2019 American Control Conference (ACC)
Philadelphia, PA, USA, July 10-12, 2019

Moving Obstacle Avoidance and Topology Recovery for Multi-agent Systems

Han Wang, Yushan Li, Wenbin Yu, Jianping He, and Xinping Guan

***Abstract*—This paper proposes a novel moving obstacle avoidance algorithm for multi-agent systems. The method has robustness in maintaining formation shape. Even if link failure occurs among agents when avoiding obstacles, the communication topology of the system can be recovered based on the conditions we obtain. The main idea includes two parts, i) a flexible function of relative velocities and positions between agents and obstacles is designed to avoid moving/stationary obstacles, and ii) based on initial adjacent matrix and graph connection characteristic, a topology recover mechanism is proposed to guarantee formation shape and no extra links are involved. The proposed algorithm has the following advantages: i) It is able to recover formation shape, even if some links among agents are broken while avoiding moving obstacles; ii) In the process of rebuilding communication links, the proposed algorithm can protect communication topology from being altered maliciously. Furthermore, we obtain conditions of the topology recovery for both directed and undirected graph. Some simulations are conducted to demonstrate the efficiency of the proposed algorithm.**

I. Introduction

Formation control is one of the most actively studied topics within the realm of multi-agent systems, aiming to drive agents to achieve prescribed shape. Consensus-based method is one of the most common way [1]. There are many applications related to formation control such as unmanned vehicles, wheeled robots, satellites, etc [2] [3]. It is remarkable that, formation control and path planning, the two research topics have a large overlap, where obstacle avoidance is a major part [4]. In real situations, an agent of the formation must have the ability to avoid collision with other agents and obstacles. It has received more and more attentions to design superior collision/obstacle avoidance algorithm.

Design an efficient obstacle avoidance function is the key challenge for the avoidance algorithm. Initially, artificial potential field (APF) is raised in [5]. The key idea is that the target has attraction force and obstacles have repulsion force to the agents, respectively, producing a potential field. Based on this approach, [6] presents a methodology for exact robot motion planning and control that unifies the purely kinematic path planning problem with the lower level feedback controller design. The author in [7] further raises virtual force field, which combines certainty grids for obstacle representation with artificial potential fields for navigation. This method is usually used when there are numeric obstacles. However, these researches just consider static obstacles. The methods are desired to apply to moving obstacles.

The authors are with the Department of Automation, Shanghai Jiao Tong University, and Key Laboratory of System Control and Signal Processing, Ministry of Education of China, Shanghai 200240, China. E-mail: l: JTUniversity_WH@sjtu.edu.cn, yushan_li@sjtu.edu.cn, yuwenbin@sjtu.edu.cn, jphe@sjtu.edu.cn, xpguan@sjtu.edu.cn. This research work is sponsored by the National Key R&D Program of China 2017YFE0114600, National Natural Science Foundation of China (61803261), and Shanghai Natural Science Foundation of China (18ZR1421100).

There are two main challenges for moving ones: i) sampling and predicting [8] the velocities of obstacles; ii) designing collision-free function. Some literatures investigate sampling problem. Authors in [9] use sampling and stochastic prediction to predict obstacle's motion. Others focus on algorithm design. In [10], a randomized motion planner for agents is design to achieve collision-free, where the obstacle is assumed to be known. In [11], the author represents a concept named velocity obstacle to achieve obstacle avoidance by selecting the velocity from feasible velocity set. [12] further investigates the problem for non-linear velocity obstacle.

Recent works mainly combine formation control and obstacle avoidance. Formation control and obstacle avoidance for multi agent systems with non-holonomic constraints is studied in [13]. The authors in [14] and [15] concern about the dynamic model of multi robot systems. Authors in [16] propose a virtual and behavioral structure, in which the agents are modeled by electric charge. Based on this structure, a rotational potential field is applied to avoid obstacles. In addition, many researches take specific applications into account. For instance, [17] improves potential function and behavior rules to effectively control the formation of a multiple autonomous underwater vehicle system in uncertain environment. In [18], the author considers the shape of the agents and proposes an algorithm, which can be used to rectangular agents.

On the other hand, some works focus on possible topology switching while avoiding obstacles in multi-agent systems. In real systems, especially when the agents are avoiding obstacles, they may move out of their neighbors' communication range. Thus, it's significant to investigate the formation problem with switching topology while avoiding obstacles. [19] further proposes a Lyapunov function to deal with average consensus problem of the flock system with obstacle avoidance. [20] [21] further investigate consensus problem with directed switching topology. Authors in [22] combine obstacle avoidance and formation control for switching topology, but they just consider to change Laplacian matrix in their formation, with no realistic conditions given to maintain consensus while avoiding obstacles. [23] takes account of communication range in a second order system, provides a sufficient condition on the initial velocities, positions and the communication range which guarantees consensus of the system. [24] further investigates the system with probable package loss.

