# OPTIMAL AND GLOBAL AUTONOMOUS NAVIGATION IN ENVIRONMENTS WITH CONVEX OBSTACLES

Spine title: Optimal and Global Autonomous Navigation in Environments with Convex Obstacles

(Thesis format: Monograph)

by

ISHAK CHENIOUNI

Graduate Program
in
Electrical and Computer Engineering

A thesis submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy (Ph.D.)

Department of Electrical and Computer Engineering
Lakehead University

# Abstract

Motion planning for autonomous navigation in unknown environments cluttered with obstacles is a fundamental challenge in robotics, requiring efficient, safe, and reliable strategies for path planning. This thesis introduces two novel autonomous navigation strategies for vehicles operating in static, unknown $n$-dimensional environments cluttered with convex obstacles. The first strategy proposes a continuous feedback controller that steers a vehicle safely to a target destination in a *quasi-optimal* manner within a "sphere world," where each obstacle is enclosed by a sphere-shaped boundary. Under this approach, the robot avoids obstacles by navigating along the shortest path on the surface of the cone enclosing the obstacle and proceeds directly toward the target when no obstacles obstruct the line of sight. This controller guarantees almost global asymptotic stability in two-dimensional (2D) environments under specific obstacles configurations. An extension of this method is also developed for real-time navigation in unknown, static 2D environments with sufficiently curved convex obstacles, maintaining the same stability guarantees. Simulation and experimental results demonstrate the practical effectiveness of this approach in navigating real-world environments.

While the first strategy ensures almost global asymptotic stability only under specific conditions related to the obstacles configuration and for 2D environments, the second strategy aims to provide a more robust solution with stronger stability guarantees. This second strategy introduces a hybrid feedback controller designed to navigate a vehicle in static $n$-dimensional Euclidean spaces cluttered with spherical obstacles. This approach ensures safe convergence to a predefined destination from any initial position within the obstacle-free workspace while optimizing obstacle avoidance. A novel switching mechanism is proposed to alternate between two operational modes: the *motion-to-destination* mode and the *obstacle-avoidance* mode, ensuring global asymptotic stability regardless of the obstacles' configuration. Numerical simulations in both known and unknown 2D and 3D environments, along with experimental validation in a 2D setting, demonstrate the effectiveness the proposed approach.

These strategies provide robust solutions for autonomous navigation in static, unknown environments, contributing to the advancement of safe, efficient, and optimal motion planning techniques for robotic systems in complex, obstacle-laden spaces.

*To all those whom I love...*

# Acknowledgements

First and foremost, all praise and thanks be to Allah, the bestower of all blessings.

I extend my heartfelt appreciation to my thesis supervisor, Professor Abdelhamid Tayebi, for his unwavering mentorship, insightful guidance, and steadfast support throughout my research journey. His immense passion for research and commitment to excellence have been instrumental in shaping my work on this thesis and will undoubtedly continue to inspire my future endeavors. The principles of integrity and rigor he embodies have profoundly influenced my academic pursuits and personal development, making me a better individual.

I also wish to acknowledge my co-supervisor, Dr. Soulaimane Berkane, for his engaging discussions and invaluable contributions, which greatly enriched my research and aided in the successful completion of this thesis.

I am grateful to the examination committee for their time and consideration in reviewing my work.

A heartfelt thank you goes to my friends who stood by me throughout this journey, especially those who have shared these amazing five years with me. Your encouragement and camaraderie have been invaluable.

Lastly, I express my deepest gratitude to my family for their boundless love and support. I am particularly indebted to my wife for her patience, understanding, and encouragement. Most importantly, I thank my parents for being my guiding light in times of uncertainty and my source of hope during challenging moments.

# Contents

**Abstract** **i**

**Acknowledgements** **iii**

**List of Figures** **vii**

**List of Tables** **xiii**

**List of Abbreviations** **xiv**

**List of Symbols** **xv**

**1 Introduction** **1**

1.1 Motivation . . . . . 1

1.2 Literature review . . . . . 2

1.2.1 Plan-and-track approach . . . . . 2

1.2.2 Feedback-based approach . . . . . 3

1.3 Thesis contributions . . . . . 12

1.4 Thesis outline . . . . . 14

**2 Background and Preliminaries** **15**

2.1 General notations . . . . . 15

2.2 Projection maps . . . . . 15

2.3 Subsets of the Euclidean space . . . . . 16

2.4 Hybrid dynamical systems . . . . . 18

**3 Safe and Quasi-Optimal Autonomous Navigation in Environments with Convex Obstacles** **21**

3.1 Introduction . . . . . 21

3.2 Problem formulation . . . . . 23

3.3 Sets definition . . . . . 24

3.4 Control design . . . . . 25

3.4.1 Single obstacle case . . . . . 25

3.4.2 Multiple obstacles case . . . . . 26

3.4.3 Characterization of the generated trajectories . . . . . 27

3.5 Safety and stability analysis . . . . . 30

3.6 Sensor-based navigation using a 2D LiDAR range scanner . . . . . 37

3.6.1 Convex obstacles . . . . . . . . . . 42
3.7 Numerical simulations . . . . . . . . . . 44
3.7.1 Sensor-based implementation in *a priori* unknown environments . 50
3.8 Experimental validation . . . . . . . . . . 53
3.8.1 Experimental settings . . . . . . . . . . 54
3.8.2 Experimental results . . . . . . . . . . 55

**4 Hybrid Feedback Control for Global Navigation with Locally Optimal Obstacle Avoidance in $n$-Dimensional Spaces 57**
4.1 Introduction . . . . . . . . . . 57
4.2 Problem formulation . . . . . . . . . . 58
4.2.1 Sets definition . . . . . . . . . . 59
4.3 Main results . . . . . . . . . . 60
4.3.1 Control design . . . . . . . . . . 60
4.3.2 Safety and stability analysis . . . . . . . . . . 66
4.3.3 Continuity and optimality . . . . . . . . . . 67
4.4 Sensor-based implementation . . . . . . . . . . 71
4.4.1 Two-dimensional spaces . . . . . . . . . . 71
4.4.2 Three-dimensional spaces . . . . . . . . . . 73
4.5 Numerical simulations . . . . . . . . . . 75
4.5.1 Implementation with global knowledge of the environment . . . . 75
4.5.2 Sensor-based implementation . . . . . . . . . . 78
4.6 Experimental validation . . . . . . . . . . 80

**5 Conclusions 85**
5.1 Summary . . . . . . . . . . 85
5.2 Perspectives . . . . . . . . . . 86

**Bibliography 88**

**A Proofs of Chapter 3 96**
A.1 Proof of Lemma 3.1 . . . . . . . . . . 96
A.2 Proof of Lemma 3.2 . . . . . . . . . . 97
A.3 Proof of Lemma 3.3 . . . . . . . . . . 98
A.4 Proof of Lemma 3.5 . . . . . . . . . . 99
A.5 Proof of Theorem 3.1 . . . . . . . . . . 100
A.6 Proof of Lemma 3.6 . . . . . . . . . . 101
A.7 Proof of Lemma 3.7 . . . . . . . . . . 101
A.8 Proof of Theorem 3.2 . . . . . . . . . . 104
A.9 Proof of Lemma 3.8 . . . . . . . . . . 104
A.10 Proof of Theorem 3.3 . . . . . . . . . . 104
A.11 Proof of Lemma 3.9 . . . . . . . . . . 106
A.12 Proof of Theorem 3.4 . . . . . . . . . . 106

**Appendices 96**

**B Proofs of Chapter 4** **108**
B.1 Proof of Lemma 4.2 . . . . . . . . . . . . . . . . . . . . . . . . . . . . 108
B.2 Proof of Theorem 4.1 . . . . . . . . . . . . . . . . . . . . . . . . . . . 108
B.3 Proof of Lemma 4.3 . . . . . . . . . . . . . . . . . . . . . . . . . . . . 112
B.4 Proof of Proposition 4.1 . . . . . . . . . . . . . . . . . . . . . . . . . 113

# List of Figures

1.1 The shortest path in a tangent visibility graph from two different initial positions. The initial positions are represented by red squares, and the destination is represented by a red dot. The blue trajectories correspond to the shortest path to the destination. . . . . . . . . . . . . . . . . . . . . 4

1.2 Navigation under APF-based control. In the left figure (a), the robot, starting from the initial position (green dot), converges safely to the destination (red dot). In the right figure (b), the robot is trapped in a local undesired minimum of the APF. . . . . . . . . . . . . . . . . . . . . . . 4

1.3 Contour plot of a NF in a sphere world with five internal obstacles. In the left Figure (a), the contour plot shows the polarity of the NF with the target (red dot) as the unique minimum. In the right figure (b), the contour plot shows three local minima (pink crosses). . . . . . . . . . . . 6

1.4 Generation of saddle points (blue dots) by the negative gradient of a NF in a sphere world. The attraction regions of the saddle points are represented by the red segments and the target by the red dot. . . . . . . . . . . . . 6

1.5 Contour plots of a NF in a star world with five internal obstacles. . . . . 7

1.6 Local obstacle-free space (green) of a point robot (blue) obtained by the maximum margin separating hyperplanes of the robot body (blue) and obstacles (grey). The metric projection of the red target onto the local obstacle-free space is represented by a black dot. In the left figure (a), the robot assumes knowledge of adjacent obstacles to construct a local obstacle-free space. In the right figure (b), the robot has an onboard LiDAR that scans a limited region (orange) and then can build a local obstacle-free space from the detected obstacles portions. . . . . . . . . 9

1.7 Illustration of the obstacle curvature condition in (Arslan and Koditschek, 2019, Assumption 2) with elliptical obstacles. The condition is that the obstacle must be entirely contained within the ball whose center is the target (red) and whose radius is equal to the distance between the target and the stationary point (blue). Obstacle $\mathcal{O}_1$ satisfies the condition, while obstacle $\mathcal{O}_2$ does not. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 10

1.8 A DistBug-like trajectory in a workspace cluttered with elliptical obstacles. 11

1.9 Safe navigation under a hybrid Bug-like strategy in a 2D workspace cluttered with elliptical obstacles. The black portions of the trajectory represent the stabilization mode, the orange portions of the trajectory represent the avoidance mode, the black dot represents the initial position, and the red dot represents the target. . . . . . . . . . . . . . . . . . . . . . . . 12

3.1 2D representation of the sets in Section IV. . . . . . . . . . . . . . . . . . . . 25

3.2 Successive projections of the control $u_d$ in a two-dimensional sphere world with four obstacles. At step zero, the nominal control $u_0 = u_d$ is projected onto the cone enclosing obstacle $\mathcal{O}_2$, where obstacle $\mathcal{O}_2$ is the closest blocking obstacle to the destination $x_d$ among the set $\mathcal{LO}(x, x_d)$ of blocking obstacles between the position $x$ and the destination $x_d$. The resultant intermediary control $u_1$ is considered for the projection at step 1 where $\hat{c}_2$, the point at which the line directed by $u_1$ is tangent to $\mathcal{O}_2$, plays the role of the destination. The same operation is repeated until step 3 where the set of blocking obstacles $\mathcal{LO}(x, \hat{c}_4)$ is empty. The velocity control will be then $u = u_3$. The sets $\mathcal{L}_i$ represent the unstable undesired equilibria associated with each obstacle. . . . . . . . . . . . . . . . . . . . . . . . . . 28

3.3 Quasi-optimal trajectory in 2D workspace. . . . . . . . . . . . . . . . . . . 30

3.4 Optimal and quasi-optimal trajectories shown in green and blue, respectively. 31

3.5 Two-dimensional illustration of Bouligand's tangent cones in a sphere world. The pink regions represent Boulgand's tangent cones $\mathcal{T}_{\mathcal{F}}(x)$ at position $x$. Inside the workspace, the Bouligand tangent cone is the Euclidean space, and on the boundary of the workspace, it is a half-space. . . . . . 32

3.6 Illustration of obstacles that generate, and those that do not generate, undesired equilibria. Obstacles 2 and 3 generate undesired equilibria, but obstacles 1 and 4 do not. Unlike obstacles 1 and 4, obstacle 2 does not satisfy the conditions in Lemma 3.5, where although obstacle 2 is crossed by the central half-line $\mathcal{L}_i^e$ associated with obstacle 3, its center is not included in the hats of the cones enclosing the obstacles preceding it (*i.e.,* obstacle 4, 1 and 3). . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 34

3.7 Illustration of invariant cells and nests. The figure on the left shows an example of four cells associated with four obstacles, where the boundaries of each cell are formed by the boundary of the free space (blue arcs) and the undesired equilibria (red segments). The figure on the right shows the nests obtained from the cells in the figure on the left. $\mathbf{Nest}_1$ and $\mathbf{Nest}_2$ are equal to $\mathbf{Cell}_3$ and $\mathbf{Cell}_1$, respectively. $\mathbf{Nest}_3$ is the union of the two adjacent cells, $\mathbf{Cell}_2$ and $\mathbf{Cell}_4$, and $\mathbf{Nest}_0$ is a special nest that includes all the undesired equilibria not included in the other nests. . . . . . . . . 35

3.8 Quasi-non-navigable two-dimensional space. . . . . . . . . . . . . . . . . . 36

3.9 The sensor-based control strategy. Figure (a) shows the (purple) arcs returned by the sensor from the obstacles detected at position $x$. Figure (b) shows the extended version of the detected arcs (blue arcs) and the virtual cone enclosing the active extended arc (green arc) with the necessary practical parameters for the sensor-based control law. . . . . . . . . . . 38

3.10 Safety consideration in a sensor-based case. In Fig. (a), the projection $u$ of $(x_d - x)$ onto the cone enclosing the arc $L_1$ of the list $\mathcal{LA}(x)$ fails to satisfy the safety condition where $u$ crosses the obstacle. In Fig. (b), the projection lies on the active arc $\tilde{L}^*$ of the list $\mathcal{LA}_e(x)$ and meets the safety condition. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 39

3.11 Illustration of the workspace for the sensor-based case. Fig. (a) shows the truncated shadow regions of each obstacle where obstacles $\{2, 3, 4\}$ are the progeny of obstacle 1, and obstacle $\{5\}$ is the progeny of obstacle 4. Fig. (b) highlights the practical shadow regions of each obstacle in pink where the union of theses regions represents the practical blind set while the white region represents the practical visible set. . . . . . . . . . . . 40
3.12 A scenario of a discontinuity occurrence in a 2D workspace populated by two obstacles at time $t = t'$. Initially, the active arc is the arc detected from obstacle 1, but arriving at position $x(t')$, it switches to the arc detected from obstacle 2, changing the robot's heading abruptly from the tangent to obstacle 1 to the tangent to obstacle 2 (pointing towards the endpoint $\tilde{c}^*$), creating a discontinuity in the control. . . . . . . . . . . . . . . . . . . 42
3.13 Representation of the shadow region and the practical shadow region of a convex obstacle. In the figure on the left, the obstacle satisfies the curvature condition, while in the figure on the right, the obstacle does not satisfy this condition. . . . . . . . . . . . . . . . . . . . . . . . . . . . 43
3.14 Smoothing of polygonal obstacle corners. . . . . . . . . . . . . . . . . . . 44
3.15 Trajectories generated by our approach, Algorithm 1, (black solid trajectories) and the optimal path found by DA in tangent visibility graphs (blue dashed trajectories), in 10 different environments. The target location is indicated with a red dot. Only 10 trajectories among 100 starting from randomly selected initial positions were plotted, while the summary of the complete experiments is reported in Table 3.1, where each subfigure illustrates the results in the corresponding space in Table 3.1. Spaces (5), (6), and (8) show examples where the quasi-optimal trajectories do not match the shortest path obtained by the DA. . . . . . . . . . . . . . . . . . . . 46
3.16 Effect of the nested projections on the optimality of the generated trajectory. In Fig. (a), a single obstacle is considered, and the trajectory generated by our approach corresponds to the DA trajectory (shortest path). In Fig. (b), a second obstacle is considered. The trajectory generated by our approach differs from the DA trajectory (shortest path). . . . . . . . 47
3.17 Trajectories generated by our approach, SH, NF and DA in a two-dimensional sphere world. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 48
3.18 Robot safe navigation from eighteen different initial positions in a three-dimensional sphere world. . . . . . . . . . . . . . . . . . . . . . . . . . . . 49
3.19 Comparison of paths generated by our approach, SH, and NF in a single two-dimensional sphere world. . . . . . . . . . . . . . . . . . . . . . . . . 49
3.20 Invariance of the nests. . . . . . . . . . . . . . . . . . . . . . . . . . . . . 49
3.21 Navigation in 2D sphere world. . . . . . . . . . . . . . . . . . . . . . . . . 50
3.22 Sensor-based navigation in unknown 2D convex worlds. . . . . . . . . . . 52
3.23 Time-stamped shots of Turtlebot 3 navigating a Gazebo world. . . . . . . 53
3.24 RPLIDAR-A1M8 onboard of Turtlebot 4. The figure on the left shows the Turtlebot 4 and the RPLIDAR-A1M8. The figure on the right illustrates the robot frame and the LiDAR frame. . . . . . . . . . . . . . . . . . . . . 54

3.25 A schematic representation of the network and data communication flow in our experimental setup. . . . . . . . . . . . . . . . . . . . . . . . . . . 55

3.26 Safe navigation of Turtlebot 4 in an environment with four boxes and four punching bags as obstacles. The top figure shows the configuration of the workspace with a highlighted initial and final positions. The bottom figure is a plot of the robot's trajectory in an orthographic projection top view of the workspace. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 56

4.1 2D representation of the sets in Section 4.2.1. . . . . . . . . . . . . . . . . . 61

4.2 2D illustration of the flow and jump sets for the *motion-to-destination* mode associated with obstacle $\mathcal{O}_k,\ k \in \mathbb{I}$. . . . . . . . . . . . . . . . . . . . . 62

4.3 Construction of the control in the *obstacle-avoidance* mode for a 2D case. 64

4.4 2D illustration of the flow and jump sets for the *obstacle-avoidance* mode associated with obstacle $\mathcal{O}_k,\ k \in \mathbb{I}$. . . . . . . . . . . . . . . . . . . . . . . . 65

4.5 Illustration of the property in Lemma 4.3 in a 3D space. In the left figure (a), the virtual destinations $x_k^{\pm 1}$ belong to the plane $\mathcal{PL}(x_d, c_k, x(t_0^k, j_0^k))$, resulting in a trajectory belonging to the same plane for the obstacle $\mathcal{O}_k$, where the *obstacle-avoidance* mode is represented by the orange curve and the *motion-to-destination* mode is represented by the blue curve. In the middle figure (b), the virtual destinations $x_k^{\pm 1}$ do not belong to the plane $\mathcal{PL}(x_d, c_k, x(t_0^k, j_0^k))$, resulting in a trajectory that does not belong to the plane $\mathcal{PL}(x_d, c_k, x(t_0^k, j_0^k))$. The right figure (c) shows that the trajectory generated in figure (b) does not belong to a single plane but to two different planes, as the virtual destinations are not on the plane $\mathcal{PL}(x_d, c_k, x(t_0^k, j_0^k))$. 69

4.6 Effects of selecting the navigation mode according to Proposition 4.1. In the left figure (a), when first entering the jump set $\mathcal{J}_k^0$ through the hysteresis region (pink region), the mode $m$ switches to *obstacle-avoidance* mode where the blue trajectory is generated by selecting the virtual destination $x_k^1$ and the green trajectory is generated by selecting the closest virtual destination $x_k^{-1}$ to the robot's position. Both trajectories are smooth, but the blue trajectory is longer than the green trajectory. In the right figure (b), the blue trajectory is longer and non-smooth. The green (blue resp.) region in the jump set $\mathcal{J}_k^0$ is where, if the robot operates in the *motion-to-destination* mode, the mode $m$ can only jump to $m = -1$ ($m = 1$ resp.). . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 71

4.7 Obstacle reconstruction from sensor data. . . . . . . . . . . . . . . . . . . 74

4.8 Robot navigation trajectories from ten different initial positions in a 2D workspace cluttered with circular obstacles. The blue trajectories are generated by our hybrid approach and the black trajectories are generated by the hybrid approach proposed in (Berkane et al., 2022). The target is represented by the red dot. . . . . . . . . . . . . . . . . . . . . . . . . . . 76

4.9 Robot navigation trajectories from ten different initial positions in a 3D workspace cluttered with spherical obstacles. The blue trajectories are generated by our hybrid approach and the black trajectories are generated by the hybrid approach proposed in (Berkane et al., 2022). The target is represented by the red dot. . . . . . . . . . . . . . . . . . . . . . . . . . . 77

4.10 Robot navigation trajectories from ten different initial positions in a 2D workspace cluttered with circular obstacles. The trajectories generated by our hybrid approach are represented by the blue trajectories, the trajectories generated by the quasi-optimal approach are represented by the green dashed curves, and the target is represented by the red dot. The red dash-dotted line segments represent the sets of undesired equilibria of the quasi-optimal control and the red areas represent the attraction regions of the undesired equilibria (nests). . . . . . . . . . . . . . . . . . . . . . . . 79

4.11 Robot navigation trajectories from ten different initial positions in a 3D workspace cluttered with spherical obstacles. The trajectories generated by our hybrid approach are represented by the blue trajectories, the trajectories generated by the quasi-optimal approach are represented by the green dashed trajectories, and the target is represented by the red dot. . 80

4.12 Robot navigation trajectories from ten different initial positions in a 2D workspace cluttered with circular obstacles. The blue trajectories are generated by the sensor-based implementation of our hybrid approach, and the orange trajectories are generated by our hybrid approach when the environment is *a priori* known. The red dot represents the target. . . . . 81

4.13 Robot navigation trajectories from ten different initial positions in a 3D workspace cluttered with spherical obstacles. The blue trajectories are generated by the sensor-based implementation of our hybrid approach, and the orange trajectories are generated by our hybrid approach when the environment is *a priori* known. The red dot represents the target. . . 82

4.14 The top figure shows the workspace configuration with initial and final positions highlighted. The bottom figure is a plot of the robot's trajectory in an orthographic projection of the workspace's top view. . . . . . . . . 83

4.15 a) Example navigation scenario in a priori unknown environment, showing the robot's trajectory generated by the proposed hybrid feedback control approach (blue) compared to alternative methods. (b) Performance comparison highlighting the path length and computational efficiency of the proposed approach. The proposed approach generates paths similar to our previously proposed quasi-optimal (QO) approach (Cheniouni et al., 2023a) while avoiding the issue of undesired equilibria in QO approach. The details of this experiment are reported in Section 4.6. The complete experiment can be visualized in the video available online `https://youtu.be/KzUNLwQ5lMo`. . . . . . . . . . . . . . . . . . . . . . . 84

5.1 Nest generation (green region) in a 3D sphere world with two obstacles. . 87

5.2 Generation of the largest nest (union of the colored planar regions representing cells) in a 3D sphere world with multiple obstacles. . . . . . . . . 87

A.1 Shortest path in a single-obstacle sphere world. . . . . . . . . . . . . . . . . 97

A.2 The left figure shows that for all position $x \in \mathcal{AR}_1 \setminus \mathring{\mathcal{H}}(y, c_1)$, the control cannot point inside $\mathcal{H}(y, c_1)$. In the right figure, obstacle 2 is placed such that $c_2 \in \mathring{\mathcal{H}}(y, c_1)$, the intermediary control $u_1$ cannot point inside $\mathcal{H}(y, c_1)$ and $y = x^*_{2,1}$. Then, the intermediary control $u_1$ cannot point into the center $c_2$ at any position $x \in \mathcal{AR}_2$ which implies that $\mathcal{L}_2$ is an empty set. 100

A.3 Illustration of the nests (attraction regions of the undesired equilibria.) . 103

# List of Tables

3.1 Number of perfect matches between the paths generated by our approach (Algorithm 1) and those found by DA in tangent visibility graphs, for 100 runs with 100 randomly selected initial positions. . . . . . . . . . . . . . . 45

3.2 The relative length difference of the paths, shown in Fig. 3.17, generated by the NF and SH approaches with respect to our approach. . . . . . . 48

3.3 The relative length difference of the paths, shown in Fig. 3.21, generated by Algorithm 2, with two sensor radial ranges ($R = 2m$ and $R = 4m$), with respect to Algorithm 1. . . . . . . . . . . . . . . . . . . . . . . . . . 51

3.4 Experimental setup and control parameters. . . . . . . . . . . . . . . . . 55

4.1 The relative length difference of the paths, shown in Fig. 4.8, generated by the hybrid approach proposed in (Berkane et al., 2022) with respect to our hybrid approach in a 2D workspace. . . . . . . . . . . . . . . . . . . . 77

4.2 The relative length difference of the paths, shown in Fig. 4.9, generated by the hybrid approach proposed in (Berkane et al., 2022) with respect to our hybrid approach in a 3D workspace. . . . . . . . . . . . . . . . . . . . 77

4.3 The relative length difference of the paths, shown in Fig. 4.10, generated by the hybrid approach with respect to the quasi-optimal approach in a 2D workspace. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 78

4.4 The relative length difference of the paths, shown in Fig. 4.11, generated by the hybrid approach with respect to the quasi-optimal approach in a 3D workspace. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 78

# List of Abbreviations

| | | |
|---|---|---|
| APF | – | Artificial Potential Field |
| VG | – | Visibility Graph |
| TVG | – | Tangent Visibility Graph |
| RRT | – | Rapidly Exploring Random Tree |
| PRM | – | Probabilistic Roadmaps |
| FMT | – | Fast Marching Tree |
| NF | – | Navigation Function |
| GAS | – | Globally Asymptotically Stable |
| AGAS | – | Almost Globally Asymptotically Stable |
| QP | – | Quadratic Program |
| SH | – | Separating Hyperplane |
| MSH | – | Modified Separating Hyperplane |

# List of Symbols

| | | |
|---|---|---|
| $\mathbb{R}$ | – | the set of real numbers |
| $\mathbb{R}_{\geq 0}$ | – | the set of non-negative real numbers |
| $\mathbb{R}_{>0}$ | – | the set of positive real numbers |
| $\mathbb{N}$ | – | the set of natural numbers |
| $\mathbb{R}^n$ | – | the set of $n$-dimensional vectors or $n$-dimensional Euclidean space |
| $\mathbb{S}^{n-1}$ | – | the set of $n$-dimensional unit vectors |
| $\mathcal{T}_x\mathcal{Q}$ | – | the tangent space of smooth manifold $\mathcal{Q}$ at $x \in \mathcal{Q}$ |
| $f : M \to N$ | – | a mapping from $M$ to $N$ |
| $F : M \rightrightarrows N$ | – | a set-valued mapping from $M$ to $N$ |
| $x \mapsto f(x)$ | – | a mapping of an element $x$ to a map $f(x)$ |
| $\nabla_x f$ | – | the gradient of $f$ at $x$ |
| $\dot{x}$ | – | the derivative, with respect to time, of the state $x$ |
| $x^+$ | – | the state of a hybrid system after a jump (state reset) |
| $\lVert x \rVert$ | – | the Euclidean norm of a vector $x \in \mathbb{R}^n$ |
| $\arg\min$ | – | the argument of the minimum |
| $I_n$ | – | an $n \times n$ identity matrix |

# Chapter 1

# Introduction

## 1.1 Motivation

Autonomous robots have become an integral part of modern technology, permeating a variety of domains, from daily life tasks (Yamazaki et al., 2012) and industrial operations (Czimmermann et al., 2022) to more specialized fields such as commercial applications (Noh et al., 2024), aerial surveillance (Savkin and Huang, 2022), underwater exploration (Vidal et al., 2017), space exploration (Guzzetti and Baoyin, 2019), and ground mobility (Jiang et al., 2022). In the medical field, these robots are also increasingly utilized in critical procedures such as surgery and drug administration through microrobots (Haidegger, 2019; Jang et al., 2019). The transformative impact of these technologies is evident in sectors like healthcare, where autonomous systems have enhanced the precision of minimally invasive surgeries, and in industries where robots are streamlining production and logistics.

The widespread adoption of autonomous robots across various environments emphasizes the need for robust control systems and advanced planning algorithms. In aerial and underwater robotics, for example, real-time path planning is essential for ensuring safe navigation, while in ground mobility and spacecraft exploration, dynamic adaptation to unstructured environments is crucial. In rescue missions, the shortest path is another necessity for successful navigation. Furthermore, microrobots used in drug delivery must precisely navigate the human body to administer medication at targeted locations. Each of these robotic tasks requires sophisticated control schemes and planners to ensure that robots can navigate unknown environments, perform tasks accurately, and satisfy optimality criteria. Whether it is a drone surveying a vast landscape, a robot exploring the depths of the ocean, or a microrobot delivering medication into a patient's bloodstream, the success of these operations hinges on advanced algorithms that optimize task execution while guaranteeing safety and reliability.

