Trajectory Planning and Control for Cooperative Manipulators in Constrained Environments

Mayank Sewlia, Christos K. Verginis, and Dimos V. Dimarogonas

I. Introduction and Problem Formulation

The cooperative manipulation problem involves controlling multiple robots to jointly manipulate a shared object typically one that cannot be handled by a single robot alone. This collaboration offers several advantages, including load sharing and the ability to manipulate heavier or more complex payloads. However, it also introduces challenges such as coordinating motion between agents, distributing forces evenly, performing decentralised planning,and translating high-level task specifications for the object into low-level control actions at each robot's joints. Cooperative manipulation systems find applications in human-robot interaction, aerial payload transport, manipulation in cluttered environments, and industrial automation [1].

To specify tasks on the manipulated object, we employ Signal Temporal Logic (STL), a continuous time and space formalism which specifies both temporal and spatial constraints on the manipulated object. In the following, we present the dynamics of the system along with the problem statement.

A. System Dynamics

The system consists of N mobile robotic arms rigidly grasping an object. Let $\{W\}$ be the inertial frame. The mobile base of the ith arm is associated with a frame $\{B_i\}$, and its end-effector is associated with a frame $\{E_i\}$, as shown in Figure 1. The position of the base in the $\{W\}$ frame is denoted by $b_i \in \mathbb{R}^2$, and the position of the end-effector by $p_i \in \mathbb{R}^3$. The dynamics of each mobile manipulator are governed by,

$$M_i(q)\ddot{q}_i + C_i(q,\dot{q})\dot{q}_i + g_i(q) = \tau_i \tag{1}$$

where $q_i = \begin{bmatrix} b_i^\top & \theta_i^\top \end{bmatrix}^\top \in \mathbb{R}^{n_i}$, with $b_i \in \mathbb{R}^2$ denotes the position of the base of the ith arm, and $\theta_i \in \mathbb{R}^{n_i-2}$ representing its joint angles. The relationship between q_i and p_i is governed by the forward kinematics,

$${}^{W}X_i^{E_i} = f(q_i) \tag{2}$$

where

$${}^{W}X_i^{E_i} = \begin{bmatrix} {}^{W}R_i^{E_i} & p_i \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix}$$

represents the homogeneous transformation matrix from the $\{E_i\}$ frame to the $\{W\}$ frame, and $f(q_i)$ is a function that maps the joint configuration q_i to the end-effector position p_i. The agent dynamics (1) are defined in joint space. The coupled object-agent dynamics in task space are given by,

$$\tilde{M}\dot{v}_o + \tilde{C}v_o + \tilde{g} + \tilde{w} = Gu. \tag{3}$$

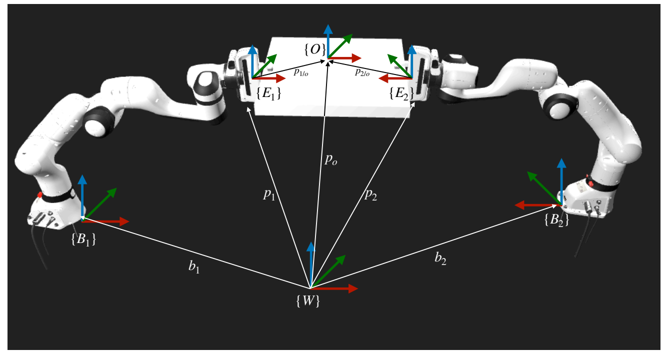


Fig. 1: Visualisation of frames of two panda arms rigidly grasping an object(we consider panda arms developed by Franka Emika [2]

The above equation is obtained by balancing forces exerted by the agents to the forces experienced by the object.

Problem 1: Consider a system of N agents rigidly grasping an object, and let an STL formula φ specify the desired spatio-temporal behaviour of the object state x_o. The objective is to design control torques τ_i for each agent $i = 1, \ldots, N$ in accordance with the dynamics in (1), such that the resulting x_o satisfies φ, i.e., $x_o \models \varphi$.

II. Methodology

The proposed solution approachis illustrated in Figure 2, and is as follows:

- *MAPS^2:* Compute a desired collision-free trajectory for the object x_{o_d} using MAPS^2 (see [3]).
- *Footprint Planner:* The problem of directly computing the joint angles given the desired object trajectory x_{o_d} quickly becomes intractable, as it involves optimising over $N \times n_i$ states, along with non-convexities introduced by obstacles and forward kinematics. Moreover, optimising over a time horizon $[0, t_f]$ with T total time steps results in $T \times N \times n_i$ decision variables. To alleviate this challenge, we decompose the planning problem for mobile manipulators into two parts: planning for the mobile bases and planning for arms.
  Given a desired x_{o_d}, our goal is to compute a collision-free set of base trajectories $b_{des} = \begin{bmatrix} b_{1\text{des}}^\top & \cdots & b_{N\text{des}}^\top \end{bmatrix}$. Key considerations include:
  - The base trajectories should not intersect with one another.
  - The base trajectories should avoid collisions with obstacles.
  - The base velocities should satisfy motion constraints.

