Robust distributed control within a curve virtual tube for a robotic swarm under self-localization drift and precise relative navigation

Yan Gao | Chenggang Bai | Quan Quan

School of Automation Science and Electrical Engineering, Beihang University, Beijing, People's Republic of China

Correspondence
Quan Quan, School of Automation Science and Electrical Engineering, Beihang University, Beijing 100191, People's Republic of China. Email: qq_buaa@buaa.edu.cn

Funding information
National Natural Science Foundation of China, Grant/Award Numbers: 61973015, 62103335

Abstract
To guide the movement of a robotic swarm in a corridor-like environment, a curve virtual tube with no obstacle inside was designed in our previous work. This article generalizes the controller design to the condition that all robots have self-localization drifts and precise relative navigation, where the flocking algorithm is introduced to reduce the negative impact of the self-localization drift. Similar to the “many wrongs principle,” it is shown that the cohesion behavior and the velocity alignment behavior are able to reduce the influence of the position measurement drift and the velocity measurement error, respectively. For convenience in practical use, a modified vector field controller with five control terms is put forward. Finally, the effectiveness of the proposed method is validated by numerical simulations and real experiments.

KEYWORDS
flocking, many wrongs principle, robotic swarm, self-localization drift, virtual tube

1 | INTRODUCTION

In recent years, there has been an increasing interest in robotic swarm systems operating in a corridor-like environment. Corridor-like environments mentioned here refer to narrow indoor corridors, indoor openings (such as doorframes and windows), outdoor complex environments (such as forests and urban environments), narrow water channels, etc. In this process, not only should each robot avoid collisions with obstacles, but all robots need to avoid collisions with each other as well.

To solve the problem of robots operation in a corridor-like environment, many well-designed methods have been proposed, which can be classified as follows: formation, multi-robot trajectory planning, and control-based methods. All robots in the formation usually remain in prespecified poses with respect to their adjacent robots and make transformation operations. The inter-robot collision avoidance is achieved mainly by the fixed geometry. When the formation needs to pass through some narrow spaces or corridors, necessary transformations have to be carried out. The main weakness of the formation is its limited scalability and adaptability. The multi-robot trajectory planning produces collision-free trajectories for all robots with a higher-order continuity. Given a specific goal point, any robot should find a discrete geometric path in the global map first and then optimize the path to a feasible trajectory locally with no conflict with obstacles or other robots’ planned trajectories. However, due to the sharp increase in computational complexity and communication pressure, trajectory planning may become infeasible when multiple robots operate densely. The control-based methods are widely used for the robotic swarm because of their simplicity and accessibility, which are most appropriate for a large robotic swarm, such as the potential-based method, the vector field method, and the control barrier function.
FIGURE 1  A curve virtual tube is designed to guide a quadcopter swarm in an indoor corridor-like environment.

method. Control-based methods usually use a simple controller and they have a good quality to achieve a fast and reactive response to a dynamic environment with low demand for computation and communication resources. Besides, the flocking control is also a type of control-based method. In 1986, Reynolds introduced the Boids model that led to the creation of the first computer animation of flocking. There are three flocking behaviors in the Boids model: velocity alignment behavior, collision avoidance behavior, and cohesion behavior.

In our previous work, a curve virtual tube is proposed for guiding the robotic swarm in a corridor-like environment. A generating curve is inside the curve virtual tube. The curve virtual tube takes advantage of both the multi-robot trajectory planning and the control-based methods. All robots share one planned curve virtual tube and use the same distributed controller for swarm coordination. There is no obstacle inside the curve virtual tube, and robots only need to guarantee no collision with each other or the tube boundary. As shown in Figure 1, the concept of the curve virtual tube is similar to the lane for autonomous road vehicles, the safe flight corridor for quadcopters, and the multi-drone skyway framework CORRIDRONE. The curve virtual tube limits the shape of the robotic swarm to a certain extent, which is similar to Reference 17. In References 18 and 19, the authors make robots track a desired trajectory while avoiding collision with each other. In this process, robots are commanded to keep a formation or keep the swarm cohesive. To guide and control a robotic swarm within a curve virtual tube, we propose a distributed vector field controller, which is a type of control-based method. However, there is no uncertainty considered in Reference 14, which means that all robots can obtain the information precisely and execute the command exactly. This assumption is too ideal for real practice. With regard to control-based methods, some solutions have been proposed to deal with uncertainties in research papers. In References 20, 21, the authors analyze the influence of norm-bounded disturbances of a quadcopter and a fixed-wing UAV in the deviation from the target curve. In Reference 22, the authors present a robust guidance strategy to deal with additive perturbations representing norm-bounded uncertainties.

Most of the studies on control-based methods only consider the influence of norm-bounded matched disturbances. The negative impact of the self-localization drift is rarely considered, which includes position measurement drift and velocity measurement error. In many cases, robots operate in GPS-denied environments and have odometry systems, such as high-precision inertial odometry, wheel odometry, and visual-inertial odometry to get their ego-position and velocity information. It is well known that the position information obtained from state-of-the-art odometry systems is accurate in the short term, but it drifts over time due to accumulated errors. The velocity information obtained from odometry systems is not precise, either. As a result, robots may have collisions with each other and have chaos in the swarm. Besides, position measurement drifts may also cause robots to reach wrong destinations.

In this paper, the robust curve virtual tube passing-through control problem is summarized and solved. To achieve collision avoidance among robots, a traditional method is to share self-observation positions with the known initial positions among robots, which suffers heavily from position measurement drift and communication uncertainties. The controller proposed in this paper can work autonomously without wireless communication and other robots’ IDs, whose premise is that all robots have omnidirectional relative localization equipment to achieve precise relative navigation. Robots can get their neighboring robots’ relative position and relative velocity precisely. No robot needs to identify other robots carefully,
thus the system structure is greatly simplified. The specific relative navigation methods can be found in References 29-31, which are mainly achieved by vision-based methods. It should be noted that “precise relative navigation” here means that the accuracy of the relative navigation between robots is far higher than that of the self-localization of any robot. In real practice, the relative navigation is not perfect, either. For the simplicity of the analysis, the relative navigation between robots is assumed as “precise” in this paper. Hence, precise relative navigation can be viewed as an ideal mathematical assumption.

To modify the original controller in Reference 14, three kinds of flocking behaviors of the Boids model are introduced, which include the velocity alignment behavior, collision avoidance behavior, and cohesion behavior.13 Specifically, the original controller already includes a collision avoidance term to guarantee safety among robots. This paper shows that the cohesion behavior and the velocity alignment behavior can reduce the negative impact of the position measurement drift and the velocity measurement error, respectively. A modified vector field controller is generated by adding two corresponding terms to the original controller. Such a modification method is similar to the well-known “many wrongs principle”,32,33 which implies that group cohesion can act to suppress individual movement directional error so that group navigational accuracy reliably exceeds the one achieved by a singleton. The “many wrongs principle” is also called “wisdom of the crowd” in some literature.34 However, it should be noted that the proposed modified controller cannot always guarantee all robots to keep moving inside the curve virtual tube under the self-localization drift and precise relative navigation, which is unrealistic when the position measurement drift is very large. Such a safety guarantee may be possible if the tube is a physical object and can be measured. For example, the authors in Reference 35 provide a robust controller for a single robot to keep itself inside a physical tunnel.

The major contributions of this paper are summarized as follows:

1. The flocking behaviors of the Boids model are introduced to reduce the negative impact of the self-localization drift when the robotic swarm is passing through a curve virtual tube. The modification principle is similar to the “many wrongs principle.”
2. Formal proofs are proposed to show that the cohesion behavior and the velocity alignment behavior have the ability to reduce the negative impact of position measurement drift and velocity measurement error, respectively.

2 | PRELIMINARIES AND PROBLEM FORMULATION

2.1 | Robot model

2.1.1 | Robot kinematics model

The robotic swarm consists of $M$ mobile robots in $\mathbb{R}^2$. Each robot has a double-integrator holonomic kinematics

\[
\begin{align*}
\dot{\mathbf{p}}_i &= \mathbf{v}_i, \\
\dot{\mathbf{v}}_i &= \mathbf{a}_{ci},
\end{align*}
\]

in which $\mathbf{a}_{ci} \in \mathbb{R}^2$ indicates the acceleration command of the $i$th robot, and $\mathbf{p}_i, \mathbf{v}_i \in \mathbb{R}^2$ stand for the position and velocity of the $i$th robot, respectively. The controller proposed in our previous work14 is a type of vector field controller. Hence, a hierarchical control architecture is necessary. First the vector field $\mathbf{v}_{ci} \in \mathbb{R}^2$ is designed, and then an acceleration command $\mathbf{a}_{ci}$ is designed to make the robot track $\mathbf{v}_{ci}$. As the vector field gives the moving direction and speed of the robot, it corresponds to the robot’s velocity.12,21 Hence, the vector field $\mathbf{v}_{ci}$ is also called the velocity command in the following.

