

# Motion Planning of Nonholonomic Cooperative Mobile Manipulators

Keshab Patra<sup>1</sup>, *Student Member, IEEE*, Arpita Sinha<sup>2</sup>, *Member, IEEE*, and Anirban Guha<sup>\*1</sup>

**Abstract**—We propose a real-time implementable motion planning technique for cooperative object transportation by nonholonomic mobile manipulator robots (MMRs) in an environment with static and dynamic obstacles. The proposed motion planning technique works in two steps. A novel visibility vertices-based path planning algorithm computes a global piece-wise linear path between the start and the goal location in the presence of static obstacles offline. It defines the static obstacle free space around the path with a set of convex polygons for the online motion planner. We employ a Nonlinear Model Predictive Control (NMPC) based online motion planning technique for nonholonomic MMRs that jointly plans for the mobile base and the manipulator’s arm. It efficiently utilizes the locomotion capability of the mobile base and the manipulation capability of the arm. The motion planner plans feasible motion for the MMRs and generates trajectory for object transportation considering the kinodynamic constraints and the static and dynamic obstacles. The efficiency of our approach is validated by numerical simulation and hardware experiments in varied environments.

**Index Terms**—Collaborative mobile manipulators, kinodynamic motion planning, collaborative object transportation, nonholonomic system, constrained optimization

## I. INTRODUCTION

Robotic systems became integral to automation in manufacturing, remote exploration, warehouse management, and other areas. Cooperative multiple MMRs garner attention due to their low cost, small size, redundancy in heavy or oversized object transportation, and fixture-less multipart assembly requiring more Degrees of Freedom (DoF). A cooperative MMR system extends workspace coverage, flexibility, and redundancy with added complexity in robot coordination, communication, and motion planning. Multiple MMRs leverage the mobile bases’ locomotion ability and the arms’ manipulation ability for object transportation and manipulation in a large workspace.

Nonholonomic mobile bases are widespread in robotic applications because of their advantages in a reduced number of actuators, simplified wheels, and better load-carrying capacity. It also works better on uneven ground surfaces. Introducing a nonholonomic mobile base for the MMRs includes non-integrable constraints on kinematics. The nonholonomic mobile base’s non-integrable kinematics constraints impact motion planning with its restricted side-wise motion capability. Hence, there are more intricacies in cooperative motion

<sup>1</sup>Keshab Patra and Anirban Guha are with the Department of Mechanical Engineering, Indian Institute of Technology Bombay, Powai, Mumbai-400076, India e-mail: keshabpatra19@gmail.com, anirbanguha1@gmail.com

<sup>2</sup>Arpita Sinha is with Systems & Control Engineering, Indian Institute of Technology Bombay, Powai, Mumbai-400076, India e-mail: arpita.sinha@iitb.ac.in

planning and trajectory generation than in the holonomic counterpart.

The study on collaborative manipulators started with a virtual linkage model [1] representing the collaborative manipulation systems to generate closed-chain constraints [2] between the object and the MMRs for motion synchronization and coordination. The dual arm cooperative control problem has been addressed by NMPC [3]. The coordination scheme for multi-MMR cooperative manipulation and transportation comprises of centralized [4], decentralized [5], [6] and distributed [7], [8], [9] control framework. Online task allocation [10] algorithm ensures efficient utilization of the capabilities of the cooperative manipulators. The collision-free navigation started with a variational-based method [11] that demonstrated static obstacle avoidance for a two MMR system with poor scalability. Dipolar inverse Lyapunov functions fused with the potential field-based navigation function [12] plan collision-free motion in static environments to transport deformable material by multiple MMRs with a little scope of formation control.

Constrained optimization-based motion planning technique [13] for holonomic MMRs in dynamic environments uses obstacle-free convex polygons around the formation in the position-time space. It optimizes the holonomic MMRs’ pose to retain the cooperative MMR system inside the obstacle-free polygon. A geometric path planning approach was proposed [14], [15] for multiple MMRs transporting an object for static obstacle avoidance. A rectangular passageway-based approach [14] is used to find the optimal system width and moving direction in the static obstacle-free area. These methods do not include motion constraints and guarantee feasible motion for nonholonomic MMRs.

A kinematic motion planning technique [16] plans for spatial collaborative payload manipulation using a hierarchical approach. The technique’s conservative approximation of the obstacles as uniform cylinders highly restricts navigation in tight spaces with high aspect ratios polygonal obstacles. MPC-based motion planning techniques for static obstacle avoidance have been presented in [17], [18]. An alternating direction method of multipliers-based distributed trajectory planning algorithm [19] plans trajectory in a static environment. A distributed formation control technique [20] utilizes constrained optimization for object transportation in a static environment. The formation moves along a predefined reference trajectory to avoid collision with the static obstacles. Motion Planning for deformable object transportation [21], [22] in a static environment uses optimization techniques. A reciprocal collision avoidance algorithm [23] combined with MPC for multiple

robots does not maintain any formation. These generic planning algorithms cannot be used as they do not maintain the rigid formation required for collaborative MMRs. An NMPC-based kinodynamic motion planning technique [24] plans motion for object transportation by multiple MMRs in static and dynamic environments. The proposed planning technique is limited to holonomic MMRs and environments with convex static obstacles.

We proposed a motion planning technique for collaborative object transportation using nonholonomic MMRs that eliminates the shortcomings. The proposed planning technique removes the restriction of convex obstacles and works with any polygonal static obstacles. The planning technique works in two steps: offline path planning and online motion planning. In offline path planning, the planner computes the shortest feasible piece-wise linear path between the start and the goal using visibility vertices [25] based technique considering the static obstacle. After computing the path, the convex polygon around path segments is computed for online motion planning. The online motion planner computes a kinodynamic feasible motion plan for the MMRs while transporting the object in a dynamic environment, taking the global path as an initial guess. The major contributions of this article are in the following:

- 1) A novel visibility vertices-based offline path planning algorithm finds the shortest path in static environments with any convex and concave polygonal shape static obstacles. The algorithm is capable of finding paths through narrow corridors.
- 2) A convex polygon computation algorithm to compute convex polygons around the path segments utilizing visible vertices of the path segments.
- 3) An NMPC-based online motion planning technique jointly plans for nonholonomic MMR's base and the manipulator. The planner computes kinodynamic feasible collision-free motion for the multiple MMRs in a dynamic environment.

## II. PROBLEM FORMULATION

A system of  $n$  nonholonomic MMRs grasps an object at its periphery as shown in Fig. 1 to collaboratively transport an object from a start location to a goal location in the obstacle-free region.  $\{w\}$  defines the world fixed reference frame. An object coordinate frame  $\{o\}$  is attached to the object center of mass (CoM), and each MMR has its own body coordinates  $\{b_i\}$  attached to the center of its mobile base. Without specific mention, all the quantities are defined in the world fixed frame  $\{w\}$ . The collaborative manipulation system is defined in the following subsections.

