RAMPAGE: Towards Whole-body, Real-Time and Agile Motion Planning in Unknown Cluttered Environments for Mobile Manipulators

Yuqiang Yang, Fei Meng, Zehui Meng, Chenguang Yang, *Fellow, IEEE*

***Abstract*—It's an open problem for a mobile manipulator (MM) with a differential mobile base (MB) to quickly detect surrounding obstacles using onboard sensors and find whole-body collision-free trajectories under kinematic constraints in unknown cluttered environments. To this end, we propose a framework integrating perception, planning, and control, named RAMPAGE. First, we utilize a layered map including an occupancy grid map and Euclidian signed distance field to realize one-shot collision checking between obstacles and the MM, whose body is approximated by sphere decomposition. Second, based on the guidance of different topological paths, a novel hierarchical trajectory searching method is developed to find an initial whole-body kinodynamic trajectory for the MM. Finally, we derive and implement a whole-body trajectory optimization method in an augmented Lagrangian differential dynamic programming (AL-DDP) form, yielding an optimal feedforward-feedback controller for free during the optimization. Extensive benchmark comparisons and experiment demonstrate that the proposed framework can find near-optimal whole-body collision-free trajectories in real time with a high success rate. The high-frequency controller accurately tracks them at an average speed of 1.86m/s among unknown dynamic obstacles with a maximum speed of 2m/s in simulation. Pick-and-place experimental results further verify our method's effectiveness and efficiency in the real world.** [1]

***Index Terms*—mobile manipulator, motion planning.**

I. Introduction

MOBILE manipulator (MM) robots have shown great potential to assist people by incorporating the benefits of mobility from a mobile base (MB) and dexterity from a manipulator. They are required to pick some objects and place them somewhere else without colliding with their surroundings. However, the unstructured indoor environment containing tables, chairs, and pedestrians poses safety and efficacy concerns, since it makes the robot hard to quickly find and accurately execute a feasible trajectory in the constrained workspace. Specifically, there are three main challenges to safely and efficiently accomplish tasks for the MM in unstructured environments containing unknown static or dynamic obstacles: 1) conquering limited perception capability of onboard sensors and non-convex body, i.e., to build a useful environmental map and check collision in real time, 2) enhancing motion planning efficiency under constraints, i.e., to find solutions in a high-dimensional state space (imposed by the MB and manipulator hybrid system) under kinematic and obstacle constraints, 3) deriving a high-performance control strategy, i.e., to improve the tracking accuracy and robustness when executing the planned trajectory in highly dynamic environments [1].

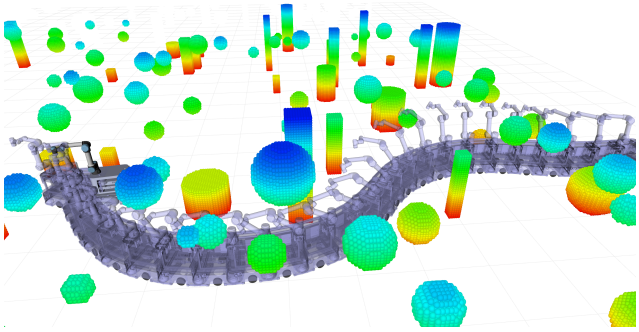

Fig. 1. Planned whole-body trajectory using proposed RAMPAGE.

Existing studies [2–5] usually use a single-layer occupancy map (OGM) to represent the environment and then decompose the obstacles and the MM body into a series of convex polyhedra for collision checking. However, the clutter of complex environments and the nonconvexity of the MM body lead these one-to-one checking methods to be quite time-consuming. After acquiring the body and obstacles' information in the environment, early work tends to separately plan a trajectory to let MB approach the target and then plan the robot arm's motions for manipulation [6–8]. Although there are off-the-shelf planning solutions to the subproblems, such as graph search methods [9] for the MB and trajectory optimization algorithms [10, 11] for the manipulator, the total solution space is dramatically reduced, resulting in low manipulability. Significant research on whole-body MM was derived recently [4, 12–16]. Nevertheless, few works can achieve reliable whole-body collision avoidance in real time

Manuscript received October 12, 2023; revised January 5, 2024 and January 31, 2024; Accepted February 19, 2024(Corresponding author is Chenguang Yang.)

Yuqiang Yang is with School of Automation Science and Engineering, South China University of Technology, Guangzhou, 510641, China.

Fei Meng is with the Department of Electronic Engineering, The Chinese University of Hong Kong, Hong Kong SAR, China.

Zehui Meng is with Application Innovation Laboratory, Huawei Technologies Co. Ltd., Shenzhen, China.

Chenguang Yang is with Department of Computer Science, University of Liverpool, Liverpool, L69 3BX, UK. Email: cyang@ieee.org.

[1]Supplementary video and materials for benchmark are available at https://github.com/yuqiang-yang/TIE-Supplementary-video.

under collision and kinematic constraints due to their poor trade-off between probabilistic completeness and computation costs. For example, [16] takes a method similar to full-state sampling to search the trajectory for the MM, which has a high success rate but takes dozens of seconds. [12] simplies uses a collision-free trajectory of the end-effector to initialize the optimization, which takes little time but has a low success rate in cluttered environments. Furthermore, studies like [6, 16] are only suitable for well-defined static environments rather than more practical dynamic ones because they plan a path without any time information instead of a kinodynamic trajectory like ours. In addition, existing studies usually design planning and control modules independently, resulting in a poor tracking performance of the MM due to the error accumulation and unfit parameters.

To address the problems mentioned above, the main contribution of this paper is three-fold. First, a real-time layered map containing an OGM as a low-level map and an ESDF as a high-level map is used to represent the obstacle occupancy and distance information. Given the spherical decomposition of the MM body, our collision detection process maintains a consistent short run time, regardless of increasing environmental complexity. Second, we propose a novel hierarchical trajectory searching method. Various topological paths guide the MB to explore different topological areas, and then spatial-temporal rapidly exploring random tree (ST-RRT) finds feasible trajectories in the high dimensional spatial-temporal configuration space for the MM. The proposed techniques can determine an initial trajectory within approximately 70 ms, boasting a high success rate while maintaining an optimal balance between efficacy and planning speed. Third, we formulate an integrated planning and control (IPC) optimization problem in the AL-DDP form to get the optimal feedforward-feedback (F-F) control policy for the MM in unknown cluttered environments. It efficiently optimizes and tracks a whole-body trajectory without any finely tuned controller, such as the PID and LQR. Finally, extensive experiments and benchmark comparisons are conducted, showing that the MM takes about 100 ms time in one planning round to plan the high-speed trajectory while avoiding collisions against dynamic obstacles at a high success rate.

II. Related works

The majority of literature divides the mobile manipulation problem into sequential subproblems. The authors in [8] used the Djikstra algorithm to search a collision-free trajectory for the MB to approach the goal, then employed the RRT-Connect to find a feasible solution for the manipulator to grasp the object. A location-aware sampling and connection strategy was proposed to plan the MB trajectory at first and simultaneously check if the manipulator collides with obstacles during the motion of MB [6]. If so, the MM was reconfigured to a free space with a fixed MB. However, the non-holonomic constraint of the MB is not considered during the planning, making the MB frequently turn in place to execute the planned trajectory. Similarly, the MM motion planning problem was solved by finding solutions for the MB and manipulator separately [7]. Although the above separate planning methods are easy to implement, they are inefficient for manipulation.

