Motion planning and safe object handling for a low-resource mobile manipulator as human assistant

Rosario Francesco Cavelli, Pangcheng David Cen Cheng, Marina Indri
Dipartimento di Elettronica e Telecomunicazioni
Politecnico di Torino
Corso Duca degli Abruzzi 24, 10129, Torino, Italy
{rosario.cavelli, pangcheng.cencheng, marina.indri}@polito.it

***Abstract*—Nowadays, mobile manipulators can support humans in daily life tasks while sharing the same workspace. These robots are usually requested to perform pick-and-place actions that involve objects that must be handled with care, since they may hurt the human operators. To optimize their utility as smart assistants, they require autonomous grasping pose generation, object recognition, and pose estimation capabilities. In addition, since they work in dynamic environments, adaptability is essential, hence predefined starting positions for grasping actions should be avoided. These demands are even more challenging for robots with limited computational capabilities.**

In this paper, we propose an approach that demonstrates how to improve the capabilities of a low-resource mobile manipulator. First, an easy way to model the robot as a unique system for holistic motion planning is developed. Then, we propose a lightweight approach to generate the grasping point to pick a requested item that relies only on the available CPU. Finally, a simple yet flexible solution that involves human feedback is adopted to let the robot handle potentially dangerous objects, while ensuring the operator's safety.

The proposed solution has been developed in ROS1 and experimentally tested on the LoCoBot mobile manipulator in a laboratory environment.

***Index Terms*—Mobile manipulation, autonomous pick and place, ROS1, human-robot collaboration**

I. Introduction

During the last two decades, robots have become more widespread and increasingly used in various fields, ranging from industries and production chains to public places and houses. Lots of changes have occurred since robots are no longer confined in safety cages and they can interact with humans. The developments in the field of Human-Robot Interaction (HRI) allowed to evolve from a simple coexistence between humans and robots, to cooperation between them, letting them share (at least partially) the same workspace at the same time while working to fulfil the same goal, and finally to their full collaboration, which involves intentional contact and exchange of information [1], [2].

There exist different robot types to be used to meet the application's requirements. Mobile manipulators are systems that combine a mobile base and a manipulator, so they have the motion capabilities of mobile robots, coupled with the dexterity and agility of manipulators [3]. This results in high versatility and efficiency, making them ideal for tasks such as pick-and-place and material handling in large working spaces. By integrating the advantages of both robot categories, mobile manipulators serve as versatile service robots suitable for personal and professional applications. The main benefits behind such systems are risk reduction, since mobile manipulators can be deployed to perform dangerous tasks, hence reducing the risk of injuries for human workers. More importantly, they give the possibility of enhancing the productivity and efficiency of human operators, allowing them to focus on those tasks where their expertise cannot be substituted, but also improving their comfort and wellbeing in Industry 4.0 [4]–[7].

Mobile manipulators also come with challenges, since to let them correctly deal with objects it is necessary to coordinate the motion of the base and the robotic arm. Depending on how the mobile manipulator is treated, it is possible to group the planning algorithms into two classes: (i) separate-subsystems planners, where the mobile base and the arm are considered independent, and hence two distinct planners are used for them, and (ii) holistic planners, where the entire robot is modelled as a single system, and hence planning is performed considering both the arm and the base [8].

Another crucial task for this type of robot is how to interact with an object to correctly handle it. There are several ways to address this task, depending on the level of autonomy given to the robot. Thanks to the recent advances made in the Artificial Intelligence (AI) and Machine Learning (ML) fields, more intelligent robots can generate the best grasping pose to use, depending on the particular object, its current pose and other aspects that may influence the way the robot has to handle the requested item. Although the high performance and high degree of autonomy obtained using these approaches, the computational resources they usually need is a crucial aspect that must be taken into account.

This paper aims at providing an easy-to-use holistic planner for a generic mobile manipulator, along with a description of a simple and lightweight method to generate the grasping poses, as well as a flexible solution to correct them, using the human feedback to properly handle potentially dangerous objects. In this way, the risks for humans present in the workspace can be reduced. The proposed solution is developed for low-resource mobile agents using a distributed setting and takes inspiration from recent works, as discussed in Section II. After such an overview, the proposed approach is described in Section III, while Section IV illustrates the experimental setup and reports the results of the carried out tests. Finally, Section V draws some conclusions and open issues for possible future works.

II. Related works

A. Motion planning for mobile manipulators

Separately planning the motion for the mobile base and the manipulator may be convenient from a computational point of view, but employing two independent planners generally results in a sub-optimal plan. To overcome this issue, it is convenient to model the mobile manipulator as a single high degree of freedom (DOF) system. This strategy allows the robot to simultaneously reach the desired mobile base's pose and the manipulator's goal, i.e., the end-effector (EE) pose, as illustrated in Figure 1.

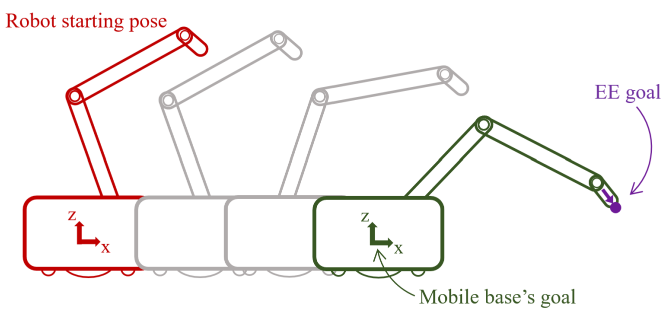


Fig. 1: Visualization of the robot's motion obtained using a holistic planner.

