Scalable distributed algorithms for multi-robot near-optimal motion planning ⋆

Guoxiang Zhao [a], Minghui Zhu [a]

[a] *School of Electrical Engineering and Computer Science, Pennsylvania State University, University Park, PA, 16802 USA*

Abstract

This paper investigates a class of motion planning problems where multiple unicycle robots desire to safely reach their respective goal regions with minimal traveling times. We present a distributed algorithm which integrates decoupled optimal feedback planning and distributed conflict resolution. Collision avoidance and finite-time arrival at the goal regions are formally guaranteed. Further, the computational complexity of the proposed algorithm is independent of the robot number. A set of simulations are conduct to verify the scalability and optimality of the proposed algorithm.

Key words: robotic motion planning, multi-robot optimal coordination, scalability

1 Introduction

The recent booming capability of mobile computing and surging development of advanced communication technologies stimulate the emergency of mobile robotic networks. Extended from single-robot systems, multi-robot systems exceed their antecedents in the following two aspects mentioned in Cao et al. (1997): 1) missions, such as those that require visits to multiple locations, may be hard or even impossible for a single robot but are attainable to a robotic network; 2) for some problems, commissioning multiple simple robots can be cheaper and more flexible than building a single powerful robot. Recent successful applications include Amazon's Kiva system in D'Andrea (2012), Uber's driverless taxis and Intel's drone show on 2018 Winter Olympics.

Motion planning of mobile robotic networks is a fundamental problem, where a set of control policies is identified to command a team of mobile robots from initial states to their respective goal sets, while abiding dynamic constraints and environmental rules. It is well-known that the motion planning problem for a single robot is computationally challenging. In particular, the generalized mover's problem in Reif (1979) is shown to be PSPACE-hard in degrees of freedom. Most single-robot planning algorithms fall into three categories: cell decomposition (Brooks & Lozano-Perez, 1985, Schwartz & Sharir, 1983), roadmap Canny (1988) and potential field (Khatib, 1986, Koren & Borenstein, 1991). As pointed in Karaman & Frazzoli (2011), these methods can be impractical in high-dimensional spaces as the first two methods suffer from heavy computations while potential field planners can be trapped in local minima. Sampling-based algorithms are shown to be effective in addressing computational inefficiency. Rapidly-exploring Random Tree (RRT) algorithm in LaValle & Kuffner (2001) can quickly return a feasible path. However, it is shown in Karaman & Frazzoli (2011) that RRT is not asymptotically optimal; i.e., the length of the shortest path returned by RRT is worse than the global minimum with probability one. Searching for the optimal path is harder than searching for a feasible path. For example, the shortest path problem in the 3D space is NP-hard in the number of obstacles as shown in Canny & Reif (1987). Recently, Karaman & Frazzoli (2011) proposes RRT*, which is both computationally efficient and asymptotically optimal. That is, RRT* only scales constantly with respect to (w.r.t.) RRT in computational complexity but guarantees that the returned solution converges to the optimum almost surely.

Motion planning of multiple robots is even more computationally challenging. Centralized planners; e.g., Schwartz & Sharir (1983), treat all the robots as a single entity. As pointed out in Schwartz & Sharir (1983), centralized planners are not scalable since the worst-case computational complexity grows exponentially

⋆ This work was partially supported by the grants NSF ECCS-1710859 and NSF CNS 1830390.

Email addresses: gfz5014@psu.edu (Guoxiang Zhao), muz16@psu.edu (Minghui Zhu).

as the robot number. Consequently, many alternative planners have been proposed to address the scalability issue. In decoupled planning (Kant & Zucker, 1986, Siméon et al., 2002), each robot independently plans its path and a coordination scheme is used to resolve conflicts. In priority planning (Buckley, 1989, Erdmann & Lozano-Perez, 1987), each robot is assigned with a unique priority such that (s.t.) lower-ranked robots make compromises for higher-ranked ones. These algorithms achieve scalability at the expense of optimality. In Zhu et al. (2014) and Jha et al. (2015), game theory is leveraged to coordinate multiple robots and convergence to Nash equilibria is ensured. However, Nash equilibria may not be socially optimal and the algorithms rely on steering functions, whose solutions are only known for a limited class of dynamic systems as pointed in Karaman & Frazzoli (2013), including double integrator, Dubins' car and the Reeds-Shepp car.

Distributed control of mobile robots has been extensively studied in recent years (Bullo et al., 2009, Yan et al., 2013, Zhu & Martínez, 2015). A widely used idea is to encode the network-wide goal of interest as a team objective function and the control laws of individual robots are derived from their partial derivatives of the team objective function. The synthesized control laws only require local information exchanges and thus are scalable w.r.t. network expansion. This idea has been successfully applied to many multi-robot missions; e.g., consensus Cao et al. (2008), formation control (Panagou & Kumar, 2014, Ren & Beard, 2008), vehicle routing Frazzoli & Bullo (2004) and sensor deployment (Cortés et al., 2004, Schwager et al., 2009). These algorithms usually use Lyapunov analysis to guarantee asymptotic convergence to certain critical points of the team objective functions. However, they do not guarantee that aggregate running costs; e.g., travelling times and energy consumptions, are minimized.

Our work is also relevant to receding-horizon control or model predictive control (MPC). A (centralized) MPC controller solves a finite-horizon open-loop optimal control problem and applies the first element of the computed control sequence. As addressed in Bemporad & Morari (1999), Mayne et al. (2000), MPC is capable of handling control and state constraints and it is robust to uncertainties and disturbances. It is shown in Grune & Rantzer (2008) that the N-horizon policy of a centralized MPC controller can consistently approximate the infinite-horizon optimal policy in terms of their (infinite-horizon) value functions as the search horizon N goes to infinity. Recently, distributed MPC has been widely applied to coordinate multiple mobile robots for various missions including vehicle platooning in Dunbar & Caveney (2011), Li et al. (2016), formation control in Dunbar & Murray (2006), Keviczky et al. (2007), Zhu & Martínez (2013) and trajectory optimization in Kuwata & How (2010), Kuwata et al. (2007). Compared to centralized MPC, distributed MPC can better cope with large-scale and complex systems, especially in computational efficiency, robustness to failures and simplification of system maintenance Mayne (2014). Distributed MPC requires robots to solve a collective finite-horizon optimal control problem at each time instant. The problem could be difficult to solve when the robot dynamics is nonlinear and state and input constraints are imposed. To reduce online computational burden, this paper proposes a coordination scheme where each robot computes its optimal policy offline ignoring other robots and quickly computes its evasive velocity with respect to other objects in its online execution.

Contribution statement: In this paper, we propose a distributed algorithm to coordinate a fleet of unicycle robots in a cluttered environment. The algorithm integrates decoupled optimal feedback planning and distributed conflict resolution. In particular, each robot individually plans its optimal motion offline and evades from other objects within its sensing range in online execution. Since the computational complexity of the decoupled planning is independent of the robot number and conflict resolution is fully distributed, the algorithm is scalable w.r.t. the robot number. When the robot-obstacle density is not high, the optimality of individual planners can lead to the near-optimality of the distributed algorithm; i.e., its loss in optimality is minor compared to its centralized counterpart. Safe and finite-time arrival at the goal regions is formally ensured. Simulations confirm the scalability and near-optimality of the proposed algorithm. Preliminary results are included in Zhao & Zhu (2019) where all the proofs are removed due to space limitation.

2 Problem Formulation

Consider a team of mobile robots, labeled by $\mathcal{V} \triangleq \{1, \ldots, N\}$, in an environment equipped with a global coordinate frame with basis (e^x, e^y). Each robot is modeled as a circular disk with radius $d_r > 0$. The state of robot $i \in \mathcal{V}$ is denoted by $q_i \triangleq [x_i \ y_i \ \theta_i]^T \in \mathcal{C}_i \triangleq X_i \times \mathbb{R}$ consisting of the position $r_i \triangleq [x_i \ y_i]^T \in X_i$ and the orientation θ_i w.r.t. the global coordinate frame (e^x, e^y). Each robot is equipped with a body-attached coordinate frame with basis $e_i^x \triangleq [\cos\theta_i \ \sin\theta_i]^T$ and $e_i^y \triangleq [-\sin\theta_i \ \cos\theta_i]^T$. See Fig. 1 for illustration. Each

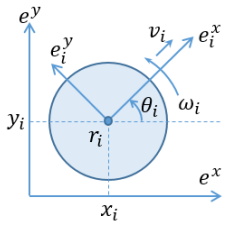


Fig. 1. Robot i in the global coordinate.

robot i is a unicycle and its motion w.r.t. (e^x, e^y) is governed by:

$$\dot{q}_i = f_i(q_i, u_i) \triangleq \begin{bmatrix} \cos\theta_i & 0 \\ \sin\theta_i & 0 \\ 0 & 1 \end{bmatrix} u_i, \tag{1}$$

where $u_i \triangleq [v_i \ \omega_i]^T \in U_i = \mathbb{R}^2$ is the control of robot i consisting of the linear velocity v_i and the angular velocity ω_i. As (Atınç et al., 2020, Bullo et al., 2009, Egerstedt & Hu, 2002, Ma et al., 1999, Panagou & Kumar, 2014), both v_i and ω_i could be any real number. We refer to the vector $v_i e_i^x$ as the velocity of robot i.

The environment is packed with circular obstacles $\mathcal{V}^O$. Obstacle $\ell \in \mathcal{V}^O$ is centered at r_ℓ^O and has radius d_o; i.e., $X_\ell^O \triangleq \{p \in \mathbb{R}^2 | \|p - r_\ell^O\| \leq d_o\}$, where $\|\cdot\|$ is the 2-norm. A robot aims at reaching a goal region while keeping a certain distance away from the obstacles and other robots to ensure its safety. The goal region for robot $i \in \mathcal{V}$ is denoted by $X_i^G \subseteq X_i^F \triangleq \{p \in X_i | \|p - p'\| \geq d_r, \forall p' \in X_\ell^O \text{ and } \ell \in \mathcal{V}^O\}$, the free space for robot i, while the set of goal states is denoted by $\mathcal{C}_i^G \triangleq X_i^G \times \mathbb{R}$. The minimum safety distances between two robots and a robot-obstacle pair are denoted by $d_s \triangleq 2d_r$ and $d_{so} \triangleq d_r + d_o$ respectively.

Each robot has limited capabilities to communicate with nearby robots and sense local environment. Specifically, each robot can broadcast messages to nearby robots within a communication range $d_c > d_s$. As in Bullo et al. (2009), we define the communication graph at position r by $\mathcal{G}^C(r, d_c) \triangleq (\mathcal{V}, \mathcal{E}^C(r, d_c))$, where $r \triangleq \prod_{i \in \mathcal{V}} r_i$ is the collection of all robots' positions, $\mathcal{E}^C(r, d_c) \triangleq \{(i, j) | \|r_i - r_j\| \leq d_c, i \neq j\}$ is the collection of communication links and each link (i, j) indicates that the messages of robot i can be transmitted to robot j. Notice that the communication graph is undirected. The communication range d_c may be omitted when no confusion is caused. Similarly, each robot can measure the states of other robots or the positions of static obstacles within a sensing range $d_{sense} > \max\{d_s, d_{so}\}$. Throughout the paper, we assume the sensing range equals to the communication range; i.e., $d_c = d_{sense}$. These two quantities are larger than d_s, and the gap will be used to avoid inter-robot collisions. The set of neighboring robots of robot i is defined by $\mathcal{N}_i(r, d_c) \triangleq \{j \in \mathcal{V} | (j, i) \in \mathcal{E}^C(r, d_c)\}$, whose messages can be received by robot i and whose states can be measured by robot i. Denote the neighboring robots whose indices are larger than robot i by $\mathcal{N}_i^+(r, d_c) \triangleq \{j \in \mathcal{N}_i(r, d_c) | j > i\}$.

The problem of interest in this paper is to synthesize distributed controllers under which the robots can safely reach their respective goal regions and meanwhile their traveling times are minimized. As mentioned in the introduction, the problem is computationally prohibited if perfect optimality is targeted at. Instead, we will trade minor optimality loss for scalability. That is, we will design distributed controllers which are scalable w.r.t. the robot number and meanwhile near-optimal. Collision avoidance and finite time arrivals at goal regions will be formally guaranteed. Scalability and near-optimality will be validated using MATLAB simulations.

3 Notation and Assumption

Throughout the paper, denote the vector from the center of object j to the center of robot i by $r_{ji} \triangleq r_i - r_j$. Specially, when object j is an obstacle, $r_{ji} \triangleq r_i - r_j^O$. The direction of r_{ji} is defined as $e_{ji} \triangleq \frac{r_{ji}}{\|r_{ji}\|}$. Denote the angle between a given vector $p \in \mathbb{R}^2$ and e^x by $Arg(p)$.

Throughout this paper, an assumption on the reachability of the initial state is imposed. We introduce the concept of reachability before the assumption.

