Real-Time Sensor-Based Feedback Control for Obstacle Avoidance in Unknown Environments

Lyes Smaili [1]   Soulaimane Berkane [1,2]

[1]University of Quebec in Outaouais, QC, Canada
[2]Lakehead University, ON, Canada

American Control Conference, July 8-12, 2024

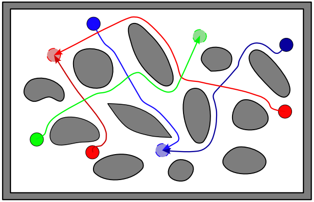
Introduction

Global approaches

- Navigation functions (Koditschek et al., 1990).
- Navigation transform (Loizou, 2017).
- Hybrid feedback (Sanfelice et al., 2006; Berkane et al., 2019; Casau et al., 2019; Berkane et al., 2021).

Local approaches

- Artificial potential fields (Khatib, 1986; Koditschek et al., 1990).
- Extended navigation functions (Lionis et al., 2007; Filippidis et al., 2011).
- Separating hyperplanes (Arslan et al., 2019).
- Navigation through safety velocity cones (Berkane, 2021).



- Navigate unknown environments filled with obstacle.
- **Objective**: design a sensor-based feedback control to ensure safe navigation.

Notations and Definitions

Reach of sets

The skeleton $\mathbf{Sk}(\mathcal{A})$:

$$\mathbf{Sk}(\mathcal{A}) := \{x \in \mathbb{R}^n : \mathbf{card}(\mathbf{P}_{\overline{\mathcal{A}}}(x)) > 1\}.$$

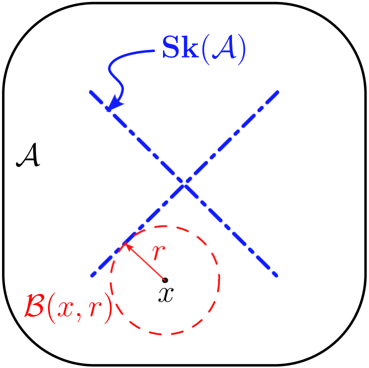
The reach of $\mathcal{A}$ at $x \in \overline{\mathcal{A}}$ is defined as

$$\mathbf{reach}(\mathcal{A}, x) := \begin{cases} 0, & x \in \partial\overline{\mathcal{A}} \cap \overline{\mathbf{Sk}(\mathcal{A})}, \\ \sup\{r > 0 : \mathbf{Sk}(\mathcal{A}) \cap \mathcal{B}(x, r) = \emptyset\}, & \text{otherwise.} \end{cases}$$

The reach of the set $\mathcal{A}$ is given by

$$\mathbf{reach}(\mathcal{A}) := \inf_{x \in \mathcal{A}} \{\mathbf{reach}(\mathcal{A}, x)\}.$$

The set $\mathcal{A}$ has **positive reach** if $\mathbf{reach}(\mathcal{A}) > 0$.



Notations and Definitions

Sets of class $\mathcal{C}^{k,l}$

- $k \in \mathbb{N}$: the **differentiability order**.
- $l \in [0, 1]$: is the **Hölder continuity**.

- If $l = 0$: the kth derivative is bounded.
- If $l = 1$: the kth derivative is Lipschitz continuous.

Notations and Definitions

Tangent cone

We denote by $\mathbf{T}_{\mathcal{A}}(x)$ the **tangent cone** to $\mathcal{A}$ at a point $x \in \mathbb{R}^n$, and is given by

$$\mathbf{T}_{\mathcal{A}}(x) = \Big\{ z \in \mathbb{R}^n : \lim_{\tau \to 0^+} \frac{\mathbf{d}_{\mathcal{A}}(x + \tau z)}{\tau} = 0 \Big\}.$$

Nagumo's theorem

Consider the system $\dot{x} = f(x)$. For each initial condition $x(0) \in \mathcal{X}$, where $\mathcal{X}$ is a closed set, we assume that the system admits a unique solution. Then the set $\mathcal{X}$ is forward invariant if and only if

$$f(x) \in \mathbf{T}_{\mathcal{X}}(x), \forall x \in \mathcal{X}.$$

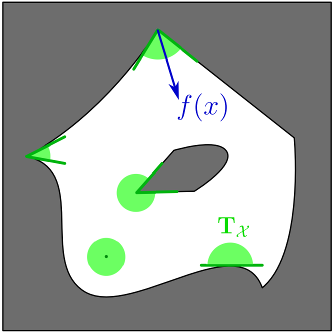


Figure: Tangent cones and Nagumo's Theorem Application.

