Mobile manipulator base placement optimization using an ellipsoidal reachability model

Mobile manipulator base placement optimization using an ellipsoidal reachability model

Rosario Francesco Cavelli, Pangcheng David Cen Cheng, Marina Indri
Department of Electronics and Telecommunications
Politecnico di Torino
Corso Duca degli Abruzzi, 24, 10129, Torino, Italy
{rosario.cavelli, pangcheng.cencheng, marina.indri}@polito.it

***Abstract*—The mobile manipulator's base positioning is strategic to optimize the execution of different tasks within a specific workspace. Base placements provided by the human intuition yield suboptimal results, and existing methods are usually based on computationally expensive maps built to represent the distribution of the manipulator's dexterity or the base positions effective for reaching a specific target, but not both simultaneously. This paper proposes a novel optimization-based method for mobile manipulator's base placement, focused on a compact mathematical model of the manipulator reachability space. By representing it through two concentric ellipsoid equations, such a model enables efficient evaluation of end-effector reachability while guiding the base placement. This model is exploited in the optimization problem to autonomously reposition the mobile base, ensuring pose reachability with high dexterity and minimizing collision risk. The approach is validated experimentally using a mobile manipulator in a real-world laboratory and assigning target poses with varying difficulty levels. The results demonstrate the effectiveness of the obtained base poses and the adaptability of the method to different scenarios.**

***Index Terms*—Mobile manipulators, Robot base placement optimization, Reachability space**

I. Introduction

Mobile manipulators (MMs) are increasingly deployed across many scenarios and applications. Their ability to navigate and interact with the environment makes them ideal for tasks such as pick-and-place operations, inspection activities, and object transportation. These tasks typically involve two actions: the robot first reaches the operational site and then executes the required manipulation. Therefore, determining an optimal pose for the mobile base is crucial, as it can significantly influence the task's success. Early strategies for MM placement heavily relied on human intuition and trial and error. Although humans possess an intuitive understanding of their environment and their manipulation capabilities, task-oriented robot placement based on intuition often results in suboptimal positioning, frequent base repositioning, or task failure. To enhance robot autonomy, some studies tried to replicate human behavior through Artificial Intelligence (AI) techniques, relying primarily on learning-by-demonstration and reinforcement learning. While effective, these approaches are computationally demanding and often difficult to generalize across different platforms. More recent efforts have focused on enabling MMs to understand their own manipulation capabilities. Reachability Maps (RMs) are used to sample the 3D space, identifying regions reachable by the end-effector (EE) and evaluating dexterity through manipulability indices [1]. Moreover, Inverse Reachability Maps (IRMs) are derived from RMs to suggest optimal base placements. However, these methods are time-consuming in the construction of the model and the base placement computation [2]. Alternatively, optimization-based methods have been proposed, where the optimal base placement is found by optimizing a task-specific metric. Nevertheless, these approaches typically do not explicitly model the robot's manipulation capabilities, which can limit flexibility and autonomy [3]. To address these limitations, in this paper, a unified novel model of the Reachability Space (RS) of MMs based on ellipsoid equations is proposed. Unlike traditional approaches, this unified model offers a compact and continuous formulation, which can jointly represent EE reachability and dexterity, and guide autonomous base placement. This RS model is then embedded within a proper optimization problem to determine base poses that inherently account for manipulation performance and collision avoidance, leveraging the redundancy introduced by the mobile base. The proposed approach can be adapted to a wide range of tasks and customized to specific operational requirements.

The paper is organized as follows: Section II provides an overview of relevant related works, while the mathematical model of the manipulator RS and the definition of the optimization problem to position the robot base are presented in Section III and Section IV, respectively. Section V presents the experimental setup along with the results, and finally, Section VI concludes the paper, highlighting possible improvements.

II. Related works

The problem of determining optimal base poses for different kinds of robots has been addressed by exploiting diverse methodologies. In early studies, positioning robots to perform different tasks depended on humans. The operators had to manually select robot base positions using their intuition and expertise related to the task to complete, usually going through an iterative procedure based on trials and errors [4]. Although some more recent studies tried to mix human intuition with optimization problems [5], these approaches still suffer from inefficiency, suboptimal solutions, and scalability issues for large workpieces, as noted in [6]. A first step towards the

