid and brittle, causing high costs in case of failures.

Another very important issue is the **safety guarantee**. Unlike in simulation tasks, we need to be very careful that learning algorithms are safe, reliable and predictable in real scenarios. This is especially true when moving to applications that require safe and correct behaviors with high confidence, such as surgery or household robots taking care of the elderly or the disabled. There are also other challenges including but not limited to the algorithm explainability, the learning speed, and high-performance computational equipment requirements. [10]

Learning positioning policies for mobile manipulation operations with deep reinforcement learning The work proposed in [8] is a practical example of a DRL-based approach facing these challenges and the limitations to overcome (as well as the potentialities). The mobile platform in figure 1.6 is used in an industrial environment for an approaching task. The robot learns to navigate to the desk where the target object

is located, and then uses the MoveIt! planner to check whether the trajectory to pick the object is feasible. The robot's localization is based on AMCL and the learned policy serves as a controller for local navigation tasks. Their work shows many shortcomings of this approach, such as the integration between the DRL policy and localization package, with noise in the real environment affecting negatively the performance of the robot. As a result, the video presented shows an inefficient and jiggly movement because of the non-smooth control policy. They mention the necessity of mounting a stereo camera for navigation, to reduce the errors in the localization and improve the navigation of the robot.



Figure 1.6: KUKA robot mounted on a mobile platform for pick and place tasks in industrial environments [8]

Deep Reinforcement Learning Based Mobile Robot Navigation Using Sensor Fusion The problem of unstable and imprecise navigation with learned policies is overcome in the paper [23], which proposes a DRL-based approach for navigation in dynamic environments. The proposed method is based on the Deep Deterministic Policy Gradient (DDPG) algorithm, which is a model-free, off-policy actor-critic algorithm that uses deep neural networks to represent the policy and the critic functions. The proposed method is

evaluated in a simulated environment where the robot learns to navigate effectively and smoothly while avoiding unknown dynamic obstacles. This work shows the right direction towards more robust navigation and obstacle avoidance systems.

1.4. Whole-body mobile manipulator control

In most control approaches to mobile manipulation, base and manipulator operations are separated, since at any given time only one primary control objective is active. This separation principle is then augmented by a switching layer that determines the currently pertinent control objectives. The advantage of such a control formulation lies in its simplicity, i.e., priorities can be separated among the arm and the base with individually designed different control algorithms employed for each subsystem [20].

Instead, we refer to **whole-body control** as a unified control framework that considers the mobile manipulator as a single system (arm manipulator mounted on a mobile wheeled/legged robot). Despite the disadvantage that unified control needs to adhere to a single control framework, it allows for the exploitation of mobile manipulation in the true sense of the term, wherein the manipulator and mobile base can be controlled at the same time. This can lead to several advantages during task achievement and makes the robot more dynamic in terms of its capabilities. The formulation for this type of control involves considering the onboard manipulator as an extended joint space of the mobile base, where the motion controller considers both the base and manipulator state. As a result, base control can be completed simultaneously without affecting much the performance of the end-effector manipulability [20].

1.4.1. MPC+IK for articulated object manipulation

Articulated Object Interaction in Unknown Scenes with Whole-Body Mobile Manipulation In a work conducted by the universities of Zurich and Toronto [11], the researchers demonstrated a successful application of whole-body control for a mobile manipulator. The proposed system introduces a two-stage architecture for autonomous interaction with large articulated objects in unknown environments.

In the first stage, an object-centric planner focuses solely on the object, providing the action-conditional sequence of states for manipulation using RGB-D data. The second stage involves an agent-centric planner that formulates whole-body motion control as an optimal control problem, ensuring safe tracking of the generated plan, even in scenes with moving obstacles. The system proposed in [11] demonstrates effectiveness in handling

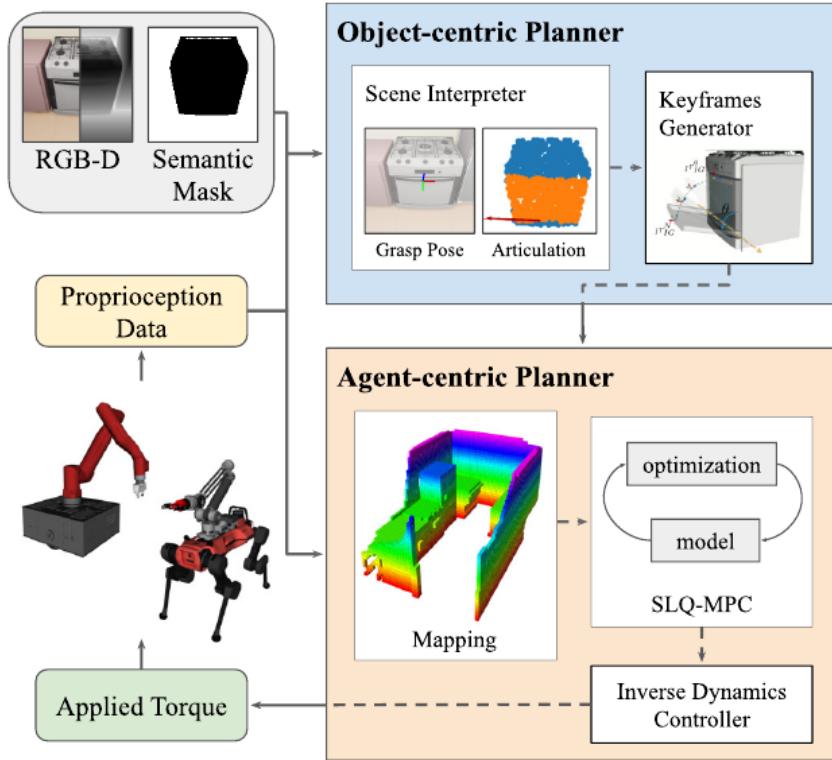


Figure 1.7: The two-level hierarchy in the proposed framework. The object-centric planner comprises a scene interpreter and keyframe generator. It uses perceptual information to generate task space plans. The agent-centric planner follows the computed plan while satisfying constraints and performing online collision avoidance. [11]

complex static and dynamic kitchen settings for both wheel-based and legged mobile manipulators. A comparison with other agent-centric planners reveals a higher success rate and lower execution time. Hardware tests on a legged mobile manipulator further confirm the system’s capability to interact with various articulated objects in a real kitchen. The approach combines **object-centric** and **agent-centric** planning, leveraging MPC-based solutions for improved success rates and reduced execution times, particularly in articulated object manipulation scenarios. The contributions include the extension of collision-free whole-body MPC for mobile manipulation, benchmarking in hyper-realistic simulation, and successful hardware experiments showcasing the system’s real-world applicability.

The work [11], published in 2022, is a successful real-world application of whole-body control using an MPC-based approach combined with IK solvers for articulated object manipulation. Furthermore, they achieved good performance in both dynamic and static real-world environments, which is a very challenging task for most of the existing methods.