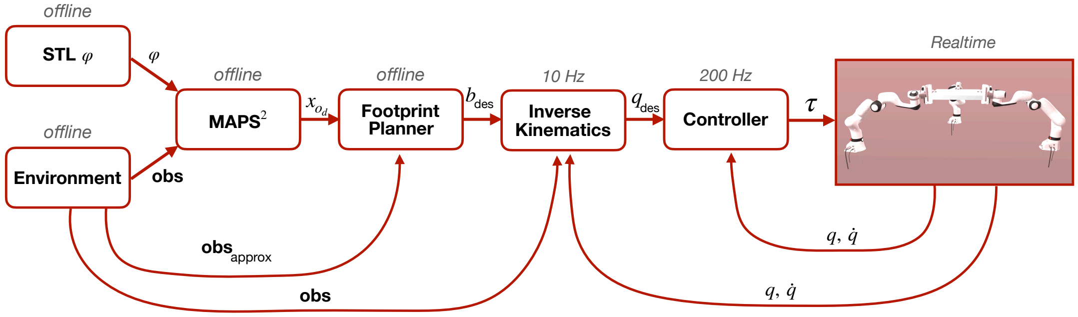


Fig. 2: Proposed solution architecture.

- The base trajectories should be feasible with respect to the desired object trajectory x_{o_d}.
- The base trajectories should remain mutually feasible, i.e., the agents should not drift too far apart from one another releasing the object.

This problem can be formulated as a trajectory optimisation problem, where the objective is to find b_{des} that satisfies the above constraints. Let the trajectory x_{o_d} be discretised into T time steps over the horizon $[0, t_f]$.

$$\min_{b_{des}} \quad \sum_{k=1}^{T} \sum_{i=1}^{N} \sum_{\substack{j=1 \\ i \neq j}}^{N} w_i \left(\|b_{i\text{des}}[k] - b_{j\text{des}}[k]\|^2 - \alpha_i^2 \right)^2 \tag{4a}$$

$$\text{s.t.} \quad \left\| \frac{1}{N} \sum_{i=1}^{N} b_{i\text{des}}[k] - x_{o_d}(t) \right\|^2 \leq \epsilon \tag{4b}$$

$$-\eta \leq b_{i\text{des}}[k+1] - b_{i\text{des}}[k] \leq \eta, \quad \forall\, i = 1, \ldots, N \tag{4c}$$

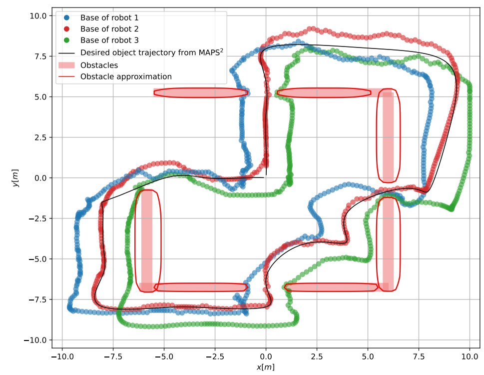
$$b_{\text{des}}[k] \in \mathcal{W} \setminus \mathcal{O} \tag{4d}$$

$$b_{\text{des}}[0] = x_{o_d}(0) \tag{4e}$$

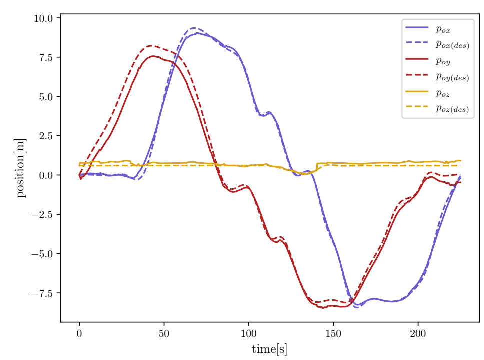
$$b_{\text{des}}[t_f] = x_{o_d}(t_f) \tag{4f}$$

- 4a defines the cost function that penalises the deviation of the squared distance between any two bases from the target squared distance α_i^2. If the arms grasp the object in non-homogeneous configurations, then α_i will vary depending on the kinematic reach from each base to its end-effector. The weights w_i are positive scalars that can be tuned based on the desired strictness in maintaining the target inter-base distances.
- 4b ensures that the centroid of the base positions remains close to the desired object trajectory x_{o_d}. The object trajectory here considers only the x and y coordinates, as z is fixed for the bases. This constraint maintains the proximity of the centroid formed by all base positions to x_{o_d} generated by the MAPS^2 algorithm, which already provides a global trajectory that avoids local minima—an advantage we aim to transfer to the base trajectories.

• *InverseKinematics:* We solve an optimisation problem to determine the desired joint angles based on the obstacles and the footprint trajectory from above.
• *Control Design:* The control torques τ tracks the desired joint angles generated by the inverse kinematics algorithm.



(a) At $t = 105s$



(a) At $t = 205s$

A. Simulations

The solution to (4) is shown in Figure 3a and the tracking error is shown in Figure 4a.

References

[1] T. G. Sugar and V. Kumar, "Control of cooperating mobile manipulators," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 1, pp. 94–103, 2002.
[2] S. Haddadin, "The franka emika robot: A standard platform in robotics research," *IEEE Robotics & Automation Magazine*, 2024.
[3] M. Sewlia, C. K. Verginis, and D. V. Dimarogonas, "Maps ^2: Multi-robot autonomous motion planning under signal temporal logic specifications," *arXiv preprint arXiv:2309.05632*, 2023.