Notations and Definitions

Tangent cone

We denote by $\mathbf{T}_{\mathcal{A}}(x)$ the **tangent cone** to $\mathcal{A}$ at a point $x \in \mathbb{R}^n$, and is given by

$$\mathbf{T}_{\mathcal{A}}(x) = \Big\{ z \in \mathbb{R}^n : \lim_{\tau \to 0^+} \frac{\mathbf{d}_{\mathcal{A}}(x + \tau z)}{\tau} = 0 \Big\}.$$

Nagumo's theorem

Consider the system $\dot{x} = f(x)$. For each initial condition $x(0) \in \mathcal{X}$, where $\mathcal{X}$ is a closed set, we assume that the system admits a unique solution. Then the set $\mathcal{X}$ is forward invariant if and only if

$$f(x) \in \mathbf{T}_{\mathcal{X}}(x), \forall x \in \mathcal{X}.$$

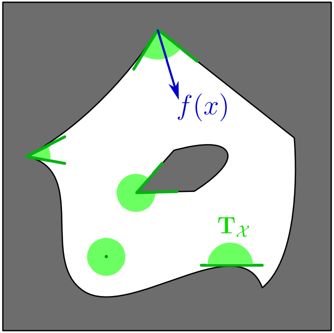
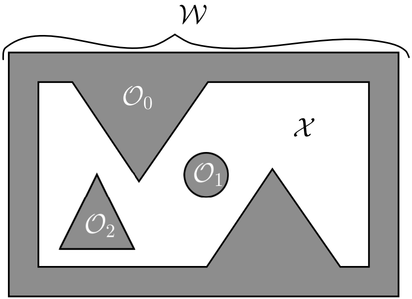


Figure: Tangent cones and Nagumo's Theorem Application.

Problem Formulation

Workspace - Free Space



- Workspace $\mathcal{W}$: closed subset of the Euclidean space $\mathbb{R}^n$.
- Obstacles $\mathcal{O}_i$, $i = 1, ..., M$: open subsets in $\mathbb{R}^n$ strictly contained in $\mathcal{W}$.
- Free space $\mathcal{X}$: a subset in $\mathbb{R}^n$,

$$\mathcal{X} := \mathcal{W} \setminus \bigcup_{i=1}^{M} \mathcal{O}_i. \tag{1}$$

Problem Formulation

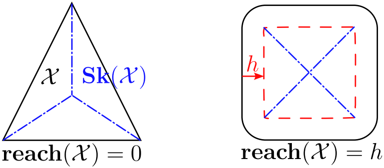
Workspace - Free Space

Assumption 1

The set $\mathcal{X}$ has a **positive reach**, *i.e.*,

$$\mathbf{reach}(\mathcal{X}) > 0. \quad (2)$$

- In other words, there exists a positive real $h > 0$ such that any point $x \in \mathcal{X}$, with $\mathbf{d}_{\complement\mathcal{X}}(x) < h$, has a **unique projection** $\mathbf{P}_{\partial\mathcal{X}}(x)$.



- If Assumption 1 holds, then

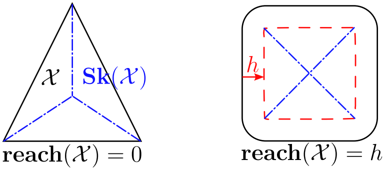
$$\mathbf{d}_{\mathcal{O}_i,\mathcal{O}_j} > 2h, \quad \forall i, j \in \mathbb{M} \text{ with } i \neq j, \quad (3)$$

Assumption 1

The set $\mathcal{X}$ has a **positive reach**, *i.e.*,

$$\mathbf{reach}(\mathcal{X}) > 0. \tag{2}$$

- In other words, there exists a positive real $h > 0$ such that any point $x \in \mathcal{X}$, with $\mathbf{d}_{\complement\mathcal{X}}(x) < h$, has a **unique projection** $\mathbf{P}_{\partial\mathcal{X}}(x)$.



