Design and Analysis of a Potential-Based Controller for Safe Robot Navigation in Unknown GPS-Denied Environments with Strictly Convex Obstacles *

Hamed Farivarnejad^{1,*}, Spring Berman¹

^aSchool for Engineering of Matter, Transport and Energy, Arizona State University (ASU), Tempe, AZ 85287, USA

Abstract

In this paper, we propose an obstacle avoidance controller for a disk-shaped holonomic robot with double-integrator dynamics and local sensing. The control objective is for the robot to converge to a target velocity while avoiding collisions with strictly convex obstacles in an unbounded environment. We assume that the robot has no information about the location and geometry of the obstacles, has no localization capabilities, and can only measure its own velocity and its relative position vector to the closest point on any obstacles in its sensing range. We first propose a potential-based controller for the case with a single obstacle, and we prove that the robot safely navigates past the obstacle and attains the desired velocity. For the case with multiple obstacles, we propose a switching control scheme in which the robot applies the single-obstacle controller for the closest obstacle at each instant. We investigate the correctness of this switching control law and demonstrate the absence of local stable equilibrium points that would trap the robot. We validate our analytical results through simulations of a robot that uses the proposed controllers to successfully navigate through an environment with strictly convex obstacles of various shapes and sizes.

Keywords: Obstacle avoidance, Virtual potential field, Switching control system

1. Introduction

Obstacle avoidance has been a challenging topic in the control of robotic systems and has been extensively studied by researchers over the past few decades. Numerous approaches have been proposed to prevent robots from colliding with obstacles in their workspace, ranging from heuristic solutions [1] to algorithmically rigorous [2, 3, 4, 5] and/or mathematically rigorous [6, 7, 8] motion planning and control schemes. These approaches can be categorized according to their requirements on the robot's localization capabilities and prior knowledge about the environment. We first describe key developments in obstacle-avoidance control schemes, along with their requirements, and then summarize our contribution in the context of this prior work.

Many existing obstacle avoidance strategies require the robot to have global localization as well as information about the exact shapes and locations of the obstacles. One pioneering solution with these requirements, first proposed in [9] in the 1980's, uses the concept of virtual potential fields. Subsequent approaches based on potential fields include [10], which assumes an environment that contains circular obstacles with known centers and radii, and [7], in which harmonic potential fields, which satisfy Laplace's equation, are used to guarantee collision-free robot navigation to a target position on the do-

Preprint submitted to Systems and Control Letters

main boundary. The construction of potential fields called navigation functions on bounded manifolds was a significant development that enabled the design of control laws for exact robot navigation to destinations in generalized sphere worlds [8, 11]. These control laws require accurate robot localization and prior information about the locations of the obstacles and the equations of their boundaries. Numerous works have adapted the navigation function approach to different scenarios. In [12], a combination of harmonic potentials and navigation functions is proposed as a solution when the free space can be decomposed into a chain of connected polygons. In [13], a navigation function-based strategy is merged with the dynamic window approach [14] to produce faster robot convergence to a destination in dynamic environments. An algorithm for automatic tuning of the parameters of navigation functions for sphere worlds is presented in [15]. In [16], navigation functions are designed such that the robot can asymptotically track a moving target in environments with obstacles. Recently, a modified navigation function-based approach was proposed in [17] to produce robot convergence to the minimum of a globally convex potential function in an environment with arbitrary convex obstacles.

Control schemes that use *barrier certificates* [18] and *barrier functions* [19] have been recently developed for scenarios where there are unsafe or undesired regions in a dynamical system's state space that its trajectories must avoid. These methods require knowledge of the exact boundary of the unsafe or undesired region, which is the set of obstacles when the objective is collision-free robot navigation. In [20], a control barrier function scheme is proposed to prevent collisions among the robots in a swarm, and it also prevents collisions between the robots

 $^{^{\}star}$ This research was supported by ONR Young Investigator Award N00014-16-1-2605.

^{*}Corresponding author

Email addresses: hamed.farivarnejad@asu.edu (Hamed Farivarnejad), spring.berman@asu.edu (Spring Berman)

and static or dynamic obstacles. This control approach requires knowledge of the the centers and radii of the circles that virtually bound the obstacles.

Another category of work on obstacle avoidance can be characterized by the dependence of the proposed control strategies on only approximate knowledge about the locations and geometries of the obstacles. In [21], a sliding mode controller is presented for tracking the gradient of potential fields that are constructed based on the smallest circle that bounds each obstacle. In the recent work [22], a stochastic navigation function-based approach is proposed that requires a priori estimates of the obstacle geometries and locations according to a probability distribution (a belief space). It is shown that if the robot follows a stochastic approximation of the gradient of the navigation function, convergence to the desired destination and obstacle avoidance are guaranteed with a probability of one. The recent work [23] proposes a sensor-based feedback law that uses a Voronoi diagram for the environment which the robot computes online. While this approach applies to environments with unknown convex obstacles, it requires an assumption on the obstacle curvature (Assumption 2 in [23]).

Other obstacle avoidance strategies do not require prior knowledge about the obstacles, but are subject to different restrictions or rely on other available information. A modified potential field-based method is presented in [24] for the case where the target robot position is very close to one of the obstacles, and it is extended to environments with moving obstacles in [25]. Even though the proposed controller does not require any knowledge about the shapes and positions of the obstacles, it cannot eliminate all local minima in the environment that can trap the robot. In [26], a stochastic source-seeking scheme is proposed for a GPS-denied environment with a signal that is directly measurable by the robot. The robot is allowed to contact the boundaries of the environment and the obstacles and travels around these boundaries, maintaining contact with them, until it finds a feasible direction to the source of the signal in the free space.

In addition to the works described above, which focus on designing controllers with theoretical guarantees in particular types of environments, numerous other works focus on developing obstacle avoidance strategies that, while not necessarily amenable to theoretical analysis, are convenient to implement using typical sensors on physical robotic platforms. For example, visual sensing approaches for estimating the distance and velocity of nearby obstacles are described in [27] and [28] for terrestrial and aerial applications, respectively. The work [29] proposes a combination of a visual servoing control scheme and a velocity estimation algorithm for obstacle avoidance by a legged robot and an omnidirectional wheeled robot.

In this paper, we present a controller that stabilizes a holonomic finite-dimensional robot to a constant desired velocity in an unknown, unbounded environment and prevents its collision with arbitrary strictly convex obstacles, as well as its entrapment between the obstacles. Obstacle avoidance is enforced by a repulsive term in the controller that is based on the gradient of a virtual potential field. The proposed controller is suitable for applications in which it is necessary to regulate the velocity of a robot and navigate it safely through an unknown, obstacle-filled environment where precise position feedback is absent, unreliable, or not required. For example, underwater robots may lack accurate global position information via odometry or GPS, only obtaining GPS readings when they surface periodically. A multi-robot control problem that involves velocity regulation and does not require position feedback is flocking control of a group of agents [30], which may need to avoid unanticipated obstacles along their way while stabilizing their velocities and maintaining group cohesion [31].

To summarize, the novel contribution of this paper is a robot controller for velocity regulation and obstacle avoidance with all of the following properties:

- The controller does not require that the robot have exact or approximate global position information or *a priori* information about the locations, geometries, or configuration of obstacles in the environment.
- The robot has no predefined trajectory and operates autonomously with minimal capabilities: it can only measure its own velocity and its relative position vector to the closest point on any nearby obstacles within its sensing range.
- The controller has theoretical guarantees on performance; specifically, it can be proved that a robot with this controller will converge to a desired velocity without colliding with obstacles or becoming entrapped by local minima.

The organization of the paper is as follows. In Section 2, we define the problem statement and the terminology that we use throughout the paper. In Section 3, we present the structure of the proposed controller. The closed-loop dynamics of the robot with the controller are analyzed for the case of environments with a single obstacle in Section 4. In Section 5, the controller design and analysis are extended to the case of environments with multiple obstacles. Numerical simulation of the robot's motion with the proposed controller is given in Section 6. Finally, the paper is concluded in Section 7.

2. Problem Statement

We consider a disk-shaped holonomic robot that moves in a planar unbounded domain with second-order dynamics (a double-integrator model), $\ddot{q} = u$, where $q = (x, y)^T \in \mathbb{R}^2$ denotes the position of the robot's center in a global reference frame and $u \in \mathbb{R}^2$ is the robot's control input. A physical realization of such a robot is an omnidirectional mobile robot that can move in any direction in a plane at each time instant [32]. We assume that the domain contains multiple strictly convex obstacles. The control objective is for the robot to attain a desired velocity v_{des} while avoiding collisions with the obstacles. The *x*-axis of the global reference frame is defined along the direction of v_{des} , without loss of generality. We now define two terms that we will frequently use throughout the paper.

Definition 2.1. *The line from the robot's current position q that is normal to the obstacle's boundary intersects the boundary*

at the **projection point**. This point and its position vector are denoted by P and q_p , respectively, as shown in Fig. 1.

Definition 2.2. The vector $\mathbf{q} - \mathbf{q}_p$ from the projection point to the robot's current position is called the **collision vector**. This vector is denoted by \mathbf{d} and is shown in red in Fig. 1.