### A. Mobile Manipulator

The mobile base of  $i^{th}$  MMR is defined with pose  $q_{m,i} = [p_i^T, \phi_i]^T$  where  $p_i \in \mathbb{R}^2$  and  $\phi_i \in \mathbb{R}$  are the position and orientation of the mobile base in  $\{w\}$ . The manipulator of MMR  $i$  has  $n_i$  number of joints. The joint states of the manipulator  $i$  is defined as  $q_{a,i} = [q_{a,i1}, q_{a,i2}, \dots, q_{a,in_i}]^T$ . The  $i^{th}$  EE's position and orientation is defined in  $\{w\}$  as

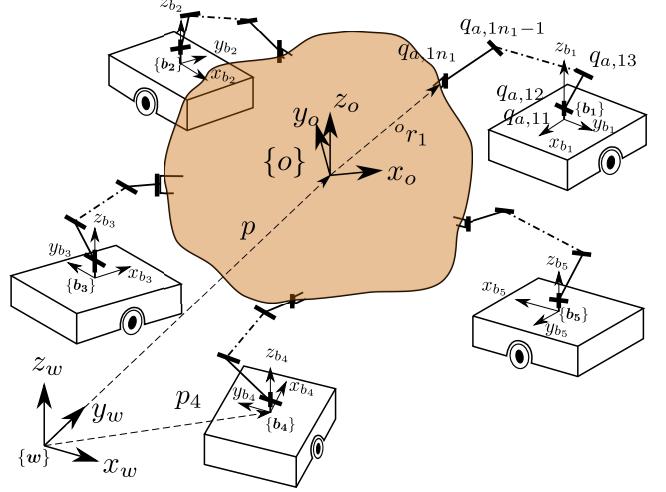


Fig. 1. Formation of five non-holonomic MMRs holding an object. The MMRs grasped the object to transport collaboratively from one place to another.

$pe_{e,i} \in \mathbb{R}^3$  and  $\phi_{ee,i} \in \mathbb{R}^3$ . The joint state of the manipulator is defined by  $q_i = [q_{m,i}^T, q_{a,i}^T]^T$ . The  $i^{th}$  non-holonomic MMR's first-order dynamics  $\dot{q}_i = [v_i \cos(\phi_i), v_i \sin(\phi_i), \omega_i, \dot{q}_{a,i}]^T$  is considered where the control inputs are mobile base's linear and angular velocities  $v_i, \omega_i$  respectively and the manipulator's joint velocities  $\dot{q}_{a,i}$ , therefore,  $u_i = [v_i, \omega_i, \dot{q}_{a,i}]$ .

We represent the coupled first order system dynamics for  $i^{th}$  MMR by a discrete-time non-linear system as

$$q_i^{k+1} = f(q_i^k, u_i^k) \quad (1)$$

where  $k$  is the discrete time step. The admissible states and control inputs are defined by Eqn. 2

$$\underline{q}_{a,i} \leq q_{a,i} \leq \bar{q}_{a,i} \quad (2a)$$

$$\underline{u}_i \leq u_i \leq \bar{u}_i \quad (2b)$$

for all  $i \in [1, n]$ , where  $\underline{q}_{a,i}, \bar{q}_{a,i}$  represents the manipulator's joint position limit vector and  $\underline{u}_i, \bar{u}_i$  are the admissible control limits. The set of admissible states  $\mathcal{Q}_i$  and control inputs  $\mathcal{U}_i$  are indicated by joint position and velocity vectors' limit (Eqn. (2)),  $\mathcal{Q}_i = [\underline{q}_{a,i}, \bar{q}_{a,i}], \mathcal{U}_i = [\underline{u}_i, \bar{u}_i]$

### B. Collaborative Formation

The multi-MMR formation  $\mathcal{F}$  (Fig 1) of  $n$  MMRs grasps a rigid object by the EEs. The  $i^{th}$  EE grasps the object at  ${}^o r_i$ , where the superscript  $o$  indicates it's reference frame  $\{o\}$ . The formation configuration is defined by  $\mathcal{X} = [p^T, o^T, Q^T]^T$ , where  $p \in \mathbb{R}^3$  is the position and  $o \in \mathbb{R}^3$  is the orientation of the object CoM,  $Q = [q_1^T, q_2^T, \dots, q_n^T]^T$  is the configuration of  $n$  MMRs. The space occupied by the formation is defined as  $\mathcal{B}(\mathcal{X})$ .

### C. Environments

A structured and bounded environment having both static and dynamic obstacles is defined as  $W$ .  $\mathcal{O}$  represents the set of static obstacles in  $W$ . We consider the static obstacles to

be vertically upright. The static obstacle-free workspace is defined by

$$W_{free} = W \setminus \mathcal{O} \in \mathbb{R}^2 \quad (3)$$

The static obstacle map is known apriori. The set of dynamic obstacles within the sensing zone of the MMRs is defined as  $\mathcal{O}_{dyn}$ . The start position  $p_s$  and the goal position  $p_g$  are in the obstacle-free space  $W_{free}$ .

The planning objective is to design a motion planning framework for cooperative object transportation such that

- 1) the MMRs can cooperatively transport the object from the start to goal position without any collision.
- 2) the generated motion is kinodynamically feasible and within the admissible limits of the MMRs.
- 3) the planner can handle static concave obstacles directly.
- 4) the control input of the MMRs are minimized.

### III. MOTION PLANNING

We solve the motion planning problem for cooperative multi-MMRs in two steps: offline path planning and online motion planning shown in Fig. 2. In the offline path planning step, we compute a static obstacle-free linear piece-wise continuous path ( $S$ ) between the start and the goal location using our proposed offline path planner in Section III-A.

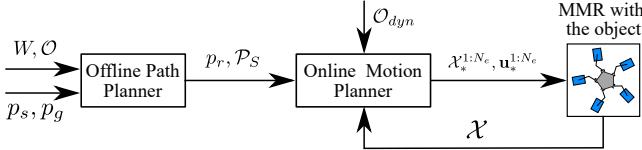


Fig. 2. Two step motion planning process: offline path planning and online motion planning.

The path planner smooths the generated path  $S$  to a continuous time normalized reference trajectory  $p_r(c_t)$  using cubic Bézier curve with normalized time parameter  $c_t \in [0, 1]$ . In the next step, an online motion planner (Section III-B) computes a feasible motion plan for the collaborative MMRs in receding horizons. The motion planner generates a kinodynamically feasible trajectory for object transportation in the dynamic environment using  $p_r(c_t)$  as an initial guess. The generated trajectory is free from collision with the static and dynamic obstacles and collision among the MMRs. The path planner smooths the generated path  $S$  to a continuous time normalized reference trajectory  $p_r(c_t)$  using cubic Bézier curve with normalized time parameter  $c_t \in [0, 1]$ . In the next step, an online motion planner (Section III-B) computes a feasible motion plan for the collaborative MMRs in receding horizons. The motion planner generates a kinodynamically feasible trajectory for object transportation in the dynamic environment using  $p_r(c_t)$  as an initial guess. The generated trajectory is free from collision with the static and dynamic obstacles and collision among the MMRs.

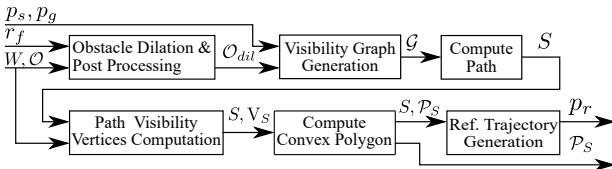


Fig. 3. Offline path planning process.

#### A. Global Path Planner

The global path planner computes a static obstacle-free path for the MMRs' base between the start and goal location in offline. It employs visibility vertices finding algorithm [25].

The post-processing step computes a convex obstacle-free polygon around the computed path segment. Fig. 3 shows the outline of the path planning process.

1) *Path Computation:* The global path planner dilates all the static obstacle  $\mathcal{O}$  by a distance  $r_f$  so that we consider the formation  $\mathcal{F}$  as a point at the CoM  $p$  of the object. The dilation distance  $r_f$  is the radius of a circle located at  $p$ , inside which the formation could always be enclosed. The planner substitutes the mutually intersecting dilated obstacles  $\mathcal{O}_{dil}$  with their union. A visibility vertices finding algorithm [25] creates a visibility map considering  $\mathcal{O}_{dil}$  and including the start and the goal. The vertices from the visibility map are added as node  $\mathcal{V}$  to a graph  $\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathcal{W})$ . An edge  $\mathcal{E}$  between two nodes is added if they are mutually visible to each other using the visibility map. The Euclidean distance between two nodes defines the weight  $\mathcal{W}$  of the edge connecting them. A graph search algorithm computes the shortest linear piecewise path  $S$  between the start  $p_s$  and the goal  $p_g$  location. Fig. 4 shows the computed path  $S$  with linear segments  $S_1, S_2, S_3, S_4$  and vertices  $w_1, w_2, w_3, w_4, w_5$ .