Solutions where the mobile manipulator is considered as two distinct sub-systems, are anyway available in literature, generally using well-known motion planners for mobile robots and manipulators. In [9], the authors used Dijkstra's algorithm to find a suitable path for the mobile platform, while the manipulator motion was planned using the Open Motion Planning Library (OMPL). The solution proposed in [10] makes use of the A* and the Timed-Elastic Band (TEB) algorithms as global and local planners, respectively, for the mobile base, whereby the manipulator's motion planning relies on the RRT* planner. Castaman et al. proposed in [11] a new method for Task and Motion Planning (TAMP), called Receding Horizon TAMP (RH-TAMP). Such an approach has been tested using a mobile manipulator, leveraging Dijkstra's algorithm as a planner for the mobile base, while the RRT-Connect algorithm was in charge of the manipulator motion planning. RRT-Connect was also used in [12] for both the mobile base and the manipulator, but the planning was carried out separately. Tests were conducted in a simulated kitchen using a double-arm mobile manipulator. For the pick-and-place operations, the objective for the mobile base was predetermined, while the manipulator's goal configuration was chosen from pre-calculated end-effector poses to avoid collisions. Such a strict use of pre-computed mobile base's poses and arm trajectories can seriously compromise the efficiency of the robot, especially if the items to pick are moved within the environment. A slightly more flexible approach has been proposed in [13], where the authors used the A* algorithm for the mobile base planning and RRT for the manipulator's motion, while picking and unloading objects. A set of possible locations from where it is possible to grasp and unload objects was pre-recorded, and the best one was selected considering the Euclidean distance from the current robot's position.

Using independent planners offers the advantage of easy debugging and issue investigation for each sub-module, although it leads to a slower and unnatural movement that is generally sub-optimal for the entire robot. Additionally, the desired EE pose belonging to the manipulator's reachability space strongly depends on the final base pose. To overcome all these issues, as demonstrated in [14], holistic planners can be adopted, resulting in an optimal and more harmonious movement, that also removes the need of fixing a mobile base's pose from where to start the grasping action.

In [15], the authors explicitly calculated the free space within which the robot operates, and subsequently, through sampling, they determined the desired goal pose for the mobile manipulator. The explicit representation and computation of free space is burdensome, and it would require recalculating it from scratches for any change in the environment.

In [16], Corke et al. proposed a holistic motion planner based on the solution of a Quadratic Problem (QP). The results demonstrated the effectiveness of holistic planners, obtaining a faster, smoother and more natural motion of the entire robot. The main drawback of this solution is that obstacles are not considered as constraints of the optimization problem, hence they are not taken into consideration during the motion planning. Moreover, the QP used to obtain the trajectory makes use of information that strongly depends on the type of robot that is being used, including the extended Jacobian matrix of the high-DOF system in use and its joint constraints. For these reasons, the proposed solution cannot be considered an easy-to-use solution to solve the holistic motion planning problem, and it cannot be employed in dynamic environments.

In [17], the application of deep reinforcement learning techniques to mobile manipulation tasks is discussed, addressing the challenges of coordinating a mobile base and a manipulator in unstructured environments. Although its effectiveness, the use of deep reinforcement learning requires lots of computational resources and effort to find optimal policies for the design of the correct interaction between the robot and the environment. Moreover, the training process is strictly dependent on the robot used, requiring to start from scratches each time a different one is considered.

Such drawbacks are quite common among holistic planners, since they are platform-dependent, meaning that a proposed solution is designed to work on a specific robot. Also, due to the intrinsic redundancy of a mobile manipulator, they require high computational resources and time to find a motion plan.

In order to exploit the advantages of holistic solutions, in this paper we propose an approach to obtain an easy-to-use yet effective holistic planner, trying to overcome the platform-dependency problem, and making it feasible also for low-resource mobile manipulators.

B. Generation of the Grasping Pose

Grasp pose generation is a crucial task in robotics, particularly in applications where a robot needs to manipulate objects using one or more arms. The primary objective is to determine a suitable pose for the robot's end-effector to successfully grasp an item. Various methods have been proposed in literature to address this challenge; the most recent approaches use Neural Networks (NNs) to generate optimal grasping poses.