978-1-5386-7926-5/$31.00 ©2019 AACC

However, exiting works basically assume that connection can be established with any nearby nodes, which is not desired to maintain a prescribed formation shape and may even give rise to vulnerabilities of the system. Motivated by this, we design an obstacle avoidance algorithm, where the links between agents can is recovered only when they exist initially. In this architecture, links among agents can be protected from being altered maliciously. We combine formation protocol and moving obstacle avoidance algorithm with possible switching topology, and derive a condition, under which the topology of the system can be safely recovered when it is broken.

Main contributions of this paper are summarized as follows:

- A novel obstacle avoidance algorithm is proposed, which is based on a flexible function of relative velocity and relative position between agents and obstacles. This algorithm has strong robustness in maintaining formation of the multi-agent system.
- For the proposed protocol, a topology recovery mechanism is designed, such that communication graph is recovered without being altered while formation is restored.
- Based on multi-agent systems, relations between number of broken edges and topology recovery for both directed graph and undirected graph are demonstrated.
- Extensive simulations are conducted to prove the effectiveness for algorithm and correctness of conditions.

The rest of this paper is organized as follows: In Section II concepts from graph theory are reviewed, formation control of the system and sampling of obstacles are presented. In Section III design of collision avoidance algorithm are shown. Conditions of topology recovery are obtained in Section IV. Simulation results are shown in Section V. Conclusions are given in Section VI.

II. PRELIMINARIES

To avoid collision between agents and obstacles for multi-agent systems, an avoidance function is usually added to formation control protocol. Moreover, for moving obstacles, sampling and prediction of their motions are necessary. In this section, firstly, some preliminary concepts from graph theory are reviewed; secondly, formation control algorithm is shown; thirdly, sampling of obstacle is discussed.

A. Graph Theory

Let $\mathcal{G}$ =($\mathcal{V}$,$\mathcal{E}$) be an directed graph to model the information flow among agents,where $\mathcal{V}$ = {1,...,N} is the set of nodes and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ denotes the set of edges. Every node in the graph represents an agent and every directed edge represents an information flow channel. An edge $(i, j) \in \mathcal{E}$ indicates that the agent j transmits some information to agent i. The adjacency matrix $A = [a_{ij}] \in \mathbb{R}^{N\times N}$ of a graph $\mathcal{G}$ =($\mathcal{V}$, $\mathcal{E}$) with N nodes specifies the interconnection topology of the multi-agent system. In the graph, a_{ij} = 1 if $(i, j) \in \mathcal{E}$, else a_{ij} = 0. Let $\mathcal{N}_i$ = {i ∈ $\mathcal{V}$: $a_{ij} \neq 0$} denote the neighbors of node i. Define Laplacian matrix L of the graph $\mathcal{G}$ as $L = D-A$, where D = diag($A \cdot \mathbf{1}$). Here $\mathbf{1}$ = $[1\ 1\ 1\ ...\ 1]^{\mathrm{T}} \in \mathbb{R}^N$ denotes the vector of ones, which is also one of the right eigenvector of L corresponding to eigenvalue, λ_1 = 0, and $L \cdot \mathbf{1} = 0$

B. Formation Control

Consider a group of mobile agents with kinematic models given by

$$\begin{bmatrix} \dot{x}_i(k) \\ \dot{y}_i(k) \end{bmatrix} = \begin{bmatrix} v_{ix}(k) \\ v_{iy}(k) \end{bmatrix}$$

where $i = 1, ..., N$, $p_i(k) = [x_i(k)\ y_i(k)]^{\mathrm{T}}$ describes the position of agent i in the plane $\mathbb{R}^2$, $v_i(k) = [v_{ix}(k)\ v_{iy}(k)]^{\mathrm{T}}$ denotes the velocity of agent i at time t, respectively. To achieve desired formation shape, the control law is designed by

$$p_i(k+1) = p_i(k) + u_i^{form}(k), \tag{1}$$

$$u_i^{form}(k) = \varepsilon \sum_{j=1}^{N} a_{ij}(p_j(k) - p_i(k) - r_{ij}), \tag{2}$$

where r_{ij} represents the predefined formation distance between agent i and agent j.

If the graph of the system has a spanning tree, consensus can be reached in the end, and the stability and robustness of the system have been proved in [2].

