

# Trajectory Tracking for Multi-Manipulator Systems in Constrained Environments\*

Mayank Sewlia<sup>a</sup>, Christos K. Verginis<sup>b</sup>, Dimos V. Dimarogonas<sup>a</sup>

<sup>a</sup>*Division of Decision and Control, Department of Electrical Engineering, KTH Royal Institute of Technology, Stockholm, Sweden*

<sup>b</sup>*Signals and Systems, Department of Electrical Engineering, Uppsala University, Uppsala, Sweden*

---

## Abstract

We consider the motion planning problem for a mobile multi-manipulator system grasping an object in an obstacle cluttered and highly constrained environment. The task is to move the object while satisfying spatio-temporal requirements expressed using Signal Temporal Logic (STL). Beyond avoiding static obstacles, the system must also navigate through narrow passages that induce discrete geometric “modes” requiring coordinated reconfiguration of all the manipulators, all while satisfying the task imposed on the object. This coupling of discrete transitions and continuous motions naturally gives rise to a hybrid systems problem. Our approach computes collision-free trajectories for both the grasped object and the mobile bases, followed by online inverse kinematics and a feedback controller to track these trajectories in real time. The resulting pipeline manages the hybrid interaction between high-level task specifications, discrete mode changes caused by spatial constraints, and the continuous dynamics of the mobile manipulators. We demonstrate and validate our method in high-fidelity physics simulations using three Franka Emika Panda mobile manipulators rigidly grasping an object.

---

## 1. Introduction

A multi-manipulator system consists of multiple robotic arms sharing the load of carrying an object and transporting it to a desired location. When each robotic arm is mounted on a mobile base, the system is referred to as a mobile multi-manipulator system. Such systems are applicable in warehouse automation, automated construction, and search-and-rescue operations, where robots can autonomously navigate while carrying heavy payloads [1], [2], [3]. These applications naturally involve hybrid interactions: the environment introduces discrete regions (free space, passages, tight corners), while the robots continuously try to adjust their configurations to avoid obstacles and maintain task requirements.

The underlying task demands precise coordination among all mobile manipulators so as to avoid inducing unnecessary internal forces on the object, prevent self-collisions and inter-robot collisions, and ensure that the object satisfies its assigned spatio-temporal specification. The challenge becomes even greater in the presence of obstacles and narrow passages, which introduce discrete mode changes. For instance, if a passage is narrower than the convex hull of the combined system, the robots must transition into a different geometric configuration, reorienting and repositioning themselves without compromising the continuity of the object trajectory. This interplay between discrete structural changes in the environment and continuous control of the robots forms a natural hybrid systems setting.

This paper studies the problem of finding trajectories for all joints of the mobile multi-manipulator system such that the grasped object satisfies the STL defined task while the entire system avoids obstacles. Unlike previous works that focus primarily on moving around obstacles, we explicitly consider the problem of traversing narrow passages. These passages require coordinated reconfiguration of the entire manipulator system while still tracking the trajectory for the grasped object. Our framework can be run in real time, enabling coordinated motion planning and control across multiple mobile manipulators while satisfying STL constraints on the grasped object. It consists of both offline and online components. In the offline phase, we construct a map of the environment and compute a collision-free trajectory for the object using our recently developed MAPS<sup>2</sup> algorithm [4]. MAPS<sup>2</sup> algorithm generates trajectories

---

\*This work was supported by the ERC CoG LEAFHOUND, the Swedish Research Council (VR) and the Knut och Alice Wallenberg Foundation (KAW).

*Email addresses:* [sewlia@kth.se](mailto:sewlia@kth.se) (Mayank Sewlia), [christos.verginis@angstrom.uu.se](mailto:christos.verginis@angstrom.uu.se) (Christos K. Verginis), [dimos@kth.se](mailto:dimos@kth.se) (Dimos V. Dimarogonas)

for multi-robot systems satisfying STL specifications. Next, we generate collision free trajectories for all mobile bases using a trajectory optimisation method. The online phase is divided into a slower, computationally intensive module and a faster feedback module. The slower module computes the joint angles of all manipulators given the object pose and base positions, while the faster module employs a feedback controller to track the desired joint angles. These modules still run online but at different frequencies allowing for realtime implementation. We demonstrate our approach in high-fidelity, physics-based simulations using three Panda mobile manipulators. These simulations highlight the ability of the framework to maintain safety, avoid collisions, and satisfy spatio-temporal specifications while traversing narrow passages.

### 1.1. Related Work

A variety of planning algorithms for mobile manipulators have been studied in the literature, with an excellent overview provided in [5]. Broadly, motion planning for mobile manipulator systems can be categorised into three approaches: grid-based, sampling-based, and optimisation-based planners.

Grid-based planners, such as [6], [7], and [8], discretise the reachable space of the mobile manipulator and employ search-based methods to generate feasible trajectories as sequences of discretised regions. Other works, such as [9], construct inverse reachability maps to determine the joint configurations of the base and arms for a given end-effector location.

Sampling-based planners, on the other hand, rely on heuristics and approximations of the collision-free space. For example, [10] proposes an asymptotically optimal manipulation planner by extending sampling-based roadmaps to the joint configuration space of both the manipulator and the object. Similarly, [11] extends RRT to prehensile tasks, while [12] introduces a bi-directional informed RRT\* in which hyper-ellipsoidal regions are constructed for subsets of the configuration space of the mobile base and manipulators to iteratively refine trajectories. While sampling-based methods are highly effective in moderate dimensions (up to approximately 10), they scale poorly in cluttered environments, as the heuristics required to find collision-free paths become increasingly complex. Moreover, post-processing is often needed to enforce dynamical feasibility without compromising collision avoidance [13]. Nonetheless, these methods retain desirable theoretical guarantees, such as probabilistic completeness and asymptotic optimality.

Optimisation-based planners often rely on sampling, either to generate good warm starts or within the solver itself, due to the nonlinear and nonconvex nature of the underlying problems. These methods are well suited to mobile manipulator systems, as they allow explicit enforcement of kino-dynamic and task-level constraints. The Graph of Convex Sets (GCS) framework [14] extends shortest-path search to hybrid spaces using convex sets and mixed-integer convex programming, while the Safe Boxes method [15] builds on this idea by computing smooth, time-parameterised trajectories over precomputed convex cells. TrajOpt [16] formulates motion planning as sequential convex optimisation with signed-distance-based collision constraints, and CHOMP [13] uses functional-gradient optimisation over continuous trajectories, though it is sensitive to initialisation and local minima. The GCS formulation becomes computationally slow even for dual-arm collision free motion, sometimes requiring up to 20 minutes to compute a trajectory. In contrast, TrajOpt and CHOMP are fundamentally single robot planners that require extensive customisation to handle a mobile multi-manipulator system, and they do not natively support closed kinematic chain constraints.

In the realm of multi-manipulator systems, recent work has advanced both control strategies and hybrid task-motion planning. For low-level control under contact uncertainty, [17] proposes a distributed, event-triggered adaptive controller for teams manipulating with rolling contacts and unknown dynamics. Their scheme eliminates the need for force/torque sensing and guarantees slip avoidance through internal-force regulation. The work in [18] introduces timed abstractions for distributed cooperative manipulation: a model-free controller with prescribed performance and no inter-agent communication that lifts continuous dynamics to a finite timed transition system, thereby enabling the satisfaction of Metric-Interval Temporal Logic tasks over workspace partitions. At the planning level, [19] address multi-robot geometric task-and-motion planning (MR-GTAMP) by combining motion-planner-derived reachability and occlusion information with a mixed-integer program to guide Monte Carlo Tree Search over collaborative action sequences. In [20], the authors couple a completeness-guaranteed configuration-space cell decomposition with distributed low-level control (specifically, reference governors and prescribed-performance control) to achieve safety and convergence for teams transporting a rigidly grasped payload in clutter, without requiring continuous inter-robot communication. The closest work to our approach is [21], which formulates a kinodynamic motion planning framework for collaborative object transportation by multiple mobile manipulators in dynamic environments. Their method combines a global piecewise-linear path

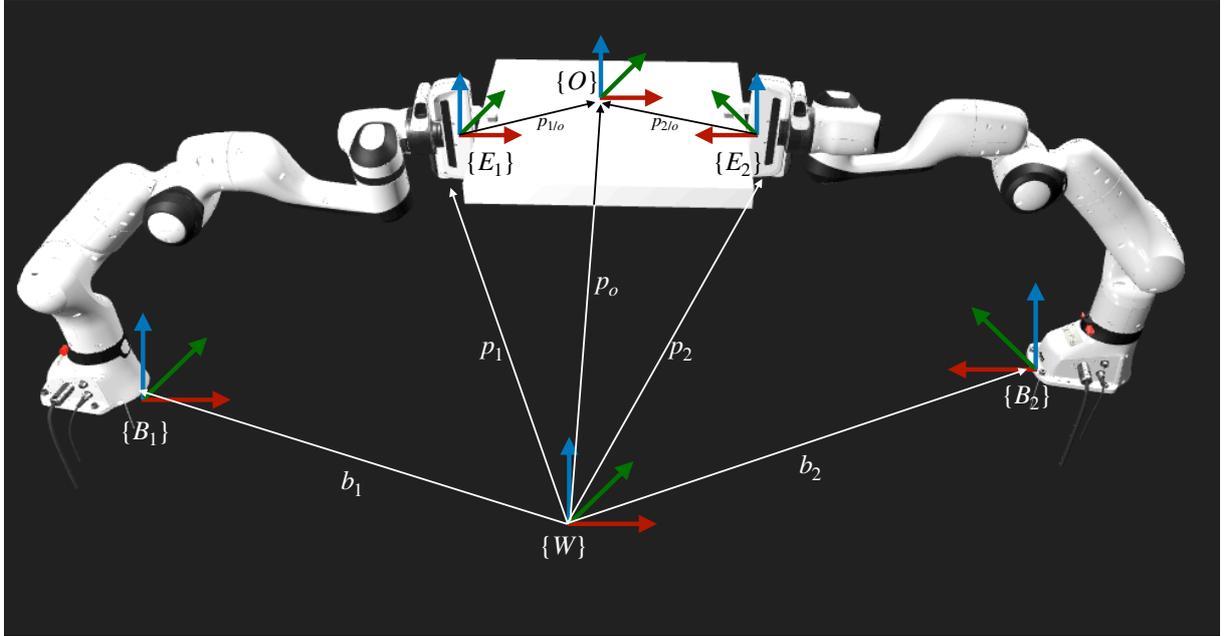


Figure 1: Visualisation of the frames for two Franka Emika Panda arms rigidly grasping an object [23].

planner with a local receding-horizon trajectory generator that accounts for kinodynamic constraints, obstacle avoidance, and joint planning of both the bases and manipulators. However, obstacle avoidance is performed in 2D, where the robots are projected onto a plane, and the framework does not permit reconfiguration of either the manipulators or the bases. As a result, collisions may still occur in 3D, since the entire manipulator-base-object structure is treated as rigid.