In [18], the authors proposed to modify an AlexNet Convolutional Neural Network (CNN) model pre-trained on ImageNet, by adding 18 million new parameters in the fully connected layers for the task of predicting grasp location. The high number of parameters requires high-performing hardware to compute predictions, making this method not suitable for low-resource robots. In [19], the authors developed a customized CNN, called Generative Grasping CNN (GG-CNN), to address this task. This network directly generates optimal grasping poses from images captured by an RGB-D camera. Notably, GG-CNN is relatively small, and the entire grasping pipeline takes only 16 ms to execute a grasping action, although the desired pose involves picking the item from the top, a working condition that is not always feasible for mobile manipulators. In [20], a novel grasping pose generation algorithm is introduced, focusing on the interaction between object picking and placing in cluttered scenes. Using an RGB-D camera on the robotic arm's wrist, a 3D render of the placement scene is constructed through 3D convolutional layers. A single-depth image of the object is collected to reconstruct its 3D model. Cross-correlating this information produces an affordance placement map. Despite its high accuracy, the method is time-consuming, due to the extensive depth image collection needed for scene reconstruction before each grasping task. In [21], the authors developed a technique to generate the grasping pose depending on the action that must be carried out, thus dealing with the Robotic Object Affordance issue. The method implements an NN that incorporates both 2D images and 3D object models. The images depict a human engaging with the object, each labeled to specify the task being performed. The network's training phase aims at replicating a form of learning by demonstration, leveraging the images as the representation from which the robot learns.

The solution that we propose in this paper can be applied to mobile manipulators with diverse computation capabilities, and it allows to pick items from different heights compatible with the robot's characteristics. The key idea behind this solution is to adapt a CNN usually exploited to complete the object recognition problem, merging the information coming from a depth sensor like an RGB-D camera.

III. Proposed Approach

The approach proposed in this paper aims at improving the motion capabilities of a mobile manipulator by planning and following a collision-free trajectory, which is unique for both the mobile base and the arm. Furthermore, the developed scheme allows the mobile manipulator to recognize known objects and generate the best grasping point to handle also those that are potentially dangerous. Even though the steps followed to design the proposed solution can be applied to different robots, the implementation is based on a LoCoBot mobile manipulator [22], providing a working case study for the research community. Some preliminary results have been developed in [23], whose code is available at [24]. Details about the LoCoBot model and sensors/hardware setup can be found in Section IV-A.

This section is divided into two parts: the first one describes the main steps to obtain an easy-to-use motion planner able to find a trajectory for the high-DOF system, while the second one unfolds all implementation aspects to make the robot able to recognize known objects and to generate a suitable grasping point.

A. Problem scenario

The deployment of mobile manipulators in pick-and-place operations offers exceptional flexibility in environments such as plants and warehouses. This allows humans to instruct the robot to search, pick and place specific products, so as to build a more versatile and efficient operational setup.

A typical example of a working environment for a mobile manipulator is a warehouse, in which it is necessary to grasp stored items and release them in a depot. This environment includes static obstacles like shelves and walls, as well as dynamic obstacles with different levels of predictability, ranging from other mobile agents to humans. Also, the objects that the mobile manipulator should handle may have different shapes and characteristics, which make them potentially dangerous when the robot interacts with humans, if they are not correctly handled. For instance, those items with sharp or pointy edges must be grasped and moved adequately and more carefully.

B. Holistic motion planning for a mobile manipulator

From the analysis of the related works, it is evident the need to quickly develop a framework capable of (i) solving the holistic planning problem, and (ii) exploiting the main planning algorithms available in literature.

The solution proposed in this paper is based on the framework MoveIt! [25], which is one of the most used ROS packages for motion planning. It is an open-source planning framework that provides several planners, and it constitutes a flexible way to generate a complete motion plan considering different constraints. Starting from the Universal Robot Description File (URDF), which contains all the information about the physical properties and constraints of the robot, it is possible to create a customized version of the MoveIt! package. The MoveIt! setup assistant is in charge of loading the URDF file, extracting all important information and creating the entire package, following the requests provided by the user through a convenient graphical interface. The information extracted from the URDF makes possible the self-collision checking, the definition of some planning groups, and the choice of the motion planning algorithm, taking into account the selected EE.

MoveIt! gives also the possibility to change the robot description by adding virtual joints, representing additional DOFs beyond the actuated physical ones of a robotic system. The main aspect of virtual joints is that a single one is able to provide more than one DOF, depending on its type; e.g., the planar virtual joint provides 3 additional DOFs, while the floating one adds 6 of them. They are crucial for handling various planning scenarios and enabling more flexible manipulation tasks. In particular, they are used in the proposed approach just to represent the manipulator and the mobile base as a single system.

In this way, virtual joints could be used to enhance the motion planning abilities of a mobile manipulator. Indeed, by adding a planar virtual joint at the bottom of the mobile base, it is possible to make any planner in the customized MoveIt! package aware that the entire robot can be relocated to obtain a collision-free trajectory and reach a desired EE's pose.

Since a mobile manipulator's base is expected to move in a 2D environment, a planar virtual joint could be simply added while configuring the package. Considering a reference frame located at the robot base, as depicted in Figure 2, a planar virtual joint, without specifying any other constraint, would give the robot the possibility of translating along the X and Y directions and rotating about the Z-axis. However, even if a planar virtual joint can be added to the current configuration in the planning framework, it is converted to a fixed one before planning, giving no effective contribution to this task.

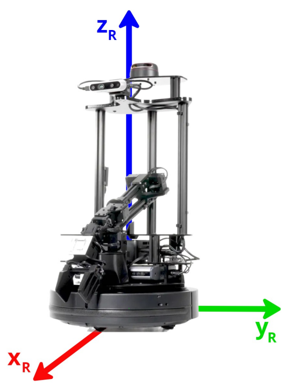


Fig. 2: Visualization of the reference frame R centered with the robot.