C. Sampling of Obstacle

To avoid collision between moving obstacles with agents, possible trajectory of obstacles needs to be known. In real situations, agents are able to detect the obstacles and sample their positions to predict velocities, then their trajectories can be estimated. In this problem, sense-time is defined as the time instant at which the agent senses the obstacle position and velocity. Control-period is defined as the time interval during which the robot performs sensing, predicting and planning for collision motion. It is assumed that control-period can be divided by sense-time, which means that agents can get real velocities of obstacles when they need to avoid obstacles. Thus, the whole process consists of sampling and avoidance. Thus, sampling process is presented as follows.

Consider holonomic kinematic model of moving obstacle. Let $p_o(k) = [x_o(k)\ y_o(k)]^{\mathrm{T}}$ be the position of obstacle o in the plane $\mathbb{R}^2$, and $v_o(k) = [v_{ox}(k)\ v_{oy}(k)]^{\mathrm{T}}$ denote the velocity of obstacle o at time t respectively.

Every agent has a sensor which can scan the environment with the sampling period t_s, and bounded sampling range R, which is a bounded constant. If one obstacle exits in the range, agents can predict its velocity by two sampling period. We use $P_o(t)$ and $V_o(t)$ to represent the position and velocity of the obstacle when it is detected. We have

$$V_o(t_2) = \frac{P_o(t_2) - P_o(t_1)}{t_2 - t_1}, \tag{3}$$

where t_2 and t_1 represent two sampling moments, and $t_2 - t_1$ = t_s. After $P_o(t_2)$ and $P_o(t_1)$ are sampled, we calculate the velocity of the obstacle by (3) at real time.

III. Main Results

A. Problem Analysis

We define $\vec{V}_i(k) = [\hat{v}_{ix}(k) \quad \hat{v}_{iy}(k)]$, and $\vec{P}_i(k) = [\hat{p}_{ix}(k) \ \hat{p}_{iy}(k)]$ as the relative velocity and relative position between agent i and obstacle within the sensor detecting range. They are calculated by

$$\vec{V}_i(k) = \begin{bmatrix}\dot{x}_i(k) - \dot{x}_o(k) & \dot{y}_i(k) - \dot{y}_o(k)\end{bmatrix}^T,$$

$$\vec{P}_i(k) = \begin{bmatrix}x_o(k) - x_i(k) & y_o(k) - y_i(k)\end{bmatrix}^T.$$

Some possible positions of agents and obstacles are shown in Fig. 1. R and r are the radius of detection and protection areas around each agent, respectively. The arrow, which denotes relative speed is shown as $\vec{V}_i(k)$, arrow between agent and obstacle denotes relative location, which is shown as $\vec{P}_i(k)$. In this condition, obstacles can be regarded as static. Fig. 1 shows some possible positions of obstacles and agents.

To avoid obstacles, an avoidance function is usually used. In this work, the function is designed as

$$u_i^{obs}(k) = \eta \frac{L_i}{|\ell_o(k) - \ell_i(k)|}, \tag{4}$$

where ℓ is used to represent x and y. $L_i = [L_{ix} \ L_{iy}]^T$ is a function of $\vec{V}_i(k)$ and $\vec{P}_i(k)$, $u_i^{obs}(k) = [u_{ix}^{obs}(k) \ u_{iy}^{obs}(k)]^T$, η is a positive const. In this design, the fractional value will increase if agents and obstacles stay closer. And the function L_i, is designed to change the velocities of agents to avoid obstacles as well as minimally change the velocity directions to maintain formation.

Definition 1. *Let d_m be the distance between the obstacle and the expected trajectory of one agent, which is decided by relative velocity between agents and obstacles. Then, d_m is calculated by*

$$d_m = \sqrt{\frac{(Kx_i(k) - y_i(k) + y_o(k) - Kx_o(k))^2}{K^2 + 1}}, \tag{5}$$

where K is the slope of $\vec{\mathrm{V}}_i(k)$, satisfying

$$K = \tan\beta = \frac{\hat{v}_{iy}(k)}{\hat{v}_{ix}(k)}.$$

Note that d_m represents probable collision situation between agents and obstacles. If d_m increases with time and becomes bigger than the detection radius of agents, obstacle avoidance is achieved.

B. Algorithm Design

To achieve obstacle avoidance, firstly we give a lemma as follows, which is a basic part in the proof of avoidance function achievement.