Definition 3.1. *A set $\mathcal{C}_i^G \subseteq \mathcal{C}_i$ is reachable from $q_{i,1} \in \mathcal{C}_i$ within finite time if there exists a state $q_{i,2} \in \mathcal{C}_i^G$ and a controller $u_i : [0, t_i] \to U_i$ for some $t_i \in [0, +\infty)$ s.t. the trajectory of system (1) from initial state $q_{i,1}$ under the controller u_i satisfies $q_i(t_i) = q_{i,2}$ and $\|r_i(\tau) - r_\ell^O\| \geq d_{so}, \forall \ell \in \mathcal{V}^O$, $\|r_i(\tau) - r_j\| \geq d_s, \forall r_j \in X_j^G, j > i$ and $\tau \in [0, t_i]$.*

Define Reach($\mathcal{C}_i^G$) as the set of states from which the goal region $\mathcal{C}_i^G$ is reachable within finite time.

Assumption 3.1. *For each robot $i \in \mathcal{V}$, $q_{i,0} \in$ Reach($\mathcal{C}_i^G$).*

Algorithm 1 Distributed optimal controller for robot i

while $r_i \notin X_i^G$ **do**
  Measure the states of nearby robots in $\mathcal{N}_i^+(r, d_c)$;
  $(v_i^A, \omega_i^A) = \mathcal{A}_i(q_i)$;
  if $\mathcal{N}_i^+(r, d_c) \neq \emptyset$ **then**
    if `receive`$_{i,j^*}(u_{j^*})$==`TRUE` **then**
      if $(e_i^x)^T e_{j^* i} \neq 0$ **then**
        `singularity=FALSE`;
        $(v_i, \omega_i) = \mathcal{R}_i^1(q_i, v_i^A, j^*, q_{j^*}, u_{j^*})$;
      else
        $(v_i, \omega_i) = \mathcal{R}_i^2(q_i, j^*, q_{j^*}, u_{j^*})$;
      end if
    end if
  else
    `singularity=FALSE`;
    $(v_i, \omega_i) = (v_i^A, \omega_i^A)$;
  end if
  Broadcast (v_i, ω_i) to robots in $\mathcal{N}_i(r, d_c)$;
end while

4 Algorithm Statement

In this section, we propose a distributed hybrid controller to solve the problem of interest. The corresponding pseudo codes are summarized in Algorithm 1. Robot i's controller distinguishes three cases:

- Case 1: it is moving in a robot-free environment.
- Case 2: it detects a neighboring robot with larger index coming from front or back;
- Case 3: it senses a robot with larger index approaching sideways.

Case 1 indicates that robot i can safely proceed and only needs to avoid static obstacles and the control is given in lines 3 and 15. Case 2 shows that robot i needs to steer away immediately and the control is given in line 8. Case 3 implies that robot i needs to move away immediately and avoid computational singularity at the same time and the control is given in line 10. Robot i chooses its controller mode by checking out the following two conditions one by one: 1. whether there exists a nearby robot with larger index around robot i as line 4; 2. whether the nearby robot $j^* \in \mathcal{N}_i^+(r, d_c)$ is in singularity condition as line 6. The computation of controller modes in Case 2 and 3 requires the reception of the control of robot j^*. This is accomplished by procedure $\texttt{receive}_{i,j^*} \in \{\texttt{TRUE}, \texttt{FALSE}\}$ defined in line 5. In addition, a variable $\texttt{singularity} \in \{\texttt{TRUE}, \texttt{FALSE}\}$ is leveraged to facilitate the computation of controller mode in Case 3 and its details will be discussed later.

At the end of the iteration, robot i broadcasts its control to all nearby robots as line 17. We now proceed to explain each case in details.

Case 1: Robot-free environment and evasion from static obstacles

Robot i is associated with attractive controller $\mathcal{A}_i$ which can command robot i from any state in $\text{Reach}(\mathcal{C}_i^G)$ to its goal region safely with minimum time when other robots are ignored. That is, $\mathcal{A}_i$ only solves a single-robot optimal motion planning problem. There are a number of numerical algorithms to synthesize $\mathcal{A}_i$; e.g., Cardaliaguet et al. (1999). Its synthesis is out of scope of this paper.

Case 2: Evasion from robots when $(e_i^x)^T e_{j^*i} \neq 0$ Each robot i at any time only considers its nearest robots with the largest index $j^* \in \mathcal{N}_i^+(r, d_c)$; i.e., $j^* \triangleq \max \arg\min_{j \in \mathcal{N}_i^+(r,d_c)} \|r_i - r_j\|$. Recall that $d_c > d_s$. A sufficient condition to avoid inter-robot collision between i and j^* is that the velocity of robot i along e_{j^*i} equals to that of robot j^*; that is,

$$(v_i e_i^x)^T e_{j^*i} = (v_{j^*} e_{j^*}^x)^T e_{j^*i}. \tag{2}$$

The linear velocity of robot i that satisfies (2) is referred to as the critically evasive linear velocity of robot i w.r.t. robot j^*, and denoted by $v_{i|j^*}^* \triangleq \frac{(v_{j^*} e_{j^*}^x)^T e_{j^*i}}{(e_i^x)^T e_{j^*i}}$. See Figure 2 for $(e_i^x)^T e_{j^*i} > 0$ and Figure 3 for $(e_i^x)^T e_{j^*i} < 0$.

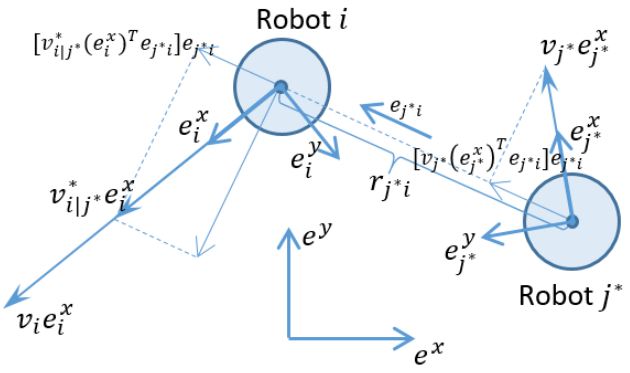


Fig. 2. Robot i evades from robot j^* when robot i is pointing away from robot j^*; i.e., $(e_i^x)^T e_{j^*i} > 0$.

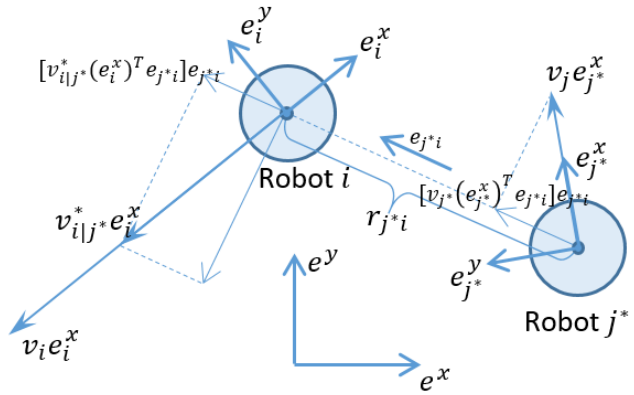


Fig. 3. Robot i evades from robot j^* when robot i is pointing towards robot j^*; i.e., $(e_i^x)^T e_{j^*i} < 0$.

The critically evasive linear velocity $v_{i|j^*}^*$ acts as the boundary of the set of feasible linear velocities for robot i. Specifically, if robot j^* shows up behind robot i; i.e., $(e_i^x)^T e_{j^*i} > 0$ as Figure 2, robot i may need to accelerate to avoid rear-end collision; i.e., $v_i \geq v_{i|j^*}^*$. If robot j^* appears in front of robot i; i.e., $(e_i^x)^T e_{j^*i} < 0$ as Figure 3, robot i needs to slow down or even turn around; i.e., $v_i \leq v_{i|j^*}^*$. Then a candidate for v_i is v_i^A, the linear velocity term of $\mathcal{A}_i(q_i)$, which indicates the optimal linear velocity for robot i to reach its goal region at q_i. Robot $i \in \mathcal{V}$ chooses v_i as the one from v_i^A and $v_{i|j^*}^*$ whose magnitude is larger. As for its angular velocity ω_i, robot i applies a proportional control to steer its angle θ_i to $Arg(r_{j^*i})$, the angle of the direction r_{j^*i} s.t. it points away from robot j^*. The repulsive controller

$\mathcal{R}_i^1 : \mathcal{C}_i \times \prod_{j\in\mathcal{N}_i^+(r,d_c)}(\mathcal{C}_j \times U_j) \to U_i$ is then summarized as follows:

$$\begin{aligned} v_i &= \begin{cases} \max\{v_i^A, v_{i|j^*}^*\}, & \text{if } (e_i^x)^T e_{j^*i} > 0; \\ \min\{v_i^A, v_{i|j^*}^*\}, & \text{if } (e_i^x)^T e_{j^*i} < 0; \end{cases} \\ \omega_i &= k_i(Arg(r_{j^*i}) - \theta_i). \end{aligned} \tag{3}$$

See Algorithm 2 for the corresponding pseudo codes.

Algorithm 2 Repulsive controller $\mathcal{R}_i^1$ for robot $i \in \mathcal{V}$ when $(e_i^x)^T e_{j^*i} \neq 0$

Input: Robot index $i \in \mathcal{V}$ and state q_i, the optimal linear velocity v_i^A, the nearest higher-ranked robot j^* and its state q_{j*} and control u_{j^*}

Output: Control (v_i, ω_i)

$v_{i|j^*}^* = \frac{v_{j^*}(e_{j^*}^x)^T r_{j^*i}}{(e_i^x)^T r_{j^*i}}$;
if $(e_i^x)^T r_{j^*i} > 0$ **then**
  $v_i = \max\{v_i^A, v_{i|j^*}^*\}$;
else
  $v_i = \min\{v_i^A, v_{i|j^*}^*\}$;
end if
$\omega_i = k_i(Arg(r_{j^*i}) - \theta_i)$;

Case 3: Evasion from robots when $(e_i^x)^T e_{j^*i} = 0$

Notice that the aforementioned repulsive controller requires $(e_i^x)^T e_{j^*i} \neq 0$. In other words, when robot j^* is approaching robot i at a direction perpendicular to the orientation of robot i and robot j^* has already contacted robot i, robot i cannot actively avoid the collision. A solution to solve this dilemma is to alert robot i in advance and command it to steer away. Recall that robot i cannot actively avoid the collision when the velocity of robot i w.r.t. robot j^*; i.e., $v_i e_i^x - v_{j^*} e_{j^*}^x$, is pointing to robot j^*. For this case, condition $(e_i^x)^T e_{j^*i} = 0$ implies that the distance between the two robots will decrease. See Figure 4 for illustration.

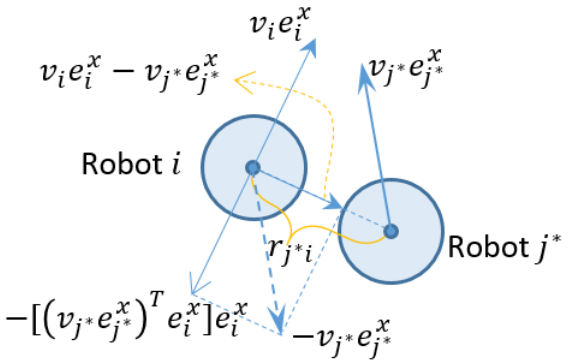


Fig. 4. Robot i cannot evade from robot j^* when $(e_i^x)^T e_{j^*i} = 0$ and robot j^* is moving to robot i.