To overcome this problem, it is possible to create a customized virtual joint that effectively extends the robot's motion abilities by modifying the robot's URDF. Each customized joint defined in such a file adds 1 DOF, hence, in order to recreate the planar virtual joint defined by MoveIt!, three auxiliary joints have been added at the base of the mobile manipulator, and directly linked to the root frame of the robotic arm: (i) a prismatic joint allowing motion along the X-axis, (ii) a second prismatic joint attached at the end of the first one moving along the Y-axis, and (iii) a revolute joint able to rotate about the Z-axis. Using this modified URDF to generate the MoveIt! package, it is possible to request a path planning for the holistic system using the planning functionalities provided by the framework. Moreover, thanks to the explicit use of three joints to build a virtual planar one, it is possible to use their constraints to limit the movement of the robot within the workspace.

Using this approach, it is possible to easily adapt the planning framework to obtain an easy-to-use holistic planner that allows to employ the different planning algorithms. In addition, it leads to more harmonious and natural movements of the entire robot, if compared with the ones obtained from the combination of the plans coming from independent planners governing the mobile base and the arm.

C. Grasping pose generation

In Section II-B, we described the most common approaches concerning the grasping pose generation problem. Most of them make use of data coming from an RGB-D camera to feed NNs, hence they generate the optimal grasping point directly from what the robot can see. However, these algorithms usually require dedicated hardware (e.g., GPUs), which is not always available in the robots' onboard computers. To face this challenge, we designed a lightweight approach to generate the optimal grasping pose, that can be also used by low-resource robots. Our solution employs a YOLOv5 CNN [26], allowing not only to understand whether an object is visible or not in the current scene, but also to compute an estimation of its position with respect to the image frame. This latter information is provided as a bounding box (BB), uniquely characterized by the center, the height, and the width. This network has been chosen because it offers various architectures with different sizes and parameter counts; this flexibility allows users to select and employ the one that aligns better with their specific requirements.

The BB information of an object is further processed to approximate the center of mass (CoM) of the object, assumed to be equivalent to the BB's center; the actual mass distribution details are neglected, as incorporating them would involve coding for each specific object or training another network. The CoM computed is considered as the grasping point from which the object can be picked.

Using only the BB prediction, it is possible to obtain the computation of the grasping point in a 2D reference frame aligned with the image frame. To acquire its coordinates in the 3D world, integration with additional data from the camera is essential. This involves leveraging the depth image provided by the camera, which is processed to transform it into a 2D matrix. This latter data structure is accessed using the CoM's coordinates computed before to retrieve its corresponding depth value. As it is evident in Figure 3, the result of this operation is not only a grasping point for the robot's EE, but also an approximation of the object's 3D pose as a translation of the actual depth camera's pose.

D. Dangerous objects handling

In order to reduce the risk for humans collaborating with the mobile manipulator, it is important to correctly identify the most dangerous part of an item, and possibly modify the grasping point to adequately pick it. Grasping the object just from the part including its dangerous elements reduces the risk for humans to be hurt, since such unsafe elements are indirectly hidden. However, letting a robot to autonomously understand which is the most dangerous part of an object is a challenge that has proved very hard to face. To overcome this problem, we designed a flexible yet effective interface through which the robot exploits the collaboration with the human operator to individuate the dangerous part and to correct the

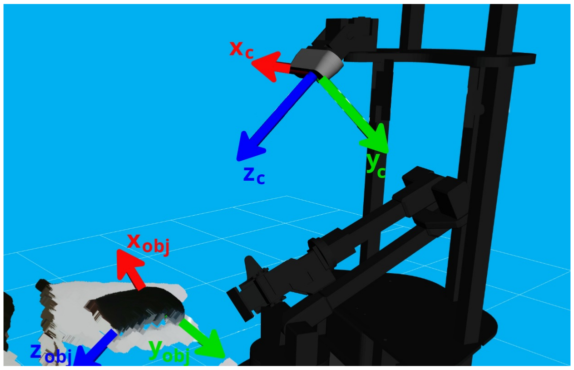


Fig. 3: Visualization of the estimated object's pose. The axis with subscript "c" indicates the reference frame of the depth camera, while those with subscript "obj" indicate the object's pose.

grasping point. Following an approach similar to what is done in Interactive Machine Learning (IML) [27], if the requested object to pick may hurt people, the grasping point is corrected by asking the human operator to indicate the most dangerous part of the item, by clicking on the image representing the last frame captured by the robot's camera. Using this feedback, the grasping point is modified, and the object is picked out directly from its most dangerous part, thus reducing the risks for the human operator. An example of the described interface is provided and discussed in Section IV-D.

IV. Experimental validation

A. Hardware and software setup

The LoCoBot WX250 mobile manipulator [22] was used to validate the proposed approach. It is composed of a Kobuki mobile platform with differential wheels, and a WidowX250 6-DOF manipulator with a parallel gripper. The robot senses the surrounding environment thanks to an RPLIDAR A2M8 (360° 2D LIDAR) and an Intel RealSense D435 (Stereo RGB-D), used for both manipulation and navigation tasks. The robot's PC has reduced computational capabilities (it has only 8 GB RAM), so all the computational and data processing tasks are assigned to the available Intel i3-CPU (8th Gen). Also, the lack of a dedicated GPU prevents the use of large NNs.