Lemma 1. *If the obstacle comes into detection range of agent i, and $d_m < r$, then, $\hat{p}_{ix}(k)\hat{v}_{ix}(k)$ and $\hat{p}_{iy}(k)\hat{v}_{iy}(k)$ are not less than 0 at the same time, i.e.*

$$\hat{p}_{ix}(k)\hat{v}_{ix}(k) + \hat{p}_{iy}(k)\hat{v}_{iy}(k) > 0, \ \forall \ k. \tag{6}$$

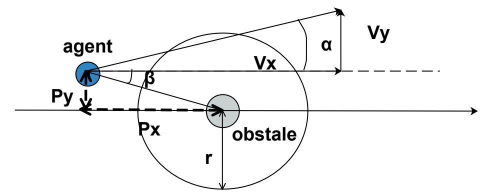


Fig. 1: One possible situation

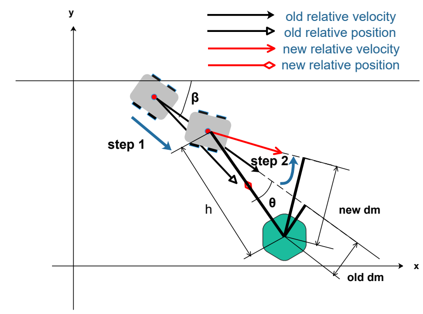


Fig. 2: Process of obstacle avoidance

Proof. The sign of $\hat{p}_{i\ell}(k)\hat{v}_{i\ell}(k)$ means whether agent i and obstacle stay closer or further on ℓ axis. $\hat{p}_{i\ell}(k)\hat{v}_{i\ell}(k) > 0$ means they stay closer, or they get further. If collision occurs, the sign of $\hat{p}_{ix}(k)\hat{v}_{ix}(k)$ and $\hat{p}_{iy}(k)\hat{v}_{iy}(k)$ are not both negative at the same time. And for that the x-acceleration speed depends on y-coordinate, $\hat{v}_{ix}(k)$ and $\hat{p}_{ix}(k)$ are not zero for any short period of time. This condition is same as that of y. Thus, the problem is divided into three possible situations.

- If $\hat{p}_{ix}(k)\hat{v}_{ix}(k) \geq 0$, $\hat{p}_{iy}(k)\hat{v}_{iy}(k) \geq 0$, for that $\hat{v}_{ix}(k)$ and $\hat{v}_{iy}(k)$ are not zero all the time, then $\hat{p}_{ix}(k)\hat{v}_{ix}(k) + \hat{p}_{iy}(k)\hat{v}_{iy}(k) \neq 0$. Thus, in this situation, (12) holds.
- If $\hat{p}_{iy}(k)\hat{v}_{iy}(k) < 0$, $\hat{p}_{ix}(k)\hat{v}_{ix}(k) > 0$. Consider at one time, one obstacle come into detection range of agent i, so $|\hat{p}_{iy}(k)| < r$, where r denotes protection range. Fig. 2 shows this condition, and one observes that

$$\tan\alpha = \frac{|\hat{v}_{iy}(k)|}{|\hat{v}_{ix}(k)|}, \tan\beta = \frac{|\hat{p}_{iy}(k)|}{|\hat{p}_{ix}(k)|},$$

  and we have $\alpha + \beta < \frac{\pi}{2}$, thus

$$\tan\alpha\tan\beta = \frac{|\hat{v}_{iy}(k)||\hat{p}_{iy}(k)|}{|\hat{v}_{ix}(k)||\hat{p}_{ix}(k)|} < 1,$$

  thus, (6) holds.
- If $\hat{p}_{ix}(k)\hat{v}_{ix}(k) < 0$ and $\hat{p}_{iy}(k)\hat{v}_{iy}(k) > 0$, exchanging x and y, the proof is similar as above.

These three situations include all possible situations, thus, we have proved the lemma. □

Theorem 1. *If the parameters L_{ix} and L_{iy} in (4) satisfy the follow conditions, then d_m will increase when agent i approaches obstacle.*

$$\begin{bmatrix}L_{ix} \\ L_{iy}\end{bmatrix} = \begin{bmatrix}sgn(\vec{P}_i(k) \otimes \vec{V}_i(k))H_i(k) \\ -sgn(\vec{P}_i(k) \otimes \vec{V}_i(k))J_i(k)\end{bmatrix}, \tag{7}$$

where

$$\begin{bmatrix} H_i(k) \\ J_i(k) \end{bmatrix} = \begin{bmatrix} sgn(\hat{p}_{ix}(k)\hat{v}_{ix}(k)\hat{v}_{iy}(k)) \\ sgn(\hat{p}_{iy}(k)\hat{v}_{iy}(k)\hat{v}_{ix}(k)) \end{bmatrix}, \tag{8}$$

and $\otimes$ is cross product. If the Fractional denominator of L_{ix} and L_{iy} equal to 0, we set the value of them to 1 and -1, respectively.

If the agents detect the obstacles, the direction of their movement is determined by L_{ix} and L_{iy}. We can prove that the algorithm can successfully realize obstacle avoidance by Lemma 1. and some more analysis. Proof of the theorem is omitted here due to the space limit.