Recent works focus on whole-body collaborative motion planning for MM. [4] treats the MM with an omnidirectional MB as a generalized high-dimensional manipulator and directly samples in that space to avoid dynamic obstacles. However, the motions of all obstacles need to be predefined, and the common non-holonomic constraint of MB is not considered. In [13, 14], the authors formulated the whole-body manipulation task as a quadratic program problem, in which the manipulability, orientation, smoothness, and tracking performance are optimized. However, it is hard to robustly avoid collisions in dynamic cluttered environments since the methods, reactive controllers in essence, do not take prediction and pre-reaction into consideration. [17] decomposes the free space of the environment with a set of convex polyhedrons and restricts every link in the polyhedrons during planning. [16] improves bi-directional full-state sampling efficiency through attention techniques but still takes tens of seconds to plan a collision-free path in a cluttered factory. The works most similar to ours are [12, 15] which check collision by ESDF and address the whole-body collision-free manipulation problem by the model predictive control (MPC). However, their planning or control procedure often gets stuck in some infeasible local minima in cluttered environments because only a collision-free end-effector trajectory is used to guide the optimizatiotn. Besides, they use a barrier function to wrap the collision cost in the Euclidian Signed Distance Fields (ESDF), whose value will increase exponentially and cause numerical instability during optimization when the MMs body approaches the obstacles. To the best of our knowledge, no other publication has achieved real-time agile motion planning for MMs in cluttered environments using solely on-board sensors.

III. Problem Formulation

Consider a MM composed of a differential-drive MB and a six DoFs manipulator. $\mathcal{C} = \mathcal{C}_{free} \cup \mathcal{C}_{obs}$ represents MM's configuration space containing the collision-free and collision subspaces. $\boldsymbol{q}(t) = [\boldsymbol{q}_b^{\mathrm{T}}\ \boldsymbol{q}_m^{\mathrm{T}}]^{\mathrm{T}} \in \mathcal{C}$ denotes the configuration vector of MM, where $\boldsymbol{q}_b = [x\ y\ \theta]^{\mathrm{T}} \in \mathbb{R}^3$ is the configuration including the position and orientation of MB on the ground and $\boldsymbol{q}_m = [q_1, \ldots, q_6]^{\mathrm{T}} \in \mathbb{R}^6$ is that representing joint angles of the manipulator.

Then, we define the state vector of MM as $\boldsymbol{x}(t) := [\boldsymbol{q}^{\mathrm{T}}\ \boldsymbol{v}^{\mathrm{T}}]^{\mathrm{T}} \in \mathcal{X} \subset \mathbb{R}^{17}$, where $\mathcal{X}$ is the state space and $\boldsymbol{v} = [v_b\ \omega_b\ \dot{\boldsymbol{q}}_m^{\mathrm{T}}]^{\mathrm{T}}$ is the velocity of MM. v_b and ω_b are the linear and angular velocities of the MB, respectively. We aim to find a collision free and smooth trajectory $\boldsymbol{\pi} = \{\boldsymbol{x}(0), \ldots, \boldsymbol{x}(t_f)\}$ connecting the start $\boldsymbol{q}_s$ and goal $\boldsymbol{q}_g$ configurations. We formulate the integrated planning and control problem (IPC) as:

$$\begin{aligned} \arg\min_{\boldsymbol{x},\boldsymbol{u}}\ & \varphi(\boldsymbol{x}(t_f), t_f) + \int_{t_0}^{t_f} \underbrace{\boldsymbol{u}^T \boldsymbol{R} \boldsymbol{u} + \bar{\boldsymbol{x}}^T \boldsymbol{Q} \bar{\boldsymbol{x}} + o(\boldsymbol{x})}_{l(\boldsymbol{x}(t), \boldsymbol{u}(t), t)} dt \\ s.t.\ & \dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}) \qquad \bar{\boldsymbol{x}} = \boldsymbol{x} - \boldsymbol{x}_0 \\ & \boldsymbol{u}_{min} < \boldsymbol{u} < \boldsymbol{u}_{max} \qquad \boldsymbol{x}_{min} < \boldsymbol{x} < \boldsymbol{x}_{max} \end{aligned} \tag{1}$$

where $\boldsymbol{u}(t) = [a_v, a_w, \ddot{\boldsymbol{q}}_m^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{R}^8$ is the acceleration input of MM, $\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u})$ is system kinematics model, $o(\boldsymbol{x})$ is the cost for collision avoidance, and $\boldsymbol{x_0}$ is the initial trajectory to guide the optimization to converge to collision-free local minima. $\varphi(\boldsymbol{x}(t_f), t_f)$ is the terminal cost and is similar to the integrand but without the part about u.

IV. Methodology

The inputs of our framework are the environment point-cloud from on-board sensors, the odometry of MB, current and target states of MM. Subsection IV-A reconstructs the obstacle information through a layered map and makes one-shot collision checking. Subsection IV-B finds various feasible kinodynamic MB trajectories in differential topological areas, based on which subsection IV-C samples in the spatial-temporal state space to quickly obtain an initial whole-body trajectory. Furthermore, subsection IV-D derives an AL-DDP solver for the IPC to optimize the whole-body trajectories and gain the optimal F-F controller.

A. Layered Map Construction and Collision Checking

The low-level map is an occupancy grid map (OGM), which records the probability of whether some obstacles occupy a certain position. After a frame of point cloud and corresponding odometry are fed, the raycasting algorithm [18] will traverse a ray's traveled grids from the camera to a certain obstacle point. The occupancy probability of that obstacle point's grid will be increased while the probability of all the grids between the camera and the point will be decreased. The high-level map is an Euclidian signed distance field (ESDF), which records the distance from the current voxel to its nearest obstacle. Based on the real-time updated OGM, the ESDF is updated at 20 Hz using an $O(N)$ time complexity algorithm [19]. It's worth noting that the real-time updated layered map enables the MM to perceive dynamic obstacles even though no special attention is paid to predicting obstacle tracks.

Fig. 2 illustrates a planar ESDF example. Combined with the sphere decomposition that approximates the MM body with several spheres, collision checking can be easily done by comparing the spheres' radii with corresponding ESDF values at their centers.The sphere center positions $\boldsymbol{c_i}$ can be calculated by truncating the forward kinematics chain of the spheres $\mathcal{FK}_i(\boldsymbol{x})$ as follows,

$$\boldsymbol{c}_i = \mathcal{FK}_i(\boldsymbol{x}) \quad i = 1, 2, ..., M. \tag{2}$$

We define $\mathcal{S}$ as the set of spheres and deem there is no self-collision if the centers satisfy the following inequations,

$$d(\boldsymbol{c}_i) - r_i > 0, \forall i \tag{3}$$

$$||\boldsymbol{c}_i - \boldsymbol{c}_j|| - r_i - r_j > 0, \forall i, j \in \mathcal{S}, \tag{4}$$

where r_i is the radius of the ith sphere and $d(c_i)$ is the distance to its nearest obstacle that can be directly read from ESDF.