## 1.2 Literature review

The development of robotic autonomous navigation began in the mid-20th century, spurred by advancements in computing and control systems. Early research focused on basic mobile robots such as "Shakey" robot in the 1960s, which could navigate and make decisions based on tactile sensors and a camera. In the 1970s and early 1980s, several basic algorithms for path planning and collision avoidance have been developed, relying on ultrasonic range finders. The 1980s saw the introduction of Oussama Khatib's groundbreaking Artificial Potential Field (APF) approach, which transformed the robotic navigation. This reactive navigation approach represented a shift from traditional navigation methods. Several extensions and modifications to the APF approach have been introduced over the years and new sophisticated autonomous navigation techniques have emerged due to the advancements in processing and sensors technologies, intelligent algorithms and control theory.

The autonomous robot navigation strategies can mainly be grouped into two approaches; the *plan-and-track approach* and the *feedback-based approach*. The plan-and-track approach is a two-stage process in which a collision-free path is first generated based on a geometric model of the workspace and then tracked by a low-level control. The generated path is eventually refined to satisfy differential constraints in order to design a trajectory that can be tracked by feedback control. Path optimality is another interesting problem frequently considered when generating paths with this approach, where the shortest path problem has received the most attention. On the other hand, the feedback-based approach is a reactive (sensor-based) approach allowing to directly design a feedback control input (low-level control) that steers the robot to a target location along a collision-free trajectory without any *a priori* path planning.

### 1.2.1 Plan-and-track approach

Path planning is the most crucial phase in the plan-and-track approach. In this phase, a roadmap of the obstacle-free configuration space is constructed, and then search algorithms such as Dijkstra (Dijkstra, 1959) or A* (A star) (Hart et al., 1968) determine the shortest feasible path connecting a valid initial/final goal pair. Path planners adopt two methods for constructing configuration space roadmaps: combinatorial and sampling-based (LaValle, 2006). The combinatorial methods rely on an exact and explicit representation of the configuration space, leading to complete algorithms and exact shortest paths. One of the earliest combinatorial methods is the visibility graph (VG) approach (Nilsson, 1969), designed for two-dimensional environments cluttered with polygonal obstacles. It connects every two visible vertices between all possible pairs of obstacles (the line segment connecting the vertices must belong to the obstacle-free space), thereby forming a graph. By connecting a valid initial/final goal pair to the resulting graph, it is possible to determine the shortest path, shown to belong to the VG (Nilsson, 1969), using search algorithms. An optimized version of the VG was proposed in (Rohnert, 1986), called the tangent visibility graph (TVG), where the graph is reduced to the minimal sub-graph containing the shortest path (Kamon et al., 1998).

In this optimized graph, only the common tangents (also called bitangents) between each pair of obstacles are taken into account. The TVG was extended in (Laumond, 1987; Liu and Arimoto, 1992) to solve the shortest path problem in two-dimensional environments with curved obstacles, as illustrated in Fig. 1.1. Unfortunately, this approach is limited to 2D environments (LaValle, 2006; Latombe, 2012). It was shown in (Canny and Reif, 1987) that the shortest path problem in 3D environments with polyhedral obstacles is NP-hard, and the best that can be obtained is an approximation to the shortest path (Latombe, 2012). The sampling-based methods do not require an explicit representation of the obstacle regions but use a sampling scheme to explore the configuration space (LaValle, 2006), resulting in weaker notions of completeness, such as probabilistic completeness in the sense that the probability that the planner fails to return a solution, if one exists, decays to zero as the number of samples approaches infinity (Barraquand et al., 1996). In this category, we find single-query planning algorithms, such as the rapidly exploring random trees (RRTs) (LaValle, 1998), which gradually construct a search tree for a pair (initial position/final position), or multiple-query algorithms, such as the probabilistic roadmaps (PRMs) (Kavraki et al., 1996), which connect several (initial position/final position) pairs to a configuration space roadmap and then search for a feasible path. The path planning algorithms such as RRTs, PRMs, expansive space trees (Hsu et al., 1997; Phillips et al., 2004), Lazy-PRM (Bohlin and Kavraki, 2000), sampling-based roadmap of trees (Plaku et al., 2005), or rapidly exploring roadmap (Alterovitz et al., 2011) guarantee the probabilistic completeness but generally provide non-optimal paths. The variants RRT* and PRM* (Karaman and Frazzoli, 2011), RRT$^{\#}$ (Arslan and Tsiotras, 2013), or the fast marching tree (FMT*) (Janson et al., 2015) are asymptotically optimal, that is, the cost (often the arc length function) of the returned solution converges probabilistically to the optimum (Janson et al., 2018). Reactive path planners are local approaches that rely on sensors to plan local paths in real-time. Bug algorithms are examples of reactive planners that operate in 2D environments under two modes of motion: motion towards the target and boundary following. The first two bug algorithms, Bug1 and Bug2, introduced in (Lumelsky and Stepanov, 1986), rely on contact sensors. Other variants have appeared since then using range sensors, such as VisBug (Lumelsky and Skewis, 1990) and TangentBug (Kamon et al., 1998). The performance of the different bug algorithms depends strongly on the geometry of the workspace, as shown in the comparative study in (Ng and Bräunl, 2007).

### 1.2.2 Feedback-based approach

A popular approach in the feedback-based category is the artificial potential field (APF) approach proposed by Khatib in (Khatib, 1986). The idea of this approach is to consider that the robot is moving in a force field where the target location generates an attractive force and the obstacles generate repulsive forces. The superposition of the potential of the respective forces (attractive and repulsive forces) constitutes the APF of the workspace. The negative gradient of the APF steers the robot safely to reach its target (at the minimum of the APF) as shown in Fig. 1.2-(a). Unfortunately, local minima are generated in some APF constructions due to the superposition of different potentials, resulting in convergence to the target not being guaranteed, as shown in Fig. 1.2-(b).

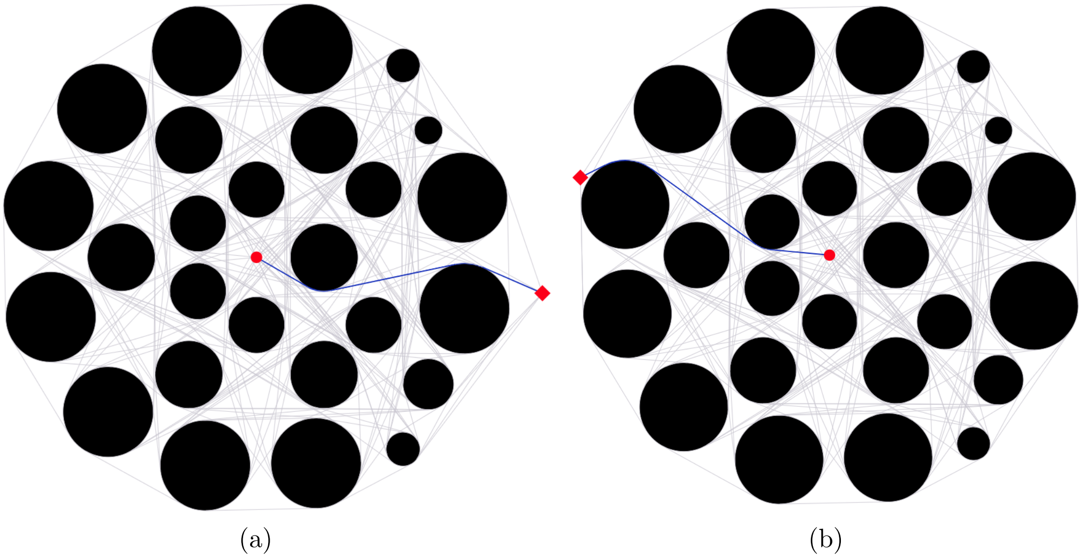


Figure 1.1: The shortest path in a tangent visibility graph from two different initial positions. The initial positions are represented by red squares, and the destination is represented by a red dot. The blue trajectories correspond to the shortest path to the destination.

The problem of local minima was solved by the navigation function (NF) approach

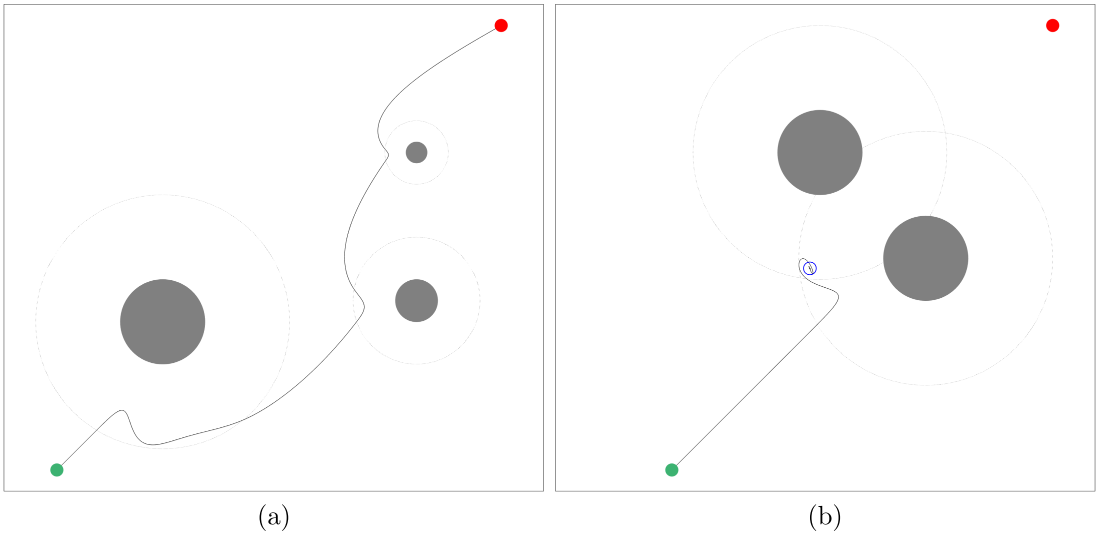


Figure 1.2: Navigation under APF-based control. In the left figure (a), the robot, starting from the initial position (green dot), converges safely to the destination (red dot). In the right figure (b), the robot is trapped in a local undesired minimum of the APF.

proposed by Rimon and Koditcheck (Koditchek and Rimon, 1990) to navigate sphere worlds[1]. The NF is a polar potential function (*i.e.,* it has a unique minimum at the target location) if a tuning parameter is large enough. The negative gradient of a well tuned NF, taken as a control input, makes the destination as a unique minimum of the NF. The example in Fig. 1.3 shows the contour plot of a NF with two different tuning parameters. In Fig. 1.3-(a), the NF is well-tuned, and the destination (red dot) is its unique minimum. In Fig. 1.3-(b), the tuning parameter is not large enough to make the NF polar, where one can observe the generation of three local minima illustrated by pink crosses. Although well-tuned NFs do not generate local minima, their negative gradient must generate at lest as many saddle points as obstacles in the workspace. The stable manifolds associated to these unstable saddle points are of zero Lebesgue measure. This topological obstruction induced by the motion space involving time-invariant continuous vector fields prevents global navigation, where, at best, almost global asymptotic stability[2] (AGAS) of the target location is achieved (Koditchek and Rimon, 1990). Therefore, NFs ensure that the target can be reached safely from any initial position in the free space except from a set of zero Lebesgue measure associated to the saddle points (undesired equilibria). These saddle points and their attraction sets are represented, respectively, by the blue dots and the red segments in Fig. 1.4. In (Rimon and Koditchek, 1992), a tunable diffeomorphism was proposed for navigation in star worlds[3] by mapping these complicated spaces to sphere worlds in which the NF was initially designed. Figure 1.5 is an example of a star world with five obstacles diffeomorphic to its model sphere world in Fig. 1.3-(a). Although the existence of a threshold for the tuning parameters of both the NF and the diffeomorphism is theoretically guaranteed, their calculation is not simple (Loizou, 2017). In (Loizou, 2011b; Loizou, 2017), a tuning-free navigation function was proposed in point worlds[4]. The navigation is accomplished with AGAS guarantees by diffeomorphically mapping sphere worlds to point worlds using the proposed tuning-free "Navigation Transformation". Given the simplicity of the point world model, a time-abstracted solution to the motion-planning problem is guaranteed. Recently, a one-step tuning-free navigation transformation from star worlds to point worlds was proposed in (Constantinou and Loizou, 2020), achieving AGAS of a target location in star worlds.

Based on the extension of the NFs to multi-robot NFs presented in (Loizou and Kyriakopoulos, 2002), a decentralized version was proposed in (Dimarogonas and Kyriakopoulos, 2007) for multi-agent systems with a limited sensing range. A tuning-free and locally computable NF was proposed in (Lionis et al., 2007) to navigate sphere worlds by restricting the influence of every obstacle to its neighborhood. Similarly, a local polynomial NF was designed in (Lionis et al., 2008) followed by a local diffeomorphism to map spaces with convex obstacles to sphere worlds. This NF is local in the sense that it only

[1] A sphere world is a ball of dimension $n$ punctured by a finite number of $n$-dimensional disjoint smaller balls (Koditchek and Rimon, 1990).

[2] An equilibrium point is almost globally asymptotically stable if it is stable and attractive from all initial conditions except from a set of zero Lebesgue measure.

[3] A star world is a ball of dimension $n$ punctured by a finite number of star-shaped sets possessing a distinguished "center point" from which rays cross their boundary once and only once (Rimon and Koditchek, 1992).

[4] The $n$-dimensional Euclidean space (or an $n$-dimensional ball of infinite radius) punctured by a finite number of points (Loizou, 2017).

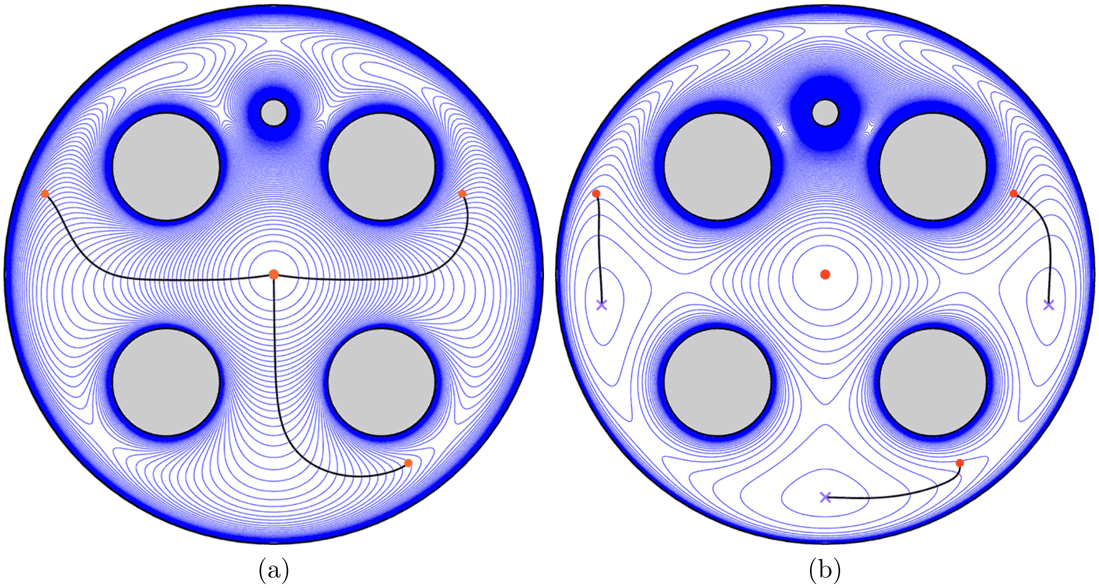


Figure 1.3: Contour plot of a NF in a sphere world with five internal obstacles. In the left Figure (a), the contour plot shows the polarity of the NF with the target (red dot) as the unique minimum. In the right figure (b), the contour plot shows three local minima (pink crosses).

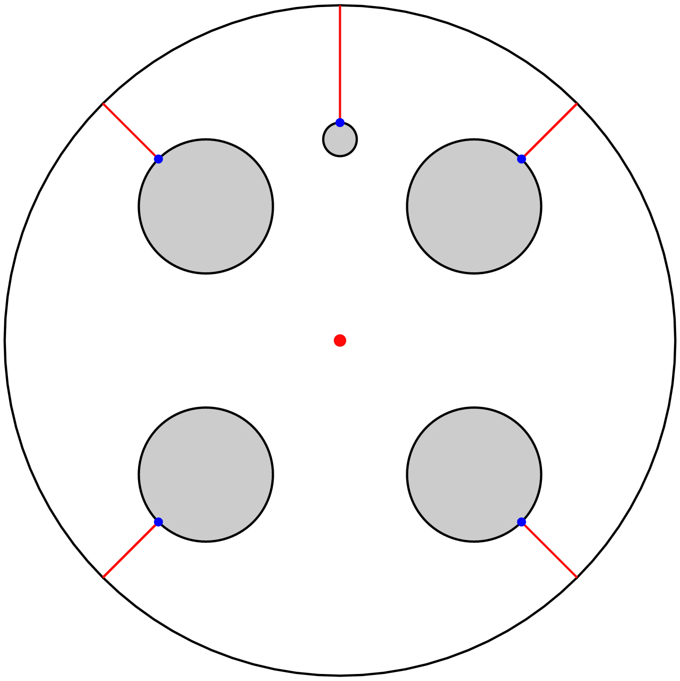


Figure 1.4: Generation of saddle points (blue dots) by the negative gradient of a NF in a sphere world. The attraction regions of the saddle points are represented by the red segments and the target by the red dot.

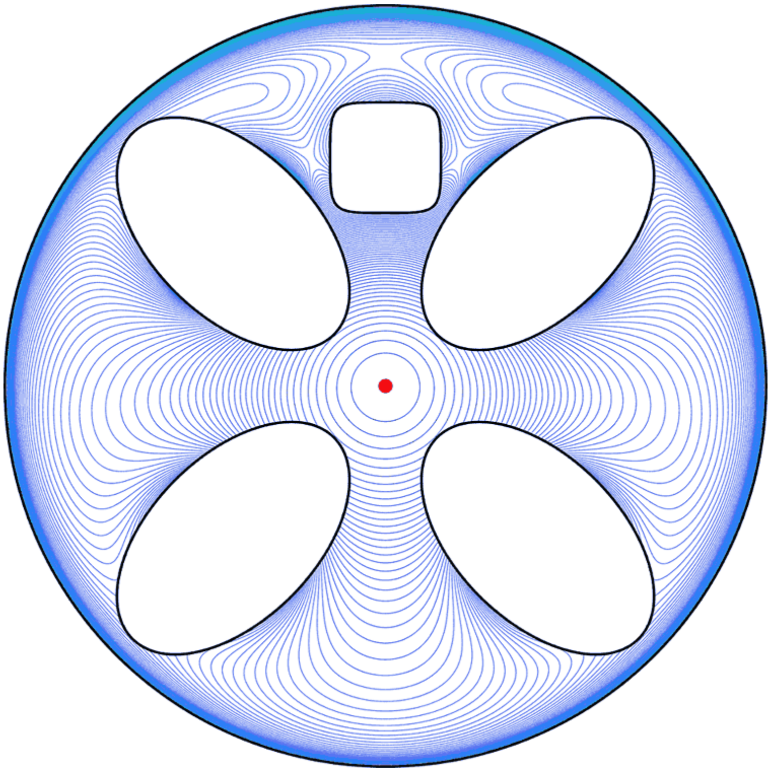


Figure 1.5: Contour plots of a NF in a star world with five internal obstacles.

requires complete information on current and adjacent obstacles. An explicit formulation of the lower bound on the tuning parameter of the Rimon-Koditcheck NF has been proposed in (Filippidis and Kyriakopoulos, 2011), followed by an automatic tuning algorithm solving the navigation problem in unknown sphere worlds. The Rimon-Koditchek NF was shown in (Filippidis and Kyriakopoulos, 2012; Filippidis and Kyriakopoulos, 2013; S. Paternain and Ribeiro, 2018) to be valid for more general worlds than sphere worlds, in particular, sufficiently curved worlds. In these worlds, the target is sufficiently far apart from the borders of the obstacles relative to their flatness (*i.e.,* the ball centered on the target, whose radius is the distance between the target and the farthest point of any given obstacle in the workspace, fully encompasses that obstacle). This curvature condition was relaxed in (Kumar et al., 2022; Kumar et al., 2020) by correcting the NF gradient so that the world appears spherical to the robot using the Hessian of the obstacle functions (functions describing the geometry of the obstacles) and the objective function (usually the Euclidean distance between the robot position and the target position). In (S. Paternain and Ribeiro, 2018), a switched controller is used for sensor-based navigation, where the NF is updated each time an obstacle is detected. However, the complete shapes of the obstacles are assumed to become known once the obstacles are detected, and the absence of local minima is not guaranteed. A stochastic extension of the NFs was proposed in (Paternain and Ribeiro, 2020) to navigate the same worlds as in the deterministic case when local and noisy sensor information of the environment is available. Recently, a tuning-free NF based on harmonic potentials has been proposed in (Loizou and Rimon, 2021) to navigate worlds topologically equivalent to sphere worlds, using symmetric sectors range scanners. These worlds are assumed to be unknown, but the complete shape of their obstacles becomes known once detected.

Another solution to the local minima problem is the use of harmonic potential functions. The authors of (Connolly et al., 1990; Connolly, 1992) drew an analogy between the navigation problem and charge distribution and fluid flow, and proposed a numerical construction of a harmonic potential function using the superposition of Dirichlet and Neumann solutions. Another construction of harmonic potential functions has been proposed in the context of real-time obstacle avoidance in (Kim and Khosla, 1992), based on the panel method, which approximates obstacles of arbitrary shape by polygonal shapes by surrounding the obstacles with virtual panels. In the same context but in dynamic environments, the solution of (Feder and Slotine, 1997) uses a global model of the environment but does not always guarantee safety and convergence. However, in static environments, safety and convergence are always guaranteed. The use of the harmonic potential functions has been extended in (Pimenta et al., 2008) to the multi-robot case. The swarm is modeled as an incompressible fluid based on smoothed particle hydrodynamics, where the global function solves the pattern generation task in the presence of static obstacles. The construction of a navigation function based on harmonic potential functions in a two-dimensional space was proposed in (Loizou, 2011a). An adaptive law is designed for a tuning parameter that ensures non-degeneracy of the critical points. In addition, this navigation function enjoys the properties of harmonic functions that allow a natural decoupling of the effects of the obstacles and the goal, but most importantly, it frees the navigation function from the offline tuning phase required in (Koditchek and Rimon, 1990). The author considered both kinematic and dynamic holonomic systems and provided almost global convergence results. In (Loizou, 2012), a navigation function based on harmonic functions has been proposed to navigate complex 3-D workspaces, wherein no systematic and constructive approach was given. Recently, the construction of a harmonic-based artificial potential field, similar to the one in (Loizou, 2011a), relying on a harmonic transformation from an arbitrary bounded 2D space to a punched disk, was proposed in (Vlantis et al., 2018). Moreover, unlike in (Loizou, 2011a), the target location can be on the boundary of the workspace. To deal with a large and complex workspace, the author partitions it into rooms where each room is associated with a transformation, and the transition from one room to another is smooth. Furthermore, all these transformations form an Atlas of harmonic maps. Although free of local minima, harmonic potentials are extremely difficult to construct in a navigation context (Vlantis et al., 2018). Indeed, the only well-structured harmonic-based artificial potential fields are not themselves harmonic, and only solve the problem of motion planning in 2D spaces (Loizou, 2011a; Vlantis et al., 2018). For the 3D case, no systematic construction has been proposed for the transformations, but only a case study has been presented in (Loizou, 2012).

In (Arslan and Koditschek, 2016), the authors proposed a reactive navigation approach by constructing a convex obstacle-free local set around the robot using the hyperplanes separating the robot from the neighboring obstacles, as shown by the green polygonal shape in Fig. 1.6-(a), and then steering the robot towards the projection of the target location onto the boundary of this convex set. The proposed feedback control ensures safe navigation in a sphere world and convergence to the destination from almost everywhere except from a set of zero Lebesgue measure. This work was extended in (Arslan and Koditschek, 2019) to navigate in environments similar to those in (S. Paternain and Ribeiro, 2018) containing sufficiently curved obstacles as illustrated in Fig. 1.7, using obstacle sensors (see Fig. 1.6b for an illustration). An extension of this work was carried out in (Vasilopoulos and Koditschek, 2018), where the workspace contains unknown convex obstacles and a class of geometrically known obstacles whose numbers and locations are unknown. Obstacles in the latter class have the shape of a non-convex star and are called "familiar obstacles". Once the familiar obstacles are detected and identified, they are transformed by diffeomorphism into disks, unlike the unknown convex obstacles, which retain their shape as detected. A new challenge was brought to the workspace in (Vasilopoulos et al., 2020), where non-convex obstacles are replaced by polygonal obstacles with possible overlap. A hybrid feedback control is then designed to safely steer the robot to the target. As a further challenge in (Vasilopoulos et al., 2021), authors considered a moving target.

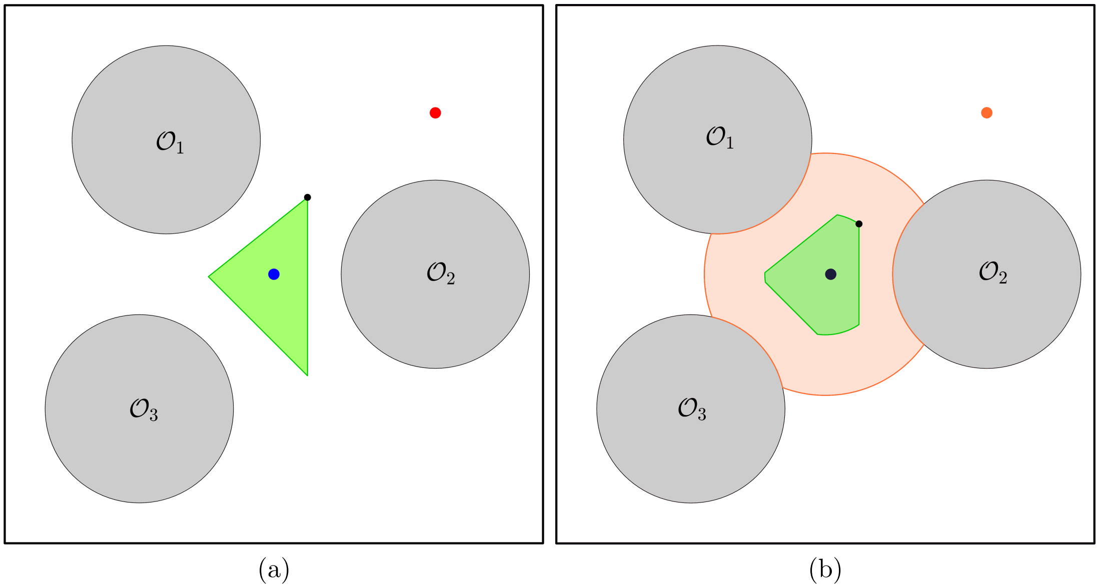


Figure 1.6: Local obstacle-free space (green) of a point robot (blue) obtained by the maximum margin separating hyperplanes of the robot body (blue) and obstacles (grey). The metric projection of the red target onto the local obstacle-free space is represented by a black dot. In the left figure (a), the robot assumes knowledge of adjacent obstacles to construct a local obstacle-free space. In the right figure (b), the robot has an onboard LiDAR that scans a limited region (orange) and then can build a local obstacle-free space from the detected obstacles portions.

Another sensor-based autonomous navigation approach, relying on Nagumo's theorem (Nagumo, 1942), was proposed in (Berkane, 2021; Lyes and Soulaimane, 2024) to navigate in environments with sufficiently curved obstacles similar to those considered in (S. Paternain and Ribeiro, 2018; Arslan and Koditschek, 2019). This approach mimics the Bug algorithms, more precisely the DistBug algorithm (Kamon and Rivlin,

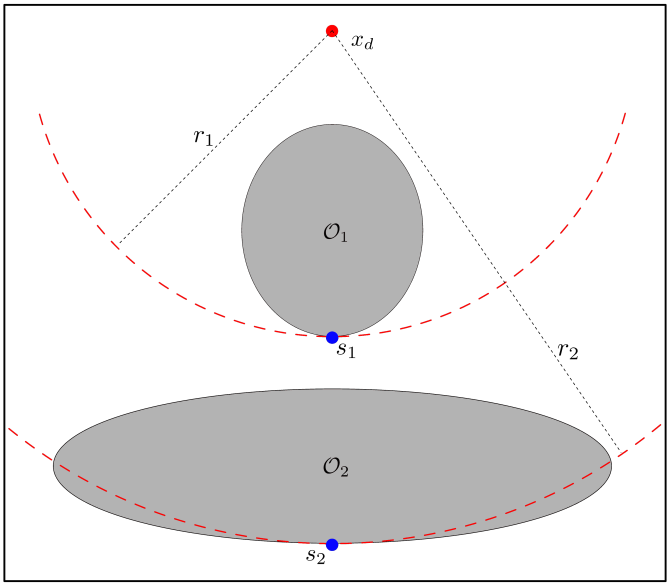


Figure 1.7: Illustration of the obstacle curvature condition in (Arslan and Koditschek, 2019, Assumption 2) with elliptical obstacles. The condition is that the obstacle must be entirely contained within the ball whose center is the target (red) and whose radius is equal to the distance between the target and the stationary point (blue). Obstacle $\mathcal{O}_1$ satisfies the condition, while obstacle $\mathcal{O}_2$ does not.