We make the following assumptions about the robot's specifications and capabilities. The robot has a circular shape with radius r. It does not have global position information (e.g., GPS) and has no prior knowledge of the obstacles' locations and shapes. The only information provided to the robot is the target velocity v_{des} . The robot can measure its own velocity, for example by using tachometers or a velocity estimation algorithm based on optical flow [33]. It can measure its heading in the global frame, e.g., using a compass.¹ It can also identify the boundaries of nearby obstacles within its local sensing range, which is assumed to be a circle with radius δ_c . We assume that at each time instant, the robot can measure its distance from each obstacle within its sensing range, for example, using infrared sensors or LIDAR. This distance is the length of the collision vector d, according to the *Projection Theorem* in [34]. We also assume that the robot can measure the angle ϕ_d of the vector -d in its body-fixed frame, e.g. using LIDAR. By adding $\phi_d + \pi$ rad to the robot's heading in the global frame, the robot can obtain the angle of d in the global frame, which we denote by θ_d . This angle is required in the proposed control law described in Section 3.

Given this minimal and completely local information, we first seek a control law that can solve the following problem.

Problem 2.3. We consider an unbounded domain that contains a single strictly convex obstacle with an arbitrary boundary ² described by $\beta(x, y) = 0$, where $\beta : \mathbb{R}^2 \mapsto \mathbb{R}$ is at least twice continuously differentiable.

We design a robot control law that uses only the local measurements available to the robot to ensure that the robot:

- (1) asymptotically converges to the desired velocity v_{des} ,
- (2) does not collide with the obstacle, and
- (3) is never trapped in a neighborhood of the obstacle.

After designing a control law that solves this problem, we consider an unbounded environment with multiple strictly convex obstacles, in which the following assumption is satisfied.

Assumption 2.4. We define the closest pair of obstacles in the environment as the two obstacles with the shortest distance between their boundaries. We assume that this distance is larger than the diameter 2r of the robot.

We confirm that the controller proposed for Problem 2.3 guarantees the properties described in the following problem.



Fig. 1: A schematic representation of the robot, an obstacle, the projection point, the collision vector, a virtual potential field constructed by the robot, and the associated global reference frame.

Problem 2.5. We consider an unbounded domain that contains a finite number m > 1 of strictly convex obstacles with arbitrary boundaries described by $\beta_i(x, y) = 0$, where each $\beta_i : \mathbb{R}^2 \mapsto \mathbb{R}, i \in \{1, ..., m\}$ is at least twice continuously differentiable. We assume that Assumption 2.4 about the distance between obstacles in the domain is satisfied. During its motion, the robot implements the control law designed to solve Problem 2.3 for the obstacle that is closest to its current position. We confirm that this control law ensures that the robot:

- (1) asymptotically converges to the desired velocity v_{des} ,
- (2) does not collide with any obstacle, and
- (3) is never trapped by any set of obstacles.

3. Controller Design

The proposed control law is a combination of a regulatory term, which stabilizes the robot's velocity to v_{des} , and a repulsive term that is based on the gradient of a virtual potential field.

3.1. Definition of the virtual potential field

The robot constructs a virtual potential field φ around the point *P*. This field is designed to satisfy four properties:

(i) φ is only a function of $\delta := (||\boldsymbol{d}|| - r) \in \mathbb{R}_{>0}$, the distance between the robot's boundary and the point \boldsymbol{q}_P .

(ii) $\varphi(\delta) \to \infty$ as $\delta \to 0$.

(iii)
$$\frac{d}{d\delta}\varphi(\delta) \to \infty$$
 as $\delta \to 0$.

(iv) $\varphi(\delta)$ and $\frac{d}{d\delta}\varphi(\delta)$ decrease monotonically to 0 as $\delta \to \delta_c$, and equal zero when $\delta \ge \delta_c$.

Note that by property (i), the potential field has circular level sets around q_P , as shown in Fig. 1.

To this end, we define the potential field as follows:

$$\varphi(\delta) = \begin{cases} p\frac{\delta}{\delta_c} + \left(\frac{\delta_c}{\delta}\right)^p - (p+1), & 0 < \delta \le \delta_c, \\ 0, & \delta_c < \delta, \end{cases}$$
(1)

¹Note that the robot's ability to measure its heading (or orientation in general) does not contradict the assumption that it lacks global position information. The orientation of a mobile robot is often measured by its on-board sensors, such as a compass, IMU, or gyroscope, and not necessarily by an external localization system.

²The assumption that the obstacle is *strictly* convex excludes the possibility that its boundary contains straight segments.

where *p* is a strictly positive real constant. We can easily confirm that the function in Eq. (1) has properties (i)-(iii). The function also satisfies property (iv), and is therefore continuous and differentiable for every $\delta \in \mathbb{R}_{>0}$. This potential field may introduce extremely large forces into the control law that exceed the saturation limits of the actuators when the robot moves very close to the obstacle's boundary. However, as described in Section 4.3, we can enforce an upper bound on the actuation forces if we impose a limit on the robot's speed.

3.2. Robot control law

The proposed control law is the following combination of a stabilizing term and a repulsive term:

$$\boldsymbol{u} = -\boldsymbol{K}(\dot{\boldsymbol{q}} - \boldsymbol{v}_{des}) - \boldsymbol{K}_R \nabla_d \varphi(\delta), \qquad (2)$$

in which K = kI and $K_R = k_R I$, where k, k_R are positive gains and $I \in \mathbb{R}^{2\times 2}$ is the identity matrix, and $\nabla_d \varphi(\delta)$ is the gradient of the potential field with respect to d.

Remark 3.1. The gradient of φ with respect to **d** can be written as:

$$\nabla_{d}\varphi(\delta) = \frac{d\varphi}{d\delta} \boldsymbol{e}_{d} = \begin{cases} \frac{p}{\delta_{c}} \left(1 - \left(\frac{\delta_{c}}{\delta}\right)^{p}\right) \boldsymbol{e}_{d}, & 0 < \delta \le \delta_{c} \\ \boldsymbol{0}, & \delta_{c} < \delta \end{cases}$$
(3)

where e_d is the unit vector along d. The calculation of $\nabla_d \varphi$ is provided in Appendix A. Since p and δ_c are known parameters, and we assume that the robot can measure δ and the direction of d (see Section 2), the robot can therefore calculate $\nabla_d \varphi(\delta)$ using only these local measurements.

Remark 3.2. We emphasize that the control law in Eq. (2) relies solely on local measurements: the robot's velocity \dot{q} and the magnitude and direction of the collision vector d, all of which can be measured by sensors on-board the robot. The robot does not need global position information or knowledge about the locations and geometric properties of the obstacles.

4. Analysis of Robot Dynamics for Single-Obstacle Case

We now investigate the robot's closed-loop dynamics with the control law in Eq. (2) and prove that this control law achieves the three objectives described in Problem 2.3. First, we define four terms that will be used in our analysis.

Definition 4.1. The *free space* is the subset of the domain that excludes the obstacle's boundary and interior.

Definition 4.2. The obstacle's front area is the subset of the free space in which $\delta \in (0, \delta_c]$ and $v_{des}^T \nabla_d \varphi \ge 0$.

Definition 4.3. The obstacle's back area is the subset of the free space in which $\delta \in (0, \delta_c]$ and $v_{des}^T \nabla_d \varphi < 0$.

Definition 4.4. The safe area is the subset of the free space that excludes the obstacle's front and back areas.

The areas defined above are illustrated for an arbitrary strictly convex obstacle in Fig. 2.



Fig. 2: Illustration of an obstacle's *front* and *back* areas as well as the *safe* area. The dashed orange lines are parallel to the direction of the desired velocity, and the solid blue lines are normal to the dashed orange lines. The blue arrows illustrate the gradient of the potential field.

4.1. Velocity convergence analysis

We can write the closed-loop dynamics of the robot with the proposed control law as:

$$\ddot{\boldsymbol{q}} + \boldsymbol{K}(\dot{\boldsymbol{q}} - \boldsymbol{v}_{des}) + \boldsymbol{K}_R \nabla_d \varphi(\delta) = \boldsymbol{0}.$$
(4)

By setting $\ddot{q} = \dot{q} = 0$ in Eq. (4), we obtain $Kv_{des} = K_R \nabla_d \varphi(\delta)$. Since the obstacle is *strictly* convex, given any two distinct points at the same distance δ from the obstacle, the direction of the gradient $\nabla_d \varphi(\delta)$ (i.e., the direction of e_d) at these points cannot be identical. Also, we can confirm that $\frac{d\varphi}{d\delta}$ in Eq. (3) is a strictly monotonic function for $\delta \in (0, \delta_c)$. Thus, the equation $Kv_{des} = K_R \nabla_d \varphi(\delta)$ has unique solutions for δ and e_d , which we denote by δ_e and e_{d_e} , respectively. Consequently, Eq. (4) has a unique equilibrium point at which

$$\delta_e = \delta_c \left(1 + \frac{k \left\| \mathbf{v}_{des} \right\|}{pk_R} \right)^{-1/(p+1)}, \quad \boldsymbol{e}_{\boldsymbol{d}_e} = -\boldsymbol{e}_{\boldsymbol{v}_{des}}, \tag{5}$$

where $e_{v_{des}}$ is the unit vector along v_{des} . We can check that $\delta_e \in (0, \delta_c)$. The repulsive vector field $-\nabla_d \varphi$ has a component in the opposite direction of v_{des} everywhere in the front area of the obstacle, and it has a component in the same direction as v_{des} everywhere in the back area. Thus, the position where the term $K_R \nabla_d \varphi$ negates the term $-K v_{des}$ in Eq. (4) must be in the front area, and so the equilibrium where the robot stops at a distance δ_e from the obstacle must be in this area (see Fig. 3).

