2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

Real-time Whole-body Motion Planning for Mobile Manipulators Using Environment-adaptive Search and Spatial-temporal Optimization

Chengkai Wu[1,2,*], Ruilin Wang[1,*], Mianzhi Song[1], Fei Gao[3], Jie Mei[2], Boyu Zhou[1,†]

***Abstract*— Mobile manipulators have recently gained significant attention in the robotics community due to their superior potential in industrial and service applications. However, the high degree of freedom associated with mobile manipulators poses challenges in achieving real-time whole-body motion planning. To bridge the gap, this paper presents a motion planning method capable of generating high-quality, safe, agile and feasible trajectories for mobile manipulators in real time. First, we present a novel environment-adaptive path searching method, which can generate paths in real-time in various environments by adaptively adjusting searching dimension based on environment complexity. Additionally, we propose a real-time spatial-temporal trajectory optimization method that takes into account the whole-body safety, agility and dynamic feasibility of mobile manipulators. Moreover, task constraints are applied to ensure that the trajectory can fulfill specific task requirements. Simulation and real-world experiments demonstrate that our method is capable of generating whole-body trajectories in real-time in challenging environments. We will release our code to benefit the community.**

I. Introduction

Recently, mobile manipulators have garnered widespread attention from researchers owing to their potential in industrial and service applications. Compared to traditional fixed-base manipulators, mobile manipulators possess the ability to manipulate objects and move, allowing them to handle a broader range of tasks [1], including inspection, cargo transportation, and housekeeping. This study focuses on the challenge of motion planning for wheeled mobile manipulators in complex environments.

Finding a real-time, feasible, and high-quality path for a mobile manipulator remains a significant challenge. Existing methods [2] divide the mobile manipulator into multiple subsystems and plan the path of each subsystem sequentially. However, the simple hierarchical strategy lacks environmental adaptablility, which causes massive sampling and unpredictable computation time. On the other hand, other methods [3]–[5] consider all the degrees of freedom of mobile manipulators simultaneously. However, they require substantial computational time to identify a satisfactory path due to the high dimensionality of the searching problem.

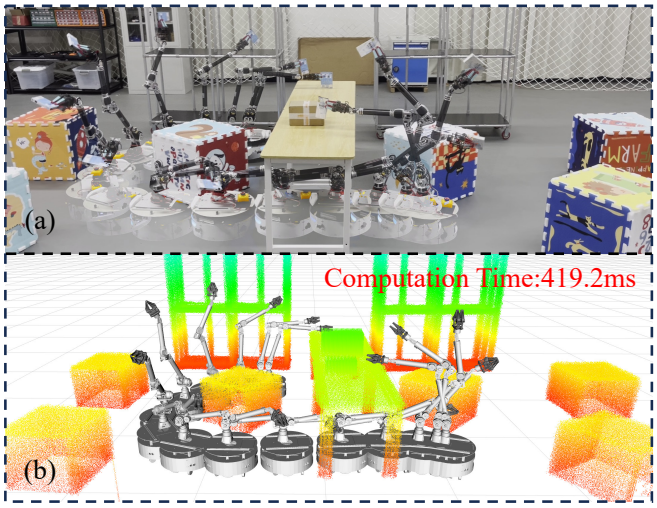

Fig. 1: The agile and safe motion for the mobile manipulator, which involves grasping objects, navigating beneath tables, traversing narrow passages, and ultimately reaching the goal state. The movement diagram of the real-world view (a) and point cloud view (b). The planning time on the onboard computer is 419.2ms.

An additional challenge stems from the absence of a real-time trajectory generation method that can achieve agile, non-conservative, and safe motion in complex environments while ensuring dynamic feasibility and accomplishing specific tasks. To address this challenge, optimization-based approaches [6, 7] are employed. However, achieving agile motion requires simultaneous optimization of all joint configurations and execution time, resulting in increased optimization variables. Ensuring non-conservative and safe motion requires a precise collision model, which in turn extends the time required for collision checking. To generate a dynamically feasible and safe trajectory for task completion, it becomes imperative to impose nonlinear constraints on the optimization problem. These factors together result in a challenging trajectory optimization problem.

To tackle these challenges, we introduce a real-time whole-body motion planning method for mobile manipulators, which comprises an environment-adaptive path searching and a spatial-temporal trajectory optimization. The environment-adaptive search primarily explores paths for the mobile base and manipulator in a decoupled manner, reducing problem dimensionality and efficiently finding high-quality paths in environments with moderate obstacle density. When confronted with highly narrow regions, it dynamically identifies the corresponding sections and performs dedicated local

* **Equal contribution**. † **Corresponding Author**.
[1] School of Artificial Intelligence, Sun Yat-sen University, Zhuhai, China.
[2] School of Mechanical Engineering and Automation, Harbin Institute of Technology Shenzhen, Shenzhen, China.
[3] State Key Laboratory of Industrail Control Technology, Institute of Cyber-Systems and Control, Zhejiang University, Hangzhou, China.
{22s153131@stu.hit,jmei@hit}.edu.cn, hustsmz@gmail.com, fgaoaa@zju.edu.cn
{wangrlin9@mail2,zhouby23@mail}.sysu.edu.cn

whole-body path searching, ensuring completeness in challenging scenarios. In the trajectory optimization, we leverage the MINCO trajectory class [8] as our representation, decoupling the number of optimization variables from the number of constraint points to simplify the problem. To maximize the utilization of safe space, we approximate the mobile manipulator as a collection of collision spheres. To address nonlinear constraints, we employ a constraint transcription method to eliminate them, transforming the constrained optimization problem into an unconstrained one. Simulations and real-world experiments validate our method's capability to generate real-time flexible motion with a high success rate in typical complex scenarios. Our contributions can be summarized as:

1) An environment-adaptive path searching method that finds a high-quality path for the mobile manipulator in real time while ensuring completeness.
2) A spatial-temporal optimization method capable of generating a smooth, agile, safe and dynamically feasible trajectory that can meet specific task requirements in real time.
3) We demonstrate the effectiveness of the proposed method in simulation and real-world experiments. The code will be released at `https://github.com/SYSU-STAR/REMANI-Planner`.

