Communication-Efficient Formation Maintenance for Multi-Robot System with a Safety Certificate

Anirudh Aynala, Made Widhi Surya Atman, and Azwirman Gusrialdi

Abstract— **This paper presents a real-time cooperative control algorithm for multi-robot systems to perform formation maintenance and navigate through the environment while avoiding collision with static obstacles and among different group formations of robots. The control algorithm is computed by solving a quadratic program and utilizing the control barrier function to efficiently incorporate multiple control objectives in a unified manner and provide a safety certificate. In addition, a novel method for collision avoidance among different formations of robots is presented which only requires a representative robot in each formation to communicate with each other and thus reduces the required communication between the robots. The cooperative control algorithm is verified using a robotic experimental testbed.**

Index Terms— **Control barrier function, Multi-robot systems, Collision avoidance, Formation maintenance.**

I. INTRODUCTION

Control of multi-robot systems have been heavily researched due to its potential practical applications in different domains including modern day military, environmental monitoring, and industrial sectors. In contrast to a single robot, deploying multi-robot systems is advantageous as it provides robustness to a single robot failure and allows the completion of complex tasks faster and more effective.

Formation control is one of the mostly investigated problems in control of multi-robot systems where the general objective is to drive the robots to achieve a given constraints on their distances [1]. In this paper, we are in particular interested in one class of formation control problems where the robots are required to maintain inter-robot distances so that the formation can maneuver as a single rigid body (also known as rigid formation), motivated by their applications in cooperative load transportation, e.g., in a warehouse [2] and robotic valet parking system [3]. Specifically, the multi-robot systems need to maintain their rigid formation while ensuring the safety by avoiding collision with the obstacles and among different formations of robots.

A variety of approaches (e.g., a method by combining a consensus-based formation control and potential function, a control algorithm based on control barrier function (CBF), a method based on deep reinforcement learning) have been proposed in the literature to achieve some of the previously mentioned control objectives, see the work [2], [4]–[9] just to name a few. However, all the work mentioned above do not consider collision avoidance among different group formations of robots. To the best of our knowledge, formation maintenance with collision avoidance among different group formations has received little attention to date. Note that the work in [10] considers the problem of multi-group collision avoidance for UAV swarm. However, the method cannot be directly applied when the robots in each group are also required to maintain their rigid formation.

This paper proposes a novel cooperative control algorithm for multi-robot systems to navigate (i.e., go to predefined locations) while maintaining their formation and ensuring collision avoidance with static obstacles and among different group formations (Section III). To this end, we utilize the CBF to take into account all the control specifications in a unified manner. Furthermore, by exploiting the fact that the formation moves as a rigid body, a novel method for collision avoidance among different group formations of robots is proposed which only requires communication between a representative robot from each formation, hence reducing the required communication between the robots. The proposed cooperative control algorithm is finally demonstrated and evaluated using a robotic experimental testbed (Section IV).

II. PRELIMINARY AND PROBLEM SETTING

This section briefly introduces the concepts of graph theory and control barrier certificates. Then, we define the problem settings within this paper.

A. Graph Theory

Interaction (e.g., information exchange) between robots in a network can be modelled using an *undirected graph* denoted by $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{A})$ with a set of nodes (representing the robots) $\mathcal{V} = \{1, 2, \ldots, n\}$, a set of edges (links) $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$, and a weighted adjacency matrix $\mathcal{A} \in \mathbb{R}^{n \times n}$. An edge $(i, j) \in \mathcal{E}$ denotes that node i can retrieve node j's information, and vice-versa. A *path* is a sequence of nodes $(i_1, i_2, \ldots, i_p), p > 1$, such that $(i_j, i_{j+1}) \in \mathcal{E}$ for $j = 1, \ldots, p-1$. An undirected graph is said to be *connected* if there is a path between any pair of distinct nodes. A non-negative entries on the i-th row and j-th column of $\mathcal{A}$ denotes the respective weights of the edges $(i, j) \in \mathcal{E}$.

A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with $n \geq 2$ is *rigid* iff there exists a subset $\mathcal{E}_R \subseteq \mathcal{E}$ with $|\mathcal{E}_R| = 2n - 3$ such that for graph $\mathcal{G}_R = (\mathcal{V}, \mathcal{E}_R)$, each *(Laman)* subgraph $\mathcal{H} = (\mathcal{V}_H, \mathcal{E}_H)$ of $\mathcal{G}_R$ with $\mathcal{V}_H \subseteq \mathcal{V}$, $|\mathcal{V}_H| \geq 2$ and $\mathcal{E}_H := \{(i, j) \in \mathcal{E}_R \mid i, j \in \mathcal{V}_H\}$, satisfies the property of $|\mathcal{E}_H| = 2|\mathcal{V}_H| - 3$, see [11]. Here, the graph $\mathcal{G}_R$ is called a *minimally rigid graph* where no

This work was supported by the Academy of Finland under the Academy Project decision number 330073. The results in this paper are based on the results of the M.Sc. thesis of the first author.