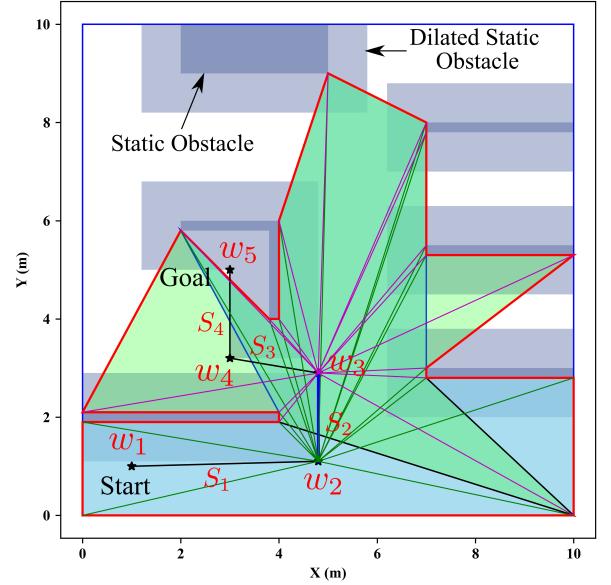


Fig. 4. Path Polygon for  $S_2$  computed using the visible vertices of  $w_2$  and  $w_3$ .

2) *Convex Polygon Computation:* Now, we consider the static obstacles without dilution and the vertices  $w_i$  of the path. We again compute the vertices  $V_s$  that are present in the static environment and visible from  $w_i$ , (Fig. 4). A simple polygon is defined for each vertex  $w_i$ ,  $\forall i$  by cyclically connecting its visible vertices. This polygon remains in the static obstacle-free region  $W_{free}$ . The union of polygons defined for  $w_i$  and  $w_{i+1}$  of a path segment  $S_i$  defines a static obstacle-free simple polygon around  $S_i \in S$ . Fig. 4 shows the computed polygon for the path segment  $S_2$  using the polygons around  $w_2$  and  $w_3$ . The union of the two polygons (green and blue) in red boundary defined around  $w_2$  and  $w_3$  provides a static obstacle-free polygon  $P_{cc,2}$  around  $S_2$ . A set of polygon  $\mathcal{P}_{cc}$  for all the path segments is computed similarly. The computed polygons are generally concave. Convexification is needed for

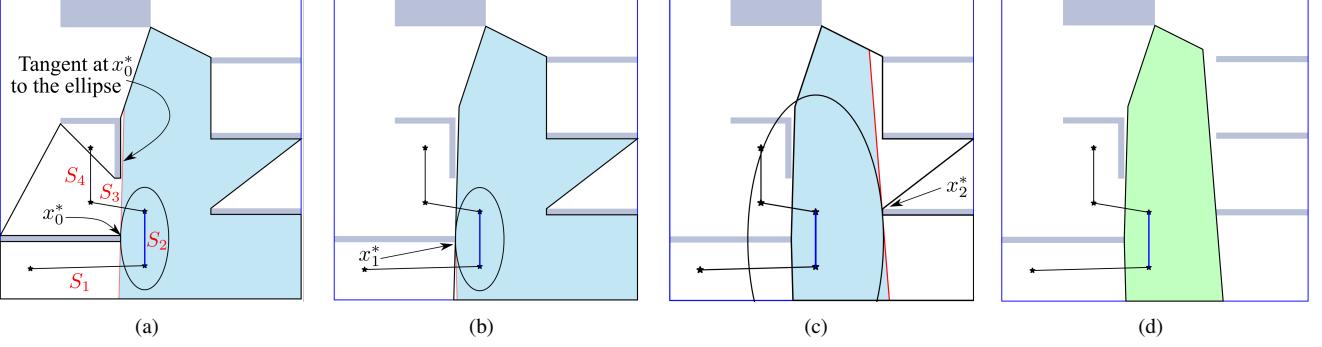


Fig. 5. The polygon convexification process step by step. For the path segment 2 in Fig. 5(a), an ellipse touching the nearest concave vertex of the polygon has been formed and a tangent (red line) to the ellipse at this point has been drawn. The tangent cuts the polygon bounded by black edges and the polygon (sky blue) containing the path segment has been kept. The ellipse has been dilated keeping the aspect ratio same in Fig 5(b) till it touches the nearest concave vertex of the new polygon. Here a very small portion of the polygon is cut by the tangent to the ellipse at this concave vertex. The process continues till any concave vertex remains and a convex polygon is formed (green polygon in Fig. 5(d).)

the static obstacle avoidance constraints in motion planning optimizations. We compute a set of convex polygons  $\mathcal{P}$  as a subset of their original concave polygons in  $\mathcal{P}_{cc}$  by ellipse fitting technique illustrated in the Algo. 1.

We define an ellipse in the ground plane as

$$\kappa(C, d) = \{x = C\bar{x} + d : \|\bar{x}\| \leq 1, x \in \mathbb{R}^2\} \quad (4)$$

where  $C$  is a  $2 \times 2$  symmetric positive definite matrix that maps the deformation of a unit radius circle ( $\|\bar{x}\| \leq 1$ ) to an ellipse.  $C$  is decomposed as  $C = R^T \Lambda R$ , where  $R$  is a rotation matrix that aligns the ellipse axes to the world reference frame axes and  $\Lambda = \text{diag}(a, b)$  is a diagonal scale matrix. The diagonal elements  $a$  and  $b$  of  $\Lambda$  refer to the length of the ellipse semi-major and minor axes.  $d$  defines the center of the ellipse.

Algo. 1 initialized a polygon  $P$  with the concave polygon  $P_{cc,i}$ . It finds the concave vertices  $\mathcal{V}_{cc}$  of  $P$ . If  $\mathcal{V}_{cc}$  is not empty, then it convexifies  $P$  in line 4–17. The algorithm computes the half-plane representation of  $P$  in line 19. In the polygon convexification process, the algorithm fits an ellipse  $\kappa_0$  center  $d$  at the midpoint of the path segment  $S_i \in S$ . The major axis is aligned with the path segment  $S_i$ , the semi-major axis length  $a = 0.5\text{length}(S_i) + r_f$ . The ellipse is inflated in the minor-axis direction till it touches the nearest concave points  $x_0^*$  to  $d$  and an ellipse  $\kappa_0$  is formed in line 4–8. The tangent to the ellipse  $\kappa_0$  at point  $x_0^*$  defines the inequality representation  $H_0 = \{x : a_0^T x \leq b_0\}$ . After obtaining  $H_0$  in line 9, we cut the polygon with  $H_0$  and keep the polygon inside the  $H_0$  as  $P$  (Fig. 5(a)). We discard the concave vertices outside  $H_0$  from  $\mathcal{V}_{cc}$ . If there is any concave vertices left  $\mathcal{V}_{cc} \neq \emptyset$  the ellipse  $\kappa_0$  is dilated to form an ellipse  $\kappa_1$  in line 14 keeping the aspect ratio same till it touches the nearest concave vertices  $x_1^* \in \mathcal{V}_{cc}$  to  $d$ . The tangent to the ellipse at point  $x_1^*$  defines the inequality  $H_1 = \{x : a_1^T x \leq b_1\}$ . After obtaining  $H_1$  in line 9, we cut the polygon with  $H_1$  and keep the polygon inside the  $H_1$  as  $P$  in Fig. 5(b). The convexification process in line 11–16 is repeated till there is no concave vertex left in the polygon  $\mathcal{V}_{cc} \neq \emptyset$  (Fig. 5(b)–5(d)). After eliminating all the concave vertices, the polygon  $P$  becomes convex (Fig. 5(d)), and the half-plane representation of  $P$  is returned in line 17.