autonomous selection of robot base poses has been done by building a model of the RS of a manipulator. The Reachability Map (RM), first introduced in [7], represents a well-known method to build such a model. RMs give a clear understanding of which EE poses can be reached from a given position of the robot's base, and with which level of dexterity. With the aim of finding the optimal base pose for manipulators, the concept behind RMs has been inverted, leading to Inverse Reachability Maps (IRMs) [8]. These maps suggest where to position the base of a manipulator in order to reach the desired EE pose. Makhal et al. [9] applied the concept of inverse reachability to an MM and used an IRM to let the robot autonomously find a suitable placement of the mobile base to reach a given EE pose. A drawback of RM and IRM is that they are static and ignore environmental changes. To overcome this limitation, a Dynamic RM has been proposed in [10], which is able to represent any updates on the RS of a manipulator. The authors in [11] proposed a neural network (named Inv-Reach Net) that updates the IRM of a manipulator to reflect the changes in the working environment. Such a network tries to predict how the robot's reachability is influenced and how its base placement should be consequently updated. The knowledge of the manipulation ability described by both the RM and IRM has also been used in conjunction with optimization approaches. In [12], an RM was used to speed up a proper optimization problem designed to optimize the base placement of a fixed-base robot and camera mounting position. The studies conducted in [13], [14] aim to find the optimal base poses for MMs as a result of an optimization problem. The objective functions were designed to minimize the time required to complete manipulation tasks, by reducing navigation distance and maximizing the manipulability of the EE. The different positions of the robot were inferred using an IRM. The main drawback of the methods based on the use of RMs and IRMs is that they can be computationally onerous and really time-consuming. Some alternative approaches try to address the problem of autonomously relocating an MM through constrained optimization techniques. Yu et al. [6] developed an optimization problem to relocate a painting robot to ensure the best manipulability while considering joint-level constraints and singularity avoidance. An optimization framework was proposed in [15] to maximize the total stiffness resulting from proper robot positioning and needed for machining large parts, while considering different constraints like joint range, speed, singularity, and collision avoidance. Authors in [16] developed an optimization problem whose objective was to position a humanoid robot minimizing EE pose errors. The constraints of the problem resulted in a task-driven mathematical model of the robot RS. However, the reliance on predefined constraints can limit the adaptability of the methods to tasks that were not explicitly considered during the problem formulation. A more explicit mathematical model of the RS is employed in some other studies, proposing also an alternative to RMs. In [17], the workspace is modeled geometrically by dividing it into multiple tori, which are used to approximate the reachable areas of the MM. Such geometrical representation is used for an optimization problem that aims at minimizing the number of robot repositionings during manipulation tasks. Zbiss et al. [18] faced the problem of determining the base location of multiple fixed-base manipulators working simultaneously to paint a car. The workspace of each robot was represented using an ellipsoidal shape. These shapes were then used to optimally position the robot to maximize the robot's coverage and minimize the risk of collisions between them. Mathematical models of manipulators' RS can be effectively used to find optimal base poses, but they may require a high level of accuracy and several observations on the task to perform. Recent studies have integrated Reinforcement Learning (RL) with reachability reasoning to enhance mobile manipulation. In [4], BASENET combines Graph Neural Networks and Layered Learning to compute optimal base poses and action sequences without relying on precomputed RMs. Similarly, authors in [19] proposed an RL approach where a Base Pose Prediction Network (BP-Net) predicts efficient base poses and pre-grasp configurations, minimizing task execution time. In [20], reachability priors are incorporated into RL, using an IRM to guide base placement decisions and improve learning efficiency. However, RL-based methods often require extensive training time, rely heavily on simulation environments, and may struggle to generalize to unseen or highly dynamic scenarios. A MATLAB toolbox was developed in [21] to visualize and analyse the reachability of robotic arms. However, this toolbox seems to be limited only to fixed-base manipulators and does not indicate all the reachable points of the EE.

III. Mathematical model of the RS

The core of the novel method proposed in this paper is an optimization problem that leverages a closed-form mathematical model of the robot's optimal RS. This model extends the procedure introduced in our previous work [22], where the goal was to define an ellipsoid equation containing the entire RS of a manipulator. In contrast, the approach presented here focuses on obtaining a mathematical model that encloses only the subset of reachable points associated with the highest dexterity. To this end, a set of target poses is first generated and used to construct the RM of the manipulator. This map serves as the input to an optimization problem, whose solution yields the desired mathematical model.

A. Targets pose generation for the EE

The starting input is a point cloud representing the positions that the robot's EE can reach. This 3D data structure is obtained by applying our method presented in [22], which uses only the robot's kinematic model defined in the corresponding URDF (Unified Robot Description Format) file [23]. Building upon this foundation, in this paper, a reachability measure is assigned to each point in the cloud, indicating the number of poses it can be reached with. A straightforward way to compute such a metric is to define a set of different orientations and to compute with how many of them a point can be reached by the manipulator's EE. To address this task, an approach similar to the one presented in [9] has been exploited. A sphere with an arbitrary radius is generated around each of the N points

in the RS cloud. The surface is sampled using the *Fibonacci algorithm* [24], which is a deterministic procedure that aims at obtaining n_s equally spaced samples. Such samples are used to generate the roll-pitch-yaw (RPY) angles describing the orientation of the EE target poses. To compute the desired angles, the vector going from each point on the surface towards the center of the sphere is defined as:

$$\mathbf{v} = [x_0 - x_s, y_0 - y_s, z_0 - z_s]^T, \tag{1}$$

where $\mathbf{x_0} = [x_0, y_0, z_0]^T$ is the center of the sphere, and $\mathbf{x_s} = [x_s, y_s, z_s]^T$ is the surface sample. The vector $\mathbf{v}$ is then normalized to obtain the versor $\mathbf{v_n} = [v_{n,1}, v_{n,2}, v_{n,3}]^T$:

$$\mathbf{v_n} = \frac{\mathbf{v}}{||\mathbf{v}||_2}. \tag{2}$$

The RPY angles describing the orientation of $\mathbf{v_n}$ with respect to the axes of the reference frame centered in the sphere are computed as:

$$roll = 0, \tag{3}$$

$$pitch = -\operatorname{atan2}(v_{n,3}, \sqrt{{v_{n,2}}^2 + {v_{n,1}}^2}), \tag{4}$$

$$yaw = \operatorname{atan2}(v_{n,2}, v_{n,1}). \tag{5}$$

These equations are applied to each sphere sample, yielding a set of n_s RPY angles. They are combined with the coordinates of the sphere center to generate all the EE target poses required to compute the reachability measure for that point. This process can be iteratively applied to each point in the RS cloud to compute its reachability measure. However, the deterministic nature of the adopted sampling algorithm allows a strong optimization of the computation process. Once the number of desired surface samples n_s has been set, their coordinates with respect to the reference frame fixed in the center of each sphere are always the same, and so will be the set of RPY angles. Due to these properties, instead of sampling N different spheres, each centered in one point of the RS cloud, it is possible to sample the surface of only one sphere, with an arbitrary center, and compute the RPY angles corresponding to that sample. Each reachable point is then coupled with the computed triplets of RPY angles, obtaining a total of n_s EE poses per point, as shown in Figure 1.