Notice that as long as the condition $(e_i^x)^T e_{j^*i} = 0$ does not hold, robot i can instantly evade from robot j^* by applying repulsive controller $\mathcal{R}_i^1$. To invalidate the condition $(e_i^x)^T e_{j^*i} = 0$, a strategy for robot i is to take a linear velocity larger than $(v_{j^*} e_{j^*}^x)^T e_i^x$ when $\|r_{j^*} - r_i\| > d_s$. More specifically, the repulsive controller $\mathcal{R}_i^2 : \mathcal{C}_i \times \prod_{j\in\mathcal{N}_i^+(r,d_c)}(\mathcal{C}_j \times U_j) \to U_i$ is shown below:

$$\begin{aligned} v_i =& (v_{j^*} e_{j^*}^x)^T e_i^x + \delta_i (v_{j^*} e_{j^*}^x)^T e_i^y; \\ \omega_i =& 0, \end{aligned} \tag{4}$$

where $\delta_i \triangleq [(\frac{\|r_{j^*i}(t_i)\|}{d_s})^2 - 1]^{-\frac{1}{2}} + 1$ is the piecewise constant evading coefficient and t_i is the most recent switching instant when robot i activates the repulsive controller $\mathcal{R}_i^2$. For example, robot i applies (4) on time interval $[t_1, t_2]$ for some $t_2 > t_1 \geq 0$ and applies other controller mode before t_1. Then throughout the application of (4) on $[t_1, t_2]$, δ_i is a constant and its computation only depends on $\|r_{j^*i}(t_1)\|$. In the second term of v_i in (4), δ_i ensures that the distance from robot j^* to the ray of $v_i e_i^x - v_{j^*} e_{j^*}^x$ is larger than the safety distance d_s. See Figure 5 for illustration. Denote the angle between $v_i e_i^x - v_{j^*} e_{j^*}^x$ and $-e_{j^*i}$ by ψ and the distance from robot j^* to $v_i e_i^x - v_{j^*} e_{j^*}^x$ by $D > d_s$. In addition, let $O_i A \triangleq -(v_{j^*} e_{j^*}^x)^T e_{j^*i} e_{j^*i}$ be the projection of $-v_{j^*} e_{j^*}^x$ onto e_{j^*i}, $O_i B \triangleq v_i e_i^x - v_{j^*} e_{j^*}^x$ be the relative velocity of robot i w.r.t. robot j^* and C be the projection of O_j onto the ray $v_i e_i^x - v_{j^*} e_{j^*}^x$. Decompose $v_{j^*} e_{j^*}^x$ onto e_i^x and e_{j^*i}, we have $O_i B = v_i e_i^x - (v_{j^*} e_{j^*}^x)^T e_i^x e_i^x - (v_{j^*} e_{j^*}^x)^T e_{j^*i} e_{j^*i}$. Then $AB = O_i B - O_i A = v_i e_i^x - (v_{j^*} e_{j^*}^x)^T e_i^x e_i^x$, and plugging (4) into it, we have $|AB| = |\delta_i (v_{j^*} e_{j^*}^x)^T e_i^y|$. Therefore, it holds that $\sin\psi = \frac{|AB|}{|O_i B|} = \frac{|\delta_i (v_{j^*} e_{j^*}^x)^T e_i^y|}{\|v_i e_i^x - v_{j^*} e_{j^*}^x\|}$. Since $\triangle O_i C O_{j^*}$ and $\triangle O_i A B$ are similar, we also have $\sin\psi = \frac{|O_{j^*} C|}{|O_i O_{j^*}|} = \frac{D}{\|r_{j^*i}(t_i)\|} > \frac{d_s}{\|r_{j^*i}(t_i)\|}$. Then solving for δ_i in terms of d_s and $\|r_{j^*i}(t_i)\|$, we arrive at the value of δ_i.

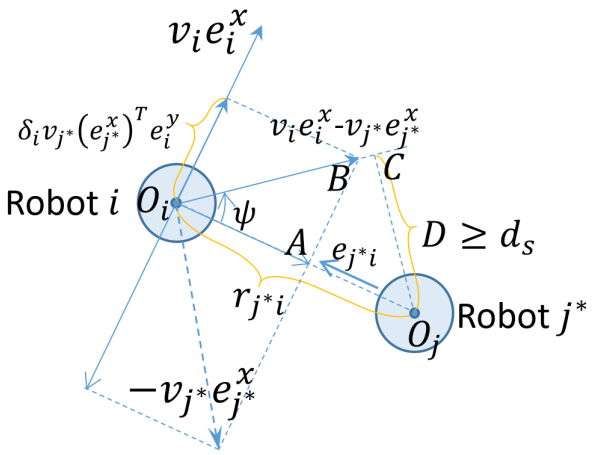


Fig. 5. Robot i evades from robot j^* when $(e_i^x)^T e_{j^*i} = 0$.

The corresponding pseudo codes are shown in Algorithm 3. In the algorithm, a variable `singularity`$\in$ {`TRUE`, `FALSE`} indicates whether the condition $(e_i^x)^T e_{j^*i} = 0$ is activated. When robot i executes controller $\mathcal{R}_i^2$, it checks the value of `singularity`. If `singularity==FALSE`; i.e., the condition $(e_i^x)^T e_{j^*i} = 0$ is not active before, $\mathcal{R}_i^2$ calculates δ_i and sets `singularity` as `TRUE` for record; otherwise; i.e., the condition is already active, $\mathcal{R}_i^2$ simply keeps the last δ_i and proceeds to apply (4). This variable is set `FALSE` as long as condition $(e_i^x)^T e_{j^*i} = 0$ does not hold.

Algorithm 3 Repulsive controller $\mathcal{R}_i^2$ for robot $i \in \mathcal{V}$ when $(e_i^x)^T e_{j^*i} = 0$

Input: Robot index $i \in \mathcal{V}$ and state q_i, the nearest higher-ranked robot j^* and its state q_{j*} and control u_{j^*}

Output: Control (v_i, ω_i)

1: **if** `singularity==FALSE` **then**
2: $\quad \delta_i = 1/\sqrt{(\frac{\|r_{j^*i}(t_i)\|}{d_s})^2 - 1} + 1$;
3: $\quad$ `singularity=TRUE`;
4: **end if**
5: $v_i = (v_{j^*} e_{j^*}^x)^T e_i^x + \delta_i (v_{j^*} e_{j^*}^x)^T e_i^y$;
6: $\omega_i = 0$;

Remark 4.1. *Algorithm 1 is scalable w.r.t. the robot network size. On one hand, the attractive controllers solve single-robot optimal motion planning problems and are synthesized offline in parallel. Parallel computation renders that the computational complexity of numerically computing the optimal controllers is independent of the robot number. On the other hand, real-time decision-making of the robots is spatially distributed and only based on states and controls of nearby robots and obstacles.*

5 Analysis

In this section, we show that Algorithm 1 guarantees collision avoidance and finite time arrival at goal region.

5.1 Scheduling

In this subsection, we analyze the possible deadlock issue in Algorithm 1. Among the controller modes proposed in Section 4, the repulsive controllers $\mathcal{R}_i^1$ and $\mathcal{R}_i^2$ rely on the controls of neighboring robots, and this could possibly cause deadlock in communication and decision making. For any time $t \geq 0$, define the control-dependency graph $\mathcal{G}^D(t) \triangleq \{\mathcal{V}, \mathcal{E}^D(t)\} \subseteq \mathcal{G}^C(t)$, where $\mathcal{E}^D(t) \triangleq \{(j,i) \in \mathcal{E}^C(t) | j \in \mathcal{N}_i^+(r(t), d_c)\}$ is the collection of edges (j,i) representing the dependency of robot i's control on robot j's. This definition follows from line 5 in Algorithm 1. Notice that $\mathcal{G}^D(t)$ is a directed graph. A deadlock happens when there exist two distinguish robots s.t. the control of one robot depends on the other's; i.e., there exist $i, j \in \mathcal{V}, i \neq j$ and two paths $\{(i,i_1),(i_1,i_2),\ldots,(i_m,j)\}$, $\{(j,j_1),(j_1,j_2),\ldots,(j_{m'},i)\} \subseteq \mathcal{E}^D(t)$ for some $m, m' \leq N-2$. That is, a deadlock happens if and only if $\mathcal{G}^D(t)$ is cyclic. The following lemma shows that $\mathcal{G}^D(t)$ is acyclic by contradiction.

Lemma 5.1. *For any time $t \geq 0$, $\mathcal{G}^D(t)$ is acyclic.*

Proof. Fix $t \geq 0$ and pick any robot $i \in \mathcal{V}$. Notice that $\forall (j,i) \in \mathcal{E}^D(t)$, we have $j > i$. Assume there exists a loop $\{(i_1,i_2),(i_2,i_3),\ldots,(i_{m-1},i_m),(i_m,i_1)\} \subseteq \mathcal{E}^D(t)$. Then it holds true that $i_1 > i_2 > \cdots > i_m > i_1$ and thus $i_1 > i_1$, which causes a contradiction. Hence there is no loop in $\mathcal{G}^D(t)$ and the lemma is then proven. □

5.2 Collision avoidance

In this subsection, we show that each robot avoids collisions with other objects under mild assumptions. Define the region where robot i contacts obstacles by $S_i^O \triangleq \{r_i \in X_i | \exists \ell \in \mathcal{V}^O \text{ s.t. } \|r_\ell^O - r_i\| = d_{so}\}$. Denote the closure, boundary and interior of a set S by $\bar{S}$, ∂S and $int(S)$ respectively. Denote the indicator function by $\mathbf{1}_X(x)$ s.t. $\mathbf{1}_X(x) = 1$ if $x \in X$ and $\mathbf{1}_X(x) = 0$ otherwise. The following two assumptions are imposed.

Assumption 5.1. *For all $t \geq 0$ and $i \in \mathcal{V}$, it holds true that $|\mathcal{N}_i^+(r(t), d_c)| + \mathbf{1}_{S_i^O}(r_i(t)) \leq 1$.*

Assumption 5.2. *In the execution of Algorithm 1, no Zeno behavior occurs.*

Assumption 5.1 indicates, if robot i contacts an obstacle, no robot with a larger index is inside the communication/sensing range; if any robot with a larger index is inside the communication/sensing range, robot i cannot address any obstacle. Assumption 5.2 rules out a fundamental issue unique to hybrid systems due to over-abstraction of modeling, Zeno behavior, which stalls computer simulations Zhang et al. (2001) and makes it hard to analyze post-Zeno performance. Roughly speaking, Zeno behavior describes a scenario where a robot switches its controller modes for infinitely many times within finite time. It is a common practice in the analysis of hybrid systems to ignore such ill but rare phenomena and focus on the performance of the controller such as collision avoidance Pallottino et al. (2007), Tomlin et al. (1998), optimality Rantzer & Johansson (2000), Xu & Antsaklis (2004) and stability Fu et al. (2015), Zhang & Shi (2009). In Section 5.4, we will show that Assumptions 5.1 and 5.2 are mild.

We proceed to characterize the number of objects involved in a collision under Assumption 5.1. A list of notations are introduced to facilitate subsequent analysis. Define the safe region by $S \triangleq \{r \in \prod_{i \in \mathcal{V}} X_i | \forall i \in \mathcal{V} \text{ and } \forall j \in \mathcal{V} \setminus \{i\}, \|r_i - r_j\| \geq d_s, \text{ and } \|r_i - r_\ell^O\| \geq d_{so}, \forall \ell \in \mathcal{V}^O\}$; i.e., no inter-robot or robot-obstacle collision occurs. Define the buffered safe region by $\hat{S} \triangleq$

$\{r \in S | |\mathcal{N}_i^+(r, d_c)| + \mathbf{1}_{S_i^O}(r_i) \leq 1, \forall i \in \mathcal{V}\}$ and the set includes the states which are in the safe region and also satisfy Assumption 5.1. The condition $|\mathcal{N}_i^+(r, d_c)| \leq 1$ for $r \in S$ implies that at most one robot with a larger index is inside the buffer ring which is centered at robot i with radii d_s and d_c. For any $r \in S$, $r \notin \hat{S}$ if and only if $|\mathcal{N}_i^+(r, d_c)| + \mathbf{1}_{S_i^O}(r_i) \geq 2$ for some i. Notice that $\partial\hat{S} = \hat{S}_1 \cup \hat{S}_2 \cup \hat{S}_3$, where $\hat{S}_1 \triangleq \{r \in S | \exists i < j \text{ s.t. } \|r_i - r_j\| = d_s\}$ implies that a pair of robots contact each other, $\hat{S}_2 \triangleq \{r \in S | \exists i \in \mathcal{V} \text{ and } \ell \in \mathcal{V}^O \text{ s.t. } \|r_i - r_\ell^O\| = d_{so}\}$ implies that a robot contacts an obstacle and $\hat{S}_3 \triangleq \{r \in S | \exists i < j \text{ and } i < k \text{ s.t. } \|r_i - r_j\| \leq d_c \text{ and } \|r_i - r_k\| = d_c\}$ implies that three robots are on the verge of violating Assumption 5.1. For any $r \in S$, two scenarios can violate Assumption 5.1: $|\mathcal{N}_i(r, d_c)| \geq 2$, or $|\mathcal{N}_i(r, d_c)| \geq 1$ together with $\mathbf{1}_{S_i^O(r_i)} = 1$ for some $i \in \mathcal{V}$. In the above relationships, $\hat{S}_3$ characterizes the first scenario $|\mathcal{N}_i(r, d_c)| \geq 2$. We do not need to consider $\hat{S}_4 \triangleq \{r \in S | \exists i < j \text{ s.t. } \|r_i - r_j\| = d_s \text{ and } \exists \ell \in \mathcal{V}^O \text{ s.t. } \|r_i - r_\ell^O\| = d_{so}\}$ for the second scenario, as $\hat{S}_4 \subseteq \hat{S}_2$. Note that $\hat{S}_1$, $\hat{S}_2$ and $\hat{S}_3$ are not disjoint.

Define $S^O \triangleq \{r \in \partial S | \nexists i < j \in \mathcal{V}, \ell \in \mathcal{V}^O \text{ s.t. } \|r_i - r_\ell^O\| = d_{so}, \|r_i - r_j\| = d_s\}$; i.e., at least one robot contacts another object but no robot simultaneously contacts an obstacle and any robot with a larger index. Similarly, define $S^R \triangleq \{r \in \partial S | \nexists i < j < k \text{ s.t. } \|r_i - r_j\| = \|r_i - r_k\| = d_s\}$; i.e., at least one robot contacts another robot but no robot simultaneously contacts two robots with larger indices. We let $S^A \triangleq \partial\hat{S} \cap int(S) = \hat{S}_3 \cap int(S)$. The next lemma implies collisions under Assumption 5.1 only involve at most two robots.