1997), by smoothly switching between a target stabilizing controller and an obstacle avoidance controller. As a result, safety and AGAS are guaranteed, which we illustrate through the example in Fig. 1.8. In (Prajna and Jadbabaie, 2004; Prajna, 2003), a new variant of Nagumo's theorem was introduced in the form of barrier certificates to guarantee the safety of nonlinear and hybrid systems. By analogy to Control Lyapunov Functions (CLFs), Control Barrier Functions (CBFs) have been introduced in (Wieland and Allgöwer, 2007) to design feedback controllers with safety guarantees for dynamical systems. CBFs and CLFs were unified in (Ames et al., 2014; Ames et al., 2017) through quadratic programs (QPs) for the design of navigation controllers guaranteeing the stabilization of a desired target location with safety guarantees. Although elegant and efficiently combining safety and stabilization, the QP-based framework, proposed in (Ames et al., 2014; Ames et al., 2017), suffers from the generation of stable undesired equilibria (Reis et al., 2021; Tan and Dimarogonas, 2024). A modified version of the QP-based control was proposed in (Reis et al., 2021; Tan and Dimarogonas, 2024) to eliminate certain types of undesired equilibria while ensuring local asymptotic stability of the target location.

None of the aforementioned works has achieved global asymptotic stability[5] (GAS) of the target using time-invariant continuous state feedback due to the topological obstruction pointed out in (Koditchek and Rimon, 1990). As an alternative, hybrid feedback

[5] An equilibrium point is globally asymptotically stable if it is stable and attractive from all initial conditions.

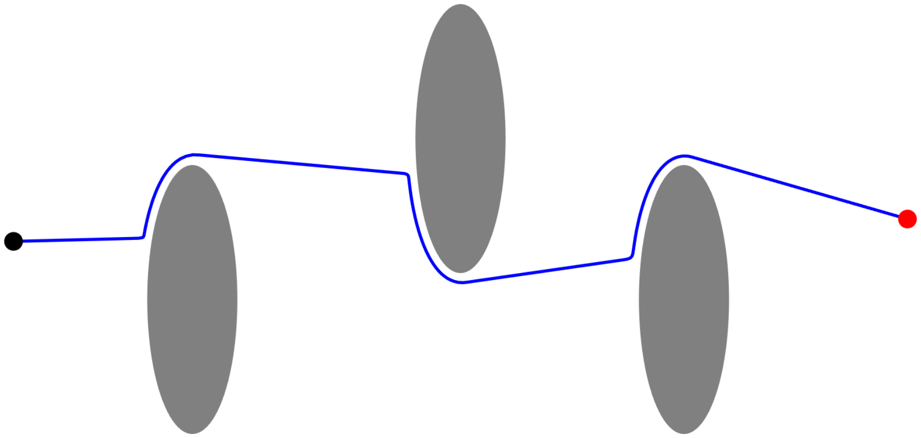


Figure 1.8: A DistBug-like trajectory in a workspace cluttered with elliptical obstacles.

controllers have been proposed in the literature to achieve GAS of the target location. A hybrid state feedback control providing robust global asymptotic stabilization of a target location while avoiding a single obstacle was proposed in (Sanfelice et al., 2006). This approach was extended in (Poveda et al., 2018) to autonomously navigate a group of vehicles to an unknown source, emitting a signal of measurable intensity in an environment containing a single obstacle. Hybrid state feedback, guaranteeing robust GAS of a target location while avoiding the neighborhood of a point obstacle, was proposed in (Braun et al., 2018) and extended in (Braun et al., 2021) to the case of multiple obstacles. The work in (Berkane et al., 2019) achieves GAS of a desired target location in $n$-dimensional Euclidean spaces with a single spherical obstacle. Similar to the Bug strategy, the proposed hybrid scheme switches between two navigation modes, namely, stabilization and avoidance modes, generating a discontinuous control input. An extension was proposed in (Berkane et al., 2022) for $n$-dimensional Euclidean spaces cluttered with sufficiently disjoint ellipsoidal obstacles. The strategy of this hybrid approach is illustrated in Fig. 1.9 by a trajectory that safely connects an initial position (black dot) to a final destination (red dot) while avoiding three elliptical obstacles, with the black portion of the trajectory representing the stabilization mode and the orange portion of the trajectory representing the avoidance mode. Recently, a similar hybrid strategy was proposed in (Sawant et al., 2023) to safely navigate in *a priori* unknown two-dimensional environments with arbitrary convex obstacles using onboard range scanners. The proposed hybrid feedback control guarantees GAS of the target location and generates continuous control input. This approach was further extended in (Sawant et al., 2024a) to navigate in two-dimensional environments with non-convex obstacles. Although the above-mentioned hybrid feedback-based approaches provide GAS results, the generated trajectories are not optimal in terms of length.

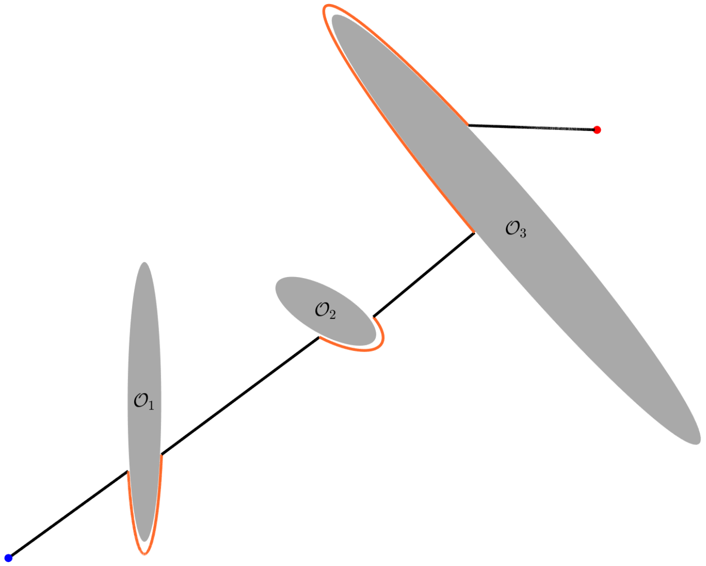


Figure 1.9: Safe navigation under a hybrid Bug-like strategy in a 2D workspace cluttered with elliptical obstacles. The black portions of the trajectory represent the stabilization mode, the orange portions of the trajectory represent the avoidance mode, the black dot represents the initial position, and the red dot represents the target.

## 1.3 Thesis contributions

In this dissertation, we aim to bridge the gap between the plan-and-track approaches, which generally prioritize path-length optimality, and feedback-based approaches, which are favored for their practicality and robustness. By integrating the principles of path-length optimization into feedback-based strategies, this work seeks to improve the efficiency of navigation systems while retaining their practicality. In particular, we address the problem of autonomous navigation in $n$-dimensional workspaces cluttered with convex obstacles using continuous and hybrid feedback controllers. Below, we summarize the main contributions of the present thesis.

Chapter 3 presents a continuous feedback control strategy for autonomous navigation in $n$-dimensional sphere worlds, and two-dimensional environments with arbitrary convex and sufficiently curved obstacles. A sensor-based implementation of our control approach in *a priori* unknown 2D environments with arbitrary convex and sufficiently curved obstacles is also proposed and experimentally validated. The main contributions of the proposed approach are summarized as follows:

1. The proposed continuous feedback control generates *quasi-optimal* trajectories in terms of length. The generated trajectories are often the shortest, as illustrated through extensive simulation results.

2. Except for the restrictions imposed by the standard separation conditions of Assumptions 3.1 and 3.2, the environment can be highly dense, and the destination can be located arbitrarily close to the boundaries of the obstacles.

3. The reactive (sensor-based) version of our approach applies to *a priori* unknown 2D environments with arbitrary convex and sufficiently curved obstacles, and ensures almost global asymptotic stabilization of the target location.

4. A successful implementation of the reactive autonomous navigation approach on a Turtlebot 4 platform, exclusively relying on onboard sensors.

The results presented in this chapter have been published in (Cheniouni et al., 2023b; Cheniouni et al., 2024c).

Chapter 4 proposes a hybrid feedback control strategy for safe autonomous navigation in $n$-dimensional Euclidean spaces with spherical obstacles. The proposed strategy relies on two different modes of operation: the *motion-to-destination* mode, where the robot moves directly toward the target when it has a clear line-of-sight, and the locally-optimal *obstacle-avoidance* mode when the robot does not have a clear line-of-sight to the target location. The main contributions of the proposed approach are summarized as follows:

1. The proposed hybrid feedback control strategy ensures safe navigation in $n$-dimensional Euclidean spaces with spherical obstacles while providing GAS guarantees for the target location.

2. The proposed hybrid feedback controller produces continuous vector fields.

3. The proposed approach enables the robot to optimally avoid obstacles (*i.e.,* via the shortest path) when they block the path to the target.

4. Navigation under the proposed control strategy applies to *a priori* unknown 2D and 3D environments, relying solely on range scanners. A successful implementation of the proposed approach on a Turtlebot 4 platform, exclusively relying on onboard sensors, further validates its practicability.

The results presented in this chapter are reported in (Cheniouni et al., 2024a) and (Cheniouni et al., 2024b).

## List of publications

The materials presented in this dissertation are based on the following publications:

**Journal Articles**:

- I. Cheniouni, S. Berkane, and A. Tayebi, “Safe and Quasi-Optimal Autonomous Navigation in Environments with Convex Obstacles,” *IEEE Transactions on Automatic Control*, accepted, 2024.

- I. Cheniouni, S. Berkane, and A. Tayebi, “Hybrid Feedback Control for Global Navigation with Locally Optimal Obstacle Avoidance in $n$-Dimensional Spaces,” *IEEE Transactions on Robotics*, submitted, 2024. `https://arxiv.org/abs/2412.20320`

**Peer-Reviewed Conference Proceedings**:

- I. Cheniouni, S. Berkane, and A. Tayebi, "Hybrid Feedback Control for Global and Optimal Safe Navigation," In *proc. of the American Control Conference (ACC)*, Toronto, Canada, pp. 1131-1136, 2024.
- I. Cheniouni, A. Tayebi, and S. Berkane, "Safe and Quasi-Optimal Autonomous Navigation in Sphere Worlds," In *proc. of the American Control Conference (ACC)*, San Diego, CA, USA, pp. 2678-2683, 2023.

## 1.4 Thesis outline

This thesis is organized as follows:

**Chapter 2** presents the notations, background and preliminaries used throughout the thesis. Section 2.1 provides the general notations used in this thesis. Section 2.2 presents the projection maps used throughout the thesis. Section 2.3 provides the mathematical definitions of several geometrical subsets of $\mathbb{R}^n$ used throughout this work. Finally, Section 2.4 presents the hybrid systems framework used in this dissertation.

**Chapter 3** tackles the autonomous navigation problem in environments with convex obstacles considering continuous feedback-based controllers. Section 3.4 presents the design of the proposed feedback controller for autonomous navigation in $n$-dimensional sphere worlds. Safety and stability properties are studied in Section 3.5. The feedback controller proposed in Section 3.4 is extended in Section 3.6 to *a priori* unknown planar environments with convex obstacles relying on onboard range scanners. Section 3.7 presents numerical simulation results. Section 3.8 reports the experimental results.

**Chapter 4** is dedicated to the design and stability analysis of a hybrid feedback controller solving the autonomous navigation problem with local optimal obstacle avoidance maneuvers in $n$-dimensional Euclidean spaces cluttered with spherical obstacles. Section 4.3 is devoted to the design of the proposed hybrid controller and the analysis of the properties of the resulting closed-loop system. The analyzed properties include forward invariance, stability, and optimality. Section 4.4 presents a sensor-based implementation of the proposed hybrid feedback control presented in Section 4.3 for autonomous navigation in two- and three-dimensional environments using range scanners. Simulation results follow in Section 4.5. Experimental results are presented in Section 4.6.

**Chapter 5** summarizes the findings of this thesis and presents some possible future directions.

**Appendices A and B contain the detailed proofs of the lemmas and theorems stated throughout this thesis.**

# Chapter 2

# Background and Preliminaries

## 2.1 General notations

Throughout the thesis, $\mathbb{N}$, $\mathbb{R}$, $\mathbb{R}_{\geq 0}$ and $\mathbb{R}_{>0}$ denote the set of natural numbers, real numbers, non-negative real numbers and positive real numbers, respectively. The Euclidean space and the unit $n$-sphere are denoted by $\mathbb{R}^n$ and $\mathbb{S}^n$, respectively. The Euclidean norm of $x \in \mathbb{R}^n$ is defined as $\|x\| := \sqrt{x^\top x}$ and the angle between two non-zero vectors $x, y \in \mathbb{R}^n$ is given by $\angle(x, y) := \cos^{-1}(x^\top y/\|x\|\|y\|)$. The identity matrix is denoted by $I_n \in \mathbb{R}^{n\times n}$. The interior, the boundary, and the closure of a set $\mathcal{A} \subset \mathbb{R}^n$ are denoted by $\mathring{\mathcal{A}}$, $\partial\mathcal{A}$, and $\overline{\mathcal{A}}$, respectively. The relative complement of a set $\mathcal{B} \subset \mathbb{R}^n$ with respect to a set $\mathcal{A} \subseteq \mathbb{R}^n$ is denoted by $\mathcal{A} \setminus \mathcal{B}$. The distance of a point $x \in \mathbb{R}^n$ to a set $\mathcal{A} \subset \mathbb{R}^n$ is defined as $d(x, \mathcal{A}) := \inf\limits_{q\in\mathcal{A}} \|q - x\|$.

## 2.2 Projection maps

Projection maps are essential tools in linear algebra for transforming vectors relative to specific subspaces, and are useful to develop our proposed navigation strategies in this thesis. In particular, we recall here three types–elementary reflector, elementary parallel projection, and elementary orthogonal projection–each with distinct properties relevant to our proposed approaches.

- **The elementary reflector**
  Let $v \in \mathbb{S}^{n-1}$ be a unit vector, we define the elementary projection map as (Meyer, 2000):

  $$\pi_r(v) := I_n - 2vv^\top. \tag{2.1}$$

  Then, for any vector $x \in \mathbb{R}^n$, the vector $\pi_r(v)x$ represents the reflection of $x$ about the hyperplane orthogonal to $v$.

- **The elementary parallel projection**
  Given a unit vector $v \in \mathbb{S}^{n-1}$, then the elementary parallel projection map is defined

as (Meyer, 2000):

$$\pi^{\|}(v) := vv^\top. \tag{2.2}$$

Consequently, for any vector $x \in \mathbb{R}^n$, the vector $\pi^{\|}(v)x$ represents the projection of $x$ onto the line generated by $v$.

- **The elementary orthogonal projection**
  Let $v \in \mathbb{S}^{n-1}$ be a unit vector, then the elementary orthogonal projection map is defined as (Meyer, 2000):

$$\pi^{\perp}(v) := I_n - vv^\top. \tag{2.3}$$

  Therefore, for any vector $x \in \mathbb{R}^n$, the vector $\pi^{\perp}(v)x$ represents the projection of $x$ onto the hyperplane orthogonal to $v$.

Each of the above projection maps has unique properties that make them useful in different contexts. Given a unit vector $v \in \mathbb{S}^{n-1}$, the following relationships hold:

$$\pi_r(v)v = -v, \qquad \pi_r(v)\pi_r(v) = I_n, \tag{2.4}$$
$$\pi^{\|}(v)v = v, \qquad \pi^{\|}(v)\pi^{\|}(v) = \pi^{\|}(v), \tag{2.5}$$
$$\pi^{\perp}(v)v = 0, \qquad \pi^{\perp}(v)\pi^{\perp}(v) = \pi^{\perp}(v). \tag{2.6}$$

These properties highlight the idempotent nature of projection operators, as well as their ability to maintain or cancel out specific vector components relative to $v$.

## 2.3 Subsets of the Euclidean space

In this section, we define sets that will help model the workspace and rigorously formulate our navigation problem. These sets are also necessary for the design of the navigation strategies proposed in this thesis.

- **Ball**
  Given $x \in \mathbb{R}^n$ and $r \in \mathbb{R}_{>0}$, a ball centered at $x$ with radius $r$ is given by

$$\mathcal{B}(x, r) := \{q \in \mathbb{R}^n |\ \|q - x\| \leq r\}. \tag{2.7}$$

- **Line**
  Let $x \in \mathbb{R}^n$ and $y \in \mathbb{R}^n$ where $x \neq y$, then a line passing through the points $x$ and $y$ is defined as

$$\mathcal{L}(x, y) := \{q \in \mathbb{R}^n | q = x + \lambda(y - x),\ \lambda \in \mathbb{R}\}. \tag{2.8}$$

- **Half line**
  Let $x \in \mathbb{R}^n$ and $v \in \mathbb{R}^n \setminus \{0\}$, then the half line starting from $x$ with direction $v$ is defined as

$$\mathcal{L}_h(x, v) := \{q \in \mathbb{R}^n | q = x + \lambda v,\ \lambda \geq 0\}. \tag{2.9}$$

- **Line segment** Let $x \in \mathbb{R}^n$ and $y \in \mathbb{R}^n$ where $x \neq y$, then a line segment joining the points $x$ and $y$ is given by

$$\mathcal{L}_s(x, y) := \{q \in \mathbb{R}^n | q = x + \lambda(y - x),\ \lambda \in [0, 1]\}\,. \tag{2.10}$$

- **Hyperplane**
  Given $x \in \mathbb{R}^n$ and $v \in \mathbb{R}^n \setminus \{0\}$, the hyperplane passing through $x$ and orthogonal to $v$ is defined as

$$\mathcal{P}_{=}(x, v) = \left\{q \in \mathbb{R}^n | v^\top (q - x) = 0\right\}. \tag{2.11}$$

  $\mathcal{P}_{=}(x, v)$. The closed negative half-space (resp. open negative half-space) is denoted by $\mathcal{P}_{\leq}(x, v)$ (resp. $\mathcal{P}_{<}(x, v)$) and the closed positive half-space (resp. open positive half-space) is denoted by $\mathcal{P}_{\geq}(x, v)$ (resp. $\mathcal{P}_{>}(x, v)$).

- **Cone**
  Given $x \in \mathbb{R}^n$, $v \in \mathbb{R}^n$, and $\varphi \in (0, \frac{\pi}{2}]$, a conic subset of $\mathcal{A} \subseteq \mathbb{R}^n$, with vertex $x$, axis $v$, and aperture $2\varphi$ is defined as

$$\mathcal{C}^{\Delta}_{\mathcal{A}}(x, v, \varphi) := \left\{q \in \mathcal{A} | \|v\| \|q - x\| \cos(\varphi) \Delta v^\top (q - x)\right\}, \tag{2.12}$$

  where $\Delta \in \{\leq, <, =, >, \geq\}$, with " $=$ ", representing the surface of the cone, " $\leq$ " (resp. " $<$ ") representing the interior of the cone including its boundary (resp. excluding its boundary), and " $\geq$ " (resp. " $>$ ") representing the exterior of the cone including its boundary (resp. excluding its boundary). The set of vectors parallel to the cone $\mathcal{C}^{=}_{\mathbb{R}^n}(x, v, \varphi)$ is defined as follows:

$$\mathcal{V}(v, \varphi) := \left\{w \in \mathbb{R}^n |\ \ w^\top v = \|w\| \|v\| \cos(\varphi)\right\}. \tag{2.13}$$

  In what follows, we state a property of cones sharing the same vertex (Berkane et al., 2019, Lemma 1).

  **Lemma 2.1** *Let $c, v_{-1}, v_1 \in \mathbb{R}^n$ such that $\angle(v_{-1}, v_1) = \psi$ where $\psi \in (0, \pi]$. Let $\varphi_{-1}, \varphi_1 \in [0, \pi]$ such that $\varphi_{-1} + \varphi_1 < \psi < \pi - (\varphi_{-1} + \varphi_1)$. Then*

$$\mathcal{C}^{\leq}_{\mathbb{R}^n}(c, v_{-1}, \varphi_{-1}) \cap \mathcal{C}^{\leq}_{\mathbb{R}^n}(c, v_1, \varphi_1) = \{c\}. \tag{2.14}$$

  Lemma 2.1 limits the angular aperture of two cones with the same vertex such that they do not intersect (their interior, including the boundaries) except at their common vertex.

- **Bouligand's tangent cone**
  Given a closed set[1] $\mathcal{K}$, the tangent cone to $\mathcal{K}$ at $x$ is defined as (Georges, 1932)

$$\mathrm{T}_{\mathcal{K}}(x) := \{z : \lim_{\tau \to 0} \inf \tau^{-1} d(x + \tau z, \mathcal{K}) = 0\}. \tag{2.15}$$

[1] A closed set is a set that contains its limit points.

> Bouligand's Tangent cone represents all possible directions from $x$ that keep us close to the set $\mathcal{K}$ when an infinitesimally small step is taken. For example, if $x$ is in the interior of the set $\mathcal{K}$, Boulogand's tangent cone will be the whole space (the Euclidean space in our case). Another example is when $x$ is on the boundary of $\mathcal{K}$ and $\mathcal{K}$ has a smooth boundary, Bouligand's tangent cone is then the tangent space of $\mathcal{K}$ at $x$. Bouligand's tangent cone has expanded the notion of tangency from tangency to smooth boundaries to tangency to nonsmooth and irregular boundaries, which helped show sets' invariance of broader geometrical classes.

Although some of these sets belong to basic geometrical classes, they play an essential role in a clear and rigorous representation of the robot's environment and simplify the design and proof of the results in this dissertation.

## 2.4 Hybrid dynamical systems

Hybrid systems are dynamic systems that combine continuous evolution and discrete jumps of the state, allowing a better representation of physical systems, such as mechanical systems with impacts. The hybrid model used in this thesis is the framework of hybrid systems developed in (Goebel et al., 2009; Goebel et al., 2012) that describes the continuous evolution with differential inclusions and discrete jumps with difference inclusions. Consider a manifold $\mathcal{X}$ embedded in $\mathbb{R}^n$ and its tangent space $T\mathcal{X}$. A general model for hybrid systems takes the form:

$$\begin{cases} \dot{x} \in \mathrm{F}(x) & x \in \mathcal{F} \\ x^+ \in \mathrm{J}(x) & x \in \mathcal{J} \end{cases} \tag{2.16}$$

where

- $\mathcal{F} \subset \mathcal{X}$ is the flow set where the continuous evolution of the state is allowed,
- $\mathcal{J} \subset \mathcal{X}$ is the jump set where the discrete state evolution is allowed,
- The flow map $\mathrm{F} : \mathcal{X} \rightrightarrows T\mathcal{X}$ governs the continuous flow on $\mathcal{F}$,
- The jump map $\mathrm{J} : \mathcal{X} \rightrightarrows \mathcal{X}$ governs the discrete jumps on $\mathcal{J}$.

Note that $\rightrightarrows$ denotes a set-valued mapping, and $x^+$ denotes the value of $x$ after an instantaneous jump. The hybrid system (2.16) is defined by its data and denoted as $\mathcal{H} = (\mathcal{F}, \mathrm{F}, \mathcal{J}, \mathrm{J})$. It is worth noting that the hybrid system framework (2.16) is generic in the sense that it can represent purely continuous dynamics when the flow set equals the state space $\mathcal{X}$ and the jump set is empty, or vice versa to represent purely discrete dynamics.

Solutions of a hybrid system are parameterized by the time $t \in \mathbb{R}_{\geq 0}$ spent in the flow set and the number of jumps $j \in \mathbb{N}$ of the state. The structure that represents this parameterization, known as a *hybrid time domain*, is denoted by $E \subset \mathbb{R}_{\geq 0} \times \mathbb{N}$ and defined as follows

$$E = \bigcup_{j=0}^{J-1} ([t_j, t_{j+1}], j)$$

for some finite sequence of time $0 = t_0 \leq t_1 \leq t_2 \leq \ldots \leq t_J$ with the last interval, if existent, being possibly of the form $[t_{J-1}, T) \times \{J\}$ and $T$ finite or $T = +\infty$. For any $(t, j)$ and $(t', j')$ belonging to a hybrid time domain, one has $(t, j) \leq (t', j')$ if and only if $t \leq t'$ and $j \leq j'$.

**Definition 2.1** *(Goebel et al., 2012) A hybrid arc is a function $x : \mathrm{dom}\ x \to \mathcal{X}$, where* dom $x$ *is a hybrid time domain, for each fixed $j$, $t \to x(t, j)$ is a locally absolutely continuous function on the the interval*

$$I^j = \{t : \ (t, j) \in E\}$$

The hybrid arc $x$ is a solution to the hybrid system $\mathcal{H}$, given in (2.16), if $x(0, 0) \in \mathcal{F} \cup \mathcal{J}$ and the following conditions are satisfied (Goebel et al., 2012):

- Flow condition: for each $j \in \mathbb{N}$ such that $I^j$ has nonempty interior with

$$\dot{x}(t, j) \in \mathrm{F}(x(t, j)), \text{ for almost all } t \in I^j, \tag{2.17}$$

$$x(t, j) \in \mathcal{F}, \text{ for all } t \in [\min I^j, \sup I^j). \tag{2.18}$$

- Jump condition: for each $(t, j) \in \mathrm{dom}\ x$ such that $(t, j + 1) \in \mathrm{dom}\ x$ with

$$x(t, j + 1) \in \mathrm{J}(x(t, j)), \tag{2.19}$$

$$x(t, j) \in \mathcal{J}. \tag{2.20}$$

The following definitions characterize the nature of a hybrid system solution:

**Definition 2.2** *(Goebel et al., 2012) A solution $x$ to $\mathcal{H}$ is said to be* maximal *if there is no other solution $x^*$ to $\mathcal{H}$ such that* dom $x$ *is a proper subset of* dom $x^*$ *and $x(t, j) = x^*(t, j)$ for all $(t, j) \in$* dom $x$.

**Definition 2.3** *(Goebel et al., 2012) A solution $x$ to $\mathcal{H}$ is said to be* complete *if its domain, namely* dom $x$, *is unbounded.*

**Definition 2.4** *(Goebel et al., 2012) A solution $x$ to $\mathcal{H}$ is said to be* precompact *if it is complete and bounded.*

**Remark 2.1** *It follows from the above definitions that every complete solution of $\mathcal{H}$ is maximal, but the opposite is not necessarily true.*

Three basic conditions/assumptions were introduced for the framework (2.16) to guarantee the existence of solutions, the robustness of stability in the presence of small perturbations, and other important properties (Goebel et al., 2012, Assumption 6.5). These hybrid basic conditions are satisfied by the hybrid system $\mathcal{H}$ if

- the flow set $\mathcal{F}$ and the jump set $\mathcal{J}$ are closed sets in $\mathcal{X}$,
- the flow map $\mathrm{F} : \mathcal{X} \rightrightarrows T\mathcal{X}$ is outer semicontinuous [2] and locally bounded[3] relative to $\mathcal{F}$, and the set $\mathrm{F}(x)$ is nonempty and convex for every $x \in \mathcal{F}$.
- The jump map $\mathrm{J} : \mathcal{X} \rightrightarrows \mathcal{X}$ is outer semicontinuous and locally bounded relative to $\mathcal{J}$, and J nonempty for every $x \in \mathcal{J}$.

[2] This means that for every $x_0 \in \mathcal{X}$, one has $\limsup_{x \to x_0} \mathrm{F}(x) \subseteq \mathrm{F}(x_0)$.

[3] This means that for each $x_0 \in \mathcal{F}$, there exists a neighborhood $N_{x_0}$ of $x_0$ such that all sets in the range of F (*i.e.*, $\mathrm{F}(x)$ for $x \in N_{x_0}$) are bounded sets.