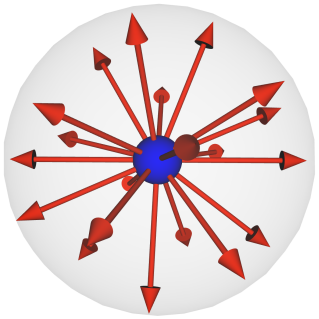


Fig. 1: Poses (in red) generated by combining a reachable point (in blue) with all RPY orientations obtained by sampling the sphere surface (in grey).

B. Reachability measure

Once the n_s EE poses have been computed for each point in the RS cloud, it is possible to compute the reachability measure of such a point. For each pose, the Inverse Kinematics (IK) problem is solved using the PyKDL library [25], a Python library for robot kinematics and dynamics computations. Solutions of the different IK problems are also checked for self-collision using the PyBullet library [26], widely adopted for its physics-based simulation and control features. The value of the reachability measure assigned to each point is equal to the number of EE target poses for which a collision-free IK solution exists. By using this method, the reachability measure of each point varies from 0 to n_s, meaning that the given point is reachable with none or all the different orientations considered, respectively. Figure 2 shows the RM for a LoCoBot WX250 [27] obtained using the described procedure.

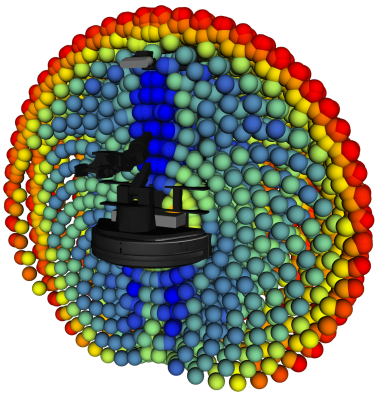


Fig. 2: Reachability map obtained for the LoCoBot WX250. Each reachable point is represented with a color that recalls its reachability measure: the highest in blue, the lowest in red.

C. Mathematical model of the optimal RS

The obtained RM is used as an input to a proper optimization problem, whose output is the desired mathematical model. It is based on the equations of two concentric ellipsoids, such that the space between them encloses the part of the RS with the highest reachability measure. The equation of each ellipsoid can be written in the following form:

$$\left(\frac{x - x_c}{A}\right)^2 + \left(\frac{y - y_c}{B}\right)^2 + \left(\frac{z - z_c}{C}\right)^2 = 1, \tag{6}$$

where x_c, y_c, $z_c \in \mathbb{R}$ are the coordinates of the center, and A, B, $C \in \mathbb{R}^+$ are the lengths of the semi-axes.

To obtain the ellipsoid equations, the optimization problem described in our early work [22] has been extended. The main differences concern the number of optimization variables and the objective function. In our early study, only one ellipsoid was considered to represent the entire RS, and the optimization variables were the six parameters in (6); in addition, all the points in the RS cloud had the same weights. Here, nine optimization variables are used: the coordinates

of the center $x_c, y_c, z_c \in \mathbb{R}$, the axis of the outer ellipsoid $A_{out}, B_{out}, C_{out} \in \mathbb{R}^+$, and the axis of the inner ellipsoid $A_{inn}, B_{inn}, C_{inn} \in \mathbb{R}^+$. Moreover, different weights are assigned to each point in the RS cloud to select only the ones with the highest reachability measure. The optimization problem can be defined as:

$$\begin{aligned} \underset{\substack{x_c,y_c,z_c,\\ A_{out},B_{out},C_{out},\\ A_{inn},B_{inn},C_{inn}}}{\arg\min} \quad & \gamma_{wt}\left(\sum_{\mathbf{p}_i \in \mathcal{P}_{out}} w_{\mathbf{p}_i} - \sum_{\mathbf{p}_j \in \mathcal{P}_{inn}} w_{\mathbf{p}_j}\right) + \\ & + \gamma_{vol}\left(V_{ell,out} + V_{ell,inn}\right) \\ s.t.: \quad & A_{out}, B_{out}, C_{out} > 0, A_{inn} \leq A_{out}, \\ & B_{inn} \leq B_{out}, C_{inn} \leq C_{out}. \end{aligned} \tag{7}$$

The first part of the objective function is the algebraic sum of the weights $w_{\mathbf{p}_i}$ and $w_{\mathbf{p}_j}$ of all the points contained in sets $\mathcal{P}_{out}$ and $\mathcal{P}_{inn}$, weighted by γ_{wt}. Denoting with $\mathcal{P}$ the set of all the points $\mathbf{p}$ building the RS point cloud, $\mathcal{P}_{out}$ and $\mathcal{P}_{inn}$ are defined as:

$$\mathcal{P}_{out} = \left\{\mathbf{p}_i \in \mathcal{P} \mid (\mathbf{p}_i - \mathbf{x_c})^T \mathbf{E_{out}} (\mathbf{p}_i - \mathbf{x_c}) \leq 1\right\}, \tag{8}$$

$$\mathcal{P}_{inn} = \left\{\mathbf{p}_j \in \mathcal{P} \mid (\mathbf{p}_j - \mathbf{x_c})^T \mathbf{E_{inn}} (\mathbf{p}_j - \mathbf{x_c}) \leq 1\right\}, \tag{9}$$