---

#### Algorithm 1 Polygon Convexification

---

**Input:** Concave Polygon  $P_{cc,i}$ , Path segment  $S_i$   
**Output:** Convex Polygon  $P$

---

```

1:  $P \leftarrow P_{cc,i}; j \leftarrow 0$ 
2:  $\mathcal{V}_{cc} \leftarrow \text{ConcaveVertices}(P)$ 
3: if  $\mathcal{V}_{cc} \neq \emptyset$  then
4:    $d \leftarrow \text{Midpoint}(S_i); R$ : major axis along  $S_i$ 
5:    $a \leftarrow 0.5\text{length}(S_i) + r_f$ 
6:    $x_j^* \leftarrow \text{NearestVertex}(\mathcal{V}_{cc}, d)$ 
7:    $C_j, b \leftarrow \text{FindEllipse}(R, a, d, x_j^*)$      $\triangleright$  Use  $\|\bar{x}_j^*\| = 1$ 
8:    $\kappa_j \leftarrow (C_j, d)$ 
9:    $a_j \leftarrow 2C_j^{-T}C_j^{-1}(x_j^* - d); b_j \leftarrow a_j^T x_j^*$ 
10:   $P, \mathcal{V}_{cc} \leftarrow \text{DiscardVertices}(a_j, b_j, P, \mathcal{V}_{cc})$ 
11:  while  $\mathcal{V}_{cc} \neq \emptyset$  do
12:     $j = j + 1$ 
13:     $x_j^* \leftarrow \text{NearestVertex}(\mathcal{V}_{cc}, d)$ 
14:     $\kappa_j \leftarrow \text{DilateEllipse}(\kappa_0, x_j^*)$ 
15:    repeat line 9 – 10
16:  end while
17:   $P : \mathbf{A} \leftarrow [a_0^T, a_1^T, \dots]^T, \mathbf{b} \leftarrow [b_0, b_1, \dots]^T$ 
18: else
19:   $P : (\mathbf{A}, \mathbf{b}) \leftarrow \text{HalfPlanes}(P)$ 
20: end if
21: return  $P(\mathbf{A}, \mathbf{b})$ 

```

---

Fig. 6 shows a set of convex polygons  $\mathcal{P}$  in light green around the path segment  $S$  computed from  $P_{cc}$  using the Algo. 1. Every segment of the piece-wise linear path  $S$  remains inside any convex polygon  $P \in \mathcal{P}$  defined in  $W_{free}$ . We add intermediate control points in red dots in Fig. 6 in the path segment as additional control points on the path  $S$ . The control points are used to generate time-normalized smooth trajectory  $p_r(c_t)$  from the  $S$  using a Bézier curve with normalized time parameter  $c_t \in [0, 1]$ . These intermediate control points are inserted when a new convex polygon appears along the path  $S$  from the start point. A control point is added for the last path segment while exiting the intersection area of the last two polygons. The generated quadratic Bézier curve

would remain in  $W_{free}$  as any three consecutive control points remain within a single convex polygon. The time-normalized reference trajectory  $p_r(c_t)$  is used as an initial guess for the online motion planner.

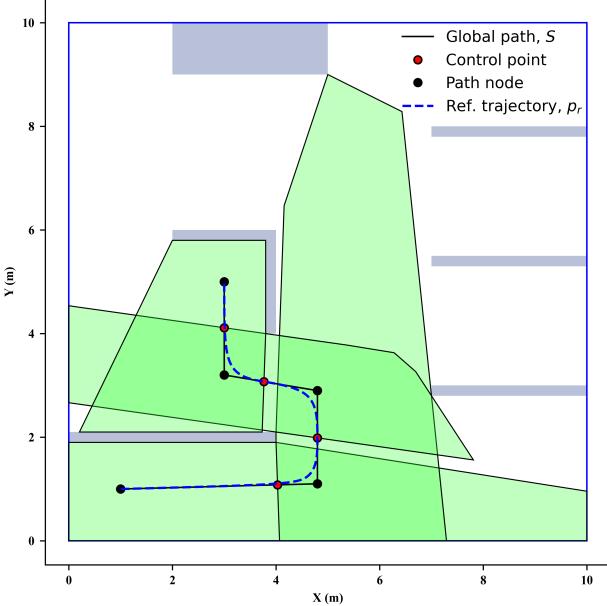


Fig. 6. Convex polygons around the path segments of  $S$ .

### B. Online Motion Planner

The global path planner in Section III-A computes a static obstacle-free path  $S$ . It does not delve into the motion constraints of the MMRs, dynamic obstacle avoidance, and the collision among the MMRs. We formulate an online motion planner as a constrained nonlinear optimization problem incorporating kinodynamic constraints. It uses the smoothed global reference trajectory  $p_r(c_t)$  as an initial guess. The motion planner computes the optimal motion for the MMRs by solving the optimization problem in Eqn. 5. The online motion planner solves the optimization problem in receding horizons for  $N_h$  horizon segment with time  $T_h$  to reduce the computational burden. The MMRs execute the computed motion plan of a horizon for an execution time  $T_e$  ( $T_e < T_h$ ). We choose the execution time  $T_e$  so that the computation time for a horizon is guaranteed to be less than  $T_e$ , and the MMRs can start the next plan once it finishes executing the current plan. The online motion planning optimization is given as

$$\mathcal{X}_*^{0:N_h}, \mathbf{u}^{0:N_h} = \arg \min \sum_{k=0}^{N_h-1} J(\mathcal{X}^k, u^k) + J_{N_h} \quad (5a)$$

$$\text{s.t. } q^{k+1} = f(q^k, u^k) \quad (5b)$$

$$\mathcal{B}(\mathcal{X}) \subset \mathcal{W}_{free} \quad (5c)$$

$$\mathcal{B}(\mathcal{X}) \cap \mathcal{O}_{dyn} = \emptyset \quad (5d)$$

$$H_i v_{<x,y>} \leq h_i, \quad (5e)$$

$$H_{(i+1)\%n} v_{<x,y>} \geq h_{(i+1)\%n}, \quad (5f)$$

$$0 \leq v_z \leq Z_h, \forall v \in \mathcal{V}_i(q_i), \forall i \in [1, n] \quad (5g)$$

$$q_{a,i}^k \in \mathcal{Q}_i, u_i^k \in \mathcal{U}_i, \forall i \in [1, n] \quad (5h)$$

$$g_i(\mathcal{X}) = 0, \forall i \in [1, n] \quad (5i)$$

$$\mathcal{X}^0 = \mathcal{X}(0) \quad (5j)$$

where the superscript  $k$  refers to the discrete time step. Section III-B1 illustrates the cost function in Eqn. (5a). Eqn. (5b) represents the state transition function (Eqn. (1)) of the system. Eqn. (5c) and (5d) account for the static and dynamic obstacles avoidance constraints detailed in Section III-B2 and Section III-B3. The self collision avoidance described in Section III-B4 is ensured by the constraints in the Eqn. (5e)-(5g). The set of admissible states and control inputs are defined in Eqn. (5h) and elaborated in Section III-B4. The grasp constraints described in Section III-B5) are maintained by Eqn. (5i). Eqn. (5j) defines the initial state of the formation in a planning horizon  $N_h$ .

