

# DEFORM: Adaptive Formation Reconfiguration of Multi-robot Systems in Confined Environments

Jin Li<sup>†</sup>, Yang Xu<sup>†</sup>, Member, IEEE, Xiufang Shi, Member, IEEE, and Liang Li, Member, IEEE

**Abstract**—Achieving desired formation patterns without collisions is rather challenging for multi-robot systems in unknown obstacle-rich and confined environments, especially in narrow corridor scenes containing large-volume obstacles. To address this, we propose an adaptive formation reconfiguration method that can dynamically switch to the optimal formation pattern based on the current obstacle distribution. Specifically, we develop a novel obstacle-free maximum passable width detection method to formulate recursive optimization problems, which can determine the currently best formation shape and refine local goals away from obstacles. Then, we design a task assignment module for the temporary leader robot and a consensus-based distributed formation controller for each robot using model predictive control to ensure rapid convergence to the suggested formation shape. In addition, we utilize the potential field approach for each robot to improve collision avoidance. Extensive Gazebo simulations and real-world experiments in confined and obstacle-rich scenes verify the efficient formation convergence of our methods compared to the previous methods.

**Index Terms**—Multi-robot systems, collision avoidance, formation control, model predictive control, underactuated robots

## I. INTRODUCTION

FORMATION maneuver is a fundamental capability for multi-robot systems, enabling robot swarms to efficiently accomplish collaborative tasks such as object transportation [1], environment exploration [2], and surveillance [3]. In an environment with large obstacles, the robots in the formation are forced to pass through one side of the obstacle due to environmental constraints. In [4], [5], the widely used optimization-based methods integrate formation maintenance and collision avoidance into the objective function to achieve formation control. However, in environments with large obstacles, the priority of collision avoidance is much higher than that of formation maintenance, often resulting in formation distortion to avoid obstacles. Another more suitable approach is to adaptively determine the optimal formation based on environmental information. This is also characterized as the adaptive formation reconfiguration problem. Recent advances in multiple soft robots also enable the robots to sense and formation shape transitions in obstacle-rich scenes, such as the boundary-constrained soft robot system [6]–[8]. However, this soft robot system needs the subunits to act as a whole entity, which can not be applied to a decentralized system directly. To

J. Li and X. Shi are with the College of Information Engineering, Zhejiang University of Technology, Hangzhou 310023, China (email: jinli@zjut.edu.cn; xiufangshi@zjut.edu.cn).

Y. Xu and L. Li are with the College of Control Science and Engineering, Zhejiang University, Hangzhou, 310027, China (email: xuyang94@zju.edu.cn; liang.li@zju.edu.cn).

Source code is available online at <https://github.com/NeSC-IV/DEFORM>.  
Digital Object Identifier (DOI): see top of this page.

<sup>†</sup> indicates equal contribution.

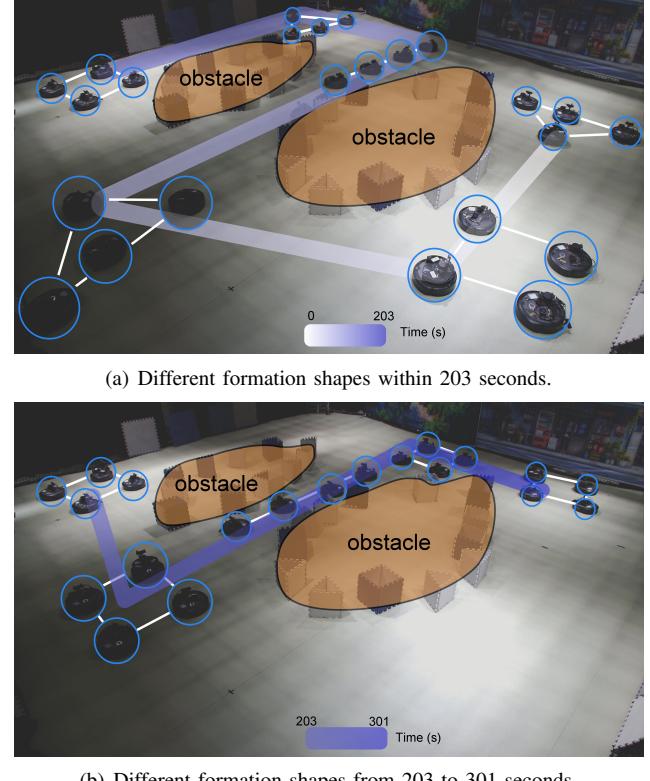


Fig. 1. An example of formation navigation in a real-world environment with large-volume obstacles. The known global path (a gradient of the blue line) guides robots through specific areas. The white line shows the outline of the formation shape.

summarize, adaptively switching formation shapes in real-time based on surrounding unexplored environments is essential for multi-robot systems, which is still an open problem.

To this end, we propose an adaptive formation reconfiguration method for the collision-free formation maneuver of multiple nonholonomic mobile robots in an unknown global environment, allowing for automatic switching and rapid convergence to the optimal formation pattern. The leader robot calculates the maximally feasible width based on the surrounding obstacles to choose the most appropriate formation pattern in the predefined shape library. In the phase of formation reconfiguration, the task allocation algorithm will determine the relative positions of robots and allocate a new leader robot with the best visibility within the formation. Further, we extend Park’s method [9] and propose a distributed formation control algorithm for multiple nonholonomic mobile robots, integrating model predictive control (MPC) and formation consensus. Experimental results show that a task allocation algorithm that minimizes the total travel distance of the

formation, combined with robust distributed formation control, significantly improves the formation convergence time compared to cutting-edge methods. As an appropriate formation can reduce the risk of collisions, we formulate a potential field-based, unconstrained optimization problem to get the best local goal when the safe distance gets lower than the threshold, which can further prevent collisions.

In summary, the contributions of this letter are as follows:

- 1) An environment-adaptive optimal formation pattern selection strategy, offering the currently best formation shape based on the maximum passable width detected by the alterable leader robot, lowers the formation collision risk in obstacle-rich and confined scenes.
- 2) A formation reconfiguration framework, comprising efficient task assignment based on the desired formation pattern and consensus-based distributed MPC formation controller, allows for quick formation shape transition.
- 3) Extensive simulations in various scenes, including large and obstacle-rich ones, and real-world experiments using a nonholonomic robot team (as in Fig. (1)) validate the efficiency and feasibility of our method in practice.

## II. RELATED WORK

### A. Multi-robot Formation Control

Over the past few decades, numerous studies [10] have been proposed to achieve the formation control of multi-robot systems, including leader-following [11], virtual structures [12], potential fields [13] and behavior-based approach [14]. In particular, the formation shape rendezvous problem has been widely studied. Wang *et al.* [15] propose distributed goal assignments based on local sensing information and resolve conflicts among local assignments using local task swapping. Morgan *et al.* [16] form a specific shape using auction assignment and trajectory optimization in a distributed manner. However, these works are limited to transforming to the predetermined formation without considering obstacles. Park *et al.* [9] propose a particle swarm optimization-based formation controller to conduct formation reconfiguration. However, this controller utilizes a reduced particle search space, which may lead to the absence of feasible solutions.