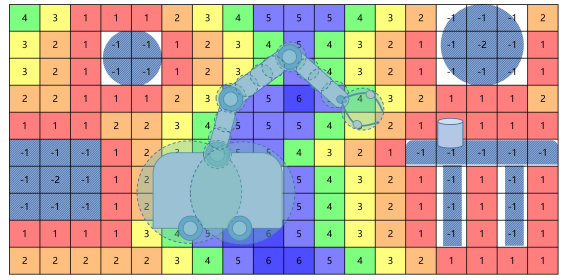

Fig. 2. A planar ESDF example of a mobile manipulator grasping a can on a table. The desk, circle, and square-shaped grids in dark blue are obstacles. The dashed circles are sphere decomposition results of the robot body approximation. The value shown in every grid represents the distance between that grid and its nearest obstacle.

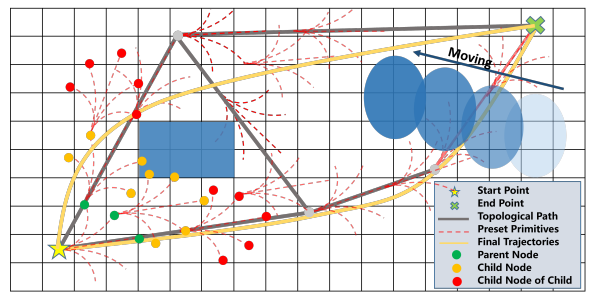

Fig. 3. A simple example for the topo-guided kinodynamic trajectory searching of Algorithm 1. The grey curves are topological paths to guide the kinodynamic searching. The dashed red curves are preset motion primitives to propagate the state. Blue rectangle and ellipsoid are static and dynamic obstacles, respectively.

The penalty function to keep the MM away from obstacles in (1) is derived as follows,

$$o(x) = \sum_{i=1,..,M} p(d(\boldsymbol{c}_i) - r_i) + \sum_{i,j \in \mathcal{S}} p(||\boldsymbol{c}_i - \boldsymbol{c}_j|| - r_i - r_j) \tag{5}$$

$$p(y) = \begin{cases} 0 & y > s_d \\ (s_d - y)^3 & 0 < y < s_d \\ 3s_d(s_d - y)^2 - 3s_d^2(s_d - y) + s_d^3 & y < 0, \end{cases} \tag{6}$$

where s_d is the desired clearance between the robot body and obstacles. The $\frac{\partial d}{\partial c}$ can be numerically calculated by trilinear interpolation in ESDF while the $\frac{\partial c}{\partial x}$ is the common jacobian matrix at given points of the MM. Therefore, the collision cost gradient w.r.t. state $\boldsymbol{x}$ during optimization can be subsequently retrieved using the chain rule.

B. Topo-guided Kinodynamic Searching for MB

Traditional hierarchical planning frameworks suffer from probabilistic incompleteness problems, i.e., they fail to find a feasible solution even if one exists in complex environments. In our framework, different topological paths are used to guide the kinodynamic searching to explore different areas, significantly increasing the effectiveness of handling cluttered static and dynamic obstacles. A simple example is shown in Fig. 3. The algorithm is shown in Alg. 1, where $\boldsymbol{x}_{b0}$ and $\boldsymbol{x}_{bg}$ are the MB's initial and goal states, respectively.

The generation of topological paths is to sample randomly to build a road map and then prune based on topological

Algorithm 1 Topo-guided Kinodynamic Trajectory Searching

Input: $\boldsymbol{x}_{b0}$ $\boldsymbol{x}_{bg}$, resolution, ESDF map
Output: multiple $\boldsymbol{x}_b^i(t)$ $t \in [t_0^i, t_g^i]$

1: $\mathcal{TP} \leftarrow$ sampleTopoPath()
2: $\mathcal{O} \leftarrow \emptyset$, $\mathcal{G}$.addNode(Node(x_{b0}, t_0))
3: **for** $i = 0, \ldots, size(\mathcal{TP}) - 1$ **do**
4: $\quad\mathcal{O}$.pushAllNode($\mathcal{G}$)
5: $\quad$**while** $\mathcal{O}$ is not empty **do**
6: $\quad\quad n_0 \leftarrow \mathcal{O}$.popOrderedNode()
7: $\quad\quad$**if** shot(n_0.state, x_g) **then**
8: $\quad\quad\quad$break $\quad\quad\quad

Algorithm 2 Spatial-Temporal RRT

Input: $\boldsymbol{x_0}, \boldsymbol{x_g}$, $\boldsymbol{x}_b(t)$ $t \in [t_0, t_g]$
Output: $\boldsymbol{x(t)}$ $t \in [t_0, t_g]$

```
T ← ∅, T.addNode(Node(x_0, t_0))
repeat
    if uniformSample(0,1) < goalBias then
        n_s = Node(x_g, t_g)
    else
        n_s = randomSample()
    n_n = T.findNearest(n_s)  ▷ Spatial-Temporal distance
    if ||n_s.x_a − n_n.x_a||_∞ > q̇_max Δt then
        n_s.t = n_n.t + ||Δx/v_max||_∞
    if d_st(n_n, n_s) > d_max then
        n_e = interpolate(n_n, n_s, d_max/d_st(n_n, n_s))
    else
        n_e = n_s
    n_e.x_b = x_b(n_e.t)
    if checkMotionValid(n_n, n_e) then
        n_e.parent = n_n
        T.addNode(n_e)
until Node(x_g, t_g) in T
x(t) = T.postprocess()
return x(t)
```

The trajectory $\boldsymbol{\pi}$ found by naive ST-RRT is usually curvy and of bad quality, which needs a post process. As shown in Fig. 4.c, we merge the nodes close enough with a radius r to make $\boldsymbol{\pi}$ smooth. Besides, we randomly choose two nodes of $\mathcal{T}$ and try to make a shortcut between them. If feasible, the nodes and edges between them are eliminated to get a shorter trajectory.

D. AL-DDP Solver for integrated planning and control

1) Motivation: In this section, we will derive an AL-differential dynamic programming (AL-DDP) based optimization solver for the IPC problem (1). Existing solvers can be mainly divided into direct and indirect methods. Direct methods usually discretize all optimization variables into many knot points and optimize in a high-dimensional full-state space. To get the optimal feedforward policy, the optimization problems are usually solved via some off-the-shelf nonlinear programming (NLP) or sequential quadratic programming (SQP) solvers such as OSQP, IPOPT, and HPIPM. Direct methods are easy to implement and handle constraints but suffer from $O(N^3m^3)$ time complexity due to the inverse of the high-dimensional Hessian matrix, where N is the number of time steps, and m is the system dimension. Furthermore, the control policy $(u(t_k), x(t_k))$ cannot be attained until the optimization is totally converged when the system dynamical constraint $f(x)$ is satisfied. These disadvantages limit the real-time planning for MM in cluttered and dynamic environments. On the contrary, DDP, a popular indirect method, exploits the Markov chain form of the optimal control problem to decompose the large optimization problem into a series of small ones. It's fast and has a linear time complexity of $O(Nm^3)$, facilitating real-time control. Furthermore, both the optimal feedforward and feedback policy can be gained through DDP, even though the recursion is not finished. The original version of DDP is hard to handle any constraints because the resolved recursion of dynamic programming will be broken. In this paper, we extend the original DDP with augmented Lagrangian methods to handle constraints, saving time for solving general problems.