*1) Cost Function:* The cost function described in Eqn. (6) for the motion planning optimization in Eqn. (5) minimizes the control inputs and the tracking error with respect to the reference trajectory. The diagonal weight-age matrix  $\mathbf{W}_u$  for control effort minimization is provided with higher value than the weight-age matrix  $\mathbf{W}_e$  to the tracking error  $e^k$  of the object CoM. The higher weight-age to control inputs prioritize optimal effort. The lower weight values to the tracking error provides global guidance to the trajectory with flexibility to deform for dynamic obstacle avoidance and kinodynamic motion compliance.

$$J(\mathcal{X}_k, u_k) = u^k{}^T \mathbf{W}_u u^k + e^k{}^T \mathbf{W}_e e^k \quad (6)$$

We discretize the reference path  $p_r(c_t)$  into a path segment. The expected and reference position for the CoM of the object is denoted as  $p^k$  and  $p_r^{\lambda+k} = [x, y]_r^{\lambda+k}$  for the future time step  $k$  and  $\lambda$  is the index of the nearest reference path segment to  $p^0$ . The discretized path should hold the relation  $\sum_{k=0}^{N_h-1} \|p_r^{\lambda+k+1} - p_r^{\lambda+k}\| \leq v_{op} T_h$ , where  $v_{op}$  is the operational velocity of the formation. The tracking error vector  $e^k$  is defined as

$$e^k = p^k - p_r^{\lambda+k} \quad (7)$$

The terminal cost  $J_{N_h}$  is defined in Eqn. (8) similar to the tracking error with a higher weighting  $\mathbf{W}_{N_h}$ .

$$J_{N_h} = e^{N_h T} \mathbf{W}_{N_h} e^{N_h} \quad (8)$$

2) *Static Obstacles Avoidance*: The formation  $\mathcal{F}$  must remain within  $W_{free}$  ( $\mathcal{B}(\mathcal{X}) \subset \mathcal{W}_{free}$ ) to avoid collision with static obstacles.  $W_{free}$  is represented by a set of convex polygons  $\mathcal{P}$  computed by the offline path planner in Section III-A2. The inequality representation of the polygon  $P \in \mathcal{P}$  is in Eqn. (9).

$$P = \{x \in \mathbb{R}^2 : Ax \leq b, A \in \mathbb{R}^{n_f \times 2}, b \in \mathbb{R}^{n_f}\} \quad (9)$$

where  $n_f$  is the number of the sides of  $P$  and  $x$  is an interior point of  $P$ . The set of vertices of the bounding polygons of the object and the  $n$  MMRs are represented by  $\mathcal{V}(\mathcal{X})$ . The projection of  $\mathcal{V}(\mathcal{X})$  at the ground plane ( $z$ ) must remain within any polygon  $P$ . The constraints are represented as follows

$$Av_{<x,y>} \leq b - d_{safe}, \forall v \in \mathcal{V}(\mathcal{X}), P : (A, b) \quad (10)$$

where  $v_{<x,y>}$  is the  $x - y$  projection of the vertex  $v \in \mathcal{V}(\mathcal{X})$  defined in  $\{w\}$ .  $d_{safe}$  is the safety distance. The number of constraints in Eqn. ((10)) increases the computational complexity significantly. The problem can be simplified further by considering the bounding circles of the projected vertices of the MMR base, manipulators, and the object. This collision geometry reduces the number of constraints and the computational complexity. We have implemented circumscribing bounding circles for each MMR base, manipulator, and object in the ground plane. The center of the circles for the  $i^{th}$  MMR base, manipulator, and the object in the ground are located at  $p_{base,i}$ ,  $p_{arm,i}$  and  $p_{obj,i}$  with radius  $r_{base,i}$ ,  $r_{arm,i}$  and  $r_{obj,i}$ . The cyan, purple, and gray area in Fig. 7 shows the circumscribing circles for the  $i - th$  MMR base, manipulator, and object. The reduced static collision avoidance constraints are in the Eqn. (11)

$$\begin{aligned} Ap_{m,i}^k &\leq b - d_{safe} - r_{m,i}, P : (A, b) \\ \forall m \in \{base, obj, arm\}, \forall i \in [1, n], \forall k \in [1, N_h] \end{aligned} \quad (11)$$

where  $d_{safe} \in \mathbb{R}$  is the safety distance.

3) *Dynamic Obstacle Avoidance*: The formation  $\mathcal{F}$  must not collide with any of the dynamic obstacles  $\mathcal{O}_{dyn}$ . The space occupied by the formation  $\mathcal{B}(\mathcal{X})$  should not overlap with the dynamic obstacles i.e.  $\mathcal{B}(\mathcal{X}) \cap \mathcal{O}_{dyn} = \emptyset$ . We implement the dynamic obstacle avoidance by introducing a nonlinear constraint between the collision geometries of the formations and the dynamic obstacles. We estimate the state of the dynamic obstacles at the beginning of a planning horizon  $N_h$  and consider the obstacle velocity  $v_{dyn,d}$  remain the same during  $T_h$ , where  $d \in [1, n_{dyn}]$  represents the dynamic obstacles' ( $\mathcal{O}_{dyn}$ ) index and  $n_{dyn}$  represents the number of dynamic obstacle sensed at the beginning of planning horizons. The collision geometry of the dynamic obstacles are considered as circles with radius  $r_{dyn,d}$ ,  $\forall d \in [1, n_{dyn}]$  located at  $p_{dyn,d}^k, \forall k \in [1, N_h]$  in the ground plane. The positions  $p_{dyn,d}^k = p_{dyn,d} + v_{dyn,d}kT_c$ ,  $\forall d, \forall k \in [1, N_h]$  is obtained with the positions  $p_{dyn,d}$  and velocity  $v_{dyn,d}$  estimated at the beginning of each planning horizon.

The same collision geometry defined for the static obstacle avoidance in Section III-B2 for the base, manipulator of

the MMRs, and the object are utilized here. The nonlinear constraints defined in Eqn. (12) ensure that the obstacle does not intersect with the formation.

$$\begin{aligned} \|p_{dyn,d}^k - p_m^k\| &\geq r_{dyn,d} + r_m + d_{safe} \\ \forall m \in \{base, obj, arm\}, \forall d \in [1, n_{dyn}], \forall k \in [1, N_h] \end{aligned} \quad (12)$$

4) *Self Collision Avoidance*: For collision avoidance with the object and the other MMRs, the  $i^{th}$  MMR needs to be within the convex wedge shown in Fig. 7 defined by two infinite vertical planes  $\mathcal{H}_i$  and  $\mathcal{H}_{(i+1)\%n}$  as shown in Fig. 7. The convex wedge specifies the workspace for the  $i^{th}$  MMR free from movements of the neighboring MMRs. The planes  $\mathcal{H}_i$  and  $\mathcal{H}_{(i+1)\%n}$  can be defined dynamically considering the admissible states and workspace of MMRs' base fixing the object. Here, we have equally divided the space around the periphery of the object, starting at the CoM of the object for each MMR, as the grasping point is equispaced. The vertical plane  $\mathcal{H}_i$  is defined as follows.

$$\begin{aligned} \mathcal{H}_i = \{x \in \mathbb{R}^2 : H_i x \leq h_i, H_i \in \mathbb{R}^{1 \times 2}, h_i \in \mathbb{R}\} \\ 0 \leq z \leq +\infty \end{aligned} \quad (13)$$

The self-collision avoidance for the  $i^{th}$  MMR is defined in the following

$$H_i v_{<x,y>} \leq h_i, H_{(i+1)\%n} v_{<x,y>} \geq h_{(i+1)\%n} \quad (14)$$

where  $\mathcal{V}_i(q_i)$  is the set of vertices of the  $i^{th}$  MMR.