- If Assumption 1 holds, then

$$\mathbf{d}_{\mathcal{O}_i,\mathcal{O}_j} > 2h, \quad \forall i,j \in \mathbb{M} \text{ with } i \neq j, \tag{3}$$

Problem Formulation

Workspace - Practical Free Space

- Consider a ball-shaped robot with a radius R and centred at $x \in \mathbb{R}^n$.
- Let ϵ be a positive real and satisfies the following inequality:

$$0 < R < \epsilon < h. \quad (4)$$

- We define the **practical free space** as follows:

$$\mathcal{X}_\epsilon := \{x \in \mathbb{R}^n : \mathbf{b}_{\complement\mathcal{X}}(x) \geq \epsilon\} \subset \mathcal{X}. \quad (5)$$

Problem Formulation

Workspace - Practical Free Space

- Consider a ball-shaped robot with a radius R and centred at $x \in \mathbb{R}^n$.
- Let ϵ be a positive real and satisfies the following inequality:

$$0 < R < \epsilon < h. \tag{4}$$

- We define the **practical free space** as follows:

$$\mathcal{X}_\epsilon := \{x \in \mathbb{R}^n : \mathbf{b}_{\complement\mathcal{X}}(x) \geq \epsilon\} \subset \mathcal{X}. \tag{5}$$

- Consider a ball-shaped robot with a radius R and centred at $x \in \mathbb{R}^n$.
- Let ϵ be a positive real and satisfies the following inequality:

$$0 < R < \epsilon < h. \tag{4}$$

- We define the **practical free space** as follows:

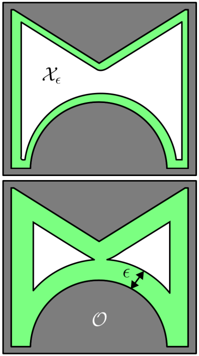
$$\mathcal{X}_\epsilon := \{x \in \mathbb{R}^n : \mathbf{b}_{\complement\mathcal{X}}(x) \geq \epsilon\} \subset \mathcal{X}. \tag{5}$$

- Consider a ball-shaped robot with a radius R and centred at $x \in \mathbb{R}^n$.
- Let ϵ be a positive real and satisfies the following inequality:

$$0 < R < \epsilon < h. \tag{4}$$

- We define the **practical free space** as follows:

$$\mathcal{X}_\epsilon := \{x \in \mathbb{R}^n : \mathbf{b}_{\complement\mathcal{X}}(x) \geq \epsilon\} \subset \mathcal{X}. \tag{5}$$



Problem Formulation

Dynamical System

We consider the first-order robot dynamics

$$\dot{x} = u, \qquad u \in \mathbb{R}^n. \tag{6}$$

- The robot is operating in the free space $\mathcal{X}$ and restricted to stay in the **practical free space** $\mathcal{X}_\epsilon$.
- Let x_d be the **goal** position.

Objectives

finding $u = \kappa(x, x_d, \mathcal{X}_\epsilon)$, such that:

- κ is **locally Lipschitz continuous**.
- κ is **sensor-based**.
- **safety** is guaranteed, *i.e.*, x must stay in $\mathcal{X}_\epsilon$.
- the robot's position is **asymptotically stabilized** at x_d.

- For the sake of simplicity, $\kappa(x, x_d, \mathcal{X}_\epsilon)$ will be written as $\kappa(x)$.

Problem Formulation

Dynamical System

We consider the first-order robot dynamics

$$\dot{x} = u, \qquad u \in \mathbb{R}^n. \tag{6}$$

- The robot is operating in the free space $\mathcal{X}$ and restricted to stay in the **practical free space** $\mathcal{X}_\epsilon$.
- Let x_d be the **goal** position.

Objectives

finding $u = \kappa(x, x_d, \mathcal{X}_\epsilon)$, such that:

- κ is **locally Lipschitz continuous**.
- κ is **sensor-based**.
- **safety** is guaranteed, *i.e.,* x must stay in $\mathcal{X}_\epsilon$.
- the robot's position is **asymptotically stabilized** at x_d.

- For the sake of simplicity, $\kappa(x, x_d, \mathcal{X}_\epsilon)$ will be written as $\kappa(x)$.

Distance-Based Smooth Controller

Feedback Control Design

- To ensure safety, we apply **Nagumo's theorem**, *i.e.*, we need

$$u \in \mathbf{T}_{\mathcal{X}_\epsilon}(x), \quad \forall x \in \mathcal{X}_\epsilon. \tag{7}$$

- To guarantee **convergence**, one must keep u as close as possible to the **nominal controller** κ_0.
- This consists of solving the *nearest point problem* formulated as follows

$$\min_u ||u - \kappa_0(x)||^2 \text{ subject to } u \in \mathbf{T}_{\mathcal{X}_\epsilon}(x), \forall x \in \mathcal{X}_\epsilon, \tag{8}$$

- The **solution** for u to this **optimization problem** is equivalent to finding the **projection** of the nominal control $\kappa_0(x)$ onto the tangent cone $\mathbf{T}_{\mathcal{X}_\epsilon}(x)$.