Lemma 5.2. *It holds that* $\partial\hat{S} \subseteq (S^O \cap S^R) \cup S^A$.

Proof. Since $\partial\hat{S} \subseteq \bar{\hat{S}} \subseteq \bar{S}$, we partition $\partial\hat{S}$ by $\partial\hat{S} \cap int(S) = S^A$ and $\partial\hat{S} \cap \partial S$. To show $\partial\hat{S} \cap \partial S \subseteq S^O \cap S^R$, we show that $\partial S \setminus (\partial S \cap \partial\hat{S}) \supseteq \partial S \setminus (S^O \cap S^R)$. Notice that the left-hand side can be simplified as $\partial S \setminus \partial\hat{S}$, and the right-hand side is equal to $(\partial S \setminus S^O) \cup (\partial S \setminus S^R)$.

We first show $\partial S \setminus \partial\hat{S} \supseteq \partial S \setminus S^O$. Take $r \in \partial S \setminus S^O$. Then there are two different robots $i < j$ which contact each other and robot i contacts an obstacle ℓ. This indicates that $\mathcal{N}_i^+(r, d_c) \supseteq \{j\}$ and $\mathbf{1}_{S_i^O}(r_i) = 1$. Hence $|\mathcal{N}_i^+(r, d_c)| + \mathbf{1}_{S_i^O}(r_i) \geq 2 > 1$, which contradicts Assumption 5.1. Therefore, $r \in \partial S \setminus \partial\hat{S}$. It shows that $\partial S \setminus S^O \subseteq \partial S \setminus \partial\hat{S}$.

Then we show $\partial S \setminus \partial\hat{S} \supseteq \partial S \setminus S^R$. Take $r \in \partial S \setminus S^R$. Then there exist three distinct robots $i < j < k$ and robot i contacts both robot j and k; i.e., $\|r_i - r_j\| = \|r_i - r_k\| = d_s$. This indicates that $\mathcal{N}_i^+(r, d_c) \supseteq \{j, k\}$, which contradicts Assumption 5.1. Therefore, $r \in \partial S \setminus \partial\hat{S}$. It shows that $\partial S \setminus S^R \subseteq \partial S \setminus \partial\hat{S}$.

In summary, we have $\partial\hat{S} \subseteq (S^O \cap S^R) \cup S^A$. □

Collision avoidance of Algorithm 1 is formally guaranteed by the following theorem.

Theorem 5.1. *If Assumption 5.1 and 5.2 hold and* $r(0) \in int(S)$*, then* $r(t) \in S, \forall t \geq 0$.

Proof. Fix $i \in \mathcal{V}$. Define $T_i^{R,1}, T_i^{R,2}, T_i^A \subseteq [0, +\infty)$ as the times when robot i switches to controller $\mathcal{R}_i^1, \mathcal{R}_i^2$ and $\mathcal{A}_i$ respectively. Then define the set of switching instants for robot i by $T_i \triangleq T_i^{R,1} \cup T_i^{R,2} \cup T_i^A$. If T_i is finite, we append $+\infty$ as the last time to switch. Sort the times in T_i in the ascending order by $T_i = \{t_k\}_{k \geq 0}$ with $t_0 = 0$. Notice that Zeno behavior will not occur in the execution of Algorithm 1 as suggested in Assumption 5.2.

Consider the performance of robot i on interval $[t_k, t_{k+1})$ for $t_k \in T_i^{R,1} \cup T_i^{R,2}$ and $t_{k+1} \geq t_k$. When $t_{k+1} = t_k$, an impulsive switch happens and the dwell time is zero. In such case, the time interval $[t_k, t_{k+1})$ reduces to a singleton $\{t_k\}$. We now proceed to prove the collision avoidance by induction. Per the notations in Section 3, we refer to the two robots in the communication/sensing range by $i < j^*$. The proof is to show the time derivative of the squared distance $\dot{D}_{ij^*}$ is non-negative on the boundary $S^R \cup S^O$, where $D_{ij}(t) \triangleq \|r_i(t) - r_j(t)\|^2$.

Claim 5.1. *Let* $t_k \in T_i^{R,1}$*. If for any* $j \in \{j' \in \mathcal{V} | j' > i\}$*,* $D_{ij}(t_k) > d_s^2$ *and for any* $\ell \in \mathcal{V}^O$*,* $\|r_i(t_k) - r_\ell^O\| > d_{so}$*, then it holds true that* $D_{ij}(t) > d_s^2, \forall j \in \{j' \in \mathcal{V} | j' > i\}$ *and* $\|r_i(t) - r_\ell^O\| > d_{so}, \forall \ell \in \mathcal{V}^O, t \in [t_k, t_{k+1})$.

Proof. We first consider the non-impulsive switch; that is, $t_{k+1} > t_k$. It follows from Assumption 5.1 that $\mathcal{N}_i^+(r(t), d_c) = \{j^*\}$ and $\mathbf{1}_{S_i^O}(r_i(t)) = 0$, for every $t \in [t_k, t_{k+1})$. Then for all $t \in [t_k, t_{k+1})$ and $j \in \{j' \in \mathcal{V} | j' > i, j' \neq j^*\}$, we have $D_{ij}(t) > d_c^2 > d_s^2$; for all $\ell \in \mathcal{V}^O$, we have $\|r_\ell^O - r_i(t)\| > d_{so}$. Fix $t \in [t_k, t_{k+1})$ and focus on $D_{ij^*}(t)$.

The time derivative of D_{ij^*} is $\dot{D}_{ij^*}(t) = 2[(v_i e_i^x)^T r_{j^*i} - (v_{j^*} e_{j^*}^x)^T r_{j^*i}]$. Then it follows from (3) and lines 1 and 3 in Algorithm 2 that when $(e_i^x)^T e_{j^*i} > 0$,

$$\begin{aligned}\dot{D}_{ij^*}(t) \geq & 2[(v_{i|j^*}^* e_i^x)^T r_{j^*i} - (v_{j^*} e_{j^*}^x)^T r_{j^*i}] \\ = & 2[(\frac{(v_{j^*} e_{j^*}^x)^T e_{j^*i}}{(e_i^x)^T e_{j^*i}} e_i^x)^T r_{j^*i} - (v_{j^*} e_{j^*}^x)^T r_{j^*i}] = 0;\end{aligned}$$

when $(e_i^x)^T e_{j^*i} < 0$, same result holds. Then integrate the above equation over $[t_k, t)$ for some $t \in [t_k, t_{k+1})$, we

have $D_{ij^*}(t) = D_{ij^*}(t_k) + \int_{t_k}^{t} \dot{D}_{ij^*}(\tau)d\tau \geq D_{ij^*}(t_k) > d_s^2$, where the last inequality follows from claim statement.

Now we proceed to analyze the safety of robot i when an impulsive switch happens; i.e., $t_k = t_{k+1}$. In such case, the velocity $\dot{q}_i(t_k)$ of robot i may change instantly but its state $q_i(t_k)$ will remain the same. The state of robot j^* will also remain the same. Then it follows from the claim statement that $D_{ij^*} > d_s^2$. This completes the proof of collision avoidance. □

Claim 5.2. *Let $t_k \in T_i^{R,2}$. If for any $j \in \{j' \in \mathcal{V} | j' > i\}$, $D_{ij}(t_k) > d_s^2$ and for any $\ell \in \mathcal{V}^O$, $\|r_\ell^O - r_i(t_k)\| > d_{so}$, then it holds true that $D_{ij}(t) > d_s^2, \forall j \in \{j' \in \mathcal{V} | j' > i\}$ and $\|r_\ell^O - r_i(t)\| \geq d_{so}, \forall \ell \in \mathcal{V}, t \in [t_k, t_{k+1})$.*

Proof. We first consider the non-impulsive switch; that is, $t_{k+1} > t_k$. It again follows from Assumption 5.1 that $\mathcal{N}_i^+(r(t), d_c) = \{j^*\}$ and $\mathbf{1}_{S_i^O}(r_i(t)) = 0, \forall t \in [t_k, t_{k+1})$. Then for all $t \in [t_k, t_{k+1})$ and $j \in \{j' \in \mathcal{V} | j' > i, j' \neq j^*\}$, we have $D_{ij}(t) > d_c^2 > d_s^2$; for all $\ell \in \mathcal{V}^O$, we have $\|r_\ell^O - r_i(t)\| > d_{so}$. We now focus on the distance between robot i and robot j^*; i.e., D_{ij^*}. Fix $t \in [t_k, t_{k+1})$. Since $\|r_{j^*i}(t_k)\| > d_s$, $\delta_i(t_k) > 0$ exists. Moreover, it follows from line 2 of Algorithm 3 that $\delta_i(t) = \delta_i(t_k) > 0$ is constant on $[t_k, t_{k+1})$. Rewrite D_{ij^*} as $D_{ij^*}(t) = D_{ij^*}(t_k) + \int_{t_k}^{t} \dot{D}_{ij^*}(\tau)d\tau$, where $\dot{D}_{ij^*}(\tau) = 2r_{j^*i}^T(\dot{r}_i - \dot{r}_{j^*}) = 2r_{j^*i}^T(v_i e_i^x) - 2r_{j^*i}^T(v_{j^*} e_{j^*}^x)$. Then it follows from (4) and line 5 in Algorithm 3 that

$$\begin{aligned}\dot{D}_{ij^*}(\tau) =& 2r_{j^*i}^T[(v_{j^*}e_{j^*}^x)^T e_i^x e_i^x + \delta_i(t_k)(v_{j^*}e_{j^*}^x)^T e_i^y e_i^x] \\ & - 2r_{j^*i}^T(v_{j^*}e_{j^*}^x) \\ =& 2r_{j^*i}^T[(v_{j^*}e_{j^*}^x)^T e_i^x e_i^x - (v_{j^*}e_{j^*}^x)] \\ & + 2r_{j^*i}^T[\delta_i(t_k)[(v_{j^*}e_{j^*}^x)^T e_i^y]e_i^x].\end{aligned}$$

The aforementioned equation can be simplified by substituting $e_i^x = [\cos\theta_i \ \sin\theta_i]^T$ and $e_i^y = [-\sin\theta_i \ \cos\theta_i]^T$ into $\dot{D}_{ij^*}$ and the following equations hold.

$$\begin{aligned}& (v_{j^*}e_{j^*}^x)^T e_i^x e_i^x - v_{j^*}e_{j^*}^x \\ =& v_{j^*}[(\cos\theta_{j^*}\cos\theta_i + \sin\theta_{j^*}\sin\theta_i)e_i^x - e_{j^*}^x] \\ =& v_{j^*}\begin{bmatrix}\cos\theta_{j^*}\cos^2\theta_i + \sin\theta_{j^*}\sin\theta_i\cos\theta_i - \cos\theta_{j^*} \\ \cos\theta_{j^*}\cos\theta_i\sin\theta_i + \sin\theta_{j^*}\sin^2\theta_i - \sin\theta_{j^*}\end{bmatrix} \\ =& v_{j^*}\begin{bmatrix}-\cos\theta_{j^*}\sin^2\theta_i + \sin\theta_{j^*}\sin\theta_i\cos\theta_i \\ \cos\theta_{j^*}\cos\theta_i\sin\theta_i - \sin\theta_{j^*}\cos^2\theta_i\end{bmatrix} \\ =& v_{j^*}(\cos\theta_{j^*}\sin\theta_i - \sin\theta_{j^*}\cos\theta_i)\begin{bmatrix}-\sin\theta_i \\ \cos\theta_i\end{bmatrix} \\ =& -v_{j^*}\sin(\theta_{j^*} - \theta_i)e_i^y.\end{aligned} \tag{5}$$

$$\begin{aligned}(v_{j^*}e_{j^*}^x)^T e_i^y =& v_{j^*}(-\cos\theta_{j^*}\sin\theta_i + \sin\theta_{j^*}\cos\theta_i) \\ =& v_{j^*}\sin(\theta_{j^*} - \theta_i).\end{aligned} \tag{6}$$

Then we have

$$\begin{aligned}\dot{D}_{ij^*}(\tau) =& -2r_{j^*i}^T[v_{j^*}\sin(\theta_{j^*} - \theta_i)e_i^y] \\ & + 2r_{j^*i}^T[v_{j^*}\delta_i(t_k)\sin(\theta_{j^*} - \theta_i)e_i^x].\end{aligned}$$

Recall that $\omega_i = 0$ in (4) and line 6 of Algorithm 3, which indicates $\theta_i(t) = \theta_i(t_k)$ is constant $\forall t \in [t_k, t_{k+1})$. We explicitly express the dependency on time variable τ in the aforementioned equation and it renders at the following result:

$$\begin{aligned}\dot{D}_{ij^*}(\tau) =& -2v_{j^*}(\tau)\sin(\theta_{j^*}(\tau) - \theta_i(t_k))r_{j^*i}^T(\tau)e_i^y(t_k) \\ & + 2v_{j^*}(\tau)\delta_i(t_k)\sin(\theta_{j^*}(\tau) - \theta_i(t_k))r_{j^*i}^T(\tau)e_i^x(t_k).\end{aligned} \tag{7}$$