Eq. (4) also has an invariant set \mathcal{E} that is defined as

$$\mathcal{E} = \left\{ \boldsymbol{q} \in \mathbb{R}^2 , \ \dot{\boldsymbol{q}} \in \mathbb{R}^2 \mid \dot{\boldsymbol{q}} = \boldsymbol{v}_{des} , \ \nabla_d \varphi(\delta) = \boldsymbol{0} \right\}.$$
(6)

From Eq. (3), $\nabla_d \varphi = \mathbf{0}$ implies that $\delta \ge \delta_c$. This invariant set has no intersection with the obstacle's front area, since asymptotic convergence to the desired velocity \mathbf{v}_{des} and a monotonic decrease in the value of the potential field $\varphi(\delta)$ as $\delta \to \delta_c$ (property (iv) of φ) cannot occur simultaneously in the front area.

The stability characteristics of the unique equilibrium point in Eq. (5) and the invariant set in Eq. (6) are discussed in the next two theorems.



Fig. 3: Illustration of repulsive vector field $-\nabla_d \varphi$ (red arrows) and the unique equilibrium point at distance δ_e from the obstacle, given by Eq. (5).

Theorem 4.5. Consider the unique equilibrium point of Eq. (4) for which the robot is stationary at distance δ_e , given in Eq. (5), from the boundary of the obstacle in the obstacle's front area. This equilibrium is a saddle point.

Proof. We use Lyapunov's indirect method to investigate the stability properties of this equilibrium. Toward this end, we define the state vector as $X = (q^T, \dot{q}^T)^T \in \mathbb{R}^4$ and linearize Eq. (4) about the equilibrium point, obtaining the following equation:

$$\dot{X} = \begin{bmatrix} \mathbf{0}_{2\times 2} & \mathbf{I}_{2\times 2} \\ -\mathbf{K}_R \frac{\partial}{\partial q} \nabla_d \varphi(\delta) & -\mathbf{K} \end{bmatrix} \mathbf{X}, \tag{7}$$

where $\delta \equiv \delta_e$. By Lemma 3.5 in [35], we know that the linear system in Eq. (7) has the stability properties of the system described by $\dot{q} = -\frac{\partial}{\partial q} \nabla_d \varphi(\delta_e) q$. In addition, since $q = d + q_P$ and $\varphi = \varphi(\delta)$, where $\delta = (||d|| - r)$, we can show that $\nabla_d \varphi = \nabla_q \varphi$ (this equation is similar to Equation (7) in [30] and is proved in Appendix A). Therefore, we have that $\dot{q} = -\frac{\partial}{\partial q} \nabla_d \varphi(\delta_e) q = -\nabla^2 \varphi(\delta_e) q$, where $\nabla^2 \varphi(\delta_e)$ is the Hessian of φ at the equilibrium point. The stability properties of Eq. (7) are thus determined by the eigenvalues of $\nabla^2 \varphi(\delta_e)$, which we characterize in the following lemma.

Lemma 4.6. The determinant of $\nabla^2 \varphi(\delta)$ is strictly negative for all points $\boldsymbol{q} \in \mathbb{R}^2$ such that $\delta \in (0, \delta_c)$, and consequently, the eigenvalues of $\nabla^2 \varphi(\delta)$ have opposite signs.

Proof. The Hessian of φ can be calculated as

$$\nabla^2 \varphi(\delta) = \frac{\partial}{\partial q} \left(\nabla_d \varphi(\delta) \right) = \frac{\partial}{\partial q} \left(\varphi'(\delta) e_d \right), \tag{8}$$

where $\varphi'(\delta) = \frac{d\varphi}{d\delta}$. Applying the fact that $\nabla_d \varphi = \nabla_q \varphi$, Eq. (8) can be written as

$$\nabla^2 \varphi(\delta) = \varphi''(\delta) \boldsymbol{e}_d \boldsymbol{e}_d^T + \varphi'(\delta) \left(\frac{\partial \boldsymbol{e}_d}{\partial \boldsymbol{q}}\right), \tag{9}$$

where $\varphi''(\delta) = \frac{d^2\varphi}{d\delta^2}$. By the chain rule, the partial derivative in Eq. (9) can be expressed as

$$\frac{\partial \boldsymbol{e}_d}{\partial \boldsymbol{q}} = \frac{\partial \boldsymbol{e}_d}{\partial \boldsymbol{d}} \frac{\partial \boldsymbol{d}}{\partial \boldsymbol{q}}.$$
 (10)

Since $d = q - q_P$, we have that $\frac{\partial d}{\partial q} = I$. Also, given that $e_d = [\cos(\theta_d) \sin(\theta_d)]^T$, we can confirm that

$$\frac{\partial \boldsymbol{e_d}}{\partial \boldsymbol{d}} = \frac{1}{\delta} \begin{bmatrix} \sin(\theta_d)^2 & -\cos(\theta_d)\sin(\theta_d) \\ -\cos(\theta_d)\sin(\theta_d) & \cos(\theta_d)^2 \end{bmatrix}.$$
 (11)

Using Eqs. (10) and (11), Eq. (9) can be rewritten as

$$\nabla^{2}\varphi(\delta) = \varphi''(\delta) \begin{bmatrix} \cos(\theta_{d})^{2} & \cos(\theta_{d})\sin(\theta_{d}) \\ \cos(\theta_{d})\sin(\theta_{d}) & \sin(\theta_{d})^{2} \end{bmatrix} \\ + \frac{\varphi'(\delta)}{\delta} \begin{bmatrix} \sin(\theta_{d})^{2} & -\cos(\theta_{d})\sin(\theta_{d}) \\ -\cos(\theta_{d})\sin(\theta_{d}) & \cos(\theta_{d})^{2} \end{bmatrix}.$$
(12)

Then, we can express the determinant of the Hessian as

$$\det(\nabla^2 \varphi(\delta)) = \frac{1}{\delta} \varphi'(\delta) \varphi''(\delta).$$
(13)

We can determine from Eq. (1) that $\varphi'(\delta)$ and $\varphi''(\delta)$ are strictly negative and strictly positive, respectively, for $\delta \in (0, \delta_c)$. Hence, det($\nabla^2 \varphi(\delta)$) is strictly negative for any point \boldsymbol{q} that is at a distance $\delta \in (0, \delta_c)$ from a strictly convex obstacle.

Since det($\nabla^2 \varphi$) is strictly negative, the eigenvalues of $\nabla^2 \varphi$ are both non-zero and have opposite signs, and consequently the equilibrium of the system described by $\dot{q} = -\nabla^2 \varphi(\delta_e) q$ is a saddle point. As explained in the text preceding Lemma 4.6, this implies that the equilibrium of the system in Eq. (7) is also a saddle point. Therefore, the equilibrium of Eq. (4) for which the robot is stationary at distance δ_e from the obstacle is a saddle point. We can thus conclude that the robot can only reach this equilibrium if its initial position is in a set of measure zero. In practice, the robot will not be initialized precisely in this set, and so it will never stop at the location of the saddle point.

Theorem 4.7. The invariant set \mathcal{E} described in Eq. (6) is locally asymptotically stable, and the obstacle's back area is a subset of its basin of attraction.

Proof. We cannot use Lyapunov's indirect method to study the stability of the invariant set \mathcal{E} due to the following argument. In the set \mathcal{E} , we have that $\delta \ge \delta_c$, and consequently $\varphi(\delta) = 0$ and $d\varphi(\delta)/d\delta = 0$ in this set. The linearization of Eq. (4) about each point in the set \mathcal{E} is given by Eq. (7) with $\delta \ge \delta_c$. As a result, the first two columns of the matrix in Eq. (7) are both columns of zeros, and therefore the matrix has two zero eigenvalues. Thus, we cannot determine the stability characteristics of the closed-loop system from its linearization [36].

Instead, we use LaSalle's invariance principle for this case. Toward this end, we define the velocity error $s = \dot{q} - v_{des}$. Since the desired velocity is constant, we have that $\ddot{q} = \dot{s}$. Then, the closed-loop dynamics in Eq. (4) can be rewritten as:

$$\dot{s} + Ks + K_R \nabla_d \varphi = \mathbf{0}. \tag{14}$$

We consider the following Lyapunov function:

$$V = \frac{1}{2}\boldsymbol{s}^{T}\boldsymbol{s} + k_{R}\varphi(\delta).$$
(15)

This function is positive over the entire state space and equals zero at each point in the set \mathcal{E} , since s = 0 and $\varphi(\delta) = 0$ in this set. The time derivative of this function is:

$$\dot{V} = -\mathbf{s}^T \mathbf{K} \mathbf{s} - \mathbf{s}^T \mathbf{K}_R \nabla_d \varphi + k_R (\nabla_d \varphi)^T \dot{\mathbf{d}}, \qquad (16)$$

where we have written $\dot{\varphi} = (\nabla_d \varphi)^T \dot{d}$ in the last term using the chain rule. To simplify Eq. (16), we write \dot{d} as $\dot{d} = \dot{q} - \dot{q}_P$, in which \dot{q}_P is the time derivative of the position of the projection point (see Fig. 1). Since the projection point is constrained to move along the boundary of the obstacle, its velocity \dot{q}_P is always tangent to this boundary. Moreover, the gradient of the potential field φ is normal to the boundary. Hence, we can conclude that $(\nabla_d \varphi)^T \dot{q}_P = 0$. Thus, using the relation $s = \dot{q} - v_{des}$, Eq. (16) is simplified to:

$$\dot{V} = -\mathbf{s}^T \mathbf{K} \mathbf{s} + k_R \mathbf{v}_{des}^T \nabla_d \varphi.$$
(17)

As stated in Definition 4.3, the second term on the right-hand side of this equation is negative in the obstacle's back area, and thus \dot{V} is negative definite over the entire back area of the obstacle. This means that the invariant set \mathcal{E} is locally asymptotically stable, and a set defined as $\Omega := \{X \in \mathbb{R}^4 \mid V \le c, c > 0\}$, which contains the obstacle's back area, is the simplest estimate of the basin of attraction for \mathcal{E} . The set Ω consists of all trajectories with a bounded initial velocity that start in or enter the obstacle's back area.