# Chapter 3

# Safe and Quasi-Optimal Autonomous Navigation in Environments with Convex Obstacles

## 3.1 Introduction

In this chapter, we address the problem of autonomous robot navigation in environments cluttered with convex obstacles. Most existing feedback-based navigation approaches focus mainly on safety and stability guarantees without paying much attention to the lengths of the generated trajectories. The NF-based approach is an example of a feedback-based autonomous navigation strategy that guarantees AGAS of a target location in sphere worlds (Koditchek and Rimon, 1990), star worlds (Rimon and Koditchek, 1992) and sufficiently curved convex worlds (Filippidis and Kyriakopoulos, 2012; S. Paternain and Ribeiro, 2018). Other extensions of the NF-based approach dealing with multi-robot navigation (Loizou and Kyriakopoulos, 2006; Dimarogonas et al., 2006) or sensor-based navigation (Loizou and Rimon, 2021) have also been proposed in the literature. Unfortunately, path-length optimality was not among the various extensions except the Optimal Navigation Function based approach proposed for navigation in 2D polygonal environments without internal obstacles (LaValle, 2006). This approach generates discontinuous vector fields and does not extend to 3D environments. Another example of a feedback-based approach is the purely reactive sensor-based controller proposed in (Arslan and Koditschek, 2016) for navigation in sphere worlds from almost all initial positions in the obstacle-free space. This approach was extended to deal with sufficiently curved convex obstacles (Arslan and Koditschek, 2019) and non-convex obstacles (Vasilopoulos and Koditschek, 2018; Vasilopoulos et al., 2020), but again, path-length optimality is not addressed by this approach. The same remark applies to the hybrid feedback navigation approaches that guarantee GAS of the target in $n$-dimensional Euclidean spaces with ellipsoidal obstacles (Berkane et al., 2022) or in two-dimensional environments with convex (Sawant et al., 2023) or non-convex obstacles (Sawant et al., 2024b). The generated trajectories of these hybrid feedback approaches are not length optimal and are similar to those generated by the Bug algorithms. These approaches operate under the move-to-target mode (*i.e.,* move straight to the target) until the robot hits the boundary of an obstacle (or is in close vicinity of the obstacle). Then, the control switches to the obstacle-avoidance mode, where the robot moves along the obstacle boundary (left or right direction) until it has a clear line of sight to the target location. On the other hand, some plan-and-track approaches solve the autonomous navigation problem with the generation of the shortest path as the main objective, such as the TVG-based approach (Rohnert, 1986). The TVG is a graph containing the shortest path, which can be determined by running search algorithms such as Dijkstra's algorithm on the graph. Initially designed for two-dimensional environments cluttered with polygonal obstacles, it was subsequently extended to curved obstacles (Laumond, 1987; Liu and Arimoto, 1992). Unfortunately, the TVG is limited to two-dimensional environments (LaValle, 2006; Latombe, 2012) and requires an additional step to track the determined shortest path from the search phase by a low-level controller. Other plan-and-track approaches, such as RRT* and PRM* (Karaman and Frazzoli, 2011), provide approximate shortest paths.

In this chapter, we propose a continuous feedback control strategy for safe autonomous navigation, in $n$-dimensional sphere worlds and two-dimensional environments with arbitrary convex and sufficiently curved obstacles, that generates *quasi-optimal* trajectories, as per Definition 3.1 that will be provided in subsection 3.4.3. The proposed approach relies on an iterative optimization process that projects the nominal feedback controller on the obstacles' enclosing cones. We also propose a sensor-based implementation of our approach in *a priori* unknown 2D environments with arbitrary convex and sufficiently curved obstacles, which has been experimentally validated on a differential-drive robot. The main contributions of the proposed approach are summarized as follows:

- **Quasi-optimal trajectory generation:** The generated trajectories are often the shortest, as illustrated through extensive simulation results.

- **Navigation in arbitrarily dense environments:** The environments considered can be arbitrarily dense, provided the obstacles are strictly disjointed and entirely contained within the workspace. The destination can be located arbitrarily close to the obstacle boundaries.

- **Navigation in *a priori* unknown environments:** Our approach's reactive (sensor-based) version handles sufficiently curved convex obstacles in *a priori* unknown 2D environments with AGAS guarantees. This approach is experimentally validated through implementation on a differential-drive robot.

The results presented in this chapter have been published in one conference paper (Chenioumi et al., 2023b) and one journal paper (Cheniouni et al., 2024c).

## 3.2 Problem formulation

Consider a robot at position $x \in \mathbb{R}^n$ moving inside a spherical workspace $\mathcal{W} \subset \mathbb{R}^n$ centered at the origin 0 and punctured by $m \in \mathbb{N}$ balls $\mathcal{O}_i$ such that:

$$\mathcal{W} := \mathcal{B}(0, r_0), \tag{3.1}$$

$$\mathcal{O}_i := \mathcal{B}(c_i, r_i) \subset \mathcal{W}, \quad i \in \mathbb{I} := \{1, \ldots, m\}, \tag{3.2}$$

where $r_0 > r_i > 0$ for all $i \in \mathbb{I}$. The free space is, therefore, given by the closed set

$$\mathcal{F} := \mathcal{W} \setminus \bigcup_{i=1}^{m} \mathring{\mathcal{O}}_i. \tag{3.3}$$

For $\mathcal{F}$ to be a valid sphere world, as defined in (Koditchek and Rimon, 1990), the obstacles $\mathcal{O}_i$ must satisfy the following assumptions:

**Assumption 3.1** *The obstacles are completely contained within the workspace and separated from its boundary, i.e.,*

$$\min_{a \in \mathcal{O}_i, b \in \partial\mathcal{W}} \|a - b\| > 0, \, \forall i \in \mathbb{I}. \tag{3.4}$$

**Assumption 3.2** *The obstacles are disjoint, i.e.,*

$$\min_{a \in \mathcal{O}_i, b \in \mathcal{O}_j} \|a - b\| > 0, \, \forall i, j \in \mathbb{I}, \, i \neq j. \tag{3.5}$$

Consequently, the boundary of the free space $\mathcal{F}$ is given by

$$\partial\mathcal{F} := \partial\mathcal{W} \bigcup \Big( \bigcup_{i=1}^{m} \partial\mathcal{O}_i \Big). \tag{3.6}$$

Consider the following first-order dynamics

$$\dot{x} = u, \tag{3.7}$$

where $u$ is the control input. The objective is to determine a Lipschitz continuous state-feedback controller $u(x)$ that safely steers the vehicle from almost all initial positions $x(0) \in \mathcal{F}$ to any given desired destination $x_d \in \mathring{\mathcal{F}}$. In particular, the closed-loop system

$$\dot{x} = u(x), \quad x(0) \in \mathcal{F} \tag{3.8}$$

must ensure forward invariance of the set $\mathcal{F}$, almost global asymptotic stability of the equilibrium $x = x_d$, and generates *quasi-optimal* trajectories that will be rigorously defined later in subsection 3.4.3.

## 3.3 Sets definition

In this section, we define the subsets of the free space that are needed for our proposed control design in Section 3.4. These are depicted in Fig. 3.1 and given as follows:

- The *hat* of a cone inside the workspace $\mathcal{W}$, enclosing an obstacle $\mathcal{O}_i$, of vertex $x \in \mathbb{R}^n$ and aperture $\theta_i$ is defined as follows:

$$\mathcal{H}(x, c_i) := \big\{q \in \mathcal{C}^{\leq}_{\mathcal{W}}(x, c_i - x, \theta_i(x)) | (c_i - q)^\top (x - q) \leq 0\big\}, \tag{3.9}$$

  where the angle $\theta_i(x) = \arcsin\left(r_i / \|c_i - x\|\right) \in (0, \frac{\pi}{2}]$.

- The *shadow region* of obstacle $\mathcal{O}_i$, which is the area hidden by obstacle $\mathcal{O}_i$, from which there is no line of sight to the destination, is defined as follows:

$$\mathcal{D}(x_d, c_i) := \big\{q \in \mathcal{C}^{\leq}_{\mathcal{F}}(x_d, c_i - x_d, \varphi_i) | (c_i - q)^\top (x_d - q) \geq 0\big\}, \tag{3.10}$$

  where the angle $\varphi_i = \arcsin\left(r_i / \|c_i - x_d\|\right) \in (0, \frac{\pi}{2}]$.

- The *exit set* of obstacle $\mathcal{O}_i$ separates the set $\mathcal{D}(x_d, c_i)$ and its complement with respect to $\mathcal{F}$ and is defined as follows:

$$\mathcal{S}(x_d, c_i) := \big\{q \in \mathcal{C}^{=}_{\mathcal{F}}(x_d, c_i - x_d, \varphi_i) | (c_i - q)^\top (x_d - q) \geq 0\big\}. \tag{3.11}$$

- The *blind set* is a subset of $\mathcal{F}$ where there is no line of sight to the destination, and is defined as follows:

$$\mathcal{BL} := \{q \in \mathcal{F} | \mathcal{L}_s(q, x_d) \cap \mathcal{O}_k \neq \varnothing,\ k \in \mathbb{I}\}, \tag{3.12}$$

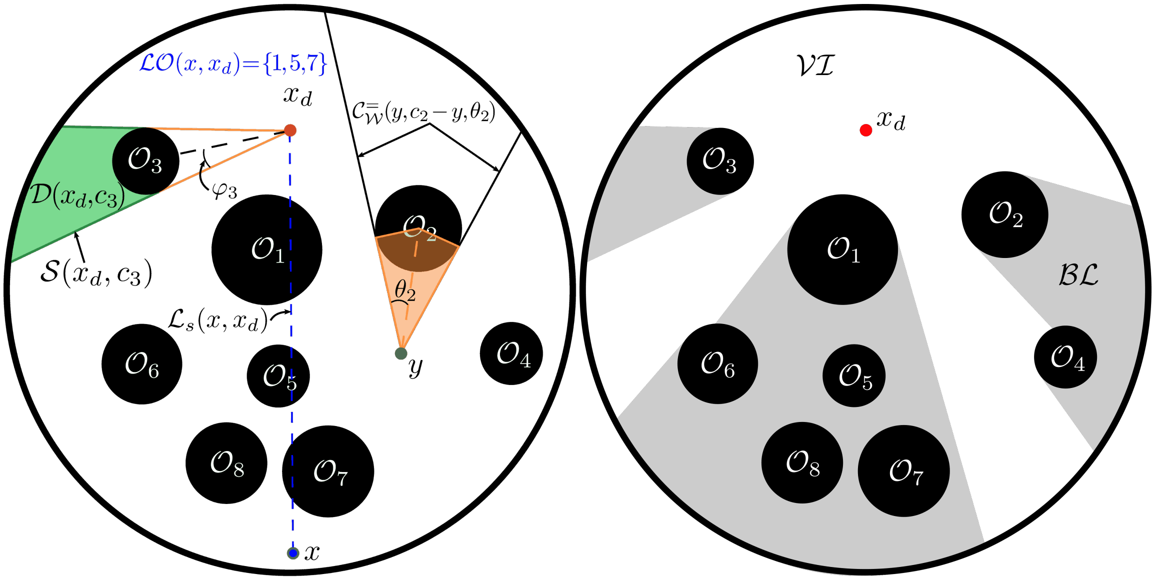
$$:= \bigcup_{i \in \mathbb{I}} \mathcal{D}(x_d, c_i). \tag{3.13}$$

- The *visible set* is the complement of the *blind set* with respect to the free space

$$\mathcal{VI} := \mathcal{BL}^c_{\mathcal{F}}. \tag{3.14}$$

- The set of *blocking obstacles* between two given positions $x$ and $y$ is the set of obstacles crossed by the line-segment $\mathcal{L}_s(x, y)$, and is defined as follows:

$$\mathcal{LO}(x, y) := \{k \in \mathbb{I} | \mathcal{O}_k \cap \mathcal{L}_s(x, y) \neq \varnothing\}. \tag{3.15}$$



(a) (b)

Figure 3.1: 2D representation of the sets in Section IV.

## 3.4 Control design

### 3.4.1 Single obstacle case

We design a preliminary control law for the single obstacle case, which will be used as a baseline in the multiple obstacle case. Let us start by considering a single obstacle $\mathcal{O}_i$ and ignoring all others. In the case where the path is clear (*i.e.*, $x$ belongs to the visible set $\mathcal{VI}$), the vehicle follows a straight line to the destination under the nominal control law $u_d(x) = \gamma(x_d - x)$ where $\gamma \in \mathbb{R}_{>0}$. In the case where the path is not clear (*i.e.*, $x \in \mathcal{D}(x_d, c_i)$), we generate a control input (vehicle's velocity) that is in the direction of the cone $\mathcal{C}^{=}_{\mathcal{F}}(x, c_i - x, \theta_i)$ enclosing the obstacle. In particular, the direction of the control input should minimize the angle between the nominal control direction, given by $(x_d - x)$, and the set of all vectors parallel to the enclosing cone, that is

$$u(x) \in \mathcal{U}(x) := \arg \min_{v_i \in \mathcal{V}(c_i - x, \theta_i)} \angle(x_d - x, v_i),\ x \in \mathcal{D}(x_d, c_i). \tag{3.16}$$

Moreover, to ensure continuity of the control input, we impose further that the control is equal to $u_d(x)$ at the exit set $\mathcal{S}(x_d, c_i) \subset \mathcal{D}(x_d, c_i)$, *i.e.*,

$$\forall x \in \mathcal{S}(x_d, c_i),\ u(x) = u_d(x), \tag{3.17}$$

The following lemma provides the solution of the optimization problem (3.16)-(3.17) and shows its uniqueness.

**Lemma 3.1** *The solution of the optimization problem* (3.16)-(3.17) *is unique and is given by*

$$u(x) = \xi(u_d(x), x, i), \tag{3.18}$$

*where* $\xi : \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{N} \to \mathbb{R}^n$ *is given by*

$$\xi(u, x, i) := \frac{\sin(\beta_i(u,x))\sin^{-1}(\theta_i(x))}{\cos(\theta_i(x) - \beta_i(u,x))}\pi^{\parallel}(\bar{\xi}_i)u, \tag{3.19}$$

*with* $\bar{\xi}_i \in \mathcal{V}(c_i - x, \theta_i)$,

$$\begin{aligned}\bar{\xi}_i &:= \frac{\sin(\theta_i(x))u}{\sin(\beta_i(u,x))\|u\|} - \frac{\sin(\theta_i(x) - \beta_i(u,x))}{\sin(\beta_i(u,x))}\frac{(c_i - x)}{\|c_i - x\|},\\ \beta_i(u,x) &:= \angle(u, c_i - x) \leq \theta_i(x).\end{aligned}$$

**Proof** See Appendix A.1.

Lemma 3.1 shows that, when $x \in \mathcal{D}(x_d, c_i)$, the control $u(x)$ is a scaled parallel projection of the nominal controller $u_d(x)$ in the direction of $\bar{\xi}_i$ which represents a unit vector on the cone enclosing the obstacle.
The continuous control strategy in the case of a single obstacle is given by

$$u(x) = \begin{cases} u_d(x), & x \in \mathcal{VI}, \\ \xi(u_d(x), x, i), & x \in \mathcal{D}(x_d, c_i). \end{cases} \tag{3.20}$$

The trajectory of the closed-loop system (3.7)-(3.20) is length-optimal as shown in the following lemma.

**Lemma 3.2** *The path generated by the closed-loop system* (3.7)*-*(3.20) *is the shortest path to the destination* $x_d$ *from every initial condition* $x(0) \in \mathcal{F} \setminus \mathcal{L}_d(x_d, c_i)$ *where* $\mathcal{L}_d(x_d, c_i) := \mathcal{D}(x_d, c_i) \cap \mathcal{L}_h(c_i, c_i - x_d)$.

**Proof** See Appendix A.2.

### 3.4.2 Multiple obstacles case

The objective in this subsection is to extend the controller (3.20) to the multiple obstacle case. The robot moves under the nominal control $u_d(x)$ when the robot has a clear line of sight to the destination (*i.e.,* $x \in \mathcal{VI}$). When there is no clear line of sight to the destination (*i.e.,* $x \in \mathcal{BL}$), one proceeds with multiple projections as described hereafter. As per (3.12), at every position $x \in \mathcal{BL}$, the blocking obstacles between $x$ and $x_d$ are represented by the set $\mathcal{LO}(x, x_d) \neq \varnothing$. Among the set $\mathcal{LO}(x, x_d)$, we select the one closest to destination $\big($*i.e.,* $i = \arg\{\min d(x_d, \mathcal{O}_k),\ k \in \mathcal{LO}(x, x_d)\}\big)$, where $u_d(x)$ is projected onto the enclosing cone of the selected obstacle $\mathcal{O}_i$ using (3.19), as in the case of a single obstacle. The resulting control vector is denoted by $u_1(x)$. The next obstacle to be considered is selected from the set of blocking obstacles $\mathcal{LO}(x, \hat{c}_i(x))$, where $\hat{c}_i(x) := x + \pi^{\parallel}(u_1(x)/\|u_1(x)\|)(c_i - x)$ is the point at which the line directed by $u_1(x)$ is tangent to obstacle $\mathcal{O}_i$. One chooses the closest obstacle to $\mathcal{O}_i$ among the set $\mathcal{LO}(x, \hat{c}_i)$ in terms of the Euclidean distance. If $\mathcal{LO}(x, \hat{c}_i(x)) = \varnothing$, the path is free. Otherwise, $u_1$ will be considered as $u_d$ for the newly selected obstacle and the same approach is followed to obtain $u_2$. Obstacle $\mathcal{O}_i$ is called an **ancestor** to the selected obstacle and the selection and projection are repeated until the path is free (see Fig. 3.2). The obstacles selected during the successive projections at a position $x$, are grouped in an ordered list $\mathcal{I}(x) \subset \mathbb{I}$ from the first obstacle $\big(\mathcal{O}_i$, such that $i = \arg\{\min d(x_d, \mathcal{O}_k),\ k \in \mathcal{LO}(x, x_d)\}\big)$ to the last one (obstacle involved in the last projection). Let $h(x) = \mathbf{card}(\mathcal{I}(x))$ be the number of required projections at position $x$. Define the map $\iota_x : \{1, \ldots, h(x)\} \to \mathcal{I}(x)$ which associates to each projection $p \in \{1, \ldots, h(x)\}$ the corresponding obstacle $\iota_x(p) \in \mathcal{I}(x)$. The set of positions involving obstacle $k$ in the successive projections is called active region and defined as $\mathcal{AR}_k := \{q \in \mathcal{BL} | k \in \mathcal{I}(q)\}$. To sum up, the intermediary control at a step $p \in \{1, \ldots, h(x)\}$ and position $x \in \mathcal{AR}_{\iota_x(p)}$ is given by the recursive formula

$$u_p(x) = \xi(u_{p-1}(x), x, \iota_x(p)), \tag{3.21}$$

with $u_0(x) = u_d(x)$ and $\xi(\cdot, \cdot, \cdot)$ as defined in Lemma 3.1. The point at which the line directed by $u_p(x)$ is tangent to obstacle $\mathcal{O}_{\iota_x(p)}$ is given by $\hat{c}_{\iota_x(p)}(x) := x + \pi^{\|}(u_p(x)/\|u_p(x)\|)(c_{\iota_x(p)} - x)$. Finally, the proposed control law is obtained by performing $h(x)$ successive projections and is given by

$$u(x) = \begin{cases} u_d(x), & x \in \mathcal{VI}, \\ u_{h(x)}(x), & x \in \mathcal{BL}. \end{cases} \tag{3.22}$$

The implementation of the control strategy (3.22) is summarized in Algorithm 1.

**Remark 3.1** *It is worth pointing out that the successive projections, involved in the control design, start from the closest obstacle to the destination. This approach enables our controller to enjoy the following features: 1) generates quasi-optimal trajectories; 2) guarantees the continuity of the control input.*

### 3.4.3 Characterization of the generated trajectories

The proposed control strategy steers the robot from an initial location $x_0 \in \mathcal{F}$ to a final destination $x_d \in \mathring{\mathcal{F}}$ by tracking a position-dependant virtual destination. A virtual destination at a position $x \in \mathcal{BL}$ is given by $P(x) := P_{h(x)}(x)$, where $P_{h(x)}(x)$ is the last in a list of successive intermediary destinations $P_p(x) := x + u_p(x)$, with $p \in \{1, \ldots, h(x)\}$, $h(x) = \mathbf{card}(\mathcal{I}(x))$ and $P_0(x) = x_d$. The point $P_p(x)$ lies on the surface of the cone enclosing the obstacle of index $\iota_x(p) \in \mathcal{I}(x)$. The intermediary destinations are designed to guarantee a minimum deviation between $(P_{p-1}(x) - x)$ and $(P_p(x) - x)$ for all $p \in \{1, \ldots, h(x)\}$. This deviation represented by the angle $\angle((P_{p-1}(x) - x), (P_p(x) - x)) = \angle(u_{p-1}(x), u_p(x))$ is the smallest possible since $u_p(x) = \xi(u_{p-1}(x), x, \iota_x(p))$, where the operator $\xi(\cdot, \cdot, \cdot)$, defined in Lemma 3.1, minimizes the angle $\angle(u_{p-1}(x), u_p(x))$ such that $u_p(x) \in \mathcal{V}(c_{\iota_x(p)} - x, \theta_{\iota_x(p)}(x))$. Recall that the set $\mathcal{V}(c_{\iota_x(p)} - x, \theta_{\iota_x(p)}(x))$ is the set of vectors parallel to the cone $\mathcal{C}^{=}_{\mathbb{R}^n}(x, c_{\iota_x(p)} - x, \theta_{\iota_x(p)}(x))$ enclosing obstacle $\mathcal{O}_{\iota_x(p)}$. The virtual destination $P(x)$, at a position $x \in \mathcal{BL}$, is the final intermediary destination obtained through the following recursive minimization process:

$$\begin{aligned} &P_p(x) := \arg \min_{\substack{y \in \mathcal{C}^{=}_{\mathbb{R}^n}(x, c_{\iota_x(p)} - x, \theta_{\iota_x(p)}(x)) \setminus \{x\} \\ \|y - x\| = u_p(x)}} \angle(y - x, P_{p-1}(x) - x), \\ &P_0(x) = x_d, \quad p \in \{1, \ldots, h(x)\}. \end{aligned} \tag{3.23}$$

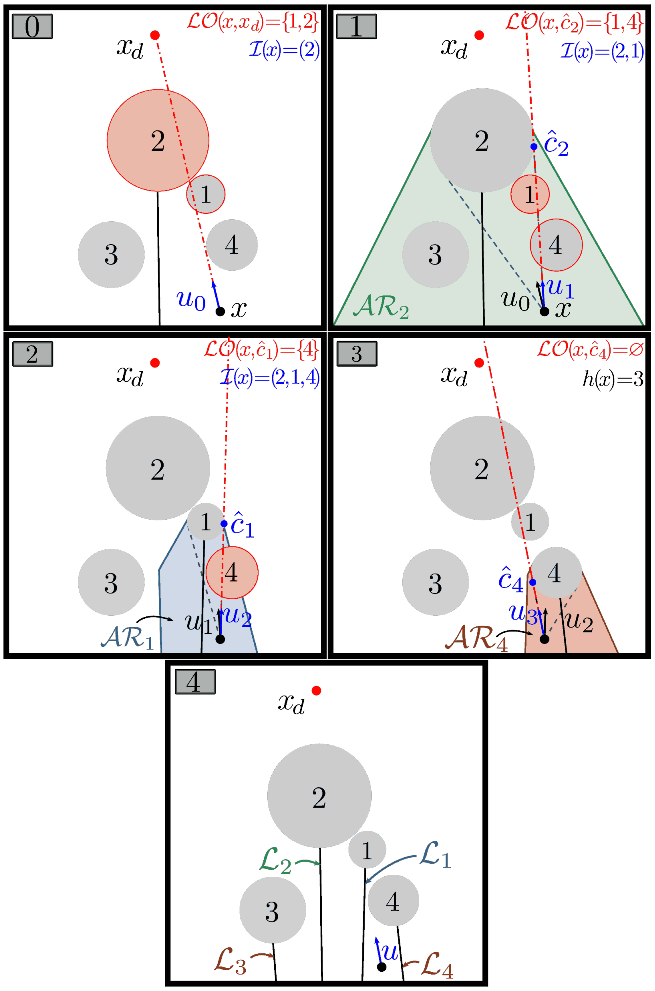


Figure 3.2: Successive projections of the control $u_d$ in a two-dimensional sphere world with four obstacles. At step zero, the nominal control $u_0 = u_d$ is projected onto the cone enclosing obstacle $\mathcal{O}_2$, where obstacle $\mathcal{O}_2$ is the closest blocking obstacle to the destination $x_d$ among the set $\mathcal{LO}(x, x_d)$ of blocking obstacles between the position $x$ and the destination $x_d$. The resultant intermediary control $u_1$ is considered for the projection at step 1 where $\hat{c}_2$, the point at which the line directed by $u_1$ is tangent to $\mathcal{O}_2$, plays the role of the destination. The same operation is repeated until step 3 where the set of blocking obstacles $\mathcal{LO}(x, \hat{c}_4)$ is empty. The velocity control will be then $u = u_3$. The sets $\mathcal{L}_i$ represent the unstable undesired equilibria associated with each obstacle.

The virtual destination coincides with the final destination (*i.e.*, $P(x) := x_d$) when $x \in \mathcal{VL}$.

Throughout this dissertation, the trajectories generated with our optimized succe‘ssive projections approach are referred to as *quasi-optimal* trajectories, and are defined as follows:

**Definition 3.1** *Given an initial position $x_0 \in \mathcal{F}$ and a final destination $x_d \in \mathring{\mathcal{F}}$, a continuously differentiable trajectory connecting $x_0$ and $x_d$ is said to be quasi-optimal if it has the shortest length when $x_0 \in \mathcal{VI}$ and when $x_0 \in \mathcal{BL}$, the tangent vector to the*

**Algorithm 1** Implementation of the control law (3.22)

```
Initialization: : $x_d$, $e_c$;
while true do
    Measure $x$;
    if $\|x - x_d\| \leq e_c$ then
        Break;
    else
        if $x \in \mathcal{BL}$ then
            $i \leftarrow \arg \min_{k \in \mathcal{LO}(x, x_d)} d(x_d, \mathcal{O}_k)$;
            while $i \neq \{\varnothing\}$ do
                Update $u$ using (3.21);
                if $\mathcal{LO}(x, \hat{c}_i(x)) = \varnothing$ then
                    $i \leftarrow \{\varnothing\}$;
                else
                    $i \leftarrow \arg \min_{k \in \mathcal{LO}(x, \hat{c}_i(x))} d(\hat{c}_i(x), \mathcal{O}_k)$;
                end if
            end while
        else
            $u \leftarrow u_d$;
        end if
        Execute $u$ in (3.8);
    end if
end while
```

*trajectory, at each $x$, points towards the virtual destination $P(x)$ obtained by the recursive minimization process* (3.23)*.*

A *quasi-optimal* trajectory, as per Definition 3.1, is a trajectory along which the vehicle's velocity, at a given location $x$, always points to a virtual destination (depending on $x$). The virtual destination, at position $x$ on the trajectory, is a result of a series of minimized deviations from the nominal direction (the direction from $x$ to $x_d$) with respect to the blocking obstacles, starting from the closest to the destination $x_d$. An example of a *quasi-optimal* trajectory is shown in Fig. 3.3 in blue color. Figure 3.3(a) and Fig. 3.3(b) illustrate the characteristics of a *quasi-optimal* trajectory where at each position $x$ on the trajectory, the tangent to the trajectory points toward the green virtual destination. This green virtual destination $P(x)$ at position $x$, for example, in Fig. 3.3(a), is obtained after three iterations of the recursive minimization process (3.23). Initially, the robot's velocity points to the destination $x_d$ (red dot). The first step is to deviate this velocity vector with a minimum angle from its initial direction to be tangent to the blocking obstacle $\mathcal{O}_4$. The obtained velocity vector will point to an intermediary destination $P_1(x)$ (orange point). The next step is to deviate the obtained velocity vector from the direction pointing to $P_1(x)$ with a minimum angle to be tangent to the blocking obstacle $\mathcal{O}_1$. The resultant velocity vector will point to the intermediary destination $P_2(x)$. In the last step, we deviate the newly obtained velocity vector with a minimum angle from the direction pointing to $P_2(x)$ to be tangent to the last blocking obstacle $\mathcal{O}_5$. The velocity vector then points to the virtual destination $P(x)$. A simulation video highlighting the characteristics of a *quasi-optimal* trajectory can be found online.[1] The generated *quasi-optimal* trajectory shown in Fig. 3.3(c) coincides with the shortest path (green curve). However, it is not always the case, as shown in Fig. 3.4, where one can observe that, for the initial position $x_0^2$, the *quasi-optimal* trajectory (blue) coincides with the shortest path (green), while for the initial position $x_0^1$, it does not. The following remark provides some additional interesting features of *quasi-optimal* trajectories in two-dimensional environments.