We expand $r_{j^*i}(\tau)$ and the following holds:

$$\begin{aligned}& r_{j^*i}(\tau) = r_{j^*i}(t_k) + \int_{t_k}^{\tau}[\dot{r}_i(s) - \dot{r}_{j^*}(s)]ds \\ =& r_{j^*i}(t_k) + \int_{t_k}^{\tau}[v_i(s)e_i^x(t_k) - v_{j^*}(s)e_{j^*}^x(s)]ds \\ =& r_{j^*i}(t_k) + \int_{t_k}^{\tau}[(v_{j^*}(s)e_{j^*}^x(s))^T e_i^x(t_k)e_i^x(t_k) \\ & + \delta_i(t_k)[v_{j^*}(s)e_{j^*}^x(s)]^T e_i^y(t_k)e_i^x(t_k) - v_{j^*}(s)e_{j^*}^x(s)]ds \\ =& r_{j^*i}(t_k) - \int_{t_k}^{\tau} v_{j^*}(s)\sin(\theta_{j^*}(s) - \theta_i(t_k))ds e_i^y(t_k) \\ & + \delta_i(t_k)\int_{t_k}^{\tau} v_{j^*}(s)\sin(\theta_{j^*}(s) - \theta_i(t_k))ds e_i^x(t_k),\end{aligned}$$

where the last equality follows from (5) and (6). Substitute this into (7) and we obtain the following equation.

$$\begin{aligned}& \dot{D}_{ij^*}(\tau) = -2v_{j^*}(\tau)\sin(\theta_{j^*}(\tau) - \theta_i(t_k))[r_{j^*i}^T(t_k)e_i^y(t_k) \\ & - \int_{t_k}^{\tau} v_{j^*}(s)\sin(\theta_{j^*}(s) - \theta_i(t_k))ds(e_i^y(t_k))^T e_i^y(t_k) \\ & + \delta_i(t_k)\int_{t_k}^{\tau} v_{j^*}(s)\sin(\theta_{j^*}(s) - \theta_i(t_k))ds(e_i^x(t_k))^T e_i^y(t_k)] \\ & + 2v_{j^*}(\tau)\delta_i(t_k)\sin(\theta_{j^*}(\tau) - \theta_i(t_k))[r_{j^*i}^T(t_k)e_i^x(t_k) \\ & - \int_{t_k}^{\tau} v_{j^*}(s)\sin(\theta_{j^*}(s) - \theta_i(t_k))ds(e_i^y(t_k))^T e_i^x(t_k) \\ & + \delta_i(t_k)\int_{t_k}^{\tau} v_{j^*}(s)\sin(\theta_{j^*}(s) - \theta_i(t_k))ds(e_i^x(t_k))^T e_i^x(t_k)] \\ & = 2v_{j^*}(\tau)\sin(\theta_{j^*}(\tau) - \theta_i(t_k))[-r_{j^*i}^T(t_k)e_i^y(t_k) \\ & + \int_{t_k}^{\tau} v_{j^*}(s)\sin(\theta_{j^*}(s) - \theta_i(t_k))ds] \\ & + 2\delta_i^2(t_k)v_{j^*}(\tau)\sin(\theta_{j^*}(\tau) - \theta_i(t_k))\end{aligned}$$

$$\int_{t_k}^{\tau} v_{j^*}(s)\sin(\theta_{j^*}(s)-\theta_i(t_k))ds.$$

Notice that the third and fifth terms are equal 0 as a fact of the perpendicularity of $e_i^x(t_k)$ and $e_i^y(t_k)$, and the fourth term is equal 0 because $\mathcal{R}_i^2$ is executed only when $r_{j^*i}^T(t_k)e_i^x(t_k)=0$. Then we integrate both sides over $[t_k,t)$ for some $t\in[t_k,t_{k+1})$ and we have the following result:

$$\begin{aligned}
&D_{ij^*}(t)-D_{ij^*}(t_k)\\
=&-2r_{j^*i}^T(t_k)e_i^y(t_k)\int_{t_k}^{t} v_{j^*}(\tau)\sin(\theta_{j^*}(\tau)-\theta_i(t_k))d\tau\\
&+(\delta_i^2(t_k)+1)[\int_{t_k}^{t} v_{j^*}(\tau)\sin(\theta_{j^*}(\tau)-\theta_i(t_k))d\tau]^2.
\end{aligned}$$

Since $r_{j^*i}^T(t_k)e_i^x(t_k)=0$, $r_{j^*i}(t_k)$ and $e_i^y(t_k)$ are in parallel; in other words, $|2r_{j^*i}^T(t_k)e_i^y(t_k)|=2\|r_{j^*i}(t_k)\|$. Then it holds that

$$\begin{aligned}
D_{ij^*}(t)=&(1+\delta_i^2(t_k))[\int_{t_k}^{t} v_{j^*}(\tau)\sin(\theta_{j^*}(\tau)-\theta_i(t_k))d\tau\\
&-\frac{r_{j^*i}^T(t_k)e_i^y(t_k)}{1+\delta_i^2(t_k)}]^2-\frac{\|r_{j^*i}(t_k)\|^2}{1+\delta_i^2(t_k)}+D_{ij^*}(t_k)\\
\geq&-\frac{\|r_{j^*i}(t_k)\|^2}{1+\delta_i^2(t_k)}+\|r_{j^*i}(t_k)\|^2\\
=&\|r_{j^*i}(t_k)\|^2\frac{\delta_i^2(t_k)}{1+\delta_i^2(t_k)}.
\end{aligned}$$

It follows from (4) that $\delta_i(t_k)=1/\sqrt{(\frac{\|r_{j^*i}(t_k)\|}{d_s})^2-1}+1$ and we have

$$\begin{aligned}
D_{ij^*}(t)\geq&\|r_{j^*i}(t_k)\|^2\frac{1}{1+\frac{1}{\delta_i^2(t_k)}}\\
>&\|r_{j^*i}(t_k)\|^2\frac{1}{1+\frac{1}{(1/\sqrt{(\frac{\|r_{j^*i}(t_k)\|}{d_s})^2-1})^2}}=d_s^2
\end{aligned}$$

for every $t\in[t_k,t_{k+1})$.

Then we consider an impulsive switch happens; i.e., $t_k=t_{k+1}$. Following the arguments in the proof of Claim 5.1, we only need to analyze $D_{ij^*}(t_k)$. Then it follows from the claim statement that $D_{ij^*}>d_s^2$. This completes the proof. □

Finally, we consider the performance of robot i on $[t_k,t_{k+1})$ for $t_k\in T_i^A$.

Claim 5.3. *Let $t_k\in T_i^A$. If for any $j\in\{j'\in\mathcal{V}|j'>i\}$, $D_{ij}(t_k)>d_s^2$ and for any $\ell\in\mathcal{V}^O$, $\|r_\ell^O-r_i(t_k)\|>d_{so}$, it holds true that $D_{ij}(t)>d_s^2,\forall j\in\{j'\in\mathcal{V}|j'>i\}$ and $\|r_\ell^O-r_i(t)\|>d_{so},\forall\ell\in\mathcal{V},t\in[t_k,t_{k+1})$.*

Proof. Then we have $\mathcal{N}_i^+(r,d_c)=\emptyset$. In such a case, robot i only actively avoids collisions with static obstacles, and this is guaranteed by the attractive controller $\mathcal{A}_i$. Therefore, for all $t\in[t_k,t_{k+1})$ and $j\in\{j'\in\mathcal{V}|j'>i,j'\neq j^*\}$, we have $D_{ij}(t)>d_c^2>d_s^2$; for all $\ell\in\mathcal{V}^O$, we have $\|r_\ell^O-r_i(t)\|>d_{so}$. This completes the proof. □

Since the initial condition satisfies $d_{ij}(0)>d_s,\forall j\in\{j'\in\mathcal{V}|j'>i\}$ and $\|r_i(0)-r_\ell^O\|>d_{so},\forall\ell\in\mathcal{V}^O$, it follows from Claims 5.1, 5.2 and 5.3 that robot i does not cause collisions w.r.t. any static obstacle or any higher-ranked robot for all $t\geq 0$. Since this holds for any $i\in\mathcal{V}$, collision avoidance is achieved for every pair of robots and every robot-obstacle pair on time interval $[0,+\infty)$. This completes the proof. □

5.3 Convergence to goal region

The following theorem formally guarantees the finite-time arrivals of all robots at their goal regions; i.e., every robot can reach its goal region within finite time without causing any collisions.

Theorem 5.2. *If Assumptions 3.1, 5.1 and 5.2 hold true and $\|r_i(0)-r_j(0)\|>d_s,\|r_i(0)-r_\ell^O\|>d_{so},\forall j\in\{j'\in\mathcal{V}|j'\neq i\},\ell\in\mathcal{V}^O$, then for system (1) under Algorithm 1, there exists $s_i<+\infty$ for $i\in\mathcal{V}$ s.t. $r_i(s_i)\in X_i^G$ and $\|r_i(\tau)-r_j(\tau)\|>d_s,\|r_i(\tau)-r_\ell^O\|>d_{so},\forall j\in\{j'\in\mathcal{V}|j'\neq i\},\ell\in\mathcal{V}^O$ and $\tau\in[0,s_i]$.*

Proof. Without loss of generality, we assume that the robots reach their goal regions in the order of their indices; i.e., robot N is the first to arrive at its goal region, and then robot $N-1$ and so on. The proof is based on induction on robot index k. Denote the induction hypothesis by $H(k):\forall i\in\{k,\dots,N\}$, there exists $s_i<+\infty$ s.t. $q_i(s_i)\in\mathcal{C}_i^G$, $\|r_i(\tau)-r_j(\tau)\|>d_s,\|r_i(\tau)-r_\ell^O(\tau)\|\geq d_{so},\forall\tau\in[0,s_i),j\in\mathcal{V}\setminus\{i\}$ and $\ell\in\mathcal{V}^O$.

Consider the case $k=N$. Since robot N has the largest index, it only executes its attractive controller $\mathcal{A}_N$. It follows from Assumption 3.1 that the attractive controller $\mathcal{A}_N$ steers robot N to reach its goal region within finite time without colliding with obstacles. In addition, it follows from Theorem 5.1 that for any robot $j<N$, it holds true that $\|r_N(\tau)-r_j(\tau)\|>d_s,\forall\tau\geq 0$. This completes the proof of $H(N)$.

Assume the hypothesis holds down to $k\leq N$. We proceed to show $H(k-1)$ also holds. It follows from $H(k)$ that each robot $i\in\{k,\dots,N\}$ has arrived at its respective goal region by $s_i<+\infty$. Notice that we assume robots arrive at goal regions in the order of their indices; i.e., $s_k\geq s_i,\forall i\in\{k,\dots,N\}$. Then after s_k, robot $k-1$ has the largest index among all robots $\{1,\dots,k-1\}$ which have not reached their goal regions. Hence robot $k-1$ only executes $\mathcal{A}_{k-1}$ after s_k. Since the trajectory

of system (1) is reversible, the initial state $q_{k-1}(0)$ of robot $k-1$ is reachable from $q_{k-1}(s_k)$. Because $q_{k-1}(0) \in Reach(\mathcal{C}_i^G)$, then $q_{k-1}(s_k) \in Reach(\mathcal{C}_i^G)$. Hence, the attractive controller $\mathcal{A}_{k-1}$ steers robot $k-1$ from $q_{k-1}(s_k)$ to reach its goal region at a finite time s_{k-1} without colliding with obstacles. In addition, it follows from Theorem 5.1 that for any robot $j < k-1$, it holds true that $\|r_{k-1}(\tau) - r_j(\tau)\| > d_s, \forall \tau \in [0, s_{k-1}]$. This completes the proof of $H(k-1)$.

Thus the induction holds for all $k \in \{1, \ldots, N\}$ and the whole proof is complete. □

5.4 Justification of Assumptions 5.1 and 5.2

In this subsection, we show that Assumptions 5.1 and 5.2 are mild. We first focus on Assumption 5.1. With slight abuse of notations, we let the buffered safe region under communication/sensing range d be $\hat{S}(d) \triangleq \{r \in S | |\mathcal{N}_i^+(r,d)| + \mathbf{1}_{S_i^O}(r_i) \leq 1, \forall i \in \mathcal{V}\}$. Let μ be the Lebesgue measure.

Lemma 5.3. *It holds true that $\mu(\hat{S}(d))$ is monotonically decreasing in $d \geq d_c$ and $\mu(\hat{S}(d_c)) = \mu(S)$.*

Lemma 5.3 shows that for a communication/sensing range d sufficiently close to d_s, the buffered safe region $\hat{S}(d)$, where Assumption 5.1 holds true, will be very close to the entire safe region S; in other words, the cases where Assumption 5.1 does not hold are negligible. It is worthy to mention that this does not indicate Assumption 5.1 will never be violated in the execution of the algorithm. On the other hand, in the simulations we will observe that Assumption 5.1 is never violated in the execution of Algorithm 1. The following theorem is instrumental to prove Lemma 5.3.