4.2. Collision avoidance analysis

We now prove that the robot will never collide with the obstacle in either its front area or back area. For the case where the robot is in the back area, the following corollary from Theorem 4.7 ensures collision avoidance:

Corollary 4.8. Since \dot{V} is negative everywhere in the obstacle's back area, V can never become unbounded in this area. This implies that φ never blows up to infinity in the back area. Hence, δ never approaches zero in this region, meaning that the robot never collides with the obstacle when it is in the back area.

Next, we analyze the case where the robot is in the obstacle's front area. For this purpose, we study the dynamics of the robot in a different coordinate system, illustrated in Fig. 4. Note that the vectors denoted by *e* in the figure are unit vectors. First, we decompose the robot's velocity \dot{q} into the sum of \dot{q}_P , the velocity of the projection point on the obstacle's boundary, and \dot{d} , the robot's velocity relative to the projection point. We describe \dot{q}_P in a tangential-normal coordinate system [37], in which $\xi \in \mathbb{R}$ denotes the scalar displacement of the projection point along the obstacle's boundary, and $\rho \in \mathbb{R}_{>0}$ denotes the radius of curvature of the boundary. In addition, we describe \dot{d} in a polar coordinate system [37], in which (as defined previously) $\delta = (||\mathbf{d}|| - r) \in \mathbb{R}_{\geq 0}$ is the distance between the robot's boundary and the projection point, and $\theta_d \in [-\pi, \pi]$ rad is the angle of the vector d in the global reference frame. Using the facts that $\dot{q} = \dot{q}_P + \dot{d}$, $e_{\delta} = e_d$, and \dot{q}_P is always tangent to the obstacle's boundary, the robot's velocity can be written in the new coordinate system as:

$$\dot{\boldsymbol{q}} = (\dot{\delta})\boldsymbol{e}_{\delta} + \left((\delta + r)\dot{\theta}_{d}\right)\boldsymbol{e}_{\theta_{d}} + (\dot{\xi})\boldsymbol{e}_{t}.$$
(18)



Fig. 4: Illustration of the coordinate systems used to derive Eqs. (20) and (21).

Therefore, the robot's acceleration is:

$$\ddot{\boldsymbol{q}} = \left(\ddot{\boldsymbol{\delta}} - (\boldsymbol{\delta} + r)\dot{\boldsymbol{\theta}}_{d}^{2}\right)\boldsymbol{e}_{\boldsymbol{\delta}} + \left((\boldsymbol{\delta} + r)\ddot{\boldsymbol{\theta}}_{d} + 2\dot{\boldsymbol{\delta}}\dot{\boldsymbol{\theta}}_{d}\right)\boldsymbol{e}_{\boldsymbol{\theta}_{d}} + (\ddot{\boldsymbol{\xi}})\boldsymbol{e}_{t} + \left(\frac{\boldsymbol{\xi}^{2}}{\rho}\right)\boldsymbol{e}_{n}.$$
(19)

Moreover, since the collision vector d always points in the direction of the normal to the boundary, we can conclude that $e_t = e_{\theta_d}$ and $e_n = -e_{\delta}$. Substituting the expressions for \dot{q} and \ddot{q} defined in Eq. (18) and Eq. (19) into Eq. (4), we can write the resulting equations of motion along the e_{δ} and e_{θ_d} directions as Eq. (20) and Eq. (21), respectively:

$$\ddot{\delta} - (\delta + r)\dot{\theta}_d^2 - \frac{\dot{\xi}^2}{\rho} + k\dot{\delta} + k_R\varphi'(\delta) - kv_{des}\cos(\theta_d) = 0, \quad (20)$$
$$(\delta + r)\ddot{\theta}_d + 2\dot{\delta}\dot{\theta}_d + \ddot{\xi} + k(\delta + r)\dot{\theta}_d + k\dot{\xi} + kv_{des}\sin(\theta_d) = 0, \quad (21)$$

in which $v_{des} = ||\mathbf{v}_{des}||$. Note that the repulsive force $-\varphi'(\delta)\mathbf{e}_d$ shows up only in Eq. (20), since $\mathbf{e}_d = \mathbf{e}_{\delta}$.

Remark 4.9. The robot is in the obstacle's back area when $\cos(\theta_d) > 0$, and it is in the front area when $\cos(\theta_d) \le 0$.

The next theorem uses Eq. (20) and Eq. (21) to prove that the robot will never collide with the obstacle in its front area.

Theorem 4.10. If the robot's trajectory starts anywhere in the obstacle's front area, then the robot will never collide with the obstacle's boundary in this region.

Proof. We consider the following function:

$$W = \frac{1}{2} \left(\dot{\delta}^2 + \left((\delta + r) \dot{\theta}_d + \dot{\xi} \right)^2 \right) + k_R \varphi(\delta) - k v_{des} \left((\delta + r) \cos(\theta_d) - \int_0^{\theta_d} \rho \sin(\sigma) d\sigma \right).$$
(22)

To confirm that this function is positive over all θ_d in the obstacle's front area, i.e. $\theta_d \in [-\pi, -\pi/2] \cup [\pi/2, \pi]$ rad, we only have to prove that $(\delta \cos(\theta_d) - \int_0^{\theta_d} \rho \sin(\sigma) d\sigma) \le 0$ for all θ_d in this set. From Remark 4.9, we see that the term $\delta \cos(\theta_d) \le 0$ for all θ_d in the front area. In addition, by Definition 6.2 in [36], the integral $\int_0^{\theta_d} \rho \sin(\sigma) d\sigma \ge 0$ for any $\theta_d \in [-\pi, \pi]$ rad, since ρ is always positive and $\sin(\theta_d)$ belongs to the sector $[0, \pi/4]$ for $\theta_d \in [-\pi, \pi]$ rad (and therefore for $\theta_d \in [-\pi, -\pi/2] \cup [\pi/2, \pi]$

rad). Therefore, $(\delta \cos(\theta_d) - \int_0^{\theta_d} \rho \sin(\sigma) d\sigma) \le 0$ for all θ_d in the front area, and hence *W* is positive over such θ_d . We note that we cannot derive a closed-form solution for the integral $\int_0^{\theta_d} \rho \sin(\sigma) d\sigma$, since ρ changes with θ_d for arbitrary strictly convex obstacles, and the obstacle shape in our scenario is unknown.

The time derivative of the function W along the trajectories of the system in Eq. (20) and Eq. (21) is given by:

$$\dot{W} = -k\left(\dot{\delta}^{2} + (\delta + r)^{2}\dot{\theta}_{d}^{2}\right) + \dot{\delta}\frac{\dot{\xi}^{2}}{\rho} - \dot{\xi}\dot{\delta}\dot{\theta}_{d} - kv_{des}\left(\dot{\xi}\sin(\theta_{d}) - \frac{d}{dt}\int_{0}^{\theta_{d}}\rho\sin(\sigma)d\sigma\right).$$
(23)

We define β as the angle of the direction of the normal to the boundary in the global reference frame. For an infinitesimal change in the projection point displacement ξ , we have that $d\xi = \rho d\beta$, where the radius of curvature ρ is approximated as constant. This implies that $\dot{\xi} = \rho \dot{\beta}$. Moreover, since e_{δ} is always normal to the boundary, we can conclude that $d\theta_d = d\beta$, and consequently, $\dot{\beta} = \dot{\theta}_d$. Using the relation $\dot{\xi} = \rho \dot{\theta}_d$, we can reduce Eq. (23) to the following expression:

$$\dot{W} = -k\left(\dot{\delta}^2 + (\delta + r)^2\dot{\theta}_d^2\right) - kv_{des}\left(\rho\dot{\theta}_d\sin(\theta_d) - \frac{d}{dt}\int_0^{\theta_d}\rho\sin(\sigma)d\sigma\right).$$
(24)

We now define $g(\theta_d) := \int_0^{\theta_d} \rho \sin(\sigma) d\sigma$. By the chain rule, the time derivative of $g(\theta_d)$ can be written as $\frac{d}{dt}g(\theta_d) = \frac{d}{d\theta_d}g(\theta_d)\dot{\theta}_d$, where $\frac{d}{d\theta_d}g(\theta_d) = \rho \sin(\theta_d)$. This leads to the cancellation of the two terms in the second set of parentheses in Eq. (24), and \dot{W} is simplified to:

$$\dot{W} = -k\left(\dot{\delta}^2 + (\delta + r)^2 \dot{\theta}_d^2\right),\tag{25}$$

from which we can conclude that $\dot{W} \leq 0$. Therefore, W never becomes unbounded in the front area, which implies that φ remains bounded in this region. By property (ii) of φ , this shows that the distance δ never approaches zero in the front area, and hence the robot never collides with the obstacle in this region.

The next theorem, which addresses the evolution of robot trajectories that begin in the obstacle's front area, completes our analysis of collision avoidance.