2.1.2 | Three areas around a robot

As shown in Figure 2, the safety area $S_i$ of the $i$th robot at the time $t > 0$ is defined as follows:

\[
S_i(t) = \{ \mathbf{x} \in \mathbb{R}^2 : \| \mathbf{x} - \mathbf{p}_i(t) \| \leq r_s \},
\]
where \( r_s > 0 \) is the safety radius. For all robots, no conflict with each other implies that \( S_i \cap S_j = \emptyset \), where \( i, j = 1, \ldots, M, i \neq j \). The detection area \( D_i \) depends on the detection range of the relative navigation equipment, which is shown as follows:

\[
D_i (t) = \{ x \in \mathbb{R}^2 : \| x - p_i (t) \| \leq r_d \}, \tag{3}
\]

where \( r_d > 0 \) is the detection radius. The set \( N_{m,i} \) is defined as the collection of all IDs of other robots within \( D_i \), which is shown as follows:

\[
N_{m,i} = \{ j : \| p_j (t) - p_i (t) \| \leq r_d, j \neq i \}. \tag{4}
\]

Finally, at the time \( t > 0 \), the cohesion area \( C_i \) of the \( i \)th robot is defined as follows:

\[
C_i (t) = \{ x \in \mathbb{R}^2 : r_c \leq \| x - p_i (t) \| \leq r_d \}, \tag{5}
\]

where \( r_c > 0 \) is the cohesion lower bound. It is required that \( r_d > r_c > r_s \). The cohesion area limits the action range of the cohesive behavior between two robots, which will be introduced in detail in the following.

### 2.1.3 Robot observation model

In this article, it is assumed that each robot has an odometry system to get its ego position and velocity in its local coordinate system. The position measurement drift of the odometry system can be modeled as a random walk of the robot’s local coordinate system.\(^{26}\) The velocity measurement error can be modeled as a zero mean white Gaussian noise.\(^{36}\) Hence, the robot observation model is expressed as follows:

\[
\dot{\hat{p}}_i = v_i + n_{p_i}, \tag{6}
\]

\[
\dot{\hat{v}}_i = v_i + n_{v_i}, \tag{7}
\]

where \( \hat{p}_i, \hat{v}_i \in \mathbb{R}^2 \) represent the self-observation position and velocity of the \( i \)th robot,

\[
n_{p_i} \sim \mathcal{N} (0, \sigma_p), \tag{8}
\]

\[
n_{v_i} \sim \mathcal{N} (0, \sigma_v) \tag{9}
\]

indicate two kinds of two-dimensional zero-mean white Gaussian noises and we have \( \sigma_p, \sigma_v > 0 \). Suppose that there is no position measurement drift in the beginning. Then, at the time \( t \), the relationship between the self-observation position
and the real position of the \( i \)th robot is shown as follows:

\[
\hat{\mathbf{p}}_i(t) = \mathbf{p}_i(t) + \mathbf{r}_p(t),
\]

where

\[
\mathbf{r}_p(t) \sim \mathcal{N}\left(0, \sigma_p \frac{L}{\Delta t}\right)
\]

and \( \Delta t \) is the observation sampling time. In the beginning, we have

\[
\mathbf{r}_p(0) = \mathbf{0}.
\]

**Remark 1.** According to the navigation devices equipped on robots, the self-observation position and velocity \( \mathbf{p}_i, \mathbf{v}_i \) have different relationships. If the self-observation position and velocity are obtained in different ways, then they are two independent observation variables, which have no mathematical relationship. Hence the relationship \( \dot{\mathbf{p}}_i = \mathbf{v}_i \) is not satisfied. If the self-observation position \( \mathbf{p}_i \) is obtained from the integral of the self-observation velocity \( \mathbf{v}_i \), we have \( \mathbf{n}_p = \mathbf{n}_v \) and \( \hat{\mathbf{p}}_i = \hat{\mathbf{v}}_i \). Our method proposed in this article is applicable to both of these cases.

Define a relative position \( \mathbf{p}_{m,ij} \) and a relative velocity \( \mathbf{v}_{m,ij} \) between the \( i \)th and \( j \)th robots, which are shown as follows:

\[
\hat{\mathbf{p}}_{m,ij} = \mathbf{p}_i - \mathbf{p}_j,  \tag{13}
\]

\[
\mathbf{v}_{m,ij} = \mathbf{v}_i - \mathbf{v}_j.  \tag{14}
\]

Besides, we have \( \hat{\mathbf{p}}_{m,ij} = \hat{\mathbf{v}}_{m,ij} \). For precise relative navigation, an assumption is proposed as follows.

**Assumption 1.** Any robot is able to obtain relative position information and relative velocity information precisely. The self-observation positions and velocities are not accurate. In other words, self-observation position information \( \hat{\mathbf{p}}_i \), self-observation velocity information \( \hat{\mathbf{v}}_i \), relative position information \( \mathbf{p}_{m,ij}, j \in \mathcal{N}_{m,i} \), and relative velocity information \( \mathbf{v}_{m,ij}, j \in \mathcal{N}_{m,i} \) are available to the \( i \)th robot.

**Remark 2.** Assumption 1 is a mathematical assumption to achieve precise relative navigation. Such an ideal assumption is just for the simplicity of the analysis in the following. In real practice, the relative navigation is not perfect, either. But the accuracy of the relative navigation between robots is, in general, far higher than the self-localization of any robot. Our method proposed in this article is applicable to the cases where relative positions and velocities can be measured more precisely than the self-observation positions and velocities.

### 2.2 Curve virtual tube model

In our previous work, \(^{14}\) a curve virtual tube is proposed for guiding a robotic swarm in a corridor-like environment. Here its definition and some necessary concepts are reviewed.

1. **Generating curve.** A generating curve \( \mathcal{C} \subset \mathbb{R}^2 \) locates inside the curve virtual tube. The generating curve ends at \( \mathbf{p}_f \in \mathcal{C} \).

   As shown in Figure 3, for any point \( \mathbf{p} \in \mathcal{C} \), define \( \mathbf{t}_c(\mathbf{p}) \in \mathbb{R}^2 \) to be the unit tangent vector pointing in the forward direction. Similarly, \( \mathbf{n}_c(\mathbf{p}) \in \mathbb{R}^2 \) is the unit normal vector directing anti-clockwise or left of the tangent direction. Then it has \( \mathbf{n}_c^T(\mathbf{p}) \mathbf{n}_c(\mathbf{p}) = 0 \).

2. **Cross section.** For any \( \mathbf{p} \in \mathcal{C} \), a cross section passing \( \mathbf{p} \) is defined as:

\[
\mathcal{C}(\mathbf{p}) = \{ \mathbf{x} \in \mathbb{R}^2 : \mathbf{x} = \mathbf{p} + \lambda(\mathbf{p}) \mathbf{n}_c(\mathbf{p}), \lambda(\mathbf{p}) \leq \lambda(\mathbf{p}) \leq \lambda(\mathbf{p}) \},
\]

where \( \lambda(\mathbf{p}) < 0 \) and \( \lambda(\mathbf{p}) > 0 \). Here, \( \mathcal{C}(\mathbf{p}_f) \) is called the finishing line or finishing cross section. Two endpoints of \( \mathcal{C}(\mathbf{p}) \) are defined as \( \mathbf{p}_i(\mathbf{p}) = \mathbf{p} + \lambda_1(\mathbf{p}) \mathbf{n}_c(\mathbf{p}) \) and \( \mathbf{p}_f(\mathbf{p}) = \mathbf{p} + \lambda_f(\mathbf{p}) \mathbf{n}_c(\mathbf{p}) \). The width of \( \mathcal{C}(\mathbf{p}) \) is expressed as \( 2r_1(\mathbf{p}) \), which is defined as \( r_1(\mathbf{p}) = \frac{1}{2} |\lambda_1(\mathbf{p}) - \lambda_f(\mathbf{p})| \). The middle point of \( \mathcal{C}(\mathbf{p}) \) is defined as \( \mathbf{m}(\mathbf{p}) = \frac{1}{2} (\mathbf{p}_i(\mathbf{p}) + \mathbf{p}_f(\mathbf{p})) \).
3. **Curve virtual tube.** The curve virtual tube $\mathcal{T}_V$ is generated by keeping cross sections always perpendicular to the tangent vectors of the generating curve, which is shown as follows:

$$\mathcal{T}_V = \bigcup_{p \in \mathcal{V}} C(p).$$  \hfill (16)

Then, the tube boundary $\partial \mathcal{T}_V$ is expressed as follows:

$$\partial \mathcal{T}_V = \{ x \in \mathbb{R}^2 : x = p_l(p) \cup p_r(p), p \in \mathcal{V} \},$$  \hfill (17)

which corresponds to two smooth boundary curves as shown in Figure 3.

Then, an assumption regarding the curve virtual tube is proposed.

**Assumption 2.** The curve virtual tube $\mathcal{T}_V$ has no self-intersection. The self-intersection means that given two different points $x, y \in \mathcal{V}, x \neq y$, we have $C(x) \cap C(y) \neq \emptyset$.

According to Assumption 2, any point inside the curve virtual tube corresponds to a specific cross section. In real practice, when the $i$th robot is inside the curve virtual tube, identifying its corresponding cross section is of the essence. Here we use the concept of the closest point to solve this problem.\(^{37}\) When $p_i \in \mathcal{T}_V$, the closest point of $p_i$ on the generating curve $\mathcal{V}$ is shown as follows:

$$p_i^* = \arg\min_{p \in \mathcal{V}} \|p - p_i\|^2.$$  \hfill (18)

According to Lemma 1 and Proposition 1 in Reference 37, the optimization problem in (18) has a unique optimal solution when Assumption 2 is satisfied. Besides, according to Lemma 2 in Reference 37, we have

$$t_i^T(p_i^*)(p_i^* - p_i) = 0,$$  \hfill (19)

which suggests that $p_i \in C(p_i^*)$. Hence, when the $i$th robot is inside the curve virtual, its corresponding cross section is $C(p_i^*)$. To simplify the symbolic representation and description, we let $C(p_i) = C(p_i^*), t_i(p_i) = t_i(p_i^*), r_i(p_i) = r_i(p_i^*), m_i(p_i) = m_i(p_i^*)$ in the following.