All authors are with the Faculty of Engineering and Natural Sciences, Tampere University, Tampere 33014, Finland. Emails: anirudhaynala18@gmail.com, widhi.atman@tuni.fi, and azwirman.gusrialdi@tuni.fi

edges can be removed without violating the rigidity of the resulting graph. Next, consider a formation of robots where the predefined inter-robot distances are given by the weighted adjacency matrix $\mathcal{A}$ associated with graph $\mathcal{G}$. If the graph $\mathcal{G}$ is rigid, then every motion of the robots which maintains the inter-robot distances given by matrix $\mathcal{A}$ will also preserve the distances of all pairs of robots. In this case, the formation is called a rigid formation [12].

B. *Control Barrier Function and QP-based Controller*

Let us consider a control affine system formulated as

$$\dot{x} = f(x) + g(x)u \tag{1}$$

with the state $x \in \mathcal{D} \subseteq \mathbb{R}^n$, control input $u \in \mathcal{U} \subseteq \mathbb{R}^m$, and vector fields f and g, which are assumed to be Lipschitz continuous. Let us introduce a continuously differentiable function $h : \mathcal{D} \subset \mathbb{R}^n \to \mathbb{R}$ and its zero superlevel set $\mathcal{C}$, i.e., $\mathcal{C} = \{x \in \mathbb{R}^n \mid h(x) \geq 0\}$. Then for the control system (1), h is a *control barrier function (CBF)* [13] if there exists a locally Lipschitz extended class $\mathcal{K}$ function α, such that

$$\sup_{u \in U} \left[L_f h(x) + L_g h(x) u\right] + \alpha(h(x)) \geq 0, \tag{2}$$

for all x in the interior of the set $\mathcal{C}$. The terms $L_f h(x)$ and $L_g h(x)$ represent the Lie derivative of h along $f(x)$ and $g(x)$, respectively.

Let us assume that $x(t)$ has a unique solution on $[t_0, t_1]$. Then, any Lipschitz continuous controller u which satisfies the constraint (2) will render the set $\mathcal{C}$ *forward invariant*, namely for every $x(t_0) \in \mathcal{C}$, the inclusion $x(t) \in \mathcal{C}$ holds for all $t \in [t_0, t_1]$. This property allows the CBF to be utilized with the goal of ensuring safety, i.e., not leaving a *safe set*. Alternatively, in the presence of existing nominal controller u_{nom}, one can utilize the following quadratic programming (QP)-based controller:

$$\begin{aligned} u(x) = \underset{u \in \mathcal{U}}{\operatorname{argmin}} \ & \|u - u_{\text{nom}}\|^2 \\ \text{s.t. } & L_f h(x) + L_g h(x) u \geq -\alpha(h(x)), \end{aligned}$$

to find the closest control input to the nominal one that satisfies the forward invariance of the set $\mathcal{C}$.

C. *Problem Setting*

Consider a scenario of multiple groups of mobile robots navigating in the 2D working area given by the (x, y)-plane. Let $\mathcal{C} = \{1, \ldots, c\}$ denote the set of identifiers for a total c groups of mobile robots. Each group $k \in \mathcal{C}$ comprises of n_k number of mobile robots, where each mobile robot can be identified by $i \in \mathcal{V}_k := \{1, \ldots, n_k\}$. The position of each mobile robot $i \in \mathcal{V}_k,\ k \in \mathcal{C}$ is denoted by $p_i = [x_i\ y_i]^T$. Furthermore, it is assumed that the position of each robot is updated according to the following kinematic model:

$$\dot{p_i} = u_i \tag{3}$$

where $u_i \in \mathbb{R}^2$ is the velocity input to be designed. Next, assume that the working area consists of o number of obstacles given by the set $\mathcal{O} := \{1, \ldots, o\}$. Without loss of generality, we assume that the obstacle's information can be retrieved by each robot's on-board computation, e.g., by using the on-board sensors or transmitted from a centralized ground station.

The objective of this paper is to design the velocity input u_i for the individual mobile robot $i \in \mathcal{V}_k$ so that it can cooperate with the robots within its own group $k \in \mathcal{C}$ to reach a given goal position for the formation while fulfilling the following subtasks:

1) *Formation maintenance:* A predefined set of robots pair within each group need to maintain a fixed inter-robot distance (with small tolerance to incorporate e.g., sensors noises).
2) *Obstacle avoidance:* Each robot ensures safe navigation without colliding to static obstacles.
3) *Group collision avoidance:* Each group of robots ensures safe navigation without colliding with other groups of robots.

III. COOPERATIVE CONTROL ALGORITHM WITH A SAFETY CERTIFICATE

In this section, we present a novel cooperative control algorithm by utilizing a CBF to incorporate all the subtask requirements in a unified manner while moving towards the goal position. In addition, a novel method for collision avoidance among different groups with reduced communication is also presented. We begin by providing the detailed specifications for fulfillment of each subtask in the subsequent subsections and then followed by presenting the proposed quadratic programming (QP)-based cooperative control algorithm.

