Simultaneous Position and Orientation Planning of Nonholonomic Multirobot Systems: A Dynamic Vector Field Approach

Xiaodong He, Member, IEEE, and Zhongkui Li, Senior Member, IEEE

Abstract—This article considers the simultaneous position and orientation planning of nonholonomic multirobot systems. Different from common research works, which only focus on final position constraints, we model the nonholonomic mobile robot as a rigid body and introduce both of position and orientation constraints for the robot’s final states. In other words, the robot should not only reach the specified position, but also point to the desired orientation simultaneously. The challenge of this problem lies in the underactuation of full-state motion planning, since three states need to be planned by mere two control inputs. To this end, we propose a dynamic vector field (DVF) based on the rigid body modeling. Specifically, the dynamics of the robot orientation is brought into the vector field, implying that the vector field is not static on the 2-D plane anymore, but a dynamic one varying with the attitude angle. Hence, each robot can move along the integral curve of the DVF to arrive at the desired position, and in the meantime, the attitude angle can converge to the specified value following the orientation dynamics. Subsequently, by designing a circular vector field under the framework of the DVF, we further study the obstacle avoidance and multirobot collision avoidance. Finally, numerical simulation and hardware experiment are provided to verify the effectiveness.

Index Terms—Dynamic vector field (DVF), motion planning, multirobot system, nonholonomic mobile robot.

I. INTRODUCTION

 MOTION planning is a fundamental problem in the field of robotics and control, which aims at finding paths or trajectories to guide mobile robots moving from initial conditions to destinations, while avoiding collisions with obstacles and other robots. One of the most typical robotic systems is the nonholonomic mobile robot, also referred to as the unicycle-type vehicle. Although neglected by a number of researchers, two facts regarding the nonholonomic mobile robot should be mentioned above all. One is that the theoretical model of such a kind of robot is essentially a rigid body rather than a point of mass [1], since the robot states include both position and orientation (or attitude). Different from a 3-degree-of-freedom (DOF) rigid body moving freely on a plane, such a robot has no lateral velocity due to the nonholonomic constraint, which naturally brings about the other fact that the nonholonomic mobile robot is indeed an underactuated system [2]. To be more specific, the robot has three DOFs (two translation DOFs and one rotation DOF), while possessing only two control inputs (one linear velocity and one angular velocity). These two facts make the motion planning for nonholonomic mobile robots more challenging and demanding.

A variety of methodologies have been proposed for motion planning, such as roadmap [3], [4], cell decomposition [5], [6], and sampling-based algorithm [7], [8]. Besides, some methods like control barrier function [9] and optimal reciprocal collision avoidance (ORCA) [10] pay more attention to obstacle and collision avoidance of multirobot systems in motion planning. However, the abovementioned results omit the robot’s kinematics or simply regard it as an integrator, so that cannot be directly applied to nonholonomic robots. Although ORCA could be extended to the nonholonomic case by implementing trajectory tracking, e.g., nonholonomic ORCA (NH-ORCA) [11], it can only guarantee boundedness rather than asymptotic convergence of the tracking error.

To deal with the nonholonomic model, some papers employ the optimization-based methods [12], [13], [14], [15], [16], [17], [18], [19]. For example, Bloch et al. [15] investigated dynamic interpolation on Riemannian manifolds and provide necessary conditions for optimality. Li et al. [16] proposed a prioritized optimization so as to compute the planning results efficiently. Cicchella et al. [17] utilized the Bernstein polynomials to transform motion planning into discrete optimization. No denying that optimization has advantages in handling unicycle models, because the nonholonomic constraints can always be included as the optimization constraints. Nevertheless, the feasibility of such optimization are unable to be guaranteed, or it suffers from extremely heavy computational burden. Another drawback is that the control inputs derived from optimization

Authorized licensed use limited to: 1558-2523 © 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.
are open loop, relying only on time, thereby are not robust to disturbances.

Given above shortages, researchers are motivated to investigate feedback motion planning algorithms for mobile robots. A typical closed-loop methodology is the velocity vector field, where a velocity vector related to the state is defined at every point in the configuration space and the integral curve of the vector field converges to the goal point. The most common vector field is defined by the gradient of a potential function, also referred to as the potential field [20], [21], [22], [23], [24], [25], but one of its inherent limitations is the possible local minima other than the desired state [26]. Particular forms of potential function, such as harmonic function [27], [28], [29] and navigation function [30], [31], [32] overcome such a drawback. However, the former has demanding computational complexity related to PDEs, while the latter is difficult to implement since a lower bound to ensure no local minima is unknown in advance.

As a matter of fact, the velocity vector field is not necessarily given by a potential gradient. Instead, it can be defined directly over the configuration space. A few works focus on motion planning via nongradient-based vector fields, like [33] and [34]. In [33], the environment with obstacles is decomposed into convex polytopes, in which simple local vector fields are defined and smoothly blended to form a global vector field convergent to the desired point. In [34], a family of 2-D analytic vector fields is proposed, which exhibits different patterns (attractive or repulsive) by choosing the value of an parameter, so that the overall vector field can finally be obtained by a suitable blending design.

Regardless of open loop or feedback algorithms, the aforementioned results rarely consider a significant yet implicit fact occurring in real-world scenarios. Apart from the position constraint, the orientation of a robot is also typically required to reach a desired direction at the final time. For instance, in the multirobot surveillance mission, the orientation of each robot should point to a certain direction so as to obtain the largest overall surveillance area. Similarly, the attitudes of missiles are generally specified in the terminal guidance so as to realize a better performance of coordinated attack. Thus, these practical demands strongly motivate us that it is necessary to incorporate the orientation constraints into motion planning. In addition, such a motivation is further strengthened from a theoretical point of view. As mentioned above, the model of a nonholonomic mobile robot is a rigid body. Then, serving as a state, the attitude angle should also be specified to be a desired value, similar to the position, rather than being left randomly, which can be referred to as full-state motion planning. Since the nonholonomic mobile robot is an underactuated system, the biggest challenge lies in how to achieve full-state planning by fewer control inputs.

Motivated by abovementioned discussions, in this article, we consider the simultaneous position and orientation planning of nonholonomic multirobot systems by designing a nongradient-based vector field. To begin with, the robot is modeled as a rigid body with nonholonomic constraint rather than a point of mass. More importantly, besides the desired position constraint, we take into account the orientation constraint at the final time instant as well. Next, different from common static vector fields on a 2-D plane, we propose a novel dynamic vector field (DVF) in the sense that the dynamics of the attitude angle are introduced into the vector field. This implies that the velocity direction at a certain point is decided by not only the position but also the orientation of robot at such a point. Thus, by moving along the integral curve of the DVF, the robot can reach the specified position at the terminal time, and meanwhile the attitude angle can converge to the desired value following the orientation dynamics. Subsequently, the control inputs or velocities are designed based on the DVF, where the feedback of an extra angle in the body-fixed frame is brought in as an additional angular velocity, with the result that the robot orientation can be adjusted to the direction of the DVF to deal with the lack of lateral velocity.

Under the framework of the DVF, the problems of obstacle avoidance and mutual-robot-collision avoidance are studied by proposing a circular vector field, where robots can move along the tangential directions of the obstacle’s boundary such that evade possible collisions. In order to guarantee the continuity of the vector field in the workspace, we blend the target navigation and obstacle/collision avoidance vector fields with a bump function, and further exclude the existence of singular points and Zeno behavior of the composite vector field. Moreover, regarding the collision avoidance among robots, we first propose the result in the scenario of two robots, and then extend it to multirobot cases, where the collision avoidance vector field is computed via a weighted sum of the vector fields in different pairs of robots. Note that both the problems of obstacle avoidance and collision avoidance are able to be solved by one circular vector field, which greatly simplifies the design of the planned trajectories.

Apart from the dynamical characteristics, the following merits also distinguish our method from the existing vector-field-based motion planning results. Compared to [33], the proposed DVF is unnecessarily derived based on cell decomposition of the environment, so that advanced high-level discrete motion planning is not required in this article. In contrast to [34], the DVF is global over the state space in the sense that the initial and final configurations can both be chosen arbitrarily, while the vector field in [34] has a separatrix (or mirror line) where the integral curves might be divergent.

The rest of this article is organized as follows. Section II provides mathematical preliminaries and formulates the problem. Section III proposes the DVF, under which obstacle avoidance and collision avoidance are handled by designing a circular vector field in Sections IV and V, respectively. Section VI provides numerical simulation and hardware experiment. Finally, Section VII concludes this article.

Commonly-used notations are defined in advance. The identity matrix in \( \mathbb{R}^n \) is denoted by \( I_n \). The base vectors of \( \mathbb{R}^3 \) are denoted by \( e_1, e_2, e_3 \). The symbol \( \Omega_{m \times n} \) represents a matrix in \( \mathbb{R}^{m \times n} \) with all zero components. The Euclidean norm of a vector is denoted by \( \| \cdot \| \). Variables denoting vectors and matrices are written in bold, while denoting scalars are not.
II. PRELIMINARIES AND PROBLEM STATEMENT

As mentioned in Section I, the model of the nonholonomic mobile robot is a rigid body. Thus, before providing the nonholonomic model, we first consider a fully actuated planar rigid body moving in the 2-D Euclidean space. Let $\mathcal{F}_e$ denote the Earth-fixed frame, and let $\mathcal{F}_b$ represent the body-fixed frame, which is attached to the center of mass of the rigid body. The position of the rigid body in $\mathcal{F}_e$ is given by a vector $p = [x \ y]^T \in \mathbb{R}^2$, and the attitude is specified by a rotation matrix $\mathbf{R}$, which depicts the rotation of $\mathcal{F}_b$ relative to $\mathcal{F}_e$. Herein, the rotation matrix $\mathbf{R}$ belongs to the special orthogonal group $\text{SO}(2) \triangleq \{ \mathbf{R} \in \mathbb{R}^{2 \times 2} | \mathbf{R}^T \mathbf{R} = \mathbf{I}_2, \det(\mathbf{R}) = 1 \}$, and $\mathbf{R}$ can be parameterized by a scalar $\theta$, i.e.,

$$\mathbf{R} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \quad (1)$$

where $\theta \in (-\pi, \pi]$ or $[-\pi, \pi)$ is interpreted as the attitude angle of the rigid body. Let $\omega \in \mathbb{R}$ and $\nu = [\nu_x \nu_y]^T \in \mathbb{R}^2$ denote the rigid body’s angular velocity and linear velocity in the body-fixed frame $\mathcal{F}_b$. Then, the kinematics of the fully actuated rigid body can be given by

$$\dot{x} = v_x \cos \theta - v_y \sin \theta$$  \hspace{1cm} (2a)

$$\dot{y} = v_x \sin \theta + v_y \cos \theta$$  \hspace{1cm} (2b)

$$\dot{\theta} = \omega.$$  \hspace{1cm} (2c)

Regarding nonholonomic mobile robots, due to no side slip of the wheels, the robot cannot move sideways. In other words, the velocity along the $y_b$-axis of the body-fixed frame $\mathcal{F}_b$ is always zero, i.e., $v_y = 0$. Such a constraint is named the nonholonomic constraint, since its reformulation in the Earth-fixed frame $\mathcal{F}_e$, which is a differential equation given by $\dot{x} \sin \theta - \dot{y} \cos \theta = 0$ cannot be integrated to be an algebraic equation. Thus, according to (2), the kinematic model of a nonholonomic mobile robot degenerates to

$$\dot{x} = v_x \cos \theta$$  \hspace{1cm} (3a)

$$\dot{y} = v_x \sin \theta$$  \hspace{1cm} (3b)

$$\dot{\theta} = \omega.$$  \hspace{1cm} (3c)

Given the fact that the nonholonomic mobile robot moves in an environment possibly populated with obstacles, we assume that each obstacle can be bounded by a circular region with radius $r_a > 0$. Moreover, suppose that multiple nonholonomic mobile robots labeled by $\mathcal{I}_Y = \{1, \ldots, N\}$ move in a common workspace simultaneously. Therefore, the potential collisions with obstacles and among robots should both be taken into account in the motion planning process. To acquire the information of obstacles and other robots positions, robot $i$ ($i \in \mathcal{I}_Y$) is assumed to have a circular sensing range with radius $R_{si}$, which can be defined by

$$S_i = \{ q \in \mathbb{R}^2 \mid \|q - p_i\| \leq R_{si} \}$$

where $p_i$ is the current position vector of robot $i$. Then, once obstacles and other robots appear into the sensing region $S_i$, robot $i$ can obtain the position information of them.