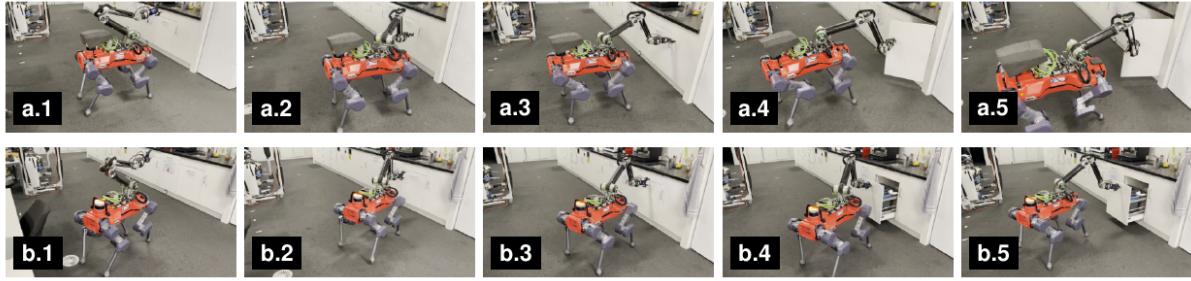


Figure 1.8: Legged mobile manipulation of articulated objects in the kitchen test scenario: (a) Drawer, (b) Cabinet. Throughout the interaction, we set the robot's gait schedule to trot. Only while grasping the handle, the robot enters stance mode. [11]

However, the proposed method is limited mostly concerning the grasping capabilities, which were hard-coded into the known interactive objects (in their case, the kitchen appliances handles), meaning that explicit behavior programming and tuning were needed for the grasping task. They propose data-centric methods to overcome these limitations.

This research proved the feasibility of this approach, which many other researchers claimed to be unfeasible and way too complex to be implemented in real-world applications. However, it is worth noting that the proposed method is not a general solution for all robot hardware configurations, and extending it to other robots would require a lot of effort and time since it boils down to an optimal control problem.

1.4.2. Deep Reinforcement Learning for high DoF control

Deep Whole-Body Control: Learning a Unified Policy for Manipulation and Locomotion The research paper [4] addresses the challenges in controlling legged manipulators with attached arms, proposing a novel approach to learn a unified policy for whole-body control using deep reinforcement learning. The standard hierarchical control pipeline is critiqued for its inefficiency, requiring significant engineering to coordinate arm and leg movements. The proposed method, Regularized Online Adaptation, aims to bridge the Sim2Real gap, and Advantage Mixing is introduced to overcome local minima during training. The authors present a low-cost legged manipulator design and demonstrate that their unified policy enables dynamic and agile behaviors across various tasks.

The paper emphasizes the limitations of current hierarchical models, advocating for learning-based methods like reinforcement learning to reduce engineering efforts and improve generalization. However, it critiques existing learning-based approaches for semi-coupling legs and arms, highlighting issues of coordination, error propagation, and non-



Figure 1.9: Framework for whole-body control of a legged robot with a robot arm attached. The left half shows how whole-body control achieves a larger workspace by leg bending and stretching. The right half shows different real-world tasks, including wiping the whiteboard, picking up a cup, pressing door-open buttons, placing, throwing a cup into a garbage bin and picking in clustered environments. [4]

smooth motions. The proposed unified policy not only allows coordination but also enhances the capabilities of individual components, such as the robot dynamically adjusting leg movements to extend the arm’s reach [4].

The challenges in scaling standard *sim2real* reinforcement learning to whole-body control are discussed, including the high degree of freedom, conflicting objectives, and dependencies between manipulation and locomotion. The paper introduces a hardware setup for a low-cost, fully untethered-legged manipulator and outlines a method for learning a unified policy to control both legs and the arm. The authors leverage causal structure in action space and regularization for domain adaptation to enhance stability and speed up learning.

The proposed method is evaluated through tasks like teleoperation and vision-guided tracking, demonstrating successful picking tasks using visual feedback from an RGB camera. Comparative analysis with a baseline method (MPC+IK) across various pick-up tasks measures success rate, average time to completion, IK failure rate, and self-collision rate. The authors conclude by acknowledging the preliminary nature of their results and highlight potential extensions, such as incorporating vision-based policies and addressing challenges in general-purpose object interaction [4].

As the authors of the paper [4] state, the main limitation (but also the core idea behind the control input) is the fact that the robot requires a human operator to provide the

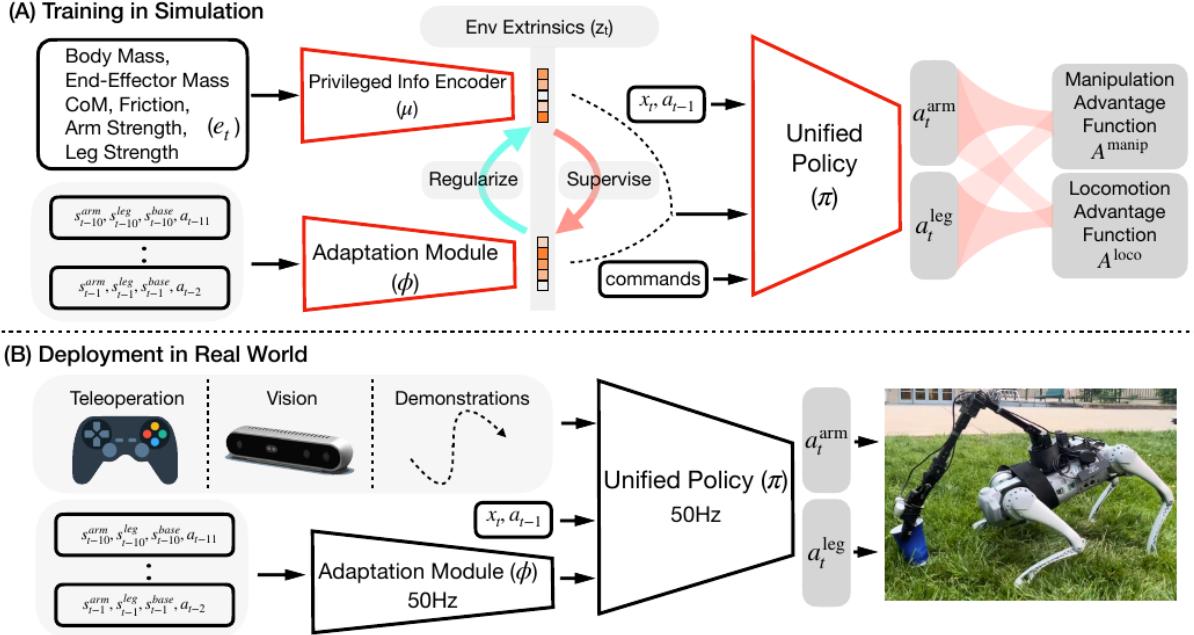


Figure 1.10: Whole-body control framework. During training, a unified policy is learned by conditioning on the environment extrinsic. During deployment, the adaptation module is reused without any real-world fine-tuning. The robot can be commanded in various modes including teleoperation, vision and demonstration replay. [4]

robot objectives, which are then translated into the control inputs. In fact, the robot is not able to autonomously plan its high-DoF motion trajectory, but instead, it is only able to either **track the end effector pose** given by the human operator or to track the April marker in the human’s hand. This limitation can be overcome with appropriate task training, but it is not a general solution. However, the proposed method is very promising, since it can achieve very good results in the robot body-hand coordination, which is a very challenging task for most of the existing methods. The movements look very smooth and natural, and the robot can perform simple tasks in a dynamic environment. The April marker tracking mode shows also the effectiveness of the use of a stereo camera for tracking the objective, meaning that promising results can be achieved in other tasks. This research used Nvidia Isaac Gym [12] as a simulation environment [14], which proved to be very powerful for training and overcoming the simulation-to-reality gap.

Learning Mobile Manipulation through Deep Reinforcement Learning [21]

This paper presents a mobile manipulation system that leverages deep reinforcement learning for unstructured environments. It integrates state-of-the-art algorithms with visual perception, adopting an efficient framework that separates visual processing from control. This design enables seamless generalization from simulation training to real-

world scenarios, utilizing only on-board sensors. Notably, the **transferability of policies** from simulation to real robots is a key strength, demonstrating the system's autonomy in grasping diverse objects across varied scenarios. The evaluation centers around a challenging mobile picking task, encompassing object recognition, collision-free robot-arm control, and object picking based on the learned policy.