A. *Formation Maintenance*

Assume that the formation for each group $k \in \mathcal{C}$ that needs to be maintained is given by a rigid graph $\mathcal{G}_k = (\mathcal{V}_k, \mathcal{E}_k, \mathcal{A}_k)$. Let us denote the i-th row and j-th element of weighted adjacency matrix $\mathcal{A}_k$ as a_{ij}^k, where $a_{ij}^k > 0$ if $(i, j) \in \mathcal{E}_k$, otherwise $a_{ij}^k = 0$. The value of $a_{ij}^k > 0$ denotes the required distance that robots i and j need to maintain while navigating to their goal positions, i.e., $\|p_i - p_j\| = a_{ij}^k,\ \forall (i, j) \in \mathcal{E}_k$.

Since it is mostly infeasible in practice (e.g., due to localization error) for two robots to maintain an exact equal distance while maneuvering, we introduce a small tolerance $\epsilon > 0$ to the inter-robot distance constraint. That is, a formation is maintained if each pair of robots $(i, j) \in \mathcal{E}_k$ is able to maintain their desired predefined distance within the value of $a_{ij}^k \pm \epsilon$. We define the following functions to formulate the specification for the formation maintenance:

$$h_{ij}^{\text{fmu}}(p_i) = -\|p_i - p_j\|^2 + (a_{ij}^k + \epsilon)^2 \geq 0 \tag{4}$$

$$h_{ij}^{\text{fml}}(p_i) = \|p_i - p_j\|^2 - (a_{ij}^k - \epsilon)^2 \geq 0 \tag{5}$$

for each pair $(i, j) \in \mathcal{E}_k$. It is clear that the distance between robots i and j will be within the range of $(a_{ij}^k - \epsilon)$ and $(a_{ij}^k + \epsilon)$ if both inqualities $h_{ij}^{\text{fml}}(p_i) \geq 0$ and $h_{ij}^{\text{fmu}}(p_i) \geq 0$ hold. Analogously, the formation for group k will be maintained if $h_{ij}^{\text{fml}}(p_i) \geq 0$ and $h_{ij}^{\text{fmu}}(p_i) \geq 0$ for all $(i, j) \in \mathcal{E}_k$.

Note that information of the relative distance to the other robot, i.e., $(p_i - p_j)$ can be retrieved by on-board sensors or via information exchange through communication network whose topology is similar to $\mathcal{G}_k$.

B. Obstacle Avoidance

Next, we present the specification to ensure collision avoidance between individual robot $i \in \mathcal{V}_k,\ k \in \mathcal{C}$ and the static obstacles. To this end, we assume that the position and size of each obstacle $s \in \mathcal{O}$ can be approximated by a circle denoted by its center $p_s^{\text{obs}} \in \mathbb{R}^2$ and its radius R_s. Let us define the following function:

$$h_{is}^{\text{avo}}(p_i) = \left\| p_i - p_s^{\text{obs}} \right\|^2 - R_s^2 \geq 0 \quad (6)$$

for each robot $i \in \mathcal{V}_k,\ k \in \mathcal{C}$ and each obstacle $s \in \mathcal{O}$. Observe that the distance between robot i and the obstacle s is ensured to be larger than R_s if $h_{is}^{\text{avo}}(p_i) \geq 0$. Hence, all the robots $i \in \mathcal{V}_k, \forall k \in \mathcal{C}$ will not collide with any obstacles as long as $h_{is}^{\text{avo}}(p_i) \geq 0$ for all $s \in \mathcal{O}$.

C. Group Collision Avoidance

Finally, let us proceed with the specification to ensure collision avoidance between different group formations of robots. To this end, we exploit the fact, which stems from the specification for formation maintenance, that each formation $k \in \mathcal{C}$ can only move or rotate as a rigid body. This means that the shape of each formation can be approximated by e.g., a circle or an ellipse. In this paper, we use an ellipse to over-approximate the formation's shape as it is less conservative compared to using a circle and it provides more maneuverability as the ellipse can also rotate.

