2019 IEEE Symposium Series on Computational Intelligence (SSCI)
December 6-9 2019, Xiamen, China

Distributed Collision-Avoidance Formation Control: A Velocity Obstacle-Based Approach

Yifan Hu
School of Mathematics
Southeast University
Jiangsu Province, Nanjing 210000
Email: yfhu@seu.edu.cn

Han Yu
School of Cyber Science
and Engineering
Southeast University, China
Email: 220184429@seu.edu.cn

Yang Zhong
School of Cyber Science
and Engineering
Southeast University, China
Email: 220184500@seu.edu.cn

Yuezu Lv
School of Mathematics
Southeast University, China
Email: yzlv@pku.edu.cn

***Abstract*—This paper proposes a discrete-time algorithm for formation control of multi-agent systems with collision avoidance, where the agents' velocity and steering constraints are also considered. To ensure the collision avoidance, the velocity obstacle and reciprocal velocity obstacle methods are introduced to modify the distributed formation algorithm, where each agent uses velocity obstacle method to avoid collision with obstacles, and reciprocal velocity obstacle method to avoid collision with other agents. In this sense, each agent has the ability to pass through complex obstacle environments by autonomously changing prefer velocity. Simulation results show that the proposed algorithm can achieve formation task and meanwhile guarantee collision avoidance.**

***Keywords*—Formation control; collision avoidance; velocity and steering constraints**

I. Introduction

In recent years, the consensus of multiple first-order or second-order integrators has been well developed [1][2], since it can be easily implemented in various cooperative control problems of multi-agent systems including formation control [3], flocking [4], economic dispatch in smart grids [5] and so on.

This paper intends to introduce the consensus algorithm to solve the formation control problem of multi-agent systems, which has broad civilian or military applications for unmanned vehicle systems such as UAVs and USVs [6]-[8]. For the formation control of multi-agent systems, the key task is to design protocols or algorithms to make the agents reach a specific formation [9]. In [10], the formation control protocol is proposed for each agent by using local state information (position, velocity) of neighboring agents. Observer-based formation is considered in [11], where the specified formation is reached by a decentralized control method. In [12], adaptive leader-follower formation control protocol is proposed to make agents achieve the specific formation task and move along the specified reference trajectory. Actuator faults and system constraints are considered in [13] and a finite-time leader-follower formation control strategy is proposed using time-varying tan-type barrier Lyapunov functions. In [14], communication costs are considered and an event-triggered formation protocol is proposed to realize the specific formation.

The major issue of the applications of the aforementioned formation algorithms in practice is that the collisions may occur, which should be absolutely forbidden during the whole formation process. In [4] and [9], the potential functions are employed in the formation algorithms, where the collision avoidance is theoretically ensured by using a Lyapunov function based approach. However, these potential function based algorithm may require biggish velocities or acceleration of the agents when they are close to obstacles. Such requirement may not be satisfied due to limited physical mobility in real applications. A collision avoidance method based on safety barrier certificates is proposed for second-order multi-agent systems in [15]. The basic idea is to choose an optimal acceleration, which is closest to the preferred acceleration in the optional acceleration set. The optional acceleration set may be empty, and the collision in such case may not be averted. To overcome such limitation, the velocity obstacle method is proposed in [16], and the reciprocal velocity obstacle method is further presented in [17]. Both these two methods make agents avoid collisions if each agent chooses velocity out of its velocity obstacle set. For the case the optional velocity set is empty, the agent can still choose suboptimal velocity.

In view of such advantage of velocity obstacle and reciprocal velocity obstacle methods, in this paper we aim at combining the distributed formation control algorithm with these two methods to formulate a novel distributed collision avoidance formation control algorithm. To achieve collision avoidance in complex obstacle environments, each agent uses velocity obstacle method to avoid collision with obstacles, and reciprocal velocity obstacle method with other agents. Moreover, the velocity and steering constraints are also considered, and thus the proposed algorithm is easier to be implemented in practice. The main contribution of this paper is that an efficient algorithm is designed, which can ensure both fast formation and collision avoidance. The effectiveness of the proposed algorithm is illustrated by simulation results.

The rest of this paper is organized as follows. In Section II, some preliminary results on graph theory, consensus and collision avoidance are summarized. In Section III, the problem of formation control with collision avoidance for agents modeled

This work was supported by the National Natural Science Foundation of China under Grant 61903083, the Natural Science Foundation of Jiangsu Province of China under Grant BK20190333, and the Fundamental Research Funds for the Central Universities No. 4007019104 and No. 2242019K40111.

as first-order integrators is stated. Section IV presents a novel distributed collision avoidance formation control algorithm. The proposed algorithm is illustrated by a specific formation task in Section V. Concluding remarks and discussions are given in Section VI.