### 2.3 Distributed controller for passing through the curve virtual tube in the ideal condition

In Reference 14, a distributed vector field controller is proposed for guiding the robotic swarm to pass through the curve virtual tube in the ideal condition. There is no uncertainty in the self-observation positions and velocities. In this subsection, the controller is reviewed briefly. More descriptions can be found in Reference 14.
As shown in Figure 3, an Euclidean distance is defined between the \(i\)th robot and the tube boundary confined in \(C(\mathbf{p}_i)\), which is shown as follows:

\[
d_{i,t} = r_t(\mathbf{p}_i) - \left\| \mathbf{p}_i - \mathbf{m}(\mathbf{p}_i) \right\|.
\]

(20)

According to two smooth functions \(\delta(x, d_1, d_2)\) and \(s(x, e_s)\) presented in Appendix A, two Lyapunov-like barrier functions are designed as follows:

\[
V_{m,ij} = \frac{k_2 \left( 1 - \delta \left( \frac{\left\| \mathbf{p}_{m,ij} \right\|}{2r_s, r_s + r_a} \right) \right)}{(1 + e_m) \left\| \mathbf{p}_{m,ij} \right\| - 2r_s \delta \left( \frac{\left\| \mathbf{p}_{m,ij} \right\|}{2r_s, e_s} \right)},
\]

(21)

\[
V_{ij} = \frac{k_3 \left( 1 - \delta \left( d_{ij, r_a, r_s} \right) \right)}{(1 + e_i) d_{ij} - r_a s \left( \frac{d_a}{r_a, e_a} \right)},
\]

(22)

where \(k_2, k_3, e_m, e_i, e_s > 0\), and \(r_a > r_s\). Besides, it is required that \(r_d > r_c > r_s + r_a\). In our previous work,\(^{14}\) the constant \(r_a\) is called avoidance radius. The detailed properties of \(V_{m,ij}\) and \(V_{ij}\) are also presented in Reference 14. With barrier functions \(V_{m,ij}\), \(V_{ij}\) available, the controller for the \(i\)th robot is proposed as follows:

\[
v_{c,i} = v_{i} \left( \mathcal{T}_{v}, \mathbf{p}_i, \mathbf{p}_{m,ij} \right) = \text{sat} \left( \mathbf{u}_{1,i} + \mathbf{u}_{2,i} + \mathbf{u}_{3,i}, v_m \right),
\]

(23)

where \(v_m > 0\) is the maximum permitted speed of all robots, and the saturation function \(\text{sat}(x, v_m)\) is defined as follows:

\[
\text{sat}(x, v_m) = \begin{cases} x & \|x\| \leq v_m, \\ v_m & \|x\| > v_m. \end{cases}
\]

(24)

The subcommands \(\mathbf{u}_{1,i}, \mathbf{u}_{2,i}\), and \(\mathbf{u}_{3,i}\) are called as finishing line approaching term, robot avoidance term and virtual tube keeping term, which are shown as follows:

\[
\mathbf{u}_{1,i} = v_m \mathbf{t}_i(\mathbf{p}_i).
\]

(25)

\[
\mathbf{u}_{2,i} = -\sum_{j \in N_m, \partial} \frac{\partial V_{m,ij}}{\partial \mathbf{m}_{ij}} \mathbf{p}_{m,ij}.
\]

(26)

\[
\mathbf{u}_{3,i} = -\frac{\partial V_{ij}}{\partial d_{ij}} \left( \mathbf{I}_2 - \mathbf{t}_i(\mathbf{p}_i) \mathbf{t}_i^T(\mathbf{p}_i) \right) \left( \frac{\partial d_{ij}}{\partial \mathbf{p}_i} \right)^T.
\]

(27)

Inside the curve virtual tube, the controller (23) is always smooth and differentiable. The velocity command \(v_{c,i}\) and its derivative \(\mathbf{v}_{c,i}\) are both available. Outside the curve virtual tube, some discontinuities may exist because the \(i\)th robot may correspond to several different cross sections, which cause the discontinuity of \(\mathbf{t}_i(\mathbf{p}_i)\). Consider a scenario where a robot is moving within a curve virtual tube, in the middle of which we have another robot. Figure 4 shows two kinds of vector fields with the swarm controller (23). The vector field in Figure 4A is more aggressive, and the one in Figure 4(B) is softer. Each vector field corresponds to a different set of parameters. In Figure 4A, the parameters in (23) are \(v_m = 3, r_s = 0.15, r_a = 0.3, k_2 = 1, k_3 = 1, e_m = 10^{-6}, e_i = 10^{-6}\), and \(e_s = 10^{-6}\). In Figure 4B, the parameters in (23) are \(v_m = 3, r_s = 0.15, r_a = 0.3, k_2 = 1.5, k_3 = 0.01, e_m = 15, e_i = 1\), and \(e_s = 0.5\). As stated in Reference 14, the more aggressive vector field can guarantee the safety of the robotic swarm in theory. However, real robots may be unable to precisely track this aggressive vector field due to physical limitations. The softer vector field is more feasible in practical use.

With the descriptions above, some extra assumptions are presented.

**Assumption 3.** All robots have the same information of the curve virtual tube.

**Assumption 4.** The robots’ initial conditions satisfy \(S_i(0) \subset \mathcal{T}_v\), \(S_i(0) \cap S_j(0) = \emptyset\), where \(i, j = 1, \ldots, M, i \neq j\).
**Assumption 5.** Once a robot arrives at the finishing line \( C(p_f) \), it will quit this curve virtual tube so other robots behind are not affected anymore.

When the controller (23) is designed to be aggressive, a theorem is stated to show the system’s safety and stability.

**Theorem 1.** Under Assumptions 2-5, suppose that (i) all robots have a single-integrator model \( \dot{p}_i = v_{ci}, i = 1, \ldots, M \); (ii) the velocity command for the \( i \)-th robot is designed as (23); (iii) \( T_V \) is wide enough for at least one robot to pass through. Then, we have \( t_1 > 0 \) such that all robots can pass through \( T_V \) at \( t \geq t_1 \), meanwhile, guaranteeing \( S_i(t) \cap S_j(t) = \emptyset, S_i(t) \subseteq T_V, t \in [0, t_1), i, j = 1, 2, \ldots, M, i \neq j \).

In this article, the robot kinematics model (1) is second-order. The objective now is to establish a control law for \( a_{ci} \) to make the robot track \( v_{ci} \) in the ideal condition. With \( v_{ci} \) and its derivative \( \dot{v}_{ci} \) available, the acceleration command for the \( i \)-th robot is proposed as follows:

\[
a_{ci} = a(v_i, v_{ci}) = k_v(v_{ci} - v_i) + \dot{v}_{ci},
\]

where \( k_v > 0 \). Then a lemma is put forward as follows.

**Lemma 1.** When the vector field \( v_{ci} \) and its derivative \( \dot{v}_{ci} \) are available, we have \( \lim_{t \to \infty} v_i(t) = v_{ci} \) if the acceleration command is designed as (28).

### 2.4 Problem formulation

With descriptions and preliminaries above, the **robust curve virtual tube passing-through control problem** is stated as follows.

**Robust curve virtual tube passing-through control problem.** Under Assumptions 1-5, design the acceleration command \( a_{ci} \) to guide and control all robots to pass through the curve virtual tube \( T_V \), meanwhile, having tendency to avoid colliding with each other \( (S_i(t) \cap S_j(t) = \emptyset) \) and keep within the virtual tube \( (S_i(t) \subseteq T_V) \) by using cross information to enhance the localization of robots, where \( i, j = 1, \ldots, M, i \neq j, t > 0 \).

### 3 ROBUST SWARM CONTROLLER DESIGN AND ANALYSIS

#### 3.1 Robust Swarm controller design

When the self-localization drift and precise relative navigation are considered, namely, Assumption 1 is satisfied, the controllers (23), (28) are rewritten as follows:

\[
v_{ci} = v(T_V, \dot{p}_i, \dot{p}_{m,i}) = \text{sat} \left( \dot{u}_{1,i} + u_{2,i} + \dot{u}_{3,i} + v_m \right),
\]

**FIGURE 4** Two kinds of vector fields (23) of a curve virtual tube. Inside the curve virtual tube, both vector fields are always smooth and differentiable.
\[ \mathbf{a}_{c,i} = \mathbf{a} (\mathbf{\dot{v}}_i, \mathbf{v}_{c,i}) = k_v (\mathbf{v}_{c,i} - \mathbf{\dot{v}}_i) + \mathbf{v}_{c,i}, \tag{30} \]

where

\[ \mathbf{\hat{u}}_{1,i} = v_m t_c (\hat{p}_i), \tag{31} \]

\[ \mathbf{\hat{u}}_{3,i} = -\frac{\partial V_{a,i}}{\partial d_{ij}} (I_2 - t_c (\hat{p}_i) t_c^T (\hat{p}_i)) \left( \frac{\partial d_{ij}}{\partial \hat{p}_i} \right)^T. \tag{32} \]

Position measurement drifts and velocity measurement errors have negative impacts on all robots. Robots may have collisions with each other and move outside the curve virtual tube. To solve this problem, the flocking behaviors of the Boids model \(^{13}\) are introduced into the modified vector field controller, which is shown as follows:

\[ \mathbf{v}_{c,i} = \mathbf{v}_{mdf} \left( r_c, \hat{p}_i, \mathbf{p}_{m,i} \right) = \text{sat} \left( \mathbf{\hat{u}}_{1,i} + \mathbf{u}_{2,i} + \mathbf{\hat{u}}_{3,i} + \mathbf{u}_{4,i} + \mathbf{u}_{5,i}, v_m \right), \tag{33} \]

where the robot cohesion term \( \mathbf{u}_{4,i} \) and the velocity alignment term \( \mathbf{u}_{5,i} \) are shown as follows:

\[ \mathbf{u}_{4,i} = -k_4 \sum_{j \in N_{m,i}} \frac{\partial V_{a,ij}}{\partial \mathbf{p}_{m,j}} \hat{p}_{m,j}, \tag{34} \]

\[ \mathbf{u}_{5,i} = -k_5 \sum_{j \in N_{m,i}} \mathbf{v}_{m,j}. \tag{35} \]

where \( k_4 > 0, k_5 > 0 \). The attractive potential function \( V_{a,ij} \) in \( \mathbf{u}_{4,i} \) will be introduced in (36).