The motion planning problem for multiple nonholonomic mobile robots can be formulated as follows.

**Problem Statement:** Regarding $N$ nonholonomic mobile robots described by (3), design linear velocity $v_i$ and angular velocity $\omega$ for robot $i$, where $i \in \mathcal{I}_Y = \{1, \ldots, N\}$, such that each controlled trajectory of the robots, which starts from the initial condition $(x_{i,0}, y_{i,0}, \theta_{i,0})$, can reach the specified destination $(x_{id}, y_{id}, \theta_{id})$, and meanwhile avoid the obstacles and mutual collisions with other robots.

III. DYNAMIC VECTOR FIELD

In this section, we construct a nongradient-based vector field to solve the motion planning problem for a nonholonomic mobile robot in an obstacle-free environment. Without loss of generality, we assume that the destination of the robot $(x_d, y_d, \theta_d)$ is chosen as $(0,0,0)$. For the sake of simplicity, we first rewrite the kinematics (2) and (3) in a more compact form, i.e., the geometric formulation of rigid body systems.

A. System Reformulation

We reformulate the kinematics of nonholonomic mobile robots to be rigid body systems on matrix Lie groups, and the standard formulation can be founded in the literature of geometric control, such as [35],[36],[37]. Specifically, based on the rotation matrix $\mathbf{R} \in \text{SO}(2)$ and the position vector $p \in \mathbb{R}^2$, we define the following matrix:

$$h = \begin{bmatrix} \mathbf{R} & p \\ 0_{1 \times 2} & 1 \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta & x \\ \sin \theta & \cos \theta & y \\ 0 & 0 & 1 \end{bmatrix} \quad (4)$$

which is called the configuration of the rigid body. All the configurations $h$, with matrix multiplication as group operation, constitute a matrix Lie group called the special Euclidean group $\text{SE}(2)$, i.e.,

$$\text{SE}(2) \triangleq \left\{ \begin{bmatrix} \mathbf{R} & p \\ 0_{1 \times 2} & 1 \end{bmatrix} \mid \mathbf{R} \in \text{SO}(2), p \in \mathbb{R}^2 \right\}.$$  \hspace{1cm} (5.1)

Thus, the rigid body’s configuration $h$ is an element in $\text{SE}(2)$. Since the configuration $h$ is uniquely defined by $\mathbf{R}$ and $p$, it can also be simply written as $h = (\mathbf{R}, p) \in \text{SE}(2)$. In geometric mechanics, the rigid body’s velocity $\omega$ and $v$ lie in the associated Lie algebra of $\text{SE}(2)$, which is defined by

$$\text{se}(2) \triangleq \left\{ \begin{bmatrix} S \\ 0_{1 \times 2} \end{bmatrix} q \mid S \in \mathfrak{so}(2), q \in \mathbb{R}^2 \right\}.$$  \hspace{1cm} (5.2)

where $\mathfrak{so}(2)$ is the associated Lie algebra of $\text{SO}(2)$ and given by $\mathfrak{so}(2) \triangleq \{ S \in \mathbb{R}^{2 \times 2} | S^T = -S \}$. Then, the velocity can be formulated in $\text{se}(2)$ as

$$\nu = \begin{bmatrix} \omega \\ v \end{bmatrix} = \begin{bmatrix} 0 & -\omega & v_x \\ \omega & 0 & v_y \\ 0 & 0 & 0 \end{bmatrix}$$  \hspace{1cm} (5.3)
and \( \theta \). Based on (4) and (5), the fully actuated kinematic model (6) can be redefined by

\[
\dot{h} = h \eta \tag{6}
\]

where the configuration \( h \) and the velocity \( \eta \) serve as the state and control input, respectively. Correspondingly, the nonholonomic kinematic model can be rewritten as (6) with an additional nonholonomic constraint \( e_i^2 \eta e_i = 0 \), where \( e_i, i = 1, 2, 3 \), are standard basis in \( \mathbb{R}^3 \).

The velocity lies in the Lie algebra \( se(2) \), while the configuration \( h \) exists in the Lie group \( SE(2) \). In order to design the velocity vector field with configuration feedback, we transform \( h \) from \( SE(2) \) to \( se(2) \) by utilizing the logarithmic map \( \log_{SE(2)} : SE(2) \rightarrow se(2) \) proposed in [37]. Let \( \Upsilon \) denote the logarithm of \( h \), and following [37, Lemma 2], we have

\[
\Upsilon = \log(h) = \begin{bmatrix} \varphi(x, y, \theta) \\ 0 \end{bmatrix} \tag{7}
\]

where \( \varphi(x, y, \theta) \) is a 2 \times 2 skew symmetric matrix with respect to \( \theta \), similar to \( \dot{\omega} \) in (5), and the vector \( \varphi(x, y, \theta) \in \mathbb{R}^2 \) is given by

\[
\varphi(x, y, \theta) = \begin{bmatrix} \chi(\theta) \\ -\chi(\theta) \end{bmatrix} \tag{8}
\]

where \( \chi(\theta) \) is defined by

\[
\chi(\theta) = \begin{cases} \frac{\theta(1 + \cos \theta)}{2 \sin \theta}, & \theta \neq 0 \\ 2, & \theta = 0. \end{cases} \tag{9}
\]

It is not difficult to verify that \( \chi(\theta) \) is a continuous differentiable function, and the derivative at \( \theta = 0 \) is 0. Once defining a matrix

\[
\Xi = \begin{bmatrix} \chi(\theta) \\ -\chi(\theta) \end{bmatrix} \tag{10}
\]

the formula (8) can be simplified as

\[
\varphi(x, y, \theta) = \Xi \eta \tag{11}
\]

where \( \eta \) is the position vector. Since \( \varphi(x, y, \theta) \) is a 2-D vector, let \( \varphi_1(x, y, \theta) \) and \( \varphi_2(x, y, \theta) \) denote its components.

We refer to the twist \( \Upsilon \in se(2) \) defined in (7) as the transformed configuration, which is simply denoted by \( \Upsilon = (\theta, \varphi) \in se(2) \). According to [37, Th. 2], the time derivative of the transformed configuration \( \Upsilon \), i.e., the rigid body kinematics (6) under the coordinates of \( \Upsilon \), is given by

\[
\dot{\Upsilon} = \mathcal{M}(\Upsilon) \eta \tag{12}
\]

where the transformed configuration \( \Upsilon \) is the state, the velocity \( \eta \) is the control input, and \( \mathcal{M}(\Upsilon) \in \mathbb{R}^{3 \times 3} \) is a state-dependent matrix satisfying

\[
\mathcal{M}(\Upsilon) \Upsilon = \Upsilon. \tag{13}
\]

The expression of \( \mathcal{M}(\Upsilon) \) is provided in [37, Lemma 5]. More information on geometric control on matrix Lie groups can be founded in [38], [39], [40], and [41].

Remark 1: The reason of reformulating the rigid body kinematics under the framework of Lie groups is that we intend to incorporate the attitude information into the state variables of translation. It can be observed from the transformed configuration \( \Upsilon \) in (7) that the attitude angle \( \theta \) describes the rotation motion, while the vector \( \varphi \) describes the translation motion. However, the translation state vector \( \varphi \) also relies on the attitude angle \( \theta \), indicating that we have one more DOF to design the velocity vector field by regarding \( \theta \) as an internal parameter. This contributes to the DVF designed in the following section.

B. Vector Field Design

Based on the transformed configuration \( \Upsilon \), we propose the following vector field \( \Gamma_d : \mathbb{R}^2 \times \mathbb{S} \rightarrow \mathbb{R}^2 \), whose components \( \Gamma^x_d \) and \( \Gamma^y_d \) are given by

\[
\Gamma^x_d = -\varphi_1(x, y, \theta) \cos \theta + \varphi_2(x, y, \theta) \sin \theta \tag{14a}
\]

\[
\Gamma^y_d = -\varphi_1(x, y, \theta) \sin \theta - \varphi_2(x, y, \theta) \cos \theta \tag{14b}
\]

where \( \theta \) serves as an internal parameter satisfying

\[
\dot{\theta} = -\theta. \tag{15}
\]

In contrast to common planner vector fields, which are maps of \( \mathbb{R}^2 \rightarrow \mathbb{R}^2 \), such as in [34], [42], and [43], one more parameter \( \theta \in \mathbb{S} \) is introduced to the vector field \( \Gamma_d \) in this article. More importantly, \( \theta \) has an explicit physical meaning, i.e., the attitude angle of the robot. To bring \( \theta \) into the vector field is naturally motivated by the fact that the nonholonomic mobile robot is essentially a kind of rigid bodies rather than a point of mass. Consequently, as a state of the rigid body, the attitude angle is not supposed to be set free at the destination. Instead, it should be guided to the specified value, like positions, after the motion planning. Furthermore, the designation of the final attitude angle has practical interpretation in the sense that it specifies the velocity direction of the subsequent motion.

Thus, we incorporate the attitude information into the vector field, leading to the result that \( \Gamma_d \) is decided by not only position but also orientation. Then, the vector field in \( \mathbb{R}^2 \) is “dynamic” instead of “static.” Herein, the word “dynamic” means \( \Gamma_d \) will vary with the initial value of \( \theta \), which evolves following (15). Therefore, even if for the same one point, the vector direction is probably different due to the attitude angle of the robot. In light of this fact, we refer to \( \Gamma_d \) as a DVF. Fig. 1 provides two DVFs with initial condition \( \theta(0) = \frac{\pi}{2} \) and \( \theta(0) = -\frac{\pi}{2} \), respectively.

![Fig. 1. Dynamic vector field \( \Gamma_d \) with different parameter \( \theta \). (a) \( \theta(0) = \frac{\pi}{2} \), (b) \( \theta(0) = -\frac{\pi}{2} \).](image)
The convergence of $\Gamma_d$ is provided in the following theorem.