2) Derivation: Let all the inequality and equality constraints in (1) be g and h, respectively. To deal with the constraints except for the kinematics, we utilize the augmented Lagrangian (AL) method. The cost function is augmented with the constraint terms, and (1) becomes

$$\min_{\boldsymbol{x},\boldsymbol{u}} L_f + \int_{t_0}^{t_f} \underbrace{l + \tilde{\mu}^{\mathrm{T}} g + \tilde{\lambda}^{\mathrm{T}} h + \frac{\rho}{2} h^{\mathrm{T}} h + \sum_{i=1}^{N_{\text{ine}}} \max(g_i, 0)}_{L_A(\boldsymbol{x}(t),\boldsymbol{u}(t),t)} \, dt \tag{9}$$

where $\tilde{\lambda}$ and $\tilde{\mu}$ are the estimated Lagrangian multipliers, ρ is the penalty weight, and N_{ine} is number of inequality. We solve the new problem in an alternating fashion, i.e., $\tilde{\lambda}$, $\tilde{\mu}$, and ρ remain fixed and we solve the continuous optimal control problem through DDP recursion, then in the next iteration, $\tilde{\lambda}$, $\tilde{\mu}$ will be updated according to the constraint violation. For the DDP recursion, we define a Cost-to-Go function as:

$$V(\boldsymbol{x}, t) = \min_{\boldsymbol{u}(t)} L_F(\boldsymbol{x}(t_f), t_f) + \int_t^{t_f} L_A(\boldsymbol{x}(t), \boldsymbol{u}(t), t) dt. \tag{10}$$

The Hamiltonian of the system is:

$$\mathcal{H}(\boldsymbol{x}(t), \boldsymbol{u}(t), V_x, t) = L_A + V_x^{\mathrm{T}} f, \tag{11}$$

where $V_x = \partial V / \partial \boldsymbol{x}$ plays the role of the costate. Then, we have the following Hamilton-Jacobi-Bellman partial differential equation (HJB PDE):

$$-\frac{\partial V(\boldsymbol{x}(t), t)}{\partial t} = \min_{\boldsymbol{u}(t)} \mathcal{H}(\boldsymbol{x}(t), \boldsymbol{u}(t), V_x, t). \tag{12}$$

Since it is hard to solve (12) for general V and f, we use a DDP method with Taylor expansion to make it tractable.

The system kinematics is linearized along the nominal trajectory at first:

$$f(\bar{x} + \delta x, \bar{x} + \delta u, t) \approx f(\bar{x}, \bar{u}, t) + A(t)\delta x + B(t)\delta u. \tag{13}$$

The Cost-to-Go function is quadratized as below:

$$V(\bar{x} + \delta x, t) \approx V(\bar{x}, t) + V_x^{\mathrm{T}} \delta x + \frac{1}{2} \delta x^{\mathrm{T}} V_{xx} \delta x, \tag{14}$$

where $V_{xx} = \partial^2 V / \partial \boldsymbol{x}^2$. Similarly, we have

$$\begin{aligned} L_A(\bar{x} + \delta x, \bar{u} + \delta u, t) \approx{}& L(\bar{x}, \bar{u}, t) + L_x^{\mathrm{T}} \delta x + L_u^{\mathrm{T}} \delta u \\ &+ \frac{1}{2} \begin{bmatrix} \delta x \\ \delta u \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} L_{xx} & L_{xu} \\ L_{ux} & L_{uu} \end{bmatrix} \begin{bmatrix} \delta x \\ \delta u \end{bmatrix}. \end{aligned} \tag{15}$$

Algorithm 3 continuous-time AL-DDP

Input: $f, g, h, J, \rho, \bar{x}, \bar{u}$
Output: $u^*(t) = d(t) + K(t)(x - \bar{x})$

Initialize $\tilde{\mu}, \tilde{\lambda}$
repeat
  Inner Loop:
  $\bar{x}(t) \leftarrow \text{rollout}(\bar{u}(t))$ ▷ System dynamics
  Linearize the system dynamics f at $\bar{x}, \bar{u}$
  Quadratize $V(x,t)$ and AL $L_A(x,u,t)$ at $\bar{x}, \bar{u}$
  $V(x,t) = \text{backwardPass}()$ ▷ Riccati equation
  $\delta u^* \leftarrow d(t) + K(t)\delta x$ ▷ After regularization
  $\alpha \leftarrow \text{linesearch}(L_A, \delta u^*)$ ▷ Backtracking Armijo rule
  $\bar{u} \leftarrow \bar{u} + \alpha \delta u^*$
  Outer Loop:
  $\tilde{\mu} \leftarrow \max(0, g(x,u)), \tilde{\lambda} \leftarrow h(x,u)$
  $\rho \leftarrow \mu\rho$
until converged $||d||_\infty < \epsilon_d$
return $K(t), d(t)$

By rewriting the left hand side of (12) through the chain rule and substituting (14) and (15) into (11), we obtain a new HJB PDE as follows:

$$\begin{aligned} -\frac{d}{dt}\Big(\bar{V} + V_x^{\mathrm{T}}\delta x + \frac{1}{2}\delta x^{\mathrm{T}} V_{xx}\delta x\Big) &= \min_{\delta u}\Big(\bar{L} + L_x^{\mathrm{T}}\delta x \\ + L_u^{\mathrm{T}}\delta u + \frac{1}{2}\begin{bmatrix}\delta x \\ \delta u\end{bmatrix}^{\mathrm{T}} &\begin{bmatrix} L_{xx} & L_{xu} \\ L_{ux} & L_{uu}\end{bmatrix}\begin{bmatrix}\delta x \\ \delta u\end{bmatrix} + V_x^{\mathrm{T}} A\delta x \\ + V_x^{\mathrm{T}} B\delta u + \delta x^{\mathrm{T}} V_{xx} A\delta x &+ \delta x^{\mathrm{T}} V_{xx} B\delta u\Big), \end{aligned} \tag{16}$$

where $\bar{V} = V(\bar{x}, t)$ and $\bar{L} = L(\bar{x}, \bar{u}, t)$.

The **necesary** condition for the minimization on the right hand size of (16) is that its gradient w.r.t δu^* equals to zero:

$$\begin{aligned} 0 &= L_{uu}\delta u^* + (L_u + B^T V_x) + (L_{ux} + B^T V_{xx})\delta x \\ \delta u^* &= -L_{uu}^{-1}(L_u + B^T V_x) - L_{uu}^{-1}(L_{ux} + B^T V_{xx})\delta x \\ \delta u^* &= d(t) + K(t)\delta x, \end{aligned} \tag{17}$$

where $L_{ux} = L_{xu}^{\mathrm{T}}$.