II. Related Works

Path planning in complex environments with high-dimensional configuration spaces often leads to time-consuming path finding processes. To address this issue, some methods [9]–[11] fold the manipulator and plan a 2D path for the mobile manipulator. However, these methods significantly waste the degree of freedom of the mobile manipulator, thus limiting its applications. Some researchers proposed dimensionality-reduction-based methods [2, 12]. These methods divides the mobile manipulator into two subsystems: the mobile base and the manipulator. It plans a path for one subsystem while keeping the other fixed. However, these approaches often generate conservative trajectories and result in inconsistent movements. Sampling-based methods [5, 13]–[16], the mainstream planning method used in high-dimensional configuration spaces, can find collision-free paths for high-dimensional systems. However, these methods bring unpredictable computation time to the mobile manipulator. The optimality of these methods comes at the expense of excessive time consumption, making it impractical for real-time applications.

Numerous methods have been proposed to address the problem of trajectory generation for wheeled mobile manipulators. Tang *et al.* [17] proved that wheeled mobile manipulator model is differentially flat and plan the trajectory in the flat output space. However, it restricts the movement of the mobile base to forward only and does not consider obstacle avoidance or dynamic feasibility. Several optimization-based methods [6, 7, 18] have been proposed to generate collision-free trajectories for high-dimensional manipulators. However, these methods cannot be executed in real time because of large number of optimization variables resulting from their trajectory representation. Recently, several Model Predictive Control (MPC)-based methods [19, 20] have been proposed, enabling coupled, dynamically feasible, and safe movements for mobile manipulators. However, these methods heavily rely on a reference path, which may lead to failure in reaching the goal in complex environment or unnecessary movements if a low-quality reference is used.

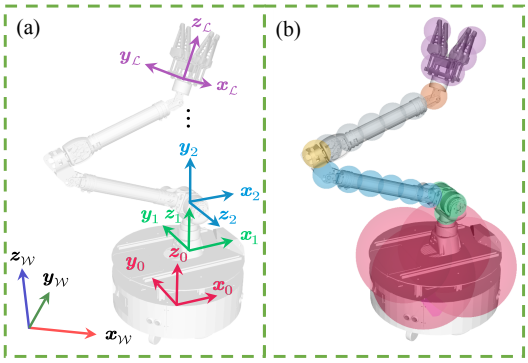

Fig. 2: The definition of coordinates (a) and the collision model (b). Each color represents a specific link to which the frame of the mobile manipulator is attached, as well as the link that the collision spheres cover.

III. Preliminaries

A. Coordinates Definition and Collision Model

In this paper, the left superscript of vectors and matrices indicates the coordinate frame in which they are expressed. We employ a total of $2+\mathcal{L}$ right-hand coordinate frames, as illustrated in Fig. 2(a). To avoid conservative motion, we approximate the mobile manipulator as a set of collision spheres, as illustrated in Fig. 2(b). ${}^{l}\boldsymbol{p}_{l,m}$ denotes the position of the center of the m-th sphere covering link l, where $l \in \{0, 1, \cdots, \mathcal{L}\}$. The centers of these spheres in the world frame $\mathcal{W}$ can be obtained through the forward kinematics of the mobile manipulator:

$$\begin{aligned} {}^{\mathcal{W}}\boldsymbol{p}_{l,m} &= {}^{\mathcal{W}}\boldsymbol{T}_{l}\,{}^{l}\boldsymbol{p}_{l,m}, \\ &\forall l \in \{0, 1, \cdots, \mathcal{L}\}, \forall m \in \{1, \cdots, m_l\}, \end{aligned} \tag{1}$$

where m_l denotes the number of spheres covering link l, ${}^{\mathcal{W}}\boldsymbol{T}_{l}$ is the homogeneous transformation from l frame to $\mathcal{W}$ frame. We use r_l to represent the radius of the collision spheres covering link l.

B. Dynamic Model

In this paper, we study on a mobile manipulator controlled by velocity. The mobile manipulator comprises a differential drive base and a $\mathcal{L}$-DOF manipulator. The states of the mobile manipulator can be represented as $\boldsymbol{x} = \left[\boldsymbol{x}_{\mathfrak{b}}^{\top}, \boldsymbol{x}_{\mathfrak{m}}^{\top}\right]^{\top}$, where $\boldsymbol{x}_{\mathfrak{b}} = [\boldsymbol{q}_{\mathfrak{b}}^{\top}, \psi]^{\top} = [q_x, q_y, \psi]^{\top}$ indicates the position and orientation of the mobile base, and $\boldsymbol{q}_{\mathfrak{m}} = [q_1, \cdots, q_{\mathcal{L}}]^{\top}$ corresponds to the joint angles of the manipulator. The control inputs for the system consist of the wheel and joint angular velocities, represented as $\boldsymbol{u} = [\omega_l, \omega_r, \dot{\boldsymbol{q}}_{\mathfrak{m}}^{\top}]^{\top}$. Additionally, $\boldsymbol{x}$, $\boldsymbol{u}$ and $\dot{\boldsymbol{u}}$ can be represented in terms of the