Notations: $\|\cdot\|$ denotes the Euclidean norm of a vector. For two matrices A and B, $A \otimes B$ denotes the Kronecker product. Function $d(\cdot,\cdot) : R^n \times R^n \to R$ represents Euclidean distance between two n dimensional vectors. $\langle \cdot,\cdot \rangle$ denotes the angle of two vectors. And $\mu(\cdot) : R^n \to R^n$ is a function which satisfies that for $\forall x \in R^n$, $\mu(x) = x/\|x\|$ if $\|x\| \neq 0$, and $\mu(x) = [0, \cdots, 0]' \in R^n$ otherwise.

II. Preliminary

A. Graph Theory

Let $G = (\mathcal{V}, \mathcal{E})$ denote a weighted digraph of n nodes with node set $\mathcal{V} = \{v_1, \cdots, v_n\}$, edge set $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$. An edge of G is denoted by $e_{ij} = (v_i, v_j)$. $\mathcal{A} = [a_{ij}]$ is the weighted adjacency matrix of G with $a_{ij} \geq 0$, where $a_{ii} = 0$ for $\forall i \in \{1, ..., n\}$ and $a_{ij} > 0$ equals to $e_{ij} \in \mathcal{E}$. Neighbors of node i are in the set $N_i = \{v_j \in \mathcal{V} : (v_i, v_j) \in \mathcal{E}\}$. The Laplacian matrix $\mathcal{L} = [l_{ij}]$ of G is induced by $\mathcal{A}$, where $l_{ii} = \sum_{j=1}^{n} a_{ij}$ and $l_{ij} = -a_{ij}$ $(\forall i \neq j)$.

B. Consensus of First-Order Multi-Agent System

Typically, we use $x_i(t) \in R^N$ to represent the state of agent i at time t. Under the following protocol

$$\dot{x}_i(t) = a \sum_{j=1}^{n} a_{ij} \left(x_j(t) - x_i(t)\right), \tag{1}$$

with communication topology containing a spanning tree, states of agents will achieve consensus exponentially as time approaches infinity [2]. This result is obtained by the property of Laplacian matrix, which is shown in the following lemma:

Lemma 1 ([2]): Given a digraph G and $\mathcal{L}$ is the Laplacian matrix of G, then $\mathcal{L}$ has at least one zero eigenvalue and all of the nonzero eigenvalues are in the open right half plane. Furthermore, $\mathcal{L}$ has exactly one zero eigenvalue if and only if G has a spanning tree.

C. Collision Avoidance

In this subsection, we briefly review two collision avoidance methods: Velocity Obstacles and Reciprocal Velocity Obstacles. For convenience, both agents and obstacles are called agents and they are translated in the plane with the shape of discs. Agent i's reference point is positioned at the disc center, which is denoted by $p_i(t) \in R^2$ and the radius is denoted by $r_i \in R$.

1) Velocity Obstacle: Set agent i with reference point $p_i(t)$, radius r_i and agent j with $p_j(t)$, r_j in the plane. A velocity obstacle set $VO_j^i(v_j(t))$ is naturally generated by agent j (regarded as an obstacle) [16], which includes all those velocities $v_i(t)$ for agent i that will result in a collision at some moment in time if agent j keeps its velocity $v_j(t)$ unchanged. The effectiveness of the VO method for agents to avoid collision with static or simple dynamic obstacles is guaranteed by assuming that obstacles will keep their velocities [17].

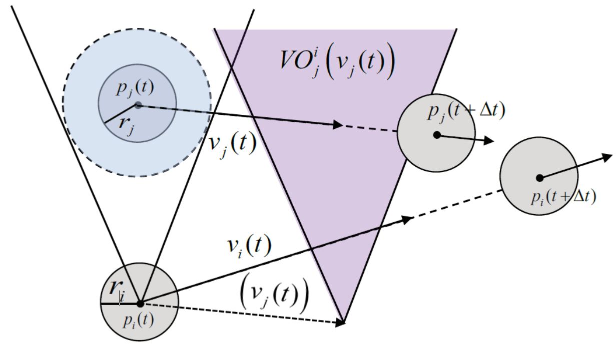

Fig. 1. Purple area represents velocity obstacle $VO_j^i(v_j(t))$ of agent j to agent i and agent i can choose velocity out of $VO_j^i(v_j(t))$ to avoid collision with agent j.

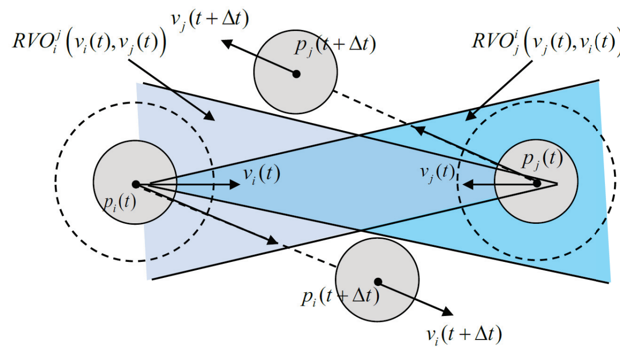

Fig. 2. Blue area represents $RVO_j^i(v_j(t), v_i(t))$ and purple area represents $RVO_i^j(v_i(t), v_j(t))$. Both agent i and j choose velocities out of the reciprocal velocity obstacle sets, and at time $t + \Delta t$ and they avoid a collision successfully.