By substituting δu^* into (16), we have:

$$\begin{aligned} -\frac{d}{dt}\Big(\bar{V} + V_x^T\delta x + \frac{1}{2}\delta x^T V_{xx}\delta x\Big) &= \Big(\bar{L} + (\frac{1}{2}d^T L_{uu} \\ + V_x^T B + L_u^T)d\Big) + \delta x^T\Big(L_x + A^T V_x + K^T(B^T V_x &\quad (18) \\ + L_u) + (L_{xu} + K^T L_{uu} + V_{xx}B)d\Big) + \frac{1}{2}\delta x^T \mathcal{H}_{xx}\delta x. \end{aligned}$$

Finally, we obtain the following **Riccati equation** of the Cost-to-Go function $V(\boldsymbol{x}, t)$ by comparing the coefficients of the constant, linear and quadratic terms:

$$\begin{aligned} -\frac{dV}{dt} &= \bar{L} - \frac{1}{2}d^{\mathrm{T}} L_{uu} d \\ -\frac{dV_x}{dt} &= L_x + A^{\mathrm{T}} V_x - K^{\mathrm{T}} L_{uu} d \\ -\frac{dV_{xx}}{dt} &= L_{xx} - K^{\mathrm{T}} L_{uu} K + A^{\mathrm{T}} V_{xx} + V_{xx} A. \end{aligned} \tag{19}$$

The terminal value of the V, V_x, V_{xx} are determined by L_F. Consequently, the inner loop of AL-DDP can be solved with the fixed Lagrange multipliers. After that, the AL method gradually updates $\tilde{\mu}$, $\tilde{\lambda}$, and ρ according to the constraints satisfaction. We describe the details of the entire continuous-time AL-DDP in Alg. 3, where $\tilde{\mu}$ and $\tilde{\alpha}$ are the estimated Lagrange multiplier to relax the constraints, and α is a constant step size. Note that in the traditional DDP for trajectory optimization, the inner loop is iterated until the optimization fully converges to a local minima, where the optimal control policy will be a pure linear feedback policy with no feedforward term (actually, it's a time-varying LQR controller [22]). However, in our real-time whole-body collision avoidance application, we only iterate AL-DDP to converge at the beginning fully. When the MM runs, the inner loop only iterates several times if the trajectory has already been feasible. This strategy has proven to work well with a reasonable warm start [12].

3) Convergence Analysis: We first analyze the convergence condition of inner loop in Alg. 3.

Theorem 1: For a general continuous optimization problem with nonlinear $f(x)$, $g(x)$, and $h(x)$, the inner loop will finally converge to local minima if L_{uu} is positive definite.

Proof. Denote the cost-to-go function $\bar{V}$ as $\bar{V}_i$ and $\bar{V}_{i+i}$ in one iteration. From definition (10) we know $-\frac{d\bar{V}}{dt} = \bar{L}$. Given a regularized positive definite L_{uu}, the delta value of $\bar{V}$ after a DDP backward pass (19) is:

$$\Delta\bar{V}(\bar{x}, t_0) = \bar{V}_{i+1} - \bar{V}_i = \int_{t_f}^{t_0} -\frac{1}{2}d^{\mathrm{T}} L_{uu} d \; dt < 0. \tag{20}$$

Therefore, the cost from start to goal always decreases in each DDP recursion, the sum value of which is positively correlated with the optimal feedforward policy $d(t)$. The convergence flag for the inner loop can be set as $||d||_\infty$ or $\Delta\bar{V}$ are small enough.

A simple but effective regularization way is to add a constant to the diagonal elements of L_{uu} and make it positive definite in each iteration, i.e. $L_{uu} = L_{uu} + \epsilon I$. □

Theorem 2: The inner loop will finally converge to the global minima with a **quadratic convergence rate** if 1) $f(x)$ is linear, 2) L_A, L_F are convex, and 3) they have continuous second-order derivatives.

Proof. HJB equation provides necessary and sufficient conditions for optimality of a continuous optimal control problem without terminal constraints. A linear $f(x)$ and convex cost function L mean that the Hamiltonian (16) is a linear combination of convex functions. Theorem 1 has shown that DDP recursion with positive definite L_{uu} will eventually converge to a local minima. $L_{uu} \succeq 0$ always holds for a convex function, and the local minima are exactly the global minima.

The AL method in the outer loop relaxes the constraint violation penalty into the Lagrange multiplier term. Compared to penalty methods that gradually increase the penalty weights ρ to satisfy a constraint well, it prevents the Hessian matrix of the DDP recursion from becoming **ill-conditioned** with a small ρ. Thus, the local convergence of the AL method can be guaranteed in our problem formulation (1), where $g(x)$ is a simple box constraint and $h(x)$ does not exist. □

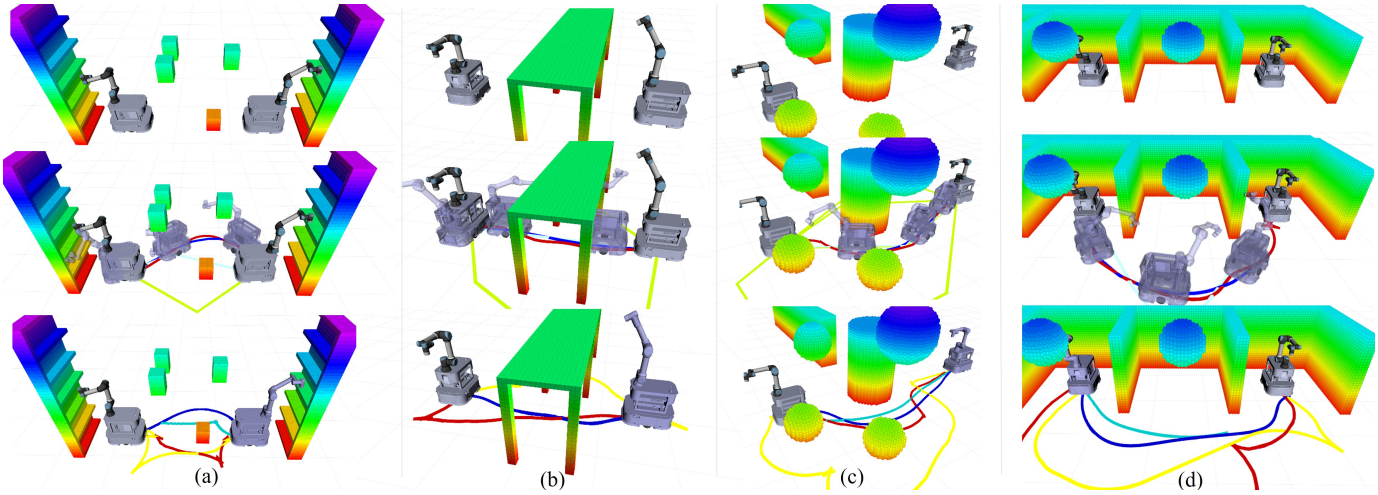

Fig. 5. Four static scenarios (a-d) with the planned results by different whole-body motion planning methods. The sub-figures in the top row show the environments with start-goal pairs of the MM. Those in the mid row show the found guiding topological paths (yellow), front-end (red), and optimized kinodynamic trajectories (blue) results by our method. Those in the bottom row show the comparison results of the MB by RS-RRT (yellow), SST (red), perceptive MPC (cyan), and our method (blue).

V. Experiments

In this section, we conduct extensive simulation and physical experiments to verify the effectiveness and efficiency of our whole-body IPC framework for a MM with a differential base.

A. Experiment Setup

The algorithms written in C++ run in the ROS Melodic open-source platform on a NUC with an Intel i7-8250 CPU, 32G RAM, and NVIDIA GTX 1080 graphics card. Our DDP optimization framework is built upon the OCS2 toolbox [23]. The time horizons of the whole-body trajectory optimization problem are 8s and 4s in simulations and real-world experiments, respectively. We employ the ODE45 integrator with adaptive step sizes for the backward Riccati and forward rollout integration. It is worth noting that the obstacles in the environment are completely unknown in prior for MM and reconstructed in real time with on-board cameras or simulated lidar with 360^circ horizontal fields of view (FOVs). The IPC parameters are listed in table I.