where $(\mathbf{p}_i - \mathbf{x_c})^T \mathbf{E} (\mathbf{p}_i - \mathbf{x_c}) \leq 1$ is the matrix form of the ellipsoid equation. The matrix $\mathbf{E}$ is defined as:

$$\mathbf{E} = \begin{bmatrix} \frac{1}{A^2} & 0 & 0 \\ 0 & \frac{1}{B^2} & 0 \\ 0 & 0 & \frac{1}{C^2} \end{bmatrix}, \tag{10}$$

where A, B and C are the semi-axes of the ellipsoid. $\mathcal{P}_{out}$ and $\mathcal{P}_{inn}$ identify the two sets of points contained inside the outer and the inner ellipsoid, respectively. The weight $w_{\mathbf{p}_i}$ of each point $\mathbf{p}_i$ is not directly equal to its reachability measure, but it is computed by adapting such a value to the minimization fashion of the problem, as:

$$w_{\mathbf{p}_i} = \left(\frac{1}{N}\sum_{j=1}^{n_s} j \cdot N_j\right) - m_i, \tag{11}$$

where N is the number of total points in the RS point cloud, n_s is the number of different orientations considered to compute the reachability measure of each point, N_j is the number of points in the cloud with a reachability measure equal to j, and m_i is the reachability measure of $\mathbf{p}_i$. The term within parentheses in (11) represents the weighted mean of the reachability measures assigned to each point. Using such an equation, a negative weight is assigned to points with a reachability measure higher than the weighted mean, while a positive weight is attributed to those with a lower value. A more detailed analysis of the first part of the objective function shows that the contribution given by the sum of the weights of the points in $\mathcal{P}_{out}$ is positive, whereas those associated with points in $\mathcal{P}_{inn}$ contribute with a negative sign. Consequently, due to the minimization nature of the problem, points with negative weights are assigned to $\mathcal{P}_{out}$, while points with positive weights are assigned to $\mathcal{P}_{inn}$. Once the optimization process is terminated, the inner ellipsoid will enclose only points located near the center with low reachability measures (i.e., with positive weights). In parallel, the outer ellipsoid will encompass points with high reachability measures (negative weights), while excluding those positioned near the boundary of the point cloud that exhibit low reachability values.

The second part of the objective function accounts for the volumes of both the outer and inner ellipsoids. This term ensures that the ellipsoids accurately fit the intended regions of the reachability space and prevents them from becoming excessively large. The mathematical model obtained for the LoCoBot WX250 is shown in Figure 3.

IV. Optimization problem to reposition a mobile manipulator

This section introduces the optimization problem formulated to reposition the base of the mobile manipulator.

A. Preliminaries

To support the formulation of the problem, three relevant reference frames are defined:

- $\mathcal{R}_B$ is the reference frame attached to the mobile base. Its x-axis is in front of the robot.
- $\mathcal{R}_{ell}$ is the reference frame with origin fixed in the center of the ellipsoids.
- $\mathcal{R}_{EE}$ is the reference frame describing the desired EE target pose. The coordinates of its origin and the orientation of its axes are defined with respect to the world fixed frame $\mathcal{R}_0$.

Figure 4a shows the defined reference frames. The axes of $\mathcal{R}_{ell}$ are aligned with those of $\mathcal{R}_B$, allowing to describe the relation between $\mathcal{R}_B$ and $\mathcal{R}_{ell}$ using a pure translation homogeneous transformation matrix T^B_{ell}. This matrix enables the conversion between the coordinates of the two frames.

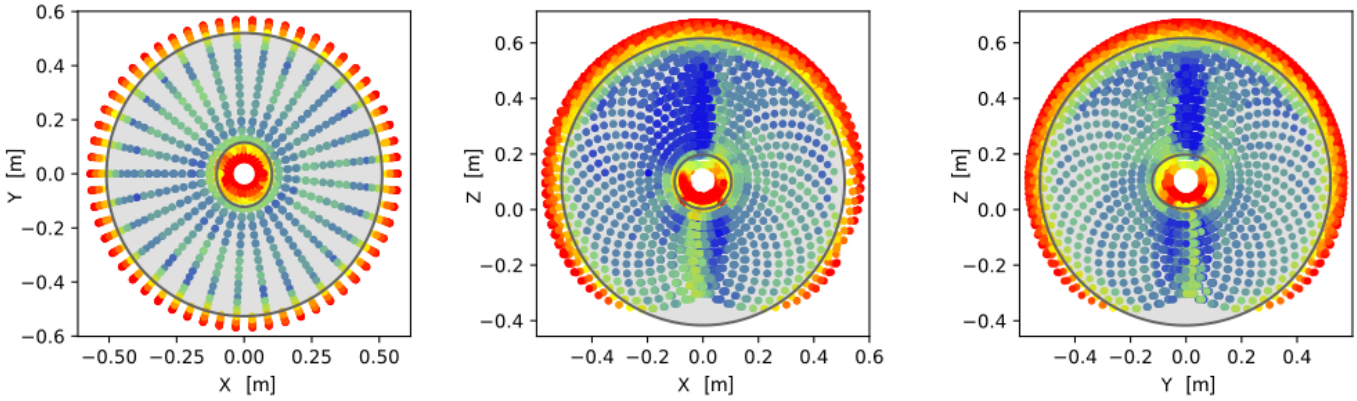
Another crucial element for the optimization problem is a representation of the environment where the robot moves. To this aim, the set $Obst$ is defined. It represents all the physical objects and obstacles of the environment, and each item within it is defined as a point cloud. In other words, $Obst$ itself is a point cloud representing the robot's surroundings. In this way, it is easier to check which objects (or parts of them) are included in the RS mathematical model and may cause collisions.

