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

Learning-based Intelligent Attack against Formation Control with Obstacle-avoidance

Yushan Li, Jianping He, Cailian Chen and Xinping Guan

Abstract—**Formation control has attracted considerable attention for its wide applications, e.g, military reconnaissance, environment exploration. However, the formation suffers additional security vulnerabilities due to its distributed fashion, networked communication and openness to outside environments. Existing works focus on detection and countermeasures for some classic attacks, e.g., Denial of Service (DoS), replay and deception attacks. Nevertheless, those attacks are generally from cyberspace and the methods are based on an assumption that the attacker has some knowledge or access to the formation system, like the system dynamics is known or internal nodes are compromised. It remains an open issue given how to design a feasible attack or under what conditions an attack can be implemented. In this paper, we aim to design a feasible and intelligent attack scheme against the obstacle-avoidance of formation control. We describe it as "intelligent" for the following: i) Without any prior information of the system dynamics, the attacker can learn the detection area and goal position of an agent by trial and observation; ii) The obstacle-avoidance mechanism is regressed using support vector regress (SVR) method; iii) The strategy exhibits attack efficiency. Furthermore, a sufficient condition is obtained to guarantee the success of the intelligent attack. Simulations illustrate the effectiveness of the proposed attack scheme.**

I. Introduction

In recent years, multi-agent systems have been a research hotspot. Formation control of mobile agents is a key problem in this field, aiming to keep a group of mobile agents in a specific geometric formation [1]. It has shown great values in many military and industrial applications, e.g, object transportation and manipulation, military search and rescue. Numerous formation control algorithms have been developed in the literature, including behavior-based methods, leader-follower methods, and virtual structure methods, to name a few. Among these approaches, consensus is a key and powerful theoretical tool. The control method combined with consensus is named as consensus-based formation control.

Consensus-based formation control usually models holonomic mobile agents as fully actuated single or double integrators [2]–[6]. For agents with non-holonomic constraints, however, their instantaneous movement is restricted [7], making the problem more challenging. Numerous research have investigated on this area [8]–[12]. Note that collision/obstacle-avoidance is a vital criterion when evaluating these methodologies, and it's also an important part of path planning. In general, path planning algorithms can be roughly divided into two categories, i.e., deterministic (like artificial potential field method), and heuristic (like evolutionary algorithm) [13]. Artificial potential field method is usually combined with consensus-based formation control algorithm. In [14], the researchers design a controller with obstacle/vehicle collision avoidance for non-holonomic agents, which guarantees to track trajectory with bounded error. [15] extends the idea and proposes a distributed formation control scheme with communication limited to neighboring agents.

When performing tasks in real environments (like military reconnaissance), the formation is faced with some security issues due to its distributed characteristic and high openness to outside world. The security shows the ability of a system to govern malicious behaviors or unanticipated events [16]. Attacks against a formation system can be roughly divided into three categories: DoS, replay, and deception attacks [17]. Much efforts have been devoted to designing corresponding countermeasures. [18] addresses the problem of ensuring trustworthy computation in a linear consensus network with misbehaving agents. The problem of estimating the state of a linear system has been studied given measurements are corrupted in [19]. In [20], undetectable and unidentifiable attacks are characterized and detection filters are designed.

However, most of the existing research rely on a baseline premise that the attacker has some knowledge or access to the formation system. For example, malicious agents have knowledge of network structure or nodes' states [18], the packets transmitted over network are corrupted [21] and links among agents are altered by an adversary [22]. Thus, there still remain potential gaps between theory and practice, concerning how to implement the attack or under what conditions the attack is possible to be launched. Motivated by this, we design a feasible and intelligent attack scheme against the obstacle-avoidance of formation control. The major challenge is that the attacker needs to figure out what kind of information is necessary, how to obtain and then use them to launch attacks. We refer to the attack as a scheme without prior information because no knowledge of the formation or access to the system is needed. First, we show how the attacker attains the necessary information of the target. Then, we propose a strategy to implement the attack. Finally, we provide simulation results.

Our study provides new insights into the security issues for formation control. The main contributions of this paper are summarized as follows:

- To the best of our knowledge, this is the first time to consider the feasibility of an attack against formation

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: yushan_li@sjtu.edu.cn, jphe@sjtu.edu.cn, cailianchen@sjtu.edu.cn, xpguan@sjtu.edu.cn. This research work is partially sponsored by the National Key R&D Program of China 2017YFE0114600, NSF of China (61622307, 61603251) and NSF of Shanghai Municipality of China(18ZR1419900).

978-1-5386-7926-5/$31.00 ©2019 AACC

control without any prior information of the system dynamics for an attacker.

- We propose an intelligent attack scheme for the attacker. It can learn the obstacle detection area of an agent through trial and observation. By the collected data and SVR method, the obstacle-avoidance mechanism is regressed. Then, an intelligent attack can be launched such that the victim agent moves into the preset trap area.
- We obtain a sufficient condition for the attacker to launch the attack successfully. Extensive simulations are conducted to illustrate the feasibility and effectiveness of our proposed scheme.