### 1.2. Outline

The remainder of the paper is structured as follows. Section 2 introduces the necessary assumptions and formally states the problem. Section 3 presents the proposed framework and the main technical results. Section 4 presents high-fidelity physics simulations, and Section 5 concludes the paper.

## 2. Problem Statement

We start by presenting the dynamics of the multi-manipulator system studied in this work and the environmental assumptions before formulating the problem.

### 2.1. System Dynamics

The system consists of  $N$  mobile manipulators rigidly grasping an object. A rigid grasp does not allow relative motion between the end effectors and the grasped object. Let  $\{W\}$  be the inertial (world) frame of reference. Each mobile manipulator is composed of a serial chain manipulator mounted on a mobile base. By  $\{B_i\}$ , we denote the frame attached to the mobile base of the  $i$ th mobile manipulator and by  $\{E_i\}$  we denote the frame attached to the end-effector of the  $i$ th mobile manipulator, 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 is given by [22],

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

where  $q_i = \begin{bmatrix} b_i \\ \theta_i \end{bmatrix} \in \mathbb{R}^{n_i}$ , with  $b_i \in \mathbb{R}^2$  denoting the base position and  $\theta_i \in \mathbb{R}^{n_i-2}$  the joint angles of the arm. Here,  $M_i(q_i)$  is the inertia matrix,  $C_i(q_i, \dot{q}_i)\dot{q}_i$  captures Coriolis and centrifugal effects,  $g_i(q_i)$  represents gravitational torques, and  $\tau_i$  denotes the input control torques applied to the system. We currently present the formulation of a holonomic mobile base, where  $b_i$  denotes the position in 2D and in Section 3.2 we show how this generalises to nonholonomic bases. The relationship between  $q_i$  and  $p_i$  is governed by the forward kinematics,

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

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  $f(q_i) : \mathbb{R}^{n_i} \rightarrow \text{SE}(3)$  is a function that maps the joint configuration  $q_i$  to the end-effector position  $p_i$ . The stacked dynamics of all mobile manipulators is given by

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) = \tau \quad (3)$$

where,  $M(q) = \text{diag}(M_1(q_1), \dots, M_N(q_N))$ ,  $C(q, \dot{q}) = \text{diag}(C_1(q_1, \dot{q}_1), \dots, C_N(q_N, \dot{q}_N))$ ,  $g(q) = [g_1(q_1)^\top \ \dots \ g_N(q_N)^\top]^\top$ , and,  $\tau = [\tau_1^\top \ \dots \ \tau_N^\top]^\top$ . The pose and velocity of the object's center of mass are denoted by  $x_o \in \mathbb{R}^6$  and  $v_o \in \mathbb{R}^6$ . The dynamics of the object are

$$\dot{x}_o = J_{o_r}^{-1}(x_o)v_o, \quad (4a)$$

$$M_o(x_o)\dot{v}_o + C_o(x_o, v_o)v_o + g_o(x_o) = \lambda_o \quad (4b)$$

where  $M_o : \mathbb{R}^3 \times \mathbb{T}^3 \rightarrow \mathbb{R}^{6 \times 6}$  is the positive-definite inertia matrix,  $C_o : \mathbb{R}^3 \times \mathbb{T}^3 \times \mathbb{R}^6 \rightarrow \mathbb{R}^{6 \times 6}$  is the Coriolis matrix and  $g_o : \mathbb{R}^3 \times \mathbb{T}^3 \rightarrow \mathbb{R}^6$  is the gravity vector. Additionally,  $J_{o_r} : \mathbb{T}^3 \rightarrow \mathbb{R}^{6 \times 6}$  is the object representation Jacobian and  $\lambda_o \in \mathbb{R}^6$  is the force vector acting on the object's center of mass. When a robot interacts with an environment, it experiences forces as a result. The single robot dynamics in (1) in joint space then becomes in task space as

$$\bar{M}(q_i)\dot{v}_i + \bar{C}(q_i, \dot{q}_i)v_i + \bar{g}(q_i) = u_i - \lambda_i, \quad (5)$$

where  $\bar{M}_i : \mathbb{R}^{n_i} \rightarrow \mathbb{R}^{6 \times 6}$  is the task-space positive definite inertia matrix,  $\bar{C}_i : \mathbb{R}^{2n_i} \rightarrow \mathbb{R}^{6 \times 6}$  the task-space Coriolis matrix,  $\bar{g}_i : \mathbb{R}^{n_i} \rightarrow \mathbb{R}^6$  the task-space gravity vector,  $u_i \in \mathbb{R}^6$  is the task-space input wrench and  $\lambda_i \in \mathbb{R}^6$  is the generalised force vector that agent  $i$  exerts on the object.

**Assumption 1.** We assume that the robots grasp the object with rigid grasps, i.e., the relative pose between the end-effector frame of the  $i$ th robot,  $\{E_i\}$ , and the object frame,  $\{O\}$ , remains constant.

The rigid grasp assumption leads to the velocity constraint,

$$J(q)\dot{q} = G^\top v_o$$

where  $J(q) = \text{diag}(J_i(q_i))$  is the block diagonal manipulator Jacobian. The object dynamics (4) are coupled to the robot dynamics (5) by the grasping constraint

$$\lambda_o = G\lambda, \quad (6)$$

where  $G := G(q) : \mathbb{R}^{\sum_{i=1}^N n_i} \rightarrow \mathbb{R}^{6 \times 6N}$  is the full row-rank grasp matrix defined as

$$G(q) = [J_{o_1}^\top(q_1) \quad J_{o_2}^\top(q_2) \quad \dots \quad J_{o_N}^\top(q_N)]$$

where  $J_{o_i} : \mathbb{R}^{n_i} \rightarrow \mathbb{R}^{6 \times 6}$  is the object-to- $i$ 'th agent Jacobian  $J_{o_i}(q_i) = \begin{bmatrix} I_3 & S(-p_{i/o}(q_i)) \\ 0_{3 \times 3} & I_3 \end{bmatrix}$ . This results in the coupled object-robot dynamics

$$\tilde{M}\dot{v}_o + \tilde{C}v_o + \tilde{g} = G^\top u \quad (7)$$

where  $\tilde{M} = M_o + G\bar{M}G^\top$ ,  $\tilde{C} = C_o + G\bar{C}G^\top + GCG^\top$ ,  $\tilde{g} = g_o + G\bar{g}$ . Observe that the position of the object can be obtained via the end effector position by the relation  ${}^W R_i^{E_i} p_{i/o} + p_i$ , see Figure 1. However, we do not design  $u$  from (7) directly, but design  $\tau$  from (3) and impose the grasping as constraints to an optimisation problem as seen in Section 3.3. The task space control can be computed using,  $\tau = J^\top G^{-\top} u$ . See [18] for more details on the coupled object-robot dynamics.

## 2.2. Signal Temporal Logic

The desired task for the object is specified using a Signal Temporal Logic (STL) formula over the centre of mass of the grasped object. Let  $x : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}^n$  be a continuous-time signal. Signal Temporal Logic [24] is a predicate-based logic with the following syntax,

$$\phi = \top \mid \mu \mid \neg\phi \mid \phi_1 \wedge \phi_2 \mid \phi_1 \mathcal{U}_{[a,b]} \phi_2 \quad (8)$$

where  $\phi_1, \phi_2$  are STL formulae and  $\mu$  is a predicate of the form  $\mu : \mathbb{R}^n \times \mathbb{R}_{\geq 0} \rightarrow \mathbb{B}$  defined via a predicate function  $p : \mathbb{R}^n \times \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}$  as

$$\mu = \begin{cases} \top & p(x, t) \geq 0 \\ \perp & p(x, t) < 0 \end{cases}. \quad (9)$$

The satisfaction relation  $(x, t) \models \phi$  indicates that signal  $x$  satisfies  $\phi$  at time  $t$  and is defined recursively as follows:

$$\begin{aligned} (x, t) \models \mu & \Leftrightarrow p(x, t) \geq 0 \\ (x, t) \models \neg\phi & \Leftrightarrow \neg((x, t) \models \phi) \\ (x, t) \models \phi_1 \wedge \phi_2 & \Leftrightarrow (x, t) \models \phi_1 \wedge (x, t) \models \phi_2 \\ (x, t) \models \phi_1 \mathcal{U}_{[a,b]} \phi_2 & \Leftrightarrow \exists t_1 \in [t+a, t+b] \text{ s.t. } (x, t_1) \models \phi_2 \\ & \quad \wedge \forall t_2 \in [t, t_1], (x, t_2) \models \phi_1. \end{aligned}$$

A signal  $x$  satisfies the *Until* operator,  $(x, t) \models \phi_1 \mathcal{U}_{[a,b]} \phi_2$ , if  $\phi_1$  holds at all times before  $\phi_2$  holds and  $\phi_2$  holds at some time instance between  $a$  and  $b$ . More details on STL semantics can be found in [25]. For brevity, we omit further details on the STL formulation and instead rely on our recently developed MAPS<sup>2</sup> algorithm [4] to generate the desired trajectory for the object from the given STL specification for a multi-robot system.

