A Novel Virtual-Structure Formation Control Design for Mobile Robots with Obstacle Avoidance

Xuanlin Chen¹, Fanghao Huang¹(✉), Yougong Zhang¹, Zheng Chen¹,³, Zheng Chen¹,²,³, Shuo Liu¹,³,⁎, Yong Nie², Jianzhong Tang² and Shiqiang Zhu¹,⁴

¹ Ocean College, Zhejiang University, Zhoushan 316021, China; 3160100578@zju.edu.cn (X.C.); huangfanghao@zju.edu.cn (F.H.); zhangyougong@zju.edu.cn (Y.Z.); zheng_chen@zju.edu.cn (Z.C.); sqzhu@zju.edu.cn (S.Z.)
² The State Key Laboratory of Fluid Power and Mechatronic Systems, Zhejiang University, Hangzhou 310027, China; ynie@zju.edu.cn (Y.N.); jztang@zju.edu.cn (J.T.)
³ Key Laboratory of Ocean Observation-Imaging Testbed of Zhejiang Province, Zhejiang University, Zhoushan 316021, China
⁴ Zhejiang Lab, Hangzhou 311100, China

* Correspondence: shuoliu@zju.edu.cn

Received: 22 June 2020; Accepted: 18 August 2020; Published: 21 August 2020

Abstract: The cooperative motion of multiple mobile robots has attracted wide attention due to its advantages in military, marine and aerospace fields, and formation control has become a significant technology in the realization of these tasks. However, most of the existing formation control designs of mobile robots do not consider the practical obstacles in the environment, and the maintenance of both formation and trajectory tracking while confronting the obstacles is still a challenging issue. Therefore, in this paper, a virtual-structure-based formation control approach is designed with obstacle avoidance for a system with multiple mobile robots. The basic trajectory is generated for each robot in the group and parameterized to keep the group in formation. A trajectory generator is then established regarding the obstacles, where a potential function is designed to adjust the basic trajectory and replan the reference trajectory to achieve obstacle avoidance. Then, a novel design for the path parameter is proposed to improve the performance of the robot group when encountering obstacles. Finally, a tracking controller is designed to achieve good tracking performance for robots, and the guaranteed performance is achieved via the Lyapunov theorem. A comparative simulation with three sets is carried out, where an objective function \(F_{obj}\) is designed to evaluate the tracking performance in the presence of obstacles. Besides this, a real experiment is implemented to further verify the effectiveness. The simulation and experimental results verify the good formation and tracking performance of the proposed design for a system with multiple mobile robots with obstacle avoidance.

Keywords: formation control; obstacle avoidance; mobile robot; virtual-structure approach

1. Introduction

With the development of automation [1–5], mobile robots have been widely applied due to their advantages in many fields, such as in environmental inspection, area reconnaissance, maritime search and rescue, etc. [6,7]. However, reliance solely on a single robot cannot handle some complicated tasks; thus, the concept of groups of multiple robots has come into being [8–10]. Formation control is a significant technology for multiple robots [11–15] that can increase the system’s robustness and efficiency, expand the range of exploration and reduce the system cost. Therefore, formation control for robots has received increasing attention over recent years.
The main objective of formation control for groups of multiple robots is to track a predefined trajectory while maintaining a certain geometric pattern for the robots [16–18]. However, with regard to the practical environment, which is always complicated by the presence of obstacles [19], the aforementioned formation objective becomes a challenge. Therefore, collision avoidance is an inevitable but significant issue in the formation control of multiple mobile robots, which can improve the flexibility of robots when undertaking more complicated tasks in the real environment. Formation control can be roughly classed as three approaches [20–23], which are leader-following, behavioral, and virtual structures, respectively; a detailed review is presented below.

The leader-following approach is the most widely used approach due to its good operability and ability to keep the formation even if the leader is perturbed by disturbances. This approach designates the leader and followers of the group. The leader tracks the predefined trajectory while the followers track the trajectory determined by the state of the leader. A leader-following formation control method for underactuated surface vehicles with model uncertainties and environmental disturbances was proposed in [24]. Gui et al. [25] proposed the leader-following approach with the combination of Lyapunov-based sliding-mode control design in the formation of multiple robots. F. Fahimi et al. [13] proposed sliding-mode control laws to control multiple robots in arbitrary formations, where the stability was guaranteed by the combination of local information and planned formation motion. L. Lapierre et al. [26] designed the leader-following approach to steer robots along identical parallel paths with the least information exchange. Schoerling et al. [27] design the cooperative controllers for robots. A. Loria et al. [28] design the controller based on the assumption that only one agent knows the information of reference trajectory. Ref. [29] proposed a leader-following approach considering the constraints of the control input. Chen et al. [30] designed a directed formation control law based on the knowledge only of the neighbor’s direction by using the integrator backstepping technique. Li et al. [31] proposed leader-following controller considering the time-varying delay and nonlinear dynamics. These studies exhibit good formation tracking performance for robots; however, since no feedback is provided by the follower, the leader-following approach cannot maintain the formation well when the follower is perturbed. This approach cannot achieve good formation tracking performance when encountering obstacles. Moreover, few applications of the behavioral approach are based on the formation control of robots due to the limited sensing range and difficulty of stability analysis [32], which makes it difficult for the robots to track the reference trajectories precisely.

In contrast to the leader-following and behavioral approaches, the entire formation is predefined as a single entity in the virtual-structure approach since it has a rigid geometric relationship. When the virtual structure moves, it produces the reference trajectory for the tracking of each robot. In [33], a virtual-structure control strategy with mutual coupling between the robots is proposed. In [16], a combination of the virtual structure and path-tracking approaches is used to derive the formation architecture, and the controller is designed in such a way that the path derivative is left as a free input to synchronize the robots’ motion. Do et al. [34] presented a constructive method to design output–feedback cooperative controllers that drive a group of robots with limited sensing ranges to perform desired formation tracking. Furthermore, in [17], this formation control method was applied to robots with limited sensing ranges to perform the desired formation. Yu et al. [35] proposed a novel controller for the trajectory tracking of nonholonomic vehicles, which could guarantee a bounded velocity. Generally, a virtual structure can maintain the formation well for multiple robots and track the predefined trajectories precisely, which is applied in this paper to design the formation controller for multiple robots.

However, the obstacles in the environment are not considered in the aforementioned references, which is an inevitable but significant issue for the practical formation control of robots to achieve the obstacle avoidance. A consensus-based cooperative formation control strategy with collision-avoidance capability for a group of multiple unmanned aerial vehicles (UAVs) was proposed in [36]. A control law based on the potential function and a formation control protocol based on graph theoretic considerations were proposed in [20,21], respectively. In [37], a control law with online
obstacle avoidance was discussed with online optimization to maintain the formation automatically. K. Do et al. [22] developed distributed controllers that could force a group of robots to perform a desired formation with no collision among any of the robots in the group. In [38], a formation strategy with the combination of the leader-following formation approach and potential function-based method was proposed.

Although some research works have been undertaken to consider collision avoidance [38–41], few can be applied directly to robots, and keeping the tracking error as small as possible in the formation when encountering the obstacles is still challenging. Thus, considering the obstacles in the environment, a virtual-structure-based formation control design for groups of multiple mobile robots is proposed in this paper. The virtual-structure approach is applied to generate the basic trajectory for each robot in the group, and the trajectory is parameterized to keep this group in formation. A trajectory generator is established with the consideration of obstacles, where the potential function is designed to adjust the basic trajectories and generate new reference velocities, which can be used to replan the reference trajectories. Subsequently, a tracking controller is designed to achieve good tracking performance, and the guaranteed performance is achieved via the Lyapunov theorem. A comparative simulation is carried out to show the formation and tracking regarding obstacles. Besides this, a real robot experiment is implemented to further verify the effectiveness of the proposed control design. The main contributions are as follows:

(A) The potential function method is applied to the virtual-structure-based approach for the consideration of obstacles; thus, the algorithm can adapt to the practical environment.