The rest of this paper is organized as follows. In Section II, the basics of formation control with obstacle-avoidance are introduced. The control methods for holonomic and non-holonomic are both considered. The attack strategy is proposed in Section III. Relative simulation results are shown in Section IV. Finally, Section V concludes this paper.

II. Consensus-based Formation Control

In this section, we first introduce some basics of graph theory. Then, we present consensus-based formation control for both holonomic and non-holonomic agents. Following this, we show how artificial potential approach and dynamic window approach work in formation control.

A. Graph Theory Basics

Let $\mathcal{G}$=($\mathcal{V}$,$\mathcal{E}$) be a directed graph to model the communication topography among agents, where $\mathcal{V} = \{v_1, \cdots, v_N\}$ is the finite set of nodes and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of edges. An edge $(v_i, v_j) \in \mathcal{E}$ indicates that v_j can receive information from v_i. The adjacency matrix $A = [a_{ij}]_{N\times N}$ of a graph is defined such that $a_{ij} > 0$ if $(v_i, v_j) \in \mathcal{E}$, else $a_{ij} = 0$.

A directed path is a sequence of nodes v_1, v_2, $\cdots$, v_r such that $(v_i, v_{i+1}) \in \mathcal{E}$, $i \in \{1, 2, \cdots, r-1\}$. A directed graph has a (directed) spanning tree if there exists at least one node having a directed path to all other nodes. Let $N_i = \{j \in \mathcal{V} : a_{ij} \neq 0\}$ denote the set of neighbors of node v_i. The Laplacian matrix of the graph $\mathcal{G}$ is defined as $P = \text{diag}(A \cdot \mathbf{1}) - A$. Note that the communication digraph $\mathcal{G}$ of the formation must have a spanning tree to achieve consensus.

B. Formation Control of Mobile agents

We adopt common algorithms for formation control. Each agent of the formation can measure the relative positions of its neighbors. For simplicity, we first introduce the consensus-based algorithm for mobile agents with holonomic constraints.

Consider a group of *N* mobile agents in an *n*-dimensional space, the dynamic of each agent is given by

$$\begin{cases} \dot{p}_i = v_i, \\ \dot{v}_i = u_i, \end{cases} \tag{1}$$

with vector position $p_i \in R^n$, speed $v_i \in R^n$ and acceleration input $u_i \in R^n$. For motion in the 2-D plane, $p_i = [x_i, y_i]'$ is the coordinates of agent *i* in X-Y coordinates. Define the constant desired position of agent *i* relative to moving formation leader p_0 as Δ_i, and the leader's velocity as v_0. A second-order control algorithm for each agent is given by

$$\begin{aligned} u_i = &\dot{v}_0 + \gamma k_v(v_0 - v_i) + k_p(p_0 + \Delta_i - p_i) \\ &+ \sum_{j\in N_i} ca_{ij}\left((p_j - \Delta_j) - (p_i - \Delta_i) + \gamma(v_j - v_i)\right), \end{aligned} \tag{2}$$

where c, γ, k_p, k_v are tuning parameters. Note that if the graph has a spanning tree and all eigenvalues of corresponding Laplacian matrix are real, setting c,γ, k_p, $k_v > 0$ can guarantee the convergence of (2) [23].

For non-holonomic mobile agents (car-like wheeled mobile agents, unicycles, etc.), their motion is controlled directly by linear velocity v and angular velocity ω or velocities of two driving wheels. The kinematics is modeled by the following non-linear ordinary differential equations (ODEs)

$$\begin{cases} \dot{x}_i = v_i \cos(\theta_i), \\ \dot{y}_i = v_i \sin(\theta_i), \\ \dot{\theta}_i = \omega_i, \end{cases} \tag{3}$$

where $\theta_i \in [0, 2\pi)$ is the orientation of agent i with respect to the x axis. As in [24], the "hand" position of the agent is defined as $\mathbf{h} = (x_h, y_h)$, which lies a distance L from the center p along the agent's axis of orientation. By simple transformation, we have

$$\begin{bmatrix} \dot{x}_h \\ \dot{y}_h \end{bmatrix} = \begin{bmatrix} \cos\theta & -L\sin\theta \\ \sin\theta & L\cos\theta \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}. \tag{4}$$

The kinematics of the hand position is holonomic for $L \neq 0$. In this way, the control problem is simplified and sufficient for the purpose of this paper. According to [25], the formation control can be achieved by

$$\begin{aligned} u_i = &\frac{1}{d_i}\sum_{j\in N_i}^{n} a_{ij}[\dot{p}_j - \gamma(p_i - p_j - \Delta_i + \Delta_j)] \\ &+ \frac{1}{d_i} a_{i0}[\dot{p}_0 - \gamma(p_i - p_0 - \Delta_i)], \end{aligned} \tag{5}$$

where d_i is in-degree of agent i, and Δ_i, p_i are the same as in (2). Utilizing (4) and (5), we obtain

$$\begin{bmatrix} v_i \\ \omega_i \end{bmatrix} = \begin{bmatrix} \cos\theta_i & \sin\theta_i \\ -\frac{1}{L_i}\sin\theta_i & \frac{1}{L_i}\cos\theta_i \end{bmatrix} \begin{bmatrix} u_{xi} \\ u_{yi} \end{bmatrix}. \tag{6}$$

The controller (5) makes the agents gradually form the desired formation such that $v_i \to v_0$ and $\omega_i \to \omega_0$.