2) Reciprocal Velocity Obstacle: Set agent i and agent j in the plane and the notations are the same as in Velocity Obstacle. The reciprocal velocity obstacle of agent j to agent i is defined as follows:

$RVO_j^i(v_j(t), v_i(t)) = \left\{ v'_i(t) | 2v'_i(t) - v_i(t) \in VO_j^i(v_j(t)) \right\}.$

The reciprocal velocity obstacle set of agent j to agent i contains all velocities for agent i that are the average of current velocity $v_i(t)$ and a velocity inside the velocity obstacle set $VO_j^i(v_j(t))$ of agent j. For a pair of agents, if they both choose velocities out of their corresponding reciprocal velocity obstacle set, safety can be guaranteed (see Fig. 2) [17].

Remark 1: The difference between this method and Velocity Obstacle is that each agent doesn't need to make all effort in adjusting velocity to avoid collision. In other words, the collision avoidance for each agent depends on both its velocity adjustment and other agents' adjustment. Thus, it has better performance than VO method when applied in cooperative collision avoidance of multiple agents.

III. Problem Statement

Consider a multi-agent system consisting of n agents, the set of agents is denoted by $\bar{N}_1 = \{1, \cdots, n\}$ and the notations of $p_i(t)$ and r_i are the position and radius of agent i. Dynamics of agents are modeled as first-order integrator, i.e.,

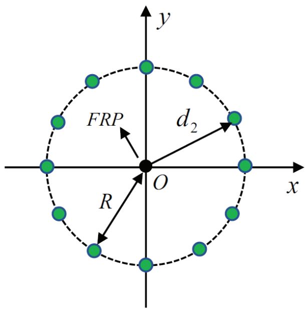

Fig. 3. A circle formation of 12 agents with radius R and FRP at origin in Cartesian coordinate. The formation positions d_i $(i = 1, \cdots, 12)$ are located around the circle evenly.

$\dot{p}_i(t) = v_i(t)$, where $v_i(t)$ denotes velocity of agent *i* at time *t*. Meanwhile, there exists a velocity limit $v_{\max} \in R_+$, i,e., $\|v_i(t)\| \leq v_{\max}$. Static obstacles are also set in the plane with the number of *m*. Here, the set of obstacles is denoted by $\bar{N}_2 = \{n+1, \cdots, n+m\}$, and $p_j(t) \equiv p_j(0)$ for $\forall j \in \bar{N}_2$.

For the sake of convenience, the following definitions are given.

Definition 1: Formation reference point (FRP) is a specified position of a formation.

Definition 2: Formation position is the set of relative positions between agents' final formation positions and FRP. Agent *i*'s formation position is denoted by $d_{\sigma(i)} \in R^2$.

Remark 2: In fact, FRP can be arbitrarily specified and different FRP determines unique formation positions for the same formation configuration. For some formation with shape of centrosymmetry, FRP can be specified by the position of symcenter for geometric significance, e.g., specify FRP of a circle formation by the circle center. And $\sigma : N_1 \to N_1$ is a bijection which matches each agent to a specified formation position.

Definition 3: Estimated formation reference position of agent *i* at time *t* is denoted by $x_i(t) \equiv p_i(t) - d_{\sigma(i)}$.

Remark 3: A specific formation is achieved if and only if estimated formation reference positions of all agents reach agreement. That is, the formation control problem is equivalent to the consensus problem of $x_i(t)$.

Collision avoidance also needs to be considered. The condition to guarantee collision avoidance can be written as follows:

$$d\left(p_i(t), p_j(t)\right) \geq r_i + r_j \tag{2}$$

for $\forall i \in \bar{N}_1$ and $\forall j \in \bar{N}_1 \cup \bar{N}_2$ during the formation process.

The objective of this paper is to design an algorithm to make agents realize a specific formation with the FRP located at a predetermined position (denoted by $x_{end} \in R^2$) while satisfying velocity limit and collision avoidance, i.e., design $v_i(t)$ to drive $p_i(t)$ converge to $x_{end} + d_{\sigma(i)}$ with the constraints $\|v_i(t)\| \leq v_{\max}$ and inequality (2) $\forall i = 1, \cdots, n$, $j = 1, \cdots, n+m$.

IV. Algorithm Design of Formation Control with Collision Avoidance

In this section, we will first solve a relevant formation control problem, then we will design the prefer velocity and finally provide a discrete-time algorithm of formation control satisfying collision avoidance and dynamic constraints.

A. *Protocol of Formation Control without Collision Avoidance*

Inspired by consensus protocol (1), we design a control protocol

$$v_i(t) = a\sum\nolimits_{j=1}^{n} a_{ij}\left(x_j(t) - x_i(t)\right) + b\left(x_{end} - x_i(t)\right), \tag{3}$$

where a, b are positive constants. Protocol (3) make each agent converge to $x_{end} + d_{\sigma(i)}$ by using estimated formation reference points of its neighbors, where the velocity limit and collision avoidance are not considered.