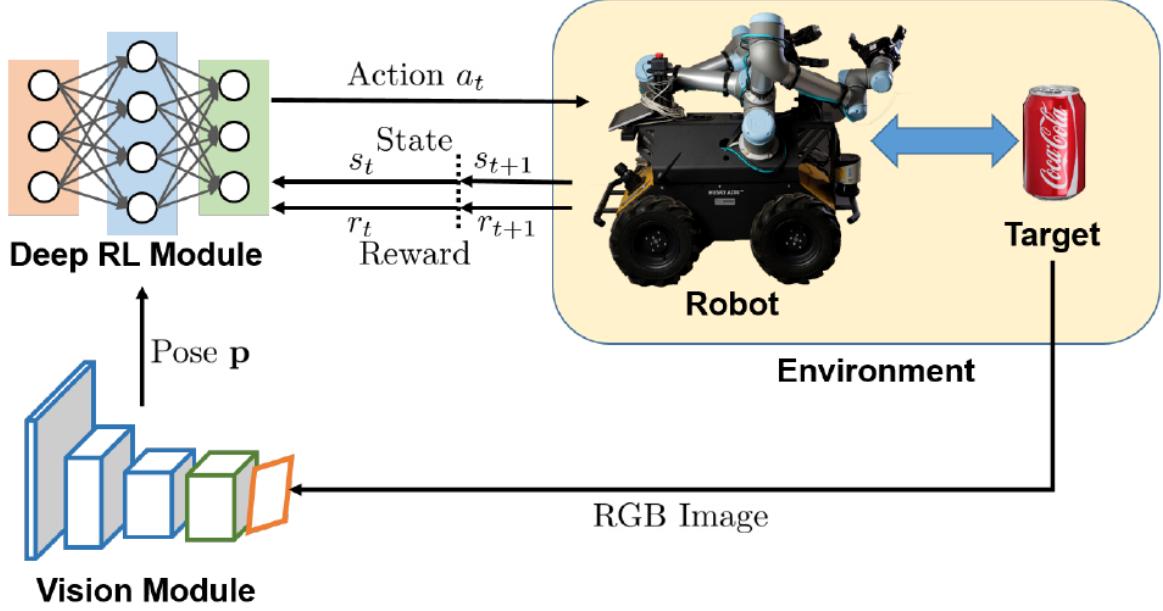


Figure 1.11: Learning-based mobile manipulation control framework. There are mainly two parts, deep reinforcement learning module and vision module. First, the vision module estimates the object 6 degrees of freedom pose p from images captured by an onboard RGB stereo camera. Then, based on the object pose p and current robot state s_t , deep reinforcement learning module predicts an action for the robot to act. A new state s_{t+1} and a reward r_{t+1} are received after action.[21]

Comparative assessments with state-of-the-art reinforcement learning algorithms highlight the stability and efficacy of the Proximal Policy Optimization (*PPO*) based system. Real-world experiments further confirm the system's ability to autonomously execute mobile grasping, overcoming challenges posed by the intricate nature of the mobile base, arm, gripper, and vision subsystems. Acknowledging differences between simulation and real-world dynamics, the paper addresses the need for closer coupling between mobile base and arm motions in future work. Overall, this work represents a significant contribution to the field, showcasing the potential of deep reinforcement learning for autonomous mobile manipulation in complex, unstructured environments.

The method proposed in [21] is very promising since it can achieve very good results in robot body-hand coordination, which is a very challenging task, especially in dynamic

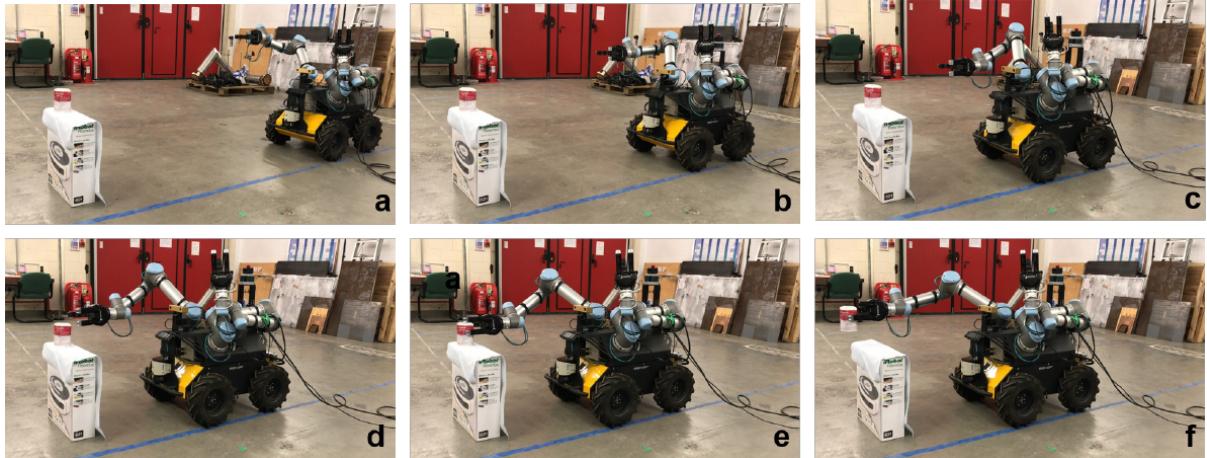


Figure 1.12: Real mobile grasping process for a soup can. (a) is starting, (b,c,d) is approaching, (e) is grasping, and (f) is picking up. [21]

environments. The movements look fluid and without any jitter, and the robot can perform simple tasks in a dynamic environment. This approach is one of the first to achieve complex control using vision-based perception together with deep reinforcement learning. However, it is far from being an optimal solution, since it achieves simple tasks in a very controlled environment.

Multi-Task Reinforcement Learning based Mobile Manipulation Control for Dynamic Object Tracking and Grasping This research paper [22] is a continuation of work demonstrated in their previous research paper [21] mentioned above. [22] addresses the challenges associated with agile control of mobile manipulators in unstructured environments, particularly focusing on dynamic object tracking and grasping. The authors propose a multi-task reinforcement learning-based control framework that aims to achieve general dynamic object tracking and grasping capabilities. The framework utilizes various dynamic trajectories as a training set, incorporating random noise and dynamics randomization to enhance policy generalization.

Experimental results demonstrate the trained policy’s ability to adapt to unseen dynamic trajectories, achieving a $0.1m$ tracking error and a 75% grasping success rate for dynamic objects. The proposed method is successfully deployed on a real mobile manipulator, showcasing its potential for real-world applications. The contributions of this work include the development of a versatile control framework and its successful deployment in unstructured environments, addressing the challenges of mobile manipulation with dynamic objects.