More complex and realistic environments, such as obstacle-rich scenes, have been considered recently. Peng *et al.* [4] optimize the yaw angle of the robot for sensing a more extensive range and use distance-based formation trajectory optimization. However, it cannot generate a user-specified shape due to the formation with rotation and translation invariance. Alonso *et al.* [17] get the feasible area for planning using the intersection of safety convex hulls of each robot. In this area, the method in [17] rearranges the desired formation and then plans local trajectories. Still, this framework cannot achieve adaptive formation transformation due to insufficient utilization of environmental information. The literature [5] considers both formation-keeping and obstacle avoidance in an obstacle-rich environment. Nevertheless, formation keeping, including affine transformation, does not take full advantage of environmental information to avoid obstacles, distorting the formation shape in narrow corridors. In contrast, we employ local sensor observations to match the appropriate formation, enabling the natural trade-off between formation-keeping and obstacle avoidance.

### B. Safe Corridors in Multi-robot Formation

The safe corridor is a commonly used technique for motion planning. Liu *et al.* [18] propose a method for generating a safe corridor by multiple iterations of convex polygons. Chen *et al.* [19] propose to generate an online safe corridor consisting of large, overlapping 3-D cubes that are simply axis-aligned, resulting in an overly conservative safe corridor. In [20], the authors propose a method for generating convex polytopes from a convex cluster of voxels, and the cluster is obtained by iterative inflation based on a spatially discretized map. The representative method IRIS [21] iteratively alternates between solving the convex polyhedron and the ellipse, utilizing quadratic programming (QP) and semi-definite programming (SDP), respectively. But it has a longer execution time for each iteration. The method in [22] directly obtains the maximally feasible width around the path and sets the desired formations around regions with width variations. Nevertheless, the use case of this method assumes a known global map. Additionally, these works have not thoroughly addressed the negatives of safety space's scale, which may drive the robot to get too close to obstacles and other robots. On the contrary, our method iteratively refines the local goal to keep the formation in the middle of obstacle-free areas, significantly reducing the probability of robot collisions in constrained environments.

## III. PRELIMINARIES AND PROBLEM STATEMENT

In this letter, we consider a 2D multi-robot system consisting of  $N$  mobile robots operating within the workspace  $\mathcal{W} \in \mathbb{R}^2$ . The obstacles in the map are unknown in advance and denoted as  $\mathcal{O}$ . The free workspace of the robots is represented by  $\mathcal{F} = \mathcal{W} \setminus \mathcal{O}$ . The collision model of each robot is defined as a circle with a radius  $\mathcal{R}$ .

### A. Nonholonomic Mobile Robots

Here, we use the unicycle model for the differential-wheeled nonholonomic mobile robot. If not specified, the state  $\mathbf{x}_i := [x_i, y_i, \theta_i]^T$  represents the 2D position and orientation of  $i$ -th robot ( $i \in \{1, \dots, N\}$ ). The input  $\mathbf{u}_i = [v_i, \omega_i]^T$  consists of the linear velocity  $v_i$  and angular velocity  $\omega_i$ . The kinematic equations for  $i$ -th robot are described as follows:

$$\dot{\mathbf{x}}_i = \begin{bmatrix} \dot{x}_i \\ \dot{y}_i \\ \dot{\theta}_i \end{bmatrix} = \begin{bmatrix} \cos\theta_i & 0 \\ \sin\theta_i & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_i \\ \omega_i \end{bmatrix} = f(\mathbf{x}_i, \mathbf{u}_i). \quad (1)$$

Note that the dimensions of the 2D position  $\mathbf{p}_{i(\cdot)} = [p_{ix(\cdot)}, p_{iy(\cdot)}, 0]^T$  and the state  $\mathbf{x}_{i(\cdot)}$  are both  $\mathbb{R}^{3 \times 1}$  in this letter, the third element of  $\mathbf{p}_{i(\cdot)}$  is set to 0 for facilitating computation.  $\mathbf{Q}_b = \text{diag}(1, 1, 0)$  is used to transform  $\mathbf{x}_{i(\cdot)}$  into  $\mathbf{p}_{i(\cdot)}$ . Other notations can be referred to Table. I. Here, we also list some basic notations used in the rest of this letter.

- $\mathbf{p}_{ig,(\cdot)}$  : Local goal of the  $i$ -th robot, including various cases:  $\mathbf{p}_{ig}$  as the local goal in the path,  $\mathbf{p}_{ig,o}$  is the local goal for avoid obstacles,  $\mathbf{p}_{ig,w}$  is the local goal after width check,  $\mathbf{p}_{ig,r}$  is the refined local goal.
- $\mathbf{p}_{ic}, \mathbf{x}_{ic}$  : Current position and state of the  $i$ -th robot.

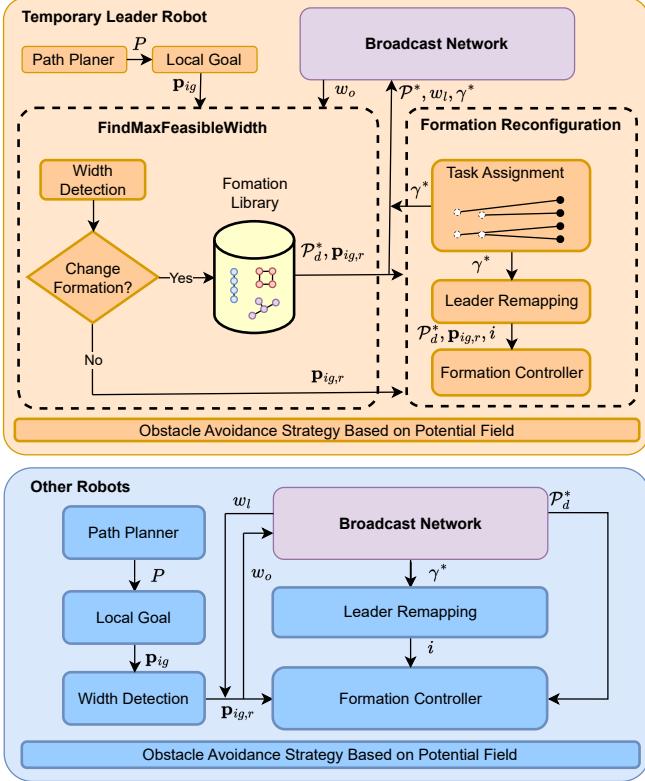


Fig. 2. An overview of our adaptive formation reconfiguration framework.

### B. Adaptive Formation Reconfiguration Problem

In this letter, a formation pattern is denoted as  $\mathcal{P}$ , and its matrix representation is given by  $\mathbf{P}_f = \{\mathbf{p}_{f,1}, \dots, \mathbf{p}_{f,N}\} \in \mathbb{R}^{3 \times N}$ , where  $\mathbf{p}_{f,i} = [p_{f,ix}, p_{f,iy}, 0]^T$  represents the relative positions of the  $i$ -th robot and  $\mathbf{p}_{f,0}$  denotes the origin in formation coordinate frame. Note that the formation pattern in this letter only defines each robot's longitudinal and lateral coordinates in the formation frame. We make the following assumptions about the formation reconfiguration problem:

*Assumption:* Each robot communicates with all other robots to share its current information, such as position and formation-related information.