(B) A novel path parameter which parameterizes the basic trajectories is designed to adjust the weight of the formation and tracking in the moving process according to the situation that the formation encounters obstacles. The system can give priority to tracking rather than formation when encountering obstacles, meaning that the tracking errors of the team can be as small as possible.

The remainder of this paper is organized as follows. In Section 2, the problem statement is given. In Section 3, the details of our formation controller are presented. Section 4, a simulation is implemented to prove the effectiveness of the proposed controller. In Section 5, a real robot experiment is carried out to further show the tracking and obstacle avoidance performance. Section 6 describes the contributions of this paper and future research.

2. Problem Statement

All symbols used in this paper are shown in Nomenclature. Consider a group of robots, shown in Figure 1, where the kinematic model of each robot can be defined as follows:

\[
\begin{align*}
\dot{x} &= v \cos \theta \\
\dot{y} &= v \sin \theta \\
\dot{\theta} &= \omega
\end{align*}
\]

where \(x\) and \(y\) are the global coordination of gravity center for each robot, \(\theta\) is the heading and \(v\) and \(\omega\) are the linear velocity and angular velocity, which are regarded as the control inputs of each robot.

Due to the nonholonomic constraint of robots, the trajectory \(R(t) = \begin{bmatrix} x(t) & y(t) \end{bmatrix}^T\) should satisfy the following equation:

\[
-\dot{x} \sin \theta + \dot{y} \cos \theta = 0
\]

where \(\theta\) is the heading angle and can be defined as \(\theta = \arctan(y/x)\).

To achieve the formation control of mobile robots, \(N\) desired trajectories \(\begin{bmatrix} x_{di} & y_{di} & \theta_{di} \end{bmatrix}^T\), \((i = 1, 2, \ldots, N)\) are planned for the tracking of the group in the formation. Specifically, since collision
an inevitable issue in formation control, especially in a complex environment, a potential function is designed in this paper to achieve collision avoidance for the group.

Figure 1. The kinematic model of a mobile robot.

Therefore, based on the above statement, the objective of this paper can be described as follows: consider the obstacles in the environment and design a formation control input \( \begin{bmatrix} \nu_i \\ \omega_i \end{bmatrix} \) to guarantee the stability of the closed-loop system and make the tracking errors as small as possible.

3. Formation Controller Design

The proposed virtual-structure-based formation control scheme is shown in Figure 2.

Figure 2. Formation control scheme.

For a group of \( N \) robots, the virtual-structure approach is applied to generate \( N \) basic desired trajectories \( \begin{bmatrix} x_{di} \\ y_{di} \end{bmatrix}^T \) for these robots to track, and the trajectories are parameterized by \( s \) in order to keep the group in formation. The desired velocities \( \begin{bmatrix} \nu_{di} \\ \omega_{di} \end{bmatrix} \) can be calculated based on the
desired trajectories containing the desired velocities and trajectories. To achieve collision avoidance via the potential function, a trajectory generator is designed to replan the reference trajectories $\begin{bmatrix} x_{ri} & y_{ri} \end{bmatrix}^T$ with the desired velocities $\begin{bmatrix} v_{di} & \omega_{di} \end{bmatrix}^T$ and adjusted velocities $\begin{bmatrix} v_{ai} & \omega_{ai} \end{bmatrix}^T$. Based on the reference velocities and trajectories, the tracking controller $\begin{bmatrix} v_i & \omega_i \end{bmatrix}^T$ is designed to achieve good formation tracking performance. Moreover, the parameter $s$ is designed to adjust the desired velocities to track with regard to the tracking errors of the whole formation, where the desired formation can be slowed down to wait for the robots to track when the system has large tracking errors. Therefore, good formation and tracking performance regarding the obstacles can be achieved by the proposed control design.

3.1. Basic Trajectory Planning

In this subsection, the basic desired trajectories $\begin{bmatrix} x_{di} & y_{di} \end{bmatrix}^T$ and velocities $\begin{bmatrix} v_{di} & \omega_{di} \end{bmatrix}^T$ are planned.

A group of $N$ robots needs $N$ desired trajectories to stay in formation. The virtual-structure approach is applied here to generate the desired trajectories. Consider a virtual structure whose center moves along the desired trajectories $R_{d0}(s) = \begin{bmatrix} x_{d0}(s) & y_{d0}(s) \end{bmatrix}^T$ with $s$ being the path parameter. Then, trajectories $R_{di}(s) = \begin{bmatrix} x_{di}(s) & y_{di}(s) \end{bmatrix}^T$ can be specified by the displacement from the center of the structure to each place-holder (shown in Figure 3), which can be calculated as follows:

$$R_{di}(s) = R_{d0}(s) + M(\theta_{d0}(s))L_{di},$$

where

$$L_{di} = \begin{bmatrix} l_{xi} & l_{yi} \end{bmatrix}^T,$$

$$M(\theta_{d0}(s)) = \begin{bmatrix} \cos \theta_{d0}(s) & -\sin \theta_{d0}(s) \\ \sin \theta_{d0}(s) & \cos \theta_{d0}(s) \end{bmatrix}.$$

$\theta_{d0}$ is the heading angle of the center of structure and can be defined as

$$\theta_{d0}(s) = \arctan((\partial y_{d0}(s)/\partial s)/(\partial x_{d0}(s)/\partial s)).$$

Each trajectory is parameterized by the path parameter $s$ so that the desired trajectories can always be in formation.

Then, the desired linear velocity and angular velocity for each robot can be calculated as

$$v_{di} = \sqrt{\frac{\partial x_{di}(s)}{\partial s}^2 + \frac{\partial y_{di}(s)}{\partial s}^2},$$

$$\omega_{di} = \frac{\frac{\partial x_{di}(s)}{\partial s}}{\frac{\partial x_{di}(s)}{\partial s}^2 + \frac{\partial y_{di}(s)}{\partial s}^2} \hat{s} + \frac{\partial^2 x_{di}(s)/\partial s^2 - \partial^2 y_{di}(s)/\partial s^2}{\partial x_{di}(s)/\partial s},$$

where the time derivative of path parameter $s$ is included in the desired linear and angular velocity, which indicates that $\hat{s}$ determines the desired formation speed; the selection of $s$ is discussed in Section 3.4.
3.2. Collision Avoidance

In this subsection, the component $\begin{bmatrix} v_{ai} & \omega_{ai} \end{bmatrix}^T$ is designed via the potential function method to adjust the basic trajectories dynamically with regard to obstacles in the environment. For easy implementation, the shape of obstacles is simplified as being round, and a single robot is also regarded as a round obstacle for other robots. The detailed design is given below. Define the potential function candidate as

$$\Gamma_i = \begin{cases} 0, & d \leq r \\ \left( \frac{d^2 - R^2}{d^2 - r^2} \right)^2, & r < d \leq R \\ 0, & d > R \end{cases}$$

(5)

where $d = \sqrt{(x_i - x_p)^2 + (y_i - y_p)^2}$ is the distance from the $i$th robot to the obstacle, and the coordinates are $(x_i, y_i)$ and $(x_p, y_p)$. $r$ is the radius of the round obstacle and $R$ is the radius of the detection region around the obstacle; note that $R > r > 0$.

Thus, this function will blow up when the robot is close to the obstacle and will be equal to zero whenever the robot is outside the detection region. It is noteworthy that there is no need to consider situation $d \leq r$ because collision will occur before that time.