It should be pointed out that the above two types of formation control are the cornerstone of most applications of mobile agents formation. In the following sections, we will propose an intelligent attack scheme that can be applied to both of them, making our attack generic in practice.

C. Obstacle-avoidance Algorithm

It's a fundamental requirement for a multi-agent formation to have the ability to avoid collision with agents and obstacles. We choose the classic artificial potential approach [26], because it's based on local information and has good adaptability considering complicated environment changes. The basic idea is that when an agent detects an obstacle or another agent, it produces a repulsive potential field, with the artificial force acting in the negative direction of the potential gradient. The algorithm is given by

$$u_{rep} = \begin{cases} k_{rep}\left(\frac{1}{\rho(p,p_{obs})} - \frac{1}{\rho_0}\right)\frac{\nabla\rho(p,p_{obs})}{\rho^2(p,p_{obs})}, \text{if } \rho(p,p_{obs}) \leq \rho_0; \\ 0 \qquad\qquad\qquad\qquad\qquad\quad, \text{if } \rho(p,p_{obs}) > \rho_0, \end{cases} \tag{7}$$

where p and p_{obs} are the coordinates of the agent and obstacle, respectively, and $\rho\left(p, p_{obs}\right) = \|p - p_{obs}\|_2$.

Regarding u_i of (2) and (5) as attraction force term, we combine u_{rep} with u_i to achieve formation control with obstacle-avoidance. Then, we have

$$u_{i,final} = u_i + u_{i,rep}, \tag{8}$$

where $u_{i,final}$ represents the final input of agent i.

For the leader of the formation, the dynamic window approach (DWA) is applied to guide the whole formation's movement. DWA is a velocity-based local planner which calculates the optimal collision-free velocity. The optimization goal is to select a heading and velocity that drives the agent to the goal with the maximum clearance from an obstacle. For more details, readers are referred to [27].

Remark 1. *Due to the leader-follower architecture of the formation, the followers only need to track the leader's movement by the consensus-based algorithm. If the leader is under attack, the influence will be spread in the whole formation. Therefore, it's very natural for an attacker to choose the leader as the direct attack target.*

III. Intelligent Attack Scheme

In this section, an intelligent attack scheme against the obstacle-avoidance of formation control is proposed. We denote the attacker as $\mathrm{R_a}$ and the victim as $\mathrm{R_v}$ hereafter. An assumption is made as follows:

Assumption 1. *The attacker can move faster than agents of the formation, and has the ability to sense objects.*

When a running agent of the formation encounters an obstacle within its detection area, it will evaluate the obstacle's influence and take corresponding action, deviating from its desired trajectory. Inspired by this, we choose the obstacle-avoidance of the formation as the focus of the intelligent attack scheme. It's assumed that the leader moves between two fixed workplace, while others follow them. This is a very common scenario in applications of formation control. Our proposed scheme consists of the following parts:

- First, without any prior knowledge, $\mathrm{R_a}$ needs to acquire basic information of $\mathrm{R_v}$ by sampling. The information should reflect the states of $\mathrm{R_v}$'s motion, such as position and heading, velocity and acceleration velocity.
- Then, $\mathrm{R_a}$ is supposed to learn $\mathrm{R_v}$'s detection area and goal position through trial and observation. Based on that, $\mathrm{R_a}$ can further collect data of $\mathrm{R_v}$'s avoiding obstacles and regress the obstacle-avoidance mechanism using SVR.
- Finally, the designed attack is launched against the formation with all necessary information obtained.

A. Pre-sampling

Denote the posture of an agent as $p = \begin{bmatrix} x & y & \theta \end{bmatrix}'$, which is updated every periodic control time T. T is a fixed small constant determined by the system. The agent's trajectory during the period T can be approximated as a straight line. $\mathrm{R_a}$ is able to measure its relative displacement with a moving object. After three consecutive sampling moments, the instantaneous variables of $\mathrm{R_v}$'s motion are estimated by

$$\begin{cases} v_{x,k+1} = \dfrac{x_{k+1} - x_k}{T}, & v_{y,k+1} = \dfrac{y_{k+1} - y_k}{T}, \\ v_{k+1} = \sqrt{v_{x,k+1}^2 + v_{y,k+1}^2}, & \theta_{k+1} = \arctan\dfrac{v_{y,k+1}}{v_{x,k+1}}, \\ a_{k+1} = \dfrac{v_{k+1} - v_k}{T}, & \omega_{k+1} = \dfrac{\theta_{k+1} - \theta_k}{T}, \end{cases} \tag{9}$$