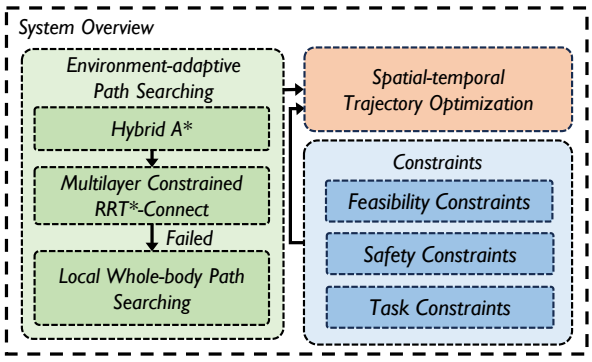

Fig. 3: The overview of the proposed real-time whole-body motion planning framework for mobile manipulators.

generalized joint configuration $\boldsymbol{q} = \left[\boldsymbol{q}_{\mathfrak{b}}^\top, \boldsymbol{q}_{\mathfrak{m}}^\top\right]^\top \in \mathbb{R}^{2+\mathcal{L}}$ and their derivatives: $\psi = \operatorname{atan2}\left(\eta\dot{q}_y, \eta\dot{q}_x\right)$,

$$\omega_{l(r)} = \frac{1}{2r_w}\left(2\eta\left\|\dot{\boldsymbol{q}}_{\mathfrak{b}}\right\|_2 \underset{(+)}{-} d_w\omega_{\mathfrak{b}}\right), \tag{2a}$$

$$\dot{\omega}_{l(r)} = \alpha_{l(r)} = \frac{1}{2r_w}\left(2\eta\frac{\dot{\boldsymbol{q}}_{\mathfrak{b}}^T\ddot{\boldsymbol{q}}_{\mathfrak{b}}}{\left\|\dot{\boldsymbol{q}}_{\mathfrak{b}}\right\|_2} \underset{(+)}{-} d_w\alpha_{\mathfrak{b}}\right), \tag{2b}$$

where r_w denotes the radius of the wheels, d_w represents the distance between two wheels. $\eta = -1$ and $\eta = 1$ indicate the backward and forward movements of the mobile base, respectively [21]. $\omega_{\mathfrak{b}}$ and $\alpha_{\mathfrak{b}}$ are angular velocity and acceleration of the mobile base, respectively [22].

C. System Overview

In this paper, we present a real-time whole-body motion planning method for mobile manipulators, illustrated in Fig.3. Firstly, we determine a path for the mobile manipulator by utilizing environment-adaptive path searching, detailed in Sec. IV. By using this path as an initial guess, we construct a spatial-temporal trajectory optimization problem considering certain constraints to obtain a safe and feasible trajectory, detailed in Sec. V, which enables the mobile manipulator to complete specific tasks.

IV. Environment-adaptive Path Searching

Our environment-adaptive path searching is illustrated in Fig. 4. Firstly, we utilize Hybrid A* [23] to find a smooth, safe, feasible and minimum-cost path for the mobile base, based on which a Multilayer Constrained RRT*-Connect (Sec. IV-C) is utilized to find a path of the manipulator. If no solution is found, we identify the interrupted part and conduct dedicated whole-body path planning locally (Sec. IV-D). This strategy ensures finding high-quality path efficiently in areas with moderate obstacle density, while also guaranteeing completeness when narrow passages are encountered.

A. Mobile Base Path Generation

At the beginning of our algorithm, we utilize Hybrid A* to determine a path for the mobile base. The path of the mobile base, denoted as $\mathcal{P}_b = \{\mathbf{P}_{b,1}, \cdots, \mathbf{P}_{b,N}\}$, consists of N sub-paths. $\mathbf{P}_{b,i} = \{\mathbf{Q}_{b,i,1}, \cdots, \mathbf{Q}_{b,i,M_i+1}\}$ represents the i-th sub-path, where $\mathbf{Q}_{b,i,j} = [p_x, p_y, \psi]^\top \in \mathbb{R}^3$ is the j-th waypoint in $\mathbf{P}_{b,i}$. Each sub-path corresponds to a forward or reverse direction, indicated by $\eta_i \in \{-1, 1\}$. Additionally, we also get the time allocation of i-th sub-path $\mathbf{T}_i \in \mathbb{R}^{M_i}$. Eventually, we get mobile base path $\mathcal{P}_b$, $\boldsymbol{\eta} = \{\eta_1, \cdots, \eta_N\}$ and time allocation $\mathbf{T} = \left[\mathbf{T}_1^\top, \cdots, \mathbf{T}_N^\top\right]^\top$.

B. Multilayer Manipulator Searching Space

Given the above path of the mobile base, we further search a feasible path of the manipulator based on it. Specifically, for each waypoint $\mathbf{Q}_{b,i,j} \in \mathbf{P}_{b,i}$, we would like to determine a manipulator state waypoint $\mathbf{Q}_{m,i,j} \in \mathbb{R}^{\mathcal{L}}$, which in combination with $\mathbf{Q}_{b,i,j}$ forms the complete state of the mobile manipulator $\mathbf{Q}_{i,j} = \{\mathbf{Q}_{b,i,j}, \mathbf{Q}_{m,i,j}\}$. In this way, we search for a complete path $\mathcal{P} = \{\mathbf{P}_1, \cdots, \mathbf{P}_N\}$ for mobile manipulator, where $\mathbf{P}_i = \{\mathbf{Q}_{i,1}, \cdots, \mathbf{Q}_{i,M_i+1}\}$ is the mobile manipulator path consisting of multiple sub-paths corresponding to those of the mobile base.

To determine the manipulator state path, we construct a searching space for the manipulator and utilize the Multilayer Constrained RRT*-Connect method (detailed in Sec. IV-C) to search for a feasible path. For each mobile base waypoint, manipulator states can be combined with it as long as certain constraints are met, these states constitute a searching space (see light green ellipses in Fig. 4(a)). Furthermore, there is a sequential relationship among the mobile base waypoints. Thus the association between manipulator states and mobile base waypoints making the manipulator searching space a multilayer structure.