The proposed solution has been developed using ROS1, in particular ROS Noetic and its packages, to accommodate the real robot's compatibility requirements. Some of the packages used are the customized MoveIt! package described in Section III-B, the RTAB-Map and SlamToolbox packages for solving the SLAM problem, and the `move_base` package to control the mobile base.

B. Environment description

Mobile manipulators are usually intended to work in indoor environments shared with humans, in which objects are stored on shelves, drawers, and cabinets. An example of such a context, scaled to a single large room, is the Robotic Laboratory at Politecnico di Torino. It is an indoor and semi-structured environment, where robots and people work together, hence, there are both static and dynamic obstacles. The dimensions of the shelves and the items to be picked are compatible with the LoCoBot WX250 mechanical structural characteristics. To address the obstacle avoidance problem, the solution involves the utilization of an occupancy map generated using Octomap [28]. Notably, the occupancy map is not permanently stored, but it is reset at the beginning of each new run. This dynamic approach enables the system to adapt to new static obstacles that may emerge when the robot is in motion, ensuring effective obstacle detection and navigation. Moreover, the occupancy grid is also necessary while picking an item from a cabinet: this latter space is closed and may be cluttered, so having a precise description of the available free space is crucial to generate a suitable collision-free trajectory.

C. Experiments

The carried out experiments are focused on the pick-and-place pipeline, to understand its real speed and the limitations of the grasping actions. Before discussing the details of the experiments, it is important to illustrate the role of the holistic planner. The execution of the computed holistic plan would be possible only if a unique controller for both the mobile manipulator's parts is available. Nevertheless, robot producers usually provide two distinct controllers, specifically designed and tuned for the arm's servomotors and the mobile base. In addition, the modified robot description, which includes the three auxiliary joints needed to build a holistic planner, would allow the mobile base to move instantaneously along the lateral direction. This latter aspect would conflict with the differential drive constraints, thus affecting the motion of the LoCoBot used for the experiments. To overcome all these issues, the holistic approach has been adopted only during planning, while the motion of the two subsystems has been kept separate. Hence, the final pose of the mobile base is extracted from the trajectory obtained by the holistic planner and used by the `move_base` package as the desired goal. In this way, the whole robot is relocated, when it is necessary, to reach the grasping pose, and the motion control of the two subsystems is demanded to the robot's controllers.

The entire sequence of actions executed by the LoCoBot WX250 is depicted in Figure 4. The mobile manipulator performs the searching phase (in red) trying to locate the requested item inside the environment and to generate a grasping pose, and, if such object belongs to those categories of potentially dangerous items, the robot requires a feedback from the human to correct the predicted grasping point (in green). Then, the holistic motion planner is used to find a trajectory for the arm and the base together (purple). If a trajectory is found, the arm and the mobile base are moved to reach the computed grasping point (pick pipeline); after having successfully grasped the object, the place pipeline is executed to release the desired object inside the depot. Figure 5 shows the highlights of the experiment conducted in the real environment:

1) After receiving a request to pick an item and extracting the list of possible places where the item usually is, the first location is reached, and the object detection task

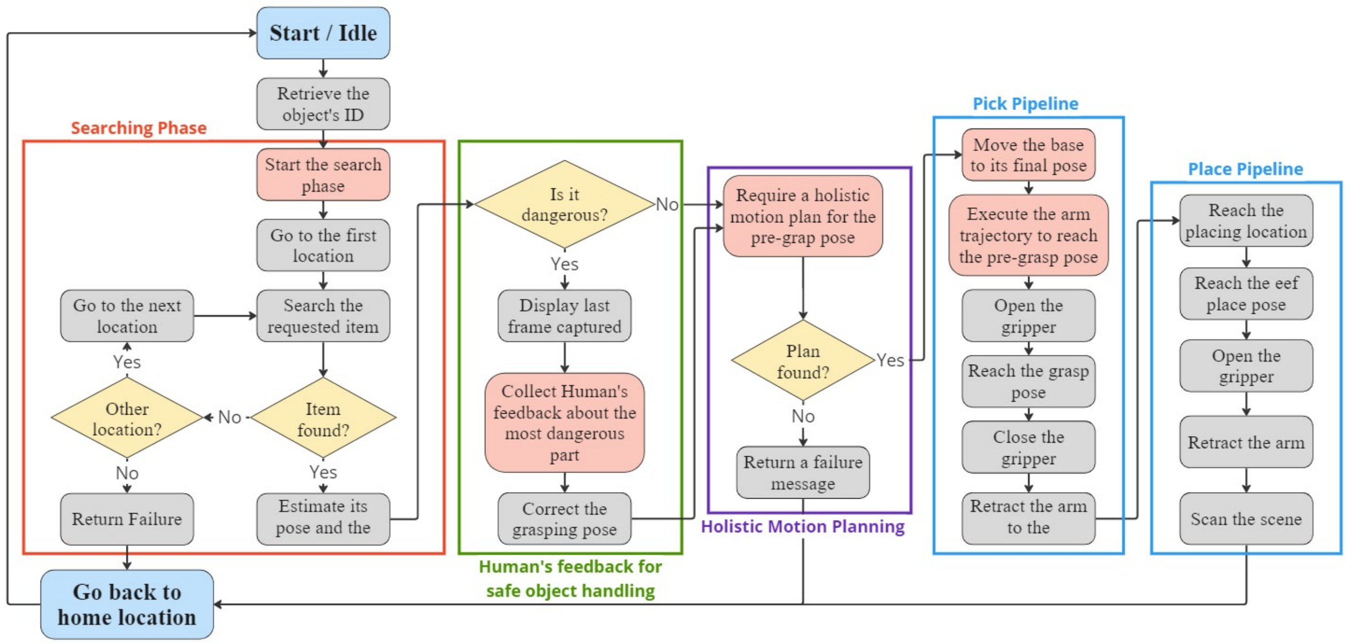


Fig. 4: Diagram describing the sequence of actions executed by the robot.