where the subscript *k* is the sampling time t_k, and *v, w, a* represent its linear, angular and accelerated velocities, respectively. To make (9) completely available, the sampling period needs to be small enough (e.g, 10ms) and $\mathrm{R_v}$ always has the last three groups of data stored to calculate (9). For holonomic mobile agents, this process is much easier with higher precision and we can directly utilize v_x, v_y. To make a unified statement, let $v_1 = v, v_2 = \omega$ if the formation agents are non-holonomic or $v_1 = v_x, v_2 = v_y$ if holonomic.

B. Intentional-learning

It's a common sense that the obstacle-avoidance of mobile agents is based on detecting its surrounding environments. The most generalized detection area is a circular region. If we do not consider obstacles behind an agent while it's moving forward, the detection area is modeled as a sector directly. Then, the primary goal of $\mathrm{R_a}$ is to infer the radius D and angle range α of the sector.

Here we denote the detection area as (D, α). When $\mathrm{R_a}$ moves close to $\mathrm{R_v}$, $\mathrm{R_a}$ makes a record of $\mathrm{R_v}$'s relative position, heading and bearing with it. After a period $\tilde{T}$(e.g, set 1s), $\mathrm{R_a}$ do the measurement again. With the two readings, $\mathrm{R_a}$ is able to calculate $\mathrm{R_v}$'s position variation Δs and heading variation $\Delta\theta$ after $\tilde{T}$. Considering $\tilde{T} = NT$, during time slot $[t_i, t_i + \tilde{T}]$, we have

$$\begin{cases} \Delta s(i) = \sqrt{\left(\Delta x_{i,i+N}\right)^2 + \left(\Delta y_{i,i+N}\right)^2}, \\ \Delta\theta(i) = \arctan\dfrac{\Delta y_{i,i+N}}{\Delta x_{i,i+N}}, \end{cases} \tag{10}$$

where $\Delta x_{i,i+N} = x(t_i + NT) - x(t_i)$ and $\Delta y_{i,i+N} = y(t_i + NT) - y(t_i)$. Note that in normal situations, $\mathrm{R_v}$ goes straight forward to the goal, i.e., $\Delta\theta = 0$. Therefore, we assume $\mathrm{R_a}$ is detected as an obstacle within (D, α) by $\mathrm{R_v}$ if $\Delta\theta \neq 0$. This whole process is illustrated in Fig. 1.

We present Algorithm 1 for $\mathrm{R_a}$ to learn $\mathrm{R_v}$'s detection area and goal position. The process is shown in Fig. 2. $\mathrm{R_a}$ records three trajectories $track(i)$ $(i = 1, 2, 3)$, which are segments

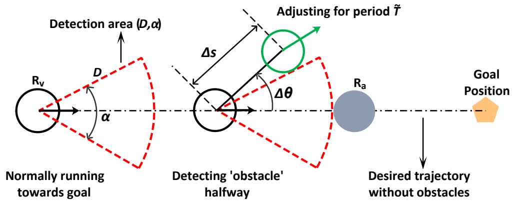

Adjusting for period $\tilde{T}$
Detection area (D,α)
Δs
D
Rv
α
Δθ
Ra
Goal Position
Normally running towards goal
Detecting 'obstacle' halfway
Desired trajectory without obstacles

Fig. 1: $\mathrm{R_a}$'s reaction after detecting $\mathrm{R_v}$ as an obstacle.

Algorithm 1: Learning $\mathrm{R_v}$'s Detection Area and Goal

```
Input: R_v's posture $P_v$, posture regulating variables $\Delta d$, $\Delta\alpha$
       for every trial
Output: detection area $(D,\alpha)$ and goal position
Initialize: R_a moves to remote posture $P_a$ such that R_a is
    directly ahead of R_v, i.e. $\varphi_r$=0;
while $\Delta\theta = 0$ do
    $d = d - \Delta d$;
    $P_a = P_a + d$;
    Calculate $\Delta\theta$ ;
    if $\Delta\theta \neq 0$ then
        Record $D$, and the following trajectory as $track1$;
    end
end
Reset: R_a moves to a posture $P_a$ such that R_a is in R_v's $+90°$
    direction with $distance = D$, i.e., $\alpha_1 = \theta + 90°$;
while $\Delta\theta = 0$ do
    $\alpha_1 = \alpha_1 - \Delta\alpha$;
    Calculate $\Delta\theta$;
    if $\Delta\theta \neq 0$ then
        Record $\alpha_1$, and the following trajectory as $track2$;
    end
end
Reset: R_a moves to another posture $P_a'$ such that R_a is in R_v's
    $-90°$ direction with $distance = D$, i.e., $\alpha_2 = \theta - 90°$. Then
    R_a does the same process again to obtain a new $\alpha_2$ and $track3$;
$\alpha = [\alpha_2, \alpha_1]$, compute goal using three trajectories;
return $(D,\alpha)$ and goal position
```

of straight lines leading to the goal. They are used to find the goal position by solving the following problem