Take the partial derivatives of $\Gamma_i$ with respect to $x_i$ and $y_i$ coordinates as

$$\frac{\partial \Gamma_i}{\partial x_i} = \begin{cases} \frac{4(R^2-r^2)(d^2-R^2)}{(d^2-r^2)^3} (x_i - x_p), & r < d \leq R \\ 0, & \text{otherwise} \end{cases}$$

(6)

$$\frac{\partial \Gamma_i}{\partial y_i} = \begin{cases} \frac{4(R^2-r^2)(d^2-R^2)}{(d^2-r^2)^3} (y_i - y_p), & r < d \leq R \\ 0, & \text{otherwise} \end{cases}$$

(7)

and define the virtual force vector $\vec{f}_{pi} = -k_{pi} \begin{bmatrix} \frac{\partial \Gamma_i}{\partial x_i} \\ \frac{\partial \Gamma_i}{\partial y_i} \end{bmatrix}^T$.

Transform this virtual force vector to the robot’s body coordination as

$$\begin{bmatrix} \Psi_{vi} \\ \Psi_{wi} \end{bmatrix} = \begin{bmatrix} \cos \theta_i & \sin \theta_i \\ -\sin \theta_i & \cos \theta_i \end{bmatrix} \vec{f}_{pi}$$

(8)
Then, the adjusted component can be obtained as
\[
\begin{align*}
\nu_{ai} & = -c_{3i} \Psi_{vi} \\
\omega_{ai} & = -c_{4i} \Psi_{wi}
\end{align*}
\]  
(9)

where \(c_{3i}\) and \(c_{4i}\) are positive constants.

In this design, the virtual force \(\Psi_{vi}\) directly acts on the reference angular velocity, and the action direction is opposite to the obstacle (see Figure 4), which means that the generator tends to turn in the direction without obstacles; correspondingly, new reference trajectories will deviate from the desired path to avoid obstacles. Another force \(\Psi_{wi}\) acts on the linear velocities, and the forward speed tends to decrease to adjust the direction when the obstacle is in front. By applying this design, the new trajectories can be guaranteed to gradually move away from the location of the obstacle.

**Figure 4.** Trajectory adjustment for collision avoidance via the potential function method.

3.3. Reference Trajectory Replanning

The reference velocities can be obtained as
\[
\begin{align*}
\nu_{ri} & = \nu_{di} + \nu_{ai} \\
& = \sqrt{\left(\frac{\partial x_{di}(s)}{ds}\right)^2 + \left(\frac{\partial y_{di}(s)}{ds}\right)^2} \dot{s} - c_{3i} \Psi_{vi} \\
\omega_{ri} & = \omega_{di} + \omega_{ai} \\
& = \frac{\partial x_{di}(s)\frac{\partial^2 y_{di}(s)}{ds^2} - \partial y_{di}(s)\frac{\partial^2 x_{di}(s)}{ds^2}}{\partial x_{di}(s)^2 + \partial y_{di}(s)^2} \dot{s} - c_{4i} \Psi_{wi}
\end{align*}
\]  
(10)

It should be noted that the adjusted component \(\begin{bmatrix} \Psi_{vi} & \Psi_{wi} \end{bmatrix}^T\) is equal to 0 when the robot is outside the detection region.

Then, define a switch coefficient
\[
\eta = \begin{cases} 
1, & r \leq d \leq R \\
0, & R < d
\end{cases}
\]  
(11)
Therefore, the reference trajectories can be replanned as

\[ R_{ri}(t) = (1 - \eta)R_{di}(t) + \eta R_{ai}(t) \]  

(12)

where \( R_{di}(t) \) are basic trajectories and \( R_{ai}(t) \) are adjusted trajectories.

The adjusted trajectories \( R_{ai}(t) \) are generated by the reference velocities when the robot gets close to the obstacle. It can be obtained as

\[ R_{ai}(t) = \begin{bmatrix} x_{ai}(t) \\ y_{ai}(t) \end{bmatrix} = \begin{bmatrix} \int_{t_0}^{t} v_{ri}(\tau) \cos \theta_{ai}(\tau) d\tau + x_{ri}(t_0) \\ \int_{t_0}^{t} v_{ri}(\tau) \sin \theta_{ai}(\tau) d\tau + y_{ri}(t_0) \end{bmatrix} \]  

(13)

where \( \theta_{ai}(t) = \int_{t_0}^{t} \omega_{ri}(\tau) d\tau + \theta_{ri}(t_0) \) and \( t_0 \) is the moment when \( \eta \) switches from 0 to 1.

**Remark 1.** When the \( i \)th robot is out of the detection region—i.e., no collision occurs—the reference trajectories \( \begin{bmatrix} x_{ri} \\ y_{ri} \end{bmatrix}^T \) are equal to the desired \( \begin{bmatrix} x_{di} \\ y_{di} \end{bmatrix}^T \). However, when the \( i \)th robot is inside the detection region and the potential function \( \Gamma_i \) blows up, the reference trajectories will have already been adjusted to deviate from the desired ones to avoid the collision. In this case, the \( \Gamma_i \) can asymptotically converge to 0 if the robots track the reference trajectories.

### 3.4. The Selection of the Path Parameter

As described in Section 3.1, \( \dot{s} \) is the time derivative of the path parameter for all desired trajectories. Equation (4) implies that \( \dot{s} \) determines the desired linear and angular velocity.

For a single mobile robot trajectory tracking problem, \( s = t \) is chosen intuitively. However, in the formation control problem, for the selection of \( s \), two situations need to be considered to be follows:

(P1) \( s \) should reflect the overall formation tracking status.

Thus, for a group of \( N \) mobile robots, define

\[ V = \sum_{i=1}^{N} \left( \frac{1}{2} \xi_{xi}^2 + \frac{1}{2} \xi_{yi}^2 + (1 - \cos \frac{\xi_{\theta i}}{2}) \right) \]  

(14)

where \( \xi_{xi}, \xi_{yi} \) and \( \xi_{\theta i} \) are tracking errors, which can be defined as

\[ \begin{bmatrix} \xi_{xi} \\ \xi_{yi} \\ \xi_{\theta i} \end{bmatrix} = \begin{bmatrix} \cos \theta_i & \sin \theta_i & 0 \\ -\sin \theta_i & \cos \theta_i & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{di} - x_i \\ y_{di} - y_i \\ \theta_{di} - \theta_i \end{bmatrix} \]  

(15)

where \( \begin{bmatrix} x_{di} \\ y_{di} \\ \theta_{di} \end{bmatrix}^T \) are basic desired trajectories. Therefore, \( V \) can reflect the formation tracking errors well and can blow up whenever any robot deviates from the desired trajectories and converge to 0 whenever all the robots are on the track.

From (4), the time derivative of path parameter \( \dot{s} \) determines the desired formation velocities. Thus, \( \dot{s} \) is expected to decrease to slow down the velocities of desired trajectories whenever \( V \) is large, which means that some robots are out of track. Then, there is plenty of time left for these robots to reach their desired trajectories. On the other hand, \( \dot{s} \) is expected to converge to 1 whenever all the robots on the track, meaning that the formation velocities can reach a desired value. In conclusion:

\[ \dot{s} \rightarrow 0, \text{when} V \rightarrow \infty, \]

\[ \dot{s} \rightarrow 1, \text{when} V \rightarrow 0, \]  

(16)
Thus, $\dot{s}$ can be designed as follows [16]:

$$
\dot{s} = 1 - \alpha \arctan V,
$$

where $\alpha \in [0, 1]$; this choice is discussed later.

