Hierarchical Motion Planning for a Barista Mobile Manipulator Preliminary Report

Francesco Marani (), Javier Gil (6563953), Núria Seguí (6299946), Simone Utili (6568688)

I. INTRODUCTION

This project addresses the motion planning problem for a **mobile manipulator** acting as an autonomous barista assistant. The robot must bring a drink from behind the bar and deliver it to a designated table while avoiding both static obstacles (e.g., tables, chairs or walls) and dynamic obstacles (e.g., walking customers). The robot must ensure smooth motion to prevent spilling the drink, so it is very important to generate a safe and efficient trajectory.

We adopt a **hierarchical motion planning framework** combining a global sampling-based planner (PRM/PRM*) with a local Model Predictive Controller (MPC). The global planner computes a feasible collision-free path in the static environment, while the local MPC refines the motion, adapts to dynamic obstacles, and ensures smooth and stable trajectory execution.

Why this approach? A hierarchical planning structure is well suited for mobile manipulation in semi-structured environments such as restaurants. PRM planners efficiently compute global paths in environments with predominantly static obstacles, while MPC enables reactive, constraint-aware motion in real time, crucial when navigating among humans.

To ensure robust development, we follow an **incremental complexity approach** explained in section V-A.

II. KINEMATIC MODEL

We use the **Albert mobile manipulator**, which consists of a Clearpath Boxer differential-drive mobile base and a 7-DOF Franka Emika Panda arm. This platform enables both 2D navigation and precise drink manipulation.

A. Mobile Base Kinematics

The Clearpath Boxer is modeled as a differential-drive robot with configuration $\mathbf{q}_b = [x, y, \theta]^T \in \mathbb{R}^3$. The nonholonomic kinematic model is:

$$\dot{x} = v\cos(\theta), \quad \dot{y} = v\sin(\theta), \quad \dot{\theta} = \omega, \tag{1}$$

where v and ω represent linear and angular velocities.

To relate these velocities to the left and right wheel speeds, they can be computed as:

$$v = \frac{R}{2}(\dot{\phi}_R + \dot{\phi}_L), \qquad \omega = \frac{R}{2L}(\dot{\phi}_R - \dot{\phi}_L),$$

where R is the wheel radius, $2L$ is the track width, and $\dot{\phi}_{R,i}, \dot{\phi}_{L,i}$ are the angular velocities of the right and left wheels, respectively.

B. Manipulator Kinematics

The 7-DOF Panda arm configuration is $\mathbf{q}_a \in \mathbb{R}^7$ with forward kinematics:

$$\mathbf{X}_b^e = f_{k,arm}(\mathbf{q}_a) \tag{2}$$

and Jacobian $\mathbf{J}_a(\mathbf{q}_a) \in \mathbb{R}^{6\times 7}$, relating joint velocities to end-effector twist in the base frame.

C. Combined System

The complete configuration space is $\mathcal{C}_i = \mathbb{R}^2 \times \mathbb{S}^1 \times (\mathbb{S}^1)^7$. The combined configuration is:

$$\mathbf{q}_i = [\mathbf{q}_{b,i}^T, \mathbf{q}_{a,i}^T]^T \in \mathbb{R}^{10} \tag{3}$$

The forward kinematics mapping to world frame is:

$$\mathbf{x}_{e,i}^w = \mathbf{X}_w^{b,i}(\mathbf{q}_{b,i}) \circ \mathbf{X}_{b,i}^{e,i}(\mathbf{q}_{a,i}) = f_k(\mathbf{q}_i) \tag{4}$$

where $\mathbf{x}_{e,i}^w \in SE(3)$ is the end-effector pose in world frame.

The differential kinematics is:

$$\dot{\mathbf{x}}_{e,i} = \mathbf{J}_i(\mathbf{q}_i)\dot{\mathbf{q}}_i \tag{5}$$

where the complete geometric Jacobian is:

$$\mathbf{J}_i(\mathbf{q}_i) = [\mathbf{J}_{b,i}(\mathbf{q}_i) \quad \mathbf{J}_{a,i}(\mathbf{q}_{a,i})] \in \mathbb{R}^{6\times 10} \tag{6}$$

Note that due to the nonholonomic constraint of the differential-drive base, the instantaneous motion is restricted, but the system remains controllable and can reach any configuration through appropriate maneuvers. The system has 10 DOF configuration space but only 9 instantaneous velocity DOF (2 base velocities + 7 arm velocities), while still being redundant for the 6-DOF end-effector task.