TABLE I
IPC Parameters

λ_1	λ_2	λ_3	λ_4	s_d	g_b	w_t	μ	R	Q
1	0.1	0.1	6	0.1	0.05	1	0.5	diag(0.5)	diag(10)

B. Benchmark for Kinodynamic Searching Methods

We first build four common scenarios including cabinets, tables, walls, floating/ground obstacles as shown in Fig. 5. The SST [24] that samples in the control space, ReedShepp-RRT (RS-RRT)[21] that samples in the state space, and a whole-body optimization method, perceptive MPC [12], are chosen as baselines. For the state sampling method, it is difficult to linearly connect two states due to the non-holonomic constraint of the differential MB. Thus, RS-RRT uses the ReedShepp curves to interpolate and connect the states. The RS-RRT and SST are implemented using the OMPL [25] with fine-tuned parameters. We randomly chose 100 start-goal pairs and compared the four methods for collision avoidance in the environments. We report their statistic results about the trajectory lengths for the arm and MB, goal errors of the MB, total planning time, and success rate in table II. Some planned trajectories are also depicted for illustration in Fig. 5 and the supplementary video. The planning time budget is 60s for each method, and it fails if the corresponding planned whole-body trajectory collides with any obstacle.

It can be observed from table II that traditional RRT often takes a longer time to find poor-quality solutions, even though it has a high success rate. Random sampling in the high-dimensional state space makes the back-end optimization usually suffer from a misleading initial value. In addition, SST method that samples in the control space is several orders of magnitude slower than ours to find solutions with small terminal errors for the MM. Therefore, they can not achieve a real-time planning in dynamic environments. The whole-body terminal errors generated by the perceptive MPC approach are large because only tracking the end-effector path cannot guarantee the accuracy of MB's terminal state. For example, the MB is not back into the parking spot in the bottom-right subfigure (cyan curve in the scenario d). Furthermore, the method always fails in the scenario b where the end-effector tends to track infeasible paths that cross over the desk due to the unreasonable path. In contrast, our hierarchical whole-body kinodynamic trajectory searching method decomposes a high-dimensional searching problem into a series of smaller ones and utilizes different topological paths to make good tradeoffs between success rate and efficiency. Our method can efficiently find high-quality initial solutions for MM in complex environments by taking both spatial and temporal constraints into account.

TABLE II
THE SUCCESS RATES, TRAJECTORY LENGTHS OF THE MB AND ARM, TOTAL COMPUTATION TIMES, AND TERMINAL STATE ERRORS FOR ST-RRT, SST, PERCEPTIVE MPC (P-MPC), AND OUR KINODYNAMIC WHOLE-BODY PATH SEARCHING METHOD IN FOUR TYPICAL SCENARIOS

Methods	scene	Success (%)	$L_b(m)$	$L_a(rad)$	Time (ms)	error (m)	scene	Success (%)	$L_b(m)$	$L_a(rad)$	Time (ms)	error (m)
RS-RRT	a	**100**	6.13	7.22	324	**0**	c	**100**	11.25	11.42	443.7	0
SST [24]		28	4.27	10.33	41583	0.34		86	8.33	9.38	18660	0.055
P-MPC [12]		41	**3.08**	8.22	84.69	0.86		69	**5.86**	**4.18**	73.68	0.557
Ours		**100**	3.31	**5.81**	**33.94**	**0**		**100**	6.16	4.65	**37.39**	**0**
RS-RRT	b	**100**	10.31	11.18	979.9	0	d	**100**	10.73	9.35	339.5	**0**
SST [24]		92	7.78	17.26	37528	0.081		56	11.92	13.77	43569	0.22
P-MPC [12]		0	-	-	-	-		65	**5.34**	**4.86**	73.48	1.25
Ours		**100**	**3.6**	**9.19**	**73**	**0**		**100**	6.26	6.29	**45.25**	**0**

C. Motion Planning in Dynamic Cluttered Environments

Before we verify the collision avoidance capability of our whole-body IPC framework in cluttered environments, we compare the tracking accuracy of different controllers. A PID controller ($P = 1$, $D = 0.5$, $I = 0.05$), time-varying Linear quadratic regulator (TVLQR) [22], open loop MPC controller, and our optimal feedback-feedforward controller are compared for tracking a planned whole-body trajectory under different gaussian noises $\mathcal{N} \sim (\mu_n, \Sigma_n)$. Note that conventional frameworks usually solve the trajectory optimization with off-the-shelf solvers and get an open-loop control policy and trajectory. Extra effort in designing and tuning a controller is required to track the trajectory. On the contrary, our AL-DDP solver can obtain an optimal feedback-feedforward controller to track the optimized whole-body trajectory from DDP iteration at no cost. The statistic results are listed in table III. e_b and e_a are the mean tracking errors for the MB and arm, respectively, and the values in parentheses represent the maximum ones. Compared with TVLQR, our controller achieves a higher tracking accuracy under all the disturbances since the integrated planning and control pattern has fewer modules and less error accumulation. Furthermore, the parameters of the individually designed TVLQR controller cannot fit as well as our integrated controller. Even though we tune the parameters of the PID controller carefully, it can only track the trajectory under a small noise and is particularly prone to oscillations at the trajectory transitions due to the high non-linearity of the differential MB. Since the open Loop controller policy is not robust against disturbances, the error increases rapidly under a biased noise. Although we can also address this problem through a receding horizon control strategy, it is much more computationally consuming since a high-frequency solution for optimal control is required. On the contrary, ours only requires a low-frequency solution because the F-F controller gained from DDP allows high-frequency low-level control without extra optimization burden. For the comparative result to existing off-the-shelf solvers, it can be seen from the supplementary materials that both solution speed and the quality achieve SOTA performance.

After obtaining a high-performance controller, we conduct the whole-body collision avoidance experiments in diverse dynamic cluttered environments as shown in Fig. 6 and the supplementary video. The goals are set to 40 meters away. The average speeds of the obstacle in three environments are 0 m/s, 0.5 m/s, and 1 m/s. The MM perceives the obstacles, constructs the OGM and ESDF, and finds a trajectory using only on-robot lidar with 360° FOV in real-time. Results in the supplementary video show that the MM can run at an average speed of 1.86 m/s in environments containing dynamic obstacles at a maximum speed of 2 m/s. Even though the tracks of dynamic obstacles are not predicted in this paper, our planning method is fast enough to quickly replan a feasible whole-body trajectory for MM in dynamic environments. Furthermore, all planned trajectories are parameterized in time, naturally enabling it to handle dynamic cases. In the mean obstacle speed of 1m/s scenes, the overall success rate is about 75%. A common failure case is that quickly moving obstacles are out of the FOV but suddenly rush towards the MM. It can be seen from the supplementary video that the MM is agile to perform dynamic obstacle avoidance according to the obstacle gradients retrieved from a high-frequency updating ESDF map.