B. Problem definition

The optimization problem designed to position the base of an MM is defined as:

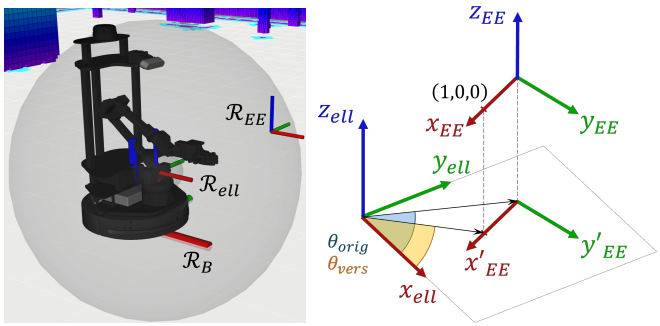
$$\begin{aligned} \underset{\mathbf{x},\theta}{\arg\min} \quad & \theta_{orig} + \theta_{vers} + \alpha|Coll| \\ s.t.: \quad & (\mathbf{x_{EE}} - \mathbf{x})^T \mathbf{E_{out}} (\mathbf{x_{EE}} - \mathbf{x}) \leq 1, \\ & (\mathbf{x_{EE}} - \mathbf{x})^T \mathbf{E_{inn}} (\mathbf{x_{EE}} - \mathbf{x}) > 1, \\ & \theta \in [0, \pi], \end{aligned} \tag{12}$$

where $\mathbf{x} \in \mathbb{R}^{3\times 1}$ and $\theta \in \mathbb{R}$ are the optimization variables, representing the coordinates of the origin of $\mathcal{R}_{ell}$ and the rotation of this reference frame about the z-axis, respectively, both defined with respect to the world fixed frame; $\mathbf{x_{EE}} \in \mathbb{R}^{3\times 1}$ is the position of the EE target. In the objective function,



(a) (b) (c)

Fig. 3: Visualization of the mathematical model obtained for the LoCoBot WX250 RS. Sections parallel to the XY, XZ, and YZ planes, and containing the center of the ellipsoids, are shown in subfigures (a), (b), and (c), respectively. The two ellipsoid equations and the space between them are visualized in light grey.



(a) (b)

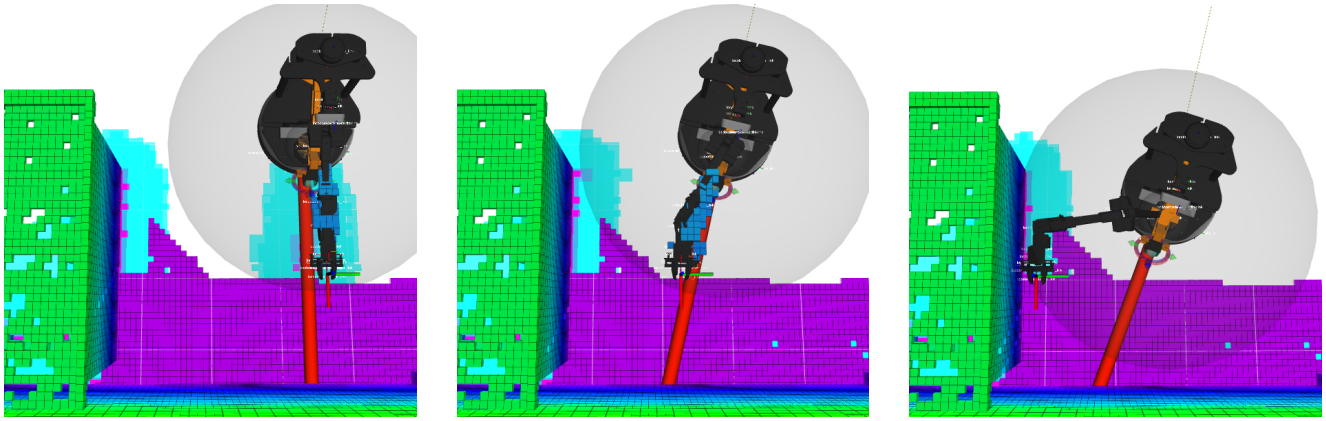
Fig. 4: (a) Mathematical model describing the optimal RS (light grey ellipsoid) together with the reference frames used for the optimization problem. (b) Projection of R_{EE} on the XY plane of R_{ell} and definition of the angles θ_{orig} and θ_{vers}.

$\theta_{orig} \in \mathbb{R}$ and $\theta_{vers} \in \mathbb{R}$ are the angles between the x-axis of $\mathcal{R}_{ell}$ and the projection on its XY plane of the vectors representing, respectively, the origin of $\mathcal{R}_{EE}$, and the point of coordinates (1,0,0) in $\mathcal{R}_{EE}$, as later discussed (Figure 4b). The last term $|Coll| \in \mathbb{R}$ represents the cardinality of the following set:

$$Coll = \left\{ \mathbf{p}_i \in Obst \,|\, (\mathbf{p}_i - \mathbf{x})^T \mathbf{E_{out}} (\mathbf{p}_i - \mathbf{x}) \leq 1 \right\}. \quad (13)$$

It contains points $\mathbf{p}_i$ of the objects in the $Obst$ set that are inside the manipulator RS, and may cause collisions. The factor $\alpha \in \mathbb{R}^+$ is used to convert $|Coll|$ from a scalar value to an angle, and it allows to adjust its impact on the pose correction.