- To ensure safety, we apply **Nagumo's theorem**, *i.e.*, we need

$$u \in \mathbf{T}_{\mathcal{X}_\epsilon}(x), \quad \forall x \in \mathcal{X}_\epsilon. \tag{7}$$

- To guarantee **convergence**, one must keep u as close as possible to the **nominal controller** κ_0.
- This consists of solving the *nearest point problem* formulated as follows

$$\min_u ||u - \kappa_0(x)||^2 \text{ subject to } u \in \mathbf{T}_{\mathcal{X}_\epsilon}(x), \forall x \in \mathcal{X}_\epsilon, \tag{8}$$

- The **solution** for u to this **optimization problem** is equivalent to finding the **projection** of the nominal control $\kappa_0(x)$ onto the tangent cone $\mathbf{T}_{\mathcal{X}_\epsilon}(x)$.

Distance-Based Smooth Controller

Feedback Control Design

- To ensure safety, we apply **Nagumo's theorem**, *i.e.*, we need

$$u \in \mathbf{T}_{\mathcal{X}_\epsilon}(x), \qquad \forall x \in \mathcal{X}_\epsilon. \tag{7}$$

- To guarantee **convergence**, one must keep u as close as possible to the **nominal controller** κ_0.
- This consists of solving the *nearest point problem* formulated as follows

$$\min_u ||u - \kappa_0(x)||^2 \text{ subject to } u \in \mathbf{T}_{\mathcal{X}_\epsilon}(x), \forall x \in \mathcal{X}_\epsilon, \tag{8}$$

- The **solution** for u to this **optimization problem** is equivalent to finding the **projection** of the nominal control $\kappa_0(x)$ onto the tangent cone $\mathbf{T}_{\mathcal{X}_\epsilon}(x)$.

- To ensure safety, we apply **Nagumo's theorem**, *i.e.*, we need

$$u \in \mathbf{T}_{\mathcal{X}_\epsilon}(x), \qquad \forall x \in \mathcal{X}_\epsilon. \tag{7}$$

- To guarantee **convergence**, one must keep u as close as possible to the **nominal controller** κ_0.
- This consists of solving the *nearest point problem* formulated as follows

$$\min_u ||u - \kappa_0(x)||^2 \text{ subject to } u \in \mathbf{T}_{\mathcal{X}_\epsilon}(x), \forall x \in \mathcal{X}_\epsilon, \tag{8}$$

- The **solution** for u to this **optimization problem** is equivalent to finding the **projection** of the nominal control $\kappa_0(x)$ onto the tangent cone $\mathbf{T}_{\mathcal{X}_\epsilon}(x)$.

Distance-Based Smooth Controller

Feedback Control Design

- When $x \in \mathbf{int}(\mathcal{X}_\epsilon)$:
  - The tangent cone $\mathbf{T}_{\mathcal{X}_\epsilon}(x) \equiv \mathbb{R}^n$.
- When $x \in \partial\mathcal{X}_\epsilon$:
  - For $\complement\mathcal{X} \neq \{\emptyset\}$, with a positive reach h, $\complement\mathcal{X}_\epsilon$ is a set of class $\mathcal{C}^{1,1}$ and the boundary $\partial\mathcal{X}_\epsilon$ is a $\mathcal{C}^{1,1}$-submanifold of dimension $(n-1)$.[a]
  - The tangent cone is given by the half-space

$$\mathbf{T}_{\mathcal{X}_\epsilon}(x) = \{z \in \mathbb{R}^n : v(x)^\top z \leq 0\}, \quad \forall x \in \partial\mathcal{X}_\epsilon, \quad (9)$$

$v(x)$: is the outward normal unit vector associated to each $x \in \partial\mathcal{X}_\epsilon$.

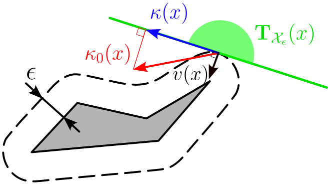


Figure: Tangent cone for smooth boundary sets.

[a] Michel C Delfour and J-P Zolésio. *Shapes and geometries: metrics, analysis, differential calculus, and optimization.* SIAM, 2011.