To be more precise, for each robot $i \in \mathcal{V}_k$ in group $k \in \mathcal{C}$, all the other robots within group $m \in \mathcal{C}, m \neq k$ can be seen as an ellipse obstacle with its center $p_m^{\text{c}} = [x_m^{\text{c}}\ y_m^{\text{c}}]^T$, the length of its semi minor axis b_m, the length of its semi major axis a_m, and its rotational angle θ_m. Hence, the objective can be recast as ensuring all the robots in each formation to navigate while avoiding the ellipses which approximate the other formations, see Fig. 1. Let us now define the rotational matrix $R(\theta_m)$ as

$$R(\theta_m) = \begin{bmatrix} \cos\theta_m & -\sin\theta_m \\ \sin\theta_m & \cos\theta_m \end{bmatrix}.$$

For each robot $i \in \mathcal{V}_k$ and a distinct group of $k \in \mathcal{C}$ and $m \in \mathcal{C}$, i.e. $k \neq m$, we can then define the following function to specify the group collision avoidance:

$$h_{im}^{\text{avf}}(p_i) = \left\| \begin{bmatrix} a_m^{-1} & 0 \\ 0 & b_m^{-1} \end{bmatrix} R(-\theta_m)(p_i - p_m^{\text{c}}) \right\|^2 - 1 \geq 0. \quad (7)$$

Observe that the robot $i \in \mathcal{V}_k$ will be outside of the ellipse defined by parameters p_m^{c}, θ_m, a_m, and b_m, if $h_{im}^{\text{avf}}(p_i) \geq 0$ always hold. As a result, we can ensure that the robots in group k will not collide with the robots in group m. Note that the formulation in (7) can also be used to replace (6) in order to approximate the static obstacle as an ellipse.

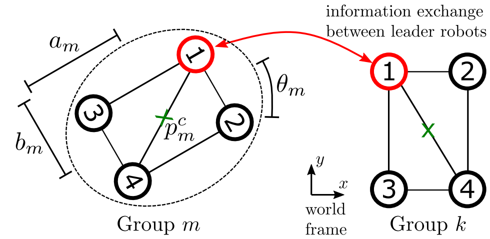


Fig. 1: Proposed approach for group collision avoidance

Let us define the information related to the group m's ellipse as $\zeta_m := \left[(p_m^{\text{c}})^T\ a_m\ b_m\ \theta_m\right]^T \in \mathcal{R}^5$. Furthermore, recall that the robots in the mth group can communicate via the communication network $\mathcal{G}_m$. In the remaining of the subsection, we describe how the robots in group m can distributively compute the information ζ_m and how all robots in the kth group can receive the information ζ_m for $m \neq k$.

First, observe that since the formation configuration (i.e., the values of a_m and b_m) is predetermined and the robots within the mth group are required to maintain their formation, the values of a_m and b_m do not change over time and thus a_m, b_m can be assigned to all robots in the mth group before their deployment. Conversely, the value of p_m^{c} and θ_m are required to be estimated/updated over time by all the robots in the mth formation. The centroid p_m^{c} is given by $p_m^{\text{c}} = \frac{1}{n_m}\sum_{i \in \mathcal{V}_m} p_i$ which can be estimated distributively by each robot $i \in \mathcal{V}_m$ via the communication network and by using the dynamic averaging consensus algorithm [14] whose reference signal of the individual robot is set to p_i. In addition, the value of θ_m can be computed according to $\theta_m = \text{atan2}\,(y_i - y_m^{\text{c}}, x_i - x_m^{\text{c}}) - \theta_i^{\text{form}}$ where θ_i^{form} denote robot i's angular coordinate with respect to formation's centroid and ellipse's major axis.

In the following, we describe how all the robots in the kth group can receive the information ζ_m from groups $m \neq k$ including the required communication. First, a robot in each group $k \in \mathcal{C}$ is selected as the leader robot responsible for exchanging ζ_k with leaders of the other groups as illustrated in Fig. 1. When $n_k \leq 4$, the leader robot can be simply selected from a robot that is a common neighbor to all other robots in its own group, e.g., robot 1 in Fig. 1. The retrieved ζ_m from other groups $m \neq k$ is then distributed to other robots within its own group k using one of the following methods. If the leader robot is connected to all other robots in the group, as for the case $n_k \leq 4$, the leader robot can send directly the information to the remaining robots in its formation. Otherwise, the other robots in the formation, including the leader, use dynamic average consensus algorithm [14] with a minor modification to propagate the information ζ_m which runs at a higher rate than the update rate of the cooperative control algorithm u_i. Specifically, for executing the dynamic average consensus the reference signal is set to $n_k\zeta_m$ for the leader robot and $\mathbf{0}_5$ for the rest of the robots in the group.

D. Distributed QP-based Controller

In the previous subsections, the specifications of each subtask have been formulated in the form of constraint

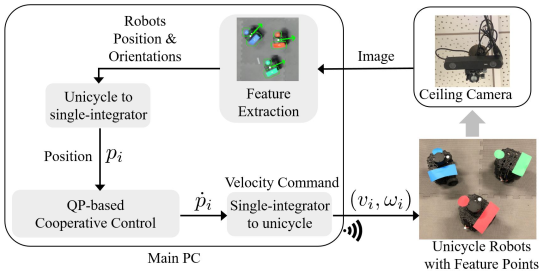


Fig. 2: Experimental system.

functions (4)–(7). Next, we design a nominal control $\hat{u}_i$ to bring the individual robot $i \in \mathcal{V}_k,\ k \in \mathcal{C}$ to its goal position and without considering the other subtasks. The nominal controller can be obtained for example through a simple PID control or the existing path-planning algorithms. Given a nominal control $\hat{u}_i$, the control design objective is then reformulated as to find the control input u_i that closely tracks the nominal control $\hat{u}_i$ and satisfies inequalities (4)–(7).