**Theorem 1:** The DVF $\Gamma_d$ given in (14) and (15) converges to $(x,y,\theta) = (0,0,0)$ asymptotically.

**Proof:** First, we prove that the transformed configuration $\mathbf{Y}$ converges to $0$ asymptotically. Let $\hat{x} = \Gamma_d^{-1}x$ and $\hat{y} = \Gamma_d^{-1}y$, and by comparing with the kinematics (2), we obtain that the velocity or the control input $\eta$ can be expressed as

$$
\eta = \begin{bmatrix}
\hat{\omega} \\
\hat{v}
\end{bmatrix}
= -\begin{bmatrix}
\hat{\theta} \\
0
\end{bmatrix}
= -\Phi(x,y,\theta)
= -\mathbf{Y}
$$

(16)

where condition (15) is utilized. Define a Lyapunov function $\Phi = \frac{1}{2}(\mathbf{Y}, \mathbf{Y})_{se(2)}$ for $\mathbf{Y} = (\hat{\theta}, \varphi) \in \mathbf{se}(2)$, where $(.,.)_{se(2)} : \mathbf{se}(2) \times \mathbf{se}(2) \rightarrow \mathbb{R}$ represents the inner product defined by $(\alpha_1, \alpha_2)_{se(2)} = \mu_1 \mu_2 + \nu_1 \nu_2$ for $\alpha_1 = (\mu_1, \nu_1) \in \mathbf{se}(2), \alpha_2 = (\mu_2, \nu_2) \in \mathbf{se}(2)$. Taking the time derivative of $\Phi$ along the trajectory of (12), we have

$$\dot{\Phi} = \langle \dot{\mathbf{Y}}, \mathbf{M}(\mathbf{Y})\eta \rangle_{se(2)}.
$$

(17)

Substituting (16) into (17),

$$\dot{\Phi} = -\langle \mathbf{Y}, \mathbf{M}(\mathbf{Y})\eta \rangle_{se(2)} = -\langle \mathbf{Y}, \mathbf{Y} \rangle_{se(2)} < 0 \quad \forall \mathbf{Y} \neq 0 \tag{18}
$$

where property (13) is used. Thus, we obtain that the transformed configuration $\mathbf{Y}$ converges to $0$ asymptotically, implying $\theta \rightarrow 0$ and $\varphi(x,y,\theta) \rightarrow 0$ as $t \rightarrow \infty$.

Next, we prove that the position vector $p = [x, y]^T$ can asymptotically converge to $0$. According to the definition of $\Xi$ in (10) and the following limit:

$$\lim_{\theta \rightarrow 0} \frac{\theta (1 + \cos \theta)}{2 \sin \theta} = \lim_{\theta \rightarrow 0} \frac{1 + \cos \theta - \theta \sin \theta}{2 \cos \theta} = 1
$$

it can be further derived that $\lim_{\theta \rightarrow 0} \det(\Xi) = 1$, where $\det(\cdot)$ represents the determinant of a matrix. Therefore, based on (11), we obtain that $\varphi(x,y,\theta) \rightarrow 0$ if and only if $p \rightarrow 0$.

**Remark 2:** The attitude angle $\theta$ given in (1) takes values in $(-\pi, \pi]$ or $[\pi, \pi)$, indicating that $\theta$ is not continuous if it varies from $\pi$ to $-\pi$ or vice versa. To deal with this problem, the DVF ensures that the $\theta$ will not pass through $\pi$ or $-\pi$ except for the initial condition. In other words, $\theta$ will monotonically decrease to 0, as guaranteed by (15). Thus, $\theta = \pi$ or $\theta = -\pi$ only probably appears at the initial time and does not influence the convergence.

In the following, we extend the DVF $\Gamma_d$ to an arbitrarily-specified final state $(x_d, y_d, \theta_d)$. Based on (4), once $(x,y,\theta) = (0,0,0)$ holds, the corresponding configuration matrix $h$ becomes a $3 \times 3$ identity matrix $I_3$. Thus, the DVF $\Gamma_d$ given in (14) and (15) can drive $h$ to $I_3$, indeed. Let $h_d$ denote the corresponding configuration matrix of the final state $(x_d, y_d, \theta_d)$. Then, the problem now becomes how to define a DVF $\Gamma_d$, which can drive the configuration $h$ to $h_d$. Motivated by the fact that $\Gamma_d$ achieves $h \rightarrow I_3$, we define a new configuration

$$\hat{h} = h_d^{-1}h
$$

(19)

which contains the information of the desired final state $h_d$. The parameterized description of $\hat{h}$, denoted by $(\hat{x}, \hat{y}, \hat{\theta})$, can be given by $\hat{x} = (x-x_d) \cos \theta_d + (y-y_d) \sin \theta_d$, $\hat{y} = -(x-x_d) \sin \theta_d + (y-y_d) \cos \theta_d$, $\hat{\theta} = \theta - \theta_d$. Based on abovementioned definitions, we present the following theorem.

**Theorem 2:** Regarding an arbitrarily-specified final state $(x_d, y_d, \theta_d)$, design a DVF $\Gamma_d : \mathbb{R}^2 \times S \rightarrow \mathbb{R}^2$, whose components $\Gamma_d$ and $\Gamma_d^y$ are given by

$$
\hat{x} = -\varphi_1(\hat{x}, \hat{y}, \hat{\theta}) \cos \hat{\theta} + \varphi_2(\hat{x}, \hat{y}, \hat{\theta}) \sin \hat{\theta}
$$

(20a)

$$
\hat{y} = -\varphi_1(\hat{x}, \hat{y}, \hat{\theta}) \sin \hat{\theta} - \varphi_2(\hat{x}, \hat{y}, \hat{\theta}) \cos \hat{\theta}
$$

(20b)

where $\hat{\theta}$ satisfies $\hat{\theta} = \hat{\theta}$. Then, the DVF $\Gamma_d$ converges to $(x,y,\theta) = (x_d, y_d, \theta_d)$ asymptotically.

**Proof:** According to Theorem 1, the DVF $\Gamma_d$ will converge to $(\hat{x}, \hat{y}, \hat{\theta}) = (0,0,0)$ asymptotically. That is, $\Gamma_d$ can drive the configuration $h$ to the identity matrix $I_3$. Then, according to the definition of $h$ in (19), we have $h_d^{-1}h \rightarrow I_3$, i.e., $h \rightarrow h_d$.
the motion planning controller for nonholonomic mobile robots is designed to be
\begin{align}
v_x &= k_v \tilde{\Gamma}_{d}^2 x \\
\omega &= -k_\omega \theta + k_a \tan^{-1} \frac{\tilde{\Gamma}_{d}^2 y}{\tilde{\Gamma}_{d}^2 x}
\end{align}

where $k_v$, $k_\omega$, and $k_a$ are all positive scalars.

Remark 3: According to the proof of Theorem 1, $k_\omega$ and $k_v$ do not affect the convergence of the DVF, but influence the convergence rate. Of course, the specific values of $k_\omega$ and $k_v$ should be decided by considering the robot’s controller saturation, the motion planning time duration, and so on. Regarding the control gain $k_a$, it cannot be too small. Otherwise, the robot is unable to adjust its orientation following the DVF. A sufficient condition for $k_a$ is that $k_a > 2k_\omega$, as shown in [44].

IV. DVF WITH OBSTACLE AVOIDANCE

This section focuses on the motion planning with obstacle avoidance. Instead of using a repulsive vector field directly, we design a circular vector field around the obstacle, and then blend it with the DVF, which is convergent to the desired configuration. We further analyze the characteristics of the proposed obstacle avoidance vector field.

A. Design of Obstacle Avoidance Vector Field

Consider finite environmental obstacles labeled by $\mathcal{I}_o = \{1, \ldots, M\}$, where $M$ is the number of the obstacles. Assume that each obstacle can be covered by a sphere located at $p_{oi} = [x_{oi} \ y_{oi}]^T$ with radius $r_{oi} > 0$, $i \in \mathcal{I}_o$. Define the following dangerous area and reactive area of the $i$th obstacle
\begin{align}
D_{oi} &= \{ q \in \mathbb{R}^2 \mid \| q - p_{oi} \| \leq r_{oi} \} \\
D_{oi} &= \{ q \in \mathbb{R}^2 \mid r_{oi} < \| q - p_{oi} \| \leq R_{oi} \}
\end{align}

where $R_{oi}$ is the radius of the reactive area $D_{oi}$ and satisfies $r_{oi} < R_{oi} \leq \min_{j \in \mathcal{I}_o} \{ R_{oj} \}$, ensuring that each robot can sense the $i$th obstacle before entering $D_{oi}$. Note that the dangerous area $D_{oi}$ is the region where the obstacle occupies, and is prohibited from being entered by the robots. Regarding the reactive area $D_{oi}$, the robot should take action to avoid possible collision with the obstacle once entering $D_{oi}$. Thus, the obstacle avoidance vector field lies in the reactive area $D_{oi}$. Before designing the vector field, several standard assumptions are imposed as follows.

Assumption 1: There holds $p_{dj} \notin \bigcup_{i \in \mathcal{I}_o} (D_{ai} \cup D_{oi})$ for all $j \in \mathcal{I}_v = \{1, \ldots, N\}$, where $p_{dj} = [x_{dj} \ y_{dj}]^T$ denotes the target position of robot $j$.

Assumption 2: There holds $\| p_{oi} - p_{oj} \| > R_{oi} + R_{oj}$ for all $i \neq j \in \mathcal{I}_o$.

Assumption 1 means that the target position of each robot cannot be covered by the dangerous area and the reactive area of any obstacle. Assumption 2 indicates that any two obstacles are sufficiently far away such that their reactive areas do not overlap. Such assumptions are reasonable and have been widely employed in the literature, e.g., [34], [45]. In addition, note that if two obstacles are too close such that Assumption 2 is violated, then these two obstacles can be regarded as one bigger obstacle to make this assumption hold.

According to Assumption 2, the nonholonomic mobile robot can merely enter the reactive area of one obstacle at each time instant. Therefore, for simplicity, we remove the subscript denoting the obstacle’s label and consider the problem of circumventing one obstacle in the following. Once the robot enters the reactive area $D_{oi}$, in order to design the obstacle avoidance vector field, we first define a repulsive vector
\begin{align}
\Gamma_r = p - p_o
\end{align}

which points from the obstacle to the nonholonomic mobile robot. Then, the distance between the obstacle and the robot can be obtained by $d = \| \Gamma_r \|$. Let $\Gamma_r^\perp$ denote a vector satisfying the following two conditions:
\begin{align}
\langle \Gamma_r^\perp, \Gamma_r \rangle_{\mathbb{R}^2} = 0, \quad \langle \Gamma_r^\perp, v \rangle_{\mathbb{R}^2} > 0
\end{align}

where $\langle \cdot, \cdot \rangle_{\mathbb{R}^2}$ denotes the inner product in $\mathbb{R}^2$, and $v$ is the velocity vector in the body-fixed frame $\mathcal{F}_B$ of the nonholonomic mobile robot. Intuitively, the vector $\Gamma_r^\perp$ is perpendicular to $\Gamma_r$ and projected positively onto the velocity vector $v$. According to the first condition in (26), we can derive that the vector $\Gamma_r^\perp$ always points to a collision-free direction with respect to the obstacle, since $\Gamma_r^\perp$ will be tangent to a certain circle located at $p$ with radius $r$ ($r_o < r \leq R_o$). Thus, all the vectors $\Gamma_r^\perp$ in the reactive area $D_{oi}$ can constitute a circular vector field surrounding the obstacle. Moreover, at the time instant when entering the reactive area $D_{oi}$, the velocity direction is actually the direction of the DVF $\Gamma_d$. Hence, the second condition in (26) can be rewritten as
\begin{align}
\langle \Gamma_r^\perp, \tilde{\Gamma}_d \rangle_{\mathbb{R}^2} > 0
\end{align}

which ensures that the robot can still move toward the destination when circumventing the obstacle.