Theorem 4.11. Almost all robot trajectories that start in the obstacle's front area will eventually leave this region and enter the back area or the safe area.

Proof. From Theorem 4.5, the only equilibrium point in the front area is a saddle, which does not attract any trajectories in this area except for trajectories that start in a particular set of measure zero. Furthermore, since there are no other equilibria in the front area, we can apply the *Index Lemma* in [36] to Eq. (14) and conclude that there is no limit cycle in this area as well. Therefore, the unstable trajectories in the front area, which emanate from the saddle point, must cross into the back area. By Theorem 4.7, there exists an asymptotically stable invariant set in the back area that attracts these trajectories.

4.3. A bound on the repulsive term in the control input

As stated in property (iii) in Section 3.1, the derivative of the potential field φ goes to infinity when the robot's distance δ from the obstacle approaches zero. Thus, when the robot is very close to the obstacle (i.e., δ is small), the repulsive term in the control input Eq. (2) could become too large to implement in practice. In this section, we establish an upper bound on this term by incorporating realistic constraints on the robot's initial velocity and sensing range.

We consider the line that is parallel to the direction of the desired velocity and passes through the saddle equilibrium point, as shown in Fig. 3. If the robot's initial position is located on this line, and its initial velocity is parallel to this line, then both the velocity stabilizing force $-K(\dot{q}-v_{des})$ and the repulsive force will be along this line at the beginning of its motion and for all future time, since there will be no other vector fields to drive the robot off this direction. This leads to one-dimensional motion of the robot along this line. Moreover, for a given initial robot speed, the velocity stabilizing force has the largest component that directly opposes the repulsive force when the robot is on this line, compared to when it is anywhere else in the obstacle's front area. Thus, the minimum feasible value for δ in the front area is achieved on this line.

When the robot moves only along this line in the front area, we have that $\ddot{\theta}_d$, $\dot{\theta}_d$, $\ddot{\xi}$, $\dot{\xi} = 0$ and $\theta_d = \pi$. Substituting these values into Eqs. (20) and (21), the equation of the robot's onedimensional motion along this line in the front area is given by:

$$\ddot{\delta} + k\dot{\delta} + k_R\varphi'(\delta) + kv_{des} = 0.$$
⁽²⁶⁾

Theorem 4.13 below proves the existence of a lower bound on δ by comparing the time response of Eq. (26), which is denoted by $\delta(t)$, to that of the following equation,

$$\ddot{\varrho} + k\dot{\varrho} = h(\varrho), \tag{27}$$

where

$$h(\varrho) := -a\varrho + b, \quad a, b \in \mathbb{R}_{>0}.$$
(28)

We first state the following Lemma, which describes the conditions that guarantee the existence of a positive lower bound for the time response of Eq. (27), denoted by $\rho(t)$.

Lemma 4.12. Given $\varrho_0 := \varrho(0) \in \mathbb{R}_{>0}^3$ and $w_0 := \dot{\varrho}(0)$ as the initial conditions for Eq. (27), and $v_{max} \in \mathbb{R}_{>0}$ as a bound for w_0 , i.e. $|w_0| \le v_{max}$, there exists a strictly positive number γ that satisfies $\varrho(t) \ge \gamma$, $\forall t \in [0, \infty)$, if a and b in Eq. (28) are large enough.

Proof. See Appendix B.
$$\Box$$

Theorem 4.13. Given $\delta_0 := \delta(0)$ and $v_0 := \dot{\delta}(0)$ as the robot's initial distance from the obstacle and its initial speed in Eq. (26), respectively, and also assuming $|v_0| \le v_{max}$, there exists a lower bound on the robot's distance from the obstacle. If the robot starts its motion in the safe area, this bound will be uniform and depend on δ_c and v_{max} .

³We assume that ρ_0 is positive, since we want to compare $\rho(t)$ with $\delta(t)$, which we proved is always positive in Theorem 4.10.



Fig. 5: Illustration of the right-hand side of Eq. (29) compared to a function in the form of Eq. (28) with ρ replaced by δ .

Proof. We rewrite Eq. (26) in the following form:

$$\ddot{\delta} + k\dot{\delta} = -k_R\varphi'(\delta) - kv_{des}.$$
(29)

The right-hand side of Eq. (29) is the orange curve in Fig. 5. From this plot, we can see that there is at least one function $h(\delta)$ in the form of Eq. (28) (the green straight line in Fig. 5) that satisfies the following condition,

$$h(\delta) \le -k_R \varphi'(\delta) - k v_{des}, \quad \forall \delta \in \mathbb{R}_{>0}, \tag{30}$$

for the type of potential field defined by Eq. (1). Hence, we can write the following differential inequality:

$$\ddot{\delta} + k\dot{\delta} \ge -a\delta + b, \quad \delta(0) = \delta_0, \ \dot{\delta}(0) = v_0.$$
 (31)

Now, considering the system in Eq. (27) with the initial conditions set as $\rho_0 = \delta_0$, $w_0 = v_0$, and defining

$$\chi_{\varrho} := \dot{\varrho} + k\varrho, \quad \chi_{\delta} := \dot{\delta} + k\delta, \tag{32}$$

we rewrite Eq. (27) and Eq. (31) as

$$\dot{\chi}_{\varrho} = h(\varrho), \quad \dot{\chi}_{\delta} \ge h(\delta).$$
 (33)

Using the comparison lemma [36], we obtain

$$\chi_{\delta} \ge \chi_{\varrho}, \quad \forall t \in [0, \infty). \tag{34}$$

Furthermore, using the expressions for χ_{δ} and χ_{ϱ} in Eq. (32), we can rewrite Eq. (34) as

$$\dot{\delta} - \dot{\varrho} \ge -k(\delta - \varrho).$$
 (35)

Using the *comparison lemma* again, we obtain

$$\delta(t) \ge (\delta_0 - \varrho_0) e^{-kt} + \varrho(t), \quad \forall t \in [0, \infty),$$
(36)

and taking into account the fact that $\delta_0 = \rho_0$, we conclude that

$$\delta(t) \ge \varrho(t), \quad \forall t \in [0, \infty). \tag{37}$$

Finally, invoking Lemma 4.12, we can write

$$\delta(t) \ge \gamma, \tag{38}$$

which gives a lower bound for $\delta(t)$ and completes the proof. \Box

If we assume that the robot starts its motion in the safe area, we can replace δ_0 with δ_c in all the calculations and obtain a closed-form solution for γ based on the procedure in Appendix B. Such a bound is a function of δ_c and v_{max} , and this bound is uniform with respect to the robot's initial condition.

To conclude this section, we take into account the fact that $-\varphi'(\delta)$ is a decreasing function with respect to δ , which allows us to establish an upper bound for the repulsive term based on the derived lower bound on $\delta(t)$ as

$$\left\|-k_R\varphi'(\delta)\boldsymbol{e}_d\right\| \le -k_R\varphi'(\gamma). \tag{39}$$

5. Analysis of Robot Dynamics for Multiple-Obstacle Case

In this section, we design a control law based on Eq. (2), which was developed for an environment with a single obstacle, and demonstrate that it achieves the three objectives described in Problem 2.5 for an environment that contains multiple strictly convex obstacles. Our solution is to define a switching control law, in which the robot applies the control law Eq. (2) for the closest obstacle that it detects in its sensing range at each time instant. This control law is a discontinuous function because the repulsive term in the control input Eq. (2) undergoes a sudden change in its direction whenever the robot crosses the *switching surface* between two obstacles, which is the loci of all points that are equidistant from the obstacles, as illustrated in Fig. 6. If there are *m* disjoint obstacles in the robot's sensing range, the control law is written as:

$$\boldsymbol{u} = -\boldsymbol{K}(\boldsymbol{\dot{q}} - \boldsymbol{v}_{des}) - \boldsymbol{K}_R \nabla_{\boldsymbol{d}^*} \varphi(\delta^*),$$

$$\delta^* = \min_{i \in \{1, \dots, m\}} \{\delta_i\},$$
 (40)

where d^* is the collision vector associated with the closest obstacle. The closed-loop dynamics of the robot with control law Eq. (40) can be written as:

$$\ddot{\boldsymbol{q}} + \boldsymbol{K}(\dot{\boldsymbol{q}} - \boldsymbol{v}_{des}) + \boldsymbol{K}_R \nabla_{\boldsymbol{d}^*} \varphi(\boldsymbol{\delta}^*) = \boldsymbol{0}.$$
(41)

Defining the state vector $X = (X_1^T, X_2^T)^T \in \mathbb{R}^4$, where $X_1 = q$ and $X_2 = \dot{q}$, we can rewrite Eq. (41) in state-space form as

$$\dot{X} = f^*(X) := \begin{bmatrix} X_2 \\ -K(X_2 - v_{des}) - K_R \nabla_{d^*} \varphi(\delta^*) \end{bmatrix}.$$
(42)

Eq. (42) is a differential equation with a discontinuous righthand side, since $\nabla_{d^*}\varphi$ may have different directions on the sides of a switching surface between two obstacles. To analyze the solutions X(t) of Eq. (42), suppose that at a given time, the robot is at distance δ_i from obstacle *i* and distance δ_j from obstacle *j*. We then replace the vector field f^* in (42) with f_i and f_j , where f_i is the vector field on the side of the switching surface that contains obstacle *i*, and f_j is the vector field on the side that contains obstacle *j* (see Fig. 6):

$$f_{i}(X) = \begin{bmatrix} X_{2} \\ -K(X_{2} - v_{des}) - K_{R} \nabla_{d_{i}} \varphi(\delta_{i}) \end{bmatrix},$$

$$f_{j}(X) = \begin{bmatrix} X_{2} \\ -K(X_{2} - v_{des}) - K_{R} \nabla_{d_{j}} \varphi(\delta_{j}) \end{bmatrix}.$$
 (43)