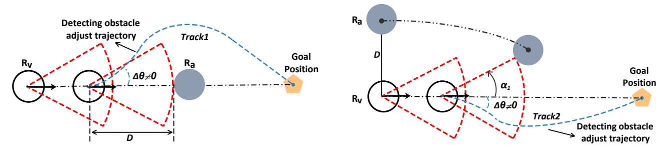
$$\bar{A}x = b, \tag{11}$$

where $\bar{A} \in R^{3\times 2}$,$\bar{b} \in R^{3\times 1}$ and $x \in R^{2\times 1}$ is the coordinates in X-Y plane. Apparently (11) is an overdetermined equation problem, which has an exact solution only when the measurements are totally accurate. Therefore, we can only obtain the least square solution $\hat{x}$ of (11). Thus, we define the estimation error of $\hat{x}$ by

$$\varepsilon_{\hat{x}} = (1/N)\sum\nolimits_{i=1}^{N} \mathrm{d}(\hat{x}, track[i]), \tag{12}$$

where $\mathrm{d}(\hat{x}, track[i])$ is the distance from $\hat{x}$ to $track[i]$, and N is the number of recorded trajectories. Since the real goal is generally a region, $\varepsilon_{\hat{x}}$ is small enough and $\hat{x}$ is acceptable.

Next, $\mathrm{R_a}$ can move inside (D,α) of one agent. In this step, $\mathrm{R_a}$ also records its relative distance d_r and bearing φ_r with $\mathrm{R_v}$, and $\mathrm{R_v}$'s heading deviation θ' with the goal point. Once $\mathrm{R_a}$ is detected, it stores two groups of data during the next period $\tilde{T}$. The data groups are defined as


Detecting obstacle adjust trajectory
Track1
Rv
Ra
Δθ≠0
D
Goal Position
Ra
D
α1
Rv
Δθ≠0
Track2
Goal Position
Detecting obstacle adjust trajectory

(a) Learning $\mathrm{R_v}$'s detection radius D. (b) Learning $\mathrm{R_v}$'s detection ang α_1.

Fig. 2: Algorithm 1: Learning $\mathrm{R_v}$'s detection area (D,α).

$$\begin{cases} input(i) = [\theta'(i), v_1(i), v_2(i), a(i), d_r(i), \varphi_r(i)], \\ output(i) = [\Delta s(i), \Delta\theta(i)]. \end{cases} \tag{13}$$

Based on this, we define the sampled feasible set $\mathcal{M}$ as

$$\mathcal{M} = \left\{ \mathop{\cup}_{i\in\mathcal{F}} (input(i), output(i)) \right\}, \\ \text{where } \mathcal{F} = \left\{ k \in Z^{+} : d_r(k) \leq D, \varphi_r(k) \in \alpha \right\}. \tag{14}$$

Then, we propose Algorithm 2, by which $\mathrm{R_a}$ collects the data set $\mathcal{M}$ and uses it as training data to learn the obstacle-avoidance mechanism of $\mathrm{R_v}$. Note $timer\_limit$ is preset trial limit for $\mathrm{R_a}$ and the classic SVR method is used.

Remark 2. *SVR method has good performance on non-linear regression and strong generalization ability when the amount of data isn't vast. It's insensitive to algorithm models.*

Now we give a detailed analysis for this algorithm. Consider the time slot $\tilde{T} = NT$, we have

$$\begin{cases} x_{k+N} - x_k = \Delta x_{k,k+N} = \sum\nolimits_{i=0}^{N-1} v_{k+i}\cdot T\cdot \cos\theta_{k+i}, \\ y_{k+N} - y_k = \Delta y_{k,k+N} = \sum\nolimits_{i=0}^{N-1} v_{k+i}\cdot T\cdot \sin\theta_{k+i}, \\ \theta_{k+N} - \theta_k = \Delta\theta_{k,k+N} = \sum\nolimits_{i=0}^{N-1} \omega_{k+i}\cdot T, \end{cases} \tag{15}$$

where v_{k+i} and ω_{k+i} are determined by (6). Ideally, $\tilde{T}$ should be set such that $\tilde{T} = T$, i.e., $\mathrm{R_a}$ does the measurement every period T. Nevertheless, $\mathrm{R_a}$'s executive time to sample and compute is beyond T, or $\mathrm{R_v}$'s displacement change during T is smaller than $\mathrm{R_a}$'s minimal detectable displacement in practice. Therefore, longer time is necessary. We only sample a group of data at one moment (e.g., time k), taking the limitation of storage and computation ability of $\mathrm{R_a}$ into consideration. The changed model (15) is given by

$$\begin{cases} x_{k+N} - x_k \approx \Delta\hat{x}_{k,k+N} = N\cdot(v_k\cdot T\cdot\cos\theta_k), \\ y_{k+N} - y_k \approx \Delta\hat{y}_{k,k+N} = N\cdot(v_k\cdot T\cdot\sin\theta_k), \\ \theta_{k+N} - \theta_k \approx \Delta\hat{\theta}_{k,k+N} = N\cdot(\omega_k\cdot T). \end{cases} \tag{16}$$