(P2) Consider a situation in which a robot becomes stuck due to obstacles or needs to deviate with a large range to get around obstacles. It will break the tracking status without doubt; then, $V$ will blow up and $\dot{s}$ will converge to 0 correspondingly, which leads to a low speed of the whole formation. Therefore, the reference velocities are expected to reach the desired velocities in this situation instead of converging to 0. By doing this, the robots who do not get close to the obstacle can maintain formation with the desired velocities, while those who are impeded by obstacles can move away from the obstacles by adjusting their desired velocities continuously. Thus, the design of $\dot{s}$ can be modified.

Define $F = \sum_{i=1}^{N} \Gamma_i$, which can indicate whether some robots become stuck. $F$ will blow up whenever any robot is impeded by obstacles. On the contrary, $F$ converges to 0 under normal conditions. When $F$ has a large value, $\dot{s}$ will converge to 1 instead of 0 despite $V$ being large. Therefore, the parameter $\alpha$ is modified to achieve the following target:

$$
\alpha \to 0, \text{ when } V \to \infty, \\
\alpha \to 1, \text{ when } V \to 0,
$$

Thus, $\alpha$ is designed as follows:

$$
\alpha(F) = -\exp(-kF),
$$

where $k$ is a positive constant.

In conclusion:

$$
\dot{s} = 1 - \exp(-kF) \arctan(V),
$$

3.5. Tracking Controller

The tracking controller is designed based on reference velocities $\left[ v_{ri} \omega_{ri} \right]^T$ and trajectories $\left[ x_{ri} y_{ri} \right]^T$.

Define the tracking errors regarding the robot’s coordination (show in Figure 5) as

$$
\begin{pmatrix}
\epsilon_{xi} \\
\epsilon_{yi} \\
\epsilon_{\theta i}
\end{pmatrix} =
\begin{pmatrix}
\cos \theta_i & \sin \theta_i & 0 \\
-\sin \theta_i & \cos \theta_i & 0 \\
0 & 0 & 1
\end{pmatrix}
\begin{pmatrix}
x_{ri} - x_i \\
y_{ri} - y_i \\
\theta_{ri} - \theta_i
\end{pmatrix}
$$

Then, the error dynamics can be obtained as follows:

$$
\begin{align*}
\dot{\epsilon}_{xi} &= \omega_{\theta i} \epsilon_{yi} - v_i + \cos \epsilon_{\theta i} v_{ri}, \\
\dot{\epsilon}_{yi} &= -\omega_{\theta i} \epsilon_{xi} + \sin \epsilon_{\theta i} v_{ri}, \\
\dot{\epsilon}_{\theta i} &= \omega_{\theta i} - \omega_{\rho i},
\end{align*}
$$

The tracking controller $\left[ v_i \omega_i \right]^T$ should be designed to have a good tracking performance for reference trajectories.

Thus, the tracking controller can be designed as

$$
\begin{align*}
v_i &= v_{ri} + c_{1i} \epsilon_{xi}, \\
\omega_i &= \omega_{\rho i} + 4v_{ri} \left( \epsilon_{yi} \cos \frac{\epsilon_{\theta i}}{2} - \epsilon_{xi} \sin \frac{\epsilon_{\theta i}}{2} \right) \\
&\quad + 4c_{2i} \sin \frac{\epsilon_{\theta i}}{2},
\end{align*}
$$
where $c_1$ and $c_2$ are positive constants.

![Diagram of robot tracking errors](image)

**Figure 5.** The $i$th robot and interpretation of tracking errors.

For the good tracking performance of the trajectory generator, one theorem can be obtained as follows.

**Theorem 1.** With the reference trajectories $[x_{ri} \ y_{ri}]^T$ and velocities $[v_{ri} \ \omega_{ri}]^T$, by applying the tracking controller (23), the closed-loop system is asymptotically stable with the achievement of the following equations:

$$
\lim_{t \to \infty} x_{ri} - x_i \to 0 \\
\lim_{t \to \infty} y_{ri} - y_i \to 0 \\
\lim_{t \to \infty} \theta_{ri} - \theta_i \to 2K\pi, K \in \mathbb{Z}
$$

which means that the tracking controller (23) shows a good tracking performance and can maintain the formation for a group of $N$ mobile robots.

**Proof.** Define

$$
\eta_i = 1 - \cos \frac{\epsilon_{\theta i}}{2}.
$$

We note that $\eta_i$ is non-negative and converges to 0 if and only if $\epsilon_{\theta i}$ converges to $4K\pi, K \in \mathbb{Z}$ [35].

Thus, $\eta_{el}$ instead of $\epsilon_{\theta i}$ is applied to design a Lyapunov function as follows:

$$
V_i = \frac{1}{2} \epsilon_{x i}^2 + \frac{1}{2} \epsilon_{y i}^2 + \eta_i + \Gamma_i,
$$

where $\Gamma_i$ is potential function (5) so that $V_i$ is non-negative and $V_i \to \infty$ as $\|\epsilon_{xi}\| + \|\epsilon_{yi}\| \to \infty$.

Taking the time derivative of $V_i$:

$$
\dot{V}_i = \dot{\epsilon}_{x i} \epsilon_{x i} + \dot{\epsilon}_{y i} \epsilon_{y i} + \frac{1}{2} \epsilon_{\theta i} \sin \frac{\epsilon_{\theta i}}{2} + \dot{\Gamma}_i
$$

(27)
where the time derivative of potential function $\Gamma_i$ has the following form:

$$\dot{\Gamma}_i = \begin{cases} \frac{4d(d^2-R^2)(R^2-r^2)}{(d^2-r^2)^3} \dot{d}, & r < d \leq R \\ 0, & d > R \end{cases} \quad (28)$$

We note that $d$ is the distance between the robot and the obstacle. When the robot is outside the collision region, $\dot{\Gamma}_i$ is equal to 0. While inside the collision region—i.e., $r < d \leq R$—then $\frac{4d(d^2-R^2)(R^2-r^2)}{(d^2-r^2)^3} \leq 0$, which means the sign of $\dot{\Gamma}_i$ depends on the sign of $\dot{d}$.

Considering Remark 1, the reference trajectories can be adjusted dynamically to avoid obstacles; thus, $d$ is guaranteed to increase when the robot is in the collision region. Therefore, it can be concluded that $\dot{d} \geq 0$; thus, $\dot{\Gamma}_i \leq 0$ always holds.

Then, substituting (22) and (23) into (27), the derivative of $V_i$ can be calculated as follows:

$$\dot{V}_i = \dot{e}_x \dot{e}_x + \dot{e}_y \dot{e}_y + \frac{1}{2} \dot{\theta}_i \sin \frac{\theta_i}{2} + \dot{\Gamma}_i$$

$$\leq \dot{e}_x \dot{e}_x + \dot{e}_y \dot{e}_y + \frac{1}{2} \dot{\theta}_i \sin \frac{\theta_i}{2}$$

$$= \left[ \dot{e}_x (v_{dx} - v_{ly}) + 2 \sin \frac{\theta_i}{2} \left( v_{dy} \cos \frac{\theta_i}{2} - v_{dl} \sin \frac{\theta_i}{2} + 1 - \frac{1}{4} \frac{\omega_i}{R} \right) \right]$$

$$= - c_1 e_{xi}^2 - 2 c_2 l_1 \sin \frac{\theta_i}{2}$$

As $V_i \geq 0, \dot{V}_i \leq 0$, then $V$ is bounded, which demonstrates that $e_{xi}$ and $e_{yi}$ are bounded in $t$. Moreover, $\dot{V}_i$ is uniformly continuous and bounded, so $\dot{V}_i \to 0$, with regard to Barbalat’s Lemma; then, $e_{xi} \to 0, \sin \frac{\theta_i}{2} \to 0$ as $t \to \infty$.