C. Multilayer Constrained RRT*-Connect

In Sec. IV-B, we introduce the multilayer structure of the searching space. Considering the high dimension of the manipulator and the multilayer nature of the searching space, we propose a Multilayer Constrained RRT*-Connect method to find the appropriate manipulator state for each waypoint.

The Multilayer Constrained RRT*-Connect has a similar structure to RRT*-Connect [4]. We will explain how to expand the searching tree under the multilayer constraint.

EXTEND* process is the most important process in both RRT*-Connect and our method, which contains 3 steps: SAMPLE, NEAREST and STEER. In SAMPLE step, we random sample a state $x_{rand} = (q_{m,rand}, h_{rand})$ (see the purple dotted node in Fig. 4(a)) which consists of a random manipulator state $q_{m,rand}$ and a random layer h_{rand}. Then, in NEAREST step, we attempt to find the nearest node $x_{nearest} = (q_{m,nearest}, h_{nearest})$ on one of the two searching tree (see Tree 1 and purple node in Fig. 4(a)). These trees are alternately selected during different sampling steps. STEER step will create a new node x_{new} (see red dot and edge in Fig. 4(a)) at layer $h_{nearest} + 1$ due to the multilayer constraint. Finally, a process similar to rewire in RRT* is performed, which only need to consider nodes in adjacent layers.

After extending a new node, we try to find a connection from the nearest node in another searching tree (see purple node in Fig. 4(b)) to x_{new} using the CONNECT process layer by layer until a collision occurs (see red dotted line

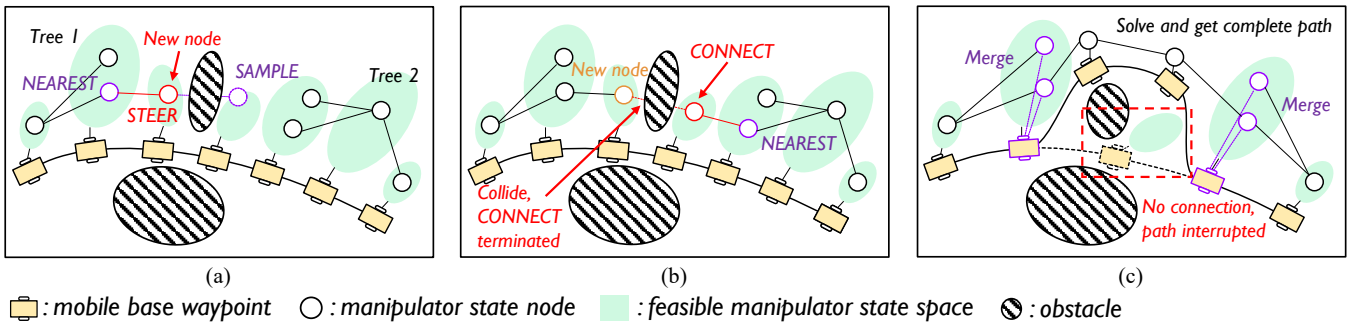

Fig. 4: The overview of the environment-adaptive path searching algorithm proposed in Sec.IV. We construct a multilayer manipulator searching space based on the mobile base path and conduct Multilayer Constrained RRT*-Connect proposed in Sec.IV-C to find a feasible path for manipulator. The EXTEND* process and CONNECT process are respectively illustrated in (a),(b). If no solution found, the local high dimension path searching is employed to complete the path (c).

in Fig. 4(b)), which is similar to Alg. 7 in [4]. This process will generate nodes within the layers where the connection passes through (see red node in Fig. 4(b)). If the predefined termination condition, such as a time limit, is not met, we repeat the above process, in which the role of two searching trees are exchanged.

D. Local Whole-body Path Searching

Our Multilayer Constrained RRT*-Connect efficiently generates feasible and high-quality paths in areas with moderate obstacles. However, due to the separate search processes for the mobile base path and manipulator path, our method is susceptible to getting stuck in highly narrow environments. In such cases, an interrupted part emerges between the two searching trees where no feasible connection can be found (as shown in Fig. 4(c)). Fortunately, the bidirectional search offers an effective means to leverage previous search results and bridge the interrupted part, eliminating the need for conducting whole-body path planning from scratch.

In this settings, we locally perform whole-body path searching, where the searching space corresponds to the whole configuration space of mobile manipulator. The initial state and goal state are set corresponding to the interrupted part of the two searching trees. We collect the manipulator states in both ends of the interrupted tree and merge them with the associated mobile base states into the whole-body states (see purple doted lines in Fig. 4(c)). Then, we formulate it as a multi-start, multi-goal path planning problem, which is solved using RRT*-Connect [4] algorithm. Our local whole-body path searching guarantees the discovery of a complete path connecting the previously interrupted part, as illustrated in Fig. 4(c). This strategy empowers our method to dynamically adapt the search space and maintain completeness even in challenging environments. Typically, it entails resolving just two simpler search problems in $\mathbb{R}^3$ and $\mathbb{R}^{\mathcal{L}}$, resulting in high-quality paths in shorter time. In the worst-case scenario, it may need to search the entire path in $\mathbb{R}^{3+\mathcal{L}}$.

V. Spatial-temporal Trajectory Optimization

Given the initial path obtained in Sec. IV, we formulate a non-linear optimization problem for whole-body trajectory planning of mobile manipulators, which generates smooth, safe and feasible trajectories that satisfies specific task constraints in real time.

A. Optimization Formulation

As we discussed in Sec. III-B, the state, input, and input derivative can be expressed in terms of the generalized joint configuration. Therefore, we choose the generalized joint configuration $\boldsymbol{q}$ as our planning space.