There will be some constraint conditions in real situations. Firstly, the obstacle is detected just when it is within the detecting range, d_R is defined as the distance between obstacle and agents, satisfying

$$d_R = \sqrt{(x_i(k) - x_o(k))^2 + (y_i(k) - y_o(k))^2}, \tag{9}$$

where O_R is defined as the detection range. Hence, the obstacle is detected if $O_R = \{x|x \leq R\}$, $d_R \in O_R$, Secondly, obstacle avoidance algorithm is executed only when d_m fulfills $O_r = \{x|x \leq r\}, \, and \; d_m \in O_r$.

The expression of d_m is given by (5), r is protection range of agents. This condition maintains that in the process of avoiding obstacles, the velocity directions of the agents are minimally modified. Thirdly, it follows from (9) that if the obstacle and agent have a equal x or y coordinate, the denominator of the fractional is zero, so if the denominator is smaller than one threshold value, q, it will be set to q. Thus,

$$|\ell_i(k) - \ell_o(k)| = q, \; if \; |\ell_i(k) - \ell_o(k)| < q,$$

Use λ_x and λ_y to substitute $|x_i(k)-x_o(k)|$ and $|y_i(k)-y_o(k)|$, then the controller in (9) is changed as

$$u_i^{obs} = \begin{cases} \eta \dfrac{L_{i\ell}}{\lambda_\ell} & if \; d_R \in O_R \;\&\; d_m \in O_r \;\&\; \lambda_\ell < q; \\ \eta \dfrac{L_{i\ell}}{q} & if \; d_R \in O_R \;\&\; d_m \in O_r \;\&\; \lambda_\ell \geq q; \\ 0 & \text{otherwise.} \end{cases}$$

Eventually, the controller of the system combines the original formation controller in (2) with obstacle avoidance controller in (4) as

$$p_i(k+1) = p_i(k) + u_i^{form}(k) + u_i^{obs}(k). \tag{10}$$

Define agents velocity matrix *v* = $[v_1(k) \; ... \; v_n(k)]^T$, agents position matrix *p* = $[p_1(k) \; ... \; p_n(k)]^T$, obstacles velocity v_o = $[v_{o1}(k) \; ... \; v_{om}(k)]^T$.

Algorithm 1: Obstacle Avoidance

Input: The initial adjacent matrix *A*, the real-time adjacent matrix *A′*, velocity matrix *v*, position matrix *p*, obstacle velocity v_o.
Output: obstacle avoidance controller u_i^{obs}
for $i = 1; i \leq n;$ **do**
  if $d_R \leq$ *detection range* **then**
    if $|\ell_i(k) - \ell_o(k)| > q\&\&d_m \leq r$ **then**
      $u_{i\ell}^{obs} = \eta \frac{L_{i\ell}}{|\ell_i(k)-\ell_o(k)|}$
    if $|\ell_i(k) - \ell_o(k)| \leq q\&d_m \leq r$ **then**
      $u_{i\ell}^{obs} = \eta \frac{L_{i\ell}}{q}$
  else
    $u_{i\ell}^{obs} = 0$
  return $u_{i\ell}^{obs}$

IV. TOPOLOGY RECOVERY CONDITION

In real situations, communication constraints are inevitable, two kinds of communication barrier may exist. Firstly, considering the communication range of the agents, agents can't transmit information to each other unless the distance among them is not longer than the threshold value. If at some points a robot is far from his neighbors beyond the range, communication between them will be interrupted. Secondly, due to environmental interference, there may be packet loss during communication. The above two cases can be reflected as the edges in the graph of the system are broken. Conditions of topology recovery are discussed in this section.

A. Conditions Discussion

Assumption 1. *Assume that the directed graph of the system which has a connected with a root node.*

This is a basic assumption, which ensures that the system can achieve consensus eventually.

Definition 2. *$\mathcal{G}$ is defined as the graph, ψ is a closed curve which divides the nodes in the graph into two sets,* S *and* Y*, and* S $\cap$ Y $= \varnothing$. *$\mathcal{S}$ and $\mathcal{Y}$ are the parts of graph which are formed by* S *and* Y.

Theorem 2. *Consider a graph that satisfies Assumption 1. When $\forall$ n edges are broken, the graph is still connected in the condition that $\forall$ ψ, there exits at least $n{+}1$ edges which break through ψ, and the edges must start from the set contains the root node to the set doesn't. And for graph that is connected and undirected, the condition is also sufficient and necessary.*

Proof. Firstly, we prove sufficiency for undirected graph. In this case, we prove that for arbitrary ψ, if k edges break through ψ, $k \geq n+1$, then $\forall$ n edges are broken, the graph $\mathcal{G}$ is still connected.