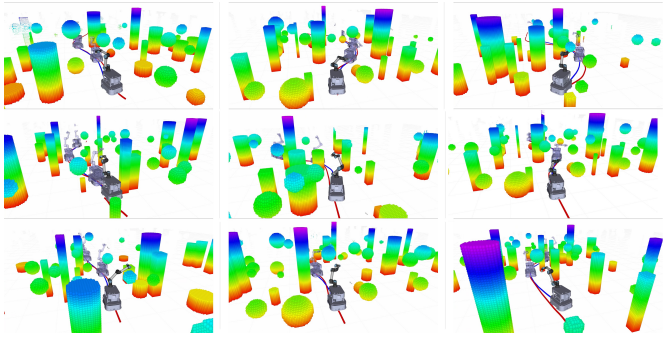

Fig. 6. Motion planning results for the mobile manipulator in dynamic cluttered environments

The elapsed times of different modules in our planning pipeline are also reported in Fig. 7. The running time of ST-RRT increases as the mean speeds of obstacles increase. The more cluttered environments are, the more difficult a manipulator can find a feasible trajectory in the temporal-spatial state space in the context of a determined base trajectory. Our topo-guided kinodynamic searching method reduces the probability of exploring a lot of infeasible space for the MB because of the guidance of the topological paths, leading to an almost constant elapsed time as the environments become denser. The total time of one AL-DDP iteration is about 30ms regardless of the speeds of obstacles, ensuring a real-time dynamic collision avoidance capability.

D. Pick-and-Place Task in an Unstructured Environment

We deploy our whole-body IPC framework for a real-world pick-and-place task to verify its efficiency. Our hardware MM platform consists of a differential MB and a UR5 manipula-

TABLE III
THE MEAN AND MAXIMUM (REPORTED INSIDE PARENTHESIS) TRACKING ERRORS FOR THE MB AND ARM BY DIFFERENT CONTROLLERS UNDER GAUSSIAN DISTURBANCES

Gaussian Noises	$\mu_n = 0.0$ $\Sigma_n = 0.0$		$\mu_n = 0.0$ $\Sigma_n = 0.1$		$\mu_n = 0.03$ $\Sigma_n = 0.1$		$\mu_n = 0.1$ $\Sigma_n = 0.1$	
	$e_b(m)$	$e_a(rad)$	$e_b(m)$	$e_a(rad)$	$e_b(m)$	$e_a(rad)$	$e_b(m)$	$e_a(rad)$
PID	0.294 (0.523)	0.093 (0.241)	0.322 (0.576)	0.119 (0.257)	0.474 (1.113)	0.078 (0.140)	1.051 (2.592)	0.224 (0.422)
TVLQR	0.050 (0.108)	0.021 (0.051)	0.047 (0.131)	0.012 (0.055)	0.067 (0.181)	0.066 (0.091)	0.220 (0.559)	0.162 (0.237)
Open MPC	0.111 (0.475)	0.036 (0.133)	0.132 (0.434)	0.093 (0.244)	1.297 (3.053)	0.754 (1.620)	4.161 (9.664)	2.369 (5.411)
Ours	**0.015 (0.032)**	**0.005 (0.019)**	**0.018 (0.040)**	**0.008 (0.0189)**	**0.025 (0.068)**	**0.038 (0.053)**	**0.118 (0.293)**	**0.101 (0.139)**

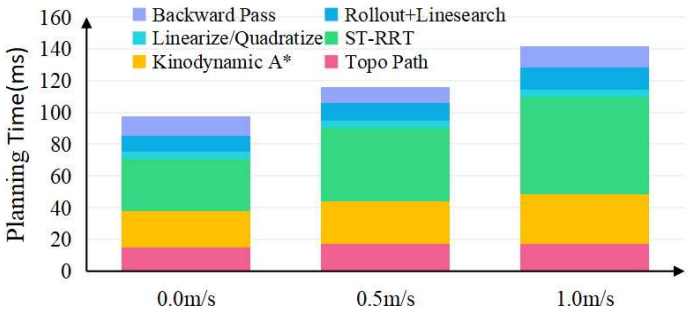

Fig. 7. Running time of different modules of our framework.

tor. Two Intel Realsense D455 cameras with limited FOVs are mounted at the back end of the base, and one Intel Realsense L515 is mounted as eye-in-hand to run a visual servo algorithm. The obstacle 3-D point clouds are generated by the camera driver using stereo vision. The MM picks an object in region A and places it in regions B and C, while avoiding collision with on-robot cameras, as shown in Fig. 8. We formulate an optimization problem with velocity and acceleration constraints in the cluttered environment and solve it with our AL-DDP.

It can be observed from Fig. 8, 9 and our supplementary videos that the MM can incrementally build the OGM and ESDF around the body using the obstacle point clouds in an unknown environment. Our framework can find smooth and short whole-body trajectories to reach the target region and grasp the object with a success rate of about 70% in 50 times attempts. The average speed of the MB and the arm is 0.42 m/s and 0.29 rad/s. Besides, once the environment layout changes, the ESDF map updates immediately and a collision-free trajectory is found in real-time, which means the MM can quickly plan and accurately execute the high-quality trajectories under our IPC framework. It's worth noting that since the MM needs to approach and grasp the objects instead of avoiding them, we filtered out the objects' pointcloud in the map, which sometimes caused the arm to knock down the objects. Therefore, an effective visual servo is still necessary for robust grasping.

It should be noticed that our IPC framework can quickly find collision-free trajectories in dynamic cluttered environments as long as sufficient perception information is fed, which has been demonstrated in Section V-C. Unfortunately, it is hard to completely capture the cluttered and dynamic obstacles surrounding the MM by the limited FOVs of on-board cameras in reality. In addition, we should constrain the robot arm's motions to avoid occlusion. If so, the solution subspace is dramatically compressed. Therefore, we only conduct physical experiments in static cluttered environments to verify our framework after we have attained the expected and remarkable results in dynamic cluttered environments in Section V-C.

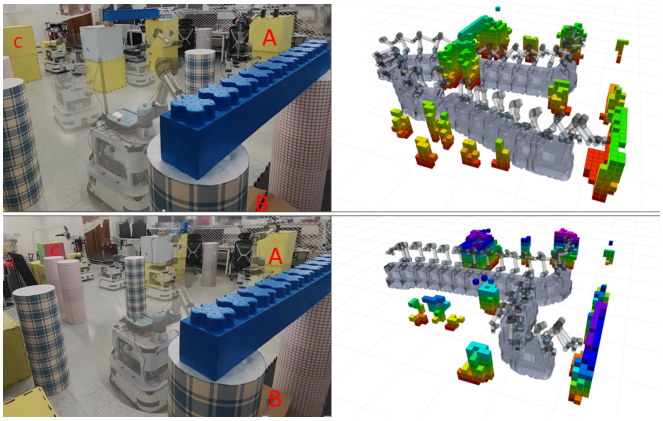

Fig. 8. Physical experiment setup and snapshots of two pick-and-place tasks. The MM picks up two objects in region A and places one in B and the other in C while avoiding collision. The real-time and near-optimal planned trajectories are smooth and short.

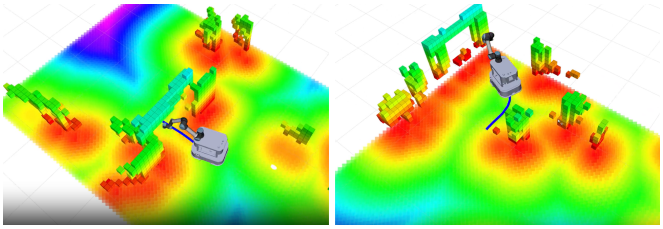

Fig. 9. Real-time constructed OGM and ESDF through only local perception. Red color means a small distance to the obstacle, while blue means a large distance.