Using the coordinates of the $\mathcal{R}_{ell}$ origin as optimization variables instead of those of $\mathcal{R}_B$ implies that the optimal position obtained from the minimization problem is not directly the position of the mobile base, but the one of the concentric ellipsoids. Nevertheless, the way $\mathcal{R}_{ell}$ and $\mathcal{R}_B$ are defined allows converting the coordinates of $\mathcal{R}_{ell}$ origin into those of the $\mathcal{R}_B$ one. In addition, the orientation of the axes of these two reference frames is the same; thus, the optimal value of θ obtained can be applied to both of them. The advantage of considering the coordinates of $\mathcal{R}_{ell}$ is that they can be simply used in both the constraints and the objective function definitions, avoiding redundant computation that would be necessary using $\mathcal{R}_B$. It is possible to notice that the ellipsoid equations defining the mathematical model of the RS are used as constraints for the proposed optimization problem. Indeed, the target EE is constrained to be inside the outer ellipsoid and outside the inner one. In this way, it is possible to ensure that it is reachable from the current mobile base pose and with a high level of dexterity. The objective function is designed to explicitly exploit the redundancy introduced by the mobile base, reducing the risk of collisions while moving the manipulator. The additional degrees of freedom (DOFs) typically allow to set the position of the mobile base on the ground and its rotation about an axis that is orthogonal to it. The optimization problem presented here aims at finding an optimal combination of them to reach the desired EE pose. The position of the base is constrained by the fact that the desired EE pose must be inside the mathematical model obtained, while no particular constraints are generally applied to the base orientation. Still, some requirements can be designed considering the typical tasks that an MM has to perform, e.g., object handling and scene inspection. Most of these operations are executed in front of the robot, and for this reason, the objective function of the proposed optimization problem has been designed to have the desired EE pose right in front of the robot. This has been achieved considering the two angles θ_{orig} and θ_{vers} as part of the objective function. By minimizing them, the optimal value of θ will lead to have the x-axes of $\mathcal{R}_{EE}$ and $\mathcal{R}_{ell}$ aligned one another, and consequently to the one of $\mathcal{R}_B$. In this way, the target EE pose will be in front of the robot. The final term of the objective function, $\alpha|Coll|$, serves to minimize the risk of collisions that may occur during the manipulator's motion. Including such a term



(a) (b) (c)

Fig. 5: Visualization of the MM base poses obtained through the optimization algorithm to reach an EE pose that is (a) far, (b) near, and (c) close to the obstacles.
The environment is represented as a set of colored voxels, while the MM optimal RS is depicted by the light gray ellipsoid. The thick red line indicates the x-axis of the base frame R_B, and the thinner one represents the x-axis of the EE frame R_{EE}.

in the objective function enables slight adjustments to the mobile base pose, facilitating the achievement of the desired EE pose while avoiding collisions.

V. Experimental validation

This section presents the experiments to validate the proposed optimization algorithm and analyzes the results. The LoCoBot WX250 [27] mobile manipulator has been used for an experimental campaign in a real-world laboratory environment, where different EE poses have been defined with varying levels of challenge. The code has been developed using Python 3.8.10 along with ROS1 Noetic and is available at [28]. To build the $Obst$ set representing the world environment, the OctoMap library [29] has been used. Consequently, the $|Coll|$ term in the objective function is computed as the number of voxels inside the ellipsoids modeling the manipulator RS.

A. Adaptability to the presence of obstacles

As discussed in Section IV, the proposed optimization problem aims at determining a base pose for the MM that enables reaching a desired EE target while accounting for potential collisions. Some experiments were performed to evaluate the method's ability to adapt the base pose to environmental conditions. In these experiments, a cluttered area of the environment was selected, and several EE target poses were defined within it. The targets were initially positioned far from obstacles and progressively moved closer, while maintaining a constant orientation. Figure 5 illustrates the behavior of the MM as the EE goal progressively moves closer to obstacles:

- Initially, the EE target is positioned far from the obstacles. The optimal base pose found yields having the x-axes of the base frame R_B and the EE frame R_{EE} aligned, as shown in Figure 5a. This demonstrates that, in the absence of environmental constraints, the optimization algorithm selects a base pose that positions the target EE pose directly in front of the robot.
- As the EE target approaches the obstacle, the manipulator adjusts its configuration, resulting in a slight misalignment between the x-axes of R_B and R_{EE} to avoid collisions (Figure 5b).
- When the EE target is located very close to the obstacle, the manipulator significantly rotates its base to maintain reachability while minimizing the number of voxels within its RS, leading to a more pronounced misalignment between the two frames' x-axes. Figure 5c depicts this situation.

This sequence clearly demonstrates the algorithm's ability to dynamically reposition the base to maintain reachability, while effectively avoiding collisions as environmental complexity increases.

B. Evaluation with different optimization algorithms

To assess the effectiveness of the proposed method, two primary performance metrics are considered: (i) the number of EE poses successfully reached, and (ii) the time required to identify an optimal base position for task execution. The optimization problem designed to reposition the base of an MM can be solved using different algorithms, and this can strongly affect the time required to find an optimal solution. For this reason, a comparative study of their impact on the method's performance has been conducted. Five iterative optimization algorithms were considered: Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Pattern Search (PS), Differential Evolution (DE), and Evolutionary Strategy (ES). To ensure experimental repeatability, the implementations provided by the pymoo library [30] were used, with all algorithms configured using their default parameters. The first objective of the comparison is to analyze the speed of convergence of each algorithm. To this aim, a population size of 15 potential solutions per generation and a termination criterion of 100 generations were adopted to ensure a fair comparison. Figure 6 reports the mean result over 10 trials

for each algorithm, where the same EE target was used but different random initial solutions were provided. PSO, DE, and ES consistently reached the optimal solution within the allocated generations. Conversely, PS converged to a local minimum, while GA failed to reach the optimum within the given iterations. Among the tested methods, ES exhibited the best performance, achieving convergence in fewer iterations.