Theorem 1: For given x_{end} and $d_{\sigma(i)}$, $p_i(t)$ will globally converge to $x_{end} + d_{\sigma(i)}$ exponentially under protocol (3) if the communication topology of the multi-agent system has a spanning tree.

Proof 1: Since $p_i(t) \equiv x_i(t) + d_{\sigma(i)}$, then $\dot{p}_i(t) \equiv \dot{x}_i(t)$. And equation (3) can be rewritten as: $\dot{x}_i(t) = a\sum\nolimits_{j=1}^{n} a_{ij} \left(x_j(t) - x_i(t)\right) + b\left(x_{end} - x_i(t)\right)$. Let $y_i(t) \equiv x_i(t) - x_{end}$, the multi-agent system can be rewritten as:

$$\dot{Y}(t) = -\left[(aL + bI_n) \otimes I_2\right] Y(t), \tag{4}$$

where $Y(t) = [y'_1(t), \cdots, y'_n(t)]'$ and $Y(t) \in R^{2n}$. By Lemma 1, all eigenvalues of the dynamic matrix in (4) are negative, which means $y_i(t)$ will globally converge to 0 exponentially, i.e., $p_i(t)$ will globally converge to $x_i(t) + d_{\sigma(i)}$ exponentially for all $i = 1, \cdots, n$.

B. *Design of Prefer Velocity*

In practice, velocity always suffers from saturation constraint due to physical limit. And the velocity designed in (3) may be beyond the saturation constraint as a, b or consensus error is large. Such drawback motivates us to design prefer velocity for agents, which is defined as:

Definition 4: Prefer velocity is a realizable velocity which guides agents' directions to achieve specified formation with suitable velocity magnitude, denoted by $v_{pref,i}(t)$.

To meet the demand of input saturation, we should make modifications on protocol (3), which consists of two terms, the consensus part (CP) $\sum\nolimits_{j=1}^{n} a_{ij}\left(x_j(t) - x_i(t)\right)$ and target movement part (TMP) $x_{end} - x_i(t)$. The control parameters a and b are used to adjust the weights of each term. Specifically, larger a can result in rapider formation achievement; while larger b can make agents arrive in target positions faster. Denote $\gamma_i(t)$ and $1 - \gamma_i(t)$ as the normalized control parameters with $\gamma_i(t) \in (0, 1)$. The following modifications are made.

$$cp_i(t) = \mu\left(\sum\nolimits_{j=1}^{n} a_{ij}\left(x_j(t) - x_i(t)\right)\right), \tag{5}$$

$$tmp_i(t) = \mu\left(x_{end} - x_i(t)\right), \tag{6}$$

where cp_i and tmp_i denote the consensus part and the target movement part respectively, $\mu(\cdot)$ modifies nonzero vectors into unit vectors. In this sense, CP and TMP are modified as unit velocity vectors or zero vectors. Combining with $\gamma_i(t)$ and $1-\gamma_i(t)$, we design prefer velocity as follows:

$$v_{pref,i}(t) = u_i(t)\mu\left(\gamma_i(t)cp_i(t) + (1-\gamma_i(t))\,tmp_i(t)\right), \quad (7)$$

with $u_i(t)$ satisfying

$$u_i(t) = \begin{cases} u, & \|c_i(t)\| \le e_f \\ u+\Delta u, & otherwise \end{cases},$$

and $\gamma_i(t)$ satisfying

$$\gamma_i(t) = \begin{cases} \gamma_1, & \|c_i(t)\| \le e_f \\ \gamma_2, & otherwise \end{cases},$$

where $c_i(t) = \sum_{j=1}^{n} a_{ij}\left(x_j(t) - x_i(t)\right)$. $\|c_i(t)\|$ and e_f represent the formation error and formation tolerant error respectively. $u+\Delta u$ denotes the velocity magnitude when $\|c_i(t)\| \le e_f$ and u denotes the velocity magnitude when the formation error is larger. u and Δu are two positive constants which satisfy $|u+\Delta u| \le v_{\max}$. And $\gamma_1, \gamma_2 \in [0,1]$ denote two proportions with $\gamma_1 < \gamma_2$.

The design of prefer velocity consists of two parts, the velocity magnitude u_i, and the velocity direction determined by μ. The initial value of the consensus error $\|c_i(t)\|$ is probably large, which makes $\gamma_i(t) = \gamma_2$ and $u_i(t) = u+\Delta u$. Large $\gamma_i(t)$ leads the prefer velocity direction to achieve a formation and high velocity magnitude makes agents achieve the formation faster. In this case, agents may not be able to reach the target, since velocity direction of achieving formation may offset the direction of approaching the target. After the consensus error of some agent i is relatively small ($\|c_i(t)\| \le e_f$), it would choose a small $\gamma_i(t)$ to reach the target, while other agents are still trying to form a formation. $u+\Delta u$ guarantees agents can catch up with the position $\left(\sum_{j=1}^{n} a_{ij}x_j(t) \middle/ \sum_{j=1}^{n} a_{ij}\right) + d_{\sigma(i)}$ to form a formation. When all agents' consensus errors are within the permitted range, the formation is realized, and directions of agents' prefer velocities mainly point at target positions, as $\gamma_i(t)$ is small.