### 2.3. Environment

Let  $\mathcal{W} \subset \mathbb{R}^3$  denote the workspace within which the robots operate, and let  $\mathcal{O} \subset \mathcal{W}$  represent the region occupied by static obstacles. We denote  $\mathcal{S}_r$  as the set that consists of all points  $p \in \mathcal{W}$  that physically belong to the coupled  $N$  mobile manipulator system, i.e., they consist part of either the volume of the robots or the volume of the grasped object. Note that these points depend on the actual value of  $q$ . Let all points  $o_p \in \mathcal{O}$  be the points that physically belong to all the obstacles;  $o_p$  are assumed to be independent of time.

**Assumption 2.** *The obstacle-free workspace  $\mathcal{W} \setminus \mathcal{O}$  is non-empty, and the obstacle region  $\mathcal{O}$  remains static over time.*

This is a standard assumption that allows planning in the obstacle free region. It is computationally infeasible to evaluate the distance between all points  $p \in \mathcal{S}_r(q)$  and all points  $o_p \in \mathcal{O}$ . Instead, we select a finite set of representative points from  $\mathcal{S}_r(q)$  that capture the geometric features of the robots and the grasped object, and a finite set of representative points on the surfaces of the obstacles  $\mathcal{O}$ . For any such robot–obstacle representative pair, we define  $d_{r,o}(q) \in \mathbb{R}$  as the signed distance between the selected points:  $d_{r,o} > 0$  indicates separation, and  $d_{r,o} < 0$  indicates penetration. We further assume that these selected points faithfully detect collisions, in the sense that

$$d_{r,o}(q) > 0 \iff \mathcal{S}_r(q) \cap \mathcal{O} = \emptyset, \quad d_{r,o}(q) \leq 0 \iff \mathcal{S}_r(q) \cap \mathcal{O} \neq \emptyset.$$

In this work, the signed distance values are obtained directly from a physics engine. For real-world deployment, an RGB-D camera is typically used, and the resulting point cloud can be processed to construct a signed distance field. A representative example is NVIDIA’s `nvblox` library, which uses the algorithm in [26] to reconstruct a signed distance field from depth data. Further, let

$$\mathcal{S}_r(q) = \bigcup_{\ell \in \mathcal{L}} \mathcal{S}_\ell(q)$$

be the union of the occupied volumes  $\mathcal{S}_\ell(q) \subset \mathcal{W}$  of rigid body  $l$  belonging to the robots and the grasped object, indexed by  $\mathcal{L}$ . Let  $\mathcal{C}_{\text{allow}} \subseteq \mathcal{L} \times \mathcal{L}$  denote the set of pairs  $(\ell_i, \ell_j)$  for which contact is allowed (e.g., between the gripper and the object or any two adjacent links), and the rest as the disallowed pairs  $\mathcal{C}_{\text{forbid}} = \{(\ell_i, \ell_j) \in \mathcal{L} \times \mathcal{L} \mid i \neq j, (\ell_i, \ell_j) \notin \mathcal{C}_{\text{allow}}\}$ . If

$$\mathcal{S}_{\ell_i}(q) \cap \mathcal{S}_{\ell_j}(q) = \emptyset \quad \forall (\ell_i, \ell_j) \in \mathcal{C}_{\text{forbid}}, \quad (10)$$

then no self-collision occurs; that is, the robots do not collide with themselves, with neighbouring robots, or with the manipulated object. Note that generating an obstacle-free reference trajectory for the centre of mass of the object does not guarantee that the object will avoid collisions, nor does it guarantee that the mobile manipulators will avoid collisions with obstacles or with each other. From here on, for simplicity we further assume that  $x_o \in \mathbb{R}^3$  and only captures the position of the object in 3D coordinates. The problem we address in this work is the following:

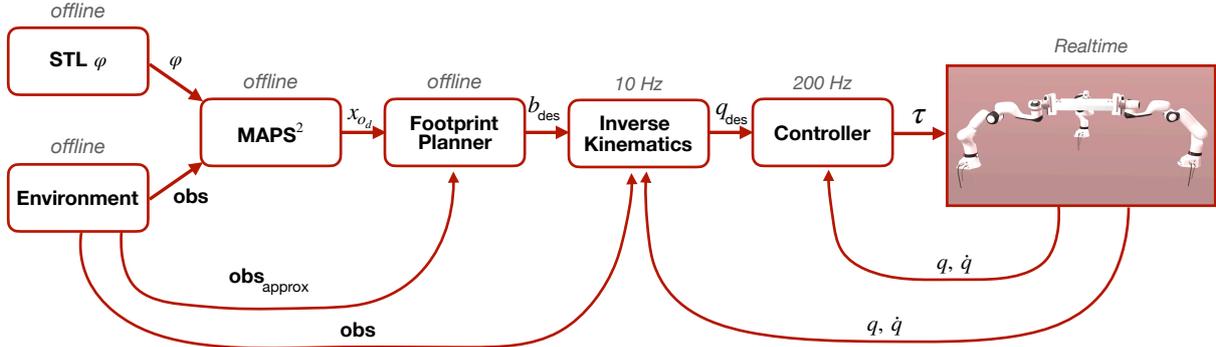


Figure 2: Proposed solution architecture.

**Problem 1.** Consider a system of  $N$  mobile manipulators rigidly grasping an object with coupled dynamics (7), and let an STL formula  $\varphi$  of the form (8) specify the desired spatio-temporal behaviour of the object state  $x_o$ . The objective is to design control torques  $\tau_i$  in (1) for each mobile manipulator  $i = 1, \dots, N$  such that the resulting  $(x_o, t) \models \varphi$ ,  $d_{r,o} > 0 \forall t$  and (10) holds.

### 3. Main Results

In this section, we present our main algorithm. We begin with an intuitive overview of the proposed approach. The method is illustrated in Figure 2 and summarised below:

- (A) Compute a desired collision-free trajectory for the object  $x_{o_d}$  using MAPS<sup>2</sup> (Section 3.1).
- (B) Determine desired collision-free base positions  $b_{ides}$  for each mobile base  $i$  using the *Footprint Planner*, based on  $x_{o_d}$  (Section 3.2).
- (C) Compute the desired joint angles  $q_{ides}$  using *Inverse Kinematics*, given  $b_{ides}$  and  $x_{o_d}$  (Section 3.3).
- (D) Design a control law to track the desired joint angles  $q_{ides}$  for each agent  $i = 1, \dots, N$  (Section 3.4).

Multi-robot manipulation in constrained environments naturally gives rise to a nonlinear hybrid systems problem. The workspace induces discrete geometric modes (e.g., free-space motion versus proximity to obstacles or narrow passages) that require coordinated reconfiguration of the entire team, while motion within each mode is governed by continuous robot and object dynamics. The proposed solution pipeline is hybrid and multi-rate: an offline layer generates an STL-satisfying object trajectory and a collision-free base footprint plan, while an online layer alternates between discrete inverse-kinematics updates at a slower rate and continuous-time feedback control at a faster rate using zero-order hold. This interaction between discrete mode changes, multi-rate computation, and continuous control places the problem and its solution squarely within the scope of nonlinear hybrid systems.

#### 3.1. MAPS<sup>2</sup>:

Based on a given STL specification, the MAPS<sup>2</sup> algorithm computes a desired collision-free trajectory for the object, represented as a sequence of waypoints

$$\{(t_k, x_{o_d}(t_k))\}_{k=0}^{K'}, \quad x_{o_d}(t_k) \in \mathbb{R}^3, \quad (11)$$

where each waypoint corresponds to a sampled state for the object. As MAPS<sup>2</sup> is a sampling-based algorithm, the resulting trajectory is non-smooth. To address this, we perform a two-stage smoothing procedure that yields a smooth continuous trajectory. First, we apply cubic spline interpolation to each coordinate of the trajectory. Writing  $x_{o_d}(t_k) = [x_{o_d}(t_k) \ y_{o_d}(t_k) \ z_{o_d}(t_k)]^\top$ , we construct cubic spline interpolants  $\tilde{x}(t), \tilde{y}(t)$  and  $\tilde{z}(t)$  such that  $\tilde{x}(t_k) = x_{o_d}(t_k), \tilde{y}(t_k) = y_{o_d}(t_k)$ , and  $\tilde{z}(t_k) = z_{o_d}(t_k)$  for  $k = 0, \dots, K'$ . Since the  $t_k$  are not evenly spaced as they are sampled randomly from a distribution, we construct a dense time grid,  $\{t_j\}_{j=0}^K$  where  $t_0 = t_{\min}$  and  $t_K = t_{\max}$  are the minimum and maximum time instances from (11). Next, we evaluate and interpolate the cubic spline on this dense time grid to obtain  $\tilde{x}_{o_d}(t_j) = [\tilde{x}(t_j) \ \tilde{y}(t_j) \ \tilde{z}(t_j)]^\top$  where  $j = 0, \dots, K$ . Spline interpolation can still preserve some oscillations from the original trajectory causing high velocities (local gradients). We therefore apply