5) *Grasp Constraints*: The grasp of the object by EE should remain at the same pose throughout the task to ensure stable formation. The grasp constraint for  $i$ -th MMR is defined as

$$p_{ee,i} = p + {}_o^w R(\psi) {}^o r_i \quad (15a)$$

$$\phi_{ee,i} = \psi \quad (15b)$$

where  $p_{ee,i}$  is  $i$ -th MMR's EE position,  ${}_o^w R$  is the rotation matrix between the object frame at its CoM and the global frame, and  ${}^o r_i$  is the position of the  $i$ -th EE defined in the object frame. We represent Eqn. (15) by  $g_i(\mathcal{X}) = 0$ .

#### IV. OBJECT TRANSPORTATION

We have validated the proposed kinodynamic motion planning algorithm in simulation and hardware experiments with the nonholonomic MMRs that accept velocity as a control input. The first order system dynamics for the nonholonomic MMR are approximated using the fourth-order Runge-Kutta method as a state transition function mentioned in Eqn. (1). The NMPC problem of the local motion planning and the nonlinear optimization of global planning is solved using the CasADi package [26] with an Interior point optimization (Iopt) method.

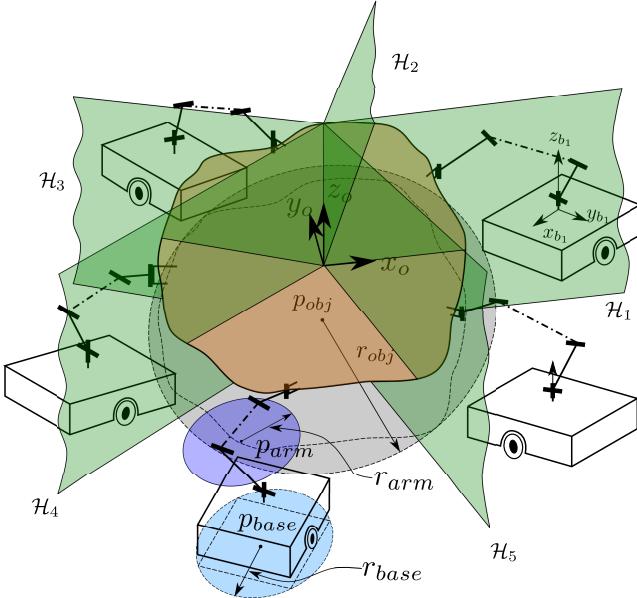


Fig. 7. The infinite convex wedge for  $i$ -th MMR is defined by the half plane  $\mathcal{H}_i$ ,  $\mathcal{H}_{(i+1)\%n}$ ,  $z = 0$  and  $z = \infty$ . The enclosing circles for MMRs' mobile base and manipulator are blue and violet, respectively, and the object is gray.

TABLE I  
DH PARAMETERS VALUE FOR THE MANIPULATORS

Joint	$d$ (m)	$a$ (m)	$\alpha$ (rad)	$\theta$ (rad)
Joint 1	0.070	0	0	$q_{a,1}$
Joint 2	0	0	$0.5\pi$	$q_{a,2}$
Joint 3	0.100	0	$-\pi$	$q_{a,3}$
Joint 4	0.125	0	$\pi$	$q_{a,4}$
Joint 5	0	0.120	$-0.5\pi$	$q_{a,5}$
Gripper	0	0	0	0

### A. Simulation

The MMRs with a differential drive mobile base have the same forwarding and reversing capabilities. The manipulator's Denavit-Hartenberg (DH) parameters are mentioned in Table I. The MMRs rigidly grasp the object at the periphery, and their grasping location remains the same throughout the entire task.

We select the operational velocity of the formation  $v_{op} = 0.15 \text{ m/s}$  and use prediction horizon time  $T_h = 9 \text{ s}$ , trajectory execution time  $T_e = 3 \text{ s}$  and the discretization time step  $T_c = 0.25 \text{ s}$ . The safety margins  $d_{safe} = 0.05 \text{ m}$  and  $d_{safe,dyn} = 0.1 \text{ m}$  for static and dynamic obstacle avoidance to keep the formation safe during object transportation. A higher margin restricts the formation from nearing the obstacles and hence reduces the obstacle-free space. The optimization weights are  $\mathbf{W}_u = diag([0.05, 0.05, 5, 0.5, 5, 0.05, 0.05, 5, 0.5, 5])$ ,  $\mathbf{W}_e = diag([0.01, 0.01])$  and  $\mathbf{W}_{N_h} = 10^5$ .

The five MMRs grasp an object at its periphery and start transporting (Fig. 8(a)) through a narrow corridor of  $1.9 \text{ m}$ . While the MMRs come out of the corridor, it encounters dynamic obstacles in Fig. 8(b), while taking a sharp left turn. The generated motion plan successfully navigates the formation, avoiding dynamic obstacles, and turns toward (Fig.

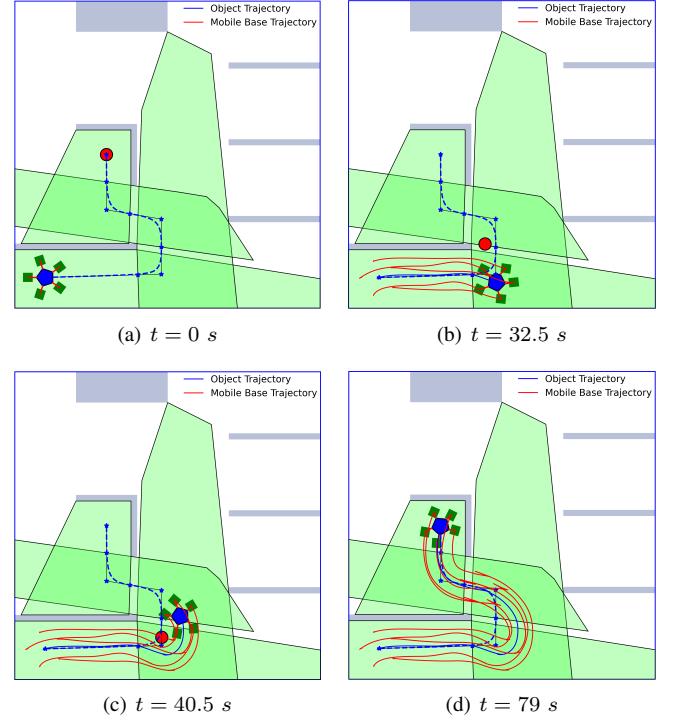


Fig. 8. The snapshots of cooperative object transportation by five MMRs (the MMR base in deep green and manipulator's arm in red line) in  $10 \text{ m} \times 10 \text{ m}$  environment. The red circular like dynamic obstacle is in its current state. The green convex polygons represents the static obstacle free region around the path. The object is being transported from the formation in Fig. 8(a) to the formation in Fig. 8(d).

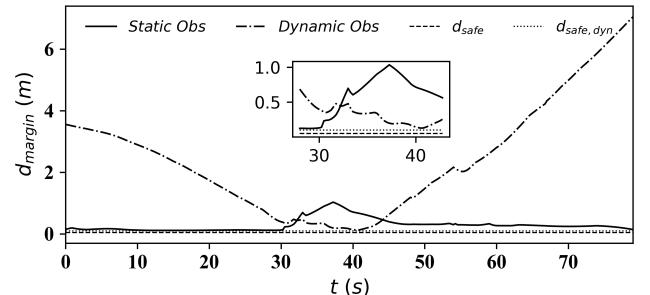


Fig. 9. Safety margin during object transportation through the narrow passages. The horizontal lines plots safety margins  $d_{safe} = 0.05 \text{ m}$  and  $d_{safe,dyn} = 0.1 \text{ m}$  is for static and dynamic obstacles.