There are at most n nodes at the end of the edges that are disconnected from their father nodes. Use ψ to surround all that nodes, and *S* to represent the set of theses nodes, *Y* to represent the else set of nodes. Since nodes in *Y* are all father nodes, thus, nodes in *Y* are connected. And initially the number of the edges which break through ψ is bigger than n, so $\mathcal{S}$ and $\mathcal{Y}$ are connected. As to $\mathcal{S}$, just repeat the process, divide it into $\mathcal{S}_1$, $\mathcal{S}_2$...$\mathcal{S}_m$, each of the graph is connected. And $\mathcal{S} \cup \mathcal{Y} \cup \mathcal{S}_1 \cup ... \cup S_m$ = $\mathcal{G}$, so that $\mathcal{G}$ is connected, the sufficiency for undirected graph is proved.

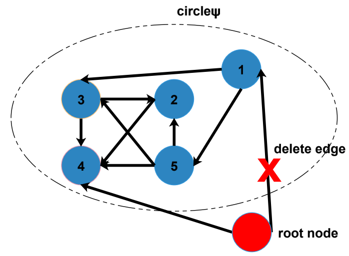
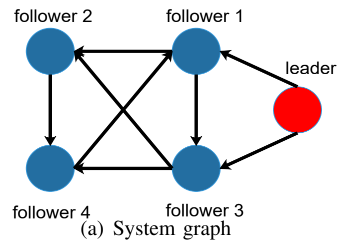


Fig. 3: Worst situation of topology break

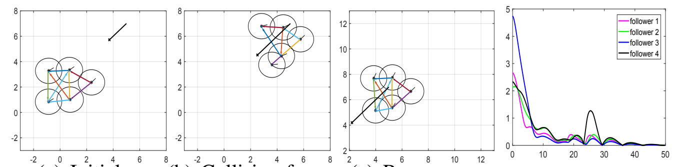


(a) System graph

$$A=\begin{bmatrix}0&0&0&1&1\\1&0&1&0&0\\1&0&0&0&1\\0&1&1&0&0\\0&0&0&0&0\end{bmatrix}$$

(b) Adjacent matrix

Fig. 4: A case of system topology recovery



(a) Initial (b) Collision-free (c) Recovery (d) Position error

Fig. 5: Linear motion

Secondly, we prove necessity for undirected graph. For this case, we need to prove that if $\forall\ n$ edges are broken and the graph is still connected, then $\forall\ \psi$, there are at least $n+1$ edges cross it.

If $\exists\ \psi$ with just k edges cross it, $k < n+1$. Then, we let the k edges which are broken all be the former k edges, the graph will be separated into two parts, since the two parts have no edges between them, the graph is disconnected.

Finally, we prove necessity for directed graph, i.e.,if $\forall\ n$ edges are broken and the graph is still connected, then $\forall\ \psi$, there are at least $n+1$ edges cross it and the edges must starts from the set which contains the root node.

Firstly, like the former proof, if the number of edges which cross ψ is less than $n+1$, we chose to break them all, thus, the graph is unconnected. Secondly, if there are $n+1$ edges cross ψ, but just n of them are point from the set which contains the root node, we break them all. In this situation, although $\mathcal{S}$, $\mathcal{Y}$, are both connected, and they connect with each other, but there is no way from the root node to every node in *S* and *Y*. The reason why the condition isn't sufficient for directed graph is shown in Fig. 4. In the figure, edge with $n = 1$ is deleted, and two edges cross ψ, the graph is connected former, but isn't connected later. □

From the discussion, we give a condition in which the system can reach consensus when n edges are broken in the graph. Furthermore, we assume that if one agent formerly communicates with another agent, and their communications are interrupted at time t_1. At time t_2, The distance between them becomes shorter than the communication range, then, the information channel is rebuilt. We use adjacent matrix A to represent the initial condition of the system, and adjacent matrix A' to represent the realtime condition of the system. Initially, $A' = A$, a_{ij} and a'_{ij} are elements of the matrix. Topology recovery protocol is given as

$$\begin{aligned} a'_{ij} &= 0,\ if\ d_{ij} > R\ \&\ a_{ij} = 1, \\ a'_{ij} &= 1,\ if\ d_{ij} \leq R\ \&\ a_{ij} = 1, \end{aligned} \tag{11}$$

where d_{ij} denotes the distance between node i and node j, R denotes the communication range.

V. SIMULATION RESULTS

In this section, simulations to verify the effectiveness of the proposed control laws are presented. Considering the system of five agents defined by adjacent matrix A, system topology graph and adjacent matrix are shown in Fig. 4.