**Remark 3.2** *In two-dimensional environments, the quasi-optimal trajectories are length-optimal between any two successive avoided obstacles. They are generated by smoothly connected lines (common tangents to pairs of obstacles) and arcs of obstacles' boundaries. These trajectories belong to the tangent visibility graph (TVG) (also known as the reduced visibility graph) that was introduced in (Rohnert, 1986) for two-dimensional environments with polygonal obstacles and shown to contain the shortest path, then extended to two-dimensional environments with curved obstacles (Laumond, 1987; Liu and Arimoto, 1992).*

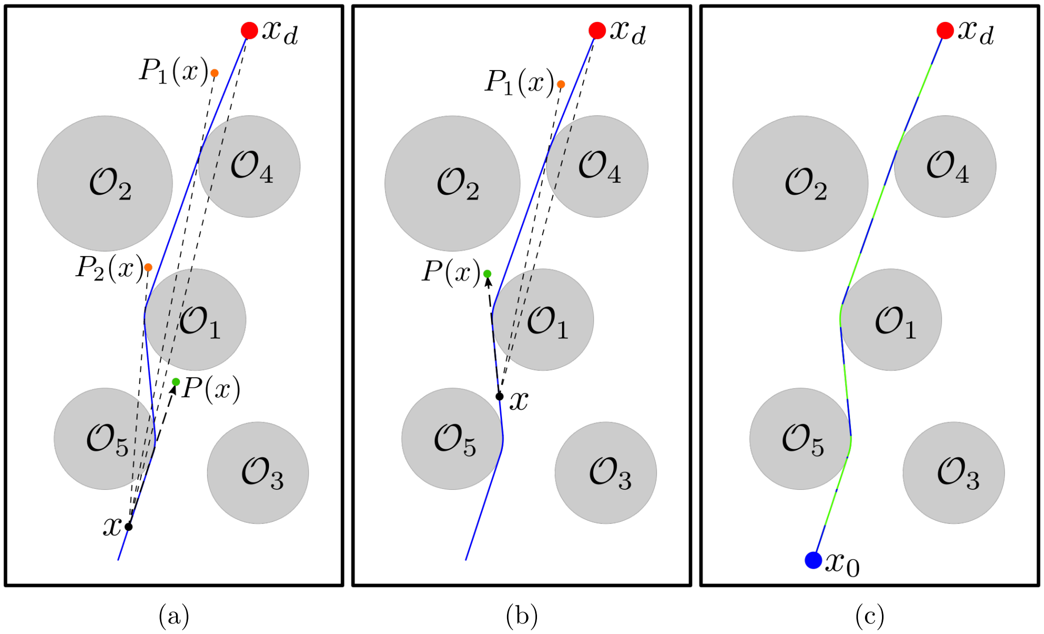


Figure 3.3: Quasi-optimal trajectory in 2D workspace.

## 3.5 Safety and stability analysis

In this section, the safety and stability properties of the closed-loop system (3.7)-(3.22) will be analyzed. Nagumo's theorem ((Nagumo, 1942; Blanchini and Miani, 2007)), offers

[1] [Online]. Available: `https://youtu.be/CzIjtsy6HBA`

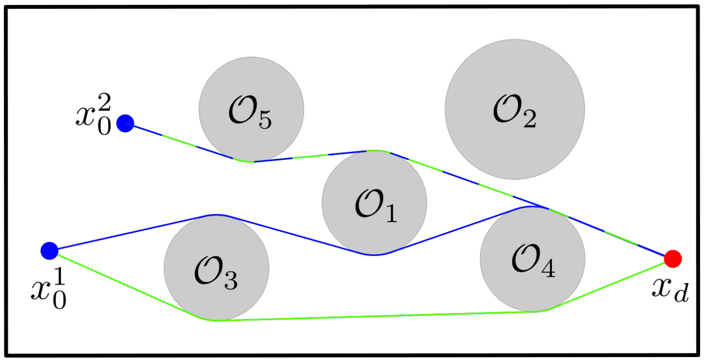


Figure 3.4: Optimal and quasi-optimal trajectories shown in green and blue, respectively.

an important tool to prove safety. One of the statements of this theorem is the one based on Bouligand's tangent defined in (2.15). In our case, when $x \in \mathring{\mathcal{F}}$, the tangent cone is the Euclidean space ($\mathrm{T}_{\mathcal{F}}(x) \equiv \mathbb{R}^n$), and since the free space is a sphere world (smooth boundary), the tangent cone at its boundary is a half-space (see Fig. 3.5). Nagumo's theorem guarantees, in a navigation problem, that the robot stays inside the free space $\mathcal{F}$. For Nagumo's condition to be satisfied, the velocity vector $u(x)$ must point inside (or is tangent to) the free space (Berkane, 2021). In what follows, we rely on Nagumo's theorem to prove the safety of the trajectories generated by our closed-loop system.

**Lemma 3.3 (Safety)** *Consider the closed set $\mathcal{F}$ described in (3.3) and the kinematic system (3.7) under the control law (3.22). Then, the closed-loop system (3.7)-(3.22) admits a unique solution for all $t \geq 0$ and the set $\mathcal{F}$ is forward invariant.*

**Proof** See Appendix A.3.

Let us look for the equilibria of the closed-loop system (3.7)-(3.22) by setting $u(x) = 0$ in (3.22). Then, from the first equation of (3.22), the equilibrium point is $x_d$. From (3.21), one can rewrite the control at step $p \in \{1, \dots, h(x)\}$ and position $x \in \mathcal{AR}_{\iota_x(p)}$, as $u_p = \sin(\beta_i)\sin^{-1}(\theta_i)\|u_{p-1}\|\bar{\xi}_i$[2] where $\iota_x(p) = i$. In the case where $u_{p-1} \neq 0$, and since $\bar{\xi}_i \in \mathbb{S}^{n-1}$, $u_p = 0$ if and only if $\beta_i = 0$. The set of positions leading to $\beta_i = 0$ is the segment (or segments) of the line, tangent to the ancestor obstacle $k = \iota_x(p-1)$, crossing the center of obstacle $i$, within the active region of obstacle $i$. When $\beta_i = 0$, the control input, at step $p-1$, is aligned with $(c_i - x)$, which is also tangent to the ancestor of obstacle $i$. This set, referred to as the set of undesired equilibria generated by obstacle $i$, is shown in Fig. 3.2 and defined as follows:

$$\mathcal{L}_i := \left\{ q \in \mathcal{AR}_i \middle|\ \beta_i(u_{p-1}(q), q) = 0, p = \iota_q^{-1}(i) \right\}. \tag{3.24}$$

Note that the destination $x_d$ is the only desired equilibrium point and that all other equilibria are undesired. The central half-line generated by obstacle $i$ in the workspace, starts from the center $c_i$ and extends the set of undesired equilibria $\mathcal{L}_i$ (as shown in Fig. 3.6), and is defined as $\mathcal{L}_i^e := \mathcal{L}_h(c_i, y - c_i)$, where $y \in \mathcal{L}_i$. Some obstacles may not

[2] For simplicity, the arguments $(x, u)$ for the angles $\beta_i$ and $\theta_i$ are omitted whenever clear from context.

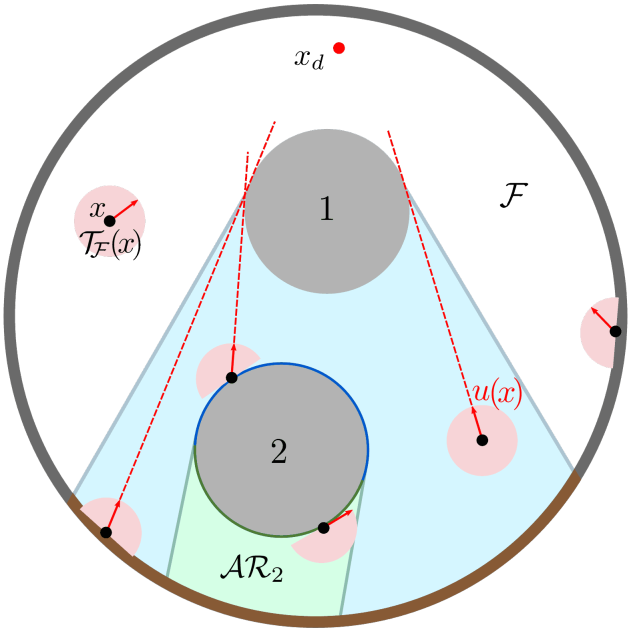


Figure 3.5: Two-dimensional illustration of Bouligand's tangent cones in a sphere world. The pink regions represent Boulgand's tangent cones $\mathcal{T}_{\mathcal{F}}(x)$ at position $x$. Inside the workspace, the Bouligand tangent cone is the Euclidean space, and on the boundary of the workspace, it is a half-space.

generate undesired equilibria, in specific configurations, as will be shown later, and in this case $\mathcal{L}_i$ and $\mathcal{L}_i^e$ are empty sets. Therefore, $u(x) = 0$ if $x \in \mathcal{L}_i$ where $i \in \mathcal{Z}$ and $\mathcal{Z}$ is the set of obstacles generating undesired equilibria. Finally, one can conclude that the set of equilibrium points of the system (3.7)-(3.22) is given by $\zeta := \{x_d\} \cup (\cup_{i\in\mathcal{Z}}\mathcal{L}_i)$. The previous developments can be summarized in the following lemma:

**Lemma 3.4** *All trajectories of the closed-loop system* (3.7)-(3.22) *converge to the set* $\zeta = \{x_d\} \cup (\cup_{i\in\mathcal{Z}}\mathcal{L}_i)$. □

The indices of obstacles crossed by the central half-line $\mathcal{L}_i^e$ of obstacle $\mathcal{O}_i$ are grouped in the set defined as $\mathcal{M}_i := \{j \in \mathbb{I} \setminus \{i\} \| \mathcal{L}_i^e \cap \mathcal{O}_j \neq \varnothing\}$ and $N_i = \mathbf{card}(\mathcal{M}_i)$. Define the map $\kappa_i : \mathcal{M}_i \to \{1, \ldots, N_i\}$ that associates to each index $k \in \mathcal{M}_i$ the corresponding order of the obstacle $\mathcal{O}_k$ according to its proximity with respect to obstacle $\mathcal{O}_i$ among the obstacles of indices in the set $\mathcal{M}_i$, where the order goes from the closest to the farthest obstacle. The set $\mathcal{M}_i^p := \{\kappa_i^{-1}(1), \ldots, \kappa_i^{-1}(p)\}$, $p \leq N_i$, contains the indices of the set $\mathcal{M}_i$ representing the $p$ first obstacles in increasing order of their distance from obstacle $i$, and $\mathcal{M}_i^0 := \varnothing$. In the following lemma, we show that under certain conditions, obstacles in the set $\mathcal{M}_i$ do not generate undesired equilibria.

**Lemma 3.5** *Let* $i \in \mathbb{I}$ *such that* $\mathcal{M}_i \neq \varnothing$. *Obstacles of indices in the set* $\mathcal{M}_i^p$, *where* $p \leq N_i$, *do not generate undesired equilibria if, for all* $k \in \mathcal{M}_i^p$, *the following conditions are satisfied:*

1. $c_k \in \mathring{\mathcal{H}}(x^*_{k,i}, c_i) \cup (\cup_{j\in\mathcal{M}_i^{p-1}} \mathring{\mathcal{H}}(x^*_{k,i}, c_j))$,

2. *$(\mathring{\mathcal{H}}(x^*_{k,i}, c_i) \cup (\cup_{j \in \mathcal{M}_i^{p-1}} \mathring{\mathcal{H}}(x^*_{k,i}, c_j))) \cap \mathcal{O}_l = \varnothing$ for all $l \in \mathbb{I} \setminus (\mathcal{M}_i^p \cup \{i\})$,*

*where $x^*_{k,i} = \arg \max_{q \in \mathcal{L}_i^e \cap \partial\mathcal{O}_k} ||c_i - q||$. Moreover, if $p = N_i$, or ($p < N_i$ and the obstacle of index $k = \kappa_i^{-1}(p+1)$ does not satisfy conditions 1) and 2)), the set $\mathcal{M}_i$ is said to be of order $\bar{N}_i = p$ which is the total number of obstacles, of indices in the set $\mathcal{M}_i$, that do not generate undesired equilibria and the set $\mathcal{M}_i^{\bar{N}_i}$ groups them.*

**Proof** See Appendix A.4.

Lemma 3.5 provides sufficient conditions so that the first $p$ obstacles, with indices in the set $\mathcal{M}_i$ and ordered according to their proximity with respect to obstacle $\mathcal{O}_i$, do not generate undesired equilibria. If $p = N_i$, or ($p < N_i$ and the $(p+1) - th$ obstacle does not satisfy these conditions), the set $\mathcal{M}_i^{\bar{N}_i}$ groups all the obstacles, with indices in the set $\mathcal{M}_i$, that do not generate undesired equilibria, where $\bar{N}_i = p$ is the number of these obstacles and the order of the set $\mathcal{M}_i$. Condition (1) requires the center of each obstacle $k \in \mathcal{M}_i^p$ to be inside the union of the hats of the cones, of vertex $x^*_{k,i}$, enclosing obstacle $i$ and the obstacles of the list $\mathcal{M}_i^p$ closer to obstacle $i$ than obstacle $k$. Condition (2) requires that the union of hats considered in condition (1) does not intersect any obstacle other than those considered in condition (1) (*i.e.,* obstacles $i$, $k$, and the obstacles closer to obstacle $i$ than obstacle $k$ among the list $\mathcal{M}_i^p$). Let us use obstacle 1 in Fig. 3.6 to verify (visually) the two conditions. The union of the hats of cones enclosing obstacles 3 and 4 (blue and green conic subsets in the left figure) includes the center of obstacle 1 and does not intersect with any obstacle other than obstacles 3, 1, and 4. Obstacle 4 satisfies the conditions, but obstacle 2 does not, as its center is outside the union of the hats enclosing obstacles 3, 4, and 1 (red, blue, and green conic subsets in the right figure).

One of the main results of this chapter is stated in the following theorem.

**Theorem 3.1** *Consider the free space $\mathcal{F} \subset \mathbb{R}^n$ described in (3.3), for $n \geq 2$, and the closed-loop system (3.7)-(3.22). Under Assumptions 3.1 and 3.2, the following statements hold:*

*i) The set $\mathcal{F}$ is forward invariant.*

*ii) All trajectories converge to the set $\zeta = \{x_d\} \cup (\cup_{i \in \mathcal{Z}} \mathcal{L}_i)$.*

*iii) The set of equilibrium points $\cup_{i \in \mathcal{Z}} \mathcal{L}_i$ is unstable.*

*iv) The equilibrium point $x_d$ is locally exponentially stable on $\mathcal{F}$.*

*v) The generated trajectories are quasi-optimal.*

**Proof** See Appendix A.5.

Theorem 3.1 shows that the desired equilibrium point $x_d$ is locally exponentially stable and that all trajectories converging to it are safe and *quasi-optimal*, in the sense of Definition 3.1. The region of attraction of the desired equilibrium is characterized in the

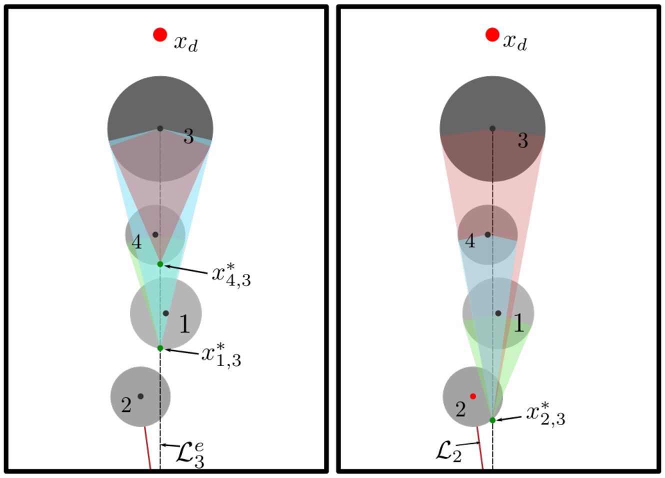


Figure 3.6: Illustration of obstacles that generate, and those that do not generate, undesired equilibria. Obstacles 2 and 3 generate undesired equilibria, but obstacles 1 and 4 do not. Unlike obstacles 1 and 4, obstacle 2 does not satisfy the conditions in Lemma 3.5, where although obstacle 2 is crossed by the central half-line $\mathcal{L}^e_i$ associated with obstacle 3, its center is not included in the hats of the cones enclosing the obstacles preceding it (*i.e.*, obstacle 4, 1 and 3).

next section for two-dimensional workspaces. Unfortunately, a complete characterization of the region of attraction has not been proved for higher dimensions $n \geq 3$. Nevertheless, our insights and extensive simulations in three-dimensional environments led us to conjecture that the equilibrium point $x_d$ is almost globally asymptotically stable, at least for $n = 3$.

## Invariant sets in two-dimensional spaces ($n = 2$)

Let $\mathcal{R}_i := \{k \in \mathcal{Z} | \mathcal{L}^e_k \cap \mathcal{O}_i \neq \varnothing,\, \mathcal{L}_k \cap \mathcal{AR}_i \neq \varnothing\}$ be the set of indices of central half-lines crossing obstacle $i$ and their set of undesired equilibria intersecting with its active region $\mathcal{AR}_i$, and note that $\mathcal{R}_i \neq \varnothing$ for all $i \in \mathbb{I}$. Obstacles crossed by more than one central half-line are represented by the set of indices $\mathbb{L} := \{k \in \mathbb{I} | \mathbf{card}(\mathcal{R}_k) \geq 2\}$. For every $i \in \mathbb{L}$, we select the out-most line segments $\mathcal{L}_k$, $k \in \mathcal{R}_i$, and we determine their intersection with the boundary of obstacle $i$, the left and right intersections being denoted by $y^l_{i,0}$ and $y^r_{i,0}$ respectively. We go through the two out-most line segments separately until they intersect with one of the line segments having an index in the set $\mathcal{R}_i$, or with the boundary of the workspace. We denote the left and right intersections by $y^l_1$ and $y^r_1$, respectively. If the workspace has yet to be reached and $(y^l_1 \neq y^r_1)$, we continue in the same way with the new line segments up to the intersection with the workspace boundary or up to the intersection between the left and right line segments (*i.e.*, $y^l_k = y^r_j$, $k, p > 0$). We group the intersection points obtained on the left and right into two lists, $Y^l_i = \{y^l_{i,0}, y^l_{i,1}, \ldots\}$ and $Y^r_i = \{y^r_{i,0}, y^r_{i,1}, \ldots\}$, respectively (see Fig. 3.7). For every two successive points $\{y^l_{i,p}, y^l_{i,p+1}\}$ of $Y^l_i$ (resp. $\{y^r_{i,p}, y^r_{i,p+1}\}$ of $Y^r_i$), we generate the right (resp. left) half-plane bounded by the line passing through these two points. The intersection of the union of the half-planes of each list forms an area that, when restricted to the active region, gives a characteristic region defined as $\chi_i := \left(\cup_{p=0}^{\mathbf{card}(Y_i^r)-2}\mathcal{P}_{\geq}\left(y^r_{i,p}, R(y^r_{i,p} - y^r_{i,p+1})\right)\right) \cap \left(\cup_{p=0}^{\mathbf{card}(Y_i^l)-2}\mathcal{P}_{\leq}\left(y^l_{i,p}, R(y^l_{i,p} - y^l_{p+1})\right)\right) \cap \mathcal{AR}_i$ where $R = \left(\begin{smallmatrix} 0 & 1 \\ -1 & 0 \end{smallmatrix}\right)$. Finally, we create a cell, deleting the characteristic regions of other obstacles inside the characteristic region of obstacle $i$, and define it as follows $\mathbf{Cell}_i := \chi_i \setminus \cup_{k \in \mathbb{L}_i} \mathring{\chi}_k$ where $\mathbb{L}_i := \{k \in \mathbb{L} | \chi_i \cap \chi_k \neq \varnothing; \forall x \in \chi_i \cap \chi_k, \iota_x^{-1}(k) > \iota_x^{-1}(i)\}$. Note that these cells are constructed so that their boundaries are formed by the undesired equilibria and the boundary of the free space, which endows them with the invariance property stated in the following lemma.

**Lemma 3.6** *Let $i \in \mathbb{L}$. The cell $\mathbf{Cell}_i$ is forward invariant for the closed-loop system* (3.7)-(3.22).

**Proof** See Appendix A.6.

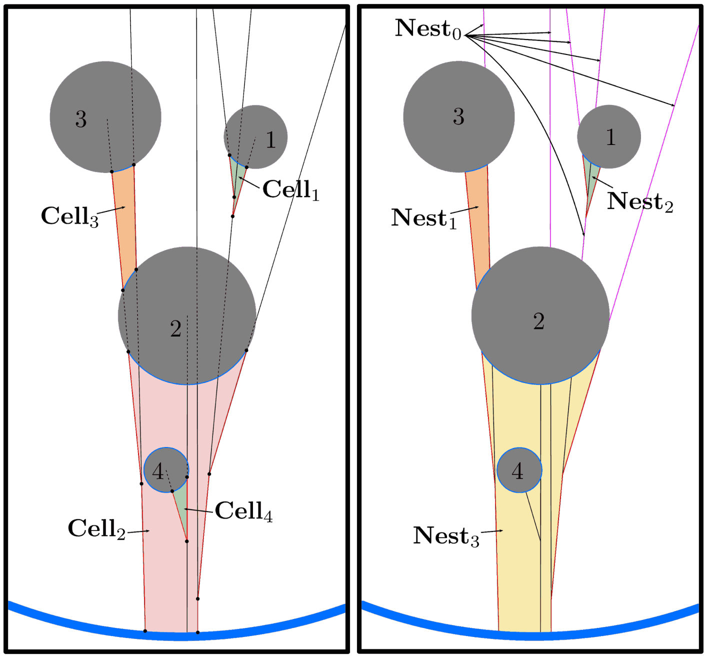


Figure 3.7: Illustration of invariant cells and nests. The figure on the left shows an example of four cells associated with four obstacles, where the boundaries of each cell are formed by the boundary of the free space (blue arcs) and the undesired equilibria (red segments). The figure on the right shows the nests obtained from the cells in the figure on the left. **Nest**$_1$ and **Nest**$_2$ are equal to **Cell**$_3$ and **Cell**$_1$, respectively. **Nest**$_3$ is the union of the two adjacent cells, **Cell**$_2$ and **Cell**$_4$, and **Nest**$_0$ is a special nest that includes all the undesired equilibria not included in the other nests.

Two cells are adjacent if they share undesired equilibria on their boundary, which is true only if $\partial\mathbf{Cell}_i \cap \partial\mathbf{Cell}_k \cap (\cup_{i\in\mathcal{Z}}\mathcal{L}_i) \neq \varnothing$. We construct nests by the union of adjacent cells, where each cell has at least one adjacent cell among the cells in that nest. **Cell**$_2$ and **Cell**$_4$

in Fig. 3.7 are examples of adjacent cells forming a nest, the union of which constitutes $\mathbf{Nest}_3$. Cells without adjacent cells form a nest with a single element, as illustrated by $\mathbf{Cell}_1$ forming $\mathbf{Nest}_2$ and $\mathbf{Cell}_3$ forming $\mathbf{Nest}_1$ in Fig. 3.7. We also construct a special nest whose cells are segments of undesirable equilibria that belong to no other regular cell, as illustrated by $\mathbf{Nest}_0$ in Fig. 3.7. Since nests are the union of invariant cells or of undesired equilibria (the special nest), nests are invariant and are denoted by $\mathbf{Nest}_k$ where $\mathbf{Nest}_0 := \cup_{i \in \mathcal{Z}} \mathcal{L}_i \setminus \cup_{k \in \mathbb{L}} \mathbf{Cell}_k$ is the special nest. Unfortunately, a nest can form a barrier around the workspace, reducing the navigable area of the free space. Such a nest can be generated by creating a circular band of adjacent cells, as shown in Fig. 3.8. In the following lemma, nests are shown to be the attraction region of the undesired equilibria.

**Lemma 3.7** *The set $\cup_k \boldsymbol{Nest}_k$ is the region of attraction of the undesired equilibria $\cup_{i \in \mathcal{Z}} \mathcal{L}_i$.*

**Proof** See Appendix A.7.

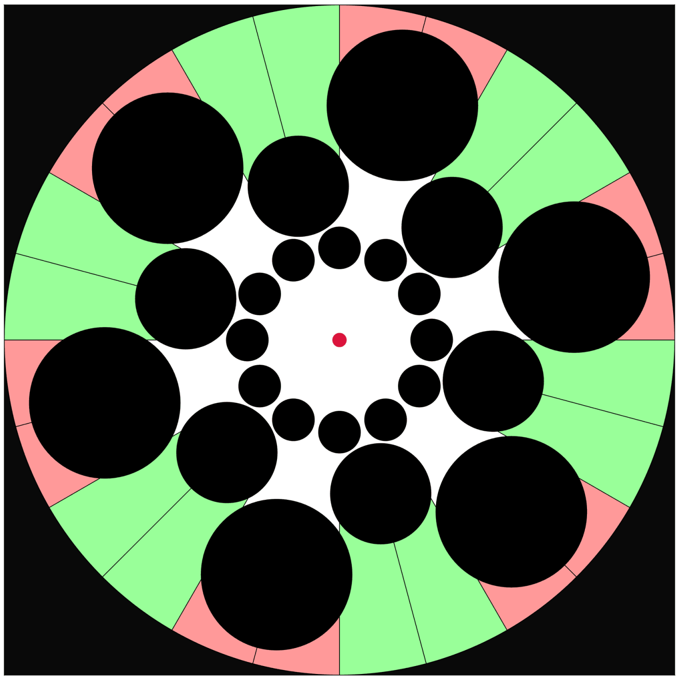


Figure 3.8: Quasi-non-navigable two-dimensional space.

Now, to ensure almost global asymptotic stability of the equilibrium point $x_d$ in two-dimensional spaces, we reduce the nests to the set of undesired equilibria by imposing the following assumption:

**Assumption 3.3** *For any $i \in \mathbb{I}$ and $k \in \mathcal{Z}$ where $i \neq k$, $\mathcal{L}_k^e \cap \mathcal{O}_i = \varnothing$, or ($\mathcal{L}_k^e \cap \mathcal{O}_i \neq \varnothing$ and $i \in \mathcal{M}_k^{\bar{N}_i}$.)*

Assumption 3.3 rules out the possibility of creating the invariant cells by imposing obstacle configurations such that $\mathbb{L} = \varnothing$ making the undesired equilibria repellers. In addition to the results of Theorem 3.1, the next theorem characterizes the attraction region of the undesired equilibria and shows almost global asymptotic stability of the destination under Assumption 3.3.

**Theorem 3.2** *Consider the free space $\mathcal{F} \subset \mathbb{R}^n$ described in (3.3), for $n = 2$, and the closed-loop system (3.7)-(3.22). Let Assumptions 3.1 and 3.2 hold. Then, items i), ii), iii), iv) of Theorem 3.1, and the following statements hold:*

*i) The equilibrium point $x_d$ is attractive from all $x(0) \in \mathcal{F} \setminus \cup_k \boldsymbol{Nest}_k$.*

*ii) From any initial position $x(0) \in \mathcal{F} \setminus \cup_k \boldsymbol{Nest}_k$, the trajectory $x(t)$ is quasi-optimal.*

*iii) Under Assumption 3.3, $\cup_k \boldsymbol{Nest}_k = \cup_{i \in \mathcal{Z}} \mathcal{L}_i$ and the destination $x_d$ is almost globally asymptotically stable.*

**Proof** See Appendix A.8.

Theorem 3.2 shows that the target location is attractive from any position in the free space, except from the nests (region of attraction of the undesired equilibria), which reduces, under Assumption 3.3, to the undesired equilibria $\cup_{i \in \mathcal{Z}} \mathcal{L}_i$ having zero Lebesgue measure. Fortunately, the nests will naturally disappear in the sensor-based case as we will see in the next section.