In [22] they use the proximal policy optimization (PPO) algorithm to train and learn a

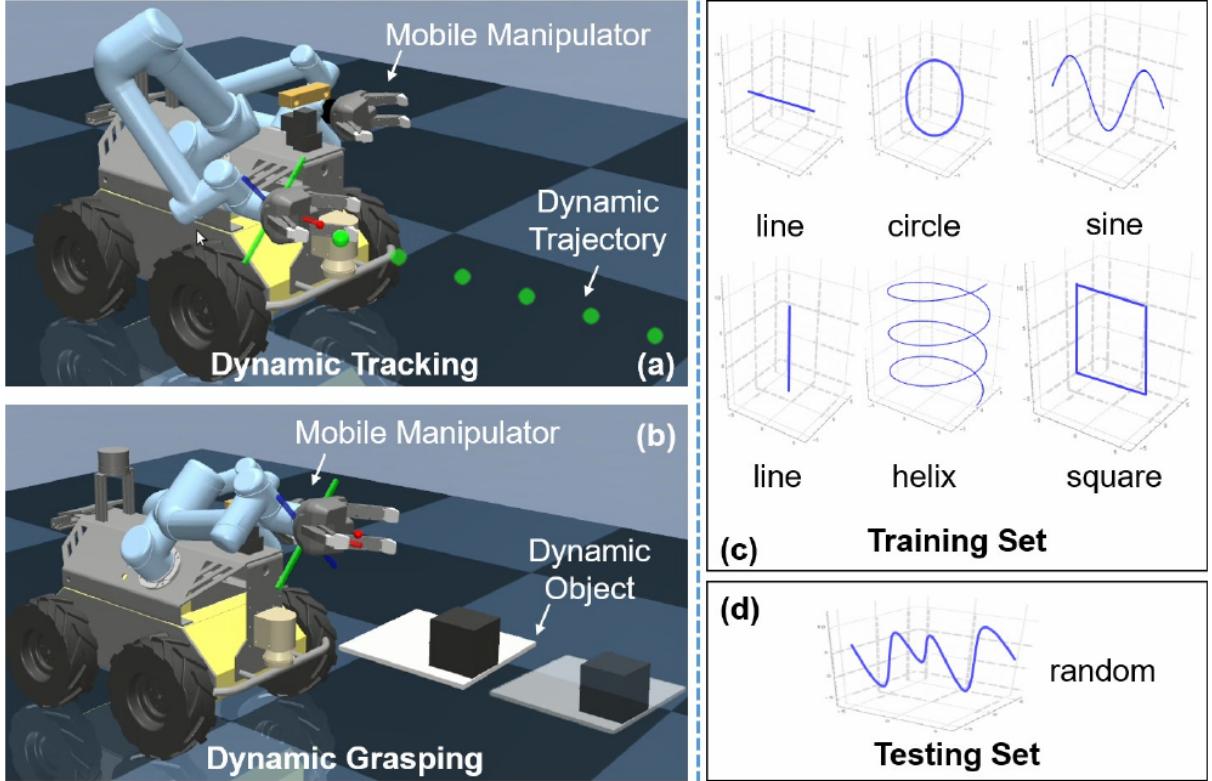


Figure 1.13: (a) Dynamic trajectory tracking task with a mobile manipulator. (b) Dynamic object grasping task with a mobile manipulator. (c) Several basic trajectories as multi-task RL training sets. (d) Random trajectories as multi-task RL testing set.[22]

policy, but the method is general and can be applied to most on/off-policy RL algorithms. PPO is one of the state-of-the-art RL algorithms that is easy to implement and tune, and performs relatively well. The policy is learned through a deep neural network.

The images 1.14 above show the training and testing process of the proposed method. They created a realistic simulation environment that enabled efficient parallel training of the policy, and simulation of the dynamic objects and tracking. They also managed to correctly transfer the learned policy to the real robot, which is a very challenging task for most of the existing methods. Artificial noise addition was essential for the training process since it allowed the policy to generalize better to unseen trajectories and environments.

Overall, this research provides valuable insights and a practical solution for advancing the field of agile mobile manipulation. This is one of the very few works that addresses the problem of dynamic object tracking and grasping, which is a very challenging task for most of the existing models, due to the difficulty in modeling and the extensive training required. Although the model doesn't show very high success rate in the tasks shown, it

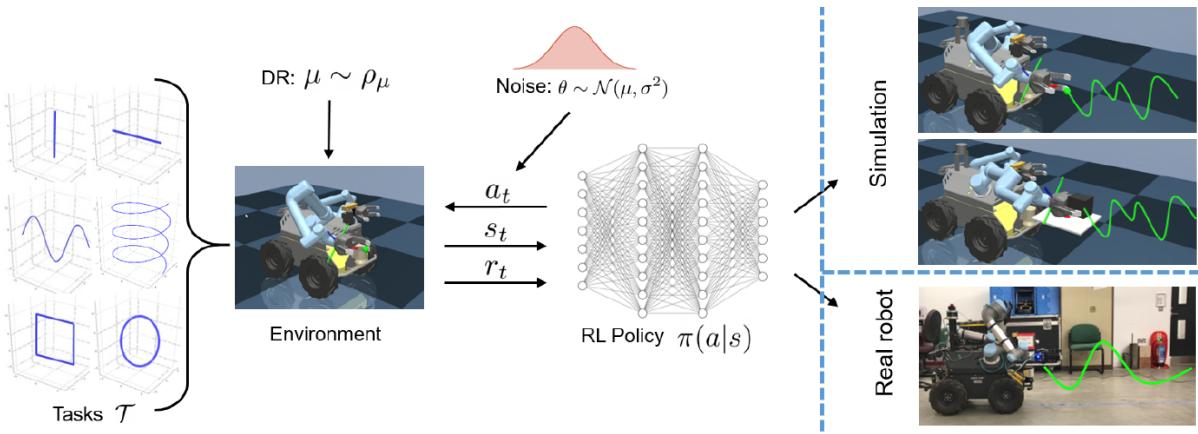


Figure 1.14: (a) In the multi-task RL training, six basic trajectories are used as the task training set to train a general policy. To improve the robustness, gaussian noise is added to the action and observation space in each training episode. (b) The RL testing includes simulation and real-world for policy evaluation.[22]

shows promising results and a direction of research that can be further explored.

1.4.3. Mobile manipulation with Imitation Learning

Imitation learning from human-provided demonstrations is a promising tool for developing generalist robots [5], as it allows people to teach arbitrary skills to robots. Indeed, direct behavior cloning can enable robots to learn a variety of primitive robot skills ranging from lane-following in mobile robots to simple pick-and-place manipulation skills to more delicate manipulation skills like spreading pizza sauce or slotting in a battery. However, many tasks in realistic, everyday environments require whole-body coordination of both mobility and dexterous manipulation, rather than just individual mobility or manipulation behaviors. Two main factors hinder the wide adoption of imitation learning for mobile manipulation.

- We lack accessible, plug-and-play hardware for whole-body teleoperation. Teleoperation for whole-body control requires a human to provide demonstrations of the task to the robot, using a specific and often expensive hardware setup. Furthermore, mobile manipulators, especially bimanual mobile manipulators, can be costly if purchased off the shelf.
- Prior robot learning works have not demonstrated high-performance bimanual mobile manipulation for complex tasks. The same goes for single manipulators in complex tasks in dynamic, cluttered or unknown environments.



Figure 1.15: Snapshots of the real robot experiments. The upper row shows a mobile tracking process in which the end-effector tries to track the target trajectory. The lower row shows a mobile grasping process in which the object moves randomly. [22]

Mobile ALOHA: Learning Bimanual Mobile Manipulation with Low-Cost Whole-Body Teleoperation On the hardware front, the researchers from Stanford University in their paper [5] present a low-cost and whole-body teleoperation system for collecting bimanual mobile manipulation data. Mobile ALOHA extends the capabilities of the original ALOHA [24], the low-cost and dexterous bimanual puppeteering setup [24], by mounting it on a wheeled base.