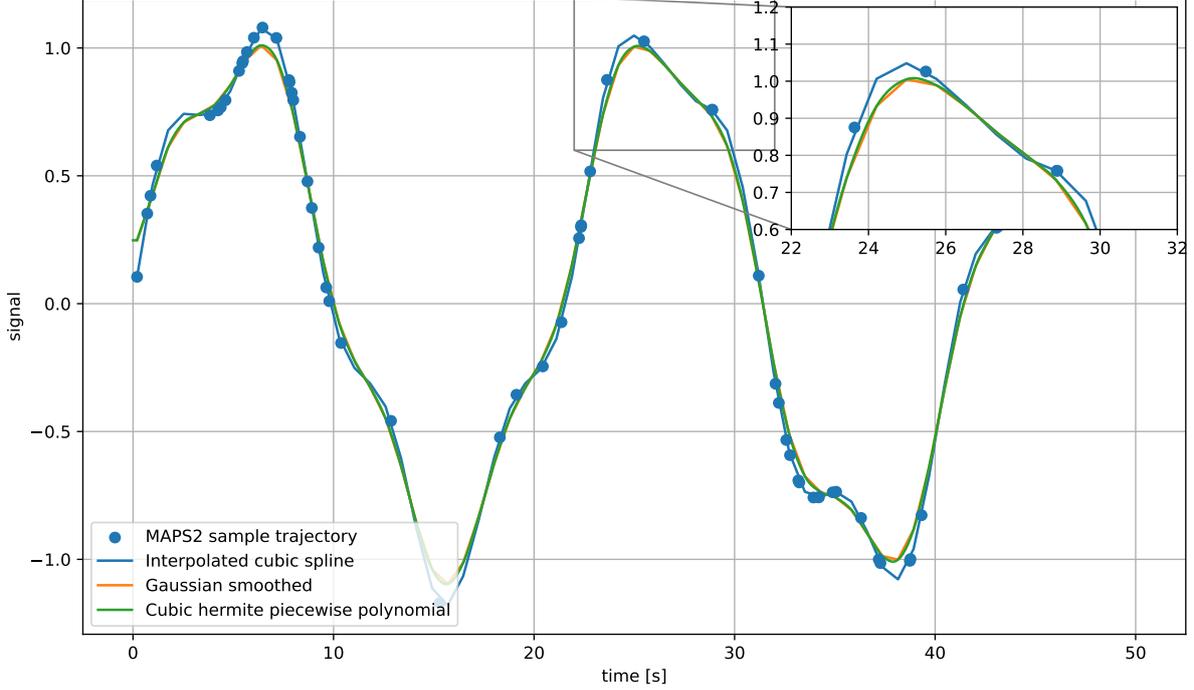


Figure 3: Post-processing of trajectories from MAPS<sup>2</sup>.

Gaussian smoothing to the interpolated sequence, implemented as a discrete convolution  $x_{o_d}^{\text{sm}}(t_j) = \sum_{\ell=0}^K w_{j-\ell} \tilde{x}_{o_d}(t_\ell)$ ,  $w_n \propto \exp\left(-\frac{n^2}{2\sigma^2}\right)$ , with the normalisation  $\sum_n w_n = 1$ . The parameter  $\sigma > 0$  controls the degree of smoothing. This produces smooth positions  $x_{o_d}^{\text{sm}}(t_j) = [x_{o_d}^{\text{sm}}(t_j) \ y_{o_d}^{\text{sm}}(t_j) \ z_{o_d}^{\text{sm}}(t_j)]$ . To obtain smooth differentiable trajectories for the solvers,  $x_{o_d}^{\text{sm}}(t_j)$  is encoded as a cubic Hermite piecewise polynomial [27] as follows,

$$x_{o_d}^{\text{ch}}(t) = h_0(s) x_{o_d}^{\text{sm}}(t_j) + h_1(s) x_{o_d}^{\text{sm}}(t_{j+1}) + h_2(s) \Delta t_j v_{o_d}(t_j) + h_3(s) \Delta t_j v_{o_d}(t_{j+1}),$$

where  $s = \frac{t-t_j}{\Delta t_j}$ ,  $\Delta t_j = t_{j+1} - t_j$ , and  $h_0, h_1, h_2, h_3$  are the standard cubic Hermite basis functions  $h_0(s) = 2s^3 - 3s^2 + 1$ ,  $h_1(s) = -2s^3 + 3s^2$ ,  $h_2(s) = s^3 - 2s^2 + s$ , and  $h_3(s) = s^3 - s^2$ . Here the velocities  $v_{o_d}(t_j)$  were computed as  $v_{o_d}(t_j) = \frac{x_{o_d}^{\text{sm}}(t_{j+1}) - x_{o_d}^{\text{sm}}(t_{j-1})}{t_{j+1} - t_{j-1}}$ ,  $j = 1, \dots, K - 1$ . This construction yields a trajectory  $x_{o_d}^{\text{ch}}(t) \in \mathbb{R}^3$  that is smooth and continuously differentiable, thus suitable for downstream optimisation solvers. We henceforth abuse the notation and denote  $x_{o_d}^{\text{ch}}(t)$  as  $x_{o_d}(t)$ . These signals are depicted in the Figure 3.

### 3.2. Footprint Planner:

Given the desired object trajectory  $x_{o_d}(t)$ , directly computing all joint angles  $q_i$  for  $i = 1, \dots, N$  quickly becomes intractable, as it requires optimising over  $N \times n_i$  decision variables and handling the non-convexities introduced by obstacles and forward kinematics. Moreover, optimising over a time horizon  $[0, t_{\text{max}}]$  with  $K$  time steps results in  $K \times N \times n_i$  decision variables. To mitigate this complexity, we decompose the planning problem into two stages: planning the mobile base trajectories and planning the manipulator arm configurations. Given  $x_{o_d}(t)$ , the main objective is to compute a collision-free set of base trajectories

$$b_{\text{des}} = [b_{1\text{des}}^\top \ \dots \ b_{N\text{des}}^\top]^\top$$

that satisfy the following requirements:

- The base trajectories of all mobile bases must not be too close such that the bases collide with each other.
- The bases must avoid collisions with obstacles.

- Finally, we need two anchor requirements. One that anchors bases close to the desired object trajectory  $x_{od}(t)$  and another that anchors bases close to each other. The  $b_{des}$  that we compute here will only function as a reference for the inverse kinematics problem later on, thus instead of computing the exact forward kinematics we only provide soft thresholds for both these anchors.

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 post-processed trajectory from MAPS<sup>2</sup>  $x_{od}$  (as described in Section 3.1) be discretised into  $K$  time steps over the horizon  $[0, t_{max}]$ , where  $t_{max}$  is the time horizon of the STL formula (the final time instance of the trajectory  $x_{od}$  from MAPS<sup>2</sup>). The optimisation problem thus is as follows,

$$\min_{b_{des}} \sum_{k=1}^K \sum_{i=1}^N \sum_{\substack{j=1 \\ i \neq j}}^N w_i \left( \|b_{ides}[k] - b_{jdes}[k]\|^2 - \alpha_i^2 \right)^2 \quad (12a)$$

$$\text{s.t.} \quad \left\| \frac{1}{N} \sum_{i=1}^N b_{ides}[k] - x_{od}(t) \right\|^2 \leq \epsilon \quad (12b)$$

$$b_{ides}[k+1] - b_{ides}[k] \in [-\eta, \eta], \quad \forall i = 1, \dots, N \quad (12c)$$

$$z_{obj}[k] - z_{x_{od}}[k] \in [-\delta, \delta] \quad (12d)$$

$$b_{des}[k] \in \mathcal{W} \setminus \mathcal{O}_{approx} \quad (12e)$$

$$b_{des}[0] = x_{od}(0) \quad (12f)$$

$$b_{des}[t_{max}] = x_{od}(t_{max}) \quad (12g)$$

Here,

- (12a) defines the cost function that penalises deviations of the squared distance between any two bases from a target squared distance  $\alpha_i^2$  ( $\alpha_i \in \mathbb{R}_{>0}$ ). This cost essentially keeps the bases close to each other such that they do not drift too far apart and release the object. The distance  $\alpha_i$  is not uniform over all pairs of bases because if the manipulators grasp the object in a non-symmetric configuration (see Fig 4, where robots 1 and 3 are farther apart than robots 1 and 2 or robots 2 and 3), then the values of  $\alpha_i$  vary according to the kinematic reach from each base to its end-effector. The weights  $w_i \in \mathbb{R}_{>0}$  are positive scalars that can be tuned to control how strictly the target inter-base distances are enforced. In this formulation, we consider all pairwise inter-base distances and penalise deviations from their desired values.
- (12b) ensures that the centroid of the base positions remains  $\epsilon \in \mathbb{R}_{>0}$  close to the desired object trajectory  $x_{od}$ . Here, only the  $x$  and  $y$  coordinates of  $x_{od}$  are considered, as the bases operate in a fixed height plane. This constraint maintains the proximity of the centroid formed by all base positions to the smoothed trajectory generated by the MAPS<sup>2</sup> algorithm, which already provides a globally feasible path - an advantage we aim to transfer to the base trajectories.
- (12c) enforces a bounded step-to-step displacement, which corresponds to a velocity bound up to the discretisation factor. The allowable displacement  $\eta \in \mathbb{R}_{>0}$  can depend on factors such as the object or payload weight, the friction properties of the floor, and the maximum admissible acceleration of the mobile bases. This constraint prevents abrupt motions between consecutive time steps and ensures that the base trajectories remain physically plausible and trackable.
- (12d) enforces a coupling between the spatial arrangement of the bases and the height of the grasped object. The average pairwise distance of the bases is computed at each time step and used to predict the object height via a simple linear model,  $z_{obj}[k] = z_{ref} - \kappa \text{spread}[k]$ , where  $\text{spread}[k] = \frac{2}{N(N-1)} \sum_{(i,j), i < j} \|b_{ides}[k] - b_{ides}[k]\|$ ,  $z_{ref}$  is a fixed distance and  $\kappa$  is a scaling factor. This predicted height is then required to remain within  $\pm\delta$  of the desired object height obtained from the continuous MAPS<sup>2</sup> trajectory. This constraint ensures that the bases do not come too close to each other thereby raising the object over the desired object height, and also are not too far apart such that the object drops below the desired object height.
- (12e) represents the obstacle-avoidance constraint in two dimensions. The obstacle region  $\mathcal{O}$  is assumed to be decomposed into convex components, and let

$$\mathcal{O}_{proj} := \{ (x, y) \in \mathbb{R}^2 \mid \exists z \in \mathbb{R} \text{ such that } (x, y, z) \in \mathcal{O} \}.$$