*Problem:* The objective is to design a multi-robot formation reconfiguration algorithm, which can narrow/widen the formation width adaptively based on the environment and automatically transition from one formation pattern  $\mathcal{P}$  to another one  $\mathcal{P}'$  in unknown scenes. Each nonholonomic robot can finish an assigned task of moving from its start position  $s_i \in \mathcal{F}$  to its desired position  $g_i \in \mathcal{F}$  while avoiding obstacles and collisions in the reconfiguration process.

### IV. SYSTEM OVERVIEW

As mentioned above, this letter aims to improve the autonomy of multi-robot formation in obstacle-rich environments using adaptive formation switching. To accomplish this, we propose a formation control framework, including width detection, formation reconfiguration, and potential field-based obstacle avoidance strategy. As shown in Fig. 2, two threads, namely the formation thread and the obstacle avoidance thread, are utilized to implement the entire algorithm.

The brief workflow of our formation thread is explained as follows. Firstly, based on the final goal  $g$ , the robot

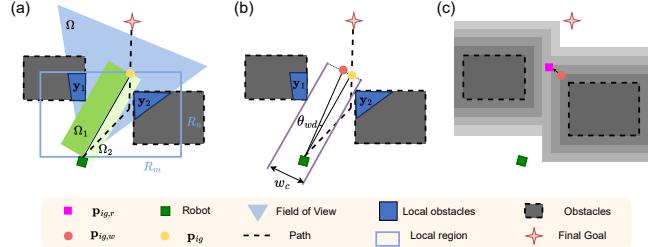


Fig. 3. Illustration of width check and refining local goal. (a) displays the relevant areas for width detection. For example,  $\Omega$  represents the camera's Field of View(FOV),  $\Omega_1$  and  $\Omega_2$  visualize the region of the constraint in Eq. (3). (b) The gap between the two purple lines represents the current width  $w_c$ , and the midpoint of the line connecting the endpoints of these two lines serves as the new local goal  $\mathbf{p}_{ig,w}$ , replacing the local goal  $\mathbf{p}_{ig}$  on the path. (c) The refined local goal  $\mathbf{p}_{ig,r}$  is a safer position near the new local goal.

plans a path  $P$  using existing algorithms, such as A\*, and extracts a fixed length along this path as the local goal  $\mathbf{p}_{ig}$ . Then, in the *FindMaxFeasibleWidth* module, the temporary leader robot obtains the desired formation  $\mathcal{P}_d^*$ , refines local goal  $\mathbf{p}_{ig,r}$ , and determines the maximum feasible width  $w_l$  (see Section V). This formation information is shared with other robots by the broadcast network and used as input for the *Formation Reconfiguration* module (see Section VI). In this module, the three sub-modules of task assignment, leader remapping, and formation controller achieve efficient formation reconfiguration. The robots in the formation share the maximum obstacle-free width in the broadcast network, allowing for formation expansion and compaction in different environments, such as  $w_l$  and  $w_o$ . Unlike the leader robot, other robots can adjust their refined local goal  $\mathbf{p}_{ig,r}$  via width detection and achieve the desired formation position by receiving formation information.

The formation thread controls multiple robots to move as a formation while switching to the obstacle avoidance thread once the distance between one of the robots and any obstacle gets less than the threshold  $d_{th,o}$ . The obstacle avoidance strategy based on potential field function is solved as an unconstrained optimization problem:

$$\min_{\mathbf{p}_{ig,o}} w_{o1} \max\{d_{th,o} - d_o(\mathbf{p}_{ig,o}), 0\}^2 + w_{o2} \|\mathbf{p}_{ig,o} - \mathbf{p}_{ic}\|^2, \quad (2)$$

where  $\mathbf{p}_{ig,o}$  is the local goal to avoid obstacles,  $w_{o1}$  and  $w_{o2}$  are the weight coefficients.  $d_o(\mathbf{p}_{ig,o})$  represents the distance between the obstacle and  $\mathbf{p}_{ig,o}$ . Notably, Eq. (2) is designed to make the robot quickly move away from the obstacle.

### V. OPTIMAL FORMATION SELECTION

As an essential part of formation control, the optimal formation selection in complex environments needs to calculate the maximum obstacle-free width only based on local information and then select the desired formation from the formation library. We develop an algorithm to sequentially solve the optimization problems of *Width check* and *Refining local goal*.

#### A. Width check

In Fig. 3(a), the local region is an axis-aligned rectangle defined by two parameters,  $R_m$  and  $R_n$ .  $R_m$  is the width occupied by the widest formation pattern ( $\mathcal{P}_1$  in Fig. 5) in the preset formation library, and the value of  $R_n$  is the difference in the longitudinal coordinates between the robot's current

position  $\mathbf{p}_{ic}$  and the local goal  $\mathbf{p}_{ig}$ . The robot uses a camera to perceive the external environment. Local obstacles are the intersection of the local region and the camera's field of view (FOV) in Fig. 3(a). Then, the QP problem of finding the maximum feasible width among these obstacles can be formulated as follows:

$$\max_{b_2, b_1} \|b_2 - b_1\|^2, \quad (3a)$$

$$\text{s.t. } \mathbf{a}^T \mathbf{y}_1 + b_1 > 0, \quad b_1 > b, \quad (3b)$$

$$\mathbf{a}^T \mathbf{y}_2 + b_2 < 0, \quad b_2 < b, \quad (3c)$$

$$|b_1 - b| < w_{mf} \|\mathbf{a}\|, \quad (3d)$$

$$|b_0 - b| < w_{mf} \|\mathbf{a}\|, \quad (3e)$$

where  $\mathbf{a} \in \mathbb{R}^{2 \times 1}$  and  $b \in \mathbb{R}$  are coefficients of the linear equation, which is determined by  $\mathbf{p}_{ic}$  and  $\mathbf{p}_{ig}$ .  $\mathbf{y}_1$  and  $\mathbf{y}_2$  represent the group of 2D position inside the local obstacles in Fig. 3(a),  $w_{mf}$  is the maximum width in Fig. 5. And  $b_1$  and  $b_2$  are optimization variables. Eq. (3b) constrains the line  $l_a : \{\mathbf{y} | \mathbf{a}^T \mathbf{y} + b_1 = 0, \mathbf{y} \in \mathbb{R}^2\}$  to region  $\Omega_1$ , while Eq. (3c) constrains the line  $l_b : \{\mathbf{y} | \mathbf{a}^T \mathbf{y} + b_2 = 0, \mathbf{y} \in \mathbb{R}^2\}$  to region  $\Omega_2$  in Fig. 3(a). Let  $b_1^*$  and  $b_2^*$  be the results of the optimization, then  $w_c = |b_1^* + b_2^*|/\|\mathbf{a}\|$  is the feasible width and  $\mathbf{p}_{ig,w}$  can be obtained from the middle of the maximum feasible region and guides the multi-robot formation to move towards regions with fewer obstacles in Fig. 3(b).