Distance-Based Smooth Controller

Feedback Control Design

- When $x \in \mathbf{int}(\mathcal{X}_\epsilon)$:
  - The tangent cone $\mathbf{T}_{\mathcal{X}_\epsilon}(x) \equiv \mathbb{R}^n$.
- When $x \in \partial\mathcal{X}_\epsilon$:
  - For $\complement\mathcal{X} \neq \{\emptyset\}$, with a positive reach h, $\complement\mathcal{X}_\epsilon$ is a set of class $\mathcal{C}^{1,1}$ and the boundary $\partial\mathcal{X}_\epsilon$ is a $\mathcal{C}^{1,1}$-submanifold of dimension $(n-1)$.[a]
  - The tangent cone is given by the half-space

$$\mathbf{T}_{\mathcal{X}_\epsilon}(x) = \{z \in \mathbb{R}^n : v(x)^\top z \leq 0\}, \quad \forall x \in \partial\mathcal{X}_\epsilon, \quad (9)$$

$v(x)$: is the outward normal unit vector associated to each $x \in \partial\mathcal{X}_\epsilon$.

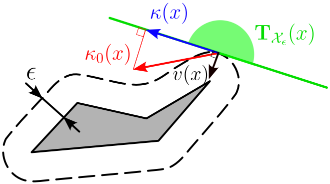


Figure: Tangent cone for smooth boundary sets.

[a] Delfour and Zolésio, *Shapes and geometries: metrics, analysis, differential calculus, and optimization.*

- When $x \in \mathbf{int}(\mathcal{X}_\epsilon)$:
  - The tangent cone $\mathbf{T}_{\mathcal{X}_\epsilon}(x) \equiv \mathbb{R}^n$.
- When $x \in \partial\mathcal{X}_\epsilon$:
  - For $\complement\mathcal{X} \neq \{\emptyset\}$, with a positive reach h, $\complement\mathcal{X}_\epsilon$ is a set of class $\mathcal{C}^{1,1}$ and the boundary $\partial\mathcal{X}_\epsilon$ is a $\mathcal{C}^{1,1}$-submanifold of dimension $(n-1)$.[a]
  - The tangent cone is given by the half-space

$$\mathbf{T}_{\mathcal{X}_\epsilon}(x) = \{z \in \mathbb{R}^n : v(x)^\top z \leq 0\}, \quad \forall x \in \partial\mathcal{X}_\epsilon, \quad (9)$$

$v(x)$: is the outward normal unit vector associated to each $x \in \partial\mathcal{X}_\epsilon$.

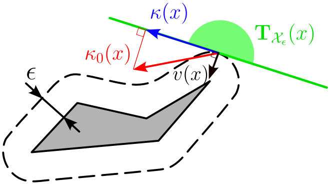
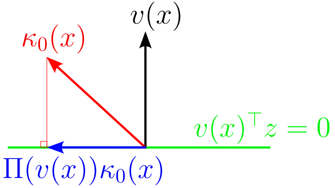


Figure: Tangent cone for smooth boundary sets.

[a] Delfour and Zolésio, *Shapes and geometries: metrics, analysis, differential calculus, and optimization.*

- The projection $\mathbf{P}(\kappa_0(x), \mathbf{T}_{\mathcal{X}_\epsilon}(x))$ is unique.
- When $v(x)^\top \kappa_0(x) > 0$, the projection reduces to the orthogonal projection onto the hyperplane $v(x)^\top z = 0$, which is given by

$$\Pi(v(x))\,\kappa_0(x) := \left(\mathbf{I}_n - v(x)\,v(x)^\top\right)\kappa_0(x). \quad (10)$$

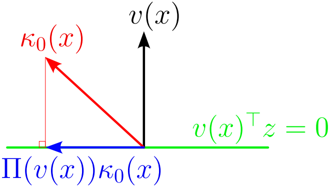


Distance-Based Smooth Controller

Feedback Control Design

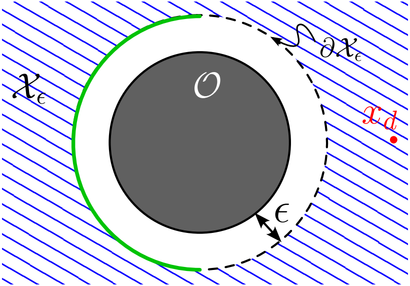
- The projection $\mathbf{P}(\kappa_0(x), \mathbf{T}_{\mathcal{X}_\epsilon}(x))$ is unique.
- When $v(x)^\top \kappa_0(x) > 0$, the projection reduces to the orthogonal projection onto the hyperplane $v(x)^\top z = 0$, which is given by

$$\Pi(v(x))\kappa_0(x) := \left(\mathbf{I}_n - v(x)v(x)^\top\right)\kappa_0(x). \quad (10)$$