This constraint ensures that the base positions remain collision free within the workspace. Each 2D obstacle is approximated by a super-ellipse in the  $(x, y)$  plane. For each obstacle  $\ell = 1, \dots, L$ , let  $c_\ell = (c_\ell^x, c_\ell^y)^\top$  denote its centre and let  $a_\ell^x, a_\ell^y > 0$  denote its shape parameters along the  $x$  and  $y$  directions, respectively. We define the corresponding super-ellipse function as

$$F_\ell(b) = \left( \frac{b_x - c_\ell^x}{a_\ell^x} \right)^4 + \left( \frac{b_y - c_\ell^y}{a_\ell^y} \right)^4, \quad b = (b_x, b_y)^\top \in \mathbb{R}^2.$$

The interior of obstacle  $\ell$  corresponds to points satisfying  $F_\ell(b) < \rho_\ell$ , and the feasible workspace is approximated by

$$\mathcal{W} \setminus \mathcal{O}_{\text{approx}} = \{ b \in \mathbb{R}^2 \mid F_\ell(b) \geq \rho_\ell, \forall \ell = 1, \dots, L \},$$

where  $\rho_\ell \geq 1$  specifies a safety margin around obstacle  $\ell$  and encapsulates the radii of the bases, and  $\mathcal{O}_{\text{proj}} \subseteq \mathcal{O}_{\text{approx}}$ . Accordingly, the obstacle-avoidance constraint imposed on the base trajectories is

$$F_\ell(b_{i\text{des}}[k]) \geq \rho_\ell, \quad i = 1, \dots, N, \ell = 1, \dots, L, k = 0, \dots, K,$$

where  $b_{i\text{des}}[k]$  denotes the planar position of base  $i$  at time index  $k$ .

- (12f) and (12g) define the initial and final conditions for the base trajectories.

Note that for nonholonomic mobile bases, the corresponding kinematic constraint can be incorporated as an additional equality constraint in the optimisation problem (12).

### 3.3. Inverse Kinematics:

Given the desired object trajectory  $x_{o_d}(t)$  and the base trajectories  $b_{\text{des}}(t)$  computed by the footprint planner, the goal of this stage is to compute a feasible joint configuration  $q_{\text{des}}(t)$  for all manipulators that realises a rigid multi-arm grasp of the object while ensuring kinematic constraints and obstacle avoidance. We denote the base trajectories  $b_{\text{des}}(t)$  as  $\bar{b}_{\text{des}}$  in this section as the inverse kinematics will compute new positions for the bases and the  $b_{\text{des}}(t)$  from the footprint planner will be used as a soft constraint. Recall that the *Footprint Planner* is an offline algorithm while the IK is an online algorithm. In the implementation, this is achieved by solving a inverse-kinematics (IK) problem at a certain frequency. The following IK problem is solved upto a fixed number of optimisation steps in order to ensure computation at a certain frequency. The considerations encoded in the IK formulation are summarised below.

- The arms along with the grasped object should avoid collisions with obstacle, i.e.,  $d_{r,o} > 0$ .
- The arms should ensure tracking of the desired base trajectories  $b_{i\text{des}}$  and the desired object trajectory  $x_{o_d}$ .
- The arms should maintain configurations that avoid kinematic singularities.
- The end-effectors should satisfy the rigid grasp constraints on the object.
- The solution must respect the joint limit constraints.

These considerations can be formulated as an optimisation problem, where the objective is to find the desired joint angles  $q_{\text{des}} = [q_{1\text{des}}^\top \ \dots \ b_{N\text{des}}^\top]^\top$  that satisfy the above requirements.

$$\min_{q_{\text{des}}} \|q_{\text{des}}[k] - q_0\|^2 + \sum_{i=1}^N \left\| \left( {}^W R_i^{E_i} p_{i/o} + p_i \right)[k] - x_{o_d}(t) \right\|^2 + c_{\text{collision}} \quad (13a)$$

$$\text{s.t.} \quad p_{i/o}[k] = p_{i/o}[0] \quad (13b)$$

$${}^O R_i^{E_i}[k] = {}^O R_i^{E_i}[0] \quad (13c)$$

$$\bar{b}_{\text{des}}[k] - \nu \leq b_{\text{des}}[k] \leq \bar{b}_{\text{des}}[k] + \nu \quad (13d)$$

$${}^W X_i^{E_i}[k] = f(q_{i\text{des}}[k]) \quad (13e)$$

- (13a) defines the overall cost function, composed of three complementary terms. The first term penalises deviations of the joint angles from a reference configuration  $q_0$ , which is selected to promote well-conditioned, collision-free postures and to provide a regularising effect that improves numerical robustness of the IK solver. The second term promotes accurate tracking of the desired object

trajectory by penalising the discrepancy between the object position, expressed as  $({}^W R_i^{E_i} p_{i/o} + p_i)$ , and the desired centre-of-mass trajectory  $x_{od}(t)$ .

The final term,  $c_{\text{collision}}$ , encodes obstacle avoidance penalty at the IK level using a smooth, signed distance based cost function. Let  $d_{r,o}(q_{\text{des}})$  (as defined in Section 2.3) denote the signed distance between a robot geometry and obstacle geometry. For each robot-obstacle representative pair, we define a local potential

$$\phi(d_{r,o}(q_{\text{des}})) = w_{r,o}(d_{r,o}(q_{\text{des}})) \exp(-\alpha (d_{r,o}(q_{\text{des}}) - d_{\text{safe}})^2),$$

where  $\alpha > 0$  is a shaping parameter,  $d_{\text{safe}} > 0$  is a prescribed safety margin, and  $w_{r,o}$  is a distance-dependent weight that down-weights pairs that are already far from contact. The total collision cost is then obtained by summing these potentials over all robot-obstacle pairs,

$$c_{\text{collision}}(q) = \sum_{(r,o)} \phi(d_{r,o}(q_{\text{des}})).$$

This construction was chosen after extensive experimentation to satisfy three requirements simultaneously: (i) negligible influence when the manipulators are well separated from obstacles, (ii) rapidly increasing cost and informative gradients as the distance approaches the safety margin, and (iii) smoothness and differentiability with respect to  $q_{\text{des}}$ , which are crucial for reliable convergence of the nonlinear optimisation.

- (13b) and (13c) together impose the rigid-grasp constraints necessary for cooperative manipulation. Constraint 13b enforces a fixed relative position between the end-effector and its corresponding grasp point on the object, ensuring that  $p_{i/o}$  does not drift over time. Constraint (13c) ensures that the relative orientation between the end-effector and the object remains constant, thereby enforcing a rigid grasp throughout the motion.
- (13d) restricts the base positions to remain within a bounded neighbourhood of the desired base trajectory  $\bar{b}_{\text{des}}[k]$ . The tolerance  $\nu \geq 0$  provides the flexibility needed to accommodate minor adjustments for collision avoidance or kinematic feasibility, while still ensuring that the mobile bases track their planned motion closely.
- (13e) enforces consistency between the decision variables and the robot's kinematics. This constraint ensures that the end-effector pose  ${}^W X_i^{E_i}[k]$  corresponds to the forward kinematics  $f(q_{i\text{des}}[k])$  of the manipulator at the desired joint configuration. It guarantees that all position and orientation expressions appearing in the optimisation are derived from physically realisable robot configurations.

**Remark 1.** *The IK optimisation problem (13) is not solved to full convergence but instead executed for a fixed number of iterations. This practice is common in nonlinear, nonconvex problems, where performing only a limited number of gradient-based steps yields a solution that is sufficiently close to a local optimum while significantly reducing computation time. The permissible iteration budget is ultimately dictated by the computational capabilities of the hardware on which the optimisation runs.*

### 3.4. Control Design

The *Inverse Kinematics* module provides the desired joint configurations  $q_{\text{des}}$  at discrete update times  $t_k$ , based on the trajectories generated by *MAPS*<sup>2</sup> and the *Footprint Planner*. Between updates, the desired configuration is held constant via a zero-order hold. Each manipulator then tracks its corresponding desired configuration using a standard joint-space Proportional-Derivative (PD) controller with gravity compensation:

$$\tau_i(t) = -k_p(q_i(t) - q_{i\text{des}}[k]) - k_v \dot{q}_i(t) + g_i(q_i(t)), \quad t \in [t_k, t_{k+1}), \quad (14)$$

where  $k_p, k_v > 0$  are tunable gains, and  $g_i(q_i)$  denotes the gravity compensation term from (1). Let

$$e_i(t) := q_i(t) - q_{i,\text{des}}[k], \quad \dot{e}_i(t) := \dot{q}_i(t).$$

Consider the Lyapunov function  $V_i(e_i, \dot{e}_i) = \frac{1}{2} \dot{e}_i^\top M(q_i) \dot{e}_i + \frac{1}{2} e_i^\top K_p e_i$ . and the dynamics (1) with disturbances  $w_i(t)$  that enter through the input channel as,

$$M_i(q_i) \ddot{q}_i + C_i(q_i, \dot{q}_i) \dot{q}_i + g_i(q_i) = \tau_i + w_i \quad (15)$$