VI. CONCLUSIONS

In this paper, we propose RAMPAGE, a complete framework including perception and integrated planning and control (IPC) modules, to generate whole-body and agile trajectories for the mobile manipulator in dynamic cluttered environments in real-time. Our hierarchical topo-guided kinodynamic trajectory searching method associated with Spatial-Temporal RRT can guarantee an initial whole-body trajectory is found in about 70ms. We solve the IPC optimization problem using AL-DDP instead of other off-the-shelf NLP solvers to quickly obtain an optimal feedforward-feedback controller that can accurately track trajectories at a high frequency. Extensive simulated and physical experiments demonstrate that our proposed framework achieves efficient planning and robust control simultaneously, which boosts the whole-body collision avoidance performance for the mobile manipulator in dynamic cluttered environments significantly. In future works, utilizing deep learning technologies to predict the motions of obstacles and plan under uncertainty [26] is promising

to further improve our collision avoidance performance in practice.

References

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

[2] S. Chitta, I. Sucan, and S. Cousins, "Moveit!" *IEEE Robotics & Automation Magazine*, vol. 19, no. 1, pp. 18–19, 2012.

[3] S. S. Srinivasa, D. Ferguson, C. J. Helfrich, G. Hollinger, J. Kuffner, and M. V. Weghe, "HERB: a home exploring robotic butler," *Autonomous Robots*, vol. 28, pp. 5–20, 2010.

[4] Y. Yang, W. Merkt, V. Ivan, and S. Vijayakumar, "Planning in time-configuration space for efficient pick-and-place in non-static environments with temporal constraints," in *2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids)*, pp. 1–9. IEEE, 2018.

[5] F. Meng, L. Chen, H. Ma, J. Wang, and M. Q.-H. Meng, "NR-RRT: Neural risk-aware near-optimal path planning in uncertain nonconvex environments," *IEEE Transactions on Automation Science and Engineering*, 2022.

[6] V. Pilania and K. Gupta, "Mobile manipulator planning under uncertainty in unknown environments," *The International Journal of Robotics Research*, vol. 37, no. 2-3, pp. 316–339, 2018.

[7] P. Stibinger and G. Broughton, "Mobile manipulator for autonomous localization, grasping and placement of construction material in a semi-structured environment," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 2595–2602, 2021.

[8] 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.

[9] D. Dolgov, S. Thrun, M. Montemerlo, and J. Diebel, "Practical search techniques in path planning for autonomous driving," *Ann Arbor*, vol. 1001, no. 48105, pp. 18–80, 2008.

[10] N. Ratliff, M. Zucker, J. A. Bagnell, and S. Srinivasa, "Chomp: Gradient optimization techniques for efficient motion planning," in *2009 IEEE international conference on robotics and automation*, pp. 489–494. IEEE, 2009.

[11] J. Schulman, Y. Duan, J. Ho, A. Lee, I. Awwal, H. Bradlow, J. Pan, S. Patil, K. Goldberg, and P. Abbeel, "Motion planning with sequential convex optimization and convex collision checking," *The International Journal of Robotics Research*, vol. 33, no. 9, pp. 1251–1270, 2014.

[12] 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.

[13] B. Burgess-Limerick, C. Lehnert, J. Leitner, and P. Corke, "An architecture for reactive mobile manipulation on-the-move," *arXiv preprint arXiv:2212.06991*, 2022.

[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] A. Heins and A. P. Schoellig, "Keep it upright: Model predictive control for nonprehensile object transportation with obstacle avoidance on a mobile manipulator," *arXiv preprint arXiv:2305.17484*, 2023.

[16] 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)*, pp. 6711–6717, 2020.

[17] 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)*, pp. 12 759–12 765. IEEE, 2021.

[18] S. Thrun, "Probabilistic robotics," *Communications of the ACM*, vol. 45, no. 3, pp. 52–57, 2002.

[19] P. Felzenszwalb and D. Huttenlocher, "Distance transforms of sampled functions," *Theory of Computing*, vol. 8, DOI 10.4086/toc.2012.v008a019, 08 2004.

[20] B. Zhou, J. Pan, F. Gao, and S. Shen, "RAPTOR: Robust and perception-aware trajectory replanning for quadrotor fast flight," *IEEE Transactions on Robotics*, vol. 37, DOI 10.1109/TRO.2021.3071527, no. 6, pp. 1992–2009, 2021.

[21] J. Reeds and L. Shepp, "Optimal paths for a car that goes both forwards and backwards," *Pacific Journal of Mathematics*, vol. 145, no. 2, pp. 367–393, 1990.

[22] C. Samson, "Time-varying feedback stabilization of car-like wheeled mobile robots," *The International Journal of Robotics Research*, vol. 12, no. 1, pp. 55–64, 1993.

[23] F. Farshidian *et al.*, "OCS2: An open source library for optimal control of switched systems."

[24] Y. Li, Z. Littlefield, and K. E. Bekris, "Asymptotically optimal sampling-based kinodynamic planning," *The International Journal of Robotics Research*, vol. 35, no. 5, pp. 528–564, 2016.

[25] I. A. Şucan, M. Moll, and L. E. Kavraki, "The Open Motion Planning Library," *IEEE Robotics & Automation Magazine*, vol. 19, no. 4, pp. 72–82, Dec. 2012.

[26] F. Meng, L. Chen, H. Ma, J. Wang, and M. Q.-H. Meng, "Learning-based risk-bounded path planning under environmental uncertainty," *IEEE Transactions on Automation Science and Engineering*, 2023.

Yuqiang Yang received the B.Eng. degree in automation control from the South China University of Technology, Guangzhou, China in 2022. He is currently working toward the master's degree in South China University of Technology, Guangzhou, China.

His research interests include motion planning and control for robots.

Fei Meng received the B.Eng. degree in electrical engineering and automation and the M.Eng. degree in control engineering from Harbin Institute of Technology, Weihai and Harbin, China, in 2016 and 2019, respectively. He is currently pursuing the Ph.D. degree with the Department of Electronic Engineering, The Chinese University of Hong Kong, Hong Kong, SAR, China.

His research interests include learning-based motion planning and control.

Zehui Meng received the Bachelor's degree of Mechanical Eigineering from National University of Singapore in 2014 and received Ph.D from National University of Singapore in 2018. He's currently a senior research engineer at Huawei co.Ltd.

His research interests include perception, motion planning and control for robots.

Chenguang Yang (Fellow, IEEE) received the B.Eng. degree in measurement and control from Northwestern Polytechnical University, Xian, China, in 2005, and the Ph.D. degree in control engineering from the National University of Singapore, Singapore, in 2010. He performed postdoctoral studies in human robotics at the Imperial College London, London, U.K from 2009 to 2010. He is Chair in Robotics with Department of Computer Science, University of Liverpool, UK. He was awarded UK EPSRC UKRI Innovation Fellowship and individual EU Marie Curie International Incoming Fellowship. As the lead author, he won the IEEE Transactions on Robotics Best Paper Award (2012) and IEEE Transactions on Neural Networks and Learning Systems Outstanding Paper Award (2022). His research interest lies in human robot interaction and intelligent system design.