Theorem 5.3 (Sard's theorem in Milnor & Weaver (1997))**.** *Let $g : \mathcal{O} \to \mathbb{R}^n$ be a smooth function s.t. all partial derivatives $\frac{\partial^k g}{\partial x_1^{k_1} \cdots \partial x_m^{k_m}}(x)$ exist and are continuous for each $x \in \mathcal{O}$, $k \in \mathbb{N}$ and $\sum_{i=1}^m k_i = k$, where the domain $\mathcal{O} \subseteq \mathbb{R}^m$ is an open set. Let $C \triangleq \{x \in \mathcal{O} | \text{rank}(J_g(x)) < n\}$, where $J_g(x)$ is the Jacobian matrix of g at $x \in \mathcal{O}$. Then the image $g(C) \subseteq \mathbb{R}^n$ has Lebesgue measure zero.*

The above theorem shows that the measure of the image of points at which the rank of the Jacobian matrix is lower than the dimension of image space, is zero. One example is given as follows. Consider $m = n = 2$ and $g(x) = Ax$, where A has rank 1. Then the image set $g(\mathbb{R}^2)$ is a line and has dimension 1; thus, its measure is zero.

Proof of Lemma 5.3. We first show the monotonicity of $\mu(\hat{S}(d))$ in d. Pick $d_1 > d_2 \geq d_s$. A sufficient condition to show function $\mu(\hat{S}(d))$ is decreasing is to show that $\hat{S}(d_1) \subseteq \hat{S}(d_2)$. Fix $r \in \hat{S}(d_1)$. We now proceed to investigate the nearby objects of each robot $i \in \mathcal{V}$. Based on whether a robot contacts an obstacle, we partition all robots into two groups $\mathcal{V}_1(r) \triangleq \{i \in \mathcal{V} | \mathbf{1}_{S_i^O}(r_i) = 1\}$ and $\mathcal{V}_2(r) \triangleq \{i \in \mathcal{V} | \mathbf{1}_{S_i^O}(r_i) = 0\}$. Then two cases arise:

- Case 1, $i \in \mathcal{V}_1(r)$. Then $|\mathcal{N}_i^+(r,d_1)| = 0$; i.e., $\forall j > i$, we have $\|r_j - r_i\| > d_1 > d_2$. That is, $|\mathcal{N}_i^+(r,d_2)| = 0$. In summary, we have $|\mathcal{N}_i^+(r,d_2)| + \mathbf{1}_{S_i^O}(r_i) = 1$;
- Case 2, $i \in \mathcal{V}_2(r)$. That is, $|\mathcal{N}_i^+(r,d_1)| \leq 1$. Then two more subcases arise:
  - Subcase 1, $|\mathcal{N}_i^+(r,d_1)| = 0$. Then we follow the arguments in Case 1 and conclude that $|\mathcal{N}_i^+(r,d_2)| = 0 \leq 1$;
  - Subcase 2, $|\mathcal{N}_i^+(r,d_1)| = 1$. That is, $\exists j^* > i$ s.t. $\|r_i - r_{j^*}\| \leq d_1$ and $\forall j > i, j \neq j^*$, $\|r_i - r_j\| > d_1 > d_2$. Then it renders at $|\mathcal{N}_i^+(r,d_2)| \leq 1$.

  To summarize, it holds true that $|\mathcal{N}_i^+(r,d_2)| + \mathbf{1}_{S_i^O}(r_i) \leq 1$ for all $i \in \mathcal{V}_2(d_1)$.

Therefore, $|\mathcal{N}_i^+(r,d_2)| + \mathbf{1}_{S_i^O}(r_i) \leq 1$ for all $i \in \mathcal{V}$, which further indicates that $r \in \hat{S}(d_2)$. Since the above holds for all $r \in \hat{S}(d_1)$, it holds true that $\hat{S}(d_1) \subseteq \hat{S}(d_2)$ and the monotonicity of $\mu(\hat{S}(d))$ in d is proven.

Now we proceed to show that $\mu(\hat{S}(d_s)) = \mu(S)$. Define $\Delta S \triangleq S \setminus \hat{S}(d_s) = \{r \in S | \exists i \in \mathcal{V} \text{ s.t. } |\mathcal{N}_i^+(r,d_s)| + \mathbf{1}_{S_i^O}(r_i) \geq 2\}$. Notice that ΔS can be decomposed as a finite union of ΔS_i; i.e., $\Delta S = \bigcup_{i \in \mathcal{V}} \Delta S_i$, where $\Delta S_i \triangleq \{r \in S | |\mathcal{N}_i^+(r,d_s)| + \mathbf{1}_{S_i^O}(r_i) \geq 2\}$. Then equivalently, we can show that each component ΔS_i has measure zero.

We explicitly express ΔS_i by enumerating the nearby objects around robot i as

$$\Delta S_i = \bigcup_{\substack{j_1, j_2 \in \mathcal{V} \\ i < j_1 < j_2}} \Delta S_{ij_1j_2} \cup \bigcup_{\substack{j \in \mathcal{V}, j > i \\ \ell \in \mathcal{V}^O}} \Delta S_{ij\ell}, \tag{8}$$

where $\Delta S_{ij_1j_2} \triangleq \{r \in S | \|r_i - r_{j_1}\| = \|r_i - r_{j_2}\| = d_s\}$ and $\Delta S_{ij\ell} \triangleq \{r \in S | \|r_i - r_j\| = d_s, \|r_i - r_\ell^O\| = d_{so}\}$. The equalities in these definitions are because these sets are subsets of the safe region S, where each element r must satisfy $\|r_i - r_j\| \geq d_s, \forall j \in \mathcal{V}$. Now we proceed to show the measure of each of the above set is zero via Sard's theorem.

Claim 5.4. *For any $i, j_1, j_2 \in \mathcal{V}$ s.t. $i < j_1 < j_2$, $\mu(\Delta S_{ij_1j_2}) = 0$.*

Proof. Without loss of generality, let $i = 1, j_1 = 2$ and $j_2 = 3$. Define an auxiliary set $X_{123} \triangleq \{r \in$

$\mathbb{R}^{2N}|\|r_1-r_2\| = \|r_1-r_3\| = d_s\} \supseteq \Delta S_{123}$ and we are going to show $\mu(X_{123}) = 0$. Notice that $\forall r \in X_{123}$, it holds true that $r_2 = r_1 + d_s e(\phi_2)$ and $r_3 = r_1 + d_s e(\phi_3)$ for some $\phi_2, \phi_3 \in \mathbb{R}$, where $e(\phi) \triangleq [\cos\phi \ \sin\phi]^T$. Then define an open set $\mathcal{O} \triangleq \mathbb{R}^{2N-2}$ and each element $\hat{r} \in \mathcal{O}$ is denoted by $\hat{r} \triangleq [\hat{r}_1^T \ \phi_2 \ \phi_3 \ \hat{r}_4^T \ \cdots \ \hat{r}_N^T]^T$, where $\hat{r}_i \in \mathbb{R}^2$. Then an auxiliary function $g : \mathcal{O} \to \mathbb{R}^{2N}$ is defined as $g(\hat{r}) \triangleq \left[\hat{r}_1^T \ \hat{r}_1^T + d_s e^T(\phi_2) \ \hat{r}_1^T + d_s e^T(\phi_3) \ \hat{r}_4^T \ \cdots \ \hat{r}_N^T\right]^T$. Clearly, g is a smooth map and $g(\mathcal{O}) = X_{123}$. The Jacobian matrix $J_g(\hat{r})$ of g is a $2N \times (2N-2)$ matrix, whose rank is certainly lower than $2N$ for all $\hat{r} \in \mathcal{O}$. Then it follows from Theorem 5.3 that $\mu(X_{123}) = \mu(g(\mathcal{O})) = 0$. Since $\Delta S_{123} \subseteq X_{123}$, it follows from the monotonicity of Lebesgue measure that $\mu(\Delta S_{123}) = 0$. Since this holds true for every $i < j_1 < j_2$, the proof is complete. □

Claim 5.5. *For any $i, j \in \mathcal{V}$, $i < j$ and $\ell \in \mathcal{O}$, $\mu(\Delta S_{ij\ell}) = 0$.*

Proof. The proof mainly follows that of Claim 5.4. Without loss of generality, we fix $i = 1$ and $j = 2$. Define $X_{12\ell} \triangleq \{r \in \mathbb{R}^{2N} | \|r_1 - r_2\| = d_s, \|r_1 - r_\ell^O\| = d_{so}\}$ and we will show that $\mu(X_{12\ell}) = 0$. For any $r \in X_{12\ell}$, rewrite r_1 and r_2 in terms of r_ℓ^O and we have $r_1 = r_\ell^O + d_{so} e(\phi_1)$ and $r_2 = r_1 + d_s e(\phi_2) = r_\ell^O + d_{so} e(\phi_1) + d_s e(\phi_2)$. We still define the same auxiliary set $\mathcal{O} \triangleq \mathbb{R}^{2N-2}$ but each element $\hat{r} \in \mathcal{O}$ is denoted by $\hat{r} \triangleq [\phi_1 \ \phi_2 \ \hat{r}_3^T \ \cdots \ \hat{r}_N^T]^T$. The corresponding auxiliary function $g : \mathcal{O} \to \mathbb{R}^{2N}$ is then defined as $g(\hat{r}) \triangleq \left[(r_\ell^O + d_{so} e(\phi_1))^T \ (r_\ell^O + d_{so} e(\phi_1) + d_s e(\phi_2))^T \ \hat{r}_3^T \ \cdots \ \hat{r}_N^T\right]^T$. This function is smooth and $g(\mathcal{O}) = X_{12\ell}$. Then it follows the arguments towards Claim 5.4 that $\mu(\Delta S_{12\ell}) = 0$. Then the claim is proven. □

Recall that ΔS_i is a finite union of sets shown in (8), all of which are of measure zero shown by Claims 5.4 and 5.5. Therefore, it follows from the monotonicity of Lebesgue measure that

$$\mu(\Delta S_i) \leq \sum_{\substack{j_1, j_2 \in \mathcal{V} \\ i < j_1 < j_2}} \mu(\Delta S_{ij_1j_2}) + \sum_{\substack{j \in \mathcal{V}, j > i \\ \ell \in \mathcal{V}^O}} \mu(\Delta S_{ij\ell}) = 0.$$

Since this holds for all $i \in \mathcal{V}$, we have $\mu(\Delta S) \leq \sum_{i \in \mathcal{V}} \mu(\Delta S_i) = 0$. This completes the proof. □

In the following, we show that Assumption 5.2 is mild. Define the coupled state variable and coupled state space by $q \triangleq \prod_{i \in \mathcal{V}} q_i$ and $\mathcal{C} \triangleq \prod_{i \in \mathcal{V}} X_i \times \mathbb{R}$ respectively. If the switching instants $T_i = \{t_1, t_2, \dots\}$ for robot i converge to a finite time; i.e., $\lim_{k \to +\infty} t_k = t_\infty < +\infty$, the time limit t_∞ is referred to as the Zeno time for robot i and the associated limit point of state variable $\lim_{k \to +\infty} q(t_k) = q_\infty$ is the Zeno point for robot i Zhang et al. (2001). The Zeno set is the set consisting of all Zeno points for each robot i; i.e., $Z \triangleq \bigcup_{i \in \mathcal{V}} Z_i$ and $Z_i \triangleq \{q_\infty \in \mathcal{C} | q_\infty \text{ is a Zeno point for robot } i\}$. The following lemma shows that Z is measure zero.

Lemma 5.4. *If Assumption 5.1 holds, the Lebesgue measure of Z is zero.*

Proof. Fix $i \in \mathcal{V}$. It follows from Assumption 5.1 that within its communication/sensing range, robot i only deals with one higher-ranked robot; in other words, robot i changes its controller mode from or to $\mathcal{A}_i$ only at its communication/sensing boundaries. In addition, controller mode $\mathcal{R}_i^2$ is activated or deactivated for robot i only when a higher-ranked robot is along the side of robot i. Therefore, robot i switches its controller mode only at $\Delta_i \triangleq \Delta_i^1 \cup \Delta_i^2$ under Algorithm 1 as shown in lines (4) and (6) of Algorithm 1, where $\Delta_i^1 \triangleq \{q \in \mathcal{C} | \|r_i - r_{j^*}\| = d_c\}$ and $\Delta_i^2 \triangleq \{q \in \mathcal{C} | \|r_i - r_{j^*}\| \leq d_c \text{ and } (e_i^x)^T r_{j^*i} = 0\}$. Therefore, at each switching instant t_k, we have $q(t_k) \in \Delta_i$ and the Zeno points must lie in Δ_i; i.e., $Z_i \subseteq \Delta_i$.

Notice that $\Delta_i^1 \subseteq \bigcup_{j>i} \Delta_{ij}^1$, where $\Delta_{ij}^1 \triangleq \{q \in \mathcal{C} | \|r_i - r_j\| = d_c\}$. Then to show $\mu(\Delta_i^1) = 0$, it suffices to show that for every $j > i$, $\mu(\Delta_{ij}^1) = 0$.