8(c)) the goal. The MMRs successfully transport the object through the narrow doors and complete the task without any collision (Fig. 8(d)). Fig. 9 plots the shortest distance  $d_{margin}$  from the formation to any static and dynamic obstacles. The  $d_{margin}$  in Fig 9 for the static and the dynamic obstacles being always positive indicates successful collision avoidance behavior of the proposed motion planning techniques.

### B. Hardware Experiments

We perform experiments with our in-house developed ROS-enabled MMRs to evaluate the motion planning algorithm in Section III-B in an environment ( $4 \text{ m} \times 4 \text{ m}$ ) with static and dynamic obstacles. The nonholonomic MMR bases are made

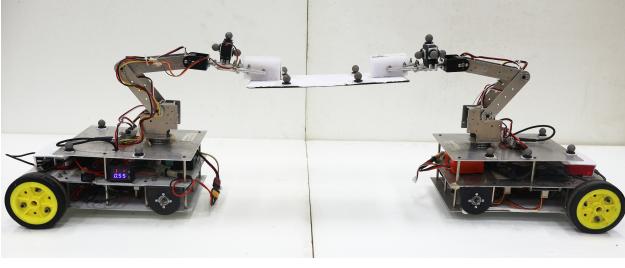


Fig. 10. Experimental Setup of two in-house developed nonholonomic MMRs.

of two disc wheels each separately driven by geared motor with an encoder. Fig. 10 shows two nonholonomic MMRs both grasped an object to transport it in an indoor environment shown in Fig. 11(a).

The manipulator of the MMRs shown in Fig. 10 is same as the manipulator used for the simulation, described in Table I, except for the joint 5. We fixed the joint 5 because of the hardware limitations. The adjusted DH parameters of the gripper are  $d = 0.120\text{ m}$ ,  $a = 0$ ,  $\alpha = 0$ , and  $\theta = 0$  after fixing joint 5. The planned trajectory and the control input for the MMRs by the online motion planner (Section III-B) are sent to the respective MMRs. The trajectory tracking controllers for the mobile base and manipulator ensure desired trajectory tracking. The trajectory of each mobile base, object, and the EE of MMRs are measured using a Vicon motion capture system.

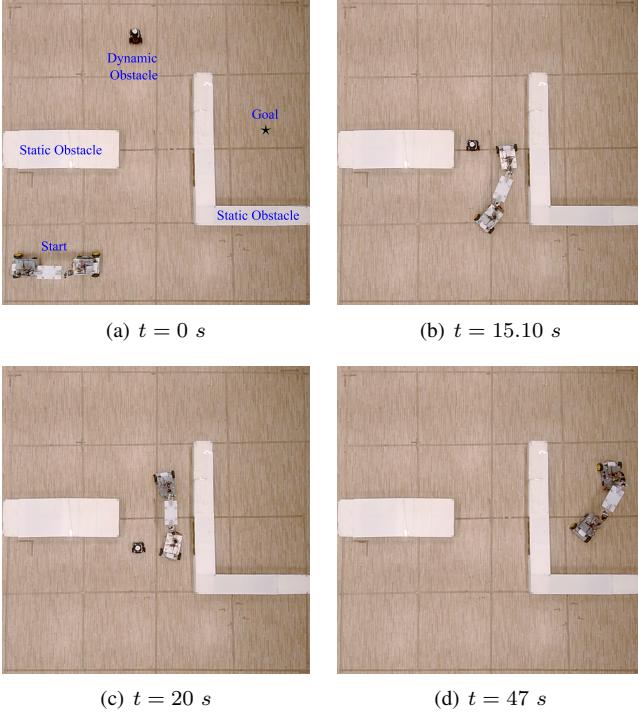


Fig. 11. Two MMRs transport the rectangular object. The MMRs encounter a dynamic obstacle and started avoidance maneuver (Fig. 11(b)). It successfully avoids the dynamic obstacle 11(c)) and reaches the goal point 11(d))

Fig. 11 shows the snap of the object transport from the start (Fig. 11(a)) to the goal (Fig. 11(d)). It encounters a dynamic

obstacle and start avoidance maneuver. Fig. 11(b) shows when the formation approaches the dynamic obstacle and finally avoids (Fig. 11(c)) the obstacle to reach the goal (Fig. 11(d)).

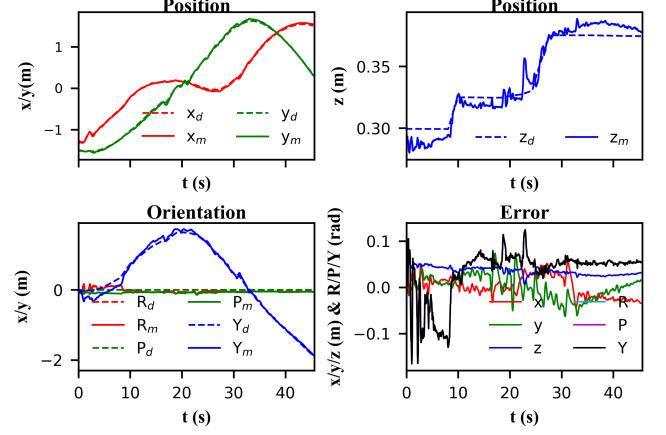


Fig. 12. Trajectory of the CoM of the object. The subscript d and m of the legend represents the planned and actual values.

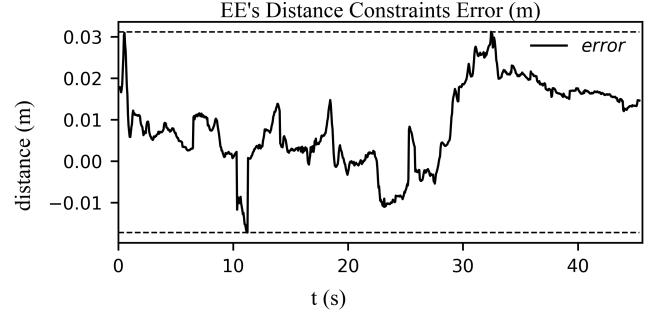


Fig. 13. The distance between the two EE during the object transportation.

Fig. 12 shows the planned and the actual trajectory of the CoM of the transported object. The position error remains within  $0.05\text{ m}$ , and the orientation error remains within  $0.15\text{ rad}$ . The  $z$  height of the object's CoM is plotted separately ( $z$  Vs  $t$  plot) in Fig. 12 to better understand the object movement in 3D. The  $z$  height increases near  $t = 10\text{ s}$  and  $t = 28\text{ s}$  before taking sharp turn to reduce the inter robot distance and turning radius. The error in fixed distance between the EEs' grasping point in Fig. 13 shows that the coordination between the MMRs has been maintained.

### C. Comparison

We compare the computational time of our proposed online motion planning technique with the holonomic MMRs' planning algorithm proposed by *Keshab et al.* [24] and *Alonso-Mora et al.* [13] in Table II. All three algorithms have been implemented in Python for motion planning of two MMRs transporting an object cooperatively in an environment with dynamic obstacles. The MMRs are the same except for the base motion constraints: nonholonomic and holonomic. We have run the planning algorithm's computation study on a Laptop equipped with AMD Ryzen 5800H CPU and 16 GB RAM. The computation time measurements are taken

for online local planning of each horizon from twenty-three cases in three different environments with varied goals and obstacle velocities. The details of the cases are available in the [attached file](#). We have also considered two dynamic obstacle cases. The results have been included in the attached [video](#). The computation time in Table II is slightly lower than the algorithm in [13] and higher than the algorithm in [24] due to its planning complexity arises because of the nonholonomic constraints of the MMRs base. The nonholonomic constraint for the mobile base reduces the solution space compared to the holonomic counterpart, which results in increased computation time for a feasible solution. The proposed approach demonstrates real-time performance using Python. We believe that significantly faster computation would be achieved with C++ implementation.