Following the same procedure as in [15], [16], it can be shown that all the inequalities (4)–(7) hold with the input selected to satisfy the constraints formulated by CBFs [13], [17]. Accordingly, for each robot $i \in \mathcal{V}_k,\ k \in \mathcal{C}$, all the specifications are satisfied if u_i is selected so that

$$\left(\frac{\partial h_{ij}^{\text{fmu}}}{\partial p_i}\right)^T u_i + \beta_1(h_{ij}^{\text{fmu}}(p_i)) \geq 0, \quad \forall (i,j) \in \mathcal{E}^k \tag{8}$$

$$\left(\frac{\partial h_{ij}^{\text{fml}}}{\partial p_i}\right)^T u_i + \beta_2(h_{ij}^{\text{fml}}(p_i)) \geq 0, \quad \forall (i,j) \in \mathcal{E}^k \tag{9}$$

$$\left(\frac{\partial h_{is}^{\text{avo}}}{\partial p_i}\right)^T u_i + \beta_3(h_{is}^{\text{avo}}(p_i)) \geq 0, \quad \forall s \in \mathcal{O} \tag{10}$$

$$\left(\frac{\partial h_{im}^{\text{avf}}(p_i)}{\partial p_i}\right)^T u_i + \beta_4(h_{im}^{\text{avf}}(p_i)) \geq 0, \quad \forall m \in \mathcal{C} \setminus k \tag{11}$$

hold with locally Lipschitz extended class $\mathcal{K}$ function β_1, β_2, β_3, and β_4. For the sake of simplicity, one can choose the same $\mathcal{K}$ function, for example by employing $\beta_1(x) = \beta_2(x) = \beta_3(x) = \beta_4(x) = \gamma x$ with a positive gain $\gamma > 0$. Finally, the control input u_i is then computed according to

$$\begin{aligned} u_i = \underset{u_i^* \in \mathbb{R}^2}{\arg\min}\ & \|u_i^* - \hat{u}_i\|^2 \\ \text{s.t. constraints } & (8)\text{–}(11) \end{aligned}. \tag{12}$$

Therefore, all the specifications are satisfied if optimization problem (12) is feasible. To this end, we leave the rigorous analysis for the existence of a solution as a future work. Remark that the proposed controller requires that the initial positions of all robots to fulfil all specifications. This can be achieved by implementing a go-to-goal position controller before solving (12) so that all the robots converge to the predefined formation.

IV. EXPERIMENTAL VERIFICATION

In this section, we demonstrate and evaluate the proposed cooperative control algorithm with a safety certificate using a planar robotic experimental testbed.

A. Experiment Setup

The experimental setup is shown in Fig. 2 with the experiment field spans 5.3m×3m. We utilize TurtleBot3 Burger robots with different color tags attached on top of each robot. A remote PC with Intel i7 processor, NVIDIA Quadro P1000 and Ubuntu 20.04 LTS operating system takes in the image data using a ZED 2 camera and performs feature extraction to estimate the robots' state (i.e., position and orientation). Hence, each robot has access to its global position and exchanges this information with their neighboring robots. The origin, i.e. location $[0\ 0]^T$, is set in the middle of the field which corresponds to the center of image data, shown as red $+$ sign in Fig. 3 and Fig. 4. The distributed QP-based control algorithm for each robot is then computed in the remote PC using the simulated information exchange between the robots. The computed control input are then sent to individual robot via Wi-Fi through ROS messages. While it is technically possible to implement the computation directly on the robot's on-board computer, the current setup eases the communication between each individual robot.

Since the single integrator input (3) is not directly implementable on the unicycle-like TurtleBot3 Burger robot, we apply the transformation based on a near identity diffeomorphism [18] resulting in the following relationship

$$\begin{bmatrix} v_i \\ \omega_i \end{bmatrix} = \begin{bmatrix} \cos(\phi_i) & \sin(\phi_i) \\ -\frac{1}{\ell}\sin(\phi_i) & \frac{1}{\ell}\cos(\phi_i) \end{bmatrix} u_i \tag{13}$$

where u_i is computed from (12), v_i, ω_i denote the linear and angular velocity of the i-th TurtleBot3 Burger robot, ϕ_i is the orientation of robot i, and $\ell \in \mathbb{R}$ is the distance of point p_i ahead of the middle point between the robot's wheels.