After obtaining a collision-free path $\mathcal{P}$ along with their corresponding forward and backward variables $\boldsymbol{\eta}$, we partition the trajectory into N segments. The i-th segment is represented as a M_i-piece $(2+\mathcal{L})$-dimensional piece-wise polynomial with a degree of $2s-1$, which is denoted as $\boldsymbol{q}_i(t)$ for $t \in [0, T_i]$:

$$\begin{cases} \boldsymbol{q}_{i,1}(t) = \boldsymbol{c}_{i,1}^T \boldsymbol{\beta}(t - \bar{T}_{i,0}) & , t \in [\bar{T}_{i,0}, \bar{T}_{i,1}] \\ \quad\vdots & \quad\vdots \\ \boldsymbol{q}_{i,M_i}(t) = \boldsymbol{c}_{i,M_i}^T \boldsymbol{\beta}(t - \bar{T}_{i,M_i-1}) & , t \in [\bar{T}_{i,M_i-1}, \bar{T}_{i,M_i}] \end{cases}$$

where $T_i = \|\mathbf{T}_i\|_1$ and $\mathbf{T}_i = [T_{i,1}, \cdots, T_{i,M_i}]^\top$ represents the time allocation. $\bar{T}_{i,j} = \sum_{k=1}^{j} T_{i,k}, j \in \{0, \cdots, M_i\}$ denotes the timestamp of the start of $\boldsymbol{q}_{i,j+1}(t)$ or the end of $\boldsymbol{q}_{i,j}(t)$. $\boldsymbol{c}_{i,j} \in \mathbb{R}^{2s\times(2+\mathcal{L})}$ is the coefficient matrix and $\boldsymbol{\beta}(t) = [1, t, \cdots, t^{2s-1}]^\top$ represents the time basis.

To generate a smooth, safe and feasible trajectory for the mobile manipulator to complete a specific task, we formulate the objective function of the trajectory generation problem as followed:

$$J = \sum_{i=1}^{N} J_{s,i} + J_{T,i} + J_{p,i}, \tag{3}$$

where $J_{s,i} = \int_0^{T_i} \boldsymbol{q}_i^{(s)}(t)^\top \boldsymbol{q}_i^{(s)}(t) dt$ is the control efforts of the trajectory. $J_{T,i} = \omega_T T_i$ is the time regulation term, which makes the execution time of the trajectory to be as small as possible.

$$J_{p,i} = \sum_{j=1}^{M_i} \sum_{d\in\mathcal{D}} \omega_d I_\Sigma(\mathcal{G}_d(\boldsymbol{q}_{i,j})), \tag{4}$$

is the penalty of the inequality constraints $\mathcal{G}_d(\boldsymbol{q}_{i,j}) \le 0$, which will be further discussed in Sec.V-B, Sec.V-C and Sec.V-D. $I_\Sigma(\cdot)$ represents the sparse and dense sampling

strategy utilized to sample the trajectory more densely around points with highly nonlinear constraints, ultimately improving the convergence rate of the optimization. In order to maintain brevity, we direct interested readers to [22] for more details about the sampling strategy. ω_T and ω_d are weights of time and penalty, respectively. And then, we can get the trajectory by solving Eq. (3) using MINCO [8] trajectory optimization.

B. Safety Constraints

This subsection introduces safety constraints for both obstacle avoidance and self-collision avoidance.

1) Obstacle Avoidance: To guarantee no collisions between the obstacle and the robot, we build the ESDF map [24] and ensure that the ESDF values $D_{ESDF}(\cdot)$ at the centers of the collision spheres, which is discussed in Sec.III-A, are greater than their respective radius r_l:

$$\begin{aligned} &\mathcal{G}_{s_o}(\boldsymbol{q}, \dot{\boldsymbol{q}}_{\mathfrak{b}}) = r_l - D_{ESDF}\left({}^{\mathcal{W}}\boldsymbol{p}_{l,m}\left(\boldsymbol{q}, \dot{\boldsymbol{q}}_{\mathfrak{b}}\right)\right), \\ &\forall l \in \{\mathcal{B}, 1, 2, \cdots, \mathcal{L}\}, \forall m \in \{1, \cdots, m_l\}, \end{aligned} \tag{5}$$

2) Self-Collision Avoidance: We check for collisions between every pair of links:

$$\begin{aligned} \mathcal{G}_{s_s}(\boldsymbol{q}_{\mathfrak{m}}) = r_{l'} + r_l - \left\| {}^{l'}\boldsymbol{p}_{l',i} - {}^{l'}\boldsymbol{T}_l\, {}^{l}\boldsymbol{p}_{l,j} \right\|_2, \\ \forall l' \in \{0, \cdots, l-1\}, \forall l \in \{1, \cdots, \mathcal{L}\}, \\ \forall i \in \{1, \cdots, m_{l'}\}, \forall j \in \{1, \cdots, m_l\}. \end{aligned} \tag{6}$$

C. Feasibility Constraints

To ensure the trajectory not to surpasses the tracking capabilities of a mobile manipulator, it is imperative to incorporate feasibility constraints within the robot's motion planning framework.

1) Wheel Constraints: For the mobile base, we impose restrictions on the angular velocity and acceleration of the left and right wheels, limiting them to the maximum values $\omega_{w,\max}$ and $\alpha_{w,\max}$ respectively:

$$\mathcal{G}_{\omega_l(r)}\left(\dot{\boldsymbol{q}}_{\mathfrak{b}}, \ddot{\boldsymbol{q}}_{\mathfrak{b}}\right) = \omega_{l(r)}^2\left(\dot{\boldsymbol{q}}_{\mathfrak{b}}, \ddot{\boldsymbol{q}}_{\mathfrak{b}}\right) - \omega_{w,\max}^2, \tag{7a}$$

$$\mathcal{G}_{\alpha_l(r)}\left(\dot{\boldsymbol{q}}_{\mathfrak{b}}, \ddot{\boldsymbol{q}}_{\mathfrak{b}}, \dddot{\boldsymbol{q}}_{\mathfrak{b}}\right) = \alpha_{l(r)}^2\left(\dot{\boldsymbol{q}}_{\mathfrak{b}}, \ddot{\boldsymbol{q}}_{\mathfrak{b}}, \dddot{\boldsymbol{q}}_{\mathfrak{b}}\right) - \alpha_{w,\max}^2. \tag{7b}$$