Simplifying (16), we obtain

$$\begin{cases} \Delta\hat{d}_{k,k+N} = N\cdot(v_k\cdot T), \\ \Delta\hat{\theta}_{k,k+N} = N\cdot(\omega_k\cdot T). \end{cases} \tag{17}$$

The SVR method is applied to regress the mapping relationship between *input* and *output* essentially. There exit inevitable model errors using (17). Ignoring the subscripts, let

Algorithm 2: Regress Obstacle-avoidance Mechanism

```
Input: R_a's detection area $(D, \alpha)$, intentional-learning's
       $timer\_limit$, $\mathcal{M} = \emptyset$
Output: Obstacle-avoidance mechanism $f$
Initialize: R_a moves to a relatively far position from R_v;
for $i \leftarrow 1$ to $timer\_limit$ do
    R_a moves into a random position in $(D, \alpha)$;
    Compute input(i)=$[\theta'(i),\ v_1(i),\ v_2(i),\ a(i),\ d_r(i),\ \varphi_r(i)]$ at
    time $t_i$;
    Wait for a time slot $\tilde{T}$;
    Compute $output(i) = [\Delta s(i), \Delta\theta(i)]$ at time $(t_i + \tilde{T})$;
    $\mathcal{M} = \mathcal{M} \cup \{input(i), output(i)\}$;
end
Use $\mathcal{M}$ and SVR method to regress $f$;
return $f$
```

$q = [\Delta x, \Delta y, \Delta\theta]'$ and $\hat{q} = [\Delta\hat{x}, \Delta\hat{y}, \Delta\hat{\theta}]'$, and present the following probability

$$P\left[\|q - \hat{q}\| \le g\right] \ge 1 - \varepsilon(g), \tag{18}$$

where g is the error between real value and estimated value, and $1 - \varepsilon(g)$ represents the confidence we have in $\hat{q}$. $\varepsilon(g)$ is a monotonic decreasing function of g. The smaller g is, the more reliable $\hat{q}$ is. Naturally, we have

$$\lim_{\tilde{T} \to T} P\left[\|q - \hat{q}\| = 0\right] = 1. \tag{19}$$

Concerning how to choose an appropriate $\tilde{T}$, we present a criteria given by

$$\tilde{T} = \min\{S_T = NT : N \in Z^+, \Delta d_{S_T} \ge d_m\}, \tag{20}$$

where Δd_{S_T} is R$_v$'s displacement during S_T, d_m is R$_a$'s minimal detectable displacement.

C. Attack Strategy

The key idea of the strategy is that R$_a$ drives R$_v$ to a preset trap (denoted as Q), which could be a pothole, a cage or an area where the communication is invalid. In fact, if a moving agent detects a static obstacle, it would go around the obstacle in an arc while towards the goal. After the obstacle is off the detection area, the agent goes directly for the goal, and we denote this trajectory as Arc_Line. Therefore, we use an arc to approximate the trajectory of R$_v$'s avoiding obstacles, then all we need to do is just to prepare the trap, decide when to stop attacking and wait for R$_v$'s running towards the trap. Note that the trap position cannot be designed arbitrarily, and we propose the following theorem for a feasible Q.

Theorem 1. *Denote the trajectory of R_a as L while obstacle-free, and the square area as S with R_v's first influenced position P_v and the goal point being the diagonals. If*

1) *Trap Q and R_a locate opposite sides of L;*
2) *Trap $Q \in Arc\_Line$ and $Q \in S$,*

then R_a can always drive R_v to Q.

It's remarkable that this theorem is relatively conservative. We believe the same idea applies to more general cases with additional consideration of specific algorithms. The attack process is illustrated in Algorithm 3.

Algorithm 3: Intelligent Attack Strategy

```
Input: obstacle-avoidance mechanism $f$, $(D, \alpha)$, trap $Q$,
       tunning constant $\beta$.
Initialize: R_a moves into detection area $(D, \alpha)$ of R_v;
while $R_a$ is running do
    R_a predicts $[\Delta s, \Delta\theta]$ of R_v;
    R_a moves $\beta\Delta s$ along $(\theta + \Delta\theta)$ direction;
    if $Q \in Arc\_Line$ and $Q \in S$ then
        R_a stops;
    end
end
```

IV. Simulation

In this section, we model non-holonomic agents as simulation object. First, the critical steps of the proposed attack scheme are shown. Then, we illustrate the effectiveness of the proposed attack strategy. Based on that, we present the simulation results of the attack against a formation.