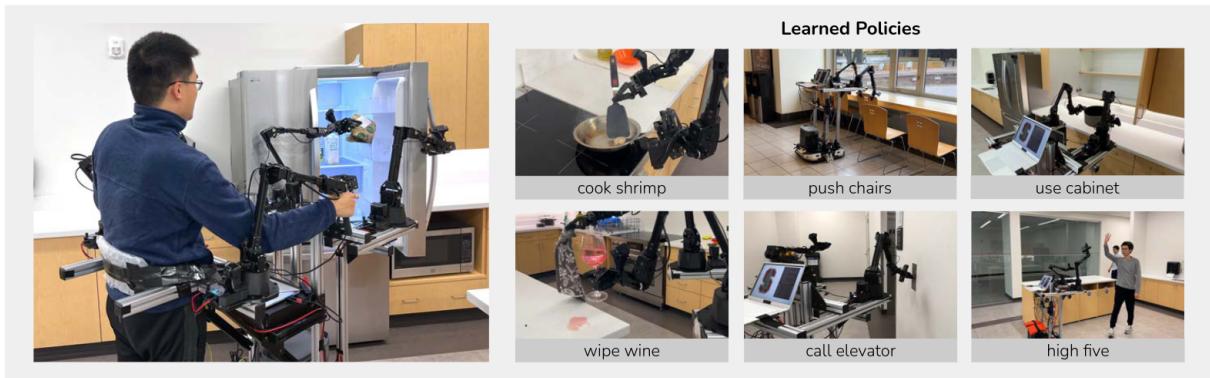


Figure 1.16: Low-cost mobile manipulation system that is bimanual and supports whole-body teleoperation. The system costs 32k USD including onboard power and compute. Left: A user teleoperates to obtain food from the fridge. Right: Mobile ALOHA can perform complex long-horizon tasks with imitation learning. [5]

The main contribution of the research paper [5] is the hardware setup of a low-cost

bimanual mobile manipulation system supporting whole-body teleoperation. Using data collected with Mobile ALOHA, performing supervised behavior cloning results in more accurate behavior mimicking. Also co-training with existing static ALOHA datasets [24] boosts performance on mobile manipulation tasks. The robot can not yet improve itself autonomously or explore to acquire new knowledge. In addition, the Mobile ALOHA demonstrations are collected by two expert operators.

The software architecture upon which the robot system in figure 1.16 relies is the one presented in [24], which is a modular architecture that allows for easy integration of functionalities. The architecture is described below.

Learning Fine-Grained Bimanual Manipulation with Low-Cost Hardware Fine manipulation tasks, such as threading cable ties or slotting a battery, are notoriously difficult for robots because they require precision, careful coordination of contact forces, and closed-loop visual feedback. Performing these tasks typically requires high-end robots, accurate sensors, or careful calibration, which can be expensive and difficult to set up. The researchers in [24] present a low-cost system that performs end-to-end imitation learning directly from real demonstrations, collected with a custom teleoperation interface. Imitation learning, however, presents its own challenges, particularly in high-precision domains: errors in the policy can compound over time, and human demonstrations can be non-stationary. To address these challenges, they developed a simple yet novel algorithm, Action Chunking with Transformers (ACT), which learns a generative model over action sequences. ACT allows the robot to learn multiple difficult tasks in the real world.

The proposed method is evaluated on a wide variety of complicated tasks, and the robot can perform well in most of them. However, their method has some limitations, mostly due to the difficulty in perception of the environment and the lack of training data. However, the architecture proposed shows very promising results from a few training examples.

In some extremely complex tasks (the ones in which even humans find some difficulty), the robots fail consistently, as reported in [24]. The failures can then be attributed to the difficulty in perception and lack of data, since the input video sequences were created from standard RGB cameras, therefore no depth information was available. Furthermore, object semantic understanding would have been very useful for the robot to understand the task deeply and perform better. The lack of data is also a big issue since the robot was trained by teleoperation. They believe that pretraining, more data and better perception are promising directions to tackle these extremely difficult tasks.

Shortcomings of imitation learning approaches The main shortcomings of imitation learning approaches are the lack of generalization and the difficulty in perception

of the environment. Imitation learning is feasible in a general context and environment proven that it is given enough data and the right perception tools:

- A large dataset of demonstrations collected with human teleoperation
- Human-made demonstrations available offline for training and evaluation
- A hardware component with a similar kinematic structure to the human demonstrator. The hardware-mimicking structure must also not occupy too much space, since it would affect the real robot's physical constraints.
- Mobile arms are controlled by each degree of freedom since it would not be possible to give a proper demonstration by inputting only the end-effector pose and getting the joints' positions by inverse kinematics.
- Appropriate depth perception and semantic object understanding would be ideal for the robot to understand the task deeply and perform better.

1.4.4. Comparison of Model-Based and Data-Driven approaches

Table 1.1: Summary of the main differences between model-based and data-driven approaches for robotic manipulator controls

| Approach | Model-Based | Data-Driven |
|--|--|--|
| Control Strategies | <ul style="list-style-type: none"> • Model Predictive Controllers (MPC) • Whole-Body Inverse Kinematics (IK) Solver | <ul style="list-style-type: none"> • Deep Reinforcement Learning (DRL) • Imitation Learning |
| Features | <ul style="list-style-type: none"> • Requires explicit modeling of system dynamics and kinematics • Suitable for simple tasks, unsuitable for complex tasks • Planning over end-effector pose or grasp in the workspace | <ul style="list-style-type: none"> • No explicit modeling of system dynamics • Learning from experience in simulation environments • High-level planning over tasks, object detection, manipulation or other objectives |
| Interpretability and Adaptability | <ul style="list-style-type: none"> • Explicit modeling implies high system interpretability • Adaptable to many tasks but requires behaviors re-programming | <ul style="list-style-type: none"> • Learned policies have very limited interpretability • Learning from experience allows high adaptability, given proper GPU-parallelized training |
| Advantages | <ul style="list-style-type: none"> • Small simulation-to-reality gap • Adaptable to many tasks but with explicit programming • No training required • Safer operation due to explicit physical limitations modeling | <ul style="list-style-type: none"> • No explicit modeling of system dynamics required • Learning from experience allows high generalization • Can perform well in unknown or dynamic environments • Can provide high body-hand movement coordination |
| Disadvantages | <ul style="list-style-type: none"> • Requires very accurate physical models for seamless integration • Doesn't perform well in complex tasks or dynamic environments • Difficult to adapt to complex tasks (low generalization) • High computational cost for the solver in high DoF systems | <ul style="list-style-type: none"> • Requires large amounts of training data and extensive training • Very long time needed to fine-tune the hyperparameters • May result in unstable and jiggly movements • May result in unsafe behaviors in real-world applications if not properly trained • Suffers a lot from the simulation-to-reality gap |

1.5. Addressing the Simulation-to-Reality Gap

The simulation-to-reality gap is a well-known problem in robotics, which is the difference between the performance of a robot in simulation and the real world. The simulation-to-reality gap is a major challenge in robotics, as it is difficult to accurately model the real world in simulation. This is especially true for mobile manipulation, where the robot must interact with the environment to perform its task.

Many works addressed this problem explicitly in the past, such as [10] and [6], proposing methods to make simulations more realistic and to improve the generalization of the learned policies.

Nvidia has developed some tools, such as Nvidia Isaac Gym [12] and Nvidia Isaac Sim [14], which are very powerful tools for simulating robots and environments. Isaac Sim is the simulation engine for Isaac Gym, which is a framework for training and testing robots in simulated environments with DRL and other tensor-based ML techniques. Isaac Gym is built on top of Nvidia Omniverse, which is a platform for real-time simulation and collaboration. Nvidia Omniverse allows also bridging the simulation with ROS thanks to the Isaac ROS bridge [13]. Overall, these tools enable reducing the simulation-to-reality gap, since they support many different robots, environments and perception sensors, so they are very powerful for training and testing DRL policies if properly set up.

A Sim-to-Real Pipeline for Deep Reinforcement Learning for Autonomous Robot Navigation in Cluttered Rough Terrain A work from Toronto [6] proposes a sim-to-real pipeline for deep reinforcement learning for autonomous robot navigation in cluttered rough terrain. Sim-to-real strategies have been developed for robot navigation tasks. For example, domain randomization can be applied to visual parameters such as texture, lighting, and object placement in synthetic environments to improve generalizability. The paper [6] addresses the **sim-to-real gap** in training robots for 3D terrain navigation by incorporating three sim-to-real strategies. Firstly, to account for depth camera measurement errors in 3D mapping that affect terrain steepness accuracy, the authors vary terrain steepness during training using a uniform distribution.