The values of $\omega_l(r)$, $\alpha_l(r)$ can be computed using Eq. (2). Moreover, if $\|\dot{\boldsymbol{q}}_{\mathfrak{b}}\|_2 = 0$, the angular velocity and acceleration of the wheel become infinite. Therefore, a constraint is imposed on the minimum velocity as follows:

$$\mathcal{G}_{\bar{v}}(\dot{\boldsymbol{q}}_{\mathfrak{b}}) = v_{\min}^2 - \dot{\boldsymbol{q}}_{\mathfrak{b}}^T \dot{\boldsymbol{q}}_{\mathfrak{b}}, \tag{8}$$

where $v_{\min} \in \mathbb{R}_{>0}$ is a extremely small value. In this paper, $v_{\min}$ is defined as 0.05.

2) Joint Constraints: For each joint l of the manipulator, where $l \in \{1, \cdots, \mathcal{L}\}$, we consider the limits on joint angle, velocity, and acceleration:

$$\mathcal{G}_{\mathfrak{q}_\theta}(q_l) = q_l - q_{l,\max}, \tag{9a}$$

$$\mathcal{G}_{\mathfrak{q}_{\bar{\theta}}}(q_l) = q_{l,\min} - q_l, \tag{9b}$$

$$\mathcal{G}_{\mathfrak{q}_\omega}(\dot{q}_l) = \dot{q}_l^2 - \omega_{l,\max}^2, \tag{9c}$$

$$\mathcal{G}_{\mathfrak{q}_\alpha}(\ddot{q}_l) = \ddot{q}_l^2 - \alpha_{l,\max}^2, \tag{9d}$$

where $q_{l,\min}$ and $q_{l,\max}$ represent the minimum and maximum joint angles of joint l, and $\omega_{l,\max}$ and $\alpha_{l,\max}$ represent the maximum joint angular velocity and acceleration of the l-th joint.

TABLE I: ABLATION STUDY RESULTS

h_b(m)	0.7	0.6	0.5	0.4
Baseline	98.16%	53.75%	2.37%	0.00%
Ours	**100.00%**	**100.00%**	**100.00%**	**100.00%**

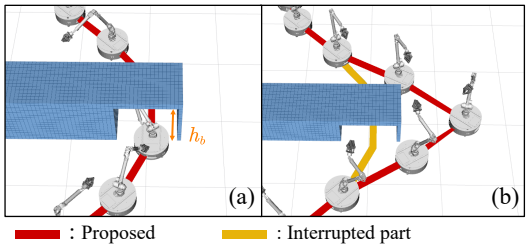

Fig. 5: The generated path in ablation study. (a): $h_b = 0.7m$. (b): $h_b = 0.4m$.

D. Task Constraints

In numerous instances [25], the end-effector of the mobile manipulator is required to follow a designated trajectory or maintain a specific orientation. To achieve these tasks, task constraints are introduced. We restrict Euclidean distance between the position of the end-effector ${}^{\mathcal{W}}\boldsymbol{p}_e$ and the target position ${}^{\mathcal{W}}\boldsymbol{p}_{\mathfrak{t}}$ within the tolerance distance d_p:

$$\mathcal{G}_{t_p}(\boldsymbol{q}) = \left\| {}^{\mathcal{W}}\boldsymbol{p}_e - {}^{\mathcal{W}}\boldsymbol{p}_{\mathfrak{t}} \right\|_2 - d_p. \tag{10}$$

And the minimum rotation angle between the orientation of the end-effector ${}^{\mathcal{W}}\boldsymbol{R}_e$ and target orientation ${}^{\mathcal{W}}\boldsymbol{R}_{\mathfrak{t}}$ is constrained within the tolerance angle d_o:

$$\mathcal{G}_{t_o}(\boldsymbol{q}) = \arccos\left(\frac{1}{2}\left(\mathrm{tr}\left({}^{\mathcal{W}}\boldsymbol{R}_{\mathfrak{t}}{}^{\top}\, {}^{\mathcal{W}}\boldsymbol{R}_e\right) - 1\right)\right) - d_o. \tag{11}$$

VI. Experiment

In this section, we conduct simulations and real-world experiments to demonstrate proposed method's efficiency.

A. Benchmark Comparison

Simulation benchmarks are conducted to demonstrate the effectiveness of our method. All simulations are performed in the Robot Operating System (ROS) on a computer equipped with Intel i7-13700KF CPU.

1) Ablation Study: We conduct an ablation study to validate the effectiveness of our local whole-body path searching method proposed in Sec.IV-D. The baseline is chosen as our method proposed in Sec. IV without local whole-body path searching. The study is conducted in scenarios involving a bridge hole and a passage (see Fig. 5). We adjust the height of the bridge (h_b) from $0.7m$ to $0.4m$ with a resolution of $0.1m$ and limit computation time to $600ms$. The success rate for both methods are shown in Table I.

As the height of the bridge decreases, the success rate of baseline quickly falls to 0. On the contrary, our method can automatically identify and complete the interrupted part (see Fig. 5(b)), resulting in a high level of completeness.

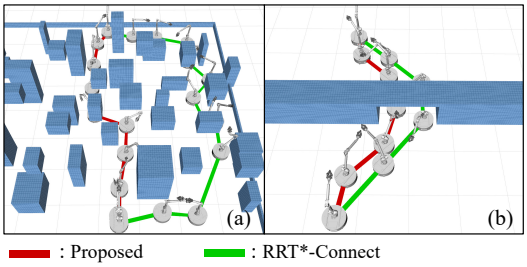

Fig. 6: The generated path in Cuboids (a) and Bridge (b).