To show $e_{yi} \to 0$ when $e_{yi} \to 0, \sin \frac{\theta_i}{2} \to 0$, define

$$f(t) = e_{yi} \sin \frac{\theta_i}{2} \quad (30)$$

Since $e_{yi}$ is bounded and $\sin \frac{\theta_i}{2} \to 0$, then $f(t) \to 0$.

Taking the time derivative of $f(t)$,

$$f(t) = e_{yi} \sin \frac{\theta_i}{2} + e_{yi} (\omega_i - \omega_i) \cos \frac{\theta_i}{2}$$

$$= (-\omega_i e_{xi} + v_{ri} \sin \frac{\theta_i}{2}) \sin \frac{\theta_i}{2} + e_{yi} (\omega_i - \omega_i) \cos \frac{\theta_i}{2}$$

$$= - 4 v_{ri} \frac{\rho_{yi}}{2} \cos \frac{\theta_i}{2} + \sin \frac{\theta_i}{2} \left[ e_{yi} \cos \frac{\theta_i}{2} (4 v_{ri} \rho_{xi} - 4 c_2) + 2 \cos \frac{\theta_i}{2} (-\omega_i e_{xi} + v_{ri} \sin \frac{\theta_i}{2}) \right]$$

$$= f_1(t) + f_2(t),$$

where $f_1(t), f_2(t)$ are defined as follows:

$$f_1(t) = - 4 v_{ri} \rho_{yi} \cos \frac{\theta_i}{2},$$

$$f_2(t) = \sin \frac{\theta_i}{2} \left[ e_{yi} \cos \frac{\theta_i}{2} (4 v_{ri} \rho_{xi} - 4 c_2) + 2 \cos \frac{\theta_i}{2} (-\omega_i e_{xi} + v_{ri} \sin \frac{\theta_i}{2}) \right]. \quad (32)$$

where $\omega_{\xi}$ can be substituted by (23).
Since \( \sin \frac{\theta_i}{2} \to 0 \), \( f_2(t) \to 0 \). \( f_1(t) \) is uniformly continuous in \( t \), the following can be derived:

\[
f_1(t) \to 0. \tag{33}
\]

We note that \( v_{ri} \neq 0 \). Thus,

\[
\epsilon_{yi} \to 0. \tag{34}
\]

Thus, we can see that \( \epsilon_{yi} \to 0 \) when \( \epsilon_{xi} \to 0 \), \( \sin \frac{\theta_i}{2} \to 0 \), the stability of the closed-loop system by applying controller \( [v_i \omega_i]^T \) is guaranteed and the controller can achieve good tracking performance.

**Remark 2.** When the \( i \)th robot is outside the detection region and no collision occurs, the reference trajectories \( [x_{ri} \ y_{ri}]^T \) are equal to the desired \( [x_{di} \ y_{di}]^T \). Thus, with Theorem 1, \( N \) mobile robots can move along the reference trajectories, where the tracking errors can converge to \( 0 \) in the situation of no collision.

**Remark 3.** When the \( i \)th robot is inside the detection region and a collision is about to occur, the reference trajectories are replanned as \( R_{ri}(t) = R_{ai}(t) \) to deviate from the desired ones considering a collision in the environment. Then, using Theorem 1 to guarantee the good tracking of reference trajectories \( R_{ri}(t) = R_{ai}(t) \), and by the decrease of parameter \( \alpha \) in \( s \), which is used to overcome the stuck situation and further track the reference trajectories, \( N \) mobile robots can move along the reference trajectories and avoid the obstacle, where the tracking errors converge to \( 0 \) in the situation of collision.

### 4. Simulation Setup

To illustrate that the group of multiple mobile robots can achieve collision avoidance with the proposed controller with tracking errors kept as small as possible, the following three sets of comparative simulations are implemented based on the MATLAB/Simulink environment.

#### 4.1. Simulation of Set 1

This simulation set is designed to illustrate that the controller with collision avoidance can show better performance in a relatively complex environment than without. To obtain a fair and effective comparison, two controllers are compared as follows:

- **C1:** Virtual-structure-based formation controller with \( P_1 \) considering collision avoidance. The control law is selected as (23), where \( c_{1i} = 10 \) and \( c_{2i} = 5 \), and the parameters in adjusted component (9) in Section 3.2 is selected as \( c_{3i} = 1 \) and \( c_{4i} = 1 \). The parameter in \( P_1 \) (17) is selected as \( \alpha = 1 \).

- **C2:** Virtual-structure consensus tracking control (proposed in [18], without collision avoidance). The control law is designed as

\[
u_i = B_i^{-1}(-k \sum_{j \in N} (r_i - r_j) - (r_i - r_d) + r_d)
\]

where

\[
\begin{bmatrix}
x_{ri} \\
y_{ri}
\end{bmatrix} = \begin{bmatrix}
x_j \\
y_j
\end{bmatrix} + \begin{bmatrix}
\cos \theta_i & \sin \theta_i \\
-\sin \theta_i & \cos \theta_i
\end{bmatrix} \begin{bmatrix}
d_x \\
d_y
\end{bmatrix},
\]

\[
\begin{bmatrix}
d_x \\
d_y
\end{bmatrix}^T
\]

is the distance between the position of the \( i \)th robot and the position of the center of the virtual structure \( k = 1 \). Then,

\[
B_i = \begin{bmatrix}
\cos \theta_i & -d_x \sin \theta_i - d_y \cos \theta_i \\
\sin \theta_i & d_x \cos \theta_i - d_y \sin \theta_i
\end{bmatrix},
\]
\[ r_d = \begin{bmatrix} x_d \\ y_d \end{bmatrix}^T \] is the desired trajectory.

The workshop is 4 m × 4 m square in the simulation set, which contains three mobile robots cooperating with each other. Assume that the shape of each robot is round, each robot can detect the position of others and obstacles in the workshop are within a certain range. A triangle formation is expected and built by the virtual structure (show in Figure 6a) and the distance \[ \begin{bmatrix} l_{x_1} \\ l_{y_1} \end{bmatrix}^T \] between the \(i\)th desired trajectory and the center of the virtual structure \[ \begin{bmatrix} x_{d_0} \\ y_{d_0} \end{bmatrix}^T \] is given as follows:

\[
\begin{align*}
\begin{bmatrix} l_{x_1} \\ l_{y_1} \end{bmatrix}^T &= \begin{bmatrix} 0 \\ 0.6 \end{bmatrix}^T \\
\begin{bmatrix} l_{x_2} \\ l_{y_2} \end{bmatrix}^T &= \begin{bmatrix} 0 \\ -0.6 \end{bmatrix}^T \\
\begin{bmatrix} l_{x_3} \\ l_{y_3} \end{bmatrix}^T &= \begin{bmatrix} -0.6 \\ 0 \end{bmatrix}^T
\end{align*}
\]

Then, the desired trajectory of the virtual center is designed as follows:

\[
\begin{align*}
x_{d_0} &= \begin{cases} 
-2 + 0.2t & t \leq T_1 \\
-1 + \sin(0.2(t - T_1)) & T_1 < t \leq T_2 \\
1 - \cos(0.2(t - T_2)) & T_2 < t \leq T_3 \\
1 + 0.2(t - T_3) & T_3 < t \leq T_4
\end{cases} \\
y_{d_0} &= \begin{cases} 
-1 & t \leq T_1 \\
-\cos(0.2(t - T_1)) & T_1 < t \leq T_2 \\
\sin(0.2(t - T_2)) & T_2 < t \leq T_3 \\
1 & T_3 < t \leq T_4
\end{cases} \\
\theta_{d_0} &= \begin{cases} 
0 & t \leq T_1 \\
-0.2(t - T_1) & T_1 < t \leq T_2 \\
-\frac{\pi}{2} + 0.2(t - T_2) & T_2 < t \leq T_3 \\
0 & T_3 < t \leq T_4
\end{cases}
\]

where \(T_1 = 5\), \(T_2 = 5 + 2.5\pi\), \(T_3 = 5 + 5\pi\), \(T_4 = 10 + 5\pi\), and the desired trajectory of the \(i\)th robot \(R_{di}\) is calculated based on \(R_{d0}\), as shown in Figure 6b. We note that the desired trajectory designed for formation maintenance has discontinuities.