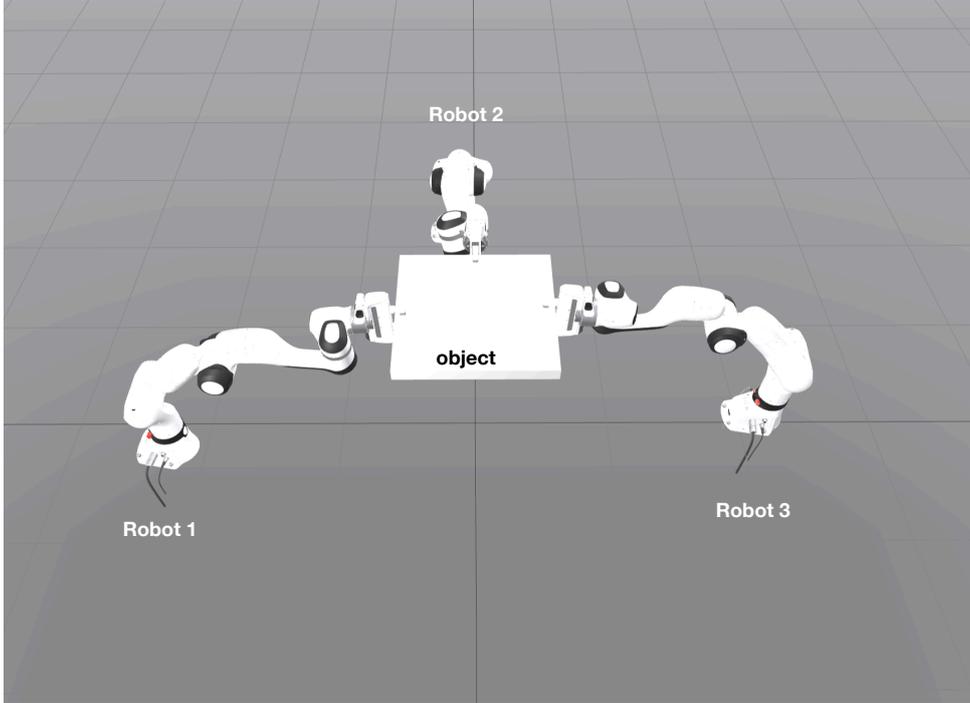


Figure 4: Setup of 3 panda arms developed by Franka Emika grabbing an object.

Substituting (14) in the above equation, its derivative satisfies

$$\dot{V}_i \leq -\dot{e}_i^\top K_v \dot{e}_i + \dot{e}_i^\top w_i(t).$$

In the disturbance-free case  $w_i(t) \equiv 0$  we obtain

$$\dot{V}_i \leq -\lambda_{\min}(K_v) \|\dot{e}_i\|^2,$$

where  $\lambda_{\min}$  is the smallest eigenvalue of  $K_c$ , so the equilibrium  $(e_i, \dot{e}_i) = (0, 0)$  is globally exponentially stable. Moreover, when  $w_i(t)$  is bounded, this inequality yields an input-to-state stability (ISS) bound for  $(e_i, \dot{e}_i)$  with respect to  $w_i(t)$ , so the tracking error remains ultimately bounded in the presence of moderate disturbances. This decentralised joint-space PD structure is sufficient in our setting, as the planning pipeline enforces multi-arm coordination and collision avoidance at the kinematic level, leaving the low-level controller to focus solely on accurate trajectory tracking.

#### 4. Simulation Results

We evaluate the proposed framework on a system composed of three Panda mobile manipulators rigidly grasping a common object, as illustrated in Fig. 4. Each manipulator is modelled in Drake [28] as a floating-base system to emulate mobile capabilities. The workspace spans a 10 [m]  $\times$  10 [m] area and contains multiple obstacles forming narrow passages of different shapes (square, triangular, and trapezoidal), shown in Fig. 5. The robots must transport the object through these passages while satisfying the given STL task specification.

##### 4.1. Task Specification

The object must visit a sequence of waypoints with tolerance  $\epsilon > 0$  while maintaining a separation of at least 0.5 m from all obstacles, this translates into the following STL task:

$$\begin{aligned} \varphi = & \mathcal{G}_{[37,38]}(\|x_o - [0 \ 8 \ 0.6]^\top\| \leq \epsilon) \wedge \mathcal{G}_{[87,88]}(\|x_o - [8 \ -1 \ 0.6]^\top\| \leq \epsilon) \wedge \\ & \mathcal{G}_{[112,113]}(\|x_o - [4 \ -4 \ 0.6]^\top\| \leq \epsilon) \wedge \mathcal{G}_{[150,151]}(\|x_o - [-5 \ -8 \ 0.6]^\top\| \leq \epsilon) \wedge \\ & \mathcal{G}_{[200,201]}(\|x_o - [-7 \ -1 \ 0.6]^\top\| \leq \epsilon) \wedge \mathcal{F}_{[220,230]}(\|x_o - [0 \ 0 \ 0.6]^\top\| \leq \epsilon) \wedge \\ & \mathcal{G}_{[0,250]}(\|x_o - \text{obs}\| \geq 0.5). \end{aligned}$$

These waypoints are shown as green markers in Fig. 5. Their placement ensures that the robots must repeatedly enter and exit narrow corridors testing collision avoidance with varying degree.

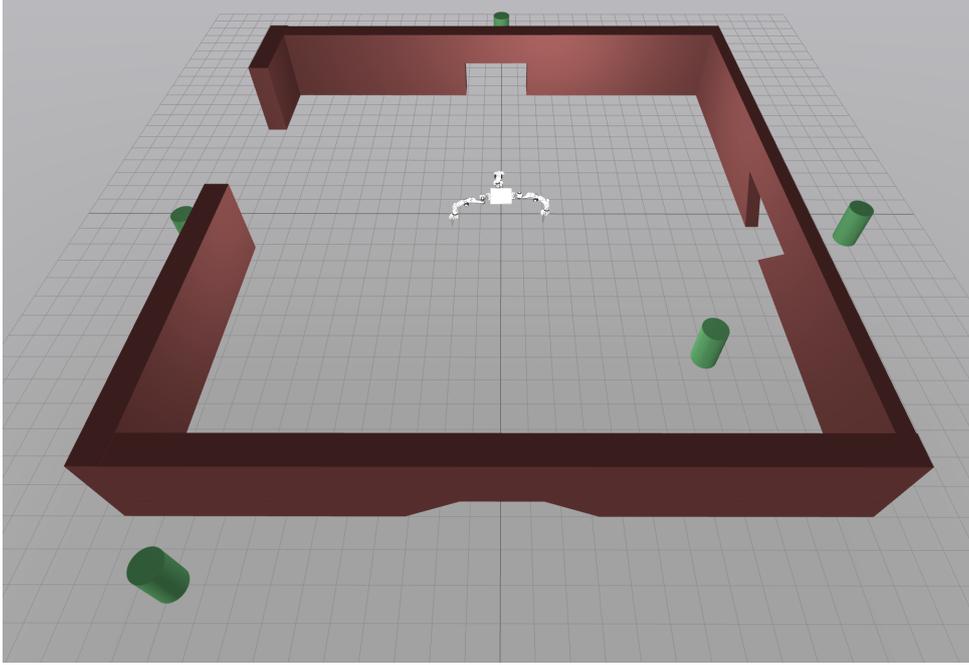


Figure 5: The workspace with obstacles shown in red, desired waypoints in green, and 3 panda mobile robots in white.

#### 4.2. Modeling Considerations

A number of practical considerations are required to faithfully simulate cooperative manipulation in high-fidelity software Drake:

- Rigid grasp modelling: Physics engines generally do not support closed kinematic chains. We therefore approximate the rigid grasps using linear bushings, which introduce mild compliance while maintaining approximate rigidity.
- Obstacle modelling in the Footprint Planner: As discussed in Section 3.2, each 2D obstacle is approximated by a super-ellipse. This avoids mixed-integer formulations that are incompatible with the solver SNOPT when used alongside nonlinear constraints. Figure 6 (left column) shows these approximations.
- AutoDiff for collision penalties: The collision cost  $c_{\text{collision}}$  requires gradients of signed distances between all robot and obstacle geometries. Drake only provides these gradients through AutoDiff models. Thus, the entire system (including three robots and the grasped object) is converted to AutoDiff to evaluate the smooth collision penalty.
- Geometric simplification of robots: For collision checking, the manipulators' collision geometry is approximated using bounding boxes, as Drake currently does not provide signed distances for meshes or convex hulls coupled with AutoDiff. These boxes are sufficient for computing smooth distance-based penalties used inside the IK problem.
- Selective activation of collision cost: The collision penalty is activated only when the object enters the vicinity of an obstacle. This reduces unnecessary computation during long collision-free segments of the trajectory.

#### 4.3. Results

Figure 6 shows the MAPS<sup>2</sup> object trajectory (top) and the resulting base trajectories from the *Footprint Planner* (bottom), together with their 2D obstacle approximations. In all cases, the bases successfully respect the super-ellipsoidal constraints.

Figure 7d displays the error between the tracked base positions and their planned trajectories. A persistent but bounded error arises from the relaxation allowed in the IK problem, which balances grasp maintenance, obstacle avoidance, and joint-limit feasibility.