The resulting control law that solves the *nearest point problem* is given by

$$u = \kappa(x) = \begin{cases} \kappa_0(x), & x \in \textbf{int}(\mathcal{X}_\epsilon) \text{ or } v(x)^\top \kappa_0(x) \leq 0, \\ \Pi(x)\kappa_0(0), & x \in \partial\mathcal{X}_\epsilon \text{ and } v(x)^\top \kappa_0(x) \geq 0. \end{cases} \quad (11)$$



- Knowing that $v(x) = -\nabla \mathbf{b}_{\complement\mathcal{X}}(x)$, $\forall x \in \partial\mathcal{X}_\epsilon$, we can rewrite the controller in terms of the **distance function**

$$\kappa(x) = \begin{cases} \kappa_0(x), & \mathbf{b}_{\complement\mathcal{X}}(x) > \epsilon \text{ or } \kappa_0(x)^\top \nabla \mathbf{b}_{\complement\mathcal{X}}(x) \geq 0, \\ \Pi(x)\kappa_0(x), & \mathbf{b}_{\complement\mathcal{X}}(x) = \epsilon \text{ and } \kappa_0(x)^\top \nabla \mathbf{b}_{\complement\mathcal{X}}(x) \leq 0, \end{cases} \tag{12}$$

- This controller is a **piecewise continuous vector field** with a discontinuity at the boundary $\partial\mathcal{X}_\epsilon$ of the **practical free space**.

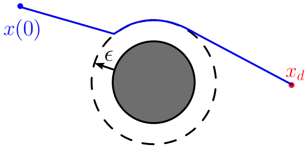


Figure: Resulting trajectory under the piecewise continuous controller.

- Knowing that $v(x) = -\nabla \mathbf{b}_{\complement\mathcal{X}}(x)$, $\forall x \in \partial\mathcal{X}_\epsilon$, we can rewrite the controller in terms of the **distance function**

$$\kappa(x) = \begin{cases} \kappa_0(x), & \mathbf{b}_{\complement\mathcal{X}}(x) > \epsilon \text{ or } \kappa_0(x)^\top \nabla \mathbf{b}_{\complement\mathcal{X}}(x) \geq 0, \\ \Pi(x)\kappa_0(x), & \mathbf{b}_{\complement\mathcal{X}}(x) = \epsilon \text{ and } \kappa_0(x)^\top \nabla \mathbf{b}_{\complement\mathcal{X}}(x) \leq 0, \end{cases} \tag{12}$$

- This controller is a **piecewise continuous vector field** with a discontinuity at the boundary $\partial\mathcal{X}_\epsilon$ of the **practical free space**.

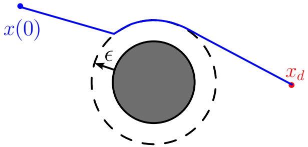


Figure: Resulting trajectory under the piecewise continuous controller.

Distance-Based Smooth Controller

Feedback Control Design

The **smoothed version of the controller** is given by

$$\kappa(x) = \begin{cases} \kappa_0(x), & \mathbf{b}_{\complement\mathcal{X}}(x) > \epsilon' \text{ or } \kappa_0(x)^\top \nabla \mathbf{b}_{\complement\mathcal{X}}(x) \geq 0, \\ \hat{\Pi}(x)\,\kappa_0(x), & \mathbf{b}_{\complement\mathcal{X}}(x) \leq \epsilon' \text{ and } \kappa_0(x)^\top \nabla \mathbf{b}_{\complement\mathcal{X}}(x) \leq 0, \end{cases} \tag{13}$$

where $0 < R < \epsilon < \epsilon' \leq h$ and

$$\hat{\Pi}(x) := \mathbf{I}_n - \phi(x)\,\nabla \mathbf{b}_{\complement\mathcal{X}}(x)\nabla \mathbf{b}_{\complement\mathcal{X}}(x)^\top, \tag{14}$$

$$\phi(x) := \min\left(1, \frac{\epsilon' - \mathbf{b}_{\complement\mathcal{X}}(x)}{\epsilon' - \epsilon}\right). \tag{15}$$

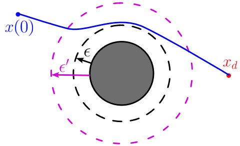


Figure: Resulting trajectory under the smooth controller.