Remark 4: The existence of e_f is to prevent agents frequently changing velocity directions when agents are very close to their positions in the formation because these directions are too sensitive to agents' positions. But the consensus tolerate error also brings inaccuracy to the formation.

C. Algorithm of Formation Control with Collision Avoidance

In this subsection, we further consider the collision avoidance, where the obstacles for each agent include both real static obstacles and other agents. As previously mentioned, VO and RVO methods generate obstacle sets of velocity and velocities in these sets may lead to a collision. Agents avoid collisions by choosing velocities out of these sets. We use VO method and RVO method to avoid collisions with static obstacles and agents respectively. Let $O_i(t)$ denote union set of all velocity obstacle sets and reciprocal velocity obstacle sets of agent i at time t, which is represented as follows:

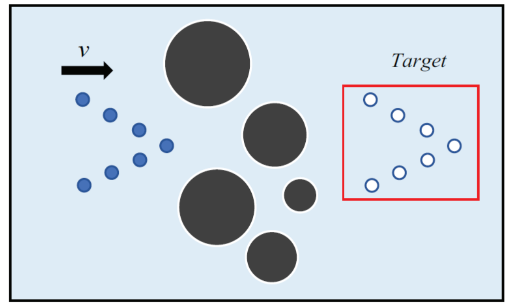

Fig. 4. Grey circles denote static obstacles which stop agents moving towards target. Blue circles denote agents which are in an arrow formation and moving towards target positions in red box with velocity v. Agents are facing a moving difficulty problem for static obstacles are too dense.

$$O_i(t) = \left(\bigcup_{j\in\bar{N}_2} VO_j^i\left(v_i(t)\right)\right) \bigcup \left(\bigcup_{k\in\bar{N}_1} RVO_k^i\left(v_k(t), v_i(t)\right)\right).$$

Let $G_i(t)$ denote velocity candidate set of agent i at time t, which represents the reachable velocity for agent i. $G_i(t)$ is a set containing all velocities v satisfying $\langle v_i(t), v\rangle \in [0,\phi]$ and $\|v\| \in [0, u_i(t)]$ with $\phi > 0$ representing agents' steering capability. Then agent i can choose its new velocity v which satisfies $v \in G_i(t)$ and $v \notin O_i(t)$. Since $v_{pref,i}(t)$ may belong to $O_i(t)$, the new velocity $v_i(t)$ can be modified by the following formula:

$$v_i(t) = \underset{v\in G_i(t)\backslash O_i(t)}{\arg\min} \|v - v_{pref,i}(t)\| \quad (8)$$

However, $G_i(t)\backslash O_i(t)$ could be an empty set in crowded environments. This means whatever velocity the agent chooses, collision will happen at some point in time. In the following, a compromise toward collision would be introduced to overcome such limitation. Let $tc_i^j(v,t_0)$ be the collision time between agent i and agent j when agent i chooses v as its velocity. $tc_i^j(v,t_0)$ can be calculated by solving $\lambda\left(p_i(t_0), v - v_j(t_0)\right) = (i \oplus j)(t_0)$ for t at time t_0 in VO method. And $tc_i^j(v,t_0)$ is solved similarly by solving $\lambda(p_i(t_0),\ 2v - v_i(t_0) - v_j(t_0)) = (i\oplus j)(t_0)$ for t at time t_0 in RVO method. The expected time to collision $t_i^{col}(v,t)$ of agent i is defined as follows:

$$t_i^{col}(v,t) = \min\ tc_i^j(v,t), \quad (9)$$

where $i \in \bar{N}_1$, $j \in \bar{N}_1 \cup \bar{N}_2$ and $j \ne i$. And formula (8) is modified as:

$$v_i(t) = \underset{v\in G_i(t)}{\arg\min}\left(\beta\frac{1}{t_i^{col}(v,t)} + \|v - v_{pref,i}(t)\|\right), \quad (10)$$

where $\beta > 0$. When $G_i(t)\backslash O_i(t) \ne \emptyset$, $t_i^{col}(v,t) = \infty$ and v_i is chosen the same as (8). When $G_i(t)\backslash O_i(t) = \emptyset$, a velocity which is considered safe within a period time and close to $v_{pref,i}(t)$ will be chosen.

For the case that agents cannot pass through some static obstacles and meanwhile keep formation (see Fig. 4), novel prefer velocity design method should be developed.

First we introduce the following definition of minimum distance:

Definition 5: Minimum distance is the minimum distance between an agent and static obstacles, denoted by $l_i(t)$ for agent i, which can be written as:

$$l_i(t) = \min_{j\in\bar{N}_2} \left(d\left(p_i(t), p_j(t)\right) - r_i - r_j\right). \tag{11}$$

$l_i(t) < 0$ means there is a collision between an agent and obstacles and $l_i(t) > 0$ means safety. Moving difficulty may happen when $l_i(t)$ is positive but close to 0.