TABLE II: PATH PLANNING RESULTS

	method	$t_p(ms)$	$l_b(m)$	$l_j(rad)$	$succ$
Cuboids	Ours	**57.22**	**7.75**	**19.82**	**100.00%**
	RRT*-Connect	119.67	10.52	36.19	43.50%
Bridge	Ours	**12.96**	**5.70**	**19.56**	**100.00%**
	RRT*-Connect	107.37	6.04	22.99	12.62%

2) Path Planning: We compare our environment-adaptive path searching method with RRT*-Connect [4]. We perform the comparison in two different scenarios, as illustrated in Fig. 6. The Cuboids scenario is an $8m \times 8m$ room with cuboids positioned both on the ground and in the sky. In the Bridge scenario, the height of the bridge is set to 0.7. To achieve real-time planning, a planning time limit of 200ms is enforced. The RRT*-Connect algorithm terminates either upon finding a path or upon reaching the time limit. We measure and record the average planning time (t_p), path length of the mobile base (l_b), path length of the manipulator joints (l_j), and success rate (succ). These results are presented in detail in Table II.

Our method achieves perfect success rate while offering an improvement in computation time and trajectory quality compared to RRT*-connect.

3) Trajectory Generation: To validate the real-time capability of our trajectory optimization method in generating feasible trajectories, we conduct a benchmark comparison with CHOMP [6] in two distinct scenarios the same as Fig. 6. The initial guess for both methods is determined using the approach proposed in Sec.IV, and we ensure that both methods have the same number of constraint points. We record the average optimization time (t_o) and success rate in simulation, detailed in Table III.

Our method achieves a significant improvement in optimization time, as well as better success rate. Besides, the trajectory generated by our method is dynamic feasible for the mobile manipulator while the one generated by CHOMP is not.

B. Real-world Door-Pushing Experiment

To demonstrate the effectiveness of our method in handling task constraints, we conduct a real-world door-pushing experiment. During pushing the door, the end-effector is required to remain relatively stationary and perpendicular to the door. Namely, the trajectory of the end-effector has to be a circle with constant height. In this experiment, $d_p = 0.05(m)$ and $d_o = 0.15(rad)$. Fig. 7 shows the process of pushing the door in real world. As demonstrated in Fig. 8, initially, the error between the target pose and the trajectory of the end-effector exceeds the tolerance limit. Then, the error quickly diminishes and remains within the tolerance.

TABLE III: TRAJECTORY GENERATION RESULT

	method	$t_o(ms)$	$succ$	$feasible$
Cuboids	Ours	**294.17**	**92.35%**	✓
	CHOMP	5791.48	68.20%	✗
Bridge	Ours	**175.67**	**100.00%**	✓
	CHOMP	2279.35	98.79%	✗

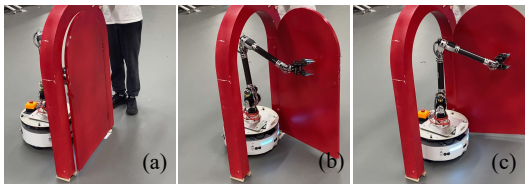

Fig. 7: The snap shot of door pushing experiment with collision-avoiding in real world.

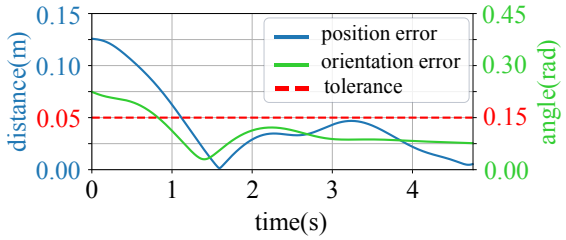

Fig. 8: Position and orientation error between target value and trajectory for door pushing task.

C. Real-world Planning in Dense Environment

To validate the feasibility of the proposed framework on a real mobile manipulator, we conduct real-world experiments in a $7m \times 5m$ craft room with tables, shelves and some square obstacles, as shown in Fig. 1. Given the goal state $\boldsymbol{x}_{goal} \in \mathbb{R}^{3+\mathcal{L}}$, the mobile manipulator is required to determine a trajectory originating from the current state that leads to $\boldsymbol{x}_{goal}$. Our platform, equipped with an onboard computer (Intel Core i7-1260P CPU), is illustrated in Fig. 2. The state of the mobile base is acquired using a motion capture system, while the manipulator's state is obtained through the built-in encoder. The point cloud map of the environment is pre-constructed using FAST-LIO2 [26]. We conduct six experiments with different start and goal states. One of the experiments is shown in Fig. 1, which shows the agile, coupled and safe motion of the mobile manipulator. The complete experiments can be viewed in the attached video.

VII. Conclusion

In this paper, we present a real-time whole-body motion planning framework for mobile manipulators. We propose an efficient environment-adaptive path searching method, which achieves high-quality, real-time path planning while ensuring completeness. Additionally, we propose a spatial-temporal trajectory optimization method that takes safety, dynamic feasibility, and task constraints into account. We conduct extensive simulations and real-world experiments to demonstrate the effectiveness of our method. In the future, we will extend this framework to cooperative transportation.

REFERENCES

[1] M. Hvilshøj, S. Bøgh, O. Skov Nielsen, and O. Madsen, "Autonomous industrial mobile manipulation (aimm): past, present and future," *Industrial Robot: An International Journal*, vol. 39, no. 2, pp. 120–135, 2012.

[2] V. Pilania and K. Gupta, "A hierarchical and adaptive mobile manipulator planner with base pose uncertainty," *Autonomous Robots*, vol. 39, pp. 65–85, 2015.