On each side of the switching surface, the robot's dynamics are described by Eq. (4), and therefore exhibit the desired velocity convergence and collision avoidance behaviors as we proved in Sections 4.1 and 4.2. On the switching surface, however, the closed-loop system (42) can have two types of solutions, depending on the directions of the vector fields f_i and f_j with respect to the switching surface. If the components of f_i and f_i that are normal to the switching surface are pointing in the same direction (Fig. 7, left), then the solution of the closed-loop system is a Carathéodory solution, which is an absolutely continuous function satisfying the integral equation corresponding to Eq. (42), $X(t) = X(t_0) + \int_{t_0}^t f^*(X(\tau)) d\tau$ [38]. In this case, the system trajectory passes through the switching surface. If the two components that are normal to the switching surface point in opposite directions (Fig. 7, right), then the system has a Filippov solution that satisfies the following differential inclusion [38], defined in terms of a convex combination of f_i and f_j :

$$\dot{\mathbf{X}} \in \boldsymbol{F}(\boldsymbol{X}) := \left\{ \alpha \boldsymbol{f}_i(\boldsymbol{X}) + (1 - \alpha) \boldsymbol{f}_j(\boldsymbol{X}) : \alpha \in [0, 1] \right\}.$$
(44)

Equation (44) describes the dynamics of the robot as:

3

$$\dot{\boldsymbol{X}} = \begin{cases} \boldsymbol{f}_i(\boldsymbol{X}), & \delta_i < \delta_j \\ \alpha \boldsymbol{f}_i(\boldsymbol{X}) + (1 - \alpha) \boldsymbol{f}_j(\boldsymbol{X}), & \delta_i = \delta_j \\ \boldsymbol{f}_j(\boldsymbol{X}), & \delta_i > \delta_j \end{cases}$$
(45)

Since the components of f_i and f_j that are normal to the switching surface are pointing in opposite directions, the system trajectory corresponding to the Filippov solution can only evolve on the switching surface. At the point where the system trajectory reaches the switching surface, there is a unique convex combination of f_i and f_j (i.e., a unique value for α in (44)) that is tangent to this surface, which defines the direction of F(X)on the surface. At each point on the switching surface, the Filippov solution is represented by the value of α for which F(X)is tangent to the surface at that point.

A trajectory corresponding to a Filippov solution often chatters about the switching surface. We note that the proposed controller, in contrast to a sliding mode controller, is not designed to stabilize the system trajectories to the switching surface. Chattering might occur for some time, but the robot will eventually leave the switching surface if certain conditions hold. Theorem 5.1 below guarantees that, under these conditions, the closed-loop system has no equilibria on the switching surface, which ensures that the robot does not become stuck between two obstacles.

Theorem 5.1. Consider an unbounded environment with at least two obstacles for which Assumption 2.4 holds true; i.e., the closest pair of obstacles is separated by a distance larger than the robot's diameter 2r. Given the discontinuous control law in Eq. (40), no equilibrium point exists on the switching surface between any two obstacles in the environment if p in Eq. (1) is sufficiently small.

Proof. Suppose that obstacles i and j are the closest pair of obstacles in an environment. By Assumption 2.4, the distance between these obstacles is greater than 2r. If there exists an



Fig. 6: Illustration of the forces that act on the robot when it detects multiple obstacles in its sensing range.



Fig. 7: A schematic representation of two vector fields that result in (*left*) Carathéodory and (*right*) Filippov solutions for a differential equation with a discontinuous right-hand side.

equilibrium point $(\dot{X} = 0)$ on the switching surface between obstacles *i* and *j*, we have that

$$\alpha \boldsymbol{f}_i(\boldsymbol{X}) + (1 - \alpha)\boldsymbol{f}_i(\boldsymbol{X}) = \boldsymbol{0}. \tag{46}$$

Using the fact that $\delta_i = \delta_j$ on the switching surface, and substituting Eq. (3) for φ and Eq. (43) for f_i and f_j , Eq. (46) becomes:

$$\boldsymbol{K}\boldsymbol{v}_{des} + k_R \frac{p}{\delta_c} \left(1 - \left(\frac{\delta_c}{\delta_s} \right)^p \right) \left(\alpha \boldsymbol{e}_{\boldsymbol{d}_i} + (1 - \alpha) \boldsymbol{e}_{\boldsymbol{d}_j} \right) = \boldsymbol{0}, \quad (47)$$

where we have defined $\delta_s := \delta_i = \delta_j$.

We now derive a conservative upper bound for the parameter p in the potential field. When the robot is on the switching surface, the repulsive force on it has the highest possible component in the direction opposite to v_{des} when $e_{d_i} = e_{d_j}$.⁴ The magnitude of the repulsive force is highest when $\delta_s = r$. Substituting $e_{d_i} = e_{d_j}$ and $\delta_s = r$ into Eq. (47), we can reduce this equation to the following scalar equation:

$$kv_{des} + k_R \frac{p}{\delta_c} \left(1 - \left(\frac{\delta_c}{r}\right)^p \right) = 0.$$
(48)

To prevent the existence of an equilibrium point, and to ensure that the robot converges to the desired velocity, we need the stabilizing term to exceed the repulsive term; i.e.,

$$kv_{des} > -k_R \frac{p}{\delta_c} \left(1 - \left(\frac{\delta_c}{r}\right)^p \right). \tag{49}$$

⁴This is a theoretical scenario that would not happen in practice; we are using it here to obtain a conservative bound on p.

We can rearrange this inequality to obtain the following upper bound on a function of p, called $\mu(p)$:

$$\mu(p) := p\left(\left(\frac{\delta_c}{r}\right)^p - 1\right) < \frac{kv_{des}}{k_R}\delta_c.$$
 (50)

If the closest pair of obstacles are both in the robot's sensing range, we know that $r \leq \delta_c$, and therefore can confirm that $\mu(p)$ in Eq. (50) is strictly increasing for positive p. Hence, we can conclude that p must be small enough for Eq. (50) to hold, which completes the proof.

The result in Theorem 5.1 can be generalized for a point that is equidistant from $l \in \{3, ..., m\}$ obstacles. At such a point, the convex combination of vector fields f_i , which defines the differential inclusion in Eq. (44), is given by $F(X) := \sum_{i=1}^{l} \alpha_i f_i(X)$, where $\alpha_i \in [0, 1]$ for all $i \in \{1, ..., l\}$ and $\sum_{i=1}^{l} \alpha_i = 1$. The vector X is an equilibrium if $\sum_{i=1}^{l} \alpha_i f_i(X) = \mathbf{0}$, which implies that

$$\boldsymbol{K}\boldsymbol{v}_{des} + k_R \frac{p}{\delta_c} \left(1 - \left(\frac{\delta_c}{\delta_s} \right)^p \right) \left(\sum_{i=1}^l \alpha_i \boldsymbol{e}_{\boldsymbol{d}_i} \right) = \boldsymbol{0}, \quad (51)$$

where $\delta_s := \delta_1 = \delta_2 = ... = \delta_l$. Again, we consider the repulsive force on the robot with the highest possible component in the direction opposite to v_{des} , which occurs when $e_{d_1} = e_{d_2} = ... = e_{d_i}$. This simplifies the summation in Eq. (51) to $\sum_{i=1}^{l} \alpha_i e_{d_i} = e_{d_1}$. Finally, setting $\delta_s = r$, Eq. (51) is simplified to Eq. (48). This shows that choosing *p* small enough to satisfy Eq. (50) will also guarantee the absence of an equilibrium at a point that is equidistant from three or more obstacles.

6. Simulation Results

To validate our controller, we simulated the motion of a diskshaped holonomic robot in environments with a single strictly convex obstacle or multiple strictly convex obstacles. The robot's radius is r = 0.1 m, and its sensing radius is $\delta_c = 0.5$ m. The desired velocity is set to $v_{des} = 0.1$ m/s along the x-axis of the global frame for all the simulations. We present results for one scenario with a single obstacle and two scenarios with multiple obstacles. In all scenarios, the robot starts its motion in the safe area to the left of the obstacles, does not have any global localization or prior information about the shapes and locations of the obstacles, and only knows the desired velocity.

6.1. Single obstacle

We first consider an environment with an elliptical obstacle. The control parameters in Eq. (2) are set to k = 1, $k_R = 0.05$, and p = 0.4. Figure 8 plots the trajectory of the robot in this environment, showing that it travels past the obstacle without collision. The robot's *x* and *y* velocity components over its trajectory are plotted versus time in Fig. 9. This figure shows that the robot quickly approaches the desired velocity until it detects the obstacle in its sensing range. Then, the controller redirects the robot so that it travels around the obstacle, as indicated by the increase in the *y* velocity. The robot converges to the desired velocity after it travels far enough from the obstacle that it cannot detect it within its sensing range.