### 3.2 Analysis: Why robot cohesion term can reduce negative impact of position measurement drift

#### 3.2.1 Attractive potential field function for robot cohesion term

According to the definition of the cohesion area \( C_i \), an attractive potential field function \( V_{a,ij} \) is defined as follows:

\[ V_{a,ij} = \delta \left( \| \mathbf{p}_{m,ij} \| , r_c, r_d \right), \tag{36} \]

where the function \( \delta (x, d_1, d_2) \) is defined in (A1). When \( \| \mathbf{p}_{m,ij} \| < r_d \), the attractive forces of the \( i \)-th and \( j \)-th robots begin to appear. When \( \| \mathbf{p}_{m,ij} \| < r_c \), the attractive forces disappear. Besides, when \( \| \mathbf{p}_{m,ij} \| = (r_c + r_d) / 2 \), the attractive forces reach the maximum. When \( r_c = 2, r_d = 4 \), the function \( V_{a,ij} (x) = \delta (x, 2, 4) \) and its negative derivative \( -\partial V_{a,ij} (x) / \partial x \) are shown in Figure 5.
Remark 3. The introduction of the cohesion term has no influence on the system’s stability and safety. As $V_{a,i}$ is limited, this function is not a “barrier” function and different from $V_{m,ij}, V_{L,i}$. Correspondingly, $u_{4,i}$ in (33) is always bounded. Hence the cohesion behavior can be seen as a soft constraint. On the contrary, collision avoidance and keeping within the virtual tube are two hard constraints. The detailed descriptions about soft and hard potential field functions can be found in Reference. 38 When the $i$th robot has safety risks, $u_{2,i}$ or $\hat{u}_{3,i}$ in (33) will become unbounded. Besides, when all the distances between any pair of robots are less than $r_c$ or larger than $r_d$, the controller (33) will degenerate into (29) with $u_{5,i}$ unconsidered. In this case, the controller (33) is still satisfied with Theorem 1 in the ideal condition.

3.2.2 Effect analysis of robot cohesion term

In the following, we are going to discuss why the robot cohesion term $u_{4,i}$ in (33) can reduce the negative impact of the position measurement drift. As shown in Figure 6(A), at the time $t = 0$, the $i$th robot locates at $p_i(0)$ and begins to pass through $\mathcal{T}_V$. Then, at the time $t = t_1$, its real position is $\hat{p}_i(t_1)$ and its measured position is $\hat{p}_i(t_1)$. To simplify the description, here the curve virtual tube $\mathcal{T}_V$ is seen as a function with respect to the time $t$. As $\hat{p}_i(t_1)$ instead of $p_i(t_1)$ is used in the controllers (29) and (33), the $i$th robot is affected by the location relationship between $\hat{p}_i(t_1)$ and $\mathcal{T}_V$, which can be considered equivalent to the $i$th robot moving inside its corresponding drifting curve virtual tube $\mathcal{T}_V \left( r_p(t_1) \right)$. At the time $t$, the drifting curve virtual tube of the $i$th robot is shown as follows:

$$\mathcal{T}_V \left( r_p(t) \right) = \{ x \in \mathbb{R}^2 : x = p - r_p(t), p \in \mathcal{T}_V \}.$$  \hspace{1cm} (37)

According to (12), when $t = 0$, we have

$$\mathcal{T}_V \left( r_p(0) \right) = \mathcal{T}_V \left( 0 \right) = \mathcal{T}_V.$$  \hspace{1cm} (38)

The definition of the drifting curve virtual tube actually transfers the position measurement drift from the robot to the curve virtual tube. There is no drift in the robot’s position and the robot is affected by $\mathcal{T}_V \left( r_p(t) \right)$. As the mean value of $r_p(t)$ is a zero vector, $\mathcal{T}_V \left( r_p(t) \right)$ of all robots are most likely distributed around $\mathcal{T}_V$. As stated in Remark 3, the introduction of $u_{4,i}$ has no influence on the system’s stability and safety. With position measurement drifts, robots keep moving inside their corresponding $\mathcal{T}_V \left( r_p(t) \right)$, which results in the fact that all robots are most likely distributed around $\mathcal{T}_V$. Meanwhile, $u_{4,i}$ in (33) makes robots move toward each other. With relative properties of the consensus algorithm, $u_{4,i}$ directs to $\mathcal{T}_V$ with a high probability, which increases with the number of robots inside $D_i$. Figure 6(B) shows an example with three robots, whose $\mathcal{T}_V \left( r_p(t) \right)$ are evenly distributed around $\mathcal{T}_V$. It can easily be observed that $u_{4,i}, i = 1, 2, 3$ all point to $\mathcal{T}_V$. In a word, the robot cohesion term $u_{4,i}$ in (33) is able to make robots have more tendencies to keep within $\mathcal{T}_V$. 

**Figure 6** (A) The relationship between the position measurement drift and the drifting curve virtual tube. (B) An example of three robots having tendencies to move towards the curve virtual tube due to the existence of $u_{4,i}$ in (33).
Remark 4. Although $u_{4,i}$ in (33) can reduce the negative impact of the position measurement drift, the main cost is the reduction of moving efficiency. All robots have to spend more time passing through the curve virtual tube. The cohesion term makes robots move towards each other. And $u_{4,i}$ of the robot in the front of the robotic swarm points most likely to the opposite direction of $u_{1,i}$. In other words, robots in the front have less force to move forward in $T_v$ due to the existence of $u_{4,i}$. As a result, although robots in the rear have more force to move forward with the help of $u_{4,i}$ from robots in the front, robots in the rear are blocked by other robots ahead. Therefore, in the modified vector field controller (33), when $k_4$ in $u_{4,i}$ becomes larger, all robots have more tendency to move inside the curve virtual tube with less moving efficiency. To support the above conclusion, a pair of comparative simulations in the ideal condition is presented in Section 4.2.1.

3.3 Analysis: Why velocity alignment term can reduce negative impact of velocity measurement error?

3.3.1 Modified acceleration command containing a velocity alignment term

In (30), an acceleration command is designed to make the robot track its velocity command. According to (1), the equation (30) is equivalent to

$$\dot{v}_i = k_v \left( v_{c,i} - v_i - n_v \right) + \dot{v}_{c,i}. \quad (39)$$

When the saturation function in the modified vector field controller (33) is not active

$$\| \hat{u}_{1,i} + u_{2,i} + \hat{u}_{3,i} + u_{4,i} + u_{5,i} \| \leq v_m, \quad (40)$$

the velocity alignment term $u_{5,i}$ in (33) can directly be transferred to the acceleration command. Hence, to facilitate the analysis, a modified acceleration command containing only $u_{5,i}$ is shown as follows:

$$a_{c,i} = a_{mdf} \left( \hat{v}_i, v_{c,i} \right) = k_v \left( v_{c,i} - \hat{v}_i + u_{5,i} \right) + \dot{v}_{c,i}. \quad (41)$$

The control gain $k_v > 0$ is shared among the equations (28), (30), (39), and (41). It should be noted that the introduction of (41) is just for comparison with (30). Similar to (39), the equation (41) is equivalent to

$$\dot{v}_i = k_v \left( v_{c,i} - v_i - n_v - k_5 \sum_{j \in N_{m,i}} \hat{v}_{m,j} \right) + \dot{v}_{c,i}. \quad (42)$$

In the following, we will compare the noise component in the velocities of (39) and (42).

3.3.2 Effect analysis of velocity alignment term in the modified acceleration command

In real practice, the curvature change of the generating curve $\Sigma$ is quite small in most cases. Besides, compared with $r_d$, the length of $\Sigma$ is much larger. Hence, $u_{1,i}$ in (33) of all robots in a detection area are almost the same. As we only discuss the relationship between velocity measurement errors and velocities of robots with respect to each other, it is assumed that, without loss of generality, all robots have the same zero velocity command, which is summarized as an assumption as follows.

Assumption 6. There are $N$ robots having a relative localization relationship. All robots have the same zero velocity command $v_{c,i} = 0$, which also results in $\dot{v}_{c,i} = 0, i = 1, 2, \ldots, N$.

Then an important lemma is stated as follows.

Lemma 2. Under Assumption 6, the velocity $v_i$ of the $i$th robot in (39) and the one in (42) both contain a two-dimensional zero-mean white Gaussian noise component, which is expressed as $n_v \sim \mathcal{N} (0, \sigma_v^2)$ and $n_v' \sim \mathcal{N} (0, \sigma_v'^2)$.
\( N \ (0, \sigma''_v) \), respectively. Then we have

\[
\begin{align*}
\sigma'_v &= \frac{k_v}{2} \sigma_v, \\
\sigma''_v &= \frac{k_v}{2} \frac{k_5 + 1}{k_5 N + 1} \sigma_v.
\end{align*}
\]

(43) \hspace{1cm} (44)

Proof. See Appendix B. ▪

Compared with the original controller (30), the modified acceleration command (41) containing a velocity alignment term reduces the variance of the noise in the velocity. The variance ratio relates to \( N \) and \( k_5 \), which is shown as follows:

\[
\frac{\sigma''_v}{\sigma'_v} = \frac{k_5 + 1}{k_5 N + 1}.
\]

(45)

For any robot in the robotic swarm, it has \( N = 1 \) when there is no other robot in its detection area. Otherwise, we have \( N > 1 \), which causes \( \frac{\sigma''_v}{\sigma'_v} < 1 \). Given \( N > 1 \), the larger \( k_5 \) is, the less \( \frac{\sigma''_v}{\sigma'_v} \) will be. In other words, when \( k_5 \) becomes larger, the variance of noise in the velocity becomes smaller, and all robots have more tendency to move inside the curve virtual tube and avoid collision with each other. However, in practical scenarios, a very high \( k_5 \) may lead to instability and safety risks of the robotic swarm. Hence, the value of \( k_5 \) should be tuned carefully with simulations and experiments.