The collision cost evolution is illustrated in Fig. 7a. Elevated values occur only when the object passes near obstacles, matching the regions where the collision cost is active. Object tracking performance is

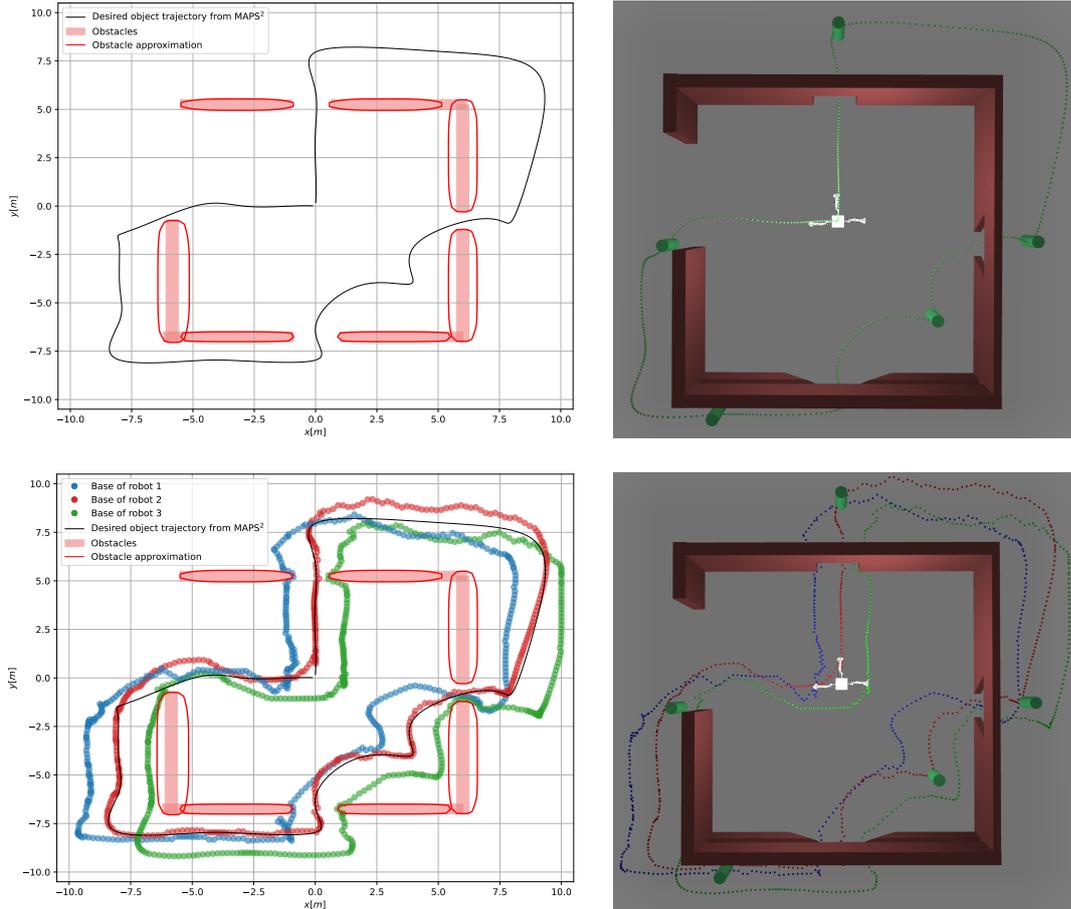


Figure 6: Desired trajectory from MAPS<sup>2</sup> along with the solution to (12)

shown in Figs. 7b and 7c, demonstrating that the manipulators successfully carry the object along the STL-specified path through all narrow passages.

Finally, Fig. 8 presents snapshots of the robots navigating the square, triangular, and trapezoidal passages, as well as avoiding an obstacle. These visualisations confirm that the framework coordinates the three arms effectively through tight environments while maintaining the rigid grasp constraints and avoiding collisions.<sup>1</sup>

## 5. Conclusion

This work proposed a hybrid, multi-rate control framework for cooperative manipulation under spatio-temporal task specifications and environmental constraints. The approach combines offline discrete planning of an STL-satisfying object trajectory and collision-free base footprints with online constrained inverse-kinematics updates and continuous-time feedback control. The resulting closed-loop system integrates discrete planning decisions with continuous nonlinear dynamics, enabling coordinated reconfiguration of multiple mobile manipulators in response to narrow passages and obstacles. High-fidelity simulations with three Panda mobile manipulators demonstrate robust trajectory tracking, collision avoidance, and successful execution of hybrid tasks in constrained environments.

## References

- [1] O. Madsen, S. Bøgh, C. Schou, R. S. Andersen, J. S. Damgaard, M. R. Pedersen, V. Krüger, Integration of mobile manipulators in an industrial production, *Industrial Robot: the international journal of robotics research and application* 42 (1) (2015) 11–18. doi:10.1108/IR-09-2014-0390. URL <https://doi.org/10.1108/IR-09-2014-0390>

<sup>1</sup>The video of the simulations can be found here: <https://youtu.be/9W4w0zR15dg>

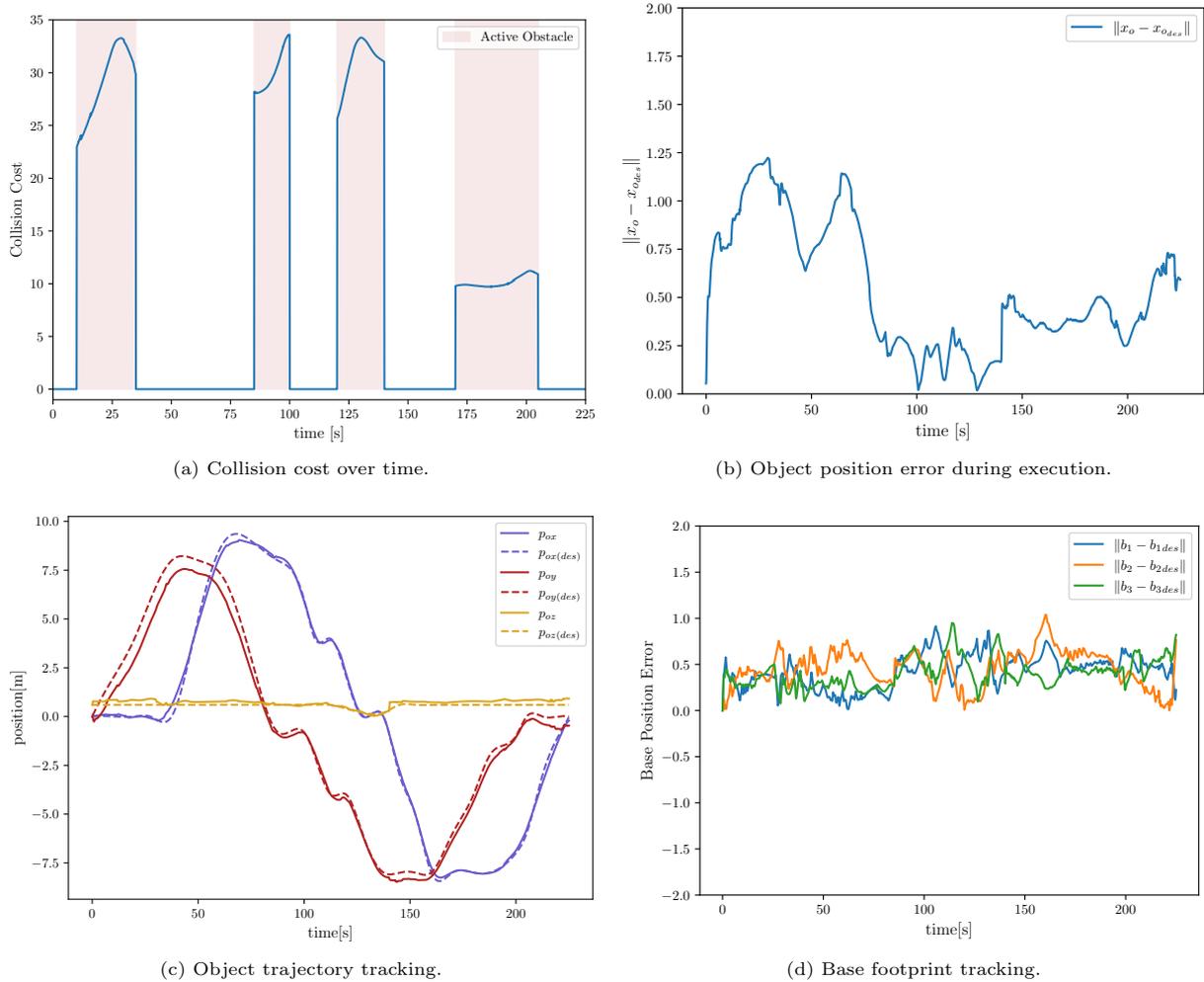
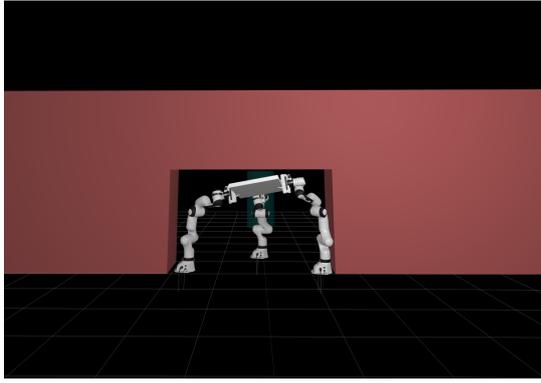
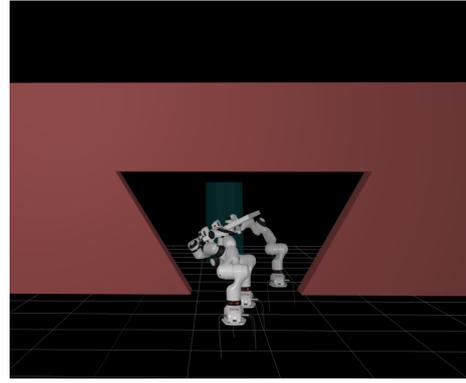