III. CHOSEN PLANNER

A. Global Planner: PRM / PRM*

The robot operates in a restaurant-like environment where static obstacles (tables, chairs, bar counters) rarely change. Thus, we adopt a **Probabilistic Roadmap (PRM or PRM*)** as the global planner.

Why PRM?

- Multi-query planner: once the roadmap is built, many table-delivery queries can be solved quickly.
- Static environment: PRM performs best when the workspace does not frequently change.
- Scalability: efficient for 2D mobile-base planning even in cluttered spaces.
- Optimal variant (PRM*): converges asymptotically to an optimal global path, beneficial for minimizing drink disturbance.

The PRM computes a collision-free reference path for the mobile base, which is subsequently tracked by the local controller.

B. Local Planner: Model Predictive Control (MPC)

MPC refines the motion around the global path and reacts to dynamic obstacles such as moving humans.

Why MPC?

- Predictive behavior for smooth, spill-free trajectories.
- Handles constraints: nonholonomic base, joint limits, obstacle avoidance.
- Allows integration of dynamic obstacles through online cost shaping.

The MPC outputs safe velocity commands for the base and arm, ensuring collision avoidance and stable drink transport.

IV. PLANNED SIMULATION ENVIRONMENT

A. Simulator

We use **PyBullet** with a custom OpenAI Gym environment, providing:

- Realistic rigid body dynamics
- Efficient collision detection
- URDF support for mobile manipulators
- Fast simulation suited for real-time MPC testing

Initial consideration of NVIDIA Isaac Sim was abandoned due to high computational requirements not all team members can satisfy.

B. Environment Setup

Restaurant-style scenario:

- **Task:** pick up a drink from behind the bar, navigate the restaurant, deliver it to the correct table.
- **Static obstacles:** tables, chairs, walls, counters (modeled as convex shapes for computational efficiency).
- **Dynamic obstacles:** customers walking around the dining area.
- **Payload:** a drink that must be transported smoothly (orientation and vibration constraints).

V. PLANNED SCENARIOS AND METRICS

A. Evaluating Scenarios

Following incremental complexity:

1) Phase 1: Mobile base only in an uncluttered environment (validate PRM path planning).
2) Phase 2: Mobile base + arm manipulation tasks behind the bar.
3) Phase 3: Drink transport in static restaurant layout.
4) Phase 4: Full scenario with dynamic humans and tight spaces.

B. Performance Metrics

- **Success rate:** Percentage of completed tasks without collisions, constraint violations or spilled drinks.
- **Planning time:** Global planner computation time (PRM/PRM* convergence)
- **Execution time:** Total task completion time
- **Tracking smoothness:** acceleration/jerk norms
- **Dynamic obstacles avoidance**: number of collisions or times the robot violates a safe distance threshold.

VI. OPEN QUESTIONS

We seek feedback on:

1) Is PRM/PRM* sufficient for the global planner in a semi-static restaurant environment?
2) For the MPC implementation: should we prioritize linear MPC with linearized dynamics (faster, simpler) or nonlinear MPC (more accurate) given the project scope?
3) How strict should drink-spill constraints be modeled in the cost function?
4) Should dynamic obstacle handling be considered a core feature or an optional extension?
5) Is modeling obstacles as convex shapes acceptable for computational efficiency, or should we implement more complex collision geometries from the start?
6) Are the chosen metrics sufficient for evaluating the planner's performance, or should we include additional measures (e.g., energy consumption)?

REFERENCES

[1] S. M. LaValle, *Planning Algorithms*, Cambridge University Press. (2006).

[2] H. Chen, X. Zang, Y. Liu, X. Zhang, J. Zhao, *A hierarchical motion planning method for mobile manipulator*. Sensors, 23(15), 6952. (2023).

[3] M. Spahn, B. Brito, J. Alonso-Mora, *Coupled Mobile Manipulation via Trajectory Optimization with Free Space Decomposition*. In 2021 IEEE International Conference on Robotics and Automation (ICRA), pp. 12759–12765. (2021).