We divide agents' moving environment into two types: crowded and uncrowded, which are denoted by 1 and 0. $signal_i(t)$ is used to record the environment type for agent i at time t:

$$signal_i(t) = \begin{cases} 1, & l_i(t) \geq l_o \\ 0, & otherwise \end{cases},$$

where l_o is a positive constant. When $l_i(t) < l_o$, agents should be vigilant and change their moving strategy. So l_o can be regarded as a vigilance distance.

There is no doubt that when an agent is in a crowded environment, its priority is to avoid collision with obstacles. Based on this fact, two situations are divided according to agents' moving environments and their neighbors' environments:

(A1) $signal_i(t) = 1$ and $signal_j(t) = 1$ for all $j_0 \in N_i$;

(A2) $signal_i(t) = 0$ or exists $j_0 \in N_i$ and $signal_{j_0}(t) = 0$, where N_i denotes number set of agent i's neighbors. The following assumption is necessary to prevent agents from switching between (A1) and (A2) repeatedly.

Assumption 1: The dwell time for the agent in situation (A2) should be larger than some positive constant T_{A2}, named as the minimal residence time.

For situation (A1), the prefer velocity of the agent i can be chosen as in (7); while for situation (A2), the prefer velocity can be designed as

$$v_{pref,i}(t) = u \cdot tmp_i(t). \tag{12}$$

Up till now, we can conclude the following Algorithm 1 to achieve formation with collision avoidance for multi-agent systems.

Algorithm 1 Formation Control with Collision Avoidance

Initialization:

1) Agent i's initial position $p_i(0)$, velocity $v_i(0)$ and radius r_i for $\forall i \in \bar{N}_1$. Agent j's initial position $p_j(0)$, velocity $v_j(0) = 0$ and radius r_j for $\forall j \in \bar{N}_2$;

2) FRP target position x_{end}, formation position $d_{\sigma(i)}$ for $\forall i \in \bar{N}_1$ and an adjacency matrix $\mathcal{A}$ of communication topology containing a spanning tree.

3) Agents' initial situation (A1), residence time $t_i^{res} = 0$ for $\forall i \in \bar{N}_1$, stop distance ε, time interval Δt and running time of the algorithm $t = 0$.

Iteration:

while $\exists i_0 \in \bar{N}_1, \|x_{i_0}(t) - x_{end}\| \geq \varepsilon$, **do**
  for all $i = 1, \cdots, n$, **do**
    Calculate agent i's actual situation at time t;
    Update agent i's current situation with actual situation if last situation was (A1) or (A2) with residence time satisfying $t_i^{res} \geq T_{A2}$. Otherwise, keep last situation of agent i;
    Calculate agent i's prefer velocity $v_{pref,i}(t)$ according to its situation based on formula (7) or (12);
    Choose $v_i(t)$ based on formula (10) as its velocity at time t;
  end for
  Update positions by $p_i(t + \Delta t) = p_i(t) + v_i(t) \cdot \Delta t$ for all $i = 1, \cdots, n$, let $t = t + \Delta t$ and $t_i^{res} = t_i^{res} + \Delta t$. For agents which change situations, let $t_i^{res} = 0$.
end while

TABLE I
EXPERIMENTAL PARAMETERS

r_i	u	Δu	γ	γ_2	e_f
2m	10m/s	4m/s	1m	0.1	0.7
l_o	β	T_{A2}	ε	Δt	ϕ
20m	10	10s	1m	0.1s	$\pi/32$

V. Simulation

In this section, simulation examples are given to illustrate the effectiveness of the proposed algorithm. Consider 30 agents to realize circle formation with radius $R = 120m$ whose initial positions are shown in Fig. 5 at $t = 0$. And the dense static obstacles are also set in Fig. 5.

Let $d_{\alpha(i)}^T = R\left[\cos\frac{2\pi i}{n}, \sin\frac{2\pi i}{n}\right]$, $v_i(0)^T = [0, 0]$ and $a_{ij} = 1$ if $d\left(d_{\sigma(i)}, d_{\sigma(j)}\right) \leq 80m$ for all $i = 1, \cdots, n$. Other experimental parameters are shown in Table 1.

The positions of each agent in different time are shown in Fig. 5 meaning that the agents can achieve fast formation and pass through obstacles with collision avoidance by breaking up the circle formation, and then reformulate formation.

$fe(t)$ denotes Formation Error at time t with expression that $fe(t) = \frac{1}{n}\sum_{i=1}^{n}\left(x_i(t) - \frac{1}{n}\sum_{j=1}^{n} x_j(t)\right)^2$, which is given in Fig. 6, demonstrating that the formation is indeed achieved. FRP Distance $d_{end} = d\left(x_{end}, \frac{1}{n}\sum_{i=1}^{n} x_i(t)\right)$, the distance between agents' estimated FRPs and x_{end}, is also presented in Fig. 7 to show that each agent can reach its predetermined final position. Global Nearest Distance $gnd(t)$, is designed based on condition (2) as

$$gnd(t) = \min_{i\in\bar{N}_1} \min_{j\in\bar{N}_1\cup\bar{N}_2, j\neq i} \left(d\left(p_i(t), p_j(t)\right) - r_i - r_j\right).$$

It can be seen from Fig. 8 that collision avoidance is guaranteed since $gnd(t) > 0$ during the simulation process.