Figure 7: Tracking errors of object and footprint over time

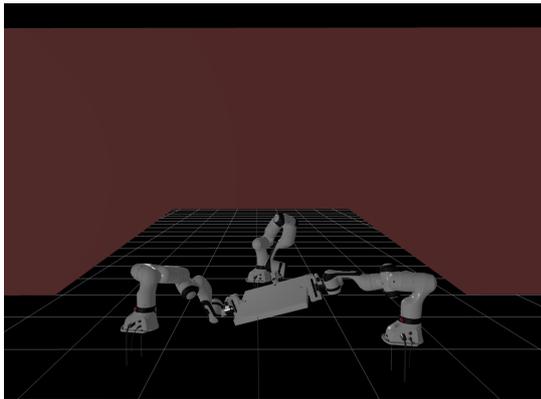
- [2] O. C. Nkwazema, L. Yin, M. Janneh, Investigation and application of robotics and automation system for construction machinery and environmental monitoring, *North American Academic Research* 4 (7) (2021) 120–130.
- [3] R. R. Murphy, S. Tadokoro, D. Nardi, A. Jacoff, P. Fiorini, H. Choset, A. M. Erkmen, *Search and Rescue Robotics*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2008, pp. 1151–1173. doi: 10.1007/978-3-540-30301-5\_51. URL [https://doi.org/10.1007/978-3-540-30301-5\\_51](https://doi.org/10.1007/978-3-540-30301-5_51)
- [4] M. Sewlia, C. K. Verginis, D. V. Dimarogonas, Maps2: Multi-robot autonomous motion planning under signal temporal logic specifications, *The International Journal of Robotics Research* (2025) 02783649251393343doi:10.1177/02783649251393343. URL <https://doi.org/10.1177/02783649251393343>
- [5] T. Sandakalum, M. H. Ang, Motion planning for mobile manipulators—a systematic review, *Machines* 10 (2) (2022). doi:10.3390/machines10020097. URL <https://www.mdpi.com/2075-1702/10/2/97>
- [6] K. Gochev, A. Safonova, M. Likhachev, Planning with adaptive dimensionality for mobile manipulation, in: *2012 IEEE International Conference on Robotics and Automation*, 2012, pp. 2944–2951. doi:10.1109/ICRA.2012.6225228.
- [7] A. K. Singh, K. M. Krishna, Coordinating mobile manipulator’s motion to produce stable trajectories on uneven terrain based on feasible acceleration count, in: *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013, pp. 5009–5014. doi:10.1109/IRoS.2013.6697080.



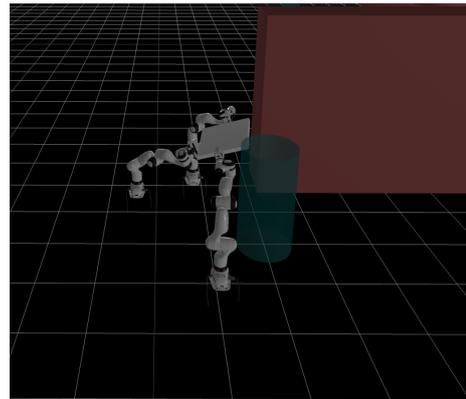
(a) Passing through a square shape.



(b) Passing through a triangle shape.



(c) Passing through trapezoid shape.



(d) Avoiding obstacle.

Figure 8: Snapshots of a simulation run.

- [8] J. Su, W. Xie, Motion planning and coordination for robot systems based on representation space, *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 41 (1) (2011) 248–259. doi:10.1109/TSMCB.2010.2051025.
- [9] T. Welschhold, C. Dornhege, F. Paus, T. Asfour, W. Burgard, Coupling mobile base and end-effector motion in task space, in: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 1–9. doi:10.1109/IROS.2018.8593534.
- [10] P. S. Schmitt, W. Neubauer, W. Feiten, K. M. Wurm, G. V. Wichert, W. Burgard, Optimal, sampling-based manipulation planning, in: *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 3426–3432. doi:10.1109/ICRA.2017.7989390.
- [11] F. Lamiroux, J. Mirabel, Prehensile manipulation planning: Modeling, algorithms and implementation, *IEEE Transactions on Robotics* 38 (4) (2022) 2370–2388. doi:10.1109/TR0.2021.3130433.
- [12] F. Burget, M. Bennewitz, W. Burgard, Bi2rrt\*: An efficient sampling-based path planning framework for task-constrained mobile manipulation, in: *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 3714–3721. doi:10.1109/IROS.2016.7759547.
- [13] N. Ratliff, M. Zucker, J. A. Bagnell, S. Srinivasa, Chomp: Gradient optimization techniques for efficient motion planning, in: *2009 IEEE International Conference on Robotics and Automation*, 2009, pp. 489–494. doi:10.1109/ROBOT.2009.5152817.
- [14] T. Marcucci, J. Umenberger, P. Parrilo, R. Tedrake, Shortest paths in graphs of convex sets, *SIAM Journal on Optimization* 34 (1) (2024) 507–532. doi:10.1137/22m1523790. URL <http://dx.doi.org/10.1137/22M1523790>
- [15] T. Marcucci, P. Nobel, R. Tedrake, S. Boyd, Fast path planning through large collections of safe boxes, *IEEE Transactions on Robotics* 40 (2024) 3795–3811. doi:10.1109/TR0.2024.3434168.

- [16] J. Schulman, Y. Duan, J. Ho, A. Lee, I. Awwal, H. Bradlow, J. Pan, S. Patil, K. Goldberg, P. Abbeel, Motion planning with sequential convex optimization and convex collision checking, *The International Journal of Robotics Research* 33 (9) (2014) 1251–1270. [arXiv:https://doi.org/10.1177/0278364914528132](https://arxiv.org/abs/1310.7530), [doi:10.1177/0278364914528132](https://doi.org/10.1177/0278364914528132).  
URL <https://doi.org/10.1177/0278364914528132>
- [17] W. S. Cortez, C. K. Verginis, D. V. Dimarogonas, A distributed, event-triggered, adaptive controller for cooperative manipulation with rolling contacts, *IEEE Transactions on Robotics* 39 (4) (2023) 3120–3133. [doi:10.1109/TR0.2023.3268595](https://doi.org/10.1109/TR0.2023.3268595).
- [18] C. K. Verginis, D. V. Dimarogonas, Timed abstractions for distributed cooperative manipulation, *Autonomous Robots* 42 (4) (2018) 781–799. [doi:10.1007/s10514-017-9672-7](https://doi.org/10.1007/s10514-017-9672-7).  
URL <https://doi.org/10.1007/s10514-017-9672-7>
- [19] H. Zhang, S.-H. Chan, J. Zhong, J. Li, P. Kolapo, S. Koenig, Z. Agioutantis, S. Schafrik, S. Nikolaidis, Multi-robot geometric task-and-motion planning for collaborative manipulation tasks, *Autonomous Robots* 47 (8) (2023) 1537–1558. [doi:10.1007/s10514-023-10148-y](https://doi.org/10.1007/s10514-023-10148-y).  
URL <https://doi.org/10.1007/s10514-023-10148-y>
- [20] P. Vlantis, C. P. Bechlioulis, K. J. Kyriakopoulos, Mutli-robot cooperative object transportation with guaranteed safety and convergence in planar obstacle cluttered workspaces via configuration space decomposition, *Robotics* 11 (6) (2022). [doi:10.3390/robotics11060148](https://doi.org/10.3390/robotics11060148).  
URL <https://www.mdpi.com/2218-6581/11/6/148>
- [21] K. Patra, A. Sinha, A. Guha, Kinodynamic motion planning for collaborative object transportation by multiple mobile manipulators (2024). [arXiv:2409.14910](https://arxiv.org/abs/2409.14910).  
URL <https://arxiv.org/abs/2409.14910>
- [22] R. M. Murray, S. S. Sastry, L. Zexiang, *A Mathematical Introduction to Robotic Manipulation*, 1st Edition, CRC Press, Inc., USA, 1994.
- [23] S. Haddadin, The franka emika robot: A standard platform in robotics research, *IEEE Robotics & Automation Magazine* (2024).
- [24] O. Maler, D. Nickovic, Monitoring Temporal Properties of Continuous Signals, in: Y. Lakhnech, S. Yovine (Eds.), *Formal Techniques, Modelling and Analysis of Timed and Fault-Tolerant Systems*, *Lecture Notes in Computer Science*, Springer, Berlin, Heidelberg, 2004, pp. 152–166.
- [25] L. Lindemann, D. V. Dimarogonas, Control barrier functions for multi-agent systems under conflicting local signal temporal logic tasks, *IEEE Control Systems Letters* 3 (3) (2019) 757–762.
- [26] L. Schmid, O. Andersson, A. Sulser, P. Pfreundschuh, R. Siegwart, Dynablox: Real-time detection of diverse dynamic objects in complex environments, *IEEE Robotics and Automation Letters* 8 (10) (2023) 6259–6266. [doi:10.1109/LRA.2023.3305239](https://doi.org/10.1109/LRA.2023.3305239).
- [27] A. Peirce, Lecture notes on variational and approximate methods in applied mathematics, [https://personal.math.ubc.ca/~peirce/M406\\_2012\\_Lecture\\_4n5.pdf](https://personal.math.ubc.ca/~peirce/M406_2012_Lecture_4n5.pdf), accessed: December 2025 (2012).
- [28] R. Tedrake, the Drake Development Team, *Drake: Model-based design and verification for robotics* (2019).  
URL <https://drake.mit.edu>