![The triangle formation setup and desired trajectories for the robot formation in set 1.](image)

**Figure 6.** The triangle formation setup and desired trajectories for the robot formation in set 1.
The radius of the robot is set as 0.175 m, and the detection region radius is set to be 0.3 m. Then, the two obstacles are set in the workshop with the radius $r = 0.2$ m, where the radius of detection region $R$ is 0.3 m, and the positions are located at $(1, 0.2)$ and $(1, 1.9)$, respectively.

For the easy illustration of the fact that the tracking error can be as small as possible with regard to obstacles in the environment, an objective function $F_{\text{obj}}$ is designed to make comparisons between the proposed controller $C_1$ and its counterpart $C_2$, where the definition of $F_{\text{obj}}$ is as follows:

$$F_{\text{obj}} = \sum_{i=1,2,3} \Gamma_i + \sum_{i=1,2,3} \left[ (x_{di} - x_i)^2 + (y_{di} - y_i)^2 + \sin\left(\frac{\theta_{di} - \theta_i}{2}\right)^2 \right]$$

(39)

The results of simulation set 1 is shown in Figures 7–11. Figures 7 and 8 show the tracking performance of the formation for robots, and the triangle represents the formation shape at that moment. From Figure 7, it can be seen intuitively that the triangle formation is maintained well in the process of tracking desired trajectories via $C_1$. We note that the robots can avoid obstacles via $C_1$ while the robots in $C_2$ cannot, since $C_2$ does not consider the obstacle avoidance in the control design.

![Figure 7](image_url) Figure 7. The formation tracking performance for robots in formation with $C_1$.

![Figure 8](image_url) Figure 8. The formation tracking performance for robots in formation with $C_2$. 
Figures 9 and 10 show the tracking errors of the whole group. It should be noted that the desired trajectories have three mutation points, which result in the blowing up of tracking errors at these moments. The tracking errors caused by trajectory catastrophe in C1 are obviously lower than those in C2. However, the group in C1 avoids the obstacles at the 20th second, which leads to an increase of tracking errors in C1 at the end of simulation.

![Figure 9. The formation tracking errors for robots in formation with C1.](image1)

![Figure 10. The formation tracking errors for robots in formation with C2.](image2)

Figure 11 shows the $F_{obj}$ of C1 and C2. Compared to the tracking error, this can better evaluate the tracking and obstacle avoidance performance. We note that the $F_{obj}$ of C1 increases slightly when the robots get close to the obstacles; this is because the desired trajectories can deviate for robots in C1 to avoid the obstacles, which results in the decrease of $\Gamma$ and means the achievement of obstacle avoidance, though the tracking error rises slightly. On the contrary, $F_{obj}$ of C2 blows up after the group first encounter the mutation point at the fifth second; this is because C2 does not consider the collision avoidance and the group gets close to the obstacle, which results in the sharp increase of $\Gamma$ although the tracking error is small. As a result, the desired trajectories suddenly change when close to the obstacles, but $F_{obj}$ of C1 is obviously lower than that of C2, which further reflects the good tracking and obstacle avoidance performance of C1.
Figure 11. The objective function $F_{obj}$ for robots in formation with C1 and C2.

Based on the result of simulation set 1 with different controllers, it can be seen that the proposed C1 has a good formation and tracking performance, which can achieve collision avoidance well and keep the tracking error as small as possible for the whole process.

4.2. Simulation of Set 2

This simulation set is designed to illustrate the effectiveness of path parameter $P_2$ compared to $P_1$ when the group encounters obstacles. Thus, two different sets are applied on C1 to make comparisons as follows:

C1–P1: Apply the proposed controller C1 in set 1 with $P_1$ (proposed in [16]) as

$$\dot{s} = 1 - \arctan(V)$$

C1–P2: Take the same controller C1 with $P_2$ (proposed in this paper) as

$$\dot{s} = 1 - \exp(-F) \arctan(V)$$

In this simulation set, the reference trajectories and workshop are set to be the same as those in set 1. Two obstacles are set in the workshop with a radius $r$ of 0.2 m, where the radius of the detection region $R$ is 0.3 m and the positions are located at $(1, 0.2)$ and $(1, 1.9)$, respectively. An additional obstacle is set to further show the compared performance, whose radius $r$ is 0.5 m with a radius of the detection region $R$ of 0.8 m, and the position is located at $(-1, -1)$. The same triangle formation as in set 1 is expected.

Similar to simulation set 1, the objective function in (39) is used to evaluate the tracking performance for robots, and Figures 12 and 13 show the tracking performance of formation maintenance for robots. The black triangle in Figures 12 and 13 is generated every fixed time, which represents the formation at this moment. Thus, as can be speculated from the intensity of the black triangle, the group under $P_1$ is impeded by the obstacles and cannot complete the trajectories at the same time, while the group under $P_2$ can complete them with better formation maintenance.

In Figure 14, the $F_{obj}$ of two groups increase sharply at 13 s, which is caused by the increase of both the potential function and tracking errors when the group first encounters obstacles. We note that the $F_{obj}$ of $P_1$ changes slightly behind $P_2$, because the path parameter designed in $P_1$ cannot decrease the weight of the formation to satisfy tracking when encountering obstacles, meaning that $P_1$ can be stuck at this moment. Then, when the group encounters obstacles that cannot be passed easily in 24 s,
the $F_{obj}$ of P1 grows quickly and maintains a high value because the group is stuck and the potential function is high, while the $F_{obj}$ of P2 rises first and then falls, which illustrates that P2 can give priority to tracking rather than formation to pass the obstacles. The results show the improvement of P2 on keeping the $F_{obj}$ small in the tracking process. Therefore, a good formation tracking performance is achieved for groups of multiple robots via C1–P2, especially when close to obstacles.

Figure 12. The formation tracking performance for robots in formation with C1–P1.

Figure 13. The formation tracking performance for robots in formation with C1–P2.

Figure 14. The objective function $F_{obj}$ for robots in formation with P1 and P2.
4.3. Simulation of Set 3

This simulation set is designed to further show the effectiveness of C1–P2 with different numbers of robots and a relatively complex workspace. The robots in this set can form linear, triangle and quadrilateral formations to track an S-shaped trajectory. The desired trajectory \[
\begin{bmatrix}
  x_{d0} \\
y_{d0}
\end{bmatrix}
\] of the center robot is given as follows:

\[
x_{d0} =\begin{cases}
  0.2t - 4 & t \leq T_1 \\
  2\sin(0.2(t - T_1) + 2) & T_1 < t \leq T_2 \\
  -0.2(t - T_2) + 2 & T_2 < t \leq T_3 \\
  -2\sin(0.2(t - T_3) - 2) & T_3 < t \leq T_4 \\
  0.2(t - T_4) - 2 & T_4 < t
\end{cases}
\]

\[
y_{d0} =\begin{cases}
  4 & t \leq T_1 \\
  2\cos(0.2(t - T_1) + 2) & T_1 < t \leq T_2 \\
  0 & T_2 < t \leq T_3 \\
  2\cos(0.2(t - T_3) - 2) & T_3 < t \leq T_4 \\
  -4 & T_4 < t
\end{cases}
\]

where \(T_1 = 30, T_2 = 30 + 10\pi, T_3 = 50 + 10\pi, T_4 = 50 + 20\pi\). The desired trajectory of the \(i\)th robot \(R_{di}\) is calculated based on \(R_{d0}\).