The topology of the system is shown in Fig. 4(a). The initial positions of the agents are set as : $[x_1\ y_1\ th_1] = [1\ \ -2\ \ 0], [x_2\ y_2\ th_2] = [-2\ \ 1\ \ 0], [x_3\ y_3\ th_3] = [1\ 1\ \pi/4], [x_4\ y_4\ th_4] = [-2\ \ -1\ \ -\pi/4], [x_5\ y_5\ th_5] = [-1\ \ -1\ \ \pi/2]$, and the formation is defined as: $\Delta x = [-1.5\ -3\ -1.5\ -3\ 0]$, $\Delta y = [1.2\ 1.2\ -1.2\ -1.2\ 0]$, Δx and Δy denote the position error among followers and the leader. Other parameters are set as follows: η = 0.5, ε = 1, R = 1, r = 0.8, q = 0.5. Three scenarios are simulated, obstacle avoidance with one linear motion obstacle, obstacle avoidance with one circular motion obstacle, and broken topology. In the former two scenarios, the topology is broken but it can be recovered. In the last scenario the topology is broken without recovery, thus, system can not reach consensus.

A. Linear Motion

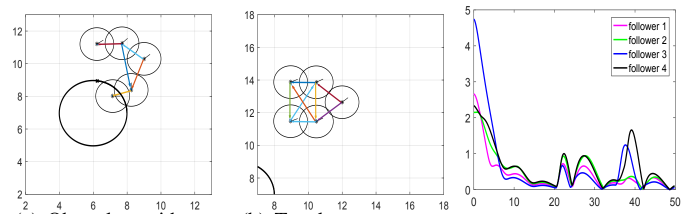
In this condition, the obstacle has initial position of $[p_x\ \ p_y] = [5\ \ 7]$ and the velocity is set as $[v_x\ \ v_y] = [-0.1\ \ -0.1]$.

Fig. 5(a) shows the formation of the system before it encounters the obstacle. It is showed in Fig. 5(b) that the information channels between agent 4 and agent 2, agent 4 and agent 1 are broken because of the obstacle avoidance of agent 4. Finally, in Fig. 5(c), all the information channel have recovered because of the formation algorithm.

Fig. 5(d) shows the x-y error between followers and the leader. There are some peaks in the figure, for the agents meet obstacles at that times. And the error converges to zero in the end, which represents that consensus is achieved.

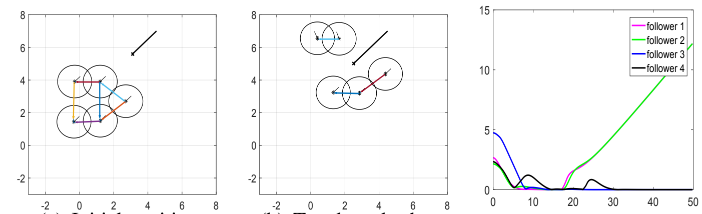
B. Angular Motion

It is shown in Fig. 6(a) that the topology is broken, and it is recovered in Fig. 6(b). In this situation the initial position of



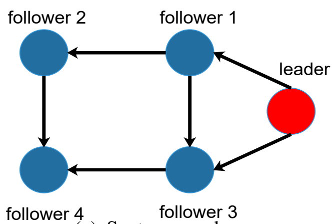
(a) Obstacle avoidance (b) Topology recovery (c) Position error

Fig. 6: Circumferential motion



(a) Initial position (b) Topology broken (c) Position error

Fig. 8: Obstacle avoidance and topology broken



(a) System graph

$$A=\begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

(b) Adjacent matrix

Fig. 7: System topology for broken case

the obstacle is set as $[p_x \ p_y] = [8 \ 7]$, and velocity $[v_x \ v_y] = [-0.4\sin(t) \ 0.4\cos(t)]$. Position error is shown in Fig. 6(c).

C. Topology Broken

In this condition, the system topology graph and adjacent matrix are shown in Fig. 7.

It's easy to check that this graph doesn't satisfy Theorem 2 if we break three edges. Fig. 8(a) shows the initial position of the system, Fig. 8(b) shows that the topology is broken, agent 1 and agent 2 are separated from the system. Fig. 8(c) shows that the system cannot maintain formation in the end.

VI. Conclusion

In this paper, a novel moving obstacle avoidance algorithm and topology recovery conditions based on multi-agent systems with limited communication range are presented. The proposed algorithm has a strong robustness in maintaining formation shape by rebuilding links among agents. System communication topology can be recovered with no extra links added while avoiding obstacles, which can maintain formation under the premise of safety. Simulations demonstrate the efficiency of the proposed algorithm and correctness of the obtained conditions. Future directions include extending the topology condition for common distributed systems and test it on real multi-robot systems.