6.2. Multiple obstacles

In the first scenario, we consider an environment with six identical circular obstacles, shown in Fig. 10. The radius of each obstacle is 1.1 m, and Assumption 2.4 is satisfied. The control parameters are set to k = 1, $k_R = 0.05$, and p = 0.32. Fig. 10 shows that the robot travels between the obstacles without colliding with them or becoming entrapped. Figure 11 plots the time evolution of the robot's velocity components, which oscillate as the robot maneuvers between the obstacles. The sudden changes in the *y* velocity component occur at times when the robot detects a new obstacle in its sensing range and begins to circumvent the obstacle. The robot converges to the desired velocity after it travels past all six obstacles.

In the second scenario, we consider an environment with four different strictly convex obstacles, shown in Fig. 12. Assumption 2.4 is satisfied, since the shortest distance between the closest pair of obstacles (Obstacles 1 and 2) is 0.3 m. As illustrated in Fig. 12, the robot travels past the obstacles without colliding with them or becoming entrapped between them. Figure 13 plots the time evolution of the robot's velocity components. The robot's velocity displays a chattering behavior between times A and B, when it passes through the narrow channel between Obstacles 1 and 2. This is due to its frequent crossing of the switching surface between these two obstacles, which indicates that its trajectory is a Filippov solution of the closed-loop dynamics (43)-(44), as described in Section 5 (Fig. 7, right). At time C, the robot crosses the switching surface between Obstacles 2 and 3. No chattering occurs at this time, since the resultant of the velocity stabilizing force and the two repulsive forces from Obstacles 2 and 3 prevent the robot from entering the narrow channel between these two obstacles. At time D, the robot crosses the switching surface between Obstacles 3 and 4. No chattering is observed at this time either, because the channel between these two obstacles is relatively wide. The absence of chattering about the last two switching surfaces indicates that the robot's trajectory through these switching surfaces is a Carathéodory solution of the closed-loop dynamics, as discussed in Section 5 (Fig. 7, left). The robot stops sensing Obstacle 3 after it passes the corresponding point D in Fig. 12 and is repelled only by Obstacle 4. Fig. 13 shows that after circumventing all the obstacles, the robot converges to v_{des} .

7. Conclusion and Future Work

We proposed an obstacle avoidance controller for a holonomic finite-dimensional robot in an unbounded, GPS-denied environment with unknown strictly convex obstacles. The controller relies only on the robot's local measurements and does not require any information about the locations and geometry of the obstacles. We first studied the case where the environment has a single obstacle and proved that with the proposed controller, no collision takes place and the robot converges to the desired velocity after it passes the obstacle. For the case of multiple obstacles, we proposed a switching control scheme and showed that the robot avoids collisions and converges to the



Fig. 8: Simulation of a disk-shaped holonomic robot's motion in an environment with a single elliptical obstacle.



Fig. 9: Time evolution of the robot's x and y velocity components in the global frame while it moves along the red trajectory shown in Fig. 8.

target velocity if it uses the controller designed for the singleobstacle case for the closest obstacle at each time instant. Moreover, the robot never becomes trapped between any pair of obstacles (i.e., there are no local stable equilibrium points) if it uses a sufficiently small p in the equation for the virtual potential field, which is used in the repulsive term of the controller.

In future work, we will modify our proposed controller to solve the obstacle avoidance problem in environments with concave obstacles and validate the controller on physical omnidirectional robots. We will also combine our controller with the controller presented in our previous work [39] in order to create a completely decentralized control strategy for multi-robot collective transport in unknown environments with obstacles.

Appendix A. Calculation of the gradients of φ with respect to *d* and *q*

We can write the gradient of φ with respect to d as

$$\nabla_{d}\varphi = \frac{\partial\varphi}{\partial d} = \frac{\partial\varphi}{\partial\delta}\frac{\partial\delta}{\partial d}.$$
 (A.1)

We represent d in terms of its components in the global coordinate frame as $d := [d_x d_y]^T$. Then, from the definition of δ , we



Fig. 10: Simulation of the robot's motion in an environment with six circular obstacles in which Assumption 2.4 is satisfied.



Fig. 11: Time evolution of the robot's x and y velocity components in the global frame while it moves along the red trajectory shown in Fig. 10.

have that $\delta = ||\boldsymbol{d}|| - r = \sqrt{d_x^2 + d_y^2} - r$. Using this expression for δ , we obtain:

$$\frac{\partial \delta}{\partial \boldsymbol{d}} = \begin{bmatrix} \frac{\partial \delta}{\partial d_x} \\ \frac{\partial \delta}{\partial d_y} \end{bmatrix} = \begin{bmatrix} \frac{d_x}{\sqrt{d_x^2 + d_y^2}} \\ \frac{d_y}{\sqrt{d_x^2 + d_y^2}} \end{bmatrix} = \frac{1}{\sqrt{d_x^2 + d_y^2}} \begin{bmatrix} d_x \\ d_y \end{bmatrix} = \boldsymbol{e_d}.$$
 (A.2)

Therefore, $\nabla_d \varphi$ is given by

$$\nabla_d \varphi = \frac{\partial \varphi}{\partial d} = \frac{\partial \varphi}{\partial \delta} e_d. \tag{A.3}$$

We also represent the position of the projection point in terms of its components in the global frame as $\boldsymbol{q}_P = [\boldsymbol{q}_{p,x} \, \boldsymbol{q}_{p,y}]^T$. Then the vector equation $\boldsymbol{q} = \boldsymbol{d} + \boldsymbol{q}_P$ can be written as

$$d_x = x - q_{p,x}, \quad d_y = y - q_{p,y}.$$
 (A.4)

The gradient of φ with respect to \boldsymbol{q} can be calculated as

$$\nabla_{q}\varphi = \frac{\partial\varphi}{\partial q} = \frac{\partial\varphi}{\partial\delta}\frac{\partial\delta}{\partial q}.$$
 (A.5)

By the chain rule, the term $\frac{\partial \delta}{\partial a}$ can be expressed as

$$\frac{\partial \delta}{\partial \boldsymbol{q}} = \frac{\partial \delta}{\partial d_x} \frac{\partial d_x}{\partial \boldsymbol{q}} + \frac{\partial \delta}{\partial d_y} \frac{\partial d_y}{\partial \boldsymbol{q}}, \tag{A.6}$$



Fig. 12: Simulation of the robot's motion in an environment with four different strictly convex obstacles in which Assumption 2.4 is satisfied. The points labeled A, B, C, and D are the locations of the robot at the corresponding times labeled in Fig. 13.



Fig. 13: Time evolution of the robot's x and y velocity components in the global frame while it moves along the red trajectory shown in Fig. 12.

which can be rewritten as

$$\frac{\partial \delta}{\partial \boldsymbol{q}} = \frac{\partial \delta}{\partial d_x} \begin{bmatrix} \frac{\partial d_x}{\partial x} \\ \frac{\partial d_x}{\partial y} \end{bmatrix} + \frac{\partial \delta}{\partial d_y} \begin{bmatrix} \frac{\partial d_y}{\partial x} \\ \frac{\partial d_y}{\partial y} \end{bmatrix}.$$
 (A.7)

Given Eq. (A.4) and the fact that $\delta = \sqrt{d_x^2 + d_y^2} - r$, we can calculate the partial derivatives in Eq. (A.7) to obtain:

$$\frac{\partial \delta}{\partial \boldsymbol{q}} = \frac{d_x}{\sqrt{d_x^2 + d_y^2}} \begin{bmatrix} 1\\0 \end{bmatrix} + \frac{d_y}{\sqrt{d_x^2 + d_y^2}} \begin{bmatrix} 0\\1 \end{bmatrix} = \frac{1}{\sqrt{d_x^2 + d_y^2}} \begin{bmatrix} d_x\\d_y \end{bmatrix} = \boldsymbol{e_d}.$$
(A.8)

Substituting this expression for $\frac{\partial \delta}{\partial q}$ into Eq. (A.5), we find that

$$\nabla_{q}\varphi = \frac{\partial\varphi}{\partial\delta}\boldsymbol{e}_{d},\tag{A.9}$$

which is identical to Eq. (A.3). Therefore, we conclude that

$$\nabla_d \varphi = \nabla_q \varphi. \tag{A.10}$$

Appendix B. Proof of Lemma 4.12

We describe a procedure for choosing *a* and *b* in Eq. (28) in order to ensure a strictly positive lower bound γ on the time response $\rho(t)$ of the system in Eq. (27). We know that any unforced scalar linear second-order system can be written in the following form [40],

$$\ddot{\varrho} + 2\zeta \omega_n \dot{\varrho} + \omega_n^2 \varrho = 0. \tag{B.1}$$

Hence, the system in Eq. (27) can be represented as

$$\ddot{\varrho} + 2\zeta \omega_n \dot{\varrho} + \omega_n^2 \varrho = b, \qquad (B.2)$$

where

$$\omega_n = \sqrt{a},\tag{B.3}$$

$$\zeta = \frac{\kappa}{2\sqrt{a}},\tag{B.4}$$

and consequently, its time response is written as

$$\varrho(t) = e^{-\zeta\omega_n t} \left(c_1 \cos(\omega_n t) + c_2 \sin(\omega_n t) \right) + b', \quad t \ge 0, \quad (B.5)$$

in which

6

$$e_1 = \varrho_0 - \frac{b}{\omega_n^2}, \tag{B.6}$$

$$c_2 = \frac{1}{\omega_n} (w_0 + c_1 \zeta \omega_n), \qquad (B.7)$$

$$b' = \frac{b}{\omega_n^2}.$$
 (B.8)

Let us choose *b* such that $c_1 = 0^5$, i.e.,

$$b := \varrho_0 \omega_n^2. \tag{B.9}$$

Then, from Eq. (B.5), we obtain the following inequality:

$$\varrho(t) = e^{-\zeta \omega_n t} c_2 \sin(\omega_n t) + b' \ge -|c_2| + b', \ t \ge 0.$$
 (B.10)

In order to ensure that $\rho(t) \ge \gamma$ for an arbitrary $\gamma \in (0, \rho_0)$, we therefore need to enforce the condition $-|c_2| + b' \ge \gamma$. To do this, we choose b' to satisfy this condition. Using Eq. (B.7) with $c_1 = 0$, this condition can be written as:

$$b' \geq |c_2| + \gamma = \frac{|w_0|}{\omega_n} + \gamma.$$
 (B.11)

Taking into account the fact that $|w_0| \le v_{max}$, we can ensure that Eq. (B.11) is true by defining b' such that:

$$b' \ge \frac{v_{max}}{\omega_n} + \gamma.$$
 (B.12)

Noting that $b' = \rho_0$ from Eq. (B.8) and Eq. (B.9), the above inequality implies that

$$\omega_n \ge \frac{v_{max}}{(\varrho_0 - \gamma)}.\tag{B.13}$$

⁵This is not the only feasible choice for b. This is the most convenient choice that facilitates the calculations.