Claim 5.6. *It holds true that $\mu(\Delta_{ij}^1) = 0, \forall j > i$.*

Proof. The proof leverages Theorem 5.3. Without loss of generality, let $j = N$. Notice that $\forall q \in \Delta_{iN}^1$, it holds true that $r_N = r_i + d_c e(\phi_N)$, where $\phi_N \in \mathbb{R}$ and $e(\phi) \triangleq [\cos\phi \ \sin\phi]^T$. Then define an open set $\mathcal{O} \triangleq \mathbb{R}^{3N-1}$ with each element $\hat{q} \in \mathcal{O}$ denoted by $\hat{q} \triangleq [\hat{q}_1^T \ \cdots \ \hat{q}_i^T \ \cdots \ \hat{q}_{N-1}^T \ \phi_N \ \hat{\theta}_N]^T$. An auxiliary function $g : \mathcal{O} \to \mathbb{R}^{3N}$ is defined as $g(\hat{q}) \triangleq [\hat{q}_1^T \ \cdots \ \hat{q}_i^T \ \cdots \ \hat{q}_{N-1}^T \ \hat{r}_i^T + d_c e^T(\phi_N) \ \hat{\theta}_N]^T$. Clearly, g is a smooth map and $g(\mathcal{O}) = \Delta_{iN}^1$. The Jacobian matrix $J_g(\hat{q})$ of g is a $3N \times (3N-1)$ matrix, and its rank is lower than $3N$ for every $\hat{q} \in \mathcal{O}$. Then it follows from Theorem 5.3 that $\mu(\Delta_{iN}^1) = \mu(g(\mathcal{O})) = 0$. Since this holds for every $j > i$, the proof is complete. □

Moreover, in terms of Δ_i^2, we may also decompose it as a finite sum of simpler sets; i.e., $\Delta_i^2 \subseteq \bigcup_{j>i} \Delta_{ij}^2$, where $\Delta_{ij}^2 \triangleq \{q \in \mathcal{C} | (e_i^x)^T r_{ji} = 0\}$. Then it suffices to show that for every $j > i$, $\mu(\Delta_{ij}^2) = 0$.

Claim 5.7. *It holds true that $\mu(\Delta_{ij}^2) = 0, \forall j > i$.*

Proof. Without loss of generality, let $j = N$. For each $q \in \Delta_{iN}^2$, we have $r_N = r_i + d e_i^y$, where $d \in \mathbb{R}$. Then define an open set $\mathcal{O} \triangleq \mathbb{R}^{3N-1}$ with each element $\hat{q} \in \mathcal{O}$ denoted by $\hat{q} \triangleq [\hat{q}_1^T \ \cdots \ \hat{q}_i^T \ \cdots \ \hat{q}_{N-1}^T \ d \ \hat{\theta}_N]^T$. An auxiliary function $g : \mathcal{O} \to \mathbb{R}^{3N}$ is defined as

$g(\hat{q}) \triangleq [\hat{q}_1^T \ \cdots \ \hat{q}_i^T \ \cdots \ \hat{q}_{N-1}^T \ \hat{r}_i^T + d[e_i^y]^T \ \hat{\theta}_N]^T$. Clearly, g is a smooth map and $g(\mathcal{O}) = \Delta_{iN}^2$. The Jacobian matrix $J_g(\hat{q})$ of g is a $3N \times (3N-1)$ matrix, and its rank is lower than $3N$ for every $\hat{q} \in \mathcal{O}$. Then it follows from Theorem 5.3 that $\mu(\Delta_{iN}^2) = \mu(g(\mathcal{O})) = 0$. Since this holds for every $j > i$, the proof is complete. □

It follows from the monotonicity of the Lebesgue measure that $\mu(Z_i) \leq \mu(\Delta_i) \leq \mu(\Delta_i^1) + \mu(\Delta_i^2) \leq \sum_{i<j} \mu(\Delta_{ij}^1) + \sum_{i<j} \mu(\Delta_{ij}^2) = 0$, where the last equality follows from Claim 5.6 and Claim 5.7. Since the aforementioned arguments hold for all $i \in \mathcal{V}$, then $\mu(Z) \leq \sum_{i \in \mathcal{V}} \mu(Z_i) = 0$ and the proof is complete. □

6 Simulations

In this section, we evaluate the performance of Algorithm 1 through simulations. The simulation environment consists of a single circular obstacle with radius $d_o = 1$ positioned at the origin. At the beginning of each simulation, N unicycle robots modeled as disks with radii $d_r = 0.25$ are uniformly distributed on a circle with radius 10 centered at the origin and the orientation of each robot points to the origin. Each robot can communicate with other robots and sense objects within range $d_c = 0.55$ and is desired to reach another robot's initial position s.t. the total displacement spans over $2\pi/3$ counter-clockwise w.r.t. the origin. See Figure 6 for illustration. Similar scenarios have been investigated in Dimarogonas et al. (2006), Olfati-Saber et al. (2007). Five different scenarios are considered, where the only controlled variable is the number of robots $N \in \{5, 10, 15, 20, 25\}$.

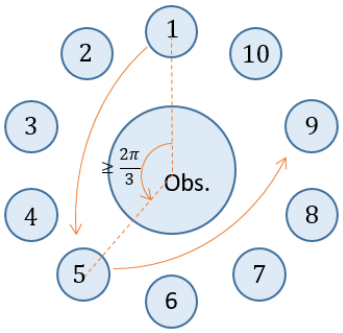


Fig. 6. Each robot desires to reach another robot's initial position as indicated by arrows.

6.1 Implementation of Algorithm 1

Algorithm 1 requires attractive controller $\mathcal{A}_i$ to solve a single-robot optimal motion planning problem. Essentially, this is a constrained optimal control problem of a nonlinear system, and no analytical solution can be derived. Hence, numerical algorithms are necessary; e.g., the algorithm in Section 3.2.4 of Cardaliaguet et al. (1999), where a continuous-time dynamic system in a continuous state space is approximated by a set-valued dynamic system over a discrete grid. Since each attractive controller $\mathcal{A}_i$ is independent of others, we compute them parallelly. The maximum linear velocity and angular velocity generated by the attractive controller are restricted by 0.5 to satisfy the assumptions imposed in Cardaliaguet et al. (1999). A regular grid of $77 \times 77 \times 20$ nodes with uniform spatial resolution 0.3 is adopted for each robot. Throughout the computations, the discrete grid and state transitions are shared among the robots.

Collision avoidance in Cardaliaguet et al. (1999) is theoretically guaranteed if allowable computation times are infinite or sufficiently large. In contrast, allowable computation times in practical scenarios are always finite, dynamic and uncertain. To address this practical issue, we slightly revise the control strategy when a robot contacts a static obstacle.

The repulsive controller $\mathcal{R}_i^O : \mathcal{C}_i \to U_i$ treats the static obstacle $\ell^* \in \mathcal{V}^O$ as a stationary robot and follows the idea of $\mathcal{R}_i^1$ in (3) to compute the evasive linear speed when robot i senses the static obstacle ℓ^* nearby; i.e., $\|r_{\ell^*} - r_i\| \leq d_{no}$ for some $d_{no} > d_{so}$. This procedure is summarized below:

$$v_i = \begin{cases} \max\{v_i^A, v_{eo}\}, & \text{if } (e_i^x)^T e_{\ell^* i} > 0; \\ \min\{v_i^A, -v_{eo}\}, & \text{if } (e_i^x)^T e_{\ell^* i} < 0; \\ v_i^A, & \text{if } (e_i^x)^T e_{\ell^* i} = 0, \end{cases}$$
$$\omega_i = \omega_i^A,$$

where $v_{eo} > 0$ is the minimum evasive speed from the obstacle so as to avoid oscillation and $(v_i^A, \omega_i^A) = \mathcal{A}_i(q_i)$ is the control of attractive controller. Compared to the original attractive controller $\mathcal{A}_i$, the revised control strategy differs in the sensing range d_{no} and the minimum evasive speed v_{eo}.

6.2 Two competitors

To verify the performance of Algorithm 1, two competitors are constructed.

Centralized coupled algorithm (CC): This type of algorithms; e.g., Algorithm 1 in Zhao & Zhu (2020), treats the entire robot team as an entity and computes a central controller to command all the robots. In the simulations, CC is only tested for $N = 2$ robots and adopts a coupled regular grid of $538,752$ nodes with spatial resolution 1.5 since CC is heavy in computations.

Centralized decoupled algorithm (CD): CD constructs an attractive controller for each robot which ignores all other robots. Different from Algorithm 1, CD sequentially computes decoupled controllers with a centralized

computer and directly implements the computed controllers without coordination among the robots. In the simulations, CD's settings follow those in Section 6.1.

6.3 Scalability

Figure 7 exhibits the computation times for five scenarios of Algorithm 1 and CD. The computation times of CD are linear w.r.t. the number of robots while those of Algorithm 1 are independent of the number of robots. Notice that the computation time for CC exceeds the computation time limit 6,000 seconds and is not shown in Figure 7. Further, Figure 8 is the zoomed-in view of Algorithm 1 in Figure 7. It shows that, for each scenario, the maximum difference between the computation times for different robots is only 0.2% of the computation time per robot.

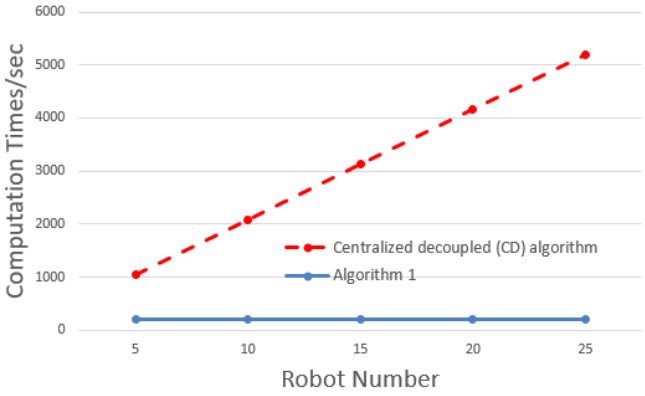


Fig. 7. The computation times of CD and Algorithm 1 at five scenarios.

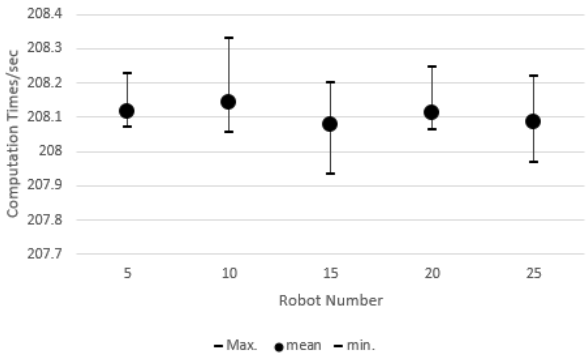


Fig. 8. The maximum, mean and minimum computation times of each robot in Algorithm 1 at five scenarios.

6.4 Near-optimality

To verify the optimality of Algorithm 1, the ideal benchmark is the CC algorithm, whose optimality and collision avoidance are simultaneously guaranteed. However, this algorithm is not scalable as it takes over 6,000 seconds to compute for just two robots, and therefore cannot be used for a large number of robots. Instead, CD is chosen as the benchmark. Notice that the total traveling times of CD are smaller than those of CC.

Throughout the simulations, Assumption 5.1 is never violated and Algorithm 1 can successfully drive all robots to their respective goal regions without causing collisions. The travelling times are summarized in Figure 9, where the total traveling times of Algoritm 1 are higher than those of CD, but the differences are small; both traveling times are almost linear w.r.t. the number of robots.

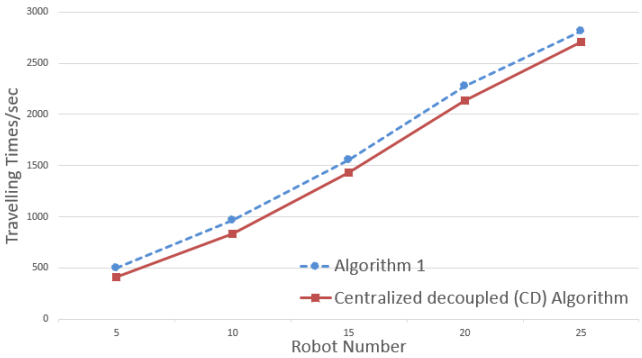


Fig. 9. The total traveling times of Algorithm 1 and the centralized decoupled algorithm (CD).

In particular, the trajectories of $N = 25$ robots driven by Algorithm 1 are shown in Figure 10. In this simulation, evasions from the static obstacle and moving robots are activated and 10 out of 25 robots apply more than one mode of Algorithm 1. The corresponding minimum distances between any two robots and any pair of a robot and the obstacle over time are shown in Figure 11. It is clear that the minimum distances are always beyond the corresponding safety distances and thus the collision avoidance is accomplished.

7 Conclusion

In this paper, a distributed multi-robot coordination algorithm is proposed. The algorithm formally guarantees collision avoidance and goal region arrival. A set of simulations are conducted to assess the scalability and near-optimality.

References

Atınç, G.M., Stipanović, D.M., & Voulgaris, P.G. (2020). A swarm-based approach to dynamic coverage control of multi-agent systems. *Automatica*, 112, 108637.