The workshop is 10 m \times 10 m square in this simulation set, which contains four obstacles. Three of the obstacles are set with a radius \(r\) of 0.2 m and a radius of the detection region \(R\) of 0.3 m. Their positions are located at \((4, 3.5), (-1.7, -0.5)\) and \((-2, 0.5)\). Another obstacle has a different size, whose radius \(r\) is 0.5 m with a radius of the detection region \(R\) of 0.8 m, and the position is located at \((2, 3)\). The properties of robots are set to be the same as those in set 1.

The results of simulation set 3 are shown in Figures 15–17. These three figures show the tracking and formation performance for different numbers of robots; the dotted line represents the desired trajectory of the virtual center and the black polygon represents the formation at this moment. The results show that the entire team have a good tracking and formation performance, which further shows that C1 proposed in this paper can adapt to different numbers of robots well and can be applied to relatively complex environments.

![Figure 15. The formation tracking performance for two robots.](image-url)
5. Experiment

To further verify the effectiveness of C1–P2 in a real environment, an experiment is implemented in which the setup is selected to be the same as the simulation setup of set 2 in Section 4.

The experiment is performed with the Quanser Unmanned Vehicle System (UVS), which includes an operator station PC, three Q-bot mobile robots and 12 OptiTrack cameras. The Q-bots are mobile robots which are driven by two driven wheels, and they can receive velocity control commands from the station PC via a WIFI network. The OptiTrack cameras are responsible for localizing objectives (i.e., Q-bots and obstacles) and send the messages to the station PC. The operator station PC acts as the host for the models running on the Q-bots and handles the processing of the OptiTrack data.

The results of experiment are shown in Figures 18 and 19. Figure 18 shows the tracking performances under C1–P2 in set 2, and Figure 19 represent the trajectories under C1–P2; the trajectories of simulation results are shown for comparison. Then, it can be seen intuitively that the experiment has
the same results as the simulation, which further verifies the effectiveness of the C1–P2 at achieving formation and tracking performance with collision avoidance in a real environment.

Figure 18. The experimental performance under C1–P2. (a–f) represent the performance every 15 s.

Figure 19. The trajectories of simulation and experimental results under C1–P2.
6. Conclusions

In this paper, a virtual-structure-based formation control design is proposed for a group of multiple mobile robots with the consideration of obstacle avoidance, where $N$ basic desired trajectories are generated in advance and parameterized by the path parameter $s$ to maintain the formation for multiple mobile robots. To achieve collision avoidance, a trajectory generator with a potential function is designed that can adjust the basic trajectories and replan the reference trajectories. Subsequently, a tracking controller is designed to achieve the good tracking performance of mobile robots, and the asymptotical stability is guaranteed via the Lyapunov theorem. A comparative simulation with three sets is implemented with the design of an objective function $F_{obj}$ to evaluate the tracking performance, and the simulation results verify the good formation and tracking performance of the proposed formation control design for multiple mobile robots with obstacle avoidance and smaller tracking errors. Finally, the real robot experiment is implemented to further verify the effectiveness of the proposed control design.

Since the potential function-based path planning approach has essential drawbacks, such as the robot group possibly becoming stuck in a deadlock, the path parameter $P_2$ proposed in this paper is developed to further improve the deadlock situation. In detail, the controller $C_1$–$P_2$ can adjust the path parameter to lower the weight of formation at this time, which maintains the good tracking and formation of the rest of this group with the sacrifice of stopping the motions of the stuck robot. However, this deadlock situation cannot be thoroughly solved because the stuck robot is left behind. Meanwhile, due to the hardware condition, the obstacle set is static, which can be relatively easily detected via the OptiTrack cameras. However, in practice, since a collision is usually instantaneous and time-varying, the online perception of obstacles and the identification of obstacle shapes becomes quite challenging, which may lead to a performance reduction in a real environment. Therefore, in future, other collision-avoidance algorithms and the online perception of obstacles and their shapes will be investigated to further improve the tracking and formation performance in the practical environment.

Author Contributions: Conceptualization, X.C. and Z.C.; methodology, X.C. and F.H.; software, X.C. and Y.Z.; validation, F.H., Y.Z. and S.L.; formal analysis, F.H.; investigation, X.C.; resources, S.L. and Y.N.; data curation, S.L. and J.T.; writing—original draft preparation, X.C.; writing—review and editing, X.C.; visualization, S.L. and S.Z.; supervision, S.L. and Z.C.; project administration, S.L., J.T. and S.Z.; funding acquisition, Z.C., Y.N. and S.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by the Natural Science Foundation of Zhejiang Province, China (No. LY19E050016), National Natural Science Foundation of China (No. 51809235 and No. 61603332), and the Strategic Priority Research Program of the Chinese Academy of Sciences (No. XDA22040202).

Conflicts of Interest: The authors declare no conflict of interest.

Nomenclature

| Notation | Description |
|----------|-------------|
| $x_i \ y_i$ | Real trajectories |
| $x_{di} \ y_{di}$ | Desired trajectories |
| $x_{ri} \ y_{ri}$ | Reference trajectories |
| $x_{ai} \ y_{ai}$ | Adjusted trajectories |
| $s$ | Path parameter |
| $\eta$ | Switch function |
| $\theta_i$ | Real angle |
| $\theta_{di}$ | Desired angle |
| $\theta_{ri}$ | Reference angle |
| $\theta_{ai}$ | Adjusted angle |
| $\Gamma_i$ | Potential function |
| $V$ | Formation error |
\[
\begin{align*}
\nu_i & \quad \omega_i \\
\nu_{di} & \quad \omega_{di} \\
\nu_{ri} & \quad \omega_{ri} \\
\nu_{ai} & \quad \omega_{ai} \\
\psi_{ri} & \quad \psi_{ai}
\end{align*}
\]

Real velocities

Desired velocities

Reference velocities

Adjusted velocities

Virtual forces

Sum of potential function

References

1. Liu, Z.; Zhang, Y.; Yu, X.; Yuan, C. Unmanned surface vehicles: An overview of developments and challenges. *Annu. Rev. Control* 2016, 41, 71–93. [CrossRef]

2. Sun, W.; Liu, Y.; Gao, H. Constrained sampled-data ARC for a class of cascaded nonlinear systems with applications to motor-servo systems. *IEEE Trans. Ind. Inform.* 2019, 15, 766–776. [CrossRef]

3. Yao, J.; Deng, W. Active Disturbance Rejection Adaptive Control of Hydraulic Servo Systems. *IEEE Trans. Ind. Electron.* 2017, 64, 8023–8032. [CrossRef]

4. Chen, Z.; Huang, F.; Yang, C.; Yao, B. Adaptive fuzzy backstepping control for stable nonlinear bilateral teleoperation manipulators with enhanced transparency performance. *IEEE Trans. Ind. Electron.* 2020, 67, 746–756. [CrossRef]

5. Chen, Z.; Huang, F.; Sun, W.; Gu, J.; Yao, B. RBF-Neural-Network-Based Adaptive Robust Control for Nonlinear Bilateral Teleoperation Manipulators With Uncertainty and Time Delay. *IEEE/ASME Trans. Mechatron.* 2020, 25, 906–918. [CrossRef]

6. Mu, D.; Zhao, Y.; Wang, G.; Fan, Y.; Bai, Y. Course control of USV based on fuzzy adaptive guide control. In Proceedings of the 2016 Chinese Control and Decision Conference (CCDC), Yinchuan, China, 28–30 May 2016.