Nevertheless, it is fairly unreasonable to construct the obstacle avoidance vector field only by $\Gamma_r^\perp$, since there must exist some situations where the nonholonomic mobile robot will never collide with the obstacle although moving in the reactive area $D_{oi}$. In such cases, it is unnecessary to construct obstacle avoidance vector field for the robot with the help of $\Gamma_r^\perp$. Otherwise, the results would become rather conservative. Therefore, in the following, we will design the obstacle avoidance vector field in the reactive area $D_{oi}$, denoted by $\Gamma_o$, according to different
situations of the robot with respect to the obstacle. The detailed construction of $\Gamma_o$ are provided as follows.

Assume that the robot has entered the reactive area $D_o$, i.e., the distance $d$ satisfying $r_o < d \leq R_o$. Let $\theta_r$ represent the angle between the velocity direction $v$ and the line segment connecting robot and obstacle, as shown in Fig. 3, which can be computed by

$$\theta_r = \arccos \frac{\langle v, -\Gamma_r \rangle_{\mathbb{R}^2}}{\|v\| \cdot \|\Gamma_r\|}.$$ 

Hence, according to the value of $\theta_r$, three different cases can be proposed as follows.

1) If $\theta_r \geq \frac{\pi}{2}$, as shown in Case 1 of Fig. 3, the robot does not have the risk of colliding with the obstacle. Then, in this case, the obstacle avoidance vector field can absolutely be chosen as the convergent DVF in the free space, i.e., $\Gamma_o = \Gamma_d$.

2) If $0 < \theta_r < \frac{\pi}{2}$, as shown in Case 2 of Fig. 3, it is possible that the robot will collide with the obstacle in a future time. Thus, we set the obstacle avoidance vector field to be $\Gamma_o = \Gamma_r$, which is defined in (26). Then, the robot will turn to the direction that is perpendicular to the radius of the obstacle so as to achieve the obstacle avoidance.

3) Except for abovementioned two cases, the rest one is $\theta_r = 0$, as shown in Case 3 of Fig. 3. Once $\theta_r = 0$, it can be derived that $\langle \Gamma_r^\perp, v \rangle = 0$, indicating the vector $\Gamma_r^\perp$ is perpendicular to the velocity $v$, so that we cannot define the obstacle avoidance vector field by checking whether the projection of $\Gamma_r^\perp$ onto $v$ is positive, as given in (26). In addition, the reason why we choose the $\Gamma_r^\perp$ positively projected onto $v$ is that such a vector can make the robot turn a smaller attitude angle for obstacle avoidance than the negatively projected one. However, when $\theta_r = 0$, the rotation angle from $v$ to $\Gamma_r^\perp$ is always $\frac{\pi}{2}$, either clockwise or anticlockwise. Thus, we can directly choose one of these two rotation directions to define the obstacle avoidance vector field $\Gamma_o$, and in this article, $\Gamma_o$ is defined by rotating $\Gamma_r$ clockwise through $\frac{\pi}{2}$, i.e., $\Gamma_o = R_{\frac{\pi}{2}} \Gamma_r$, where $R_{\frac{\pi}{2}}$ is rotation matrix given in (1) with $\theta = \frac{\pi}{2}$.

To summarize, based on abovementioned definitions under different situations, the obstacle avoidance vector field can be given by

$$\Gamma_o = \begin{cases} R_{\frac{\pi}{2}} \Gamma_r, & \theta_r = 0 \\ R_{\frac{\pi}{2}} \Gamma_r, & 0 < \theta_r < \frac{\pi}{2} \\ \Gamma_r, & \theta_r \geq \frac{\pi}{2}. \end{cases}$$

Therefore, we can present the DVF with obstacle avoidance in $\mathbb{R}^2$ as follows:

$$\Gamma_P = \begin{cases} \Gamma_d, & d > R_o \\ \Gamma_o, & r_o < d \leq R_o. \end{cases}$$

Note that the vector field given in (29) would be discontinuous at $d = R_o$. To make the vector field continuous, we introduce the following transition function:

$$\varsigma = \begin{cases} 0, & d < R_o - \epsilon \\ S(d), & R_o - \epsilon \leq d \leq R_o \\ 1, & d > R_o \end{cases}$$

where $S(d)$ is a bump function [46] valued in $[0,1]$ and $\epsilon$ is a small positive constant. It is obvious that the transition function $\varsigma$ continuously varies from 0 to 1 as the distance $d$ varies from $R_o - \epsilon$ to $R_o$. Therefore, the vector field proposed in (29) can be revised to be a continuous form, i.e.,

$$\Gamma_P = \varsigma \Gamma_d + (1 - \varsigma) \Gamma_o$$

which is the composition of $\Gamma_d$ and $\Gamma_o$. Hence, we summarize the results of obstacle avoidance as follows.

Theorem 3: Let $(x_0, y_0, \theta_0)$ denote an arbitrarily-specified final state. The DVF $\Gamma_P$ proposed in (31) asymptotically converges to $(x, y, \theta) = (x_d, y_d, \theta_d)$ and avoids the collision with the obstacle meanwhile.

Proof: The convergence of $\Gamma_d$ has been proved by Theorem 1, so that we merely need to prove that $\Gamma_o$ will not cause collision of the robot with the obstacle. According to (25)–(28), the vector field $\Gamma_o$ can be rewritten as

$$\Gamma_o = R_{\pm \frac{\pi}{2}} (p - p_o)$$

where $R_{\pm \frac{\pi}{2}}$ represents the rotation matrix with $\theta = \pm \frac{\pi}{2}$. Define the following distance function:

$$f = \|p - p_o\|^2 - \epsilon_o^2$$

then $f > 0$ holds for all $p \in D_o$. To ensure obstacle avoidance, the distance function $f$ should be guaranteed always positive. Assume that once the robot enters the obstacle’s reactive area $D_o$, the initial value of $f$ is a constant $c_0 > 0$. Taking the time derivative along the vector field $\Gamma_o$, we have

$$\frac{df}{dt} = 2(p - p_o)^T \Gamma_o = 2(p - p_o)^T R_{\pm \frac{\pi}{2}} (p - p_o) = 0$$

which demonstrates that the distance function will maintain $f = c_0 > 0$ in the future time, indicating that integral curve of the vector field $\Gamma_o$ will maintain a constant distance with respect to the obstacle, thus achieving obstacle avoidance by a circular motion.

Although Theorem 3 is derived based on only one obstacle, it can be simply extended to the case of multiple obstacles.
Regarding $M$ obstacles, the DVF is designed to be

\[ \Gamma_p = \prod_{i=1}^{M} \Gamma_d + \sum_{i=1}^{M} (1 - \zeta_i) \Gamma_{ai} \]  

(35)

where $\Gamma_d$ is the convergent DVF, $\Gamma_{ai}$ is the obstacle avoidance vector field around the $i$th obstacle, and $\zeta_i$ is the transition function regarding the $i$th obstacle.

Having obtained the vector field in (35), it would not be difficult to design the controller for the nonholonomic mobile robot. Similar to (21), we transform the vector field $\Gamma_p$ into the body-fixed frame $\mathcal{F}_b$ by

\[ \Gamma_p^b = R^T \Gamma_p \triangleq \begin{bmatrix} \Gamma_{bx}^b \\ \Gamma_{by}^b \end{bmatrix}. \]  

(36)

Then, inspired by (24), the controller can be given by

\[ v_x = k_v \Gamma_{bx}^b \]  

(37a)

\[ \omega = -k_w \sum_{i=1}^{M} \zeta_i \theta + k_a \tan^{-1} \left( \frac{\Gamma_{by}^b}{\Gamma_{bx}^b} \right) \]  

(37b)

where $k_v, k_w, k_a$ are all positive scalars.

**B. Characteristics of Obstacle Avoidance Vector Field**

In this section, we present several characteristics of the obstacle avoidance vector field. It can be seen from (31) that $\Gamma_p$ is composed of $\Gamma_d$ and $\Gamma_{ai}$, and $\Gamma_p$ possibly vanishes at certain points, which are also referred to as singular points. Thus, we analyse the existence of the singular points of $\Gamma_p$.

**Lemma 1:** The composite vector field $\Gamma_p$ defined in (31) does not have any singular points in the switching region $D_s = \{R_o - \epsilon < \|p - p_o\| < R_o\}$.

**Proof:** The composite vector field $\Gamma_p$ has singular points if and only if its norm is equivalent to 0, i.e., $\|\Gamma_p\| = 0$. The norm of $\Gamma_p$ can be computed by

\[ \|\Gamma_p\|^2 = \zeta^2 \|\Gamma_d\|^2 + (1 - \zeta)^2 \|\Gamma_{ai}\|^2 + 2\zeta(1 - \zeta)(\Gamma_d, \Gamma_{ai}) R^2. \]  

(38)

According to (28), if $\theta > \frac{\pi}{2}$, we have $\Gamma_o = \Gamma_d$. Then, $\|\Gamma_p\|$ in (38) degenerates to $\|\Gamma_p\| = \|\Gamma_d\|$. It can be observed from Theorem 2 that $\Gamma_d$ only vanishes at the destination $p_o$, while $p_d \notin D_o \cup D_s$ holds. Thus, we have $\|\Gamma_p\| = \|\Gamma_d\| \neq 0$ in $D_s$. If $\theta < \frac{\pi}{2}$, based on $\Gamma_o$ in (28), we have $\Gamma_o = \Gamma_{ai}$, and thus the vector field norm $\|\Gamma_p\|$ in (38) can be rewritten as $\|\Gamma_p\|^2 = \zeta^2 \|\Gamma_d\|^2 + (1 - \zeta)^2 \|\Gamma_{ai}\|^2 + 2\zeta(1 - \zeta)(\Gamma_d, \Gamma_{ai}) R^2$. According to the definitions in (25) and (26), the vector field $\Gamma_{ai}$ satisfies $\|\Gamma_{ai}\| = 0$ if and only if $p = p_o$, which implies the robot reaches the obstacle’s position. This is prohibited by Theorem 3, thus we will have $\|\Gamma_{ai}\| \neq 0$. Since the transition function $\zeta$ satisfies $0 < \zeta < 1$ in $D_s$, it can be derived that $\|\Gamma_p\| = 0$ if and only if $(\Gamma_d, \Gamma_{ai}) R^2 = 0$, which is yet contradictory with (27). Therefore, we conclude that $\|\Gamma_p\| \neq 0$ in $D_s$, demonstrating that the composite vector field $\Gamma_p$ has no singular points in $D_s$.