### B. Refining local goal

To provide more operational space for the formation, an unconstrained optimization is conducted as follows:

$$\begin{aligned} \mathbf{p}_{ig,r} = \arg \min_{\mathbf{p}_{ig,r}} w_{r1} \max\{d_{th,r} - d_o(\mathbf{p}_{ig,r}), 0\}^2 \\ + w_{r2} \|\mathbf{p}_{ig,r} - \mathbf{p}_{ig,w}\|^2, \end{aligned} \quad (4)$$

where  $w_{r1}, w_{r2}, d_{th,r}$  and  $d_o(\mathbf{p}_{ig,r})$  are consistent with those defined in Eq.(2). The refined local  $\mathbf{p}_{ig,r}$  is located near  $\mathbf{p}_{ig,w}$ , and  $\mathbf{p}_{ig,r}$  is farther away from the nearest obstacle, as shown in Fig. 3(c).

We develop each robot's maximum feasible width detection, outlined in Algorithm 1. In each iteration, the **WidthCheck()** and **UpdateNewLocalGoal()** are calculated in real-time to obtain  $w_c$  and  $\mathbf{p}_{ig,w}$ , respectively. Before each iteration, three conditions need to be satisfied together: the iteration counts do not exceed  $I_{max}$ , the amount of change in width is greater than  $w_{th}$ , and the absolute value of the angle  $\theta_{wd}$  (See Fig. 3(a)) is less than  $\theta_{th}$  (Line 2). The **RefineLocalGoal()** computes a safer local goal  $\mathbf{p}_{ig,r}$  and the maximum width  $w_c^*$  that determines the matrix form of the formation shape  $\mathbf{P}_f^*$  is recalculated (Lines 10–12). In practice, the direction of the formation shape needs to be aligned with the current motion direction. Therefore, the angle  $\theta^*$  is defined as the direction of motion, and  $\mathbf{R}$  is the corresponding rotation matrix, where  $\mathbf{I}_z = (0, 0, 1)^T$  (Lines 13–14). Finally, the desired formation with rotation information is obtained as  $\mathcal{P}_d^*$  (Line 15).

## VI. FORMATION RECONFIGURATION PLANNING

Fast convergence to the new formation pattern is crucial for adaptive formation reconfiguration. In this section, we focus on how to achieve fast formation reconfiguration. Firstly, the task

---

### Algorithm 1 Width Detection

**Notation:** Iteration number  $I$ , Current width  $w_c$ , Last width  $w_l$

1: **Initialize:**  $I \leftarrow 0, \theta_0 \leftarrow \arccos\left(\frac{\mathbf{p}_{ig} \cdot \mathbf{p}_{ic}}{\|\mathbf{p}_{ig}\| \|\mathbf{p}_{ic}\|}\right), \mathbf{p}_{ig,w} \leftarrow \mathbf{p}_{ig}, w_l \leftarrow 0, w_c \leftarrow 2w_{th}$   
2: **while**  $I < I_{max}$  **and**  $|\theta_{wd}| < \theta_{th}$  **and**  $|w_c - w_l| > w_{th}$   
**do**  
3:    $w_l \leftarrow w_c$   
4:    $w_c \leftarrow \mathbf{WidthCheck}(\mathbf{p}_{ic}, \mathbf{p}_{ig,w})$  ▷ detailed in Eq.(3)  
5:    $\mathbf{p}_{ig,w} \leftarrow \mathbf{UpdateNewLocalGoal}()$   
6:    $\theta_i \leftarrow \arccos\left(\frac{\mathbf{p}_{ig,w} \cdot \mathbf{p}_{ic}}{\|\mathbf{p}_{ig,w}\| \|\mathbf{p}_{ic}\|}\right)$   
7:    $\theta_{wd} \leftarrow \theta_i - \theta_0$   
8:    $I = I + 1$   
9: **end while**  
10:  $\mathbf{p}_{ig,r} \leftarrow \mathbf{RefineLocalGoal}(\mathbf{p}_{ig,w})$  ▷ detailed in Eq. (4)  
11:  $w_c^* \leftarrow \mathbf{WidthCheck}(\mathbf{p}_{ic}, \mathbf{p}_{ig,r})$   
12:  $\mathbf{P}_f^* \leftarrow \mathbf{UpdateDesiredFormation}(w_c^*)$   
13:  $\theta^* \leftarrow \arccos\left(\frac{\mathbf{p}_{ig,r} \cdot \mathbf{p}_{ic}}{\|\mathbf{p}_{ig,r}\| \|\mathbf{p}_{ic}\|}\right)$   
14:  $\mathbf{R} \leftarrow \mathbf{AngleAxisToRotationMatrix}(\theta^*, \mathbf{I}_z)$   
15:  $\mathcal{P}_d^* \leftarrow \mathbf{R} \cdot \mathbf{P}_f^*$   
16: **return**  $\mathbf{p}_{ig,r}, \mathcal{P}_d^*, w_c^*$

---

assignment module assigns each robot a new relative position (see Section VI-A). The robot assigned as the “leader” continuously monitors the surrounding environment to determine if a new formation pattern is needed. Notably, the “leader” robot may not be fixed in each task assignment (see Section VI-B). Subsequently, we propose the distributed formation control scheme (see Section VI-C) to drive the robots to the formation goals efficiently. We devise this formation reconfiguration strategy that leverages the merits of centralized leader robot remapping for fast convergence and decentralized local motion control for saving computational resources.

### A. Task Assignment

In our work, the task assignment module can allocate relative positions to robots in a new formation pattern, minimizing the total travel distance of the formation. We compute the optimal transport [23] between discrete distributions to provide the correspondence between robots and the new formation. This problem is formulated as:

$$\gamma^* = \arg \min_{\gamma} \sum_{i,j} \gamma_{i,j} \mathbf{M}_{i,j}, \quad (5a)$$

$$\text{s.t. } \gamma \mathbf{1}^N = \mathbf{1}^N; \gamma^T \mathbf{1}^N = \mathbf{1}^N, \quad (5b)$$

where the optimization variable  $\gamma$  is a matrix consisting only of 0 and 1,  $\gamma_{i,j}$  represents whether to assign robot  $i$  to the  $j$ -th position in the new formation. The vector  $\mathbf{1}^N$  is a column vector of size  $N$  where all elements equal 1.  $\mathbf{M}$  is the metric cost matrix, where each element represents the distance from the  $i$ -th robot to the  $j$ -th position in the new formation. The optimization problem formulated by Eq.(5a)-(5b) can be resolved using the open-source library POT [24].