Secondly, the paper tackles disturbances in robot motion during interactions with rough terrain, such as slippage and insufficient traction. Both 3D terrain interactions and latency from visual odometry measurements contribute to disturbances in robot travel distance and yaw rotation angle. The third strategy focuses on addressing robot pose estimation errors arising from image and feature association errors. These errors, caused by lens distortion and ambiguous features, impact the accuracy of the robot's estimated 6 DOF

pose. The paper integrates these errors into the inputs of the DRL network to improve the model's performance in the face of localization inaccuracies.

1.6. Object Detection and Grasping

Grasp planning for mobile manipulators is a challenging problem tackled in several ways in the literature. On the one hand, grasping requires coordination within a very challenging high-dimensional constrained configuration space (mobile base, manipulator, gripper). On the other hand, grasping requires detecting objects and constructing data-driven geometrical representations, to produce effective grasping plans in the presence of statistical data uncertainties. Many of the traditional grasp planners (designed for stationary manipulators) can be used for mobile manipulators once the mobile base has been fixed.

However, a **generic grasping pipeline** is desirable, which achieves arm-base-gripper coordinated grasping given the information about object pose and the operating environment. Such coordinated manipulator and mobile base motion approaches are explored to find a feasible grasp or to identify an approach position that can lead to a successful grasp. (and only the manipulator moves for grasping). This may not be optimal as grasping can happen with the mobile base and the manipulator moving when the gripper is closing [20].

Broadly, automated grasping can be categorized into the following approaches [1]:

1. grasp using prior information from scene/objects
2. grasp using hand-eye coordination through learning directly from raw sensor data
3. grasp using template matching
4. grasp by detecting proper grasping pose using deep learning-based approaches
5. other field-specific approaches

Systems in each category have one or more limitations that are detailed in the following subsections. The majority of the existing systems are static, where a robotic system is fixed in an environment surrounded by the objects in its workspace.

Automated Object Manipulation Using Vision-Based Mobile Robotic System for Construction Applications The system designed and deployed for pick-and-place in a structured construction environment [1] integrates scene understanding and autonomous navigation with object grasping. To achieve this, two stereo cameras and a robotic arm are mounted on a mobile platform. This integrated system uses a global-to-

local control planning strategy to reach the objects of interest (i.e., bricks, wood sticks, and pipes). Then, the scene perception, together with grasp and control planning, enables the system to detect the objects of interest, pick them, and place them in a predetermined location depending on the application. The system is implemented and validated in a **construction-like environment** for pick-and-place activities. The results demonstrate the effectiveness of this fully autonomous system using solely onboard sensing for real-time applications with end-effector positioning accuracy of less than a centimeter.

However, the researchers mention also the shortcomings of the system, since the robot was developed for a field-specific application and, therefore non-adaptable to more generic use case scenarios. The system uses a heuristic-based approach to detect the bricks to grasp and pick up. Furthermore, the navigation pipeline relies on a static environment with no dynamic obstacles, since the arm manipulator does not employ any collision avoidance in the trajectory planning [1].

Autonomous Robotic Manipulation: Real-Time, Deep-Learning Approach for Grasping of Unknown Objects The work [17] proposes a novel approach for grasping unknown objects. The researchers present a full grasping pipeline proposing a real-time data-driven deep-learning approach for robotic grasping of unknown objects using MATLAB and deep convolutional neural networks. The proposed approach employs RGB-D image data acquired from an eye-in-hand camera centering the object of interest in the field of view. The arm control is based on **visual servo-ing techniques**, i.e. the robot arm is controlled based on the visual feedback from the camera. Their approach aims at reducing propagation errors and eliminating the need for complex hand-tracking algorithms, image segmentation, or 3D reconstruction, which are often either infeasible or too prone to errors. The proposed approach can efficiently generate reliable multi-view object grasps regardless of the geometric complexity and physical properties of the object in question. The system employed is a 7-DoF robotic manipulator controlled with an IK solver and a parallel gripper with overactuated fingers.

One of the main limitations of the approach in [17] is the fact that the grasping pipeline is implemented in MATLAB, therefore hardly portable and hardly replicable on other robotic hardware. Furthermore grasping with a parallel gripper is very limited, and the approach can work well with a limited set of objects. The authors demonstrated a good grasping capability with many different and unknown objects, but the approach cannot be generalized well without creating a multi-view perspective of the target object, to gain more understanding of the object's shape and geometry. Although the approach is very promising, there is room for improvement, especially in the CNN for grasping pose detection, which is the core of the grasping pipeline.

1.6.1. Grasping Soft Objects

Grasping soft objects is a challenging task for robotic manipulators, as it requires precise and effective control of the robot's end-effector to avoid damaging the object. Soft objects can deform under the pressure of the robot's gripper, making it difficult to grasp them effectively. Many existing methods for grasping soft objects rely on force control to regulate the pressure applied by the robot's gripper, but these methods can be difficult to implement and may not be effective for all types of soft objects.

Instead, some other solutions rely on open-loop control strategies to grasp soft objects, which do not require force feedback but may not be as effective as force control methods. These strategies employ **soft fingers or force-compliant grippers** that can conform to the shape of the object, reducing the risk of damage during grasping. Soft fingers can be made from materials like silicone or rubber that can deform to fit the shape of the object, providing a more secure grip without applying excessive pressure.

The inherent difficulty in grasping soft objects is the lack of a well-defined grasp configuration, as the object can deform and change shape during the grasping process, as well as the inevitable deformation of the soft gripper itself. This makes it difficult to predict the behavior of the object and the gripper during the grasping process. As a result, creating simulated environments for training and testing grasping policies for soft objects is challenging, as it requires accurate modeling of the object's deformation properties and the gripper's compliance to ensure realistic results.

Many existing methods for grasping soft objects that rely on force or torque control can be difficult to adapt to different types and shapes of soft objects. This often makes the only viable solution to create control strategies pre-tuned for specific soft objects, which limits the applicability of the method to only known objects of a specific shape but can be effective for many applications. The other difficulty in implementing adaptable control strategies for grasping soft objects is the need to create ad-hoc solutions engineered for specific soft gripper configurations, which are hard to generalize to other robots' end effectors or grippers.