## 3.6 Sensor-based navigation using a 2D LiDAR range scanner

We now present a version of our approach, using a LiDAR range scanner, in an unknown two-dimensional sphere world. Assume that the robot is equipped with a sensor of 360° angular sensing range, a resolution $d\theta > 0$, and a radial sensing range $R > 0$. The measurements of the sensor, at a position $x$, are modeled by the polar curve $\rho(x, \theta) : \mathcal{F} \times \hat{\mathcal{A}} \to [0, R]$, where $\hat{\mathcal{A}} := \{0, d\theta, 2d\theta, \ldots, 360 - d\theta\}$ is the set of scanned angles, defined as follows:

$$\rho(x, \theta) := \min \left( R, \min_{\substack{y \in \partial\mathcal{F} \\ \overline{\text{atan2}}(y - x) = \theta}} \|x - y\|, \right), \tag{3.25}$$

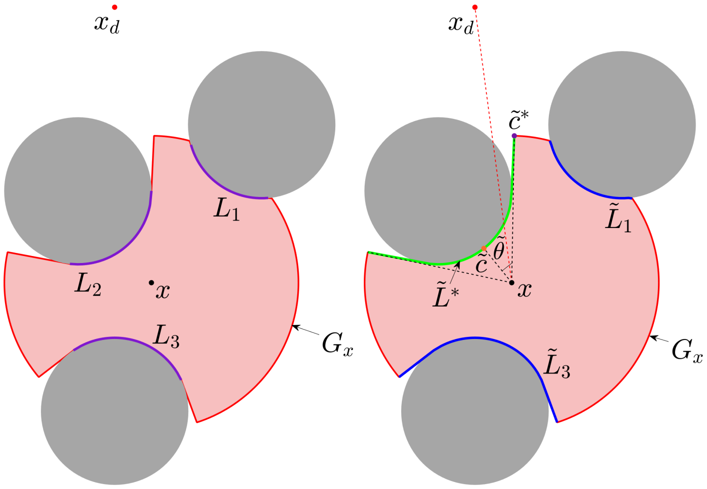
where $\overline{\text{atan2}}(v) = \text{atan2}(v(2), v(1))$ for $v \in \mathbb{R}^2$.
The Cartesian coordinates of the scanned points are modeled by the mapping $\delta(x, \theta) : \mathcal{F} \times \hat{\mathcal{A}} \to \mathcal{F}$ defined as follows:

$$\delta(x, \theta) := x + \rho(x, \theta)[\cos(\theta)\ \sin(\theta)]^\top. \tag{3.26}$$

Let $G_x(\delta)$ be the graph of the mapping $\delta$ at a position $x$ (red curve in Fig. 3.9). The set $\mathbb{I}_x \subset \mathbb{I}$ of the detected obstacles is defined as $\mathbb{I}_x := \{i \in \mathbb{I} | d(x, \mathcal{O}_i) \leq R\}$. Assume that at each position $x$, the sensor returns a list of arcs $\mathcal{LA}(x) := \left\{L_1, L_2, \ldots, L_{\tau(x)}\right\}$ from the detected obstacles corresponding to the intersection of the graph $G_x(\delta)$ and obstacles of the set $\mathbb{I}_x$, where $\tau(x) = \mathbf{card}(\mathbb{I}_x)$ as shown in Fig. 3.9(a) by the magenta arcs. Since the available information about the environment is limited by the graph $G_x(\delta)$, successive projections are impossible to apply. Therefore, we apply the single obstacle control strategy given by

$$u(x) = \begin{cases} u_d(x), & x \in \mathcal{VI}, \\ u_d(x) - \|u_d(x)\|\dfrac{\sin(\theta_i - \beta_i)}{\sin(\theta_i)}V_{ci}, & x \in \mathcal{D}(x_d, c_i), \end{cases}$$

where $\theta_i$ and $V_{ci} = (c_i - x)/\|c_i - x\|$ are, respectively, the aperture and the axis of the enclosing cone, $\beta_i$ is the angle between $u_d$ and $(c_i - x)$, and $\mathcal{D}(x_d, c_i)$ is the shadow region. To adapt the above control strategy to the sensor-based case, one proceeds as follows. At each position $x$, the detected arcs $\mathcal{LA}(x)$ are considered as obstacles. The arc crossed by the segment $\mathcal{L}(x, x_d)$ will help to create a virtual enclosing cone onto which the projection is performed. However, due to the practical model of the sensor (LiDAR), which may have low resolutions, safety is not always guaranteed when the robot is in the neighborhood of the obstacles where the velocity vector (projection of $u_d$ onto the virtual enclosing cone) may point inside the obstacle (see Fig. 3.10(a)). To overcome this problem, a list of extended arcs $\mathcal{LA}_e(x) := \left\{\tilde{L}_1, \tilde{L}_2, \ldots, \tilde{L}_{\tau(x)}\right\}$ is defined, where the endpoints of an arc $L_i$ are moved through the graph $G_x(\delta)$ until they have a radial polar coordinate equal to $R$ or they meet the endpoints of the neighboring arcs, as shown in Fig. 3.9(b). Among the extended arcs of the list $\mathcal{LA}_e(x)$, the active extended arc crossed



(a) (b)

Figure 3.9: The sensor-based control strategy. Figure (a) shows the (purple) arcs returned by the sensor from the obstacles detected at position $x$. Figure (b) shows the extended version of the detected arcs (blue arcs) and the virtual cone enclosing the active extended arc (green arc) with the necessary practical parameters for the sensor-based control law.

by the segment $\mathcal{L}(x, x_d)$ is selected and denoted by $\tilde{L}^*$. The active extended arc serves as an obstacle enclosed by a virtual cone (see Fig. 3.9(b)) from which we extract the following practical parameters:

- The virtual center

$$\tilde{c} := \arg\min_{y \in \tilde{L}^*} \|x - y\|, \tag{3.27}$$

  which gives the direction $(\tilde{c} - x)$.

- The virtual aperture

$$\tilde{\theta} := \angle(\tilde{c} - x, \tilde{c}^* - x), \tag{3.28}$$

  where $\tilde{c}^*$ is the endpoint of $\tilde{L}^*$ such that $u_d$ is between the directions $(\tilde{c} - x)$ and $(\tilde{c}^* - x)$.

- The angle

$$\tilde{\beta} := \angle(\tilde{c} - x, u_d). \tag{3.29}$$

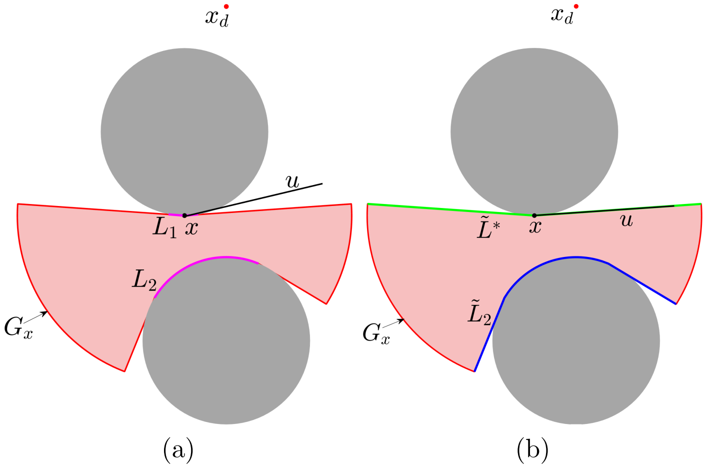


Figure 3.10: Safety consideration in a sensor-based case. In Fig. (a), the projection $u$ of $(x_d - x)$ onto the cone enclosing the arc $L_1$ of the list $\mathcal{LA}(x)$ fails to satisfy the safety condition where $u$ crosses the obstacle. In Fig. (b), the projection lies on the active arc $\tilde{L}^*$ of the list $\mathcal{LA}_e(x)$ and meets the safety condition.

Before defining the new blind and visible sets, let us define the truncated shadow region by

$$\mathcal{D}^t(x_d, c_i) := \mathcal{D}(x_d, c_i) \setminus \bigcup_{j \in \mathcal{PR}_i} \mathcal{D}(x_d, c_j), \tag{3.30}$$

where $\mathcal{PR}_i := \big\{j \in \mathbb{I} | \mathcal{D}(x_d, c_i) \cap \mathcal{D}(x_d, c_j) \neq \varnothing,\, d(x_d, \mathcal{O}_i) < d(x_d, \mathcal{O}_j)\big\}$ is the progeny of obstacle $i$ (see Fig. 3.11(a)). Since the visibility of the robot is limited to the scanning range of the sensor, let us define the practical shadow region of an obstacle $i$ as follows:

$$\tilde{\mathcal{D}}(x_d, c_i, R) := \mathcal{D}^t(x_d, c_i) \cap \mathcal{B}(c_i, r_i + R). \tag{3.31}$$

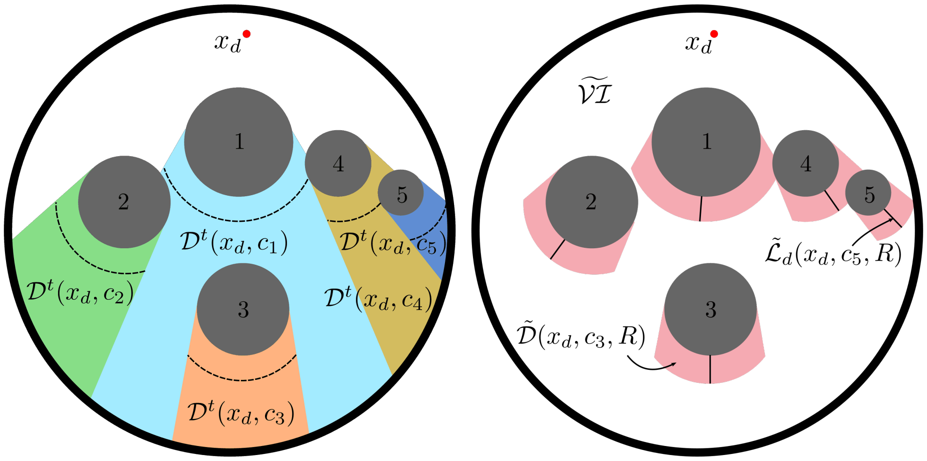
Therefore, the practical blind set is defined as follows:

$$\widetilde{\mathcal{BL}} := \bigcup_{i \in \mathbb{I}} \tilde{\mathcal{D}}(x_d, c_i, R), \tag{3.32}$$

The practical visible set is then defined as $\widetilde{\mathcal{VL}} := \widetilde{\mathcal{BL}}^c_{\mathcal{F}}$ (see Fig. 3.11(b)). Finally, the control is given by

$$u(x) = \begin{cases} u_d(x), & x \in \widetilde{\mathcal{VI}}, \\ u_d(x) - \|u_d(x)\| \dfrac{\sin(\tilde{\theta} - \tilde{\beta})}{\sin(\tilde{\theta})} \dfrac{\tilde{c} - x}{\|\tilde{c} - x\|}, & x \in \widetilde{\mathcal{BL}}. \end{cases} \tag{3.33}$$

The implementation of the sensor-based control strategy is summarized in Algorithm 2


$x_d$
1
2
3
4
5
$\mathcal{D}^t(x_d, c_1)$
$\mathcal{D}^t(x_d, c_2)$
$\mathcal{D}^t(x_d, c_3)$
$\mathcal{D}^t(x_d, c_4)$
$\mathcal{D}^t(x_d, c_5)$
$x_d$
$\widetilde{\mathcal{VI}}$
1
2
3
4
5
$\tilde{\mathcal{L}}_d(x_d, c_5, R)$
$\tilde{\mathcal{D}}(x_d, c_3, R)$

(a) (b)

Figure 3.11: Illustration of the workspace for the sensor-based case. Fig. (a) shows the truncated shadow regions of each obstacle where obstacles $\{2, 3, 4\}$ are the progeny of obstacle 1, and obstacle $\{5\}$ is the progeny of obstacle 4. Fig. (b) highlights the practical shadow regions of each obstacle in pink where the union of theses regions represents the practical blind set while the white region represents the practical visible set.

(see also Fig. 3.9).

The sensor-based control strategy (3.33) can be seen as a special case of the control strategy in the map-based scenario (*a priori* known environments) if each obstacle is considered as the unique obstacle in the workspace. In this way, the active regions become the disjoint practical shadow regions that will limit the undesired equilibria generated by each obstacle to its own practical shadow region excluding the possibility of creating invariant cells. The following lemma determines the equilibria of the closed-loop system (3.7)-(3.33) shown in Fig. 3.11(b).

**Algorithm 2** Implementation of the control law (3.33) in the closed-loop (3.8)

```
Initialization: $x_d$, $e_c$;
while true do
    Measure $x$, $G_x(\delta)$, and $\mathcal{LA}(x)$.
    if $\|x - x_d\| \leq e_s$ then
        Break;
    else
        if $\mathcal{LA}(x) \neq \varnothing$ then
            Construct $\mathcal{LA}_e(x)$.
            if $\mathcal{L}(x_d, x)$ crosses one of the extended arcs of $\mathcal{LA}_e(x)$ then
                Identify $\tilde{L}^*$.
                Determine $\tilde{c}$, $\tilde{\theta}$ and $\tilde{\beta}$ using equations (3.27), (3.28) and (3.29), respectively.
                $u \leftarrow u_d(x) - \|u_d(x)\| \frac{\sin(\tilde{\theta}-\tilde{\beta})}{\sin(\tilde{\theta})} \frac{\tilde{c}-x}{\|\tilde{c}-x\|}$;
            else
                $u \leftarrow u_d$;
            end if
        else
            $u \leftarrow u_d$;
        end if
        Execute $u$ in (9);
    end if
end while
```

**Lemma 3.8** *All trajectories of the closed-loop system (3.7)-(3.33) converge to the set* $\zeta = \{x_d\} \cup \Big(\cup_{i\in\mathbb{I}}\tilde{\mathcal{L}}_d(x_d, c_i, R)\Big)$ *where* $\tilde{\mathcal{L}}_d(x_d, c_i, R) := \mathcal{L}_h(c_i, c_i - x_d) \cap \tilde{\mathcal{D}}(x_d, c_i, R)$.

**Proof** See Appendix A.9.

Lemma 3.8 shows that the set of undesirable equilibria of the closed-loop system (3.7)-(3.33), associated with an obstacle $\mathcal{O}_i$, is the line segment starting from the antipodal point of the destination on obstacle $\mathcal{O}_i$ to the boundary of the practical shadow region. The next theorem states formally the properties of the sensor-based control strategy in two-dimensional sphere worlds.

**Theorem 3.3** *Consider the free space* $\mathcal{F} \subset \mathbb{R}^n$ *described in (3.3), for* $n = 2$, *and the closed-loop system (3.7)-(3.33). Under Assumptions 3.1 and 3.2, the following statements hold:*

*i) The set* $\mathcal{F}$ *is forward invariant.*

*ii) All trajectories converge to the set* $\zeta = \{x_d\} \cup \Big(\cup_{i\in\mathbb{I}}\tilde{\mathcal{L}}_d(x_d, c_i, R)\Big)$.

*iii) The set of undesired equilibria* $\cup_{i\in\mathbb{I}}\tilde{\mathcal{L}}_d(x_d, c_i, R)$ *is unstable.*

*iv) The equilibrium point $x_d$ is almost globally asymptotically stable on $\mathcal{F}$.*

**Proof** See Appendix A.10.

Theorem 3.3 provides the stability results obtained with the reactive sensor-based feedback control strategy, relying only on local information provided by the sensor, which is more practical and realistic than the global approach which requires *a priori* knowledge of the entire workspace. Almost global asymptotic stability is guaranteed without imposing restrictions on the obstacle configurations as in Assumption 3.3. However, the control continuity and quasi-optimality of the generated trajectories are no longer guaranteed. Fig. 3.12 shows an example of a discontinuity in our control at time $t = t'$ when the active arc $\tilde{L}^*$ passed from one obstacle to another, resulting in a sudden change in the control's direction to follow the tangent of the new obstacle.

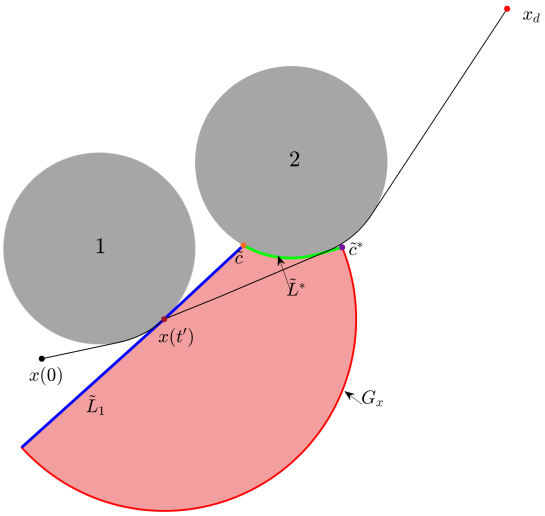


Figure 3.12: A scenario of a discontinuity occurrence in a 2D workspace populated by two obstacles at time $t = t'$. Initially, the active arc is the arc detected from obstacle 1, but arriving at position $x(t')$, it switches to the arc detected from obstacle 2, changing the robot's heading abruptly from the tangent to obstacle 1 to the tangent to obstacle 2 (pointing towards the endpoint $\tilde{c}^*$), creating a discontinuity in the control.

### 3.6.1 Convex obstacles

We consider convex sets with smooth boundaries $\tilde{\mathcal{O}}_i$ as obstacles. The free space must satisfy the separation conditions of Assumptions 3.1 and 3.2. We also assume that the following curvature condition (see, *e.g.,* (Arslan and Koditschek, 2019)) is satisfied.

**Assumption 3.4** *Obstacles are sufficiently curved at their farthest point from the target location $x_d \in \mathcal{F}$, i.e.,*

$$\tilde{\mathcal{O}}_i \subset \mathcal{B}(x_d, \|x_d - \mathrm{x}_i\|), \ \forall i \in \mathbb{I}, \tag{3.34}$$

*where* $\mathrm{x}_i \in \Big\{q \in \partial\tilde{\mathcal{O}}_i | ds_i(q)^\top(x_d - q)/\|x_d - q\| = 1\Big\}$*, and* $ds_i(q) \in \mathbb{S}^{n-1}$ *is the inward-directed gradient of the surface of obstacle* $\tilde{\mathcal{O}}_i$ *at position* $q \in \partial\tilde{\mathcal{O}}_i$*.*

Assumption 3.4 somewhat limits the flatness of an obstacle at its farthest point from the target, as illustrated in Figure 3.13.
The shadow region for a convex obstacle is redefined as $\mathcal{D}(x_d, i) := \Big\{q \in \mathcal{F} | \mathcal{L}(x_d, q) \cap \tilde{\mathcal{O}}_i \neq \varnothing\Big\}$, where the center is replaced by the index of the obstacle as a parameter (see Fig. 3.13). The practical shadow region is then defined as $\tilde{\mathcal{D}}(x_d, i, R) := \Big\{q \in \mathcal{D}^t(x_d, i) | d(q, \tilde{\mathcal{O}}_i) \leq R\Big\}$, where $\mathcal{D}^t(x_d, i)$ is the truncated shadow region defined in (3.30), substituting the center with the obstacle's index. The practical parameters and the control are the same as in (3.33). The next lemma provides the set of equilibria of the closed-loop system (3.7)-(3.33) in the case of convex obstacles.

**Lemma 3.9** *All trajectories of the closed-loop system* (3.7)-(3.33) *converge to the set* $\tilde{\zeta} = \{x_d\} \cup \Big(\cup_{i \in \mathbb{I}} \tilde{\mathcal{L}}_d(x_d, \mathrm{x}_i, R)\Big)$*, where* $\tilde{\mathcal{L}}_d(x_d, \mathrm{x}_i, R) := \mathcal{L}_h(\mathrm{x}_i, \mathrm{x}_i - x_d) \cap \tilde{\mathcal{D}}(x_d, i, R)$*.*

**Proof** See Appendix A.11.

In addition to the destination, Lemma 3.9 shows that the equilibrium points are the positions aligned with their projection $\tilde{c}$ and the destination. The same sensor-based procedure applied in sphere worlds is used, except that the elements of the list of arcs are not arcs but convex portions of the detected obstacles. In the case of convex obstacles with non-smooth boundaries, the procedure is modified where the endpoints of each detected portion are dilated with a ball of radius $r > 0$, and the cone enclosing the segment crossed by $\mathcal{L}(x, x_d)$ is enlarged, as shown in Fig. 3.14. The objective of dilating the endpoints is to smooth the corners of the obstacles. When an endpoint coincides with a non-smooth corner of an obstacle, the dilated endpoint will help to generate a smooth

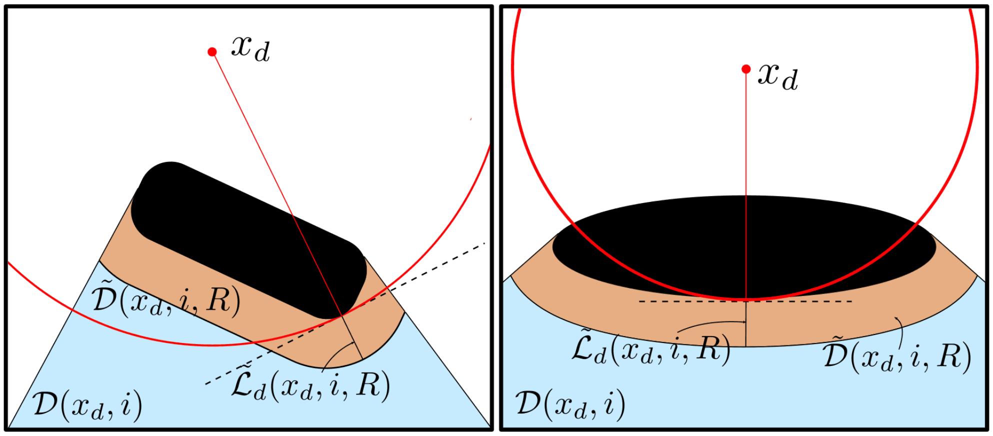

$x_d$
$\tilde{\mathcal{D}}(x_d, i, R)$
$\tilde{\mathcal{L}}_d(x_d, i, R)$
$\mathcal{D}(x_d, i)$
$x_d$
$\tilde{\mathcal{L}}_d(x_d, i, R)$
$\tilde{\mathcal{D}}(x_d, i, R)$
$\mathcal{D}(x_d, i)$

Figure 3.13: Representation of the shadow region and the practical shadow region of a convex obstacle. In the figure on the left, the obstacle satisfies the curvature condition, while in the figure on the right, the obstacle does not satisfy this condition.

trajectory. If the endpoint dilation is applied all around the boundary of an obstacle, a dilated version of this obstacle, given by $\tilde{\mathcal{O}}_i^r = \tilde{\mathcal{O}}_i \oplus \mathcal{B}(0, r)$, will be generated. Therefore, the new free space will be $\mathcal{F}_r := \mathcal{W} \setminus \bigcup\limits_{i=1}^{m} \mathring{\tilde{\mathcal{O}}}_i^r$ and the minimum separation distance will be greater than $2r$.

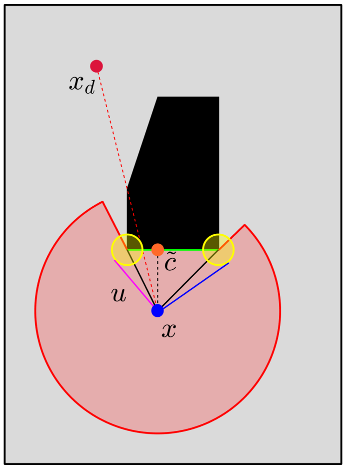


Figure 3.14: Smoothing of polygonal obstacle corners.

**Theorem 3.4** *Consider the free space $\mathcal{F} \subset \mathbb{R}^n$ described in (3.3), in the case of convex obstacles and $n = 2$ , and the closed-loop system (3.7)-(3.33). Under Assumptions 3.1, 3.2, and 3.4, the following statements hold:*

*i) The set $\mathcal{F}$ is forward invariant.*

*ii) All trajectories converge to the set $\tilde{\zeta} = \{x_d\} \cup \Big(\cup_{i \in \mathbb{I}} \tilde{\mathcal{L}}_d(x_d, \mathrm{x}_i, R)\Big)$.*

*iii) The set of undesired equilibria $\cup_{i \in \mathbb{I}} \tilde{\mathcal{L}}_d(x_d, \tilde{x}_i, R)$ is unstable.*

*iv) The equilibrium point $x_d$ is almost globally asymptotically stable on $\mathcal{F}$.*

**Proof** See Appendix A.12.

Theorem 3.4 shows that the sensor-based strategy designed for sphere worlds extends to convex worlds with obstacles satisfying the curvature condition of Assumption 3.4, and the results are preserved. For convex obstacles with non-smooth boundaries, we consider their dilated version $\tilde{\mathcal{O}}_i^r$ and the free space $\mathcal{F}_r$ which amounts to the case of obstacles with smooth boundaries.

## 3.7 Numerical simulations

To explore the extent of what our *quasi-optimal* trajectories can offer in terms of the shortest path in the multiple obstacle case, we compare the trajectories generated by our approach with the shortest paths obtained with Dijkstra's algorithm (DA) on a tangent visibility graph (TVG). We used 10 different and highly congested two-dimensional environments and 100 randomly selected initial positions in each environment. The percentage of perfect matches of the paths is reported in Table 3.1, which shows a high rate of success. Fig. 3.15 shows a sample of 10 trajectories generated from 10 randomly selected initial positions in two of the ten environments used in our simulations. A simulation video can be found online.[3] The effect of successive projections on the optimality of the path generated by our approach is illustrated in Fig. 3.16, where one can see that the path generated by our approach coincides with the shortest path in a single-obstacle workspace, while it does not in a two-obstacle workspace.

Table 3.1: Number of perfect matches between the paths generated by our approach (Algorithm 1) and those found by DA in tangent visibility graphs, for 100 runs with 100 randomly selected initial positions.

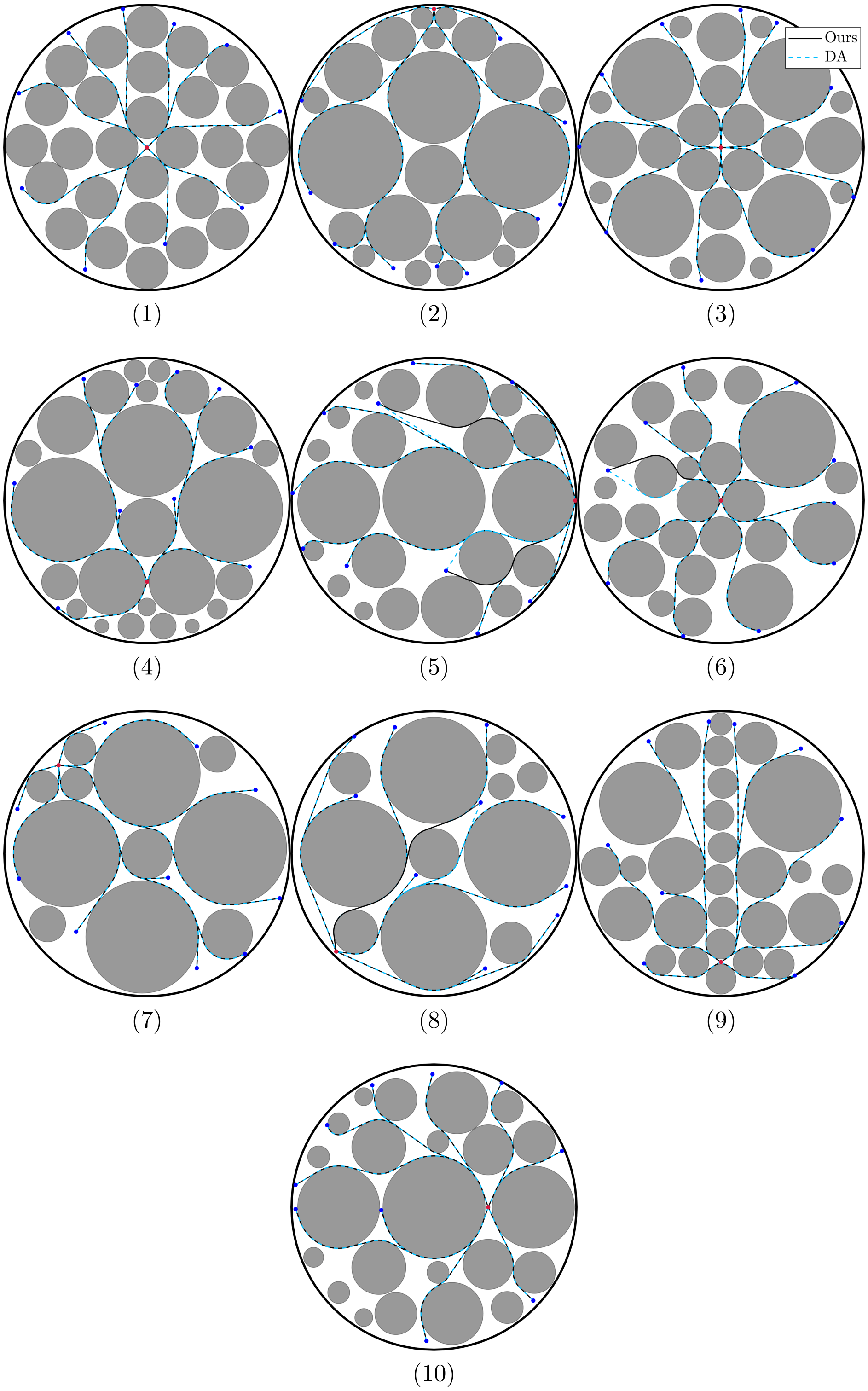
| **Space 1** | **Space 2** | **Space 3** | **Space 4** | **Space 5** |
|---|---|---|---|---|
| 100% | 98% | 100% | 100% | 81% |
| **Space 6** | **Space 7** | **Space 8** | **Space 9** | **Space 10** |
| 96% | 99% | 94% | 94% | 99% |

**Remark 3.3** *The combination TVG-DA has been used as a benchmark to test the optimality of the paths generated by our approach. The advantages of our approach w.r.t. the TVG-DA are as follows:*

- *We solve the problem from a control perspective, as our solution is feedback-based, allowing us to solve the navigation problem in one go, whereas TVG-DA only gives the shortest path to be tracked by another feedback controller.*
- *We propose a closed-form solution* (3.22) *to the autonomous navigation problem, which is more suitable for real-time implementations (computationally efficient) than searching tangent visibility graphs.*
- *We solve the navigation problem in $n$-dimensional sphere worlds while the TVG-DA is limited to paths in two-dimensional sphere worlds as the TVG is infinite in three and higher-dimensional environments.*

To visualize the properties of our approach, we consider two different scenarios. In the first scenario, we assume that the robot evolves in $\mathbb{R}^2$ where the workspace contains twenty-six obstacles, and the destination is $x_d = [0\ 0]^\top$. We run the simulation from fifteen different initial positions. In the second scenario, the considered space is $\mathbb{R}^3$, where the workspace contains eighteen obstacles, and the goal is $x_d = [0\ 0\ 0]^\top$. We run the simulation from eighteen different initial positions. A comparison of our approach with the navigation function approach (NF) (Koditchek and Rimon, 1990) and the separating

[3] [Online]. Available: `https://youtu.be/SE8w8vabxVE`



(1) (2) (3) (4) (5) (6) (7) (8) (9) (10)

Figure 3.15: Trajectories generated by our approach, Algorithm 1, (black solid trajectories) and the optimal path found by DA in tangent visibility graphs (blue dashed trajectories), in 10 different environments. The target location is indicated with a red dot. Only 10 trajectories among 100 starting from randomly selected initial positions were plotted, while the summary of the complete experiments is reported in Table 3.1, where each subfigure illustrates the results in the corresponding space in Table 3.1. Spaces (5), (6), and (8) show examples where the quasi-optimal trajectories do not match the shortest path obtained by the DA.