### B. Leader Robot Remapping

Differently, the “leader” robot here is alterable, temporary, and can be reselected based on each task assignment. The remapping of the leader robot sets a robot in a fixed position in the formation as the “leader” robot. Fig. 4 shows the

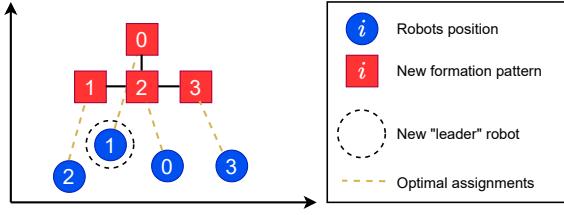


Fig. 4. Illustration of task assignment and leader robot remapping.

brief process of the task assignment and leader remapping, where the desired positions in the new formation and the current positions of the robots are marked by red boxes and blue circles, respectively, and the optimal position assignments are represented by yellow dashed lines. The position of the “leader” robot in the formation is generally predetermined to obtain the largest field of view for observing obstacles. Here, robot 1 is assigned to the 0-th position in the new formation pattern and becomes the new “leader” robot. The “leader” robot determines the time and the type of formation pattern to change, as shown in Fig. 2.

### C. MPC-based Formation Control

According to the consensus theory [25], the reference state  $\mathbf{x}_{ir}$  of robot  $i$  can be defined by other robots and the desired formation pattern:

$$\mathbf{x}_{ir} = \frac{1}{N_i + 1} \left( \sum_{j \in \mathcal{N}_i} (\mathbf{x}_j + \mathbf{p}_{f,ij}) + \mathbf{p}_{ig,r} \right), \quad (6)$$

where  $\mathcal{N}_i$  is the set of all robots except robot  $i$ , and  $N_i$  is the number of robots belonging to  $\mathcal{N}_i$ . Besides,  $\mathbf{p}_{f,ij} = \mathbf{p}_{f,i} - \mathbf{p}_{f,j}$  is the relative position between robot  $i$  and robot  $j$  in the desired formation,  $\mathbf{x}_{ie} = \mathbf{x}_{ir} - \mathbf{x}_{ic}$  is the current state error. The reference input  $\mathbf{u}_{ir} = [v_{ir}, w_{ir}]^T$  for each robot  $i$  is designed as follows:

$$v_{ir} = \begin{cases} \frac{\|\mathbf{p}_{ie}\|}{x_{th}} \cdot v_{max}, & \|\mathbf{p}_{ie}\| \leq x_{th}, \\ v_{max}, & \|\mathbf{p}_{ie}\| \geq x_{th} \end{cases}, \quad (7a)$$

$$w_{ir} = \frac{1}{\pi} \cdot w_{max} \cdot \arccos\left(\frac{\mathbf{v}_r \cdot \mathbf{p}_{ie}}{\|\mathbf{v}_r\|_2 \cdot \|\mathbf{p}_{ie}\|_2}\right), \quad (7b)$$

where  $v_{max}$  and  $w_{max}$  represent the maximum linear and angular velocities, respectively.  $\mathbf{p}_{ie} = \mathbf{Q}_b \mathbf{x}_{ie}$  is the current position error.  $x_{th}$  is the distance threshold, and  $\mathbf{v}_r \in \mathbb{R}^3$  is the unit vector indicating the current orientation of the robot. This method enables the robots to reach the local goal and rotate to the orientation with the smallest angle deviation as quickly as possible, which is necessary for nonholonomic mobile robots.

In a prediction horizon, let  $\mathbf{x}_i(t_{k+m|k}) (m = 0, \dots, M-1)$  and  $\mathbf{u}_i(t_{k+m|k})$  denote the predicted states and inputs of robot  $i$  at time step  $t_k$ , respectively,  $M$  is the number of prediction steps. The optimization problem based on the distributed

optimization MPC framework can be solved as a nonlinear constrained optimization:

$$\min_{v_i, w_i, \varepsilon} \sum_{m=0}^{M-1} \|\mathbf{x}_i(t_{k+m|k}) - \mathbf{x}_{ir}\|_{\mathbf{Q}} + \|\mathbf{u}_i(t_{k+m|k}) - \mathbf{u}_{ir}\|_{\mathbf{R}} + w_i \varepsilon^2, \quad (8a)$$

$$\text{s.t. } \mathbf{x}_i(t_{k+m+1|k}) = f(\mathbf{x}_i(t_{k+m|k}), \mathbf{u}_i(t_{k+m|k})), \quad (8b)$$

$$0 \leq v_i(t_{k+m}) \leq v_{max}, \quad (8c)$$

$$|w_i(t_{k+m|k})| \leq w_{max}, \quad (8d)$$

$$\varepsilon_{th} \leq \varepsilon \leq 0, \quad (8e)$$

$$d_{ij}(t_{k+m|k}) \geq d_{safe} + \varepsilon, \quad \forall j \in \mathcal{N}_i, \quad (8f)$$

where  $\mathbf{Q} \in \mathbb{R}_+^{3 \times 3}$  and  $\mathbf{R} \in \mathbb{R}_+^{2 \times 2}$  are diagonal weight matrices.  $\varepsilon$  is to relax constraints Eq. (8f), approximately equivalent to converting hard constraints to soft constraints. The  $d_{ij}(t_{k+m|k}) = \|\mathbf{x}_i(t_{k+m|k}) - \mathbf{x}_j\|_{\mathbf{Q}_b}$  is the Euclidean distance between the prediction positions of robot  $i$  and the position of robot  $j$  and  $d_{safe}$  is the safe distance between robots.

We define the cost function Eq. (8a) to achieve smooth and efficient formation control. The constraint Eq. (8b) denotes the discrete-time form of Eq. (1), the constraints of Eq. (8c)-(8e) can be handled by bounding the search space for  $v_i$ ,  $w_i$  and  $\varepsilon$ , and Eq. (8f) maintain a certain distance between the robots.

## VII. EXPERIMENTS

To validate our method, we conduct extensive Gazebo simulations and real-world experiments, where the leader robot can detect obstacles using an onboard depth camera. The Robot Operating System (ROS) and the TCP/IP protocol are used to ensure the reliability of data sharing in experimental environments.

### A. Gazebo Simulations

To demonstrate the efficiency and robustness of our method, we conduct Gazebo simulations compared with cutting-edge formation control methods: Quan’s method [5] and Park’s method [9]. Since Park’s method can not be directly applied to obstacle-rich scenes, we combined Park’s method with the FindMaxFeasibleWidth module (see Fig. 2) and the Obstacle Avoidance strategy (see Fig. 2). However, these modifications do not affect the comparison results in the experiments of obstacle-free scenes. To evaluate the formation error between the current shape  $\mathcal{P}_c$  and the desired one  $\mathcal{P}_d$ , we apply two evaluation metrics originally used in the baseline methods. The first metric is the average formation distance error  $\bar{e}_{dist}$  in Park’s work [9]:

$$\mathbf{x}_{id} = \frac{1}{N_i} \sum_{j \in \mathcal{N}_i} (\mathbf{x}_j + \mathbf{p}_{f,ij}), \quad (9)$$

$$\bar{e}_{dist} = \frac{1}{T_{sum}} \sum_{d=1}^D \sum_{i \in \mathcal{I}} \|\mathbf{x}_{id}(t_d) - \mathbf{x}_{ic}(t_d)\|_{\mathbf{Q}_b}, \quad (10)$$

where Eq. (9) is similar to Eq. (6), but it only considers the formation error rather than the local goal,  $T_{sum} = \sum_{d=1}^D t_d$  is the total time of completing one task corresponding to the following two sets of experiments: Section VII-A1 and