In addition to the singularity, another characteristics worth studying is the Zeno behavior [47], where the execution of a switching system undergoes an infinite number of discrete transitions in a finite length of time and the point where the Zeno behavior occurs is called Zeno point [48]. The Zeno behavior could be examined by identifying the Zeno points of the system.

According to (30) and (31), the composite vector field $\Gamma_p$ switches when crossing the following surfaces:

\[ S_{11} = \{p \in \mathbb{R}^2 : \Pi_1 \triangleq \|p - p_o\|^2 - (R_o - \epsilon)^2 = 0\} \]

\[ S_{12} = \{p \in \mathbb{R}^2 : \Pi_2 \triangleq \|p - p_o\|^2 - R_o^2 = 0\}. \]

Crossing these two surfaces indicates that the vector field $\Gamma_p$ are shaped under the effect of both $\Gamma_d$ and $\Gamma_{ai}$. If the vector field $\Gamma_p$ suffers from Zeno behavior when crossing the switching surfaces $S_{11}$ and $S_{12}$, then the robot will get stuck and chatter on $S_{11}$ and $S_{12}$ for an infinite amount of times at a finite time instant. In order to examine the robot’s behavior around the switching surfaces, we check the existence of Zeno points with respect to $S_{11}$ and $S_{12}$ in the following.

**Theorem 4:** The closed-loop trajectory of the robot does not suffer from Zeno behavior under the switching of the composite vector field $\Gamma_p$ with respect to $S_{11}$ and $S_{12}$.

**Proof:** The theorem holds, if there do not exist Zeno points in the closed-loop trajectory of the robot. We first focus on the behavior of the robot’s trajectory on either side of the switching surface $S_{11}$. Let $\tilde{p}_{11}$ and $\tilde{p}_{12}$ denote the robot’s trajectory before and after crossing the switching surface $S_{11}$, i.e., $\|p_{11}^+ - p_o\| - (R_o - \epsilon) = \tilde{d}_c$, $\|p_{11}^- - p_o\| - (R_o - \epsilon) = -\tilde{d}_c$, where $\tilde{d}_c > 0$ is sufficiently small. Let $\tilde{Z}_{i1}$ denote the set of Zeno points with respect to the switching surface $S_{11}$. Then, according to [48, Theorem 2], $\tilde{Z}_{i1}$ is given by

\[ \tilde{Z}_{11} = \{\tilde{p}_{11}(p_{11}^+), \tilde{p}_{11}(p_{11}^-), \tilde{p}_{12}(p_{11}^+), \tilde{p}_{12}(p_{11}^-) = 0\} \]

where $\tilde{p}_{11}$ and $\tilde{p}_{12}$ represent the first-order and second-order time derivatives of $\Pi_1$ along the robot’s trajectory, respectively. By computation, we obtain that $\tilde{p}_{11} = 2(p - p_o)^T p$ and $\tilde{p}_{12} = 2\|p\|^2 + 2(p - p_o)^T p$. Regarding $\tilde{p}_{11}$ and $\tilde{p}_{12}$, it has $\tilde{p}_{11}^+ = \zeta \Gamma_d + (1 - \zeta) \Gamma_o$ and $\tilde{p}_{12}^+ = \Gamma_o$, which is ensured by (31). Substituting $\tilde{p}_{11}^+$ and $\tilde{p}_{12}^+$ into $\tilde{p}_{11}$, there holds $\tilde{p}_{11}(p_{11}^+) = 2(p_{11}^+ - p_o)^T (\zeta \Gamma_d + (1 - \zeta) \Gamma_o)$ and $\tilde{p}_{11}(p_{11}^-) = 2(p_{11}^- - p_o)^T \Gamma_o$. It is obvious that $p_{11}^+ - p_o \neq 0$ and $p_{11}^- - p_o \neq 0$ since the robot cannot overlap with the obstacle. Moreover, based on the proof of Lemma 1, we have $\Gamma_d + (1 - \zeta) \Gamma_o \neq 0$ and $\Gamma_o \neq 0$. Therefore, it can be derived that $\tilde{p}_{11}(p_{11}^+) \neq 0$ and $\tilde{p}_{11}(p_{11}^-) \neq 0$. Note that there holds $\tilde{Z}_{11} \neq \emptyset$ if $\Pi_1(p_{11}^+), \Pi_1(p_{11}^-), \tilde{p}_{11}(p_{11}^+), \tilde{p}_{11}(p_{11}^-)$ are all equal to zero. We have obtained that the first-order derivatives $\Pi_1(p_{11}^+)$ and $\Pi_1(p_{11}^-)$ are nonzero. Then, no matter what values of second-order derivatives $\tilde{p}_{11}(p_{11}^+)$ and $\tilde{p}_{11}(p_{11}^-)$, it always holds $\tilde{Z}_{11} = \emptyset$, implying that there are no Zeno points with respect to the switching surface $S_{11}$. Regarding $S_{12}$, the set of Zeno points $\tilde{Z}_{12}$ is given by

\[ \tilde{Z}_{12} = \{\tilde{p}_{12}(p_{12}^+), \tilde{p}_{12}(p_{12}^-) = \tilde{p}_{12}(p_{12}^+), \tilde{p}_{12}(p_{12}^-) = 0\}. \]
It can be similarly proved that $Z_{\Pi_i} = \emptyset$. In summary, there does not exist Zeno behavior when the robot’s closed-loop trajectory goes across the switching surfaces $S_{\Pi_i}$ and $S_{\Pi_j}$. □

**Remark 4:** The points in $Z_{\Pi_i}$ ($i = 1, 2$) are named candidate Zeno points [48]. In other words, if $\tilde{p}$ is a Zeno point, then $p \in Z_{\Pi_i}$. Thus, this is a necessary but not sufficient condition for the existence of Zeno points. But we can further obtain that if $Z_{\Pi_i} = \emptyset$, then there does not exist any Zeno points, which is a sufficient condition to prove the nonexistence of Zeno behavior. As shown in the proof of Theorem 4, $Z_{\Pi_i} = \emptyset$ results from the nonsingularity of the composite vector field, as guaranteed by Lemma 1.

**V. DVF WITH COLLISION AVOIDANCE AMONG ROBOTS**

In this section, we deal with collision avoidance among multiple nonholonomic mobile robots. The collision avoidance between two robots is first taken into consideration, and then the result is extended to the multirobot case.

**A. Collision Avoidance in Two-Robot Scenario**

Motivated by the circular vector field presented in obstacle avoidance, intuitively, we can introduce a virtual obstacle between two robots so that they are able to avoid each other by avoiding the virtual obstacle. To be more specific, as shown in Fig. 4(a), regarding two robots positioned at $p_i$ and $p_j$, when there is a potential collision risk in their sensing ranges, a virtual obstacle is set on the line segment $\overline{p_ip_j}$. According to the obstacle avoidance vector field in Section IV, the robots will turn to the direction of $\Gamma_{ri}^\perp$ and $\Gamma_{rj}^\perp$, respectively, as the green vectors in Fig. 4(a), which are projected positively along $v_i$ and $v_j$, respectively. Then, two robots will follow the vectors $\Gamma_{ri}^\perp$ and $\Gamma_{rj}^\perp$ to avoid the virtual obstacle clockwise. In this way, the collision avoidance between two robots can be realized completely.

However, in Fig. 4(a), it should be noted that the velocities $v_i$ and $v_j$ lie in the different sides of the line segment $\overline{p_ip_j}$. Once $v_i$ and $v_j$ lie in the same side of $\overline{p_ip_j}$, as given in Fig. 4(b), the obstacle avoidance vectors $\Gamma_{ri}^\perp$ and $\Gamma_{rj}^\perp$, which are projected positively along $v_i$ and $v_j$ will point to the opposite rotation directions around the virtual obstacle, i.e., clockwise and anticlockwise, respectively. Then, two mobile robots will turn to such directions and cause the collision eventually. Therefore, the obstacle avoidance vector field cannot be extended to collision avoidance straightforwardly by introducing a virtual obstacle between two robots.

Based on abovementioned analysis, the key point of the collision avoidance is how to define the direction of vectors $\Gamma_{ri}^\perp$ and $\Gamma_{rj}^\perp$ so as to make them both clockwise or both anticlockwise. To solve this problem, we still consider two robots located at $p_i$ and $p_j$, and suppose that they have been into each other’s sensing range, i.e., $d_{ij} = \|p_i - p_j\| < R_s$, where $R_s = \min\{R_c, R_c\}$. In addition, we assume that the distance threshold for starting collision avoidance denoted by $R_c$ satisfies $R_c < R_s$. That is, as shown in Fig. 5, when $d_{ij} \leq R_c$, two robots are supposed to have a potential risk of collision, and the vector field should be converted from the mode of target navigation to collision avoidance.

Following the idea of virtual obstacle, we define a circular obstacle with radius $\tilde{r}$, whose position vector is given by

$$\tilde{p}_o = \frac{1}{2}(p_i + p_j).$$

(39)

Note that $\tilde{r}$ is actually the radius of the safe area for each robot. Then, $\tilde{r}$ can be regarded as the minimum safe distance for no collision. In other words, two robots will collide with each other if $d_{ij} \leq 2\tilde{r}$. Thus, the collision avoidance vector field will lie in the following area:

$$D_c = \left\{ q \in \mathbb{R}^2 \mid \tilde{r} < \|q - \tilde{p}_o\| \leq \frac{R_c}{2} \right\}$$

(40)

where $2\tilde{r} < R_c < R_s$. Let us take the robot $i$ for example. Similar to the obstacle avoidance, the repulsive vector field $\Gamma_{ri}$ can be defined by

$$\Gamma_{ri} = p_i - \tilde{p}_o.$$
Subsequently, we define the collision avoidance vector field \( \Gamma_{ci} \) for robot \( i \) as follows:

\[
(\Gamma_{ci}, \Gamma_{ri})_{\mathbb{R}^2} = 0, \quad (\Gamma_{ci}, R_2 v_i)_{\mathbb{R}^2} > 0
\]

(42)

where \( R_2 \) is rotation matrix in (1) with \( \theta = \frac{\pi}{2} \). Note that \( R_2 \) rotates anticlockwise the velocity direction \( v_i \) by \( \frac{\pi}{2} \), then the vector \( R_2 v_i \) points exactly to the \( Y^B \)-axis of the body-fixed frame \( F^B \). Thus, different from (26) in the obstacle avoidance, \( \Gamma_{ci} \) in (42) becomes the vector, which projects positively onto \( Y^B \)-axis rather than \( X^B \)-axis (i.e., the direction of \( v_i \)), as the green vectors illustrated in Fig. 5. Due to the fact that for each robot the \( Y^B \)-axis is always obtained by rotating anticlockwise \( X^B \)-axis through \( \frac{\pi}{2} \), all the robots will rotate anticlockwise to follow the vector field \( \Gamma_{ci} \) once having a potential collision risk. Therefore, the robots will move along the anticlockwise rotation direction to accomplish the collision avoidance.