In the following, we will present a further discussion. Without considering the area limitation of the detection area, we have

\[
\lim_{N \to \infty} \sigma''_v = \lim_{N \to \infty} \frac{k_v}{2} \frac{k_5 + 1}{k_5 N + 1} \sigma_v = 0
\]

(46)

according to (44). Equation (46) means that when \( N \) approaches infinity, the variance of noise in the velocity of each robot becomes zero. In this case, the noise in the velocity disappears by using relative velocity information among robots. Similarly, without considering the limitation of the robots’ safety, we can obtain that

\[
\lim_{k_5 \to \infty} \frac{\sigma''_v}{\sigma'_v} = \lim_{k_5 \to \infty} \frac{k_v}{2} \frac{k_5 + 1}{k_5 N + 1} = \frac{k_v}{2N} \sigma_v,
\]

(47)

\[
\lim_{k_5 \to \infty} \frac{\sigma'_v}{\sigma'_v} = \lim_{k_5 \to \infty} \frac{k_5 + 1}{k_5 N + 1} = \frac{1}{N}.
\]

(48)

Different from (46), the equation (47) shows that when \( k_5 \) approaches infinity, some noises still exist in the velocities of all robots. And the variance of noise in the velocity of each robot becomes \( 1/N \) times the original variance according to (48). Given a limited \( N \), noises in the velocities will not disappear no matter how large \( k_5 \) is.

### 3.4 Analysis: Control effectiveness of robust Swarm controller

With Lemmas 1-2 and the above analyses available, the main result of this paper is stated as follows.

**Theorem 2.** Under Assumptions 2–5, suppose that (i) all robots have a double-integrator model (1); (ii) the controllers for the ith robot is designed as (30), (33); (iii) \( T_V \) is wide enough for at least one robot to pass through. Then, in the ideal condition, we have \( t_1 > 0 \) such that all robots can pass through \( T_V \) at \( t \geq t_1 \), meanwhile, guaranteeing \( S_i(t) \cap S_j(t) = \emptyset \), \( S_i(t) \subset T_V, \ t \in [0, t_1), i, j = 1, 2, \ldots, M, i \neq j \). When Assumption 1 is satisfied, the modified controllers (30), (33) decrease the variance of noise in the velocities of all robots compared with original controllers (29), (30). Besides, these modified controllers induce a stronger inclination for the robotic swarm to move inside the curve virtual tube and avoid collision with each other.

Proof. In the ideal condition, \( u_{4,i} \) in (33) has no influence on the system stability and safety as stated in Remark 3. Similarly, \( u_{5,i} \) in (33) also has no influence. The reason is that \( u_{5,i} \) can be equivalent to the derivative of a limited potential function, which is easy to add to the final Lyapunov-like function. Then, according
to Theorem 1 and Lemma 1, we have \( t_1 > 0 \) such that all robots can pass through \( T_v \) at \( t \geq t_1 \), meanwhile, guaranteeing \( S_i(t) \cap S_j(t) = \emptyset \). \( S_i(t) \subseteq T_v \), \( t \in [0, t_1) \), \( i, j = 1, 2, \ldots, M \), \( i \neq j \).

Next consider the condition that Assumption 1 is satisfied. When the saturation function in (33) is not active, the controllers (30), (33) can make the robotic swarm have more tendency to move inside the curve virtual tube and avoid collision with each other according to Lemma 2. When the saturation function in (33) works, the norms of five control terms in (33) scale proportionally. The control effect of the robot cohesion and velocity alignment may become weaker. But modified controllers can still induce a stronger inclination for the robotic swarm to move inside the curve virtual tube and avoid collision with each other.

Remark 5. Compared with Theorem 1, the proposed Theorem 2 is more qualitative instead of quantitative. Complex swarm behaviors make it difficult for us to identify the quantitative control effect of our proposed controller (33). However, according to our analyses, the controller (33) statistically reduces the negative impact of the self-localization drift. Hence we use the description as “induce a stronger inclination for the robotic swarm.”

### 4 SIMULATION AND EXPERIMENT

Simulations and experiments are given to show the effectiveness of the proposed method. A video is available on [https://youtu.be/TpOWXf9six8](https://youtu.be/TpOWXf9six8) and [http://rfly.buaa.edu.cn](http://rfly.buaa.edu.cn).

#### 4.1 Comparative numerical simulations

In this subsection, the validity and feasibility of the proposed method are numerically verified in comparative simulations under the self-localization drift and precise relative navigation. Consider a scenario in which \( M = 6 \) robots pass through a predefined curve virtual tube \( T_v \). The width of \( T_v \) is always \( r_1 = 1 \text{m} \) along the generating curve. The curve length of the generating curve is 24.77\text{m}. All robots are arranged symmetrically in a rectangular space in the beginning and they satisfy the second-order model in (1). The original controllers (29), (30) and modified controllers (30), (33) are applied to guide these robots. The control parameters are \( k_2 = 1, k_3 = 1, \epsilon_m = 10^{-6}, \epsilon_t = 10^{-6}, \epsilon_s = 10^{-6}, k_4 = 2, k_5 = 1, r_s = 0.2 \text{m}, r_a = 0.3 \text{m}, r_c = 1 \text{m}, r_d = 2 \text{m}, \) and \( v_m = 1 \text{m/s} \). The variances of noises \( n_p \) and \( n_q \) are set as \( \sigma_p = 1 \text{m}^2, \sigma_v = 1(\text{m/s})^2 \). Besides, in order to make simulations closer to the practice, the simulation step of the robot model is 0.001 s, while the simulation steps of the controller and the noise generator are both 0.02 s.

Both simulations last 33 s. During the simulation, the position measurement drifts of all robots are shown in Figure 7A. Assume that the velocity measurement error of the \( i \)-th robot is shown as \( n_{v_i} = [n_{v_{ix}}, n_{v_{iy}}]^T \), where \( n_{v_{ix}}, n_{v_{iy}} \sim N(0, \sigma_v) \). Here we choose \( \sum n_{v_{ix}} \) as a representative velocity measurement error, which is shown in Figure 7B. As shown in Figure 8, three snapshots of each simulation are presented, and the safety areas of all robots are represented by circles with different colors. It can be observed that with original controllers (29) and (30), robots often move outside \( T_v \) during the simulation due to the existence of the self-localization drift. By comparison, robots usually keep moving inside \( T_v \) with modified controllers (30) and (33).

For a better comparison of the two kinds of controllers, a sum of distances \( d_{t,\text{all}} \) is introduced as the comparison metric, which is defined as:

\[
d_{t,\text{all}} = \sum_{i=1}^{M} \max(r_s - d_{t,i}, 0),
\]

where \( d_{t,i} \) is defined in (20). It has \( d_{t,\text{all}} \geq 0 \). A larger \( d_{t,\text{all}} \) means that the phenomenon of robots moving outside \( T_v \) is more common. In Figure 9, a comparison of the metric \( d_{t,\text{all}} \) is presented. The modified controllers proposed in this paper perform better and have the ability to reduce the negative impact of the self-localization drift. Besides, we also record the sum of \( d_{t,\text{all}} \) during the simulation and the passing-through time of all robots. With original controllers (29), (30), it takes 30.54 s for all robots to pass through the curve virtual tube, and the sum of \( d_{t,\text{all}} \) is 8.188 m. With modified controllers (30), (33), it takes 40.98 s for all robots to pass through, and the sum of \( d_{t,\text{all}} \) is 0.119 m.

To get a comparison conclusion with statistical significance, such comparative numerical simulations are repeated 30 times with different noises and the same parameters. Then, it is obtained that compared with original controllers (29),
(30), the average of the sum of $d_{i,\text{all}}$ during the simulation decreases by 66.5% with modified controllers (30), (33). The average passing-through time of all robots increases by 43%. This comparison result confirms that our proposed method not only has the ability to reduce the negative impact of the self-localization drift, but also reduces the moving efficiency of the robotic swarm. In real practice, to compensate for the decrease in moving efficiency, increasing the maximum permitted speed $v_m$ is a feasible option as long as physical limitations are not exceeded.

4.2 Additional numerical simulations for robot cohesion term

As stated in Section 3.2.2 and Remark 5, the discussion about the robot cohesion term is more qualitative instead of quantitative in this paper. Hence, in this subsection, some additional numerical simulations are presented to show the control effect of the robot cohesion term.