Lemma 1

We assume the free space $\mathcal{X}$ is a set of class $\mathcal{C}^{2,l}$, where $0 \leq l \leq 1$. Consider the practical free space set $\mathcal{X}_\epsilon$. Then, the smoothed control $\kappa(x)$ is **locally Lipschitz-continuous**.

Theorem 2 (Safety)

Consider the set $\mathcal{X} \subset \mathbb{R}^n$ that describes the free space. Consider the set $\mathcal{X}_\epsilon \in \mathbb{R}^n$ that describes the practical free space. Consider the closed-loop system under the locally Lipschitz-continuous control law $\kappa(\cdot)$. Then,

- the closed-loop system admits a **unique solution** and,
- the set $\mathcal{X}_\epsilon$ is **forward invariant**.

- The forward invariance of $\mathcal{X}_\epsilon$ is equivalent to the safety of the robot.

Lemma 1

We assume the free space $\mathcal{X}$ is a set of class $\mathcal{C}^{2,l}$, where $0 \leq l \leq 1$. Consider the practical free space set $\mathcal{X}_\epsilon$. Then, the smoothed control $\kappa(x)$ is **locally Lipschitz-continuous**.

Theorem 2 (Safety)

Consider the set $\mathcal{X} \subset \mathbb{R}^n$ that describes the free space. Consider the set $\mathcal{X}_\epsilon \in \mathbb{R}^n$ that describes the practical free space. Consider the closed-loop system under the locally Lipschitz-continuous control law $\kappa(\cdot)$. Then,

- the closed-loop system admits a **unique solution** and,
- the set $\mathcal{X}_\epsilon$ is **forward invariant**.

- The forward invariance of $\mathcal{X}_\epsilon$ is equivalent to the safety of the robot.

We a consider the following **nominal controller**

$$\kappa_0(x) = -k(x - x_d), \qquad k > 0. \tag{16}$$

Theorem 3 (Stability)

Under the nominal controller $\kappa_0(.)$ given by (16), we have

- the distance $||x - x_d||$ is non-increasing,
- the equilibrium point $x = x_d$ is locally exponentially stable, and
- and trajectories converge to the set $\mathcal{E} \cup \{x_d\}$, where

$$\mathcal{E} := \{x : \mathbf{b}_{\complement\mathcal{X}}(x) = \epsilon, (x - x_d) = \lambda \nabla \mathbf{b}_{\complement\mathcal{X}}(x), \lambda \in \mathbb{R}_{>0}\} \tag{17}$$

  is a set of measure zero.

- The nature of the undesired equilibria defined by the set $\mathcal{E}$ is directly related to the topology of the obstacles.

We a consider the following **nominal controller**

$$\kappa_0(x) = -k(x - x_d), \qquad k > 0. \tag{16}$$

Theorem 3 (Stability)

Under the nominal controller $\kappa_0(.)$ given by (16), we have

- the distance $||x - x_d||$ is non-increasing,
- the equilibrium point $x = x_d$ is locally exponentially stable, and
- and trajectories converge to the set $\mathcal{E} \cup \{x_d\}$, where

$$\mathcal{E} := \{x : \mathbf{b}_{\complement\mathcal{X}}(x) = \epsilon, (x - x_d) = \lambda \nabla \mathbf{b}_{\complement\mathcal{X}}(x), \lambda \in \mathbb{R}_{>0}\} \tag{17}$$

is a set of measure zero.

- The nature of the undesired equilibria defined by the set $\mathcal{E}$ is directly related to the topology of the obstacles.

- We suppose all obstacles are **convex**.

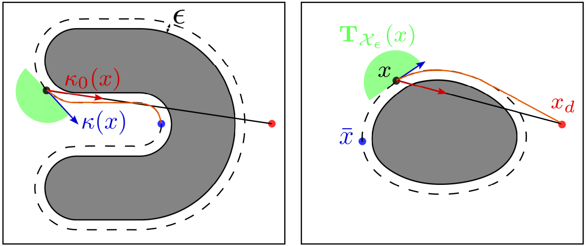


Figure: (Left) shows a non-convex obstacle for which the trajectory of the robot converges to the undesired equilibrium, and (right) shows a convex obstacle for which the trajectory converges to the desired goal x_d.