Fig. 4(a) shows that R$_a$ collects a series of *input* and *output* by sequential "intentional learning". It's a feasible means to regress the obstacle-avoidance algorithm using the data, when no prior information of that is available. Fig. 4(b) shows a simple and rough attack: once R$_a$ appears in (D, α), it predicts R$_v$'s next move, and runs in the predicted direction with faster speed and repeats this process. Ideally, R$_v$ will keep avoiding R$_a$. However, this attack cannot proceed all the time. We conclude the leading cause lies in two parts: 1) the obstacle-avoidance algorithm has taken the goal point into consideration; 2) each prediction will produce certain errors. By comparison, the proposed attack strategy, as shown in Fig. 5, enables an attacker to drive the victim to a preset trap, remedying the deficiency of the former one.

Fig. 6 shows that the attack strategy is applied to formation control. Two cases are designed where the formation shape is a triangle and a straight line, respectively. It illustrates our strategy is feasible and efficient, with the whole formation breaking down after the leader is attacked.

V. Conclusion

In this paper, we investigate the security problem of formation control. Instead of focusing on detection or countermeasures for attacks based on some assumptions about the system, we present an intelligent attack scheme against the obstacle-avoidance of formation control without any prior information of the system. The proposed scheme enables

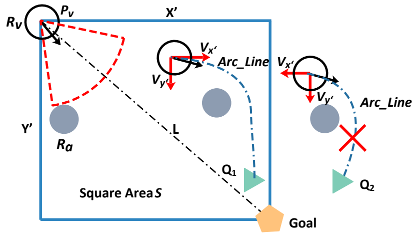
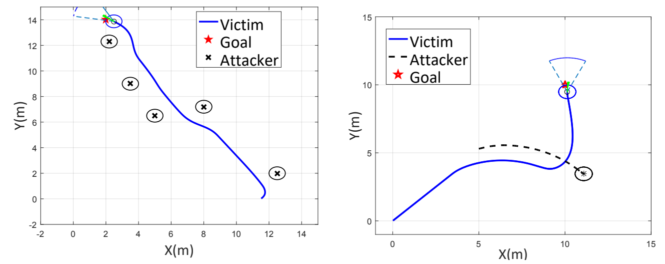


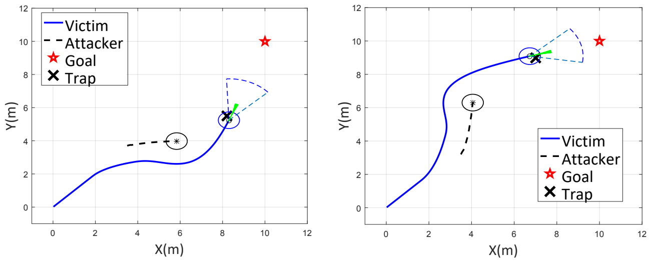
Fig. 3: An example of Theorem 1. Trap position Q_1 within S is feasible, while Q_2 outside S is not.



(a) Intentional learning: $\mathrm{R_a}$ collects data of $\mathrm{R_v}$ at '×' positions consecutively. $\mathrm{R_v}$ moves from $(11.5, 0)$ to $(2, 14)$.

(b) Simple attack: once $\mathrm{R_a}$ appears in (D, α), it makes continuous impacts on $\mathrm{R_v}$ without purpose. This illustrates attack to the right.

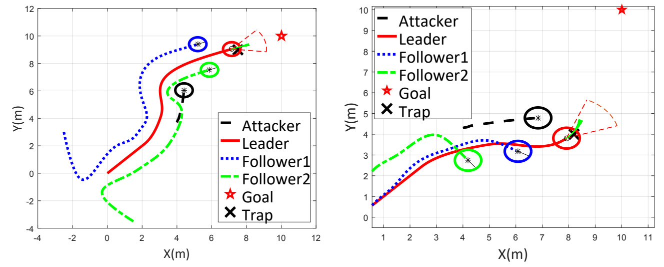
Fig. 4: Intentional learning and simple attack



(a) trap point $Q = (8.2, 5.5)$.

(b) trap point $Q = (7, 9)$.

Fig. 5: Intelligent attack: $\mathrm{R_a}$ drives $\mathrm{R_v}$ to a preset trap.



(a) trap point $Q = (8.2, 5.5)$, formation shape: triangle.

(b) trap point $Q = (7, 9)$, formation shape: straight line.

Fig. 6: Attack against the formation.

the attacker to learn the victim agent's goal position and obstacle-detection area. Furthermore, the obstacle mechanism is regressed by using collected data and SVR method. The feasibility of our proposed attack strategy is proved. Sufficient accuracy is guaranteed when the measurement is with small noises. Extensive simulations confirm the effectiveness of the intelligent attack scheme. Future directions include extending the idea to more complicated scenarios and establishing a unified mathematical framework.

References

[1] Y. Q. Chen and Z. Wang, "Formation control: A review and a new consideration," in *Proceedings of Intelligent Robots and Systems*, pp. 3181–3186, 2005.