According to \( \Gamma_{ci} \) given in (42), for robot \( i \), the DVF with collision avoidance in \( \mathbb{R}^2 \) can be proposed as follows:

\[
\Gamma_{Qi} = \varsigma_i \hat{\Gamma}_d + (1 - \varsigma_i) \Gamma_{ci}
\]

(43)

where the transition function \( \varsigma_i \) is obtained from (30) by replacing \( R_o \) with \( R_c \). The abovementioned results are summarized in the theorem given as follows.

**Theorem 5**: Consider the cooperative motion planning of two nonholonomic robots and let \((x_{di}, y_{di}, \theta_{di})\) denote an arbitrarily-specified final state for robot \( i \), \( i = 1, 2 \). The DVF \( \Gamma_{Qi} \) in (43) asymptotically converges to \((x, y, \theta) = (x_{di}, y_{di}, \theta_{di})\) and avoids the collision with the other robot.

**Proof**: Similar to the proof of Theorem 3, we need to prove that \( \Gamma_{Qi} \) will not cause collisions between the robots. Define the following distance function:

\[
f_{ij} = \|p_i - p_j\|^2 - (2\hat{r})^2
\]

(44)

which satisfies \( f_{ij} > 0 \) for all \( p_i, p_j \in D_c \), \( i, j = 1, 2 \), \( i \neq j \). Then, robot \( i \) and robot \( j \) are guaranteed to collide with each other, if \( f_{ij} > 0 \) always holds. Taking the time derivative along the vector fields \( \Gamma_{ci} \) and \( \Gamma_{cj} \), we have

\[
\frac{d}{dt}f_{ij} = 2(p_i - p_j)^T(\Gamma_{ci} - \Gamma_{cj})
\]

\[
= 2(p_i - p_j)^T R_2^T (p_i - \hat{p}_o - p_j + \hat{p}_o)
\]

\[
= 0
\]

(45)

where the definitions in (41) and (42) are utilized. The time derivative of \( f_{ij} \) in (45) implies that the distance between robot \( i \) and robot \( j \) will keep constant in the area \( D_c \). Note that the distance function \( f_{ij} \) satisfies \( f_{ij} > 0 \) at the initial time for collision avoidance, then \( f_{ij} > 0 \) will still hold in the future time. Hence, the robot \( i \) will not collide with robot \( j \) in the movement.

Based on the vector field \( \Gamma \) designed in (43), we can present the control inputs \( v_i \) and \( \omega_i \) further. Similar to aforementioned sections, the controller is still proposed with the aid of the vector field given in the body-fixed frame \( F^B \). Let \( \Gamma_{Qi}^B \) denote the formulation of \( \Gamma_{Qi} \) expressed into the body-fixed frame \( F^B \), which can be obtained by

\[
\Gamma_{Qi}^B = R^T \Gamma_{Qi} \triangleq \begin{bmatrix} \Gamma_{Qi}^{Bx} \\ \Gamma_{Qi}^{By} \end{bmatrix}.
\]

(46)

 Nonetheless, there are two facts worth noting in the design of \( v_{ci} \) and \( \omega_i \), which are different from the controllers in abovementioned sections. One fact is that the linear velocities of robot \( i \) and robot \( j \) should be equivalent, when these two robots move around the virtual obstacle for collision avoidance. In addition, the definitions in (39) and (42) implicitly guarantees that \( \Gamma_{ci} \) and \( \Gamma_{cj} \) have the same amplitudes, so that \( \frac{d}{dt}f_{ij} = 0 \) in (45) holds. Therefore, once the robots enter the collision avoidance field, we set the linear velocities to be a common constant denoted by \( v_c \), which can be decided according to the range of speed for real robots. The other fact is that compared to \( \delta \) defined in (23), the angle between the directions of \( v_i \) and \( \Gamma_{ci} \), as shown in Fig. 5, belongs to \([-\pi, \pi]\) instead of \([-\frac{\pi}{2}, \frac{\pi}{2}]\). This is because according to the definition in (42), the vector \( \Gamma_{ci} \) has positive projection onto \( Y^B \)-axis, which indicates that \( \Gamma_{ci} \) might be negatively projected onto \( X^B \)-axis (i.e., the direction of \( v_i \)). Thus, the angle from the direction of \( v_i \) to \( \Gamma_{ci} \), denoted by \( \delta_c \), should be decided based on the four-quadrant inverse tangent (atan2) rather than common inverse tangent (atan) as in (23).

In other words, the angle \( \delta_c \) is given by

\[
\delta_c = -\text{atan}2(\Gamma_{Qi}^{Bx}, \Gamma_{Qi}^{By}).
\]

(47)

Hence, based on abovementioned two facts, the controller of robot \( i \) can be designed as

\[
v_{xi} = k_v \varsigma_i \Gamma_{Qi}^{Bx} + (1 - \varsigma_i) v_c
\]

(48a)

\[
\omega_i = -k_\omega \varsigma_i \theta + k_a \text{atan}2(\Gamma_{Qi}^{Bx}, \Gamma_{Qi}^{By})
\]

(48b)

where \( k_v, k_\omega, k_a \) are all positive scalars, and \( \varsigma_i \) is the transition function similar to (30).

**B. Collision Avoidance in Multirobot Scenario**

In this section, we continue to investigate the collision avoidance problem for multiple robots. Partly motivated by [43], we propose a mixed vector field by computing the weighted sum of the collision avoidance vector fields of each pair of robots. We define a set \( N_i \), which represents the label set of the neighboring robots, which appear in the collision avoidance reactive area of robot \( i, i \in I_V = \{1, \ldots, N\} \). Specifically, the set \( N_i \) is given by

\[
N_i = \{j \in I_V : ||q_i - q_j|| < R_c, i \neq j\}
\]

(49)

where \( R_c \) is the radius of the collision avoidance reactive area. Based on the definition in (49), it is indicated that robot \( i \) has the risk of colliding with robot \( j, j \in N_i \), so that we propose an algorithm for robot \( i \) to avoid collision with robot \( j \).

Algorithm 1 illustrates how to design the collision avoidance vector field for robot \( i \) in the scenario of multirobot motion planning. Note that for each \( j \in N_i \), robot \( j \) and robot \( i \) compose a robot pair. Thus, similar to the case of two robots, first, we design the collision avoidance vector field for robot \( i \) with respect to each robot \( j \) by introducing an virtual obstacle as shown in
Algorithm 1: Collision Avoidance Vector Field of Multiple Nonholonomic Mobile Robots.

**Input:** position $p_i$, position $p_j$, $j \in N_i$

**Output:** collision avoidance vector field $\Gamma_{ci}$

1. for each $j \in N_i$ do
   2. $\bar{p}_{oi}^j = \frac{1}{2}(p_i + p_j)$;
   3. $\Gamma_{ri}^j = p_i - \bar{p}_{oi}^j$;
   4. define $\Gamma_{ci}^j$ by the following two conditions:
      $\langle \Gamma_{ci}^j, \Gamma_{ri}^j \rangle_{\mathbb{R}^2} = 0$ and $\langle \Gamma_{ci}^j, R_\frac{\pi}{2} v_i \rangle_{\mathbb{R}^2} > 0$;
   5. $d_{oi}^j = ||p_i - \bar{p}_{oi}^j||$;
4. end
5. $D_{oi} = \sum_j d_{oi}^j$;
6. if $D_{oi} = d_{oi}^j$ then // only one element in $N_i$
  7. $\Gamma_{ci} = \Gamma_{ci}^j$;
else
  8. for each $j \in N_i$ do
  9. $\beta_i^j = 1 - d_{oi}^j / D_{oi}$; // $\beta_i^j$ is weight for the $j$-th vector field
10. end
11. $(\beta_i^k, k) = \text{max}(\beta_i^j)$; // $k$ is the label of the maximum weight
12. $\Omega_\beta = \sum_j \beta_i^j$;
13. if $\beta_i^k > \Delta_m$ then // $\Delta_m$ is a threshold
14. $\beta_i^k = 1$;
15. $\beta_i^j = 0, j \neq k$;
16. else
17. $\beta_i^j = \beta_i^k / \Omega_\beta$;
18. end
19. $\Gamma_{ci} = \sum_j \beta_i^j \Gamma_{ci}^j$;
20. end

Fig. 5, and let $\Gamma_{ci}^j$ denote such a vector field. Then, we specify the weights, denoted by $\beta_i^j$, associated with each vector field $\Gamma_{ci}^j$. We remark that the weight $\beta_i^j$ take values in $[0,1]$ and is determined by the robot $i$ distance to the virtual obstacle located at $\bar{p}_{oi}^j$, which is the middle point between $p_i$ and $p_j$.

Roughly speaking, $\beta_i^j$ approaches to 1 as robot $i$ is getting close to the virtual obstacle associated with robot $j$. Next, we find the maximum of all the weights and let $\beta_i^k$ denote it, where $k \in N_i$ is the label of the maximum weight. In other words, $\beta_i^k \geq \beta_i^j$ holds for all $j \in N_i, j \neq k$. This indicates that robot $i$ is the closest to the virtual obstacle generated from robot $k$. If $\beta_i^k$ is larger than a predefined weight threshold $\Delta_m$, i.e., $\beta_i^k > \Delta_m$, where $0 < \Delta_m < 1$, we set $\beta_i^h = 1$ and $\beta_i^j = 0$ $(j \neq k)$. Otherwise, we normalize all the weights. Finally, the collision avoidance vector field for robot $i$, denoted by $\Gamma_{ci}$, is given by the weighted sum of all the components $\Gamma_{ci}^j$ with weights $\beta_i^j$.

Intuitively, the weight threshold $\Delta_m$ is employed to measure how close robot $i$ is to its nearest neighbor, i.e., robot $k$. Once $\beta_i^k > \Delta_m$, it implies robot $i$ is very close to robot $k$, leading to a high potential risk of collision. Then, in such a case, it is reasonable to make robot $i$ only consider the collision avoidance with respect to robot $k$ and follow the sole vector field $\Gamma_{ci}^k$.

In this way, the collision avoidance of multirobot system is converted to the scenario of two robots, which can be guaranteed by Theorem 5.

The results of collision avoidance among multiple robots are summarized in the following theorem.

**Theorem 6:** Let $\Gamma_{ci}$ denote the collision avoidance vector field of robot $i$, $i \in I_2$. If $\Gamma_{ci}$ is designed as in Algorithm 1, then robot $i$ is guaranteed not to collide with any other robot, i.e., $||p_i - p_j|| > 2\hat{r}$ for all $j \in I_2, i \neq j$, where $\hat{r}$ denotes the radius of safe area for each robot.

**Proof:** According to the definition in (49), we consider the robots in $N_i$ and $I_2 \setminus N_i$, respectively.

Concerning $j \in N_i$, $j \neq i$, the vector field for robot $i$ is designed by mixing the collision avoidance vector fields with respect to each robot $j$ using different weights. Although such a mixed vector field does not guarantee collision avoidance, it will guide robot $i$ to move toward a certain direction and finally enter the threshold $\Delta_m$ of a robot $k$. At this time instant, robot $k$ is the closest one to robot $i$. Then, the safety of robot $i$ can be ensured if robot $i$ does not collide with robot $k$. In other words, $||p_i - p_k|| > 2\hat{r}$ holds for all $j \in N_i$, if $||p_i - p_k|| > 2\hat{r}$ satisfies. This is actually the collision avoidance of two robots, and according to Algorithm 1, the vector field for robot $i$ in this case is directed given by the collision avoidance vector field with respect to robot $k$, which guarantees $||p_i - p_k|| > 2\hat{r}$ by the proof of Theorem 5.