Since $a = \omega_n^2$ by Eq. (B.3), Eq. (B.13) implies that *a* should be chosen such that

$$a \ge \frac{v_{max}^2}{(\varrho_0 - \gamma)^2}.\tag{B.14}$$

By Eq. (B.3) and Eq. (B.9), we have that

$$b = \varrho_0 a. \tag{B.15}$$

We can then define b according to the selected value of a.

This proof shows that the establishment of a specific lower bound γ for the time response of the system in Eq. (27) requires *a* and *b* to be chosen such that the conditions in Eq. (B.14) and Eq. (B.15) hold. Also, the selection of sufficiently large values for *a* and *b* never contradicts Eq. (30), since we can always choose values of *a* and *b* such that the corresponding green line in Fig. 5 lies below the orange curve in that figure.

References

- W. Xi, X. Tan, J. S. Baras, A hybrid scheme for distributed control of autonomous swarms, in: Proceedings of the 2005 American Control Conference, 2005, pp. 3486–3491.
- [2] S. M. LaValle, Planning algorithms, Cambridge University Press, 2006.
- [3] S. R. Lindemann, S. M. LaValle, Simple and efficient algorithms for computing smooth, collision-free feedback laws over given cell decompositions, The International Journal of Robotics Research 28 (5) (2009) 600– 621.
- [4] S. Karaman, E. Frazzoli, Sampling-based algorithms for optimal motion planning, The International Journal of Robotics Research 30 (7) (2011) 846–894.
- [5] C. Belta, V. Isler, G. J. Pappas, Discrete abstractions for robot motion planning and control in polygonal environments, IEEE Transactions on Robotics 21 (5) (2005) 864–874.
- [6] C. I. Connolly, J. B. Burns, R. Weiss, Path planning using Laplace's equation, in: Proceedings of the 1990 IEEE International Conference on Robotics and Automation, 1990, pp. 2102–2106.
- [7] J.-O. Kim, P. K. Khosla, Real-time obstacle avoidance using harmonic potential functions, IEEE Transactions on Robotics and Automation 8 (3) (1992) 338–349.
- [8] E. Rimon, D. E. Koditschek, Exact robot navigation using artificial potential functions, IEEE Transactions on Robotics and Automation 8 (5) (1992) 501–518.
- [9] O. Khatib, Real-time obstacle avoidance for manipulators and mobile robots, The International Journal of Robotics Research 5 (1) (1986) 90– 98.
- [10] R. Shahidi, M. Shayman, P. S. Krishnaprasad, Mobile robot navigation using potential functions, in: Proceedings of the 1991 IEEE International Conference on Robotics and Automation, 1991, pp. 2047–2053.
- [11] D. E. Koditschek, E. Rimon, Robot navigation functions on manifolds with boundary, Advances in Applied Mathematics 11 (4) (1990) 412–442.
- [12] D. C. Conner, A. A. Rizzi, H. Choset, Composition of local potential functions for global robot control and navigation, in: Proceedings of the 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2003, pp. 3546–3551.
- [13] P. Ogren, N. E. Leonard, A convergent dynamic window approach to obstacle avoidance, IEEE Transactions on Robotics 21 (2) (2005) 188–195.
- [14] D. Fox, W. Burgard, S. Thrun, The dynamic window approach to collision avoidance, IEEE Robotics and Automation Magazine 4 (1) (1997) 23–33.
- [15] I. Filippidis, K. J. Kyriakopoulos, Adjustable navigation functions for unknown sphere worlds, in: Proceedings of the 2011 IEEE International Conference on Decision and Control and European Control Conference, 2011, pp. 4276–4281.
- [16] C. Li, H. G. Tanner, Navigation functions with time-varying destination manifolds in star worlds, IEEE Transactions on Robotics 35 (1) (2019) 35–48.
- [17] S. Paternain, D. E. Koditschek, A. Ribeiro, Navigation functions for convex potentials in a space with convex obstacles, IEEE Transactions on Automatic Control 63 (9) (2018) 2944–2959.

- [18] S. Prajna, A. Jadbabaie, G. J. Pappas, Stochastic safety verification using barrier certificates, in: Proceedings of the 2004 IEEE International Conference on Decision and Control, 2004, pp. 929–934.
- [19] A. D. Ames, X. Xu, J. W. Grizzle, P. Tabuada, Control barrier function based quadratic programs for safety critical systems, IEEE Transactions on Automatic Control 62 (8) (2017) 3861–3876.
- [20] L. Wang, A. D. Ames, M. Egerstedt, Safety barrier certificates for collisions-free multirobot systems, IEEE Transactions on Robotics 33 (3) (2017) 661–674.
- [21] J. Guldner, V. I. Utkin, Sliding mode control for gradient tracking and robot navigation using artificial potential fields, IEEE Transactions on Robotics and Automation 11 (2) (1995) 247–254.
- [22] S. Paternain, A. Ribeiro, Safe online navigation of convex potentials in spaces with convex obstacles, in: Proceedings of the 2017 IEEE International Conference on Decision and Control, 2017, pp. 2473–2478.
- [23] O. Arslan, D. E. Koditschek, Sensor-based reactive navigation in unknown convex sphere worlds, The International Journal of Robotics Research 38 (2-3) (2019) 196–223.
- [24] S. S. Ge, Y. J. Cui, New potential functions for mobile robot path planning, IEEE Transactions on Robotics and Automation 16 (5) (2000) 615– 620.
- [25] S. Ge, Y. Cui, Dynamic motion planning for mobile robots using potential field method, Autonomous Robots 13 (3) (2002) 207–222.
- [26] E. Ramírez-Llanos, S. Martínez, Stochastic source seeking for mobile robots in obstacle environments via the SPSA method, IEEE Transactions on Automatic Control 64 (4) (2019) 1732–1739.
- [27] W. Rassameepaiboon, W. Assawinchaichote, Vehicle avoidance reaction by two step motion flow cluster, in: 2018 22nd International Computer Science and Engineering Conference (ICSEC), 2018, pp. 1–6.
- [28] R. P. Padhy, S. K. Choudhury, P. K. Sa, S. Bakshi, Obstacle avoidance for unmanned aerial vehicles: Using visual features in unknown environments, IEEE Consumer Electronics Magazine 8 (3) (2019) 74–80.
- [29] M. Ferro, A. Paolillo, A. Cherubini, M. Vendittelli, Vision-based navigation of omnidirectional mobile robots, IEEE Robotics and Automation Letters 4 (3) (2019) 2691–2698.
- [30] H. G. Tanner, A. Jadbabaie, G. J. Pappas, Stable flocking of mobile agents, part I: fixed topology, in: Proceedings of the IEEE International Conference on Decision and Control, Vol. 2, 2003, pp. 2010–2015.
- [31] D. Sakai, H. Fukushima, F. Matsuno, Flocking for multirobots without distinguishing robots and obstacles, IEEE Transactions on Control Systems Technology 25 (3) (2017) 1019–1027.
- [32] T. Bräunl, Omni-directional robots, Embedded Robotics: Mobile Robot Design and Applications with Embedded Systems (2008) 147–156.
- [33] H. W. Ho, G. C. de Croon, Q. Chu, Distance and velocity estimation using optical flow from a monocular camera, International Journal of Micro Air Vehicles 9 (3) (2017) 198–208.
- [34] D. P. Bertsekas, A. Nedich, A. E. Ozdaglar, Convex analysis and optimization, Athena Scientific, 2003.
- [35] D. E. Koditschek, The application of total energy as a Lyapunov function for mechanical control systems, Contemporary Mathematics 97 (1989) 131–157.
- [36] H. K. Khalil, Nonlinear systems, Prentice Hall, Upper Saddle River, N.J., 1996.
- [37] J. L. Meriam, L. G. Kraige, Engineering mechanics: dynamics, Vol. 2, John Wiley & Sons, 2012.
- [38] D. Liberzon, Switching in systems and control, Springer Science & Business Media, 2003.
- [39] H. Farivarnejad, S. Berman, Stability and convergence analysis of a decentralized proportional-integral control strategy for collective transport, in: Proceedings of the 2018 American Control Conference, 2018, pp. 2794–2801.
- [40] G. F. Franklin, J. D. Powell, A. Emami-Naeini, Feedback control of dynamic systems, Prentice Hall Press, 2014.