Assumption 2 (Strong Convexity)

The Jacobian matrix $\mathbf{J}_{\mathbf{P}_{\partial\mathcal{X}}}(x)$ of the metric projection of any stationary point $x \in \mathcal{E}$ onto the boundary $\partial\mathcal{X}$ of the free-space satisfy

$$\mathbf{J}_{\mathbf{P}_{\partial\mathcal{X}}}(x) \prec \frac{||x_d - \mathbf{P}_{\partial\mathcal{X}}(x)||}{\epsilon + ||x_d - \mathbf{P}_{\partial\mathcal{X}}(x)||}\mathbf{I}_n, \quad \forall x \in \mathcal{X}, \tag{18}$$

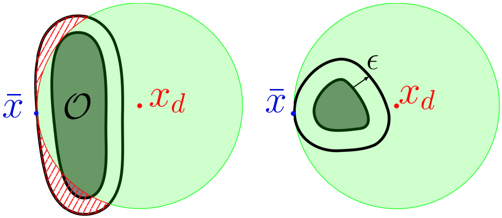


Figure: Visualization of Assumption 3: (left) shows an obstacle that doesn't meet the assumption. (Right) shows an obstacle that satisfies the assumption.

Convex Sphere Worlds

Topology Of The Obstacle Set - Strong Convexity Condition

Assumption 2 (Strong Convexity)

The Jacobian matrix $\mathbf{J}_{\mathbf{P}_{\partial\mathcal{X}}}(x)$ of the metric projection of any stationary point $x \in \mathcal{E}$ onto the boundary $\partial\mathcal{X}$ of the free-space satisfy

$$\mathbf{J}_{\mathbf{P}_{\partial\mathcal{X}}}(x) \prec \frac{||x_d - \mathbf{P}_{\partial\mathcal{X}}(x)||}{\epsilon + ||x_d - \mathbf{P}_{\partial\mathcal{X}}(x)||}\mathbf{I}_n, \quad \forall x \in \mathcal{X}, \tag{18}$$

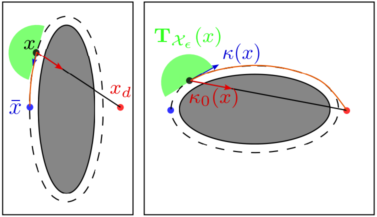


Figure: (Left) shows a flat obstacle, as viewed from the position of the vehicle, for which its trajectory converges to the undesired equilibrium.(Right) shows a strongly convex obstacle where the trajectory converges to the desired goal x_d.

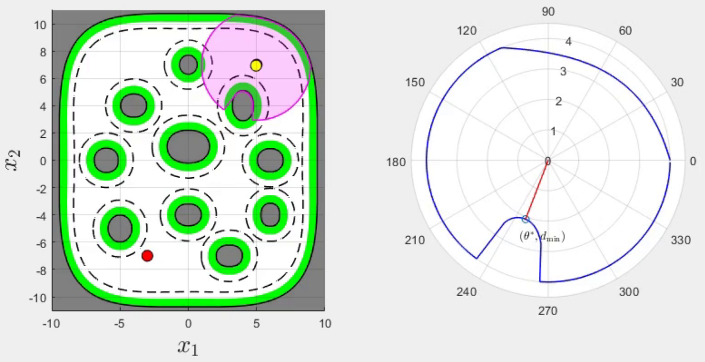
Theorem 3

In addition, if Assumption 3 holds, then

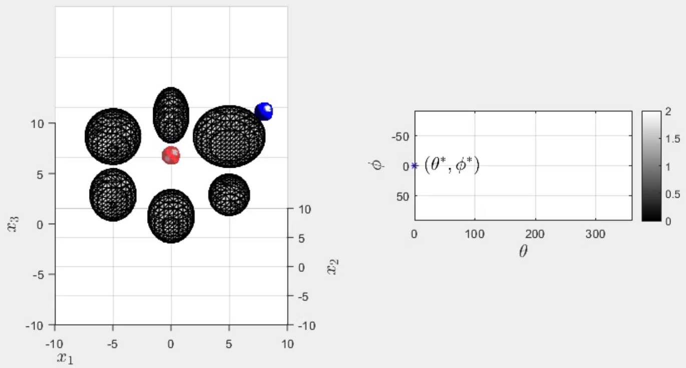
1. all the undesired equilibria $\bar{x} \in \mathcal{E}$ are unstable, and
2. the desired equilibrium x_d is locally exponentially stable and **almost globally asymptotically stable**.

Numerical Simulation

2D



Numerical Simulation
3D



Conclusion - Future Works

- Smooth feedback control law that guarantees safe navigation.
- Ensure AGAS under some topological conditions.
- Sensor-based and computationally efficient controller.

Future works:

- Relaxing the strong convexity assumption.
- Extending to higher-order dynamics.

Thank You!