Regarding $j \in I_2 \setminus N_i, j \neq i$, the distance between robot $i$ and robot $j$ satisfies $||p_i - p_j|| \geq R_c$. As given in (40), the reactive area is larger than the safe area, i.e., $R_c > 2\hat{r}$. Thus, $||p_i - p_j|| > 2\hat{r}$ holds.

Remark 5: In the collision avoidance of multiple robots, the number of the closest robot with respect to robot $i$ may be more than one. Thus, in Algorithm 1, there probably exist more than one maximum weights with different labels. Since the collision avoidance is finally converted to the case of two robots, we have to determine a maximum’s label $k$ to construct the robot pair.

In this case, we additionally consider the bearing angle of the line-of-sight of each robot pair, and specify robot $k$ as the robot, which forms a minimum bearing angle. For example, as shown in Fig. 6, robots $j_1$ and $j_2$ are the equally closest to robot $i$, and the line-of-sight’s bearing angle of these two robot pairs are given by $\theta_{i1}^j$ and $\theta_{i2}^j$. Since $\theta_{i2}^j < \theta_{i1}^j$, we choose robot $j_2$ to form a pair with robot $i$, and then the vector field can be designed accordingly.
**Remark 6:** The collision avoidance vector field $\Gamma_{ci}$ designed in Algorithm 1 might vanish at certain points, since $\Gamma_{ci}$ is a weighted mixture of different vector fields $\Gamma_{jci}$. Nevertheless, these critical points do not cause any collision because they are guaranteed being far away from the robots by choosing a suitable weight threshold $\Delta m$, which ensures to generate a sole-robot-dominated area around the nearest robot for collision avoidance. Moreover, the critical points of $\Gamma_{ci}$ are temporary and of measure zero, because they are caused by the perfect symmetry of the robots’ moving positions. Thus, the vector field can still guarantee collision avoidance among robots.

**Remark 7:** It probably occurs that the robot encounters obstacle avoidance and collision avoidance at the same time instant. Note that the collision avoidance in this article is converted into circumventing a virtual obstacle. Thus, if there exists a superimposed effect of obstacle avoidance and collision avoidance, it implies that the reactive areas of the real obstacle and virtual obstacle are overlapped. In this case, we can construct a larger virtual obstacle to cover the original real obstacle and virtual obstacle. Then, the robot could circumvent such a larger virtual obstacle to accomplish obstacle avoidance and collision avoidance simultaneously.

**Remark 8:** The collision avoidance algorithm is scalable with respect to the number of the robots. This is because the multirobot system does not rely on a predefined communication topology but takes measures in reactive behavior. Of course, the reactive method requires high computational efficiency, which is another characteristics of the proposed algorithm. The designed vector field serving as planning result only relies on the robot’s states, and we present in an analytical expression, so that it can be straightforwardly computed in real time as long as the robot’s states are obtained.

**VI. SIMULATION AND EXPERIMENT**

**A. Numerical Simulation**

In this section, the following four numerical simulation examples are provided, i.e.:

1) motion planning in an obstacle-free space;
2) motion planning with obstacle avoidance;
3) coordinated motion planning with collision avoidance;
4) coordinated motion planning with both obstacle and collision avoidance.

**Example 1:** The DVF (20) is employed in this example to verify the effectiveness of motion planner in an obstacle-free space. The initial and final states of the nonholonomic mobile robot are given in Table I, where we choose six different final states including positions and orientations. The control gains are set to be $k_\theta = 0.15$, $k_v = 0.15$, $k_a = 0.6$. We remark that this is an example for one robots under various requirements of final states, instead of coordinated motion planning of multiple robots. Fig. 7(a) depicts the initial and specified final states of the robot. It should be mentioned that the case of $(x_d, y_d, \theta_d) = (1, 2, 0)$ is a quite challenging one, because such a final position is in the lateral direction of the initial position and has a same attitude as the initial state, while the robot cannot move sideways. The simulation results of robot trajectories are provided in Fig. 7(b), which shows that the robot reaches the specified positions as well as the desired orientations. The control inputs of the robot, i.e., the angular velocity and linear velocity in these six different conditions are given in Fig. 8(a).

**Example 2:** This example considers the motion planning in an obstacle-cluttered environment, so that the DVF with obstacle avoidance should be utilized. The initial and final states of one
Fig. 9. Simulation results of motion planning in an obstacle environment. (a) $t = 0$ s. (b) $t = 8$ s. (c) $t = 20$ s. (d) $t = 50$ s.

Fig. 10. Simulation results of coordinated motion planning with collision avoidance (Scenario 1). (a) $t = 0$ s. (b) $t = 8$ s. (c) $t = 20$ s. (d) $t = 50$ s.

| Case no. | Initial condition $(x_0, y_0, \theta_0)$ | Specified final state $(x_f, y_f, \theta_f)$ |
|----------|----------------------------------------|------------------------------------------|
| 1        | $(2, 2, 0)$                            |                                          |
| 2        | $(0, 2, \frac{\pi}{2})$               | $(2, 0, 0)$                              |
| 3        | $(0, 0, \pi)$                          |                                          |

TABLE II
INITIAL AND FINAL STATES FOR MOTION PLANNING WITH OBSTACLE AVOIDANCE

robot is provided in Table II, where we choose three different initial conditions. The control gains are set to be $k_{\omega} = 0.15$, $k_v = 0.2$, $k_a = 0.8$. The radius of the obstacle is given by $r_o = 0.075$ m, while the radius of the region with obstacle-avoidance vector field is $R_o = 0.18$ m. The simulation time is chosen as $T = 50$ s, and the trajectories of the robot in three different scenarios are given in Fig. 9. It can be seen that the robot can arrive at the specified final states and avoid the obstacles meanwhile. In addition, the control inputs in these three different conditions are given in Fig. 8(b).

Example 3: In this example, we provide two scenarios of coordinated motion planning of multiple robots with collision avoidance. In the first scenario, six robots start from a line formation with parallel orientations, as shown in Fig. 10(a). Each robot is required to reach its own destination, including both of position and orientation, which is depicted in Fig. 10(a) by the arrows with the same color as the robot. The trajectories of the robots at different time instants are illustrated in Fig. 10(b)–(d), which indicates that the robots achieve the desired final states and avoid collisions with each other. The second scenario considers six robots on a circle, where each robot is expected to swap the position with the opposite one, and in the mean while, maintain the initial orientation at the final time instant. The initial conditions are given in Fig. 11(a), and the final states are illustrated in Fig. 11(d). Such a scenario cause possibly deadlocks for artificial potential or optimization methods, but the vector field proposed in this article is applicable to this problem, which can be observed from the simulation results in Fig. 11. The control gains in the abovementioned two scenarios are both chosen as $k_{\omega} = 0.15$, $k_v = 0.2$, $k_a = 0.8$, and the control inputs of the robots are provided in Fig. 8(c) and 8(d). We remark that in Scenario 2, the robots are evenly distributed on circle and construct a symmetry structure, so that their control inputs are completely equivalent, and the curves in Fig. 8(d) overlap each other.

Example 4: The last simulation example handles the multi-robot motion planning in an obstacle-cluttered environment. To this end, we further take into account the obstacle avoidance as well as the collision avoidance with each other. In this example, the number of the robots is $N = 10$ and the radius of the virtual obstacle is chosen as $\tilde{r} = 0.05$ m, so that safe distance between any pair of robots is $2\tilde{r} = 0.1$ m. Besides, the radius of the obstacle is chosen to be $r_o = 0.075$ m, while the radius of the reactive region with obstacle-avoidance vector field is $R_o = 0.18$ m. The control gains in this example are chosen as $k_{\omega} = 0.15$, $k_v = 0.2$, $k_a = 0.8$. The motions of all the robots at different time instants are depicted in Fig. 12, where the initial conditions of the robots are provided in Fig. 12(a) and...
the final states including desired positions and orientations are depicted in Fig. 12(d). It can be viewed from Fig. 12 that the trajectories of the robots do not overlap with obstacles, implying that obstacle avoidance is realized successfully. Moreover, the minimum distance among all pairs of robots at each time instant is shown in Fig. 13, which demonstrates that the safe threshold of collision avoidance is not violated in the robots’ motion.

B. Hardware Experiment

The hardware experiments are carried out on KK-Swarm, a platform of multiple wheeled mobile robots based on robot operating system. The localization system of the KK-Swarm is composed of the industrial cameras produced by the HIKROBOT company and the AprilTags visual fiducial system. The maximum linear velocity of each robot is 0.6 m/s. The workspace is of $2 \times 2$ m, the size of each robot is $119.75 \times 105.01 \times 79.07$ mm, and the obstacles are covered by circular regions with diameter 0.15 m.

Similar to the numerical simulation, we perform four examples or cases in the hardware experiment, i.e., single robot with/without obstacles, multiple robots with/without obstacles.

---

1More information about the platform of KK-Swarm is available at https://wiki.amovlab.com/public/misaro-doc/, and the open-source project is provided on https://github.com/kkswarm/kk-robot-swarm

2Experiment video is available at https://www.youtube.com/watch?v=HAg-YHIPA-w&list=TLPQMTMwNJtwMjOi8uRDJkTIA&index=2

---
Moreover, the initial conditions, final states and control gains in the hardware experiment are basically similar to those in the numerical simulation. For simplicity, we provide the video captures of two demos, which demonstrate the multirobot system motion planning in an obstacle-free/cluttered environment. The snapshots of these two experiments are presented in Figs. 14 and 15, and the minimum distance among all pairs of robots in two experiments is provided in Fig. 16, which shows the safety of the robots are guaranteed. Note that the black arrows in Fig. 14 represent the desired positions and orientations of the robots, like in Fig. 10(a), and the green arrows in Fig. 15 denote the initial conditions of the robots. Besides, the blue disks in Fig. 15 are obstacles in the environment. It can be observed from the experimental results that the wheeled mobile robots can be safely guided to the specified positions with desired orientations under the proposed motion planning algorithm.

VII. CONCLUSION

This article has studied the simultaneous position and orientation planning of multiple nonholonomic mobile robots. Such a planning problem takes into account the position and orientation requirements simultaneously, indicating that the robot can reach the goal point with a specified attitude angle. In contrast to the existing open-loop algorithms, we have proposed a novel global feedback motion planning method, namely, a DVF, under which the directions of velocity vectors over the 2-D plane are decided by both of position and orientation and the nonholonomic constraint can be handled. In addition, by blending with a circular vector field, the DVF has been extended to the cases of obstacle and collision avoidance.

In future research, we would like to investigate the deadlock and Zeno issues in multirobot collision avoidance, which are still open problems as mentioned in [11] and [18]. Similar to obstacle avoidance, multiple robots are possibly trapped into deadlock if the robots are too crowded. The deadlock problem is challenging to handle, considering that it is unreasonable to make assumptions for the robots’ motion to handle deadlock in multirobot systems. Besides, there probably exists Zeno phenomenon, since the nearest robot in collision avoidance algorithm may vary with different scenarios. Although Zeno phenomenon can be characterized by finding the Zeno points similarly as in obstacle avoidance, the analysis for multirobot systems is far from being trivial. Other potential future works include the vector-field-based motion planning under input saturations and external disturbances.