TABLE I  
FORMATION PARAMETERS

Parameter	Symbol	Value
Distance threshold in Eq. (7a)	$x_{th}$	0.5m
Maximum linear velocity	$v_{max}$	0.22 m/s
Maximum angular velocity	$w_{max}$	1.5 rad/s
State error weights	$\mathbf{Q}$	diag(0.1,0.1,0.0)
Control input error weights	$\mathbf{R}$	diag(0.02,0.02)
Weight for relaxing reciprocal avoidance	$w_\varepsilon$	100
Threshold for relaxing reciprocal avoidance	$\varepsilon_{th}$	-0.3
Safe distance for reciprocal avoidance	$d_{safe}$	0.3m
Sampling time	$t_k$	0.1
Number of Prediction Steps	$M$	40
Maximum width of formation	$w_{mf}$	2.4m
Weight of local goal in Eq. (4)	$w_{r1}$	5
Weight of formation in Eq. (4)	$w_{r2}$	3
Weight of obstacle in Eq. (2)	$w_{o1}$	2
Weight of current position in Eq. (2)	$w_{o2}$	8
Safe distance for obstacles	$d_{th,o}$	0.2
Distance considering formation	$d_{th,r}$	1.2m

Section VII-A2.  $D$  is the number of samples uniformly taken from  $T_{sum}$ ,  $\Delta t = t_{d+1} - t_d = 0.5$  is a fixed time slot.

The second metric used in [5] is based on the graph theory. Specifically, each formation mode corresponds to a graph  $\mathcal{G}$ . The weight of each edge in the graph  $\mathcal{G}$  is  $w_{ij} = \|\mathbf{p}_i - \mathbf{p}_j\|$ . The adjacency matrix  $\mathbf{A} \in \mathbb{R}^{N \times N}$  and degree matrix  $\mathbf{D} \in \mathbb{R}^{N \times N}$  of the formation pattern  $\mathcal{P}$  can be determined. Due to the limited space, the definitions of matrices  $\mathbf{A}$  and  $\mathbf{D}$  are the same as those in [5]. Thus, the corresponding Laplacian matrix is:

$$\mathbf{L} = \mathbf{D} - \mathbf{A}. \quad (11)$$

The symmetric normalized Laplacian matrix is defined by:

$$\hat{\mathbf{L}} = \mathbf{D}^{-1/2} \mathbf{L} \mathbf{D}^{-1/2}. \quad (12)$$

We can calculate the second metric of average formation similarity error  $\bar{e}_{sim}$  as follows:

$$\bar{e}_{sim} = \frac{1}{T_{sum}} \sum_{d=1}^D \|\hat{\mathbf{L}}(t_d) - \hat{\mathbf{L}}_{des}(t_d)\|_F^2, \quad (13)$$

where  $\|\cdot\|_F$  denotes the Frobenius norm,  $\hat{\mathbf{L}}(t_d)$  is the symmetric normalized Laplacian matrix of the current formation shape at time  $t_d$ ,  $\hat{\mathbf{L}}_{des}(t_d)$  is the counterpart of the desired formation shape.

The benchmark experiments are conducted on a laptop with an Intel i7-14650HX CPU and 16G RAM. The formation library contains shapes of various widths, as shown in Fig. 5. Table I lists some important parameters used in the following benchmarks and simulations. The unconstrained optimization problems (Eq. (2) and Eq. (4)) are solved by an open-source library LBFGS-Lite<sup>1</sup> and the constrained optimization problems (Eq.(3a)-(3e) and Eq. (8a)-(8f)) are solved by an open-source library CasADI [26].

1) *Convergence of Formation in Obstacle-free Scene*: Rapid convergence to the desired formation is essential for formation control tasks. To evaluate this performance, we conducted 12 sets of formation pattern switching experiments, including  $\mathcal{P}_1$  to  $\mathcal{P}_2$ ,  $\mathcal{P}_2$  to  $\mathcal{P}_3$ ,  $\mathcal{P}_3$  to  $\mathcal{P}_4$ , and so on. In an obstacle-free environment, once the formation error  $\bar{e}_{dist}$  or  $\bar{e}_{sim}$  falls below the predetermined reference value, it indicates

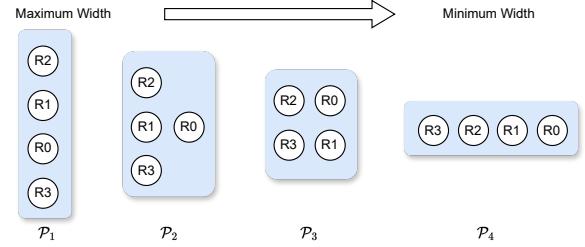


Fig. 5. The formation library contains four shapes. From left to right, the longitudinal width of the formation decreases.

TABLE II  
PERFORMANCE COMPARISON OF 3 PATTERNS TRANSITION

Method	Time Consumption(seconds)		
	$\mathcal{P}_1 \rightarrow \mathcal{P}_2$	$\mathcal{P}_2 \rightarrow \mathcal{P}_3$	$\mathcal{P}_3 \rightarrow \mathcal{P}_4$
Quan' [5]	39	47.1	52.7
Park's [9]	72.6	59.9	70.2
Ours	<b>12.5</b>	<b>12.6</b>	<b>14.6</b>

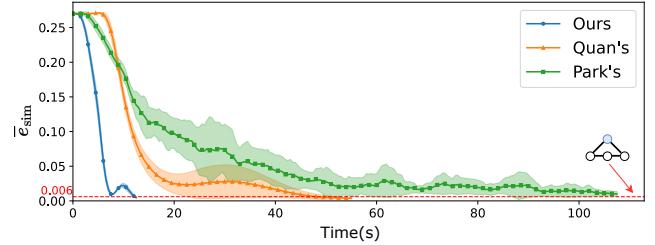


Fig. 6. Formation similarity errors  $\bar{e}_{sim}$  of different methods during the convergence from formation pattern  $\mathcal{P}_1$  to  $\mathcal{P}_2$ .

the test is completed. We repeat the same experiment five times for each method. Table II shows each method's average time of successful convergence in three specific formation pattern switching. Fig. 6 shows the trend of formation errors over time during the convergence from  $\mathcal{P}_1$  to  $\mathcal{P}_2$ . More trend plots of formation errors are in the supplementary video [27].

The results show that the formation convergence efficiency of Park's method performs worst. This is because Park's method gets stuck in local optima regarding the convergence speed to the desired formation. Potential reasons include: the lack of task allocation to reduce the total distance traveled by all robots; and the fact that Park's paper does not provide the parameters for PSO. Compared to Park's method [9], Quan's method [5] has better convergence efficiency because it uses a Quasi-Newton approach, which has a higher solving efficiency than the particle swarm optimization algorithm. However, Quan's formation convergence efficiency is still lower than our method's. In Quan's method [5], formation similarity and dynamic feasibility are both considered in the same cost function but assigned different weights, while we significantly increase the weight of the dynamic feasibility due to the relatively low speed of our robots, which prevents the formation error from converging quickly. Without this adjustment, the state of our robots would be unable to maintain alignment with the predicted trajectory.