TABLE II  
COMPUTATION TIME (IN SECONDS) COMPARISON WITH THE STATE-OF-THE-ART ALGORITHM.

	Min	Mean	Max	Standard deviation
Proposed	0.199	0.580	1.855	0.231
Keshab <i>et al.</i> [24]	0.227	0.272	0.346	0.035
Alonso-Mora <i>et al.</i> [13]	0.46	0.857	1.26	0.256

## V. CONCLUSION

Motion planning for multiple MMRs for objection transportation is a significant challenge, especially for MMRs with a non-holonomic base, due to the kinodynamic constraints that must be included in the planning and the rigid connection between the object and the manipulators. The proposed nonlinear MPC-based motion planning technique derives kinodynamically feasible collision-free trajectories in dynamic environments for non-holonomic MMRs. The proposed method optimizes control inputs for the base and the manipulators of the MMRs. Our simulations and hardware experiments indicate that the trajectory at the intersection of the obstacle-free polytope is non-smooth. In the future, we will address this shortcoming.

## REFERENCES

- [1] O. Khatib, K. Yokoi, K. Chang, D. Ruspini, R. Holmberg, and A. Casal, “Coordination and decentralized cooperation of multiple mobile manipulators,” *Journal of Robotic Systems*, vol. 13, no. 11, pp. 755–764, 1996.
- [2] P. Xu, J. Zhang, Y. Cui, K. Zhang, and Q. Tang, “Modeling and coordinated control of multiple mobile manipulators with closed-chain constraints,” *International Journal of Control, Automation and Systems*, vol. 21, pp. 1296–1308, 4 2023.
- [3] X. Zhao, Y. Zhang, W. Ding, B. Tao, and H. Ding, “A dual-arm robot cooperation framework based on a nonlinear model predictive cooperative control,” *IEEE/ASME Transactions on Mechatronics*, vol. 29, no. 5, pp. 3993–4005, 2024.
- [4] S. Erhart and S. Hirche, “Adaptive force/velocity control for multi-robot cooperative manipulation under uncertain kinematic parameters,” in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 307–314, 2013.
- [5] P. Culbertson and M. Schwager, “Decentralized adaptive control for collaborative manipulation,” in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 278–285, 2018.
- [6] C. K. Verginis, A. Nikou, and D. V. Dimarogonas, “Communication-based decentralized cooperative object transportation using nonlinear model predictive control,” in *2018 European Control Conference (ECC)*, pp. 733–738, 2018.
- [7] G.-B. Dai and Y.-C. Liu, “Distributed coordination and cooperation control for networked mobile manipulators,” *IEEE Transactions on Industrial Electronics*, vol. 64, no. 6, pp. 5065–5074, 2017.
- [8] A. Marino, “Distributed adaptive control of networked cooperative mobile manipulators,” *IEEE Transactions on Control Systems Technology*, vol. 26, no. 5, pp. 1646–1660, 2018.
- [9] Y. Ren, S. Sosnowski, and S. Hirche, “Fully distributed cooperation for networked uncertain mobile manipulators,” *IEEE Transactions on Robotics*, vol. 36, no. 4, pp. 984–1003, 2020.
- [10] K. Patra, A. Sinha, and A. Guha, “Online capability based task allocation of cooperative manipulators,” *Journal of Intelligent & Robotic Systems*, vol. 110, p. 23, 3 2024.
- [11] J. Desai and V. Kumar, “Nonholonomic motion planning for multiple mobile manipulators,” in *Proceedings of International Conference on Robotics and Automation*, vol. 4, pp. 3409–3414 vol.4, 1997.
- [12] H. Tanner, S. Loizou, and K. Kyriakopoulos, “Nonholonomic navigation and control of cooperating mobile manipulators,” *IEEE Transactions on Robotics and Automation*, vol. 19, no. 1, pp. 53–64, 2003.
- [13] J. Alonso-Mora, S. Baker, and D. Rus, “Multi-robot formation control and object transport in dynamic environments via constrained optimization,” *The International Journal of Robotics Research*, vol. 36, no. 9, pp. 1000–1021, 2017.
- [14] Z. Cao, N. Gu, J. Jiao, S. Nahavandi, C. Zhou, and M. Tan, “A novel geometric transportation approach for multiple mobile manipulators in unknown environments,” *IEEE Systems Journal*, vol. 12, no. 2, pp. 1447–1455, 2018.
- [15] J. Jiao, Z. Cao, N. Gu, S. Nahavandi, Y. Yang, and M. Tan, “Transportation by multiple mobile manipulators in unknown environments with obstacles,” *IEEE Systems Journal*, vol. 11, no. 4, pp. 2894–2904, 2017.
- [16] R. Tallamraju, D. H. Salunkhe, S. Rajappa, A. Ahmad, K. Karlapalem, and S. V. Shah, “Motion planning for multi-mobile-manipulator payload transport systems,” in *2019 IEEE 15th International Conference on Automation Science and Engineering (CASE)*, pp. 1469–1474, 2019.
- [17] A. Nikou, C. Verginis, S. Heshmati-alamdarji, and D. V. Dimarogonas, “A nonlinear model predictive control scheme for cooperative manipulation with singularity and collision avoidance,” in *2017 25th Mediterranean Conference on Control and Automation (MED)*, pp. 707–712, 2017.
- [18] F. Kennel-Maushart and S. Coros, “Payload-aware trajectory optimisation for non-holonomic mobile multi-robot manipulation with tip-over avoidance,” *IEEE Robotics and Automation Letters*, vol. 9, no. 9, pp. 7669–7676, 2024.
- [19] O. Shorinwa and M. Schwager, “Scalable collaborative manipulation with distributed trajectory planning,” in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 9108–9115, 2020.
- [20] C. Wu, H. Fang, Q. Yang, X. Zeng, Y. Wei, and J. Chen, “Distributed cooperative control of redundant mobile manipulators with safety constraints,” *IEEE Transactions on Cybernetics*, pp. 1–13, 2021.
- [21] J. Hu, W. Liu, H. Zhang, J. Yi, and Z. Xiong, “Multi-robot object transport motion planning with a deformable sheet,” *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 9350–9357, 2022.
- [22] L. Pei, J. Lin, Z. Han, L. Quan, Y. Cao, C. Xu, and F. Gao, “Collaborative planning for catching and transporting objects in unstructured environments,” *IEEE Robotics and Automation Letters*, vol. 9, no. 2, pp. 1098–1105, 2024.
- [23] R. Mao, H. Gao, and L. Guo, “A novel collision-free navigation approach for multiple nonholonomic robots based on orca and linear mpc,” *Mathematical Problems in Engineering*, vol. 2020, pp. 1–16, 6 2020.
- [24] K. Patra, A. Sinha, and A. Guha, “Kinodynamic motion planning for collaborative object transportation by multiple mobile manipulators,” 2024.
- [25] H. Choset, K. Lynch, S. Hutchinson, G. Kantor, and W. Burgard, *Principles of Robot Motion: Theory, Algorithms, and Implementations*. Intelligent Robotics and Autonomous Agents series, MIT Press, 2005.
- [26] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, “CasADi – A software framework for nonlinear optimization and optimal control,” *Mathematical Programming Computation*, vol. 11, no. 1, pp. 1–36, 2019.