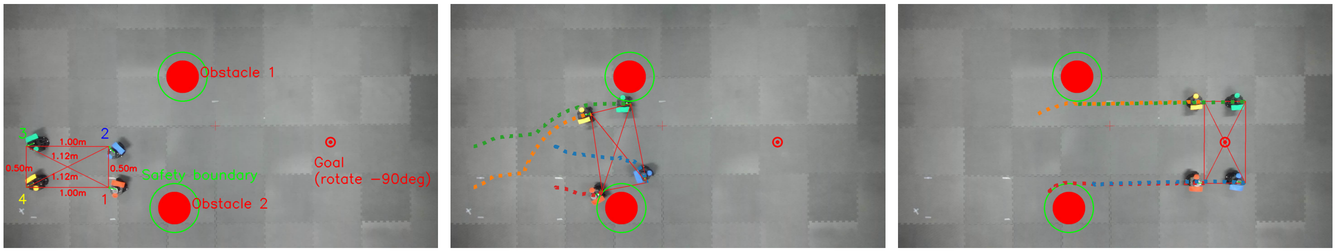
For both scenarios considered in the experiment, the nominal controller $\hat{u}_i$ used to drive the robots to their goal positions are given by the following P-controller:

$$\hat{u}_i = k_p(G_i - p_i), \qquad k_p > 0 \tag{14}$$

where k_p denotes the proportional gain and G_i denotes the i-th robot's goal position. In the experiments we set $k_p = 0.8$ and saturate the nominal input within $||\hat{u}_i||_2 \leq 0.1$m/s. The distance of look ahead point is set as $\ell = 0.06$m. The $\mathcal{K}$ functions for the formation maintenance and the obstacle avoidance are set as $\beta_1(x) = \beta_2(x) = \beta_3(x) = 1000x^3$, while the $\mathcal{K}$ function for group collision avoidance is set as $\beta_4(x) = 0.1x^3$.

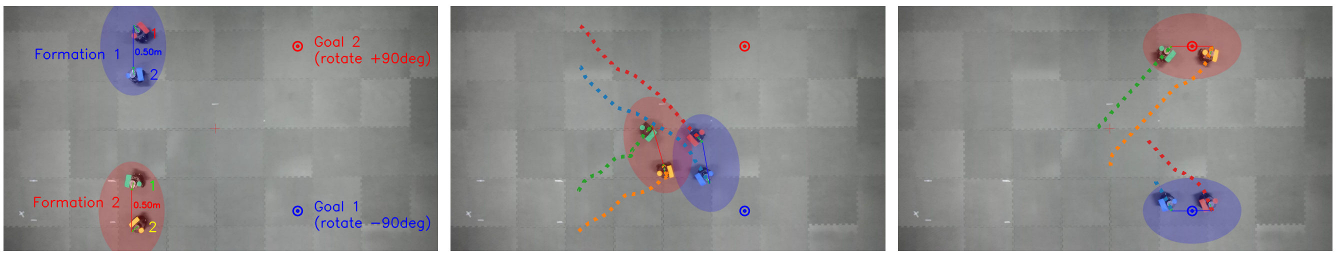
B. Scenario 1: Four-robot Formation with Static Obstacles

The main objective is to navigate a group of four robots to their goal positions while maintaining their formation and avoiding the static obstacles. For simplicity, the information regarding the obstacles position is assumed to be known. The obstacles are located at $p_1^{\text{obs}} = [-0.4\ 0.6]^T$m and $p_2^{\text{obs}} = [-0.5\ \ -1.0]^T$m with their radius set as 0.2m. Note that by using the mapping in (13) we ensure the fulfilment of the specifications for the point in ℓ distance ahead of the robot. In order to guarantee that the robots do not collide with the



(a) $t = 0s$ (b) $t = 22s$ (c) $t = 44s$

Fig. 3: Snapshots of robot formation avoiding obstacles. The dashed colored line shows the trajectory of the robots between each snapshot (for the last 22s). The video of the experiment can be viewed in `https://youtu.be/Ke9bf71z-pQ`.



(a) $t = 0s$ (b) $t = 22s$ (c) $t = 44s$

Fig. 4: Snapshots of two-robot formations avoiding each other. The dashed colored line shows the trajectory of the robots between each snapshot. The video of the experiment can be viewed in `https://youtu.be/HoJuWlrpWqA`.

obstacle, we need to take into account the size of the robot into the safety distance R_s, that is we set R_s to be 0.3m.

The predefined inter-robot distances (in meters) that need to be maintained by all the four robots is shown in Fig. 3a and the tolerance ϵ is set to 0.05m. The formation is initialized with its center located at $[-1.8 \ \ -0.6]^T$m, and the desired goal is for the formation to be rotated $-90°$ with its center at $[1.4 \ \ -0.2]^T$m.