As shown in Table II, our method achieves the fastest convergence to the desired formation. The main reasons lie in that the task assignment module reduces the overall travel distance, and our efficient formation controller performs better

<sup>1</sup><https://github.com/ZJU-FAST-Lab/LBFGS-Lite>

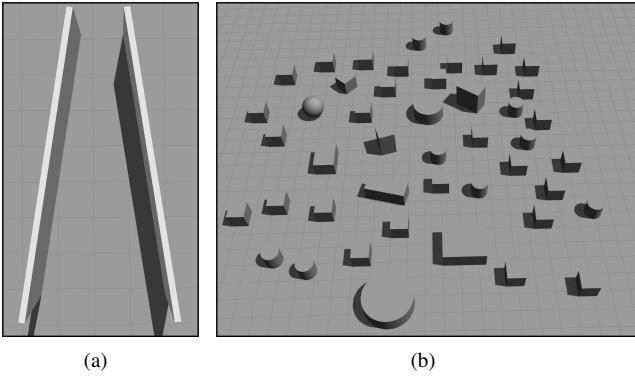


Fig. 7. The corridor and obstacle-rich scenarios.

TABLE III  
PERFORMANCE COMPARISON IN CONFINED ENVIRONMENTS

Scenario	Method	SR(%)	time(s)	$\bar{e}_{dist}$	$\bar{e}_{sim}$
Corridor	Park's [9]	80	121.17	2.65	0.25
	Quan's [5]	95	82.54	-	0.26
	<b>Ours</b>	<b>100</b>	<b>77.28</b>	<b>1.9</b>	<b>0.11</b>
Obstacle-rich	Park's [9]	75	250.28	2.51	0.23
	Quan's [5]	95	142.32	-	0.22
	<b>Ours</b>	<b>100</b>	<b>142.16</b>	<b>1.72</b>	<b>0.079</b>

in minimizing the convergence time.

2) *Formation Control in Confined Environments*: As in Fig. 7, we test a 4-robot formation in a corridor map with varying widths and an obstacle-rich map. The obstacle-rich map is of  $20\text{ m} \times 15\text{ m}$  size. The corridor map is of  $(1+3.1)\times 7.4\text{ m}$  size and has a trapezoidal shape. The velocity limit of all robots is  $0.2\text{ m/s}$ .

We demonstrate the success rate (**SR**), runtime, and formation errors of all methods for twenty tests in each map. The results are shown in Table III. Quan's method defines the formation with invariance to rotation, scale, and translation, while the  $\bar{e}_{dist}$  is used to evaluate formations with only rotational invariance. Therefore,  $\bar{e}_{dist}$  cannot correctly assess the formation navigation performance of Quan's method, and the corresponding data is not presented in Table III. Park's method [9], due to its limitations analyzed in Section VII-A1, finds the solution with a relatively low speed of convergence to the desired formation, making it difficult to quickly restore the desired formation in the presence of obstacles or formation distortions. In scenarios where obstacles are larger than the formation, the robot team is forced to stay on the same side of the obstacle. In such scenarios, Quan's method [5] prioritizes collision avoidance constraints over formation maintenance, leading to poor formation.

The results in Table III show that our method achieves better performance in terms of formation distance error  $\bar{e}_{dist}$  and formation similarity error  $\bar{e}_{sim}$ . Furthermore, our success rate is the most satisfactory in each test environment since the automatic formation transitioning of our method improves adaptability considerably in planning formation motion. Additionally, in obstacle-rich scenarios like Fig. 7(b), our approach saves more runtime than Quan's [5] since we consider multiple robots as an entire unit, reducing the detours caused by individual robots dispersedly passing through obstacles, which also benefits the communication stability of multi-robot formation. As shown in Fig. 8, our algorithm enabled four

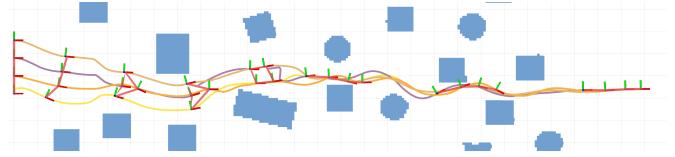
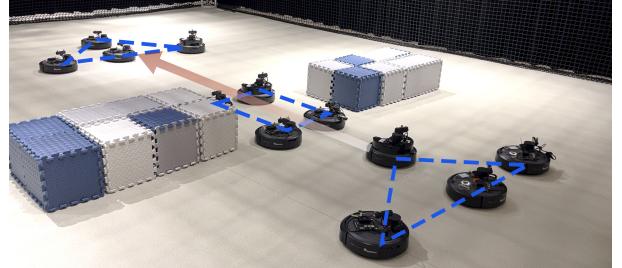


Fig. 8. Robot trajectories of formation reconfiguration experiment in the complex environment of Fig. 7(b).



(a) The formation reconfiguration in a narrow corridor.

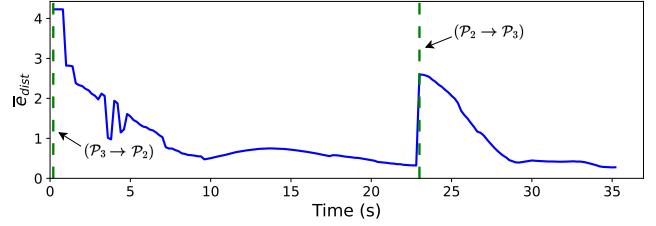
(b) Formation distance error  $\bar{e}_{dist}$  in formation reconfiguration over time.

Fig. 9. Four robots adaptively change formation patterns based on environmental information to pass through a narrow corridor.

vehicles to safely traverse the scenario depicted in Fig. 7(b) through multiple formation reconfigurations. To further validate the algorithm's performance in more complex and larger scenes containing various obstacles, a twelve-robot formation experiment was conducted in a complex environment; see supplementary video [27] for details.

3) *Formation Control Performance*: We validated the real-time performance of formation convergence for up to 15 robots in the obstacle-free scene. However, as the task assignment algorithm minimizes only the total travel distance within the formation reconfiguration, an increase in the number of robots results in a higher density of neighboring robots. This leads to more frequent activation of constraint Eq. (8f), thereby reducing solver efficiency and further impacting the system's real-time performance. And we showed that our method can find the best formation under given conditions, whether the formation library is large or small. More details are in the supplementary video [27].

## B. Real-world Experiments

We conduct real-world experiments using four mobile robots to validate the feasibility of our method. The software is running on the onboard computers (Jetson Orin Nano) in real-time. The state of each robot can be obtained via the OptiTrack system (motion capture system), and the robot swarm shares formation information via the wireless router.

The first experiment, as shown in Fig. 1, involves four mobile robots dynamically switching formations in real-time according to the maximum feasible width along a known

global path. Here, Fig. 1(a) and Fig. 1(b) represent collision-free formation control for the first 203 seconds and from 203 to 301 seconds, respectively. The scenario with various feasible widths demonstrates that our algorithm can quickly converge to the appropriate formation.