4.2.1 Comparative simulations to show reduction of moving efficiency

In Remark 4, we have stated that the main cost of the robot cohesion term $u_{4,i}$ in (33) is the reduction of moving efficiency. Here, a pair of comparative simulations in the ideal condition is presented to support our statement. In the simulations, there is no uncertainty. Consider a scenario in which $M = 4$ robots pass through a virtual tube $T_V$. The generating curve is a line segment. The width of $T_V$ is always $r_i = 1$ m. All robots satisfy the second-order model in (1). The original controllers (29), (30) and modified controllers (30), (33) are applied to guide these robots. As we only focus on the effect of the robot cohesion term $u_{4,i}$, the velocity alignment term $u_{5,j}$ in (33) is unconsidered and set as $u_{5,j} = 0$. The control parameters are $k_2 = 1, k_3 = 1, \epsilon_m = 10^{-6}, \epsilon_t = 10^{-6}, \epsilon_s = 10^{-6}, k_4 = 2, r_s = 0.2$ m, $r_a = 0.3$ m, $r_c = 0.6$ m, $r_d = 2$ m, and $v_m = 1$ m/s. Both simulations last 4 seconds. As shown in Figure 10, original controllers (29), (30) without cohesion make robots move farther than modified controllers (30) and (33) with cohesion. In other words, the robot cohesion term $u_{4,i}$ makes robots have less moving efficiency.

4.2.2 Comparative simulations under external perturbations

In this paper, we assume that the mean value of the noise $n_p$, always equals a zero vector, which is shown in (8). However, in real practice, robots may be subject to some external perturbations, such as wind, wheel slip, and electromagnetic
interference. In these cases, the mean value of $n_p$ should be set as a constant vector, whose norm is not equal to zero. In the following, a pair of comparative simulations is presented to show the effectiveness of the robot cohesion term under external perturbations.

The simulation scenario is similar to the one presented in Section 4.2.1. The difference is that the width of $T_F$ is always $r_i = 0.6m$, and the noise $n_p$ is set as $n_p \sim N\left(\begin{bmatrix} 0.01 & 0.01 \end{bmatrix}^T m/s, 1(m/s)^2\right)$. The original controllers (29), (30) and modified controllers (30), (33) are applied to guide these robots. As we only focus on the effect of the robot cohesion term $u_{4,i}$, the noise in the velocity and the velocity alignment term $u_{5,i}$ in (33) are both unconsidered. Both simulations last 25 seconds. As shown in Figure 11, although robots cohere more closely with the help of the robot cohesion term $u_{4,i}$ in (33), the modified controllers (30) and (33) cannot make robots have more tendency to stay inside the virtual tube. In a word, the

**FIGURE 8** Snapshots of comparative simulations.
4.2.3 Simulations to show effectiveness of robot cohesion term in flocking control

The cohesion behavior can reduce the negative impact of the position measurement drift not only inside the curve virtual tube but also in many common scenarios. In the following, we will show that the cohesion behavior has a similar control effect in the flocking control.
In Reference 19, the authors propose a leader–follower flocking system. Here, some simulations are carried out, which are similar to the first simulation presented in Reference 19. The leader–follower flocking system is composed of $M = 4$ “leader” robots, which satisfy the single integrator model $p_i = v_{c,i}, i = 1, \ldots, 4$. The robots are commanded to track a desired trajectory, which is shown as follows:

$$p_d(t) = \begin{bmatrix} 22 \cos(0.038t) + 5 \\ 22 \sin(0.038t) \end{bmatrix} \text{m.}$$

In the simulations, the velocity command of the $i$th robot is designed as follows:

$$v_{c,i} = \text{sat}\left( k_t \left( p_d - \hat{p}_i \right) + p_d + u_{2,i} + k_c \left( p^c - p_i \right) \cdot v_m \right),$$

where $k_t, k_c > 0$, the robot avoidance term $u_{2,i}$ is defined in (26), and $p^c = \sum_{i=1}^{M} p_i / M$ represents the position of flocking center. In Reference 19, the authors propose a practical method for all robots to estimate the flocking center with their local information. In our simulations, we assume that the flocking center $p^c$ is available to all robots. According to Assumption 1, a two-dimensional zero-mean white Gaussian noise $n_p \sim \mathcal{N}(0, 1\text{m}^2)$ exists in the self-observation position $\hat{p}_i$. Besides, as robots have precise relative navigation, the accurate position $p_i$ is used in the cohesion term $k_c \left( p^c - p_i \right)$, which has a similar cohesion effect to our proposed $u_{2,i}$ in (33). Compared with the controller proposed in Reference 19, the controller in (51) has two modifications. The first is that we replace the fuzzy separation function proposed in Reference 19 with $u_{2,i}$ to achieve collision avoidance among robots. Such a modification simplifies the description and does not affect the control effect. The second is that a saturation function is added to the controller, which only changes the norm of the velocity command without affecting the direction. As long as $v_m > \|p_d\|$, this saturation process has no negative effect on the system safety and stability.\textsuperscript{14}

Four simulations are carried out with $k_c = 0, 1, 5, 10$. Other parameters are set as $k_t = 5, r_s = 1\text{m}, r_a = 3\text{m}, \epsilon_m = 10^{-6}, \epsilon_s = 10^{-6}$, and $v_m = 2\text{m/s}$. When $k_c = 0$, robots just track the desired trajectory and avoid collision with each other. The larger $k_c$ is, the more tendency the robots will have to cohere. The simulation result of $k_c > 0$ in the ideal condition can be
**FIGURE 12**  Flocking control simulations.

**FIGURE 13**  The comparison of the sum of trajectory distances $d_{tra,all}$ in the flocking control simulations.
FIGURE 14 Snapshots of real experiments.

FIGURE 15 The comparison of the sum of distances $d_{\text{tra,all}}$ from the tube boundary of all robots in the experiments.

found in Reference 19. When there is no uncertainty, robots track the desired trajectory as a connected group during the whole process. They will not leave the desired trajectory far away. When position measurement drifts exist, the flocking trajectories with $k_c = 0, 1, 5, 10$ are shown in Figure 12. The dotted line in black denotes the desired trajectory in (50). The colored dotted lines are the trajectories of four robots. The black circle on the desired trajectory represents the desired position at four time instants $t = 0s, 40s, 80s, 120s$. It can be observed that when $k_c$ becomes larger, robots have more tendency to get closer and move near the desired trajectory. For a better comparison, a sum of trajectory distances $d_{\text{tra,all}}$
similar to $d_{\text{all}}$ in (49) is introduced as the comparison metric, which is defined as follows:

$$d_{\text{tra,all}}(t) = \sum_{i=1}^{M} \| p_d(t) - p_i(t) \|. \quad (52)$$

A larger $d_{\text{tra,all}}$ means that the phenomenon of robots moving far away from the desired trajectory is more common. In Figure 13, a comparison of the metric $d_{\text{tra,all}}$ is proposed. When $k_c$ becomes larger, the metric $d_{\text{tra,all}}$ becomes smaller finally. Besides, we can observe that there is no much difference with the curves of $k_c = 5, 10$ in Figure 13, which suggests that simply increasing $k_c$ is meaningless. The cohesion behavior has a limited effect on reducing the negative impact of position measurement drift.

In Remark 4, we have stated that the main cost of the robot cohesion term is the reduction of moving efficiency inside the curve virtual tube. However, as shown in Figure 12, the moving efficiency almost keeps the same whatever $k_c$ is. The reason is that Equation (33) generates a path controller, while Equation (51) generates a trajectory controller. The finishing line approaching term $u_{l,i}$ in (25) has an invariable norm $\| u_{l,i} \| = v_m$. In (51), the norm of the tracking term $k_t (p_d - \hat{p}_i)$ becomes larger when the robot is farther away from the desired trajectory. Hence, although robots in the front have less force to move forward due to the existence of the cohesion term $k_c (p^i - \hat{p}_i)$, the tracking term $k_t (p_d - \hat{p}_i)$ will become larger and compensate for the reduction of the moving efficiency.

### 4.3 Real experiments

In this subsection, real experiments are implemented on an open-source platform called Robotarium, which is developed by Georgia Institute of Technology.\textsuperscript{40,41} By uploading our algorithms into the Web front end, the control algorithms are verified by the user management and transferred to the experiment server. In the experiments, $M = 10$ ground mobile robots are commanded to move in a predefined curve virtual tube. Here the curve virtual tube is closed. The generating curve starts and ends at the same point. The width of $\mathcal{T}_v$ is always $r_t = 0.25m$ along the generating curve. The original controllers (29), (30) and modified controllers (30), (33) are applied to guide these robots, respectively. The control parameters are $k_2 = 1, k_3 = 1, \epsilon_m = 10^{-6}, \epsilon_t = 10^{-6}, \epsilon_s = 10^{-6}, k_4 = 2, k_5 = 1, r_s = 0.075m, r_b = 0.125m, r_c = 0.3m, r_d = 1m$, and $v_m = 0.1m/s$. From Robotarium, we can get precise position information of all robots. And the velocity information can be obtained from the difference between the current and previous positions. To simulate the precise relative navigation, relative position information and relative velocity information are directly obtained. Then, to simulate self-localization drift, random walks are are added to the positions of all robots, and white Gaussian noises are added to the velocities. The variances of noises $\mathbf{n}_p$ and $\mathbf{n}_v$ are set as $\sigma_p = 10^{-3} \text{m}^2$ and $\sigma_v = 10^{-3} \text{(m/s)}^2$.

Both experiments last 70 seconds. As shown in Figure 14, three snapshots of each experiment are presented. It can be observed that compared with original controllers (29) and (30), robots have more tendencies to keep moving inside $\mathcal{T}_v$ with modified controllers (30) and (33). In Figure 15, the comparison of the metric $d_{\text{all}}$ is presented. The modified controllers proposed in this paper perform better and have the ability to reduce the negative impact of the self-localization drift.

### 5 CONCLUSIONS

The robust curve virtual tube passing-through control problem is proposed and then solved in this paper. To reduce the negative impact of the self-localization drift, the flocking behaviors of the Boids model are introduced to the controller design. A modified vector field controller is proposed based on our previous work. Formal analyses and proofs are made to show that the cohesion behavior and the velocity alignment behavior have the ability to reduce the influence of the position measurement drift and the velocity measurement error, respectively. Finally, comparative simulations and real experiments are given to show the effectiveness and performance of the proposed method. In the future, we are going to study how to solve the problem of other types of uncertainties inside the curve virtual tube, such as broadcast delay and packet loss when wireless communication is required for robots to share their poses.