The snapshots of the experiment are shown in Fig. 3 where the robots successfully manage to avoid the obstacles in their way while maintaining the formation and reach the goal position as expected. The specification of formation maintenance can be evaluated by checking the distance between two robots as shown in Fig. 5a. It can be observed that all the distances between two robots are as specified within the given tolerance ϵ. Fig. 5b shows the time series data of the safety function for obstacle avoidance for all the robots. The values of h_{is}^{avo} decreases close to zero as robot $i = 1$ approaches obstacle $s = 2$ around $t =$ 6s to $t =$ 29s and robot $i = 3$ approaches obstacle $s = 1$ around $t =$ 19s to $t =$ 22s. The control input obtained from solving QP in (12) ensures the value of the functions stay positive, that is to avoid the obstacles.

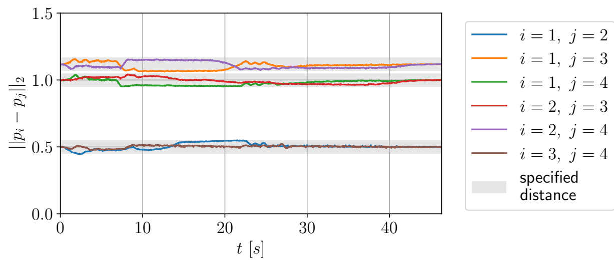
C. Scenario 2: Two-robot Formations

Next, we present the experiment results where two formations of robots avoid each other while navigating to their goal positions. The robots need to maintain their formation and maneuver as a unit to avoid the other group of robots. The distance that need to be maintained by the robots in each formation is equal to 0.5m with the tolerance ϵ is set to 0.05m. The parameters of the ellipse enclosing the formations are set as $a_k = 0.6$m, $b_k = 0.4$m, for $k = 1, 2$. The two formations are initialized with each formation center located at $[-1 \ \ 1]^T$m and $[-1 \ \ -1]^T$m for group 1 and group 2, respectively. The desired goal for group 1 is for the formation to be rotated $-90°$ with its center at $[1 \ \ -1]^T$m. The desired goal for group 2 is for the formation to be rotated $90°$ with its center at $[1 \ \ 1]^T$m. The snapshots of the experiment are shown in Fig. 4.

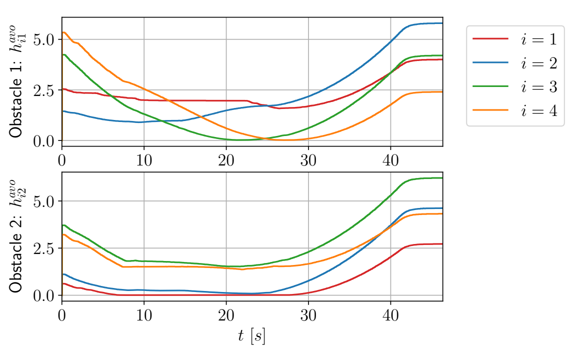
The time series plots of the distance between robots in each group are shown in Fig. 6a where the distances are kept within the specification throughout the experiment. Fig. 6b shows the time series plots of the safety function for avoiding other robot formations for the robots. The safety functions values are close to zero as the formations get closer to each other around $t = 8s$ to $t = 23s$, but the controller obtained from solving (12) keeps their values positive through out the experiment as expected. The snapshots in Fig. 4 show us that the formations successfully avoid each other and reach their respective goal positions.

V. CONCLUSIONS AND FUTURE WORK

A real-time distributed cooperative control algorithm is presented for multi-robot systems to perform formation maintenance and navigate through the environment while avoiding collision with static obstacles and among different group formations of robots. Specifically, we utilize the con-

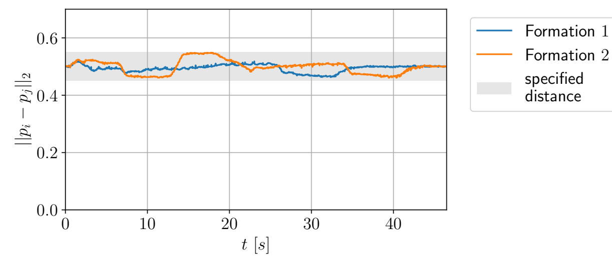


(a) $||p_i - p_j||_2$ for formation maintenance

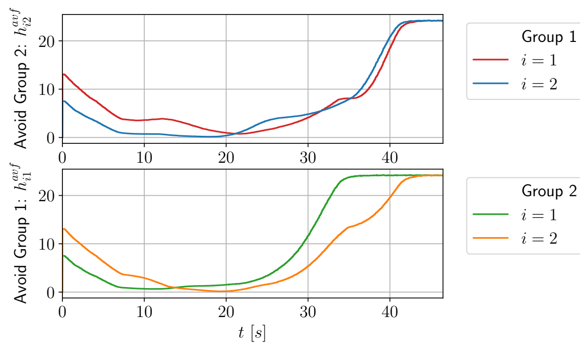


(b) h_{is}^{avo} for obstacle avoidance

Fig. 5: Fulfillment of subtasks in scenario 1.