ACKNOWLEDGMENT

The authors would like to thank the Associate Editor and the anonymous reviewers for their insightful comments and instructive suggestions. They also would like to thank the support from Junjie Wang and Yike Qiao in the hardware experiment of nonholonomic mobile robots.

REFERENCES

[1] A. M. Bloch, Nonholonomic Mechanics and Control, 2nd ed. New York, NY, USA: Springer, 2015.
[2] F. Bullo, "Stabilization of relative equilibria for underactuated systems on riemannian manifolds," Automatica, vol. 36, no. 12, pp. 1819–1834, Dec. 2000.
[3] L. E. Kavraki, P. Svestka, J. C. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," IEEE Trans. Robot. Autom., vol. 12, no. 4, pp. 566–580, Aug. 1996.
[4] P. Lehner and A. Albu-Schaffer, "The repetition roadmap for repetitive constrained motion planning," IEEE Robot. Autom. Lett., vol. 3, no. 4, pp. 3884–3891, Oct. 2018.
[5] C. Cai and S. Ferrari, "Information-driven sensor path planning by approximate cell decomposition," IEEE Trans. Syst., Man, Cybern. B, vol. 39, no. 3, pp. 672–689, Jun. 2009.
[6] R. V. Cowlagi and P. Tsotras, "Multiresolution motion planning for autonomous agents via wavelet-based cell decompositions," IEEE Trans. Syst., Man, Cybern. B, vol. 42, no. 5, pp. 1455–1469, Oct. 2012.
[7] S. Karaman and E. Frazzoli, “Sampling-based algorithms for optimal motion planning,” Int. J. Robot. Res., vol. 30, no. 7, pp. 846–894, Jun. 2011.
[8] Y. Oh, K. Cho, Y. Choi, and S. Oh, “Chance-constrained multilayered sampling-based path planning for temporal logic-based missions,” IEEE Trans. Autom. Control, vol. 66, no. 12, pp. 5816–5829, Dec. 2021.
[9] L. Wang, A. D. Ames, and M. Egerstedt, “Safety barrier certificates for collisions-free multirobot systems,” IEEE Trans. Robot., vol. 33, no. 3, pp. 661–674, Jun. 2017.
[10] J. van den, S. J. Berg, M. Guy Lin, and D. Manocha, “Reciprocal n-body collision avoidance,” in Proc. 14th Int. Symp. Robot. Res., 2011, pp. 3–19.
[11] J. Alonso-Mora, A. Breitenmoser, M. Rüfi, P. Beardsley, and R. Siegwart, “Optimal reciprocal collision avoidance for multiple non-holonomic robots,” in Proc. 10th Int. Symp. Distrib. Auton. Robot. Syst., 2013, pp. 203–216.
[12] I. I. Hussein and A. M. Bloch, “Optimal control of underactuated nonholonomic mechanical systems,” IEEE Trans. Autom. Control, vol. 53, no. 3, pp. 668–682, Apr. 2008.
A. J. Häusler, A. Saccon, A. P. Aguiar, J. Hauser, and A. M. Pascoal, “Energy-optimal motion planning for multiple robotic vehicles with collision avoidance,” IEEE Trans. Control Syst. Technol., vol. 24, no. 3, pp. 867–883, May 2016.

G. Zhao and M. V. F. Jarreto optimal multirobot motion planning,” IEEE Trans. Autom. Control, vol. 66, no. 9, pp. 3984–3999, Sep. 2021.

A. Bloch, M. Camarinha, and L. Colombo, “Dynamic interpolation for obstacle avoidance on Riemannian manifolds,” Int. J. Control, vol. 94, no. 3, pp. 588–600, Mar. 2021.

J. Li, M. Kan, and L. Xie, “Efficient trajectory planning for multiple nonholonomic mobile robots via prioritized trajectory optimization,” IEEE Robot. Autom. Lett., vol. 6, no. 2, pp. 405–412, Apr. 2021.

V. Cichella, I. Kaminer, C. Walton, N. Hovakimyan, and A. M. Pascoal, “Optimal multivelocity motion planning using Bernstein approximants,” IEEE Trans. Autom. Control, vol. 66, no. 4, pp. 1453–1466, Apr. 2021.

G. Zhao and M. Zhu, “Scalable distributed algorithms for multirobot near-optimal motion planning,” Automatica, vol. 140, Jun. 2022, Art. no. 110241.

B. Li, Y. Ouyang, Y. Zhang, T. Acmaran, Q. Kong, and Z. Shao, “Optimal cooperative maneuver planning for multiple nonholonomic robots in a tiny environment via adaptive-scaling constrained optimization,” IEEE Robot. Autom. Lett., vol. 6, no. 2, pp. 1511–1518, Apr. 2021.

S. Ge and Y. Cui, “Dynamic motion planning for mobile robots using potential field method,” Auton. Robot., vol. 30, no. 2, pp. 207–222, Nov. 2002.

L. Huang, “Velocity planning for a mobile robot to track a moving target. A potential field approach,” Auton. Robot., vol. 57, no. 1, pp. 55–63, Jan. 2019.

L. Valbuena and H. G. Tanner, “Hybrid potential field based control of differential drive mobile robots,” J. Intell. Robot. Syst., vol. 68, no. 3/4, pp. 307–322, Dec. 2012.

C. S. Karagöz, H. I. Bozma, and D. E. Koditschek, “Coordinated navigation of multiple independent disk-shaped robots,” IEEE Trans. Robot., vol. 30, no. 6, pp. 1289–1304, Dec. 2014.

B. Kovacs, G. Szayer, F. Taji, M. Burdelis, and P. Korondi, “A novel potential field method for path planning of mobile robots by adapting animal motion attributes,” Auton. Robot., vol. 82, pp. 24–34, Aug. 2016.

Y. Tian, X. Zhu, D. Meng, X. Wang, and B. Liang, “An overall configuration planning method of continuum hyper-redundant manipulators based on improved artificial potential field method,” IEEE Robot. Autom. Lett., vol. 6, no. 3, pp. 4867–4874, Jul. 2021.

Y. Koren and J. Koren, “Potential field methods and their inherent limitations for mobile robot navigation,” in Proc. IEEE Int. Conf. Robot. Autom., Sacramento, CA, USA, 1991, pp. 1398–1404.

J.-O. Kim and P. K. Khosla, “Real-time obstacle avoidance using harmonic potential functions,” IEEE Robot. Autom. Mag., vol. 8, no. 3, pp. 338–349, Jun. 1992.

S. Garrido, L. Moreno, D. Blanco, and F. M. Monar, “Robotic motion using harmonic functions as target finite elements,” J. Intell. Robot. Syst., vol. 59, no. 1, pp. 57–73, Jul. 2010.

A. A. Masoud, “Motion planning with gamma-harmonic potential fields,” IEEE Aerosp. Electron. Syst. Mag., vol. 48, no. 4, pp. 2786–2801, Oct. 2012.

E. Rimon and D. Koditschek, “Exact robot navigation using artificial potential functions,” IEEE Robot. Autom. Mag., vol. 8, no. 5, p. 511–518, Aug. 1992.

S. G. Loizou and K. J. Kyriakopoulos, “Navigation of multiple kinematically constrained robots,” IEEE Trans. Robot., vol. 24, no. 1, pp. 221–231, Feb. 2008.

C. Li and H. G. Tanner, “Navigation functions with time-varying destination manifolds in star worlds,” IEEE Trans. Robot., vol. 30, no. 1, pp. 35–48, Feb. 2019.

S. R. Lindemann and S. M. LaValle, “Simple and efficient algorithms for computing smooth, collision-free feedback laws over given cell decompositions,” Int. J. Robot. Res., vol. 28, no. 5, pp. 600–621, May 2009.

D. Panagou, “A distributed feedback motion planning protocol for multiple unicycle agents of different classes,” IEEE Trans. Autom. Control, vol. 62, no. 3, pp. 1179–1183, Mar. 2017.

R. M. Murray, Z. Li, and S. S. Sastry, A Mathematical Introduction to Robotic Manipulation. Boca Raton, FL, USA: CRC Press, 1994.

F. Bullo and A. D. Lewis, Geometric Control of Mechanical Systems: Modeling, Analysis, and Design for Simple Mechanical Control Systems. New York, NY, USA: Springer, 2005.

F. Bullo and R. M. Murray, “Proportional derivative (PD) control on the euclidean group,” California Institute Technol., Pasadena, CA, USA, Tech. Rep. CDS 95–010, Aug. 1995.

X. He and Z. Geng, “Trajectory tracking of nonholonomic mobile robots by geometric control on special Euclidean group,” Int. J. Robust Nonlinear Control, vol. 31, no. 12, pp. 5650–5707, Aug. 2021.

M. Tayefi, Z. Geng, and X. Peng, “Coordinated tracking for multiple nonholonomic vehicles on SE(2),” Nonlinear Dyn., vol. 87, no. 1, pp. 665–675, Jan. 2017.

H. Rodríguez-Cortés and M. Velasco-Villa, “On the geometric trajectory tracking controller for the unicycle mobile robot,” Syst. Control Lett., vol. 168, Oct. 2022, Art. no. 105360.

Y. Liu, Y. Zhao, and G. Chen, “Finite-time formation tracking control for multiple vehicles: A motion planning approach,” Int. J. Robust Nonlinear Control, vol. 26, no. 14, pp. 3140–3149, Sep. 2016.

Y. A. Kapitaniyuk, A. V. Proskurnikov, and M. Cao, “A guiding vector-field algorithm for path-following control of nonholonomic mobile robots,” IEEE Trans. Control Syst. Technol., vol. 26, no. 4, pp. 1372–1385, Jul. 2018.

A. A. Masoud, “Motion planning with gamma-harmonic potential fields,” IEEE Trans. Robot., vol. 94, no. 1, pp. 96–110, Jan. 2020.

M. Tayefi and Z. Geng, “Logarithmic control, trajectory tracking, and formation for nonholonomic vehicles on Lie group SE,” Int. J. Control, vol. 92, no. 2, pp. 204–224, Feb. 2019.

W. Yao, B. Lin, B. D. O. Anderson, and M. Cao, “Guiding vector fields for following occluded paths,” IEEE Trans. Autom. Control, vol. 67, no. 8, pp. 4091–4106, Aug. 2022.

R. Fry and S. McManus, “Smooth bump functions and the geometry of banach spaces: A brief survey,” Expo. Math., vol. 20, no. 2, pp. 143–183, 2002.

M. Heymann, F. Lin, G. Meyer, and S. Resmerita, “Analysis of zero behaviors in a class of hybrid systems,” IEEE Trans. Autom. Control, vol. 50, no. 3, pp. 376–383, Mar. 2005.

F. Ceragioli, “Finite valued feedback laws and piecewise classical solutions,” Nonlinear Anal., vol. 65, no. 6, pp. 984–998, Sep. 2006.