[2] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1520–1533, 2004.

[3] W. Ren, R. W. Beard, and E. M. Atkins, "Information consensus in multivehicle cooperative control," *IEEE Control Systems*, vol. 27, no. 2, pp. 71–82, 2007.

[4] Z. Meng, W. Ren, Y. Cao, and Z. You, "Leaderless and leader-following consensus with communication and input delays under a directed network topology," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 41, no. 1, pp. 75–88, 2011.

[5] J. He, W. Chen, and L. Gao, "Some distributed algorithms for quantized consensus problem," in *International Conference on Intelligent Computing*. Springer, 2009, pp. 443–452.

[6] C. Zhao, J. He, P. Cheng, and J. Chen, "Performance analysis of discrete-time average consensus under uniform constant time delays," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 11 725–11 730, 2017.

[7] K. D. Listmann, M. V. Masalawala, and J. Adamy, "Consensus for formation control of nonholonomic mobile robots," in *Proceedings of Robotics and Automation*, 2009.

[8] Z. Lin, B. Francis, and M. Maggiore, "Necessary and sufficient graphical conditions for formation control of unicycles," *IEEE Transactions on Automatic Control*, vol. 50, no. 1, pp. 121–127, 2005.

[9] W. Dong and J. A. Farrell, "Cooperative control of multiple nonholonomic mobile agents," *IEEE Transactions on Automatic Control*, vol. 53, no. 6, pp. 1434–1448, 2008.

[10] W. Wang, J. Huang, et al, "Distributed adaptive control for consensus tracking with application to formation control of nonholonomic mobile robots," *Automatica*, vol. 50, no. 4, pp. 1254–1263, 2014.

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

[12] H. Du, G. Wen, Y. Cheng, Y. He, and R. Jia, "Distributed finite-time cooperative control of multiple high-order nonholonomic mobile robots," *IEEE Trans. Neural Netw. Learn. Syst*, vol. 28, pp. 2998–3006, 2017.

[13] C. Tam, R. Bucknall, and A. Greig, "Review of collision avoidance and path planning methods for ships in close range encounters," *The Journal of Navigation*, vol. 62, no. 3, pp. 455–476, 2009.

[14] S. Mastellone, D. M. Stipanović, et al, "Formation control and collision avoidance for multi-agent non-holonomic systems: Theory and experiments," *The International Journal of Robotics Research*, vol. 27, no. 1, pp. 107–126, 2008.

[15] S. A. Barogh, E. Rosero, and H. Werner, "Formation control of non-holonomic agents with collision avoidance," *ACC*, 2015.

[16] Q. Zhu and T. Basar, "Game-theoretic methods for robustness, security, and resilience of cyberphysical control systems: Games-in-games principle for optimal cross-layer resilient control systems," *IEEE Control Systems*, vol. 35, no. 1, pp. 46–65, 2015.

[17] D. Ding, Q.-L. Han, Y. Xiang, X. Ge, and X.-M. Zhang, "A survey on security control and attack detection for industrial cyber-physical systems," *Neurocomputing*, vol. 275, pp. 1674–1683, 2018.

[18] F. Pasqualetti, A. Bicchi, and F. Bullo, "Consensus computation in unreliable networks: A system theoretic approach," *IEEE Transactions on Automatic Control*, vol. 57, no. 1, pp. 90–104, 2012.

[19] H. Fawzi, P. Tabuada, and S. Diggavi, "Secure state-estimation for dynamical systems under active adversaries," in *Proceedings of Communication, Control, and Computing (Allerton)*, 2011.

[20] F. Pasqualetti, F. Dörfler, and F. Bullo, "Attack detection and identification in cyber-physical systems," *IEEE Transactions on Automatic Control*, vol. 58, no. 11, pp. 2715–2729, 2013.

[21] M. Zhu and S. Martinez, "Stackelberg-game analysis of correlated attacks in cyber-physical systems," in *Proceedings of ACC*, 2011.

[22] Z. Feng, G. Hu, and G. Wen, "Distributed consensus tracking for multi-agent systems under two types of attacks," *International Journal of Robust and Nonlinear Control*, vol. 26, no. 5, pp. 896–918, 2016.

[23] W. Ren and R. W. Beard, *Distributed consensus in multi-vehicle cooperative control*. Springer, 2008.

[24] J. R. Lawton, R. W. Beard, and B. J. Young, "A decentralized approach to formation maneuvers," *IEEE Transactions on Robotics and Automation*, vol. 19, no. 6, pp. 933–941, 2003.

[25] W. Ren, "Consensus tracking under directed interaction topologies: Algorithms and experiments," in *Proceedings of ACC*, 2008.

[26] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," in *Autonomous Robot Vehicles*. Springer, 1986, pp. 396–404.

[27] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *IEEE Robotics & Automation Magazine*, vol. 4, no. 1, pp. 23–33, 1997.