starts (Figure 5a). It is worth noting that to correctly scan the scene, the camera is moved in four different positions, so the object detection task is performed four times.

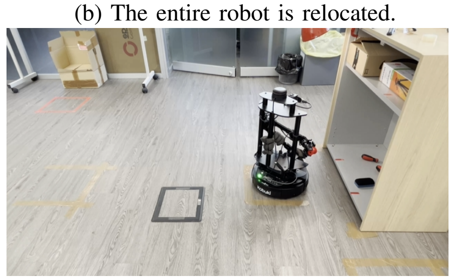
2) If the requested object is available in the current scene, the information about its BB is used to generate the grasping point.
3) The holistic planner is requested to find a collision-free trajectory to reach the EE's pre-grasp pose, which is located above the object (orthogonal to the plane where the item is placed), with the gripper pointing towards it.
4) The final pose of the mobile base is extracted and assigned as the desired goal for the mobile base's controller (Figure 5b).
5) After the mobile base has reached the correct position, the trajectory computed for the arm is executed and the pre-grasp pose is reached (Figure 5c).
6) Then, the real grasp pose is reached by the mobile manipulator and the gripper is closed (Figure 5d).
7) After having successfully grasped the desired item, the manipulator is retracted to a predefined safe pose, used

(a) The robot reaches a searching location.

(b) The entire robot is relocated.

(c) Arm moves to reach the pre-grasp pose.

(d) The EE reaches the grasp-pose.



(e) The gripper is closed to pick the item.

(f) The object is placed inside the depot.

Fig. 5: Visualization of the pick-and-place pipeline.

to avoid any possible and undesired collision of the arm with its surroundings (Figure 5e).

8) In the final phase, the mobile manipulator arrives to the designed place location and releases the item in a depot (Figure 5f).

A brief demo video showcasing the experimental validation is available at [29]. During the entire pipeline, two different planners from the OMPL are used:

1) The RRT* planner [30] is used to obtain a complete trajectory considering the mobile manipulator as a holistic system. The number of attempts available was set to 5, while the maximum planning time was set to 15 s;
2) The RRT-Connect algorithm [31] is instead used to find suitable trajectories for the arm only to go from the pre-grasp pose to the effective grasping one. In this case, the total number of planning attempts was 5, while the maximum time to find a solution was set to 10 s.

D. YOLOv5 version and dataset

Finding a suitable trade-off between the model complexity and the final performances, while considering the available computational resources, had a crucial role in determining the type of YOLOv5 CNN to use. Thanks to its low number of parameters and swift computation, the small version, YOLOv5s, has been chosen: the necessary predictions are achieved in approximately 121 ms, ensuring a rapid and lightweight object detection task, with an accuracy of the 93%. To effectively train our network, a customized dataset comprising images of objects commonly encountered in laboratories has been created. It contains about 800 RGB images of objects belonging to 15 different classes (Figure 6). Each image's label contains the expected BB in addition to the class label.

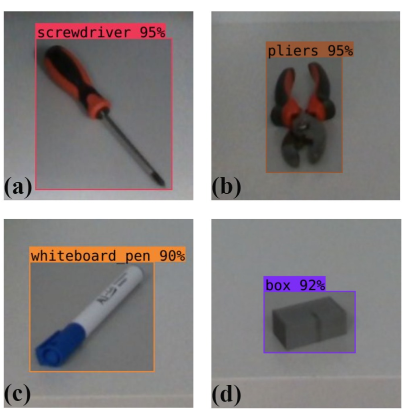


Fig. 6: Some examples of the detected bounding boxes and the class associated with them: (a) screwdriver, (b) plier, (c) whiteboard pen, (d) box.

E. Dangerous part identification feedback

After the object detection phase and before the grasp action, if the requested item belongs to the hard-coded list of dangerous ones, the robot requests a feedback from the human operator using the approach described in Section III-D. A visual representation of the last scene captured by the LoCoBot is provided to the user. The image depicts the object intended for grasping, identified by its BB, and the computed grasping point is indicated by a red dot. This information is communicated to the user, offering a clear visual reference for the robotic interaction. After the image is displayed, the user clicks on the point that represents the most dangerous one. The system computes the mean point between the predicted and user-indicated grasp points, using it as the actual grasping point (Figure 7).

Using such a flexible interface, the robot can correctly pick dangerous items from their most dangerous part, therefore, reducing the risk for the human operator and increasing safety. This can be very useful when the object must be given to the operator, instead of being released in a depot.