In the second experiment, as shown in Fig. 9(a), four mobile robots automatically choose the appropriate formation to safely pass through the narrow corridor. The initial formation shape is a triangle. When detecting that the width of the narrow corridor is smaller than the current formation's width, the formation shape changes to a rectangle. Finally, after moving away from the narrow corridor, the formation autonomously restores its original triangular shape. The formation distance error curve over time is shown in Fig. 9(b). The tests demonstrate that our method can effectively achieve the automatic expansion and compaction of the multi-robot formation adaptive to obstacle-rich environments.

## VIII. CONCLUSIONS

In this letter, we presented an adaptive formation reconfiguration algorithm for multi-robot systems operating in obstacle-rich environments. The proposed approach leverages a temporary leader robot to determine the maximum feasible width by solving a QP problem, which is then used to adjust the local goal through unconstrained optimization based on surrounding environmental information. The optimal formation pattern is automatically selected according to the computed feasible width. Our task allocation strategy, coupled with a consensus-based MPC formation controller, ensures efficient and precise formation reconfiguration. Simulation results demonstrate that our method significantly reduces formation errors compared to state-of-the-art approaches. Real-world experiments also validate the practicality and efficiency of our approach in complex, confined environments. Our future work primarily focuses on enhancing robustness, particularly by deploying more robots in large-scale outdoor environments with complex terrains, and designing multi-objective task allocation strategies to improve the scalability of the formation.

## REFERENCES

- [1] 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.
- [2] G. Sun, R. Zhou, Z. Ma, Y. Li, R. Groß, Z. Chen, and S. Zhao, "Mean-shift exploration in shape assembly of robot swarms," *Nature Communications*, vol. 14, no. 1, p. 3476, 2023.
- [3] R. Tallamraju, E. Price, R. Ludwig, K. Karlapalem, H. H. Bülthoff, M. J. Black, and A. Ahmad, "Active perception based formation control for multiple aerial vehicles," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 4491–4498, 2019.
- [4] P. Peng, W. Dong, G. Chen, and X. Zhu, "Obstacle avoidance of resilient uav swarm formation with active sensing system in the dense environment," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 10529–10535.
- [5] L. Quan, L. Yin, C. Xu, and F. Gao, "Distributed swarm trajectory optimization for formation flight in dense environments," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 4979–4985.
- [6] M. A. Karimi, D. C. Bonham, E. Lopez, A. Srivastava, and M. Spenko, "Slam and shape estimation for soft robots," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2023, pp. 9133–9138.
- [7] M. A. Karimi, V. Alizadehyazdi, H. M. Jaeger, and M. Spenko, "A self-reconfigurable variable-stiffness soft robot based on boundary-constrained modular units," *IEEE Transactions on Robotics*, vol. 38, no. 2, pp. 810–821, 2021.
- [8] K. Tanaka, Q. Zhou, A. Srivastava, and M. Spenko, "A localization framework for boundary constrained soft robots," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2023, pp. 598–603.
- [9] S. Park and S.-M. Lee, "Formation reconfiguration control with collision avoidance of nonholonomic mobile robots," *IEEE Robotics and Automation Letters*, 2023.
- [10] K.-K. Oh, M.-C. Park, and H.-S. Ahn, "A survey of multi-agent formation control," *Automatica*, vol. 53, pp. 424–440, 2015.
- [11] J. Chen, D. Sun, J. Yang, and H. Chen, "Leader-follower formation control of multiple non-holonomic mobile robots incorporating a receding-horizon scheme," *The International Journal of Robotics Research*, vol. 29, no. 6, pp. 727–747, 2010.
- [12] S. Duran and V. Gazi, "Adaptive formation control and target tracking in a class of multi-agent systems," in *Proceedings of the 2010 American Control Conference*. IEEE, 2010, pp. 75–80.
- [13] L. Sabattini, C. Secchi, and C. Fantuzzi, "Arbitrarily shaped formations of mobile robots: artificial potential fields and coordinate transformation," *Autonomous Robots*, vol. 30, pp. 385–397, 2011.
- [14] D. Xue, J. Yao, G. Chen, and Y.-L. Yu, "Formation control of networked multi-agent systems," *IET control theory & applications*, vol. 4, no. 10, pp. 2168–2176, 2010.
- [15] H. Wang and M. Rubenstein, "Shape formation in homogeneous swarms using local task swapping," *IEEE Transactions on Robotics*, vol. 36, no. 3, pp. 597–612, 2020.
- [16] D. Morgan, G. P. Subramanian, S.-J. Chung, and F. Y. Hadaegh, "Swarm assignment and trajectory optimization using variable-swarm, distributed auction assignment and sequential convex programming," *The International Journal of Robotics Research*, vol. 35, no. 10, pp. 1261–1285, 2016.
- [17] J. Alonso-Mora, E. Montijano, M. Schwager, and D. Rus, "Distributed multi-robot formation control among obstacles: A geometric and optimization approach with consensus," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 5356–5363.
- [18] S. Liu, M. Waterson, K. Mohta, K. Sun, S. Bhattacharya, C. J. Taylor, and V. Kumar, "Planning dynamically feasible trajectories for quadrotors using safe flight corridors in 3-d complex environments," *IEEE Robotics and Automation Letters*, vol. 2, no. 3, pp. 1688–1695, 2017.
- [19] J. Chen, T. Liu, and S. Shen, "Online generation of collision-free trajectories for quadrotor flight in unknown cluttered environments," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 1476–1483.
- [20] F. Gao, L. Wang, B. Zhou, X. Zhou, J. Pan, and S. Shen, "Teach-repeat-replan: A complete and robust system for aggressive flight in complex environments," *IEEE Transactions on Robotics*, vol. 36, no. 5, pp. 1526–1545, 2020.
- [21] R. Deits and R. Tedrake, "Computing large convex regions of obstacle-free space through semidefinite programming," in *Algorithmic Foundations of Robotics XI: Selected Contributions of the Eleventh International Workshop on the Algorithmic Foundations of Robotics*. Springer, 2015, pp. 109–124.
- [22] S. Guo, B. Liu, S. Zhang, J. Guo, and C. Wang, "Continuous-time gaussian process trajectory generation for multi-robot formation via probabilistic inference," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 9247–9253.
- [23] G. Peyré, M. Cuturi *et al.*, "Computational optimal transport: With applications to data science," *Foundations and Trends® in Machine Learning*, vol. 11, no. 5-6, pp. 355–607, 2019.
- [24] R. Flamary, N. Courty, A. Gramfort, M. Z. Alaya, A. Boisbunon, S. Chambon, L. Chapel, A. Corenflos, K. Fatras, N. Fournier *et al.*, "Pot: Python optimal transport," *Journal of Machine Learning Research*, vol. 22, no. 78, pp. 1–8, 2021.
- [25] W. Ren, R. W. Beard, and E. M. Atkins, "Information consensus in multivehicle cooperative control," *IEEE Control systems magazine*, vol. 27, no. 2, pp. 71–82, 2007.
- [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.
- [27] Jin Li *et al.*, "Supplementary video for 'DEFORM: Adaptive Formation Reconfiguration of Multi-robot Systems in Confined Environments,'" 2025, [Online]. Available: <https://youtu.be/Gp7ctSpoegl>.