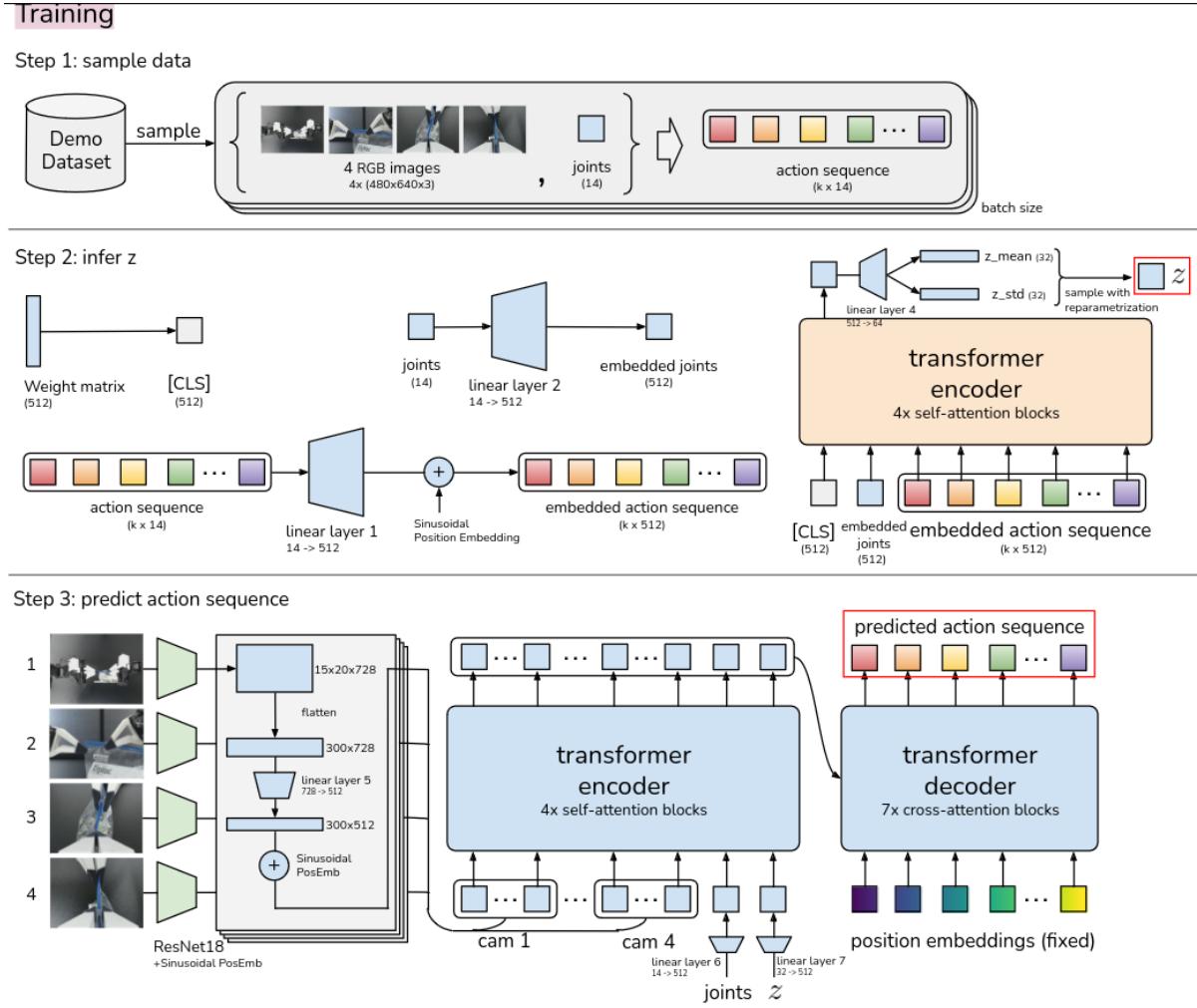


Figure 1.17: Detail architecture of Action Chunking with Transformers (ACT). First, they train ACT as a Conditional VAE (CVAE), which has an encoder and a decoder. The encoder of the CVAE compresses action sequence and joint observation into z , the style variable. The encoder is discarded at test time. The decoder or policy of ACT synthesizes images from multiple viewpoints, joint positions, and z with a transformer encoder, and predicts a sequence of actions with a transformer decoder. z is simply set to the mean of the prior (i.e. zero) at test time [24].

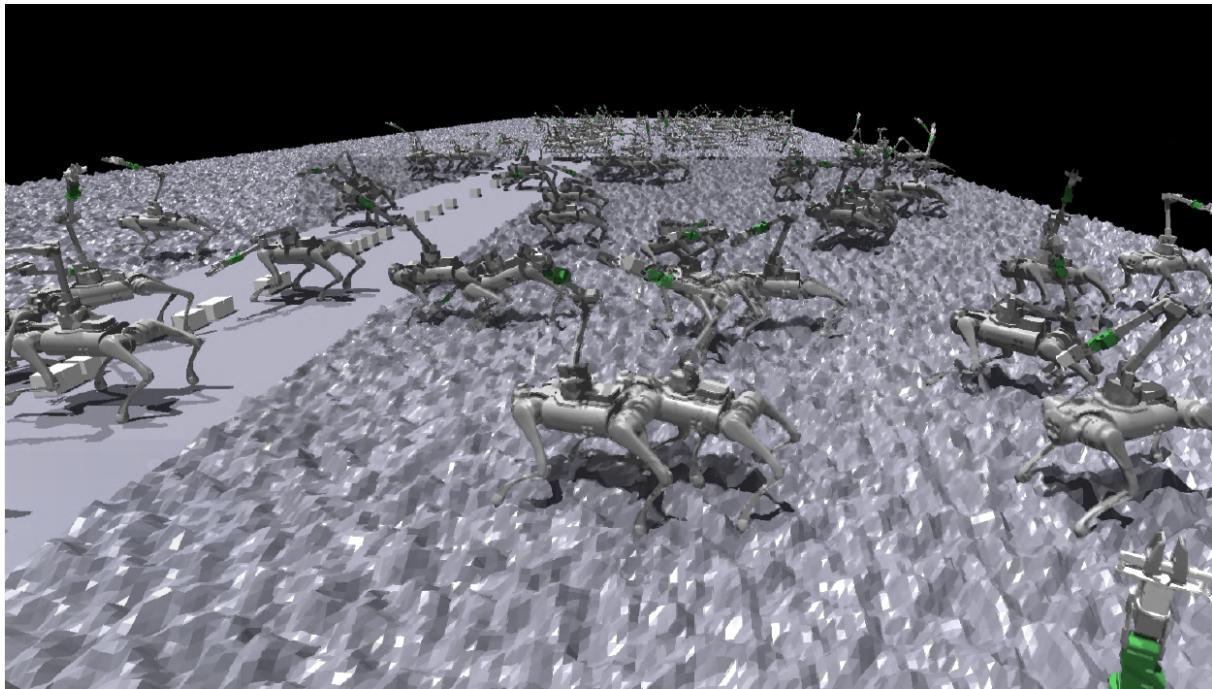


Figure 1.18: Simulated environment with Nvidia Isaac Gym [13]. The screenshot depicts a simulated environment with many legged mobile manipulators trained in parallel using DRL, as in [11]. The environment is simulated using Nvidia Isaac Sim [14]

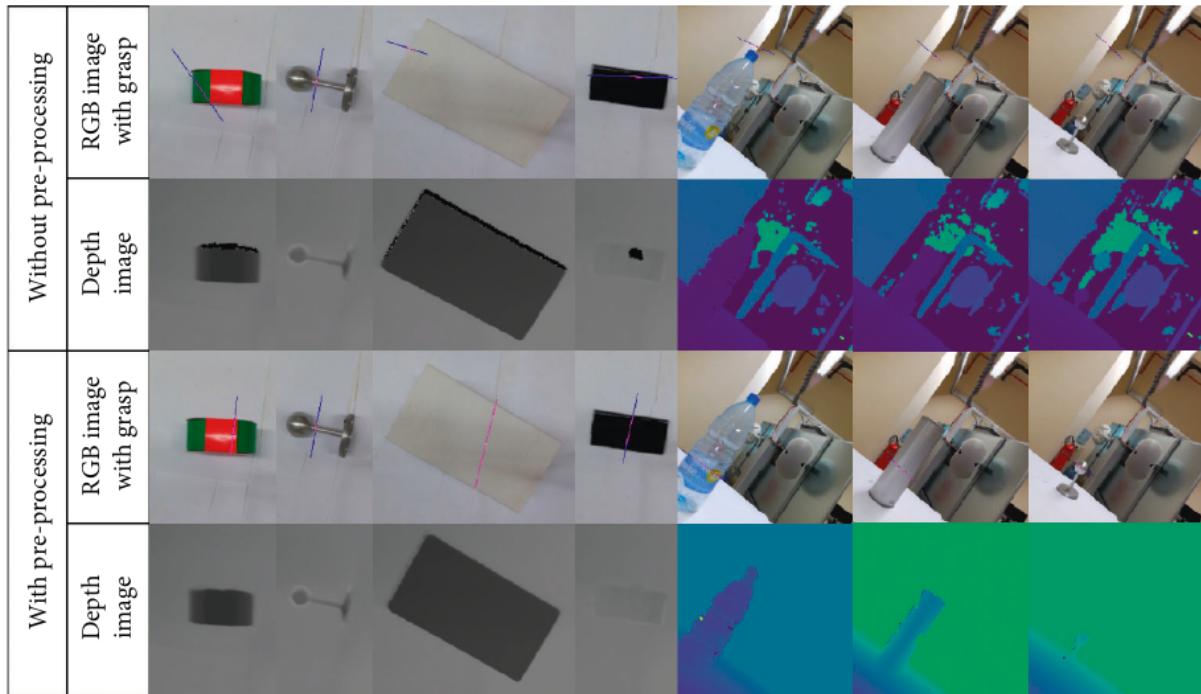


Figure 1.19: Grasp generation results: comparison between grasp generated by the GG-CNN with and without RGB-D image preprocessing for shiny and black objects [17]