### DATA AVAILABILITY STATEMENT

The data that support the findings of this study are available from the corresponding author upon reasonable request.
NOMENCLATURE

- $a_i$: acceleration command,
- $v_{c,i}$: velocity command (vector field),
- $v_i$: velocity and position of the $i$th robot.
- $\hat{p}_i$, $\dot{v}_i$: self-observation position and velocity of the $i$th robot. They are assumed as inaccurate.
- $n_{p_i}$, $n_{v_i}$: two kinds of two-dimensional zero-mean white Gaussian noises related to $\hat{p}_i$ and $\dot{v}_i$.
- $\sigma_{p_i}$, $\sigma_{v_i}$: variances of $n_{p_i}$, $n_{v_i}$.
- $r_{p_i}$: two-dimensional zero-mean random walk in $\hat{p}_i$.
- $\dot{p}_{m,j}$, $\dot{v}_{m,j}$: relative position and relative velocity between the $i$th and $j$th robots. They are precise with no noise.
- $v_m$: maximum permitted speed of all robots.
- $r_s$, $r_a$, $r_d$, $r_c$: safety radius, avoidance radius, detection radius and cohesion lower bound of all robots.
- $S_i$, $D_i$, $C_i$: safety area, detection area and cohesion area of the $i$th robot.
- $N_{m,i}$: collection of all IDs of other robots within the detection area of the $i$th robot.
- $V$, $T_Y$, $\partial T_Y$: generating curve, curve virtual tube and its boundary.
- $C(p)$: cross-sectional passing the point $p$.
- $r_i(p)$, $m(p)$: width and middle point of the cross section $C(p)$.
- $t_c(p)$: unit tangent vector pointing in the forward direction of $p$.
- $n_c(p)$: unit normal vector directing anti-clockwise or left of $t_c(p)$.
- $d_{i,j}$: euclidean distance between the $i$th robot and the tube boundary confined in $C(p)$.
- $V_{m,j}$: lyapunov-like barrier function for the $i$th and $j$th robots to avoid collision with each other.
- $V_{t,i}$: lyapunov-like barrier function for the $i$th robot to keep within the curve virtual tube.
- $V_{a,i}$: attractive potential function for the $i$th and $j$th robots.
- $u_{1,i}$, $u_{2,i}$, $u_{3,i}$: finishing line approaching term, robot avoidance term, virtual tube keeping term of the $i$th robot.
- $\hat{u}_{1,i}$, $\hat{u}_{3,i}$: finishing line approaching term and virtual tube keeping term of the $i$th robot subject to self-localization drift.
- $u_{4,i}$: robot cohesion term and velocity alignment term of the $i$th robot.
- $k_2$, $\epsilon_m$, $\epsilon_s$: parameters in $V_{m,j}$ and $u_{2,j}$.
- $k_3$, $\epsilon_r$, $\epsilon_s$: parameters in $V_{t,i}$, $u_{3,i}$ and $\hat{u}_{3,i}$.
- $k_4$: parameter in $V_{a,i}$ and $u_{4,i}$.
- $k_5$: parameter in $u_{5,i}$.
- $T_Y(r_{p_i}(t_1))$: drifting curve virtual tube of the $i$th robot at the time $t_1$.

CONFLICT OF INTEREST STATEMENT

The authors declare that there is no conflict of interests regarding the publication of this paper.

ORCID

- Yan Gao 🌐 https://orcid.org/0000-0001-5112-0561
- Chenggang Bai 🌐 https://orcid.org/0000-0003-2224-2337
- Quan Quan 🌐 https://orcid.org/0000-0001-8216-8998

REFERENCES

1. Hu J, Bhowmick P, Lanzon A. Two-layer distributed formation-containment control strategy for linear swarm systems: algorithm and experiments. *Int J Robust Nonlinear Control*. 2020;30(16):6433-6453.
2. Yang L, Yu J, Yang S, Wang B, Nelson BJ, Zhang L. A survey on swarm microrobotics. *IEEE Trans Robot*. 2022;38(3):1531-1551.
3. Dong Y, Huang J. Leader-following consensus with connectivity preservation of uncertain Euler–Lagrange multi-agent systems. *Int J Robust Nonlinear Control*. 2017;27(18):4772-4787.
4. Park J, Kim D, Kim GC, Oh D, Kim HJ. Online distributed trajectory planning for quadrotor swarm with feasibility guarantee using linear safe corridor. *IEEE Robot Autom Lett*. 2022;7(2):4869-4876.
5. Zhao S, Zelazo D. Bearing rigidity theory and its applications for control and estimation of network systems: life beyond distance rigidity. *IEEE Control Syst Mag*. 2019;39(2):66-83.
6. Xu Y, Zhao S, Luo D, You Y. Affine formation maneuver control of high-order multi-agent systems over directed networks. *Automatica*. 2020;118:109004.
7. Zhou X, Zhu J, Zhou H, Xu C, Gao F. Ego-swarm: a fully autonomous and decentralized quadrotor swarm system in cluttered environments. *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE; 2021:4101-4107.
8. Ding W, Gao W, Wang K, Shen S. An efficient b-spline-based kinodynamic replanning framework for quadrotors. *IEEE Trans Robot.* 2019;35(6):1287-1306.

9. Panagou D, Stipanović DM, Voulgaris PG. Distributed coordination control for multi-robot networks using Lyapunov-like barrier functions. *IEEE Trans Autom Control.* 2015;61(3):617-632.

10. Panagou D. Motion planning and collision avoidance using navigation vector fields. *2014 IEEE International Conference on Robotics and Automation (ICRA).* IEEE; 2014:2513-2518.

11. Wang L, Ames AD, Egerstedt M. Safety barrier certificates for collisions-free multirobot systems. *IEEE Trans Robot.* 2017;33(3):661-674.

12. Yao W, Marina dHG, Sun Z, Cao M. Distributed coordinated path following using guiding vector fields. *2021 IEEE International Conference on Robotics and Automation (ICRA).* IEEE; 2021:10030-10037.

13. Reynolds CW. Flocks, herds and schools: a distributed behavioral model. *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques.* ACM; 1987:25-34.

14. Quan Q, Gao Y, Bai C. Distributed control for a robotics swarm to pass through a curve virtual tube. *Robot Autonom Syst.* 2023;162:104368.

15. Luo Y, Cai P, Bera A, Hsu D, Lee WS, Manocha D. Porca: modeling and planning for autonomous driving among many pedestrians. *IEEE Robot Autom Lett.* 2018;3(4):3418-3425.

16. Tony LA, Ratnoo A, Ghose D. CORRIDRONE: corridors for drones, an adaptive on-demand multi-lane design and testbed. arXiv: 2012.01019. 2020.

17. Michael N, Kumar V. Controlling shapes of ensembles of robots of finite size with nonholonomic constraints. *Proc Robot.* 2008.

18. Fernando M, Liu L. Formation control and navigation of a quadrotor swarm. *2019 International Conference on Unmanned Aircraft Systems (ICUAS).* IEEE; 2019:284-291.

19. Gu D, Wang Z. Leader–follower flocking: algorithms and experiments. *IEEE Trans Control Syst Technol.* 2009;17(5):1211-1219.

20. Rezende AM, Gonçalves VM, Raffo GV, Pimenta LC. Robust fixed-wing UAV guidance with circulating artificial vector fields. *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).* IEEE; 2018:5892-5899.

21. Rezende AM, Gonçalves VM, Nunes AH, Pimenta LC. Robust quadcopter control with artificial vector fields. *2020 IEEE International Conference on Robotics and Automation (ICRA).* IEEE; 2020:6381-6387.

22. Olavo JLG, Thums GD, Jesus TA, De Araujo Pimenta LC, Torres LAB, Palhares RM. Robust guidance strategy for target circulation by controlled UAV. *IEEE Trans Aerosp Electron Syst.* 2018;54(3):1415-1431.

23. Johnson B, Albrecht C, Braman T, et al. Development of a navigation-grade MEMS IMU. *IEEE International Symposium on Inertial Sensors and Systems (INERTIAL).* IEEE; 2021: 1-4.

24. Mohamed SA, Haghbayan MH, Westerlund T, Heikkonen J, Tenhunen H, Plosila J. A survey on odometry for autonomous navigation systems. *IEEE Access.* 2019;7:97466-97486.

25. Qin T, Li P, Shen S. Vins-mono: a robust and versatile monocular visual-inertial state estimator. *IEEE Trans Robot.* 2018;34(4):1004-1020.

26. Nguyen TH, Nguyen TM, Xie L. Flexible and resource-efficient multi-robot collaborative visual-inertial-range localization. *IEEE Robot Autom Lett.* 2021;7(2):928-935.

27. Han D, Panagou D. Robust multitask formation control via parametric Lyapunov-like barrier functions. *IEEE Trans Autom Control.* 2019;64(11):4439-4453.

28. Zhang P, Chen G, Li Y, Dong W. Agile formation control of drone flocking enhanced with active vision-based relative localization. *IEEE Robot Autom Lett.* 2022;7(3):5359-6366.

29. Yan X, Deng H, Quan Q. Active infrared coded target design and pose estimation for multiple objects. *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).* IEEE; 2019:6885-6890.

30. Jayasuriya M, Ranasinghe R, Dissanayake G. Active perception for outdoor localisation with an omnidirectional camera. *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).* IEEE; 2020:4567-4574.