Bemporad, A. & Morari, M. (1999). Robust model predictive control: A survey. In *Robustness in Identification and Control*, 207–226. Springer.

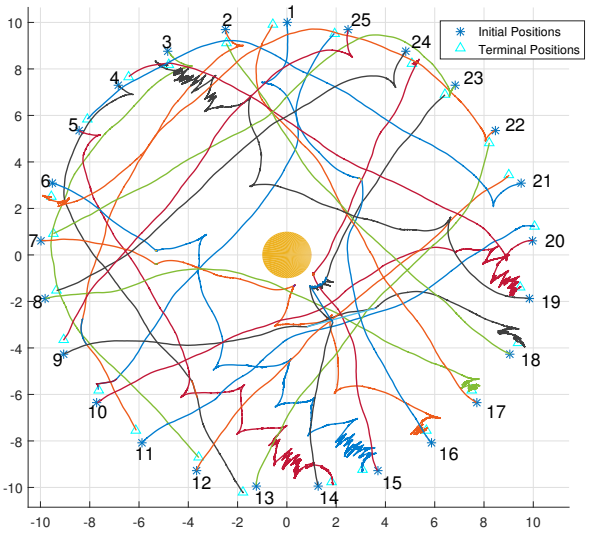


Fig. 10. Trajectories of 25 robots under the guidance of Algorithm 1.

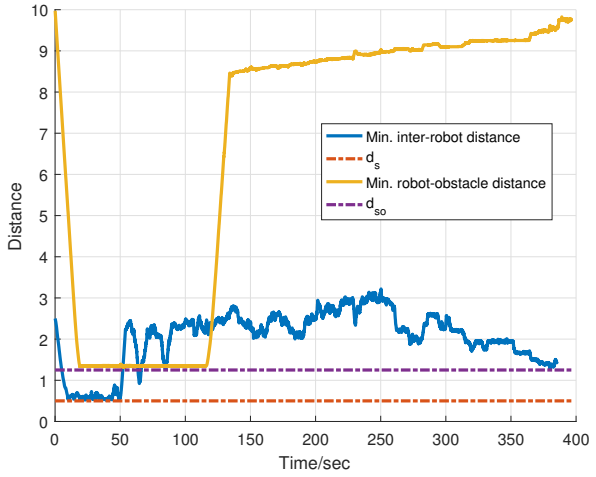


Fig. 11. Minimum distances between two robots and any robot-obstacle pair.

Brooks, R.A. & Lozano-Perez, T. (1985). A subdivision algorithm in configuration space for findpath with rotation. *IEEE Transactions on Systems, Man, and Cybernetics*, (2), 224–233.

Buckley, S. (1989). Fast motion planning for multiple moving robots. In *Proceedings of 1989 International Conference on Robotics and Automation*, 322–326.

Bullo, F., Cortés, J., & Martínez, S. (2009). *Distributed control of robotic networks: A mathematical approach to motion coordination algorithms*. Princeton University Press.

Canny, J. (1988). *The complexity of robot motion planning*. MIT Press.

Canny, J. & Reif, J. (1987). New lower bound techniques for robot motion planning problems. 49–60.

Cao, M., Morse, A.S., & Anderson, B.D. (2008). Reaching a consensus in a dynamically changing environment: A graphical approach. *SIAM Journal on Control and Optimization*, 47(2), 575–600.

Cao, Y.U., Fukunaga, A.S., & Kahng, A. (1997). Cooperative mobile robotics: Antecedents and directions. *Autonomous Robots*, 4(1), 7–27.

Cardaliaguet, P., Quincampoix, M., & Saint-Pierre, P. (1999). Set-valued numerical analysis for optimal control and differential games. In *Stochastic and Differ. Games*, 177–247. Springer.

Cortés, J., Martínez, S., Karatas, T., & Bullo, F. (2004). Coverage control for mobile sensing networks. *IEEE Transactions on Robotics and Automation*, 20(2), 243–255.

D'Andrea, R. (2012). Guest editorial: A revolution in the warehouse: A retrospective on kiva systems and the grand challenges ahead. *IEEE Transactions on Automation Science and Engineering*, 9(4), 638–639.

Dimarogonas, D.V., Loizou, S.G., Kyriakopoulos, K.J., & Zavlanos, M.M. (2006). A feedback stabilization and collision avoidance scheme for multiple independent non-point agents. *Automatica*, 42(2), 229–243.

Dunbar, W.B. & Caveney, D.S. (2011). Distributed receding horizon control of vehicle platoons: Stability and string stability. *IEEE Transactions on Automatic Control*, 57(3), 620–633.

Dunbar, W.B. & Murray, R.M. (2006). Distributed receding horizon control for multi-vehicle formation stabilization. *Automatica*, 42(4), 549–558.

Egerstedt, M. & Hu, X. (2002). A hybrid control approach to action coordination for mobile robots. *Automatica*, 38(1), 125–130.

Erdmann, M. & Lozano-Perez, T. (1987). On multiple moving objects. *Algorithmica*, 2(1-4), 477–521.

Frazzoli, E. & Bullo, F. (2004). Decentralized algorithms for vehicle routing in a stochastic time-varying environment. In *43rd IEEE Conference on Decision and Control*, volume 4, 3357–3363.

Fu, J., Ma, R., & Chai, T. (2015). Global finite-time stabilization of a class of switched nonlinear systems with the powers of positive odd rational numbers. *Automatica*, 54, 360–373.

Grune, L. & Rantzer, A. (2008). On the infinite horizon performance of receding horizon controllers. *IEEE Transactions Automatic Control*, 53(9), 2100–2111.

Jha, D.K., Zhu, M., & Ray, A. (2015). Game theoretic controller synthesis for multi-robot motion planning-part II: Policy-based algorithms. *The 5th IFAC Workshop on Distributed Estimation and Control in Networked Systems*, 48(22), 168–173.

Kant, K. & Zucker, S.W. (1986). Toward efficient trajectory planning: The path-velocity decomposition. *International Journal of Robotics Research*, 5(3), 72–89.

Karaman, S. & Frazzoli, E. (2011). Sampling-based algorithms for optimal motion planning. *International Journal of Robotics Research*, 30(7), 846–894.

Karaman, S. & Frazzoli, E. (2013). Sampling-based optimal motion planning for non-holonomic dynamical systems. In *2013 IEEE International Conference on Robotics and Automation*, 5041–5047.

Keviczky, T., Borrelli, F., Fregene, K., Godbole, D., & Balas, G.J. (2007). Decentralized receding horizon control and coordination of autonomous vehicle formations. *IEEE Transactions on Control Systems Technology*, 16(1), 19–33.

Khatib, O. (1986). Real-time obstacle avoidance for manipulators and mobile robots. In *Autonomous robot vehicles*, 396–404. Springer.

Koren, Y. & Borenstein, J. (1991). Potential field methods and their inherent limitations for mobile robot navigation. In *IEEE Conference on Robotics and Automation*, volume 2, 1398–1404.

Kuwata, Y. & How, J.P. (2010). Cooperative distributed robust trajectory optimization using receding horizon milp. *IEEE Transactions on Control Systems Technology*, 19(2), 423–431.

Kuwata, Y., Richards, A., Schouwenaars, T., & How, J.P. (2007). Distributed robust receding horizon control for multivehicle guidance. *IEEE Transactions Control Systems Technologies*, 15(4), 627–641.

LaValle, S.M. & Kuffner, J.J. (2001). Randomized kinodynamic planning. *International Journal of Robotics Research*, 20(5), 378–400.

Li, H., Shi, Y., & Yan, W. (2016). Distributed receding horizon control of constrained nonlinear vehicle formations with guaranteed γ-gain stability. *Automatica*, 68, 148–154.

Ma, Y., Kosecka, J., & Sastry, S.S. (1999). Vision guided navigation for a nonholonomic mobile robot. *IEEE Transactions on Robotics and Automation*, 15(3), 521–536.

Mayne, D.Q. (2014). Model predictive control: Recent developments and future promise. *Automatica*, 50(12), 2967–2986.

Mayne, D.Q., Rawlings, J.B., Rao, C.V., & Scokaert, P.O. (2000). Constrained model predictive control: Stability and optimality. *Automatica*, 36(6), 789–814.

Milnor, J.W. & Weaver, D.W. (1997). *Topology from the differentiable viewpoint*. Princeton University Press.

Olfati-Saber, R., Fax, J.A., & Murray, R.M. (2007). Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(1), 215–233.

Pallottino, L., Scordio, V.G., Bicchi, A., & Frazzoli, E. (2007). Decentralized cooperative policy for conflict resolution in multivehicle systems. *IEEE Transactions on Robotics*, 23(6), 1170–1183.

Panagou, D. & Kumar, V. (2014). Cooperative visibility maintenance for leader–follower formations in obstacle environments. *IEEE Transactions on Robotics*, 30(4), 831–844.

Rantzer, A. & Johansson, M. (2000). Piecewise linear quadratic optimal control. *IEEE transactions on automatic control*, 45(4), 629–637.

Reif, J.H. (1979). Complexity of the mover's problem and generalizations. 421–427.

Ren, W. & Beard, R.W. (2008). *Distributed consensus in multi-vehicle cooperative control*. Springer.

Schwager, M., Rus, D., & Slotine, J.J. (2009). Decentralized, adaptive coverage control for networked robots. *International Journal of Robotics Research*, 28(3), 357–375.

Schwartz, J.T. & Sharir, M. (1983). On the "piano movers" problem. II. general techniques for computing topological properties of real algebraic manifolds. *Advances in applied Mathematics*, 4(3), 298–351.

Siméon, T., Leroy, S., & Lauumond, J.P. (2002). Path coordination for multiple mobile robots: A resolution-complete algorithm. *IEEE Transactions on Robotics and Automation*, 18(1), 42–49.

Tomlin, C., Pappas, G.J., & Sastry, S. (1998). Conflict resolution for air traffic management: A study in multiagent hybrid systems. *IEEE Transactions on Automatic Control*, 43(4), 509–521.

Xu, X. & Antsaklis, P.J. (2004). Optimal control of switched systems based on parameterization of the switching instants. *IEEE Transactions on Automatic Control*, 49(1), 2–16.

Yan, Z., Jouandeau, N., & Cherif, A.A. (2013). A survey and analysis of multi-robot coordination. *International Journal of Advanced Robotic Systems*, 10(12), 399.

Zhang, J., Johansson, K.H., Lygeros, J., & Sastry, S. (2001). Zeno hybrid systems. *International Journal of Robust and Nonlinear Control*, 11(5), 435–451.

Zhang, L. & Shi, P. (2009). Stability, l_2-gain and asynchronous H_∞ control of discrete-time switched systems with average dwell time. *IEEE Transactions on Automatic Control*, 54(9), 2192–2199.

Zhao, G. & Zhu, M. (2019). Scalable distributed algorithms for multi-robot near-optimal motion planning. In *2019 IEEE Conference on Decision and Control*, 226–231.

Zhao, G. & Zhu, M. (2020). Pareto optimal multi-robot motion planning. *IEEE Transactions on Automatic Control*, To appear.

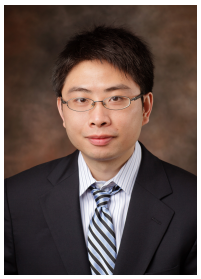
Zhu, M. & Martínez, S. (2013). On distributed constrained formation control in operator–vehicle adversarial networks. *Automatica*, 49(12), 3571–3582.

Zhu, M. & Martínez, S. (2015). *Distributed Optimization-Based Control of Multi-Agent Networks in Complex Environments*. Springer.

Zhu, M., Otte, M., Chaudhari, P., & Frazzoli, E. (2014). Game theoretic controller synthesis for multi-robot motion planning-part I: Trajectory based algorithms. In *2014 IEEE International Conference on Robotics and Automation*, 1646–1651.

Guoxiang Zhao is a PhD student in the School of Electrical Engineering and Computer Science at the Pennsylvania State University. He received the B.E. degree in mechanical engineering from Shanghai Jiao

Tong University, Shanghai, China, in 2014 and the M.S. degree in mechanical engineering from Purdue University, West Lafayette, IN, USA, in 2015. His research interests mainly focus on multi-robot optimal motion planning.



Minghui Zhu is an Associate Professor in the School of Electrical Engineering and Computer Science at the Pennsylvania State University. Prior to joining Penn State in 2013, he was a postdoctoral associate in the Laboratory for Information and Decision Systems at the Massachusetts Institute of Technology. He received Ph.D. in Engineering Science (Mechanical Engineering) from the University of California, San Diego in 2011. His research interests lie in distributed control and decision-making of multi-agent networks with applications in robotic networks, security and the smart grid. He is the co-author of the book "Distributed optimization-based control of multi-agent networks in complex environments" (Springer, 2015). He is a recipient of the Dorothy Quiggle Career Development Professorship in Engineering at Penn State in 2013, the award of Outstanding Reviewer of Automatica in 2013 and 2014, and the National Science Foundation CAREER award in 2019. He is an associate editor of the Conference Editorial Board of the IEEE Control Systems Society.