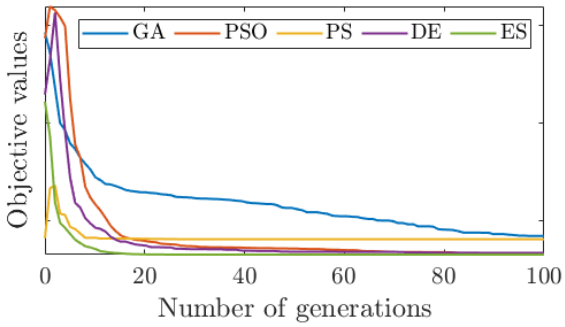


Fig. 6: Visualization of the number of generations needed by each algorithm to converge to an optimal solution.

The effectiveness of the base poses found by the optimization algorithm to reach the desired EE targets has been measured by requesting the MM to retrieve objects from ten locations, each varying in height, environmental clutter, and orientation. This diversity was introduced to replicate realistic operational scenarios to be compliant with both spatial reachability and orientation constraints. The method's efficacy has been analyzed, considering also the different optimization algorithms to further analyze their impact on the performance. Moreover, a direct comparison was made between the performances achieved by two non-expert users who manually positioned the robot to reach the EE targets and the ones obtained using the proposed approach. Figure 7 shows an example with three different poses used to validate the algorithm with different levels of complexity. Table I summarizes the achieved results. The mean time has been computed by terminating the optimization algorithm if the difference between each element of the optimal solutions obtained in two consecutive iterations is less than 10^{-3}.

Approach	Algorithm/ Operator	EE poses reached	Mean time $[s]$
Optimised positioning	GA	10/10	1.84
	PSO	10/10	2.64
	PS	8/10	0.45
	DE	9/10	2.26
	ES	10/10	9.19
Human Intuition	User 1	6/10	3.01
	User 2	7/10	2.59

TABLE I: Performances obtained by the proposed optimized positioning with different algorithms versus human intuition.

The results show the effectiveness of the proposed method in terms of the average number of EE targets successfully reached and the completion time. Moreover, they demonstrate its superiority with respect to the results obtained when the base is placed by human intuition. Indeed, the base poses selected by the non-expert users were too close to the platforms where the objects were located, hindering the possibility of finding a collision-free plan for the manipulator. Different optimization algorithms showed that mobile base poses are found in about 2 seconds using the GA, PSO, and DE algorithms, and they reached almost all the EE targets. The PS algorithm is much faster than the previous ones, but is prone to getting stuck in local minima, so it reaches a lower number of EE targets. Despite the fast convergence, the ES algorithm requires more than 9 seconds on average to find a solution. This is mainly due to its computational complexity, which ensures reaching all the EE targets at the cost of a greater execution time.

VI. Conclusions and Future Works

This paper presented a novel optimization-based method for determining optimal base poses of MMs. A mathematical model was developed using two concentric ellipsoids, whose parameters were obtained through an optimization process to encapsulate regions of the robot RS characterized by high dexterity. This model provides a computationally efficient, bidirectional tool for evaluating EE reachability and guiding base placement, offering a practical alternative to other methods based on RMs and IRMs. These ellipsoid equations were used as constraints of a second optimization problem, which is the core of the novel method proposed here. It has been designed to identify a base pose that maximizes dexterity and dynamically avoids potential collisions when the robot is in motion. Experimental validation across different scenarios (characterized by varying the EE orientation, height, and levels of environmental clutter) demonstrated the flexibility and effectiveness of the proposed method, which significantly outperforms human intuition in achieving optimal placements. Although an optimal solution can be computed within a few seconds, the computation time remains insufficient for integration into real-time closed-loop control systems. Moreover, the current problem formulation considers only single-target EE poses, restricting its applicability in tasks that require simultaneous access to multiple targets. Future work will focus on extending the framework to multi-target optimization, enabling simultaneous reachability of multiple EE poses with minimal base repositioning.

Acknowledgment

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.

(a) (b) (c)

Fig. 7: Base poses obtained by the proposed method in scenarios (with increasing level of challenges): (a) perfect alignment with the EE target; (b) slight misalignment to reduce collision risk; (c) full misalignment to enable collision-free grasping.

REFERENCES

[1] S. Thakar, S. Srinivasan, S. Al-Hussaini, P. M. Bhatt, P. Rajendran, Y. Jung Yoon, N. Dhanaraj, R. K. Malhan, M. Schmid, V. N. Krovi *et al.*, "A survey of wheeled mobile manipulation: A decision-making perspective," *Journal of Mechanisms and Robotics*, vol. 15, no. 2, p. 020801, 2023.

[2] B. Shao, N. Cao, Y. Ding, X. Wang, F. Gu, and C. Chen, "Momapos: An efficient object-kinematic-aware base placement optimization framework for mobile manipulation," *arXiv preprint arXiv:2403.19940*, 2024.

[3] T. Bachmann, K. Nottensteiner, and M. A. Roa, "Automated planning of workcell layouts considering task sequences," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 12 662–12 668.

[4] L. Naik, S. Kalkan, S. L. Sørensen, M. B. Kjærgaard, and N. Krüger, "Basenet: a learning-based mobile manipulator base pose sequence planning for pickup tasks," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2024, pp. 3666–3673.