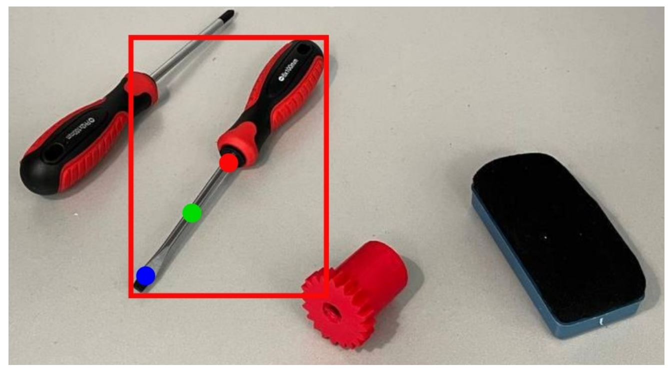


Fig. 7: Visualization of the predicted grasping point (red), the point clicked by the user (blue) and the resulting grasping point that will be used (green).

F. Results

The entire pipeline shown in Figure 4 is executed in about 2.5 minutes in an environment of approximately 20 m^2. Most of the time is spent on the robot's navigation, while the object detection task is quite quick, even if it is executed on a GPU-free system, and it is requested twice to mitigate the effect of the noise affecting the depth camera; the time required for this task, considering also the delay introduced to reduce the noise effect, is of about 10 s for 8 predictions (considering 4 camera positions, and requesting predictions twice). The RRT* has been used as a holistic planner nearly all the time and requires all the allowed 5 attempts of 15 s each to find a suitable collision-free trajectory, while the RRT-Connect planner for the robotic arm seldom uses all the attempts provided, and it is often able to find a motion plan in less than 10 s.

The interface used to collect the human feedback about the most dangerous part of an object resulted to be intuitive and easy to use, allowing one to correctly handle all the items.

The gripper's characteristics introduce challenges in grasping certain objects of the dataset. Long items, such as screwdrivers, with a CoM near one end, are often difficult to grasp successfully. Conversely, objects with a regular shape and a CoM aligned with the actual grasping point are generally picked correctly in most instances.

V. Conclusions and Future Works

The paper aims at enhancing a mobile manipulator's capabilities to serve as a safe smart assistant for humans, especially when handling hazardous objects. It introduces an easy-to-use holistic planner to optimize motion plans, expanding the manipulator's reachable space. This approach enables dynamic base positioning for grasping tasks, eliminating the need for predefined base poses from which it is possible to perform the grasping task. Further advancements could include the development of customized planners tailored for mobile manipulators to optimize base positions for specific end-effector poses. Noise affecting the depth data caused some errors in the correct prediction of the grasping pose. However, the proposed lightweight solution based on YOLOv5 CNN proved to be effective, giving more flexibility to the robots with limited computational capabilities. Moreover, the objects that the robot can pick are limited by its gripper, since they have different shapes and mass distribution, but the picking precision could be improved using point clouds with 3D-CNN instead of images augmented using depth data. Finally, regarding the dangerous parts handling, human-in-the-loop interaction is a preliminary step towards training the robot using IML algorithms. Following this fashion, using a learning-from-experience approach can make the robot more autonomous in this task, while reducing the effort needed to train it.

Acknowledgements

This study was carried out within the FAIR - Future Artificial Intelligence Research and received funding from the European Union Next-GenerationEU (PIANO NAZIONALE DI RIPRESA E RESILIENZA (PNRR) – MISSIONE 4 COMPONENTE 2, INVESTIMENTO 1.3 – D.D. 1555 11/10/2022, PE00000013). This manuscript reflects only the authors' views and opinions, neither the European Union nor the European Commission can be considered responsible for them.

References

[1] A. Hentout, M. Aouache, A. Maoudj, and I. Akli, "Human–robot interaction in industrial collaborative robotics: a literature review of the decade 2008–2017," *Advanced Robotics*, vol. 33, no. 15-16, pp. 764–799, 2019.

[2] L. Wang, R. Gao, J. Váncza, J. Krüger, X. V. Wang, S. Makris, and G. Chryssolouris, "Symbiotic human-robot collaborative assembly," *CIRP annals*, vol. 68, no. 2, pp. 701–726, 2019.

[3] P. D. Cen Cheng, F. Sibona, and M. Indri, "A framework for safe and intuitive human-robot interaction for assistant robotics," in *2022 IEEE 27th International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2022, pp. 1–4.

[4] P. Segura, O. Lobato-Calleros, A. Ramírez-Serrano, and E. G. Hernández-Martínez, "Safety assurance in human-robot collaborative systems: A survey in the manufacturing industry," *Procedia CIRP*, vol. 107, pp. 740–745, 2022.

[5] C. Weidemann, N. Mandischer, F. van Kerkom, B. Corves, M. Hüsing, T. Kraus, and C. Garus, "Literature Review on Recent Trends and Perspectives of Collaborative Robotics in Work 4.0," *Robotics*, vol. 12, no. 3, p. 84, 2023.

[6] J. Gros, D. Zatyagov, M. Papa, C. Colloseus, S. Ludwig, and D. Aschenbrenner, "Unlocking the Benefits of Mobile Manipulators for Small and Medium-Sized Enterprises: A Comprehensive Study," *Procedia CIRP*, vol. 120, pp. 1339–1344, 2023.