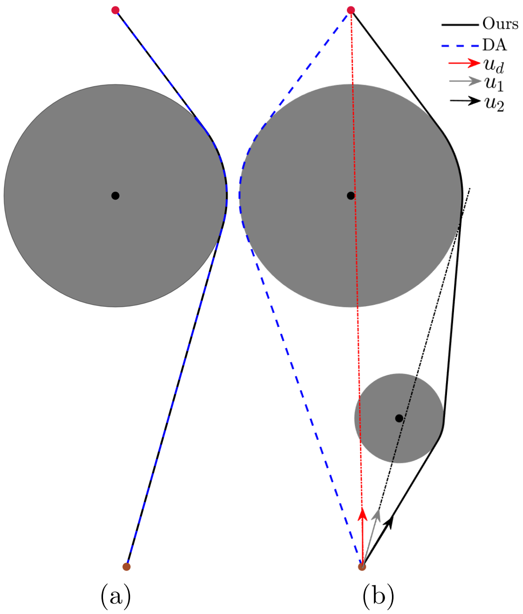


Figure 3.16: Effect of the nested projections on the optimality of the generated trajectory. In Fig. (a), a single obstacle is considered, and the trajectory generated by our approach corresponds to the DA trajectory (shortest path). In Fig. (b), a second obstacle is considered. The trajectory generated by our approach differs from the DA trajectory (shortest path).

hyperplane approach with the Voronoi-adjacent obstacle sensing model (SH) (Arslan and Koditschek, 2019) is established in the two-dimensional space. The simulation results in Fig. 3.17 and 3.18 show that all the trajectories generated by our control strategy are safe and converge to the red target. In addition, Fig. 3.17 shows the superiority of our approach over the two other methods in terms of the length of the generated collision-free paths where it generates the same paths as DA. Moreover, Table 3.2 reports the relative length difference of the paths generated by the NF and SH approaches with respect to our approach. For each initial position $p_i$, $i \in \{1, \ldots, 15\}$, in Fig. 3.17, we computed the relative length difference $RLD_i^a = 100(l_i^a - l_i^0)/l_i^0$, $a \in \{NF, SH\}$, where $l_i^{NF}$ (resp. $l_i^{SH}$) is the length of the $i$th path generated by the NF approach (resp. SH approach), and $l_i^0$ is the length of the path generated by our approach. The positive numbers in Table 3.2 indicate that, for all 15 initial conditions, our approach generated shorter paths than the NF and SH approaches. This superiority is mainly due to the uncontrolled repulsion exerted by the obstacles on the robot in the NF and SH approaches. It becomes clear in the single obstacle case where the robot is repelled even if it has a clear line-of-sight to the destination, which is shown in the simulation result in Fig. 3.19, where the pink initial positions are in the visible set while the green initial positions are in the shadow region. The trajectories generated by our approach are the shortest in terms of distance, as shown in Lemma 3.1. The simulation video of Fig. 3.18 can be found online.[4]

To test our control in a two-dimensional space that does not satisfy Assumption 3.3,

[4] [Online]. Available: `https://youtube.com/shorts/yJCdRLdQHnc`

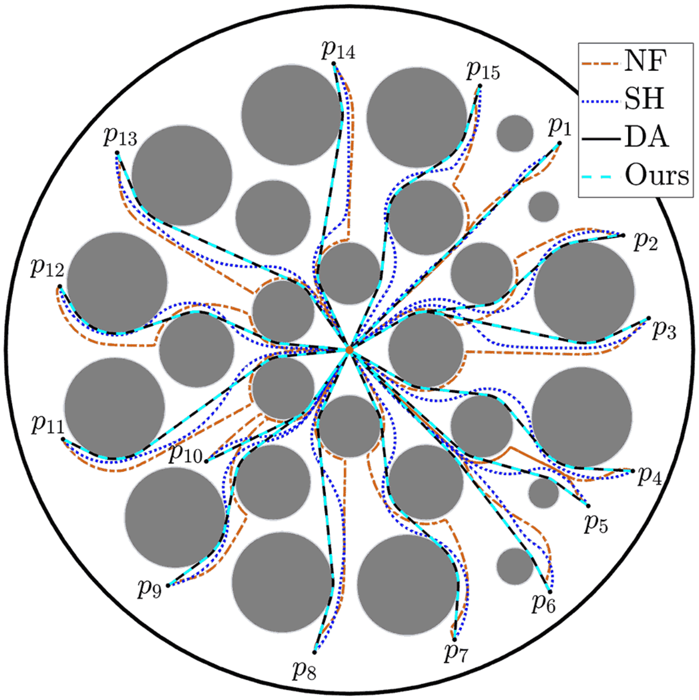


Figure 3.17: Trajectories generated by our approach, SH, NF and DA in a two-dimensional sphere world.

Table 3.2: The relative length difference of the paths, shown in Fig. 3.17, generated by the NF and SH approaches with respect to our approach.

| Paths | $RLD^{NF}$ (%) | $RLD^{SH}$ (%) |
|---|---|---|
| $p_1$ | 1.18 | 0.27 |
| $p_2$ | 11.36 | 7.59 |
| $p_3$ | 8.6 | 5.2 |
| $p_4$ | 5.93 | 7.23 |
| $p_5$ | 6.57 | 3.64 |
| $p_6$ | 4.26 | 2.43 |
| $p_7$ | 13.35 | 7.15 |
| $p_8$ | 6.6 | 3.79 |
| $p_9$ | 11.34 | 5.47 |
| $p_{10}$ | 6.63 | 2.98 |
| $p_{11}$ | 9.79 | 3.91 |
| $p_{12}$ | 14.08 | 5.05 |
| $p_{13}$ | 9.24 | 4.24 |
| $p_{14}$ | 7.23 | 4.31 |
| $p_{15}$ | 4.65 | 6.96 |

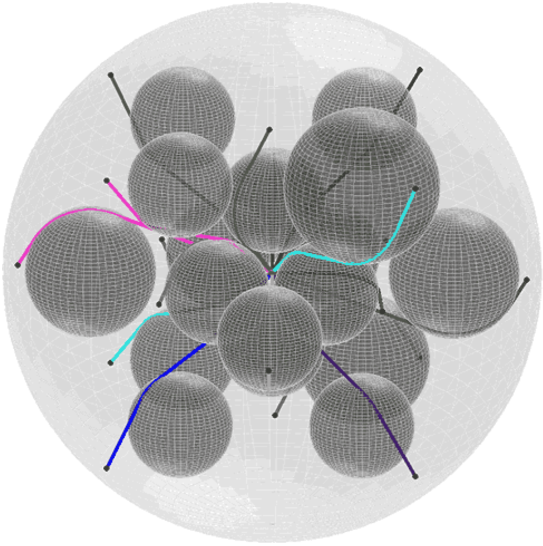


Figure 3.18: Robot safe navigation from eighteen different initial positions in a three-dimensional sphere world.

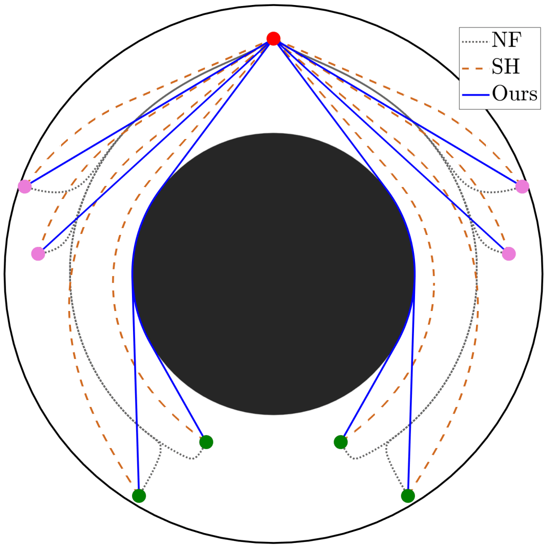


Figure 3.19: Comparison of paths generated by our approach, SH, and NF in a single two-dimensional sphere world.

We consider six different initial positions. Three are inside the nests, and the remaining three are outside but in the vicinity of their boundaries (undesired equilibria). The results of the simulation are shown in Fig. 3.20. The trajectories starting from the nests stay inside, while the three remaining trajectories reach their destination safely. We can see that nests are indeed the attraction region of the undesired equilibria.

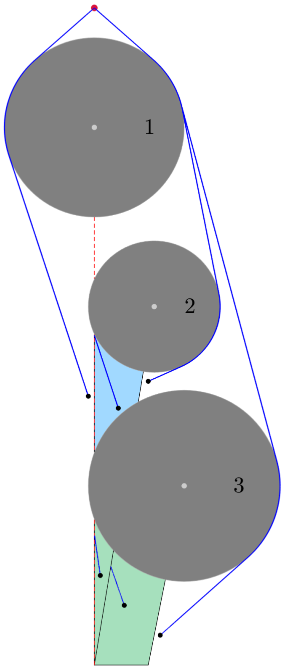


Figure 3.20: Invariance of the nests.

### 3.7.1 Sensor-based implementation in *a priori* unknown environments

#### 3.7.1.1 MATLAB simulation

We tested our sensor-based strategy (3.33) in a sphere world, as shown in Fig. 3.21, where we used a 360° LiDAR model with 1° resolution and two radial ranges $R = 2m$ and $R = 4m$. We plotted the trajectories generated by Algorithm 1 and Algorithm 2. The results in Fig. 3.21, clearly show a decrease in performance, in terms of path length, when navigating in an *a priori* unknown environment relying on a sensor. This is expected since, in the sensor-based approach, the information available about the workspace is limited to the sensor's detection zone. It is, therefore, impossible to predict *a priori* the obstacles to be avoided before the sensor detects them. In fact, under control (3.22), where global information on the environment is available, after avoiding an obstacle, the robot already knows the next obstacle to avoid, resulting in a quasi-optimal trajectory. Overall, the sensor-based approach provides a *local* optimal solution in the sense that each local avoidance maneuver is optimal when the robot is close enough to the detected obstacle. The simulation video can be found online.[5]

We also performed simulations in environments with convex obstacles satisfying the

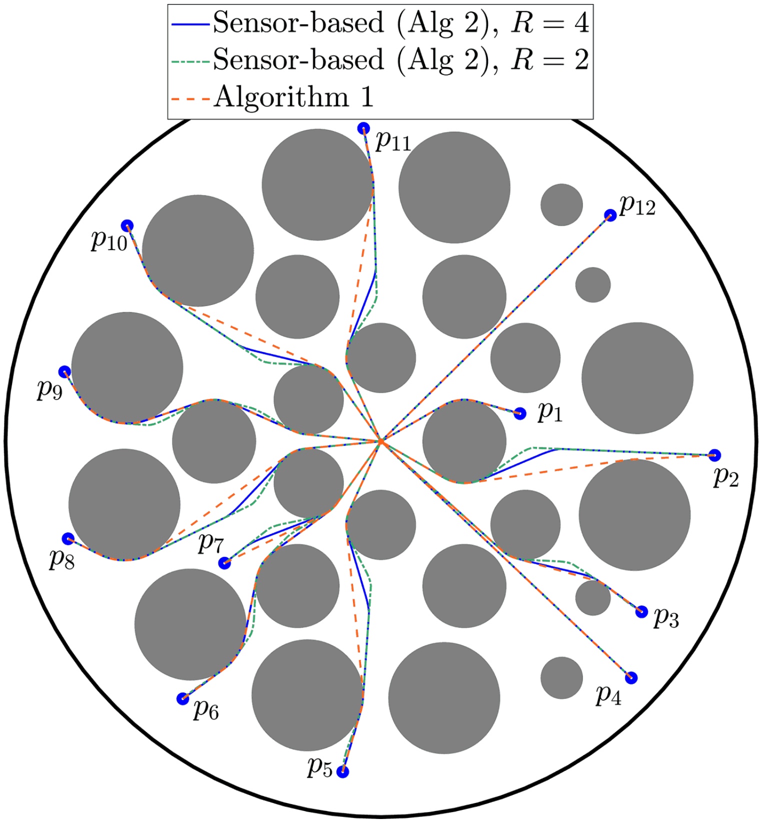


Figure 3.21: Navigation in 2D sphere world.

[5] [Online]. Available: `https://youtu.be/cnWoxi-lGvw`

Table 3.3: The relative length difference of the paths, shown in Fig. 3.21, generated by Algorithm 2, with two sensor radial ranges ($R = 2m$ and $R = 4m$), with respect to Algorithm 1.

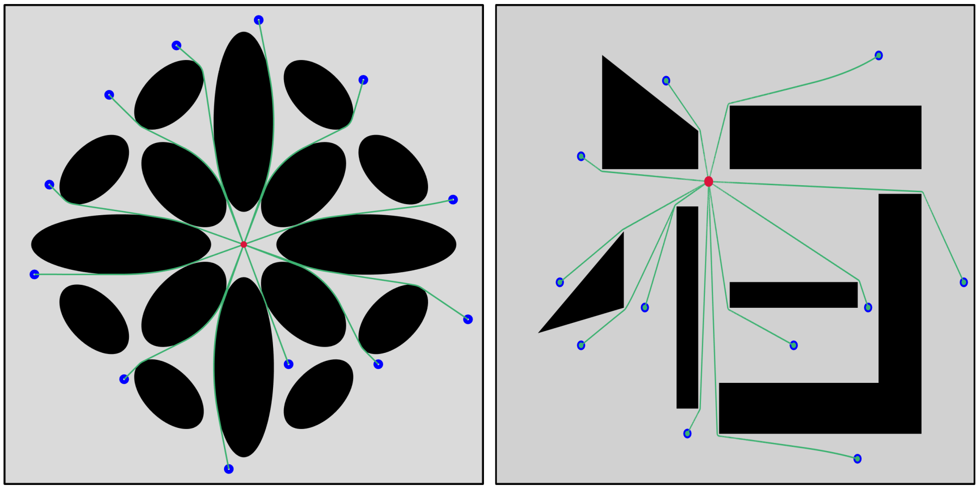
| Paths | RLD$^1$ (%) | RLD$^2$ (%) |
|---|---|---|
| $p_1$ | 0 | 0.02 |
| $p_2$ | 1.37 | 2.37 |
| $p_3$ | 0.12 | 0.8 |
| $p_4$ | 0 | 0 |
| $p_5$ | 0.72 | 1.49 |
| $p_6$ | 0 | 0.72 |
| $p_7$ | 0.43 | 1.24 |
| $p_8$ | 0.69 | 1.37 |
| $p_9$ | 0 | 0.68 |
| $p_{10}$ | 0.7 | 1.4 |
| $p_{11}$ | 0.94 | 1.84 |
| $p_{12}$ | 0 | 0 |

curvature condition in Assumption 3.4. The first simulation was performed in an environment with ellipsoidal obstacles and the second one in an environment with polygonal obstacles. The results in Fig. 3.22 show the effectiveness of the proposed approach in convex worlds with smooth and non-smooth boundaries where all the trajectories converge safely to the target (red dot). Note that the robot's navigation was successful in the environment shown in Fig. 3.22(b), although it contains an L-shaped non-convex obstacle. In fact, in view of the position of the target and during all the avoidance maneuvers, only a convex curve is detected from the boundary of this L-shaped obstacle. Simulation videos can be found online.[6]

#### 3.7.1.2 Gazebo simulation

We used the meta-operating system ROS (Noetic) running on Ubuntu (20.04.6) to implement the sensor-based control strategy (3.33) on a Turtlebot3 model and simulate real-world scenarios with Gazebo (3D dynamic simulator). Our code is written in Python, and the data analysis is performed in MATLAB. The Turtlebot model includes a 360° LiDAR with a resolution of 1°, a maximum range $R = 3.4\,m$, and a minimum range of $0.12\,m$. The robot's position and orientation are obtained by subscribing to the odometry topic provided by ROS. Zero mean Gaussian noise is added to the sensors' data where the standard deviation for the LiDAR is $0.02\,m$, the standard deviation for the position is $0.03\,m$, and for the orientation, the standard deviation is $0.035\,rd$. As the Turtlebot has a disk-shaped base of radius $r_b = 0.14\,m$, we consider the eroded workspace $\mathcal{W}_r := \mathcal{W} \setminus (\partial\mathcal{W} \oplus \mathcal{B}(0, r))$, and the dilated obstacles $\tilde{\mathcal{O}}_i^r$. The eroded free space is then

[6] [Online]. Available: `https://youtu.be/Y5dho-ptkm8` and `https://youtu.be/FZ0qxx6Gsog`



(a) Convex world with elliptical obstacles. (b) Convex world with polygonal obstacles.

Figure 3.22: Sensor-based navigation in unknown 2D convex worlds.

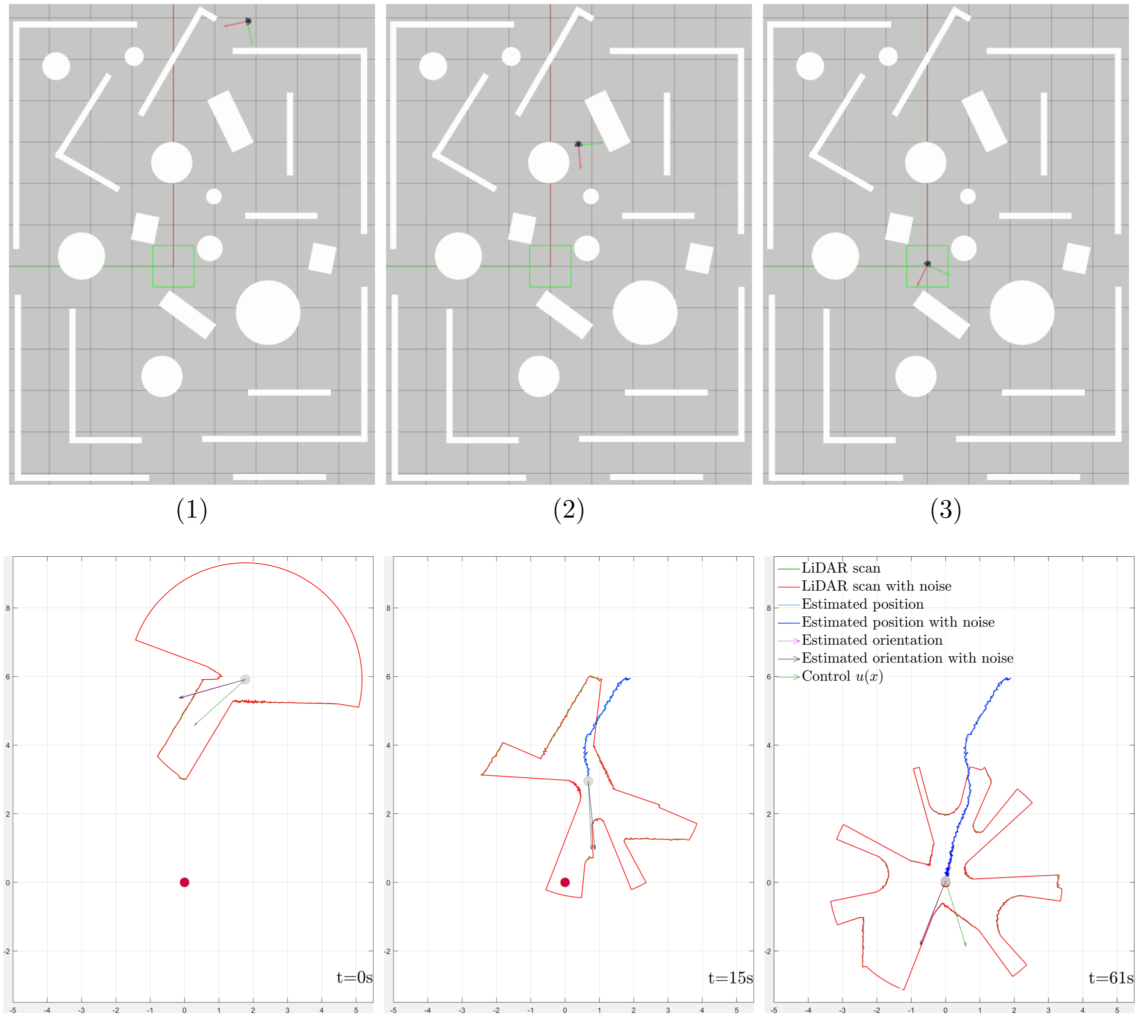
defined as $\mathcal{F}_r := \mathcal{W}_r \setminus \cup_{i\in\mathbb{I}}\tilde{\mathcal{O}}_i^r$ and for all $x \in \mathcal{F}_r$, $\mathcal{B}(x, r_b) \subset \mathcal{F}$. Considering that $x$ (the center of the robot's base) evolves in the eroded free space $\mathcal{F}_r$, and choosing the dilation parameter $r = r_b + r_s$, where $r_s = 0.11\,m$ is a safety margin, the robot is guaranteed to evolve in the free space $\mathcal{F}$. Turtlebot 3 is a differential drive robot whose kinematic model is represented by

$$\begin{cases} \dot{x} = v[\cos(\psi)\ \sin(\psi)]^\top, \\ \dot{\psi} = \omega, \end{cases} \tag{3.35}$$

where $\psi \in (-\pi, \pi]$ is the robot's orientation, and $v \in \mathbb{R}$ and $\omega \in \mathbb{R}$ are, respectively, the robot's linear and angular velocity inputs. As the control law (3.33) was designed for fully actuated robots, a transformation is required to generate adequate velocity inputs for our robot. The principal idea is to rotate the robot so that its orientation coincides with the direction of $u(x)$ obtained from (3.33), and then translate the robot with a linear velocity equal to the magnitude of $u(x)$. The direction of $u(x)$ is denoted by $\psi_d = \overline{\text{atan2}}(u(x))$, and the difference between the robot's orientation and the direction of $u(x)$ is denoted by $\Delta\psi = \psi - \psi_d \in (-\pi, \pi]$. We transform the velocity input of a fully actuated robot to the velocity inputs of a nonholonomic mobile robot with smooth switching between the rotation and translation using the following transformation (inspired from (Sawant et al., 2024b)):

$$\begin{cases} v = \min\left(v_{\max}, k_v\|u(x)\|\left(\cos(\frac{\Delta\psi}{2})\right)^{2p}\right), \\ \omega = \omega_{\max}\sin(\frac{\Delta\psi}{2}), \end{cases} \tag{3.36}$$

where $k_v > 0$, $p \geq 1$, $v_{\max} = 0.26\,m/s$ and $\omega_{\max} = 1.82\,rd/s$ are the maximum supported velocities by the robot's actuators. Larger $p$ values lead to small velocities when the robot's heading is misaligned with the direction of the control $u(x)$ (*i.e.*, $\Delta\psi \neq 0$). This minimizes the linear displacements when the robot orients its heading to match $u(x)$. This procedure allows to generate trajectories closer to the ones generated by the control $u(x)$. We created a Gazebo environment with obstacles whose dilated versions adhere to Assumption 3.4. We implemented the transformed control law (3.36) on the Turtlebot 3, where we set the gain to $k_v = 0.8$ and the exponent $p = 3$. The results are shown in Fig. 3.23, and the simulation video can be found online.[7]



(1) (2) (3)

(4) (5) (6)

Figure 3.23: Time-stamped shots of Turtlebot 3 navigating a Gazebo world.

## 3.8 Experimental validation

In this section, we use the Turtlebot 4 platform shown in Fig. 3.24(left) to implement the sensor-based control strategy (3.33) with the transformation from the fully-actuated

[7] [Online]. Available: `https://youtu.be/g1Ya9RFSgJc`

model to the differential-drive model (3.35)-(3.36).

### 3.8.1 Experimental settings

Our implementation is based on a ROS 2 (Humble) setup on Ubuntu 22.04, integrating the two main components of the TurtleBot 4, the Create 3 and the Raspberry Pi 4B (RPi4B), and a User PC (an external computer), with communication realized through a simple discovery mode. This networking mode is a multicast configuration enabling peer-to-peer communication between the various devices connected to the Wi-Fi network. The RPi4B acts as the main host of the robot's ROS 2 node and as a network gateway. It receives LiDAR data and transmits it to the network, publishing it in the /scan topic. It also relays the robot pose information supplied by Create 3 to the network, publishing it in the /odom topic, and receives the velocity commands sent by the PC user and transmits it to Create 3. The Create 3 is the mobile base equipped with essential onboard actuators and sensors such as wheel encoders and IMU for odometry and cliff sensors for safety. It is also responsible for the low-level control. The user PC receives LiDAR ranges and robot pose to execute the control algorithm and send velocity commands (linear velocity $v$ and angular velocity $\omega$) to the RPi4B, publishing them in the /vel_cmd topic. The network and data communication flow are shown in Fig. 3.25. For obstacle detection, we rely on the onboard RPLiDAR-A1MS shown in Fig. 3.24. The Lidar has a resolution of 1 deg, an angular range of 360 deg, a minimum radial range $R_{min} = 0.15m$ and a maximum radial range $R_{\max} = 12m$. The LiDAR provides measurements in its own frame (LiDAR frame), which is displaced by $-4cm$ along the $x$-axis and rotated by 90 deg with respect to the robot frame as illustrated in Fig. 3.24(right). Therefore, we first limit the measurement radial range to $R_{\max} = 2m$ (due to the workspace limitation), then transform the LiDAR data from the LiDAR frame to the robot frame so we can correctly localize the detected obstacles with respect to the robot frame. The parameters of the control (3.33) and (3.36) used in the experiment are summarized in Table 3.4.