References

[1] Y. Cai, J. He, W. Yu, and X. Guan, "Consensus-based data statistics in distributed network systems," in Proc. *IEEE CDC*, 2018.

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

[3] J. Su, J. He, P. Cheng, and J. Chen, "Actuator fault diagnosis of a hexacopter: A nonlinear analytical redundancy approach," in Proc. *MED*, 2017

[4] Y. Liu and R. Bucknall, "A survey of formation control and motion planning of multiple unmanned vehicles," *Robotica*, pp. 1–29, 2018.

[5] O. Khatib, "Real-Time Obstacle Avoidance for Manipulators and Mobile Robots". Springer New York, 1986.

[6] E. Rimon and D. E. Koditschek, "Exact robot navigation using artificial potential functions," *IEEE Trans on Robotics & Automation*, vol. 8, no. 5, pp. 501–518, 1992.

[7] J. Borenstein and Y. Koren, "Real-time obstacle avoidance for fast mobile robots," *IEEE Transactions on Systems Man & Cybernetics*, vol. 19, no. 5, pp. 1179–1187, 2002.

[8] J. He, L. Cai, and X. Guan, "Preserving data-privacy with added noises: Optimal estimation and privacy analysis," *IEEE Transactions on Information Theory*, vol. 64, no. 8, pp. 5677–5690, 2018.

[9] Y. S. Nam, B. H. Lee, and M. S. Kim, "View-time based moving obstacle avoidance using stochastic prediction of obstacle motion," *Robotics & Automaton*, vol. 2, pp. 1081–1086, 1996.

[10] D. Hsu, R. Kindel, J. C. Latombe, and S. Rock, "Randomized kinodynamic motion planning with moving obstacles," *International Journal of Robotics Research*, vol. 21, no. 3, pp. 233–255, 2000.

[11] P. Fiorini, "Motion planning in dynamic environments using velocity obstacles," *International Journal of Robotics Research*, vol. 17, no. 7, pp. 760–772, 1998.

[12] E. Rydwik, A. Bergland, L. Forsn, and K. Frandin, "Motion planning in dynamic environments: obstacles moving along arbitrary trajectories," in *Proc. IEEE ICRA*, 2001.

[13] Y. Liang and H. H. Lee, "Decentralized formation control and obstacle avoidance for multiple robots with nonholonomic constraints," in *Proc. ACC*, 2006.

[14] D. L. C. Celso and C. Ricardo, "Dynamic model based formation control and obstacle avoidance of multi-robot systems," *Robotica*, vol. 26, no. 3, pp. 345–356, 2008.

[15] T. T. Yang, Z. Y. Liu, H. Chen, and R. Pei, "Formation control and obstacle avoidance for multiple mobile robots," *Acta Automatica Sinica*, vol. 34, no. 5, pp. 588–593, 2008.

[16] H. Rezaee and F. Abdollahi, "A decentralized cooperative control scheme with obstacle avoidance for a team of mobile robots," *IEEE Transactions on Industrial Electronics*, 61(1):pp. 347–354, Jan 2014.

[17] Q. Jia and G. Li, "Formation control and obstacle avoidance algorithm of multiple autonomous underwater vehicles(auvs)based on potential function and behavior rules," in *Proc. IEEE ICAL*, 2007.

[18] T. Nguyen, H. M. La, T. D. Le, and M. Jafari, "Formation control and obstacle avoidance of multiple rectangular agents with limited communication ranges," *IEEE Transactions on Control of Network Systems*, vol. 4, no. 4, pp. 680–691, 2017.

[19] R. O. Saber and R. M. Murray, "Agreement problems in networks with directed graphs and switching topology," in *Proc. IEEE CDC*, 2003.

[20] X. Dong, Y. Zhou, Z. Ren, and Y. Zhong, "Time-varying formation control for unmanned aerial vehicles with switching interaction topologies," *Control Engineering Practice*, vol. 46, pp. 26 – 36, 2016.

[21] X. Dong and G. Hu, "Time-varying formation control for general linear multi-agent systems with switching directed topologies," *Automatica*, vol. 73, pp. 47 – 55, 2016.

[22] Y. Zhang, G. Song, G. Qiao, and J. Zhang, "Consensus and obstacle avoidance for multi-robot systems with fixed and switching topologies," in *Proc. IEEE ICRB*, 2015.

[23] X. Ai, K. You, and S. Song, "Second-order consensus for multi-agent systems with limited communication range," in *Proc. CCC*, 2014.

[24] Y. Zhang and Y. P. Tian, "Consensus of data-sampled multi-agent systems with random communication delay and packet loss," *IEEE Transactions on Automatic Control*, vol. 55, no. 4, pp. 939–943, 2010.