VI. Conclusion and Discussion

In this paper, we proposed a discrete-time algorithm of formation control with collision avoidance. The advantages of the proposed algorithm are summarized as follows. First, formation of any shape can be achieved theoretically. Second, velocity and steering constraints are considered in this algorithm

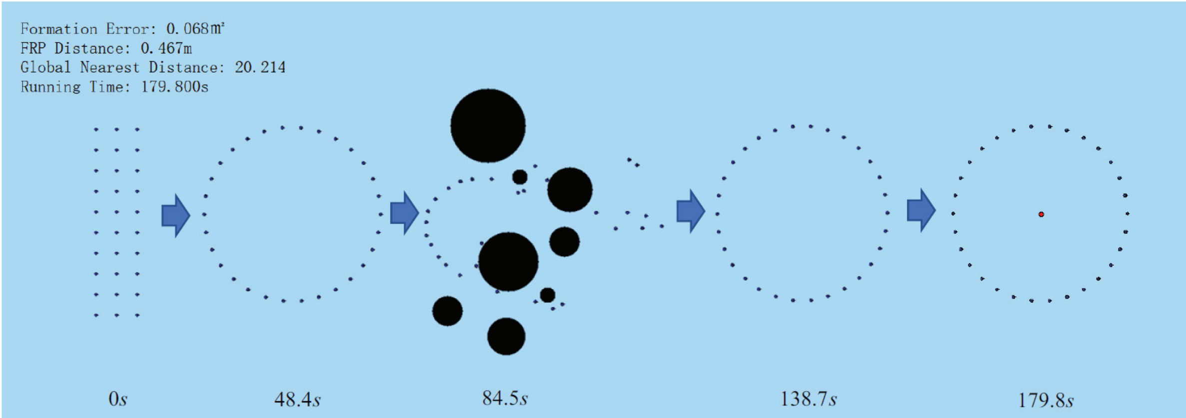

Fig. 5. 30 agents are positioned in a matrix shape at 0s. Circle formation task is given to the multi-agent system with FRP located at the red point. The system interface records agents' position change in 5 moments during the simulation process. Before facing static obstacles, agents have already achieved a circle formation and the formation is destructed when agents try to pass through obstacles. After all agents passed through obstacles, they achieve the circle formation again and finally arrive at the target formation positions.

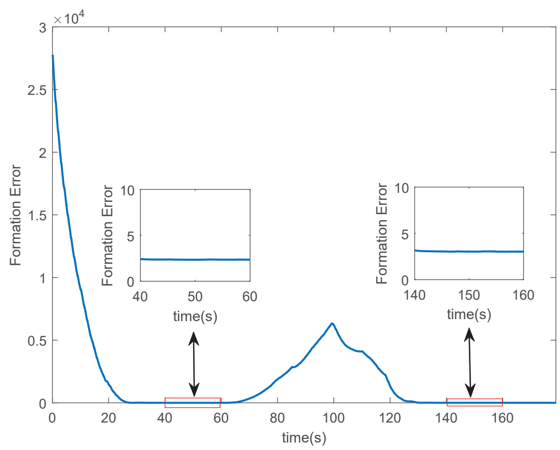

Fig. 6. Formation Error is large at first, then it decreases fast and stays near 2.5 from 30s to 60s. Formation Error increases when agents faces static obstacles. At about 100s, Formation Error decreases again and stays near 2.5 until end of the simulation.

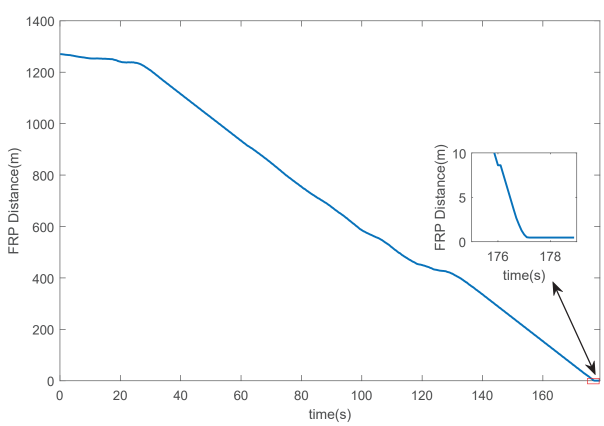

Fig. 7. FRP Distance keeps decreasing during the simulation process and it gets extremely close to 0 in the end. The index decreases faster when agents have achieved a formation.