[7] P. D. Cen Cheng, M. Indri, F. Maresca, A. Ragazzo, and F. Sibona, "Dynamic path planning in human-shared environments for low-resource mobile agents," in *2023 IEEE 32nd International Symposium on Industrial Electronics (ISIE)*, 2023, pp. 1–6.

[8] T. Sandakalum and M. H. Ang Jr, "Motion planning for mobile manipulators—a systematic review," *Machines*, vol. 10, no. 2, p. 97, 2022.

[9] H. Engemann, S. Du, S. Kallweit, P. Cönen, and H. Dawar, "OMNIVIL—An Autonomous Mobile Manipulator for Flexible Production," *Sensors*, vol. 20, no. 24, 2020. [Online]. Available: https://www.mdpi.com/1424-8220/20/24/7249

[10] P. D. Cen Cheng, M. Indri, F. Maresca, A. Ragazzo, and F. Sibona, "A software architecture for low-resource autonomous mobile manipulation," in *2023 IEEE 28th International Conference on Emerging Technologies and Factory Automation (ETFA)*. IEEE, 2023, pp. 1–8.

[11] N. Castaman, E. Pagello, E. Menegatti, and A. Pretto, "Receding horizon task and motion planning in changing environments," *Robotics and Autonomous Systems*, vol. 145, p. 103863, 2021.

[12] S. Saoji and J. Rosell, "Flexibly configuring task and motion planning problems for mobile manipulators," in *2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, vol. 1, 2020, pp. 1285–1288.

[13] A. Rastegarpanah, H. C. Gonzalez, and R. Stolkin, "Semi-autonomous behaviour tree-based framework for sorting electric vehicle batteries components," *Robotics*, vol. 10, no. 2, p. 82, 2021.

[14] J. Haviland, N. Sünderhauf, and P. Corke, "A holistic approach to reactive mobile manipulation," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 3122–3129, 2022.

[15] K. Liu, J. Sui, N. Yue, and S. Liu, "Path planning method of mobile manipulator based on the representation space," in *2016 IEEE International Conference on Mechatronics and Automation*, 2016, pp. 322–326.

[16] J. Haviland, N. Sünderhauf, and P. Corke, "A holistic approach to reactive mobile manipulation," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 3122–3129, 2022.

[17] C. Wang, Q. Zhang, Q. Tian, S. Li, X. Wang, D. Lane, Y. Petillot, and S. Wang, "Learning mobile manipulation through deep reinforcement learning," *Sensors*, vol. 20, no. 3, p. 939, 2020.

[18] L. Pinto and A. Gupta, "Supersizing self-supervision: Learning to grasp from 50K tries and 700 robot hours," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 3406–3413.

[19] D. Morrison, P. Corke, and J. Leitner, "Closing the loop for robotic grasping: A real-time, generative grasp synthesis approach," *arXiv preprint arXiv:1804.05172*, 2018.

[20] Z. He, N. Chavan-Dafle, J. Huh, S. Song, and V. Isler, "Pick2Place: Task-aware 6DoF Grasp Estimation via Object-Centric Perspective Affordance," *arXiv preprint arXiv:2304.04100*, 2023.

[21] Y. Yang, W. Zhai, H. Luo, Y. Cao, J. Luo, and Z.-J. Zha, "Grounding 3D Object Affordance from 2D Interactions in Images," *arXiv preprint arXiv:2303.10437*, 2023.

[22] T. Robotics, "LoCoBot WX250 with 6 DOF Arm (with Lidar)," [Accessed February 2024]. [Online]. Available: https://docs.trossenrobotics.com/interbotix_xslocobots_docs/specifications/locobot_wx250s.html

[23] R. F. Cavelli, "Mobile Manipulator as Smart Human Assistant for Safe Objects Handling," 2023.

[24] "Experimental test video demo," https://github.com/Saro0800/296846-MasterThesis.git, [Online; accessed April 2024].

[25] D. Coleman, I. Sucan, S. Chitta, and N. Correll, "Reducing the barrier to entry of complex robotic software: a moveit! case study," *arXiv preprint arXiv:1404.3785*, 2014.

[26] G. Jocher, A. Chaurasia, A. Stoken, J. Borovec, Y. Kwon, K. Michael, J. Fang, Z. Yifu, C. Wong, D. Montes *et al.*, "ultralytics/yolov5: v7.0-yolov5 sota realtime instance segmentation," *Zenodo*, 2022.

[27] A. Ayub, C. Nehaniv, and K. Dautenhahn, "Interactive Continual Learning Architecture for Long-Term Personalization of Home Service Robots," *arXiv preprint arXiv:2403.03462*, 2024.

[28] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "OctoMap: An efficient probabilistic 3D mapping framework based on octrees," *Autonomous robots*, vol. 34, pp. 189–206, 2013.

[29] "Experimental test video demo," https://youtu.be/dxks09ZzC6I, [Online; accessed April 2024].

[30] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *The international journal of robotics research*, vol. 30, no. 7, pp. 846–894, 2011.

[31] J. J. Kuffner and S. M. LaValle, "RRT-connect: An efficient approach to single-query path planning," in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*, vol. 2. IEEE, 2000, pp. 995–1001.