Bibliography

- [1] Khashayar Asadi et al. “Automated Object Manipulation Using Vision-Based Mobile Robotic System for Construction Applications”. In: *Journal of Computing in Civil Engineering* 35.1 (2021), p. 04020058. DOI: 10.1061/(ASCE)CP.1943-5487.0000946. eprint: <https://ascelibrary.org/doi/pdf/10.1061/%28ASCE%29CP.1943-5487.0000946>.
- [2] ETH Zurich Autonomous Systems Lab. *Go Fetch: Mobile Manipulation in Unstructured Environments*. Published on YouTube by the Autonomous Systems Lab, ETH Zurich. 2020. URL: <https://www.youtube.com/watch?v=videoID>.
- [3] Kenneth Blomqvist et al. “Go Fetch: Mobile Manipulation in Unstructured Environments”. In: (2020). DOI: 10.48550/arXiv.2004.00899. arXiv: 2004.00899 [cs.RO]. URL: <https://doi.org/10.48550/arXiv.2004.00899>.
- [4] 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. DOI: <https://doi.org/10.48550/arXiv.2210.10044>. arXiv: 2210.10044. URL: <https://manipulation-locomotion.github.io/>.
- [5] Zipeng Fu, Tony Z. Zhao, and Chelsea Finn. “Mobile ALOHA: Learning Bimanual Mobile Manipulation with Low-Cost Whole-Body Teleoperation”. In: (2024). DOI: 10.48550/arXiv.2401.02117. arXiv: 2401.02117 [cs.RO]. URL: <https://mobile-aloha.github.io/>.
- [6] Han Hu et al. “A Sim-to-Real Pipeline for Deep Reinforcement Learning for Autonomous Robot Navigation in Cluttered Rough Terrain”. In: *IEEE Robotics and Automation Letters* 6.4 (2021), pp. 6569–6576. DOI: 10.1109/LRA.2021.3093551.
- [7] Ander Iriondo et al. “Pick and Place Operations in Logistics Using a Mobile Manipulator Controlled with Deep Reinforcement Learning”. In: *Applied Sciences* 9.2 (2019). ISSN: 2076-3417. DOI: 10.3390/app9020348. URL: <https://www.mdpi.com/2076-3417/9/2/348>.

- [8] Ander Iriondo et al. “Learning positioning policies for mobile manipulation operations with deep reinforcement learning”. In: *International Journal of Machine Learning and Cybernetics* 14.9 (Sept. 2023), pp. 3003–3023. DOI: 10.1007/s13042-023-01815-8. URL: <https://doi.org/10.1007/s13042-023-01815-8>.
- [9] Sanghoon Ji et al. “Learning-Based Automation of Robotic Assembly for Smart Manufacturing”. In: *Proceedings of the IEEE* 109.4 (2021), pp. 423–440. DOI: 10.1109/JPROC.2021.3063154.
- [10] Rongrong Liu et al. “Deep Reinforcement Learning for the Control of Robotic Manipulation: A Focussed Mini-Review”. In: *Robotics* 10 (2021), p. 22. DOI: 10.3390/robotics10010022. URL: <https://doi.org/10.3390/robotics10010022>.
- [11] Mayank Mittal et al. “Articulated Object Interaction in Unknown Scenes with Whole-Body Mobile Manipulation”. In: *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2022, pp. 1647–1654. DOI: 10.1109/IROS47612.2022.9981779. URL: <https://www.pair.toronto.edu/articulated-mm/>.
- [12] NVIDIA. *Isaac Gym*. 2024. URL: <https://developer.nvidia.com/isaac-gym>.
- [13] NVIDIA. *Isaac ROS bridge*. 2024. URL: <https://developer.nvidia.com/isaac-ros>.
- [14] NVIDIA. *Isaac Sim*. 2024. URL: <https://developer.nvidia.com/isaac-sim>.
- [15] David Pick and MoveIt! Developers. *MoveIt! Documentation*. 2023. URL: <https://moveit.picknik.ai/main/index.html>.
- [16] Open Robotics. *ROS 2 Documentation*. 2023. URL: <https://docs.ros.org/en/rolling/index.html>.
- [17] Malak H. Sayour, Sharbel E. Kozhaya, and Samer S. Saab. “Autonomous Robotic Manipulation: Real-Time, Deep-Learning Approach for Grasping of Unknown Objects”. In: *Journal of Robotics* 2022 (June 2022). ISSN: 1687-9600. DOI: 10.1155/2022/2585656. URL: <https://doi.org/10.1155/2022/2585656>.
- [18] Petr Štibinger et al. “Mobile Manipulator for Autonomous Localization, Grasping and Precise Placement of Construction Material in a Semi-Structured Environment”. In: *IEEE Robotics and Automation Letters* 6.2 (Apr. 2021), pp. 2595–2602. DOI: 10.1109/LRA.2021.3061377.

- [19] Charles Sun et al. “Fully Autonomous Real-World Reinforcement Learning with Applications to Mobile Manipulation”. In: *Proceedings of the 5th Conference on Robot Learning*. Ed. by Aleksandra Faust, David Hsu, and Gerhard Neumann. Vol. 164. Proceedings of Machine Learning Research. PMLR, Nov. 2022, pp. 308–319. URL: <https://proceedings.mlr.press/v164/sun22a.html>.
- [20] Shantanu Thakar et al. “A Survey of Wheeled Mobile Manipulation: A Decision-Making Perspective”. In: *Journal of Mechanisms and Robotics* 15.2 (July 2022), p. 020801. ISSN: 1942-4302. DOI: 10.1115/1.4054611. URL: <https://doi.org/10.1115/1.4054611>.
- [21] Cong Wang et al. “Learning Mobile Manipulation through Deep Reinforcement Learning”. In: *Sensors* 20.3 (2020). ISSN: 1424-8220. DOI: 10.3390/s20030939. URL: <https://www.mdpi.com/1424-8220/20/3/939>.
- [22] Cong Wang et al. “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)*. July 2022, pp. 34–40. DOI: 10.1109/ACIRS55390.2022.9845515.
- [23] Kejian Yan, Jianqi Gao, and Yanjie Li. “Deep Reinforcement Learning Based Mobile Robot Navigation Using Sensor Fusion”. In: *2023 42nd Chinese Control Conference (CCC)*. 2023, pp. 4125–4130. DOI: 10.23919/CCC58697.2023.10240555.
- [24] Tony Z. Zhao et al. “Learning Fine-Grained Bimanual Manipulation with Low-Cost Hardware”. In: (2023). DOI: 10.48550/arXiv.2304.13705. arXiv: 2304.13705 [cs.RO]. URL: <https://tonyzhaozh.github.io/aloha/>.