[5] Q. Sun, W. Zhi, T. Zhang, and M. Johnson-Roberson, "Teaching robots where to go and how to act with human sketches via spatial diagrammatic instructions," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2024, pp. 1094–1100.

[6] Q. Yu, G. Wang, X. Hua, S. Zhang, L. Song, J. Zhang, and K. Chen, "Base position optimization for mobile painting robot manipulators with multiple constraints," *Robotics and Computer-Integrated Manufacturing*, vol. 54, pp. 56–64, 2018.

[7] J. Müller, U. Frese, and T. Röfer, "Grab a mug - object detection and grasp motion planning with the nao robot," in *2012 12th IEEE-RAS International Conference on Humanoid Robots (Humanoids 2012)*, 2012, pp. 349–356.

[8] F. Burget and M. Bennewitz, "Stance selection for humanoid grasping tasks by inverse reachability maps," in *2015 IEEE International conference on robotics and automation (ICRA)*, 2015, pp. 5669–5674.

[9] A. Makhal and A. K. Goins, "Reuleaux: robot base placement by reachability analysis," in *2018 2nd IEEE international conference on robotic computing (IRC)*, 2018, pp. 137–142.

[10] Y. Yang, V. Ivan, Z. Li, M. Fallon, and S. Vijayakumar, "idrm: Humanoid motion planning with realtime end-pose selection in complex environments," in *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, 2016, pp. 271–278.

[11] T. Sandakalum, N. X. Yao, and M. H. Ang, "Inv-reach net: Deciding mobile platform placement for a given task," in *2022 IEEE-RAS 21st International Conference on Humanoid Robots (Humanoids)*, 2022, pp. 127–133.

[12] A. Trabelsi, J. Sandoval, A. Mlika, S. Lahouar, S. Zeghloul, and M. A. Laribi, "Robot base placement and tool mounting optimization based on capability map for robot-assistant camera holder," *Robotica*, vol. 42, no. 8, pp. 2489–2510, 2024.

[13] F. Reister, M. Grotz, and T. Asfour, "Combining navigation and manipulation costs for time-efficient robot placement in mobile manipulation tasks," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 9913–9920, 2022.

[14] F. Wang, J. R. G. Olvera, and G. Cheng, "Optimal order pick-and-place of objects in cluttered scene by a mobile manipulator," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 6402–6409, 2021.

[15] Q. Fan, Z. Gong, B. Tao, Y. Gao, Z. Yin, and H. Ding, "Base position optimization of mobile manipulators for machining large complex components," *Robotics and Computer-Integrated Manufacturing*, vol. 70, p. 102138, 2021.

[16] H. Liu, Y. Ma, X. Chen, and Z. Yu, "Placement optimization for humanoid robot manipulation using a task-driven planning method," in *2024 IEEE International Conference on Cyborg and Bionic Systems (CBS)*, 2024, pp. 46–51.

[17] M. Forstenhäusler, T. Wetner, and K. Dietmayer, "Optimized mobile robot positioning for better utilization of the workspace of an attached manipulator," in *2020 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, 2020, pp. 2074–2079.

[18] K. Zbiss, A. Kacem, M. Santillo, and A. Mohammadi, "Automatic optimal robotic base placement for collaborative industrial robotic car painting," *Applied Sciences*, vol. 14, no. 19, p. 8614, 2024.

[19] L. Naik, S. Kalkan, and N. Krüger, "Pre-grasp approaching on mobile robots: A pre-active layered approach," *IEEE Robotics and Automation Letters*, vol. 9, no. 3, pp. 2606–2613, 2024.

[20] S. Jauhri, J. Peters, and G. Chalvatzaki, "Robot learning of mobile manipulation with reachability behavior priors," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 8399–8406, 2022.

[21] P. Nayyeri, K. Zareinia, and H. Bougherara, "Matlab toolbox for reachability analysis and visualization of robotic manipulators," in *2024 8th International Conference on Robotics and Automation Sciences (ICRAS)*, 2024, pp. 49–54.

[22] R. F. Cavelli, P. D. Cen Cheng, and M. Indri, "Modeling the reachability space of robotic manipulators through ellipsoid equations," *Submitted to Journal of Intelligent & Robotic Systems*, 2025.

[23] ROS, *URDF- ROS*, ROS.org, 2023, accessed: April 2025. [Online]. Available: https://wiki.ros.org/urdf

[24] Á. González, "Measurement of areas on a sphere using fibonacci and latitude–longitude lattices," *Mathematical geosciences*, vol. 42, pp. 49–64, 2010.

[25] O. Project, "PyKDL: Python bindings for the Kinematics and Dynamics Library (KDL)," 2021, Accessed: April 2025. [Online]. Available: https://github.com/orocos/orocos_kinematics_dynamics

[26] E. Coumans and Y. Bai, "Pybullet, a python module for physics simulation for games, robotics and machine learning," http://pybullet.org, 2016–2021.

[27] Trossen Robotics, *LoCoBot WX250*, Trossen Robotics, 2020, accessed: April 2025. [Online]. Available: https://www.trossenrobotics.com/locobot-wx250s

[28] "Github repository with the developed code," [Accessed: April 2025]. [Online]. Available: https://github.com/Saro0800/base_pose_opt_ws.git

[29] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "OctoMap: An efficient probabilistic 3D mapping framework based on octrees," *Autonomous Robots*, 2013. [Online]. Available: https://octomap.github.io

[30] J. Blank and K. Deb, "pymoo: Multi-objective optimization in python," *IEEE Access*, vol. 8, pp. 89 497–89 509, 2020.