31. Pavliv M, Schiano F, Reardon C, Floreano D, Loianno G. Tracking and relative localization of drone swarms with a vision-based headset. *IEEE Robot Autom Lett.* 2022;7(2):1455-1462.

32. Bergman G, Donner KO. An analysis of the spring migration of the common scoter and the long-tailed duck in southern Finland. *Tilgmann; 1964.*

33. Simons AM. Many wrongs: the advantage of group navigation. *Trends Ecol Evol.* 2004;19(9):453-455.

34. Kremer I, Mansour Y, Perry M. Implementing the “wisdom of the crowd”. *J Polit Econ.* 2014;122(3):988-1012.

35. Júnior GPC, Rezende AM, Miranda VR, et al. EKF-LOAM: an adaptive fusion of LiDAR SLAM with wheel odometry and inertial data for confined spaces with few geometric features. *IEEE Trans Autom Sci Eng.* 2022;19(3):1458-1471.

36. Delaune J, Bayard DS, Brockers R. Range-visual-inertial odometry: scale observability without excitation. *IEEE Robot Autom Lett.* 2021;6(2):2421-2428.

37. Rezende AM, Gonçalves VM, Pimenta LC. Constructive time-varying vector fields for robot navigation. *IEEE Trans Robot.* 2021;38(2):852-867.

38. Santilli M, Mukherjee P, Williams RK, Gasparri A. Multirobot field of view control with adaptive decentralization. *IEEE Trans Robot.* 2022;38(4):2131-2150.

39. Olfati-Saber R. Flocking for multi-agent dynamic systems: algorithms and theory. *IEEE Trans Autom Control.* 2006;51(3):401-420.

40. Pickem D, Glotfelter P, Wang L, et al. The robotarium: a remotely accessible swarm robotics research testbed. *2017 IEEE International Conference on Robotics and Automation (ICRA).* IEEE; 2017:1699-1706.

41. Wilson S, Glotfelter P, Wang L, et al. The robotarium: globally impactful opportunities, challenges, and lessons learned in remote-access, distributed control of multirobot systems. *IEEE Control Syst Mag.* 2020;40(1):26-44.
APPENDIX A. INTRODUCTION OF TWO SMOOTH FUNCTIONS

Two smooth functions are introduced for the design of Lyapunov-like barrier functions. The first function is a smooth “bump” function

\[
\delta(x, d_1, d_2) = \begin{cases} 
0 & x \leq d_1 \\
Ax^3 + Bx^2 + Cx + D & d_1 \leq x \leq d_2 \\
1 & d_2 \leq x
\end{cases}
\] (A1)

where \(d_1\) and \(d_2\) are two parameters, \(A = \frac{2}{(d_1 - d_2)^3}\), \(B = -\frac{3(d_1 + d_2)}{(d_1 - d_2)^3}\), \(C = \frac{6d_1d_2}{(d_1 - d_2)^3}\), \(D = \frac{d_2^3(d_1 - 3d_2)}{(d_1 - d_2)^3}\). When \(d_1 = 2, d_2 = 4\), the function \(\delta(x, 2, 4)\) is shown in the left plot of Figure 5. As shown in Figure A1, to approximate a saturation function \(\bar{s}(x) = \min(x, 1), x \geq 0\), the other smooth function is a smooth saturation function

\[
s(x, \varepsilon_s) = \begin{cases} 
x & 0 \leq x \leq x_1(\varepsilon_s) \\
(1 - \varepsilon_s) + \sqrt{\varepsilon_s^2 - (x - x_2(\varepsilon_s))^2} & x_1(\varepsilon_s) \leq x \leq x_2(\varepsilon_s) \\
1 & x_2(\varepsilon_s) \leq x
\end{cases}
\] (A2)

with \(x_2(\varepsilon_s) = 1 + \frac{1}{\tan 67.5^\circ} \varepsilon_s\) and \(x_1(\varepsilon_s) = x_2(\varepsilon_s) - \sin 45^\circ \varepsilon_s\).
APPENDIX B. PROOF OF LEMMA 2

Firstly, for the equation (39), the input and output are defined as $-\mathbf{n}_{v_j}$ and $\mathbf{v}_i$, respectively. According to (9), we have

$$-\mathbf{n}_{v_j} \sim \mathcal{N}(0, \sigma_{v_j}).$$  \hfill (B1)

The noises $\mathbf{n}_{v_j}$ and $-\mathbf{n}_{v_j}$ have the same power spectral density of each dimension $s_{n_{v_j}}(\omega)$. As we have $\mathbf{v}_{c,i} = \mathbf{0}$ and $\mathbf{v}_{c,j} = \mathbf{0}$, a transfer function matrix $\mathbf{G}(s)$ is obtained as:

$$\mathbf{G}(s) = \mathbf{G}(s) \mathbf{I}_2 = \frac{k_v}{s + k_v} \mathbf{I}_2.$$  \hfill (B2)

As $-\mathbf{n}_{v_j}$ is a two-dimensional white Gaussian noise, the power spectral density of each dimension $s_{n_{v_j}}(\omega)$ is always equal to its variance, which is shown as follows:

$$s_{n_{v_j}}(\omega) = \sigma_v.$$  \hfill (B3)

Besides, as $\mathbf{G}(s)$ is linear and time-invariant, $\mathbf{v}_i$ in (39) contains a two-dimensional zero-mean white Gaussian noise component. Then, it has

$$\sigma_{v_i}^2 = \frac{1}{2\pi} \int_{-\infty}^{+\infty} |\mathbf{G}(j\omega)|^2 s_{n_{v_j}}(\omega) d\omega = \frac{\sigma_v}{2\pi} \int_{-\infty}^{+\infty} \frac{k_v^2}{\omega^2 + k_v^2} d\omega = \frac{k_v^2}{2} \sigma_v.$$  \hfill (B4)

Secondly, for the equation (42), the input is defined as $\mathbf{u} = [-\mathbf{n}_{v_1}^\top - \mathbf{n}_{v_2}^\top \cdots - \mathbf{n}_{v_N}^\top]^\top$, and the output is defined as $\mathbf{y} = [\mathbf{v}_1^\top \mathbf{v}_2^\top \cdots \mathbf{v}_N^\top]^\top$. The relative localization relationship among $N$ robots is modeled as a simple connected undirected graph. Then, the diagonal elements of the Laplacian matrix $\mathbf{L} \in \mathbb{R}^{N \times N}$ are all $-1$, and other elements are all $0$. Then according to (42), the state equations are expressed as $\dot{\mathbf{v}} = \mathbf{A} \mathbf{v} + \mathbf{B} \mathbf{u}, \mathbf{y} = \mathbf{C} \mathbf{v}$, where $\mathbf{v} = [\mathbf{v}_1^\top \mathbf{v}_2^\top \cdots \mathbf{v}_N^\top]^\top$, $\mathbf{A} = -k_v k_5 \left( \mathbf{L} \otimes \mathbf{I}_2 + \frac{1}{k_5} \mathbf{I}_{2N} \right)$, $\mathbf{B} = k_v \mathbf{I}_{2N}, \mathbf{C} = \mathbf{I}_{2N}$. The transfer function matrix $\mathbf{H}(s)$ is shown as follows:

$$\mathbf{H}(s) = \begin{bmatrix}
H_1(s) \mathbf{I}_2 & H_2(s) \mathbf{I}_2 & \cdots & H_2(s) \mathbf{I}_2 \\
H_2(s) \mathbf{I}_2 & H_1(s) \mathbf{I}_2 & \cdots & H_1(s) \mathbf{I}_2 \\
\vdots & \vdots & \ddots & \vdots \\
H_2(s) \mathbf{I}_2 & H_2(s) \mathbf{I}_2 & \cdots & H_1(s) \mathbf{I}_2
\end{bmatrix}. \hfill (B5)

where

$$H_1(s) = \frac{k_v (s + k_v + k_5)}{(s + k_v) (s + k_v + N k_v k_5)}.$$  \hfill (B6)

$$H_2(s) = \frac{k_5^2 k_5}{(s + k_v) (s + k_v + N k_v k_5)}.$$  \hfill (B7)

Hence, for the $i$th robot, we have

$$\mathbf{v}_i(s) = -H_1(s) \mathbf{n}_{v_i} - \sum_{j=1, j \neq i}^{N} H_2(s) \mathbf{n}_{v_j}. \hfill (B8)$$

As $\mathbf{H}(s)$ is linear and time-invariant, $\mathbf{v}_i$ in (42) contains a two-dimensional zero-mean white Gaussian noise component. Then, $\sigma_{\mathbf{v}_i}''$ is calculated as:

$$\sigma_{\mathbf{v}_i}'' = \frac{1}{2\pi} \int_{-\infty}^{+\infty} |H_1(j\omega)|^2 s_{n_{v_i}}(\omega) d\omega + \sum_{j=1, j \neq i}^{N} \frac{1}{2\pi} \int_{-\infty}^{+\infty} |H_2(j\omega)|^2 s_{n_{v_i}}(\omega) d\omega. \hfill (B9)$$
Then it has

\[
\sigma'' = \frac{\sigma_v}{2\pi} \left[ \int_{-\infty}^{+\infty} |H_1(j\omega)|^2 d\omega + \frac{(N-1)\sigma_v}{2\pi} \int_{-\infty}^{+\infty} |H_2(j\omega)|^2 d\omega \right]
\]

\[
= \frac{\sigma_v k_v (N-1)(k_5N + k_5 + 2) + (k_5 + 2)(k_5N + 1)}{2N(k_5N + 1)(k_5N + 2)} + \frac{(N-1)\sigma_v k_v}{2(k_5N + 1)(k_5N + 2)}
\]

\[
= \frac{k_v}{2} \frac{k_5 + 1}{k_5N + 1} \sigma_v.
\]  (B10)