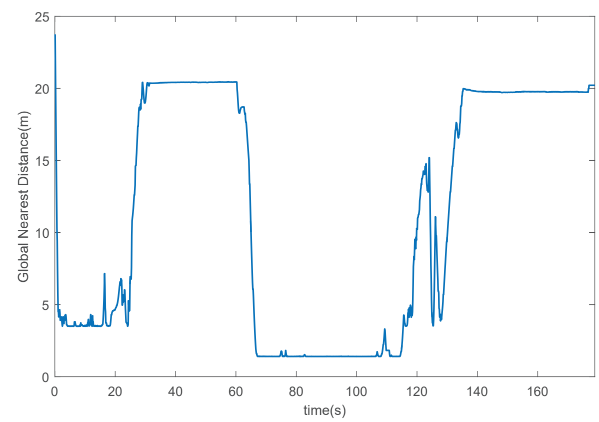

Fig. 8. Global Nearest Distance keeps greater than zero and it means there is no collision during the simulation process.

which makes it more practical. Third, this algorithm provides a simple framework as prefer velocity can be modified for any specified cooperative control task.

Future work can be done along the following two aspects. First, intermittent communication needs to be considered, where prefer velocity needs to be modified and collision avoidance still needs to be satisfied. Second, deadlock phenomenon [15] also happens during some simulation process when number of agents is large and an alternative idea is to reduce obstacle set of velocities to guarantee more safe velocities for agents to choose.

REFERENCES

[1] Olfati-Saber, R., and Murray, R. M. (2004). *"Consensus problems in networks of agents with switching topology and time-delays"*. IEEE Transactions on automatic control, 49(9), 1520-1533.

[2] Ren, W., and Atkins, E. (2007). *"Distributed multi-vehicle coordinated control via local information exchange"*. International Journal of Robust and Nonlinear Control: IFAC-Affiliated Journal, 17(10-11), 1002-1033.

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

[4] Olfati-Saber, R. (2004). *"Flocking for multi-agent dynamic systems: Algorithms and theory"* (No. CIT-CDS-2004-005). CALIFORNIA INST OF TECH PASADENA CONTROL AND DYNAMICAL SYSTEMS.

[5] Wen, G., Yu, X., Liu, Z. W., and Yu, W. (2017). *"Adaptive consensus-based robust strategy for economic dispatch of smart grids subject to communication uncertainties"*. IEEE Transactions on Industrial Informatics, 14(6), 2484-2496.

[6] Dong, X., Zhou, Y., Ren, Z., and Zhong, Y. (2016). *"Time-varying formation tracking for second-order multi-agent systems subjected to switching topologies with application to quadrotor formation flying"*. IEEE Transactions on Industrial Electronics, 64(6), 5014-5024.

[7] Wang, Y. L., and Han, Q. L. (2018). *"Network-based modelling and dynamic output feedback control for unmanned marine vehicles in network environments"*. Automatica, 91, 43-53.

[8] Li, Z., Li, C., Zhuo, S., and Chen, S. (2017, October). *"UAVs cooperative attack based on archimedes spiral"*. In 2017 17th International Conference on Control, Automation and Systems (ICCAS) (pp. 1094-1098). IEEE.

[9] Jia, Q., and Li, G. (2007, August). *"Formation control and obstacle avoidance algorithm of multiple autonomous underwater vehicles (AUVs) based on potential function and behavior rules"*. In 2007 IEEE international conference on automation and logistics (pp. 569-573). IEEE.

[10] Lin, Z., Wang, L., Han, Z., and Fu, M. (2014). *"Distributed formation control of multi-agent systems using complex Laplacian"*. IEEE Transactions on Automatic Control, 59(7), 1765-1777.

[11] Oh, K. K., and Ahn, H. S. (2012). *"Formation control of mobile agents based on distributed position estimation"*. IEEE Transactions on Automatic Control, 58(3), 737-742.

[12] Peng, Z., Yang, S., Wen, G., Rahmani, A., and Yu, Y. (2016). *"Adaptive distributed formation control for multiple nonholonomic wheeled mobile robots"*. Neurocomputing, 173, 1485-1494.

[13] Jin, X. (2016). *"Fault tolerant finite-time leaderCfollower formation control for autonomous surface vessels with LOS range and angle constraints"*. Automatica, 68, 228-236.

[14] Ge, X., and Han, Q. L. (2017). *"Distributed formation control of networked multi-agent systems using a dynamic event-triggered communication mechanism."*. IEEE Transactions on Industrial Electronics, 64(10), 8118-8127.

[15] Wang, L., Ames, A. D., and Egerstedt, M. (2017). *"Safety barrier certificates for collisions-free multirobot systems"*. IEEE Transactions on Robotics, 33(3), 661-674.

[16] Fiorini, P., and Shiller, Z. (1998). *"Motion planning in dynamic environments using velocity obstacles"*. The International Journal of Robotics Research, 17(7), 760-772.

[17] Van den Berg, J., Lin, M., and Manocha, D. (2008, May). *"Reciprocal velocity obstacles for real-time multi-agent navigation"*. In 2008 IEEE International Conference on Robotics and Automation (pp. 1928-1935). IEEE.