[3] J. Kuffner and S. LaValle, "RRT-connect: An efficient approach to single-query path planning," in *Proc. IEEE Int. Conf. Robot. Autom.*, vol. 2, 2000, pp. 995–1001.

[4] S. Klemm, J. Oberländer, A. Hermann, A. Roennau, T. Schamm, J. M. Zollner, and R. Dillmann, "RRT*-connect: Faster, asymptotically optimal motion planning," in *2015 IEEE international conference on robotics and biomimetics (ROBIO)*. IEEE, 2015, pp. 1670–1677.

[5] S. Thakar, P. Rajendran, H. Kim, A. M. Kabir, and S. K. Gupta, "Accelerating bi-directional sampling-based search for motion planning of non-holonomic mobile manipulators," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 6711–6717.

[6] M. Zucker, N. Ratliff, A. D. Dragan, M. Pivtoraiko, M. Klingensmith, C. M. Dellin, J. A. Bagnell, and S. S. Srinivasa, "Chomp: Covariant hamiltonian optimization for motion planning," *The International journal of robotics research*, vol. 32, no. 9-10, pp. 1164–1193, 2013.

[7] J. Schulman, J. Ho, A. X. Lee, I. Awwal, H. Bradlow, and P. Abbeel, "Finding locally optimal, collision-free trajectories with sequential convex optimization." in *Robotics: science and systems*, vol. 9, no. 1. Berlin, Germany, 2013, pp. 1–10.

[8] Z. Wang, X. Zhou, C. Xu, and F. Gao, "Geometrically constrained trajectory optimization for multicopters," *IEEE Transactions on Robotics*, vol. 38, no. 5, pp. 3259–3278, 2022.

[9] D. Berenson, J. Kuffner, and H. Choset, "An optimization approach to planning for mobile manipulation," in *2008 IEEE International Conference on Robotics and Automation*. IEEE, 2008, pp. 1187–1192.

[10] E. Marder-Eppstein, E. Berger, T. Foote, B. Gerkey, and K. Konolige, "The office marathon: Robust navigation in an indoor office environment," in *2010 IEEE international conference on robotics and automation*. IEEE, 2010, pp. 300–307.

[11] A. Hornung, M. Phillips, E. G. Jones, M. Bennewitz, M. Likhachev, and S. Chitta, "Navigation in three-dimensional cluttered environments for mobile manipulation," in *2012 IEEE International Conference on Robotics and Automation*. IEEE, 2012, pp. 423–429.

[12] S. Chitta, B. Cohen, and M. Likhachev, "Planning for autonomous door opening with a mobile manipulator," in *2010 IEEE International Conference on Robotics and Automation*. IEEE, 2010, pp. 1799–1806.

[13] S. LaValle, "Rapidly-exploring random trees: A new tool for path planning," *Research Report 9811*, 1998.

[14] L. Janson, E. Schmerling, A. Clark, and M. Pavone, "Fast marching tree: A fast marching sampling-based method for optimal motion planning in many dimensions," *The International journal of robotics research*, vol. 34, no. 7, pp. 883–921, 2015.

[15] S. Karaman and E. Frazzoli, "Optimal kinodynamic motion planning using incremental sampling-based methods," in *49th IEEE conference on decision and control (CDC)*. IEEE, 2010, pp. 7681–7687.

[16] D. Berenson, S. Srinivasa, and J. Kuffner, "Task space regions: A framework for pose-constrained manipulation planning," *The International Journal of Robotics Research*, vol. 30, no. 12, pp. 1435–1460, 2011.

[17] C. P. Tang, P. T. Miller, V. N. Krovi, J.-C. Ryu, and S. K. Agrawal, "Differential-flatness-based planning and control of a wheeled mobile manipulator—theory and experiment," *IEEE/ASME Transactions on Mechatronics*, vol. 16, no. 4, pp. 768–773, 2010.

[18] M. Kalakrishnan, S. Chitta, E. Theodorou, P. Pastor, and S. Schaal, "Stomp: Stochastic trajectory optimization for motion planning," in *2011 IEEE international conference on robotics and automation*. IEEE, 2011, pp. 4569–4574.

[19] J. Pankert and M. Hutter, "Perceptive model predictive control for continuous mobile manipulation," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6177–6184, 2020.

[20] M. Spahn, B. Brito, and J. Alonso-Mora, "Coupled mobile manipulation via trajectory optimization with free space decomposition," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 12 759–12 765.

[21] Z. Han, Y. Wu, T. Li, L. Zhang, L. Pei, L. Xu, C. Li, C. Ma, C. Xu, S. Shen, and F. Gao, "An efficient spatial-temporal trajectory planner for autonomous vehicles in unstructured environments," 2023.

[22] M. Zhang, C. Xu, F. Gao, and Y. Cao, "Trajectory optimization for 3d shape-changing robots with differential mobile base," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 10 104–10 110.

[23] D. Dolgov, S. Thrun, M. Montemerlo, and J. Diebel, "Path planning for autonomous vehicles in unknown semi-structured environments," *The international journal of robotics research*, vol. 29, no. 5, pp. 485–501, 2010.

[24] B. Zhou, F. Gao, L. Wang, C. Liu, and S. Shen, "Robust and efficient quadrotor trajectory generation for fast autonomous flight," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3529–3536, 2019.

[25] J.-P. Sleiman, F. Farshidian, and M. Hutter, "Versatile multicontact planning and control for legged loco-manipulation," *Science Robotics*, vol. 8, no. 81, p. eadg5014, 2023.

[26] W. Xu, Y. Cai, D. He, J. Lin, and F. Zhang, "Fast-lio2: Fast direct lidar-inertial odometry," *IEEE Transactions on Robotics*, vol. 38, no. 4, pp. 2053–2073, 2022.