(a) $||p_i - p_j||_2$ for formation maintenance



(b) h_{is}^{avf} for group collision avoidance

Fig. 6: Fulfillment of subtasks in scenario 2.

trol barrier function to efficiently incorporate multiple control objectives in a unified manner and provide a safety certificate. Furthermore, a novel method for collision avoidance among different formations is proposed which only requires a representative robot in each formation to communicate with each other. Experiments with different scenarios verified the efficacy of the proposed cooperative control algorithm. Future works are aimed towards the extension into 3D space, relaxation of the obstacle's and group formation's shapes as convex polytope, and investigation on the robustness of the proposed cooperative control against communication link and robot's failures.

References

[1] K.-K. Oh, M.-C. Park, and H.-S. Ahn, "A survey of multi-agent formation control," *Automatica*, vol. 53, pp. 424–440, 2015.

[2] S. K. Tse, Y. B. Wong, J. Tang, P. Duan, S. W. W. Leung, and L. Shi, "Relative state formation-based warehouse multi-robot collaborative parcel moving," in *The 4th IEEE International Conference on Industrial Cyber-Physical Systems*, pp. 375–380, IEEE, 2021.

[3] A. Amanatiadis, C. Henschel, B. Birkicht, B. Andel, K. Charalampous, I. Kostavelis, R. May, and A. Gasteratos, "Avert: An autonomous multi-robot system for vehicle extraction and transportation," in *IEEE International Conference on Robotics and Automation*, pp. 1662–1669, IEEE, 2015.

[4] D. Koung, I. Fantoni, O. Kermorgant, and L. Belouaer, "Consensus-based formation control and obstacle avoidance for nonholonomic multi-robot system," in *16th International Conference on Control, Automation, Robotics and Vision*, pp. 92–97, 2020.

[5] A. Mondal, L. Behera, S. R. Sahoo, and A. Shukla, "A novel multi-agent formation control law with collision avoidance," *IEEE/CAA Journal of Automatica Sinica*, vol. 4, no. 3, pp. 558–568, 2017.

[6] S. Mastellone, D. M. Stipanović, C. R. Graunke, K. A. Intlekofer, and M. W. Spong, "Formation control and collision avoidance for multi-agent non-holonomic systems: Theory and experiments," *The International Journal of Robotics Research*, vol. 27, no. 1, pp. 107–126, 2008.

[7] Z. Sui, Z. Pu, J. Yi, and S. Wu, "Formation control with collision avoidance through deep reinforcement learning using model-guided demonstration," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 6, pp. 2358–2372, 2020.

[8] A. Hegde and D. Ghose, "Multi-uav collaborative transportation of payloads with obstacle avoidance," *IEEE Control Systems Letters*, vol. 6, pp. 926–931, 2021.

[9] F. Mehdifar, C. P. Bechlioulis, F. Hashemzadeh, and M. Baradarannia, "Prescribed performance distance-based formation control of multi-agent systems," *Automatica*, vol. 119, p. 109086, 2020.

[10] L. Luo, X. Wang, J. Ma, and Y.-S. Ong, "Grpavoid: Multigroup collision-avoidance control and optimization for uav swarm," *IEEE Transactions on Cybernetics*, 2021.

[11] G. Laman, "On graphs and rigidity of plane skeletal structures," *Journal of Engineering Mathematics*, vol. 4, pp. 331–340, Oct. 1970.

[12] B. D. Anderson, C. Yu, B. Fidan, and J. M. Hendrickx, "Rigid graph control architectures for autonomous formations," *IEEE Control Systems Magazine*, vol. 28, pp. 48–63, Dec. 2008.

[13] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control barrier functions: Theory and applications," in *2019 18th European Control Conference (ECC)*, IEEE, 2019.

[14] S. S. Kia, B. Van Scoy, J. Cortes, R. A. Freeman, K. M. Lynch, and S. Martinez, "Tutorial on Dynamic Average Consensus: The Problem, Its Applications, and the Algorithms," *IEEE Control Systems Magazine*, vol. 39, pp. 40–72, June 2019.

[15] L. Wang, A. D. Ames, and M. Egerstedt, "Safety Barrier Certificates for Collisions-Free Multirobot Systems," *IEEE Transactions on Robotics*, vol. 33, pp. 661–674, June 2017.

[16] M. Egerstedt, J. N. Pauli, G. Notomista, and S. Hutchinson, "Robot ecology: Constraint-based control design for long duration autonomy," *Annual Reviews in Control*, vol. 46, pp. 1–7, Jan. 2018.

[17] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, "Control Barrier Function Based Quadratic Programs for Safety Critical Systems," *IEEE Transactions on Automatic Control*, vol. 62, pp. 3861–3876, Aug. 2017.

[18] S. Wilson, P. Glotfelter, L. Wang, S. Mayya, G. Notomista, M. Mote, and M. Egerstedt, "The robotarium: Globally impactful opportunities, challenges, and lessons learned in remote-access, distributed control of multirobot systems," *IEEE Control Systems Magazine*, vol. 40, no. 1, pp. 26–44, 2020.