7. Liu, Y.; Richard, B. Efficient multi-task allocation and path planning for unmanned surface vehicle in support of ocean operations. *Neurocomputing* 2018, 275, 1550–1566. [CrossRef]

8. Jonathan, R.T.; Beard, R.W.; Young, B.J. A decentralized approach to formation maneuvers. *IEEE Trans. Robot. Autom.* 2003, 19, 933–941.

9. Wang, P.K.C. Navigation strategies for multiple autonomous mobile robots moving in formation. *J. Robot. Syst.* 1991, 8, 177–195. [CrossRef]

10. Yamaguchi, H. Adaptive formation control for distributed autonomous mobile robot groups. In Proceedings of the International Conference on Robotics and Automation, Albuquerque, NM, USA, 25 April 1997; pp. 2300–2305.

11. Beard, R.W.; Lawton JHadaegh, F.Y. A coordination architecture for spacecraft formation control. *IEEE Trans. Control Syst. Technol.* 2001, 9, 777–790. [CrossRef]

12. Cortes, J. Global and robust formation-shape stabilization of relative sensing Networks. *Automatica* 2009, 45, 2754–2762. [CrossRef]

13. Fahimi, F. Sliding-mode formation control for underactuated surface vessels. *IEEE Trans. Robot.* 2007, 23, 617–622. [CrossRef]

14. Ghommam, J.; Calvo, O.; Rozenfeld, A. Coordinated path following for multiple underactuated auvs. In Proceedings of the OCEANS 2008—MTS/IEEE Kobe Techno-Ocean, Kobe, Japan, 8–11 April 2008; pp. 1–7.

15. Ghommem, J.; Mniif, F.; Poisson, G.; Derbel, N. Nonlinear formation control of a group of underactuated ships. In Proceedings of the OCEANS 2007—Europe, Aberdeen, UK, 18–21 June 2007; pp. 1–8.

16. Do, K.; Pan, J. Nonlinear formation control of unicycle-type mobile robots. *Robot. Auton. Syst.* 2007, 55, 191–204. [CrossRef]

17. Do, K. Practical formation control of multiple underactuated ships with limited sensing ranges. *Robot. Auton. Syst.* 2011, 59, 457–471. [CrossRef]

18. Yoshioka, C.; Namerikawa, T. Formation Control of Nonholonomic Multi-Vehicle Systems Based on Virtual Structure. In Proceedings of the 17th World Congress the International Federation of Automatic Control, Seoul, Korea, 6–11 July 2008; pp. 5149–5154.

19. Sun, X.; Wang, G.; Fan, Y.; Mu, D.; Qiu, B. Collision avoidance of podded propulsion unmanned surface vehicle with colregs compliance and its modeling and identification. *IEEE Access* 2018, 6, 55473–55491. [CrossRef]
20. Mastellone, S.; Stipanovic, D.; Spong, M. Remote Formation Control and Collision Avoidance for Multi-Agent Nonholonomic Systems. In Proceedings of the 2007 IEEE International Conference on Robotics and Automation, Roma, Italy, 10–14 April 2007; pp. 1062–1067.

21. Regula, G.; Lantos, B. Formation Control of a Large Group of UAVs with Safe Path Planning. In Proceedings of the 21st Mediterranean Conference on Control and Automation (MED’13), Crete, Greece, 25–28 June 2013; pp. 987–993.

22. Do, K. Formation control of multiple elliptical agents with limited sensing ranges. *Automatica* 2012, 48, 1330–1338. [CrossRef]

23. Liu, T.; Jiang, Z. Distributed formation control of nonholonomic mobile robots without global position measurements. *Automatica* 2013, 49, 592–600. [CrossRef]

24. Sun, Z.; Zhang, G.; Lu, Y.; Zhang, W. Leader-follower formation control of underactuated surface vehicles based on sliding mode control and parameter estimation. *ISA Trans.* 2018, 72, 15–24. [CrossRef]

25. Cui, R.; Ge, S.S.; Ho, B.V.; Choo, Y. Leader-follower formation control of underactuated autonomous under water vehicles. *Ocean Eng.* 2010, 37, 1491–1502. [CrossRef]

26. Lapierre, L.; Soetanto, D.; Pascoal, A. Coordinated motion control of marine robots. *IFAC Proc. Vol.* 2003, 36, 217–222. [CrossRef]

27. Schoerling, D.; Kleck, C.V.; Fahimi, F.; Koch, C.R.; Ams, A.; Lober, P. Experimental test of a robust formation controller for marine unmanned surface vessels. *Auton. Robot.* 2010, 28, 213–230. [CrossRef]

28. Loria, A.; Dasdemir, J.; Jarquin, N.A. Leader-Follower Formation and Tracking Control of Mobile Robots Along Straight Paths. *IEEE Trans. Control Syst. Technol.* 2016, 24, 727–732. [CrossRef]

29. Consolini, L.; Morbidi, F.; Prattichizzo, D.; Tosques, M. Leader-follower formation control of nonholonomic mobile robots with input constraints. *Automatica* 2008, 44, 1343–1349. [CrossRef]

30. Chen, Y.; Tian, Y. A backstepping design for directed formation control of three agents in the plane. Int. J. Robust Nonlinear Control 2009, 19, 729–745. [CrossRef]

31. Li, W.; Chen, Z.; Liu, Z. Leader-following formation control for second-order multiagent systems with time-varying delay and nonlinear dynamics. *Nonlinear Dyn.* 2013, 72, 803–812. [CrossRef]

32. Balch, T.; Arkin, R. Behavior-based formation control for multirobot teams. *IEEE Trans. Robot. Autom.* 1998, 14, 926–939. [CrossRef]

33. van den Broek, T.H.A.; van de Wouw, N.; Nijmeijer, H. Formation control of unicycle mobile robots: A virtual structure approach. In Proceedings of the 48th IEEE Conference on Decision and Control (CDC), Shanghai, China, 16–18 December 2009; pp. 8328–8333.

34. Do, K. Output-feedback formation tracking control of unicycle-type mobile robots with limited sensing ranges. *Robot. Auton. Syst.* 2009, 57, 34–47. [CrossRef]

35. Yu, X.; Liu, L.; Feng, G. Trajectory tracking for nonholonomic vehicles with velocity constraints. *IFAC Pap.* 2015, 48, 918–923. [CrossRef]

36. Kuriki, Y.; Namekawa, T. Consensus-based cooperative formation control with collision avoidance for a multi-UAV system. In Proceedings of the 2014 American Control Conference, Portland, OR, USA, 4–6 June 2014; pp. 2077–2082.

37. Yan, J.; Guan, X.; Luo, X. Formation and obstacle avoidance control for multiagent systems. *J. Control Theory Appl.* 2011, 9, 141–147. [CrossRef]

38. Wen, G.; Chen, C.L.P.; Liu, Y.J. Formation control with obstacle avoidance for a class of stochastic multiagent systems. *IEEE Trans. Ind. Electron.* 2018, 65, 5847–5855. [CrossRef]

39. Xia, Y.; Na, X.; Sun, Z.; Chen, J. Formation control and collision avoidance for multi-agent systems based on position estimation. *ISA Trans.* 2016, 61, 287–296. [CrossRef]

40. Loris, R.; Pedrocchi, N.; Vicentini, F.; Tosatti, L.M. Industrial compliant robot bases in interaction tasks: A force tracking algorithm with coupled dynamics compensation. *Robotica* 2017, 35, 1732.

41. Giuseppe, F.; Ivanov, D.; Peron, M.; Sgarbossa, F.; Strandhagen, J.O. Increasing flexibility and productivity in industry 4.0 production networks with autonomous mobile robots and smart intralogistics. *Ann. Oper. Res.* 2020, 9, 1–19.