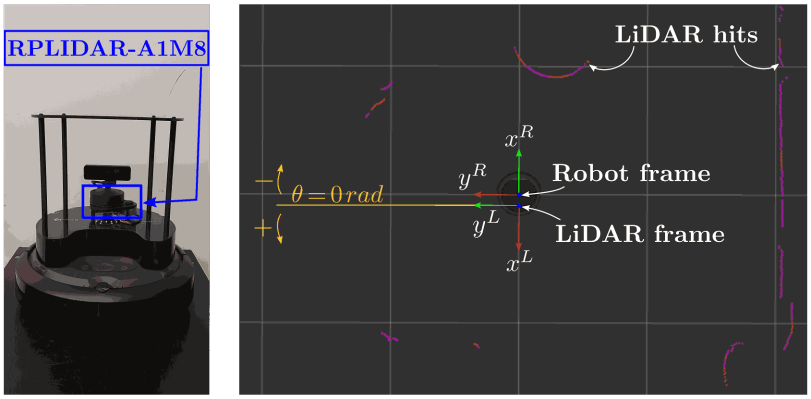


Figure 3.24: RPLIDAR-A1M8 onboard of Turtlebot 4. The figure on the left shows the Turtlebot 4 and the RPLIDAR-A1M8. The figure on the right illustrates the robot frame and the LiDAR frame.

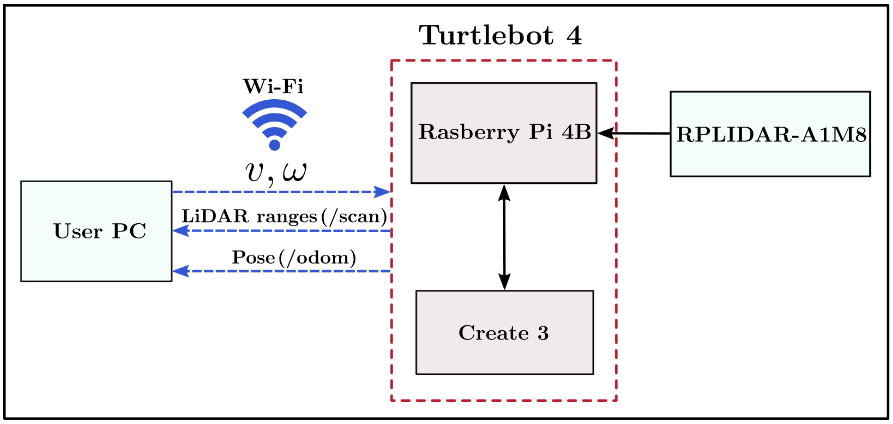


Figure 3.25: A schematic representation of the network and data communication flow in our experimental setup.

Table 3.4: Experimental setup and control parameters.

| Parameter | Value |
|---|---|
| Robot's radius | $r_b = 0.17m$ |
| Safety margin | $r_s = 0.13m$ |
| Dilation parameter | $r = r_b + r_s = 0.3m$ |
| Maximum radial range measurement | $R_{max} = 2m$ |
| Gain of the control $u(x)$ | $\gamma = 1.5$ |
| Gain used in (3.36) | $k_v = 0.1$ |
| Tuning parameter used in (3.36) | $p = 1$ |
| Maximum linear velocity | $v_{\max} = 0.31m/s$ |
| Maximum angular velocity | $\omega_{\max} = 1.9rd/s$ |

### 3.8.2 Experimental results

We set up a $6m \times 4.25m$ workspace with four boxes and four punching bags as obstacles. The robot is initially at the origin with its heading aligned with the $x$-axis ($\psi = 0$), and the target is set at the position $x_d = [5.3\ 0]^\top$. The experimental results are shown in Fig. 3.26 and in a video that can be found online.[8] The top figure of Fig. 3.26 shows the workspace configuration with the initial and final positions. In the bottom figure, the trajectory of the robot is plotted in an orthographic projection top view of the workspace. The obtained results illustrate the safe navigation of the robot from the initial position to the final destination.

[8] [Online]. Available: `https://youtu.be/Z2AWva6DYgs`

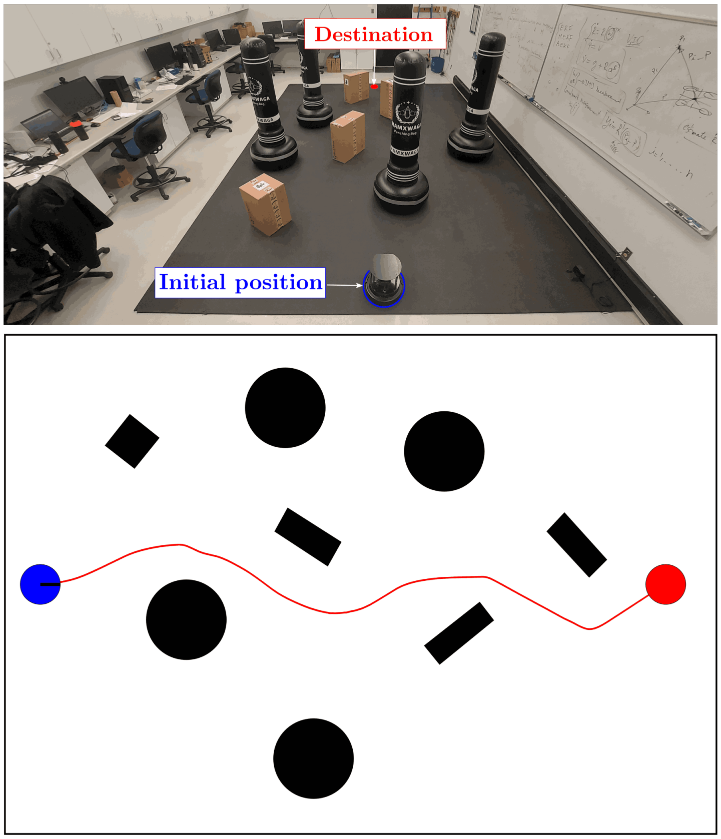


Figure 3.26: Safe navigation of Turtlebot 4 in an environment with four boxes and four punching bags as obstacles. The top figure shows the configuration of the workspace with a highlighted initial and final positions. The bottom figure is a plot of the robot's trajectory in an orthographic projection top view of the workspace.

# Chapter 4

# Hybrid Feedback Control for Global Navigation with Locally Optimal Obstacle Avoidance in $n$-Dimensional Spaces

## 4.1 Introduction

This chapter deals with the problem of autonomous navigation in $n$-dimensional Euclidean spaces cluttered with a finite number of spherical obstacles. We propose a hybrid navigation scheme, with locally optimal obstacle avoidance, leading to global asymptotic stability of the target location. A reactive sensor-based version of the proposed approach is also presented for autonomous navigation in *a priori* unknown environments.

The approach proposed in (Berkane et al., 2022) is an example of a hybrid feedback control that guarantees GAS of a target location in $n$-dimensional Euclidean spaces with elliptical obstacles. However, this approach does not generate length-optimal trajectories, does not generate continuous control inputs and does not extend easily to *a priori* unknown environments. Another example is the sensor-based hybrid feedback navigation strategy with GAS guarantees and continuous vector field generation in two-dimensional environments with convex obstacles (Sawant et al., 2023). Unfortunately, path-length optimality is not considered in this approach, and in its recent extensions to non-convex obstacles (Sawant et al., 2024a) and to three-dimensional convex obstacles (Sawant et al., 2024c). The proposed hybrid feedback-based approach proposed in this chapter is also distinguished from continuous feedback-based approaches by the fact that the latter ones achieve, at best, AGAS results (Koditchek and Rimon, 1990) and generate non-optimal paths in terms of length. One can take the example of the NF approach (Koditchek and Rimon, 1990) with its various extensions (Rimon and Koditchek, 1992; Filippidis and Kyriakopoulos, 2013; Loizou and Kyriakopoulos, 2006; Dimarogonas et al., 2006) and the QP-based control barrier functions (Reis et al., 2021; Tan and Dimarogonas, 2024) for *a priori* known environments, or the SH approach (Arslan and Koditschek, 2016) and its different extensions (Arslan and Koditschek, 2019; Vasilopoulos and Koditschek, 2018;

Vasilopoulos et al., 2020; Vasilopoulos et al., 2021) for *a priori* unknown environments.

The main contributions of the proposed approach in this chapter are summarized as follows:

- **Safety and global asymptotic stability**: The proposed control strategy ensures safe navigation in $n$-dimensional spaces with spherical obstacles while providing global asymptotic stability (GAS) guarantees for the target location, *i.e.,* convergence from all initial conditions in the workspace.
- **Continuous control input**: Unlike many hybrid strategies, the proposed hybrid feedback controller produces continuous velocity inputs, enabling smooth robot motion and ensuring practical feasibility for real-world applications.
- **Local optimal avoidance maneuvers**: By dynamically generating shortest-path maneuvers around obstacles, the controller achieves locally optimal navigation without sacrificing global stability guarantees.
- **Navigation in unknown environments**: The proposed obstacle-avoidance mechanism is fully implementable using range sensors alone, allowing navigation in both 2D and 3D environments without requiring prior global knowledge. Experimental validation on a TurtleBot 4 platform demonstrates its effectiveness in *a priori* unknown settings, as illustrated in Fig. 4.14.
- **Scalability and computational efficiency**: The control strategy is computationally lightweight and scalable to higher-dimensional spaces, making it well-suited for autonomous systems with limited onboard resources.

Compared to the quasi-optimal approach in Chapter 3, this approach ensures safe navigation from any initial position in the obstacle-free space. It does not suffer from the problem of existence of undesired equilibria regardless of the dimension of the workspace or whether it is *a priori* known or unknown.
The results presented in this chapter are based on our conference paper (Cheniouni et al., 2024a) and a submitted journal paper (Cheniouni et al., 2024b).

## 4.2 Problem formulation

Autonomous navigation in cluttered environments remains a fundamental challenge in robotics, particularly when requiring safety, efficiency, and smooth motion in high-dimensional spaces. We consider the position $x \in \mathbb{R}^n$ of a point mass vehicle evolving in the Euclidean space of dimension $n \geq 2$ punctured by $b \in \mathbb{N} \setminus \{0\}$ balls $\mathcal{O}_k := \mathcal{B}(c_k, r_k)$ of radius $r_k > 0$ and center $c_k \in \mathbb{R}^n$ where $k \in \mathbb{I} := \{1, \ldots, b\}$. The obstacle-free space is, therefore, given by the closed set

$$\mathcal{X} := \mathbb{R}^n \setminus \bigcup_{k \in \mathbb{I}} \mathring{\mathcal{O}}_k. \tag{4.1}$$

In practical applications, obstacle disjointness is a standard requirement to avoid overlapping regions, which could lead to navigation ambiguities or infeasible maneuvers. Therefore, we impose the following assumption to preserve the spherical nature of obstacles:

**Assumption 4.1** *The obstacles are pairwise disjoint, that is,*

$$\|c_k - c_j\| > r_k + r_j, \ \ \forall k, j \in \mathbb{I}, \, i \neq j. \tag{4.2}$$

We consider a velocity-controlled vehicle such that $\dot{x} = u$, where $u$ represents the control policy designed to generate trajectories in constrained $n-$dimensional spaces. This model assumes full actuation, which is a common abstraction for theoretical exploration, though practical implementations may incorporate actuation constraints. The primary objective is to design a feedback control policy for $u$ that ensures safe and efficient navigation while addressing the following challenges:

- **Global Goal Reaching**: Steer the vehicle from any initial position $x(0) \in \mathcal{X}$ to a predefined destination $x_d \in \mathring{\mathcal{X}}$, ensuring the robot consistently reaches the goal regardless of the initial conditions.

- **Obstacle Avoidance**: Prevent collisions with all obstacles $\mathcal{O}_k$, leveraging the obstacle-free space $\mathcal{X}$ to navigate safely.

- **Locally Optimal Maneuvers**: Achieve locally shortest feasible paths around obstacles, reflecting efficiency in navigation and minimizing unnecessary detours.

- **Smoothness**: At each time $t$, the control input $u(t)$ is continuous, leading to continuously differentiable trajectories suitable for practical deployment.

The proposed problem addresses a critical gap in existing navigation frameworks. Many approaches either rely on constructing explicit or local representations of the environment, which may not be feasible in real-time, or fail to guarantee smooth, optimal, and globally converging trajectories in $n$-dimensional spaces. By focusing on disjoint spherical obstacles, we establish a mathematically tractable yet practically relevant scenario that enables rigorous analysis of the control strategy. The emphasis on continuous control inputs ensures compatibility with robotic systems where abrupt changes can destabilize dynamics or degrade performance. This problem formulation sets the stage for developing a novel hybrid feedback strategy that overcomes these limitations, providing a robust solution for autonomous navigation in high-dimensional obstacle-filled environments.

### 4.2.1 Sets definition

In this subsection, we define the subsets of the free space that will be used in the design of our control proposed in Section 4.3. These subsets are illustrated in Fig. 4.1 and presented as follows:

- The *shadow region* of obstacle $\mathcal{O}_k$ is the area hidden by obstacle $\mathcal{O}_k$ from which the vehicle does not have a clear line of sight to the target. It is defined as follows:

$$\mathcal{S}_k(x_d) := \big\{q \in \mathcal{C}^{\leq}_{\mathcal{X}}(x_d, c_k - x_d, \theta(x_d, k)) | (c_k - q)^\top (x_d - q) \geq 0\big\}, \tag{4.3}$$

  where the function $\theta(q, k) \colon \mathcal{X} \to (0, \frac{\pi}{2}]$, $q \mapsto \theta(q, k) := \arcsin(r_k / \|q - c_k\|)$ assigns to each position $q$ of the free space, the half aperture of the cone enclosing obstacle $\mathcal{O}_k$.

- The *active region* of obstacle $\mathcal{O}_k$ is defined as follows:

$$\mathcal{A}_k(x_d) := \mathcal{S}_k(x_d) \cap \mathcal{B}(c_k, r_k + \bar{r}_k), \tag{4.4}$$

  where $\bar{r}_k \in (0, \hat{r}_k)$, $\hat{r}_k = \min\limits_{j \in \mathcal{I}_k(x_d)} (\|c_k - c_j\| - r_k - r_j)$, and $\mathcal{I}_k(x_d) := \{j \in \mathbb{I} | \mathcal{S}_k(x_d) \cap \partial\mathcal{O}_j \neq \varnothing\}$ is the set of obstacles hidden (fully or partially) from the destination $x_d$ by obstacle $\mathcal{O}_k$. Note that when $\mathcal{I}_k(x_d) = \varnothing$, $\hat{r}_k = \infty$ and $\mathcal{A}_k(x_d) = \mathcal{S}_k(x_d)$.

- The *exit set* of obstacle $\mathcal{O}_k$ is the lateral surface of the *active region* and is defined as follows:

$$\mathcal{E}_k(x_d) := \mathcal{C}^{=}_{\mathcal{X}}(x_d, c_k - x_d, \theta(x_d, k)) \cap \mathcal{A}_k(x_d). \tag{4.5}$$

- The *hat* of obstacle $\mathcal{O}_k$ is the upper part of the surface of the cone of vertex $x_d$ enclosing obstacle $\mathcal{O}_k$ and is defined as follows:

$$\mathcal{H}_k(x_d) := \mathcal{C}^{=}_{\mathcal{X}}(x_d, c_k - x_d, \theta(x_d, k)) \setminus \mathcal{E}_k(x_d). \tag{4.6}$$

- The *active free space* is defined as

$$\mathcal{V}(x_d) := \bigcup_{k \in \mathbb{I}} \mathcal{A}_k(x_d). \tag{4.7}$$

## 4.3 Main results

In this section, we present the design of our hybrid controller, demonstrating the forward invariance of the obstacle-free space and the stability of the target location under the proposed control scheme. Furthermore, we establish the continuity of the control input and substantiate the optimality of the obstacle-avoidance maneuvers.

### 4.3.1 Control design

The proposed hybrid control strategy consists of two operation modes: the *motion-to-destination* mode and the *obstacle-avoidance* mode. We make use of a mode selector $m \in \mathbb{M} := \{-1, 0, 1\}$ which refers to the *motion-to-destination* mode if $m = 0$, and to the *obstacle-avoidance* mode if $m = \pm 1$. Each mode of operation is activated and deactivated

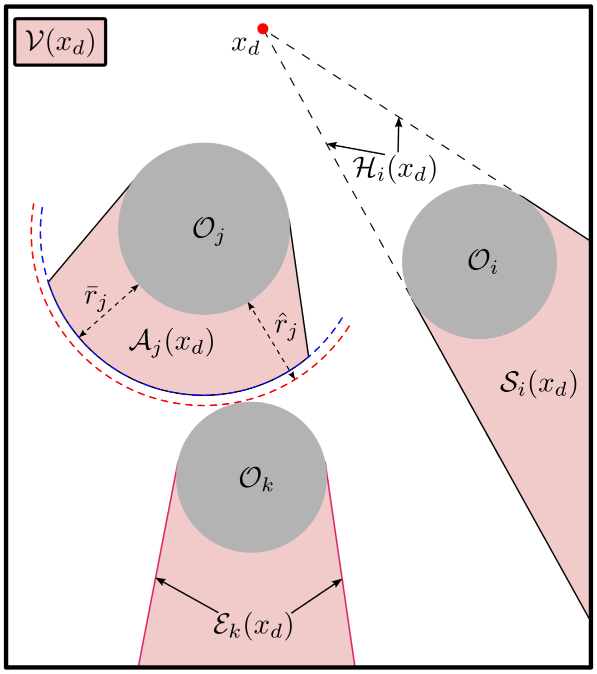


Figure 4.1: 2D representation of the sets in Section 4.2.1.

in specific regions–referred to as flow sets and jump sets–based on a carefully designed switching strategy. The proposed hybrid feedback control depends on the current position $x \in \mathbb{R}^n$, the considered obstacle $k \in \mathbb{I}$, and the operation mode $m$, and is defined as follows:

$$u(x,k,m) = m^2\alpha(x,k)\mu(x,k,m)\kappa(x,k,m) + (1 - m^2\alpha(x,k))u_d(x), \tag{4.8a}$$

$$\begin{cases} \dot{k} = 0, \\ \dot{m} = 0, \end{cases} \quad (x,k,m) \in \mathcal{F}, \tag{4.8b}$$

$$\begin{cases} k^+ \in K(x,k,m), \\ m^+ \in M(x,k,m) \end{cases} \quad (x,k,m) \in \mathcal{J}, \tag{4.8c}$$

where $u_d(x) := \gamma(x_d - x)$ is the nominal control for the *motion-to-destination* mode, $\gamma > 0$, $\kappa(x,k,m)$ is the control for the *obstacle-avoidance* mode that will be defined in Subsection 4.3.1.2. The scalar functions $\alpha$ and $\mu$, which will be defined in Subsection 4.3.1.3, ensure smooth transitions between the operation modes. The dynamics of the obstacle and mode selectors are described in (4.8b)-(4.8c) where $K(\cdot)$ is the jump map of the obstacle selector and $M(\cdot)$ is the jump map of the mode selector that will be designed in Subsection 4.3.1.3. The sets

$$\mathcal{F} := \bigcup_{m\in\mathbb{M}} (\mathcal{F}_m \times \{m\}), \ \mathcal{J} := \bigcup_{m\in\mathbb{M}} (\mathcal{J}_m \times \{m\}) \tag{4.9}$$

are, respectively, the flow and jump sets of the hybrid system where $\mathcal{F}_m$ and $\mathcal{J}_m$ are, respectively, the flow and jump sets of the operation mode $m \in \mathbb{M}$ that will be constructed in Subsections 4.3.1.1, 4.3.1.2. In the following, we define the control in each mode and its associated flow and jump sets, and then we define the jump maps that govern the transitions between these modes.

#### 4.3.1.1 Motion-to-destination mode ($m = 0$)

In this mode, the robot moves straight to the target under the nominal control $u_d(x)$. Considering obstacle $k \in \mathbb{I}$, the flow and jump sets associated with obstacle $\mathcal{O}_k$, depicted in Fig. 4.2, are defined as follows:

$$\mathcal{F}_k^0 := \overline{\mathcal{X} \setminus \mathcal{A}_k(x_d)},\ \mathcal{J}_k^0 := \mathcal{A}_k(x_d). \tag{4.10}$$

where $\mathcal{A}(x_d)$ is the active region defined in (4.4). The flow and jump sets for the mode $m = 0$, considering all obstacles, are defined as:

$$\mathcal{F}_0 := \tilde{\mathcal{F}}_0 \times \mathbb{I}, \quad \mathcal{J}_0 := \tilde{\mathcal{J}}_0 \times \mathbb{I}, \tag{4.11}$$

where the *motion-to-destination* mode is selected at each position $x$ within the intersection of the flow sets, $\tilde{\mathcal{F}}_0 := \cap_{k\in\mathbb{I}}\mathcal{F}_k^0$, for all obstacle indices $k \in \mathbb{I}$. Additionally, at each position $x$ in the union of the jump sets $\tilde{\mathcal{J}}_0 := \cup_{k\in\mathbb{I}}\mathcal{J}_k^0 = \mathcal{V}(x_d)$, where $\mathcal{V}(x_d)$ is the *active free space* defined in (4.7), a jump to the *obstacle-avoidance* mode can occur for any obstacle index $k \in \mathbb{I}$.

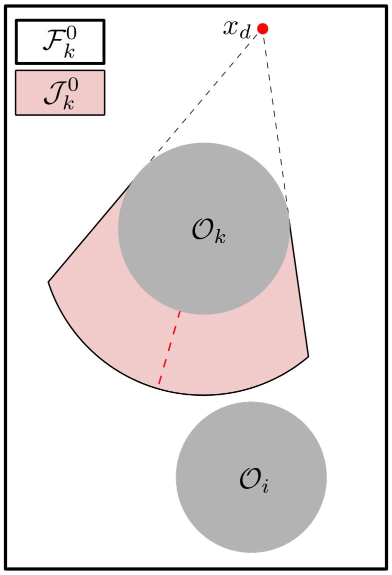


Figure 4.2: 2D illustration of the flow and jump sets for the *motion-to-destination* mode associated with obstacle $\mathcal{O}_k,\ k \in \mathbb{I}$.

#### 4.3.1.2 Obstacle-avoidance mode ($m = \pm 1$)

In this mode of operation, the robot will engage in a local optimal obstacle avoidance maneuver. To this end, we consider two virtual destinations, $x_k^1$ and $x_k^{-1}$, which are designed to be on the *hat* $\mathcal{H}_k(x_d)$, defined in (4.6), of obstacle $\mathcal{O}_k$ and symmetrical with respect to the hat axis $(c_k - x_d)$, as shown in Fig. 4.3. The introduction of two virtual destinations is motivated by the observation that each virtual destination generates a distinct set of undesired equilibria. By appropriately selecting the virtual destination during the avoidance maneuver (via hybrid feedback), the vehicle is prevented from becoming trapped at these undesired equilibria, see (Berkane et al., 2022). Moreover, by projecting the nominal control $\bar{\kappa}(x, k, m) := \gamma(x_k^m - x)$ onto the surface of the cone, with vertex at $x$, enclosing obstacle $\mathcal{O}_k$, we aim to avoid the obstacle with minimal deviation from the nominal direction, as illustrated in Fig. 4.3, thereby generating optimal obstacle maneuvers, see also (Cheniouni et al., 2023a). These virtual destinations are chosen as follows:

$$x_k^1 \in \mathcal{H}_k(x_d) \cap \mathcal{P}_{\geq}(p_k, x_d - p_k), \tag{4.12a}$$

$$x_k^{-1} = x_d - \pi_r\left(\frac{c_k - x_d}{\|c_k - x_d\|}\right)(x_k^1 - x_d) \tag{4.12b}$$

where $p_k := c_k + r_k \frac{x_d - c_k}{\|x_d - c_k\|}$ and $\|x_d - x_k^m\| =: e_k > 0$. Note that the choice of $x_k^1$ is not unique and can be any point on the hat $\mathcal{H}_k(x_d)$ of the enclosing cone within the half-space $\mathcal{P}_{\geq}(p_k, x_d - p_k)$, and satisfying $\|x_d - x_k^m\| = e_k$. The term $\kappa(x, k, m)$ used in (4.8a), which is the control in the *obstacle-avoidance mode* (*i.e.*, $m = \pm 1$), is given as follows:

$$\kappa(x, k, m) = \bar{\kappa}(x, k, m) - \tau(x, k, m)\frac{c_k - x}{\|c_k - x\|}, \tag{4.13}$$

where $\tau(x, k, m) = \|\bar{\kappa}(x, k, m)\| \sin(\theta(x, k) - \beta(x, k, m)) \sin^{-1}(\theta(x, k))$, $\beta(x, k, m) = \angle(c_k - x, \bar{\kappa}(x, k, m))$, and $\theta(x, k) = \arcsin(r_k/\|x - c_k\|) \in (0, \pi/2]$ with $k \in \mathbb{I}$ and $m \in \{-1, 1\}$. The control law (4.13) is a scaled parallel projection of the nominal controller $\bar{\kappa}(x, k, m)$, with respect to the virtual destination $x_k^m$, onto the line tangent to obstacle $\mathcal{O}_k$, ensuring a minimal angle with $\bar{\kappa}(x, k, m)$ and the continuity of $\kappa(x, k, m)$ at the *exit set*. The optimization problem with solution $\kappa(x, k, m)$ is given in the following lemma.

**Lemma 4.1** *Consider obstacle $\mathcal{O}_k$, a virtual destination $x_k^m$ and their associated active region $\mathcal{A}_k(x_k^m)$ where $k \in \mathbb{I}$ and $m \in \{-1, 1\}$. For each $(x, k, m)$, the control law $\kappa(x, k, m)$, given in (4.13), is the unique solution of the optimization problem given by*

$$\min_u \angle(x_k^m - x, v), \tag{4.14a}$$

$$s.t.\ v \in \mathcal{V}(c_k - x, \theta(x, k)), \qquad \text{if } x \in \mathcal{S}_k(x_k^m), \tag{4.14b}$$

$$v = \bar{\kappa}(x, k, m), \qquad \text{if } x \in \mathcal{E}_k(x_k^m). \tag{4.14c}$$

*where $\mathcal{S}_k(x_k^m)$ and $\mathcal{E}_k(x_k^m)$ are defined in (4.3) and (4.5) respectively.*

The proof of Lemma 4.1 is the same as the proof in Appendix A.1 when the virtual destination $x_k^m$ is considered instead of the actual destination $x_d$. Note that the constraint

(4.14c) ensures the uniqueness of the solution to the optimization problem (4.14) and the continuity of $\kappa(x,k,m)$ at the *exit set* $\mathcal{E}_k(x_k^m)$. The control law (4.13) steers the robot tangentially to the obstacle, providing a more efficient path compared to approaches like the bug algorithms, which initiate obstacle avoidance only upon contact or in close proximity to the obstacle, as seen in hybrid approaches such as (Berkane et al., 2022; Sawant et al., 2023; Sawant et al., 2024a). The equilibrium points generated by the control input (4.13) in the *obstacle-avoidance* mode can be obtained by setting $\kappa(x,k,m) = 0$ for $x \in \mathcal{A}_k(x_k^m)$, $m \in \{-1, 1\}$, and $k \in \mathbb{I}$. It is clear that $\kappa(x,k,m) = 0$ holds only if $\beta(x,k,m) = 0$. The set of equilibria is then the line passing through the center $c_k$ and the virtual destination $x_k^m$ limited by the active region $\mathcal{A}_k(x_k^m)$. It is given by $\mathcal{L}_k(x_d^m) := \mathcal{L}(x_k^m, c_k) \cap \mathcal{A}_k(x_k^m)$. The flow and jump sets of the *obstacle-avoidance* mode, considering obstacle $\mathcal{O}_k$, $k \in \mathbb{I}$, are illustrated in Fig. 4.4 and defined as follows:

$$\mathcal{F}_k^m := \mathcal{A}_k(x_k^m) \setminus \mathcal{C}_{\mathcal{X}}^{<}(c_k, v_k^m, \varphi_k^m), \mathcal{J}_k^m := \overline{\mathcal{X} \setminus \mathcal{F}_k^m}, \tag{4.15}$$

where $v_k^m := c_k - x_k^m$. To ensure that the jump from the *motion-to-target* mode to the *obstacle-avoidance* mode is valid everywhere in $\mathcal{J}_0$ (*i.e.*, $\forall (x,k) \in \mathcal{J}_0$, $M(x,k,0) \neq \varnothing$), we design the angles $\varphi_k^1, \varphi_k^{-1}$ in (4.15) as per Lemma 2.1 as follows:

$$\varphi_k^1 = \varphi_k^{-1} = \varphi < \min\left\{\frac{\angle(v_k^1, v_k^{-1})}{2}, \frac{\pi - \angle(v_k^1, v_k^{-1})}{2}\right\}. \tag{4.16}$$

Note that a conic subset is subtracted from the active region for the modes $m = \pm 1$ in (4.15), which excludes the equilibria from the flow set, leaving the system with a unique equilibrium point at the target location for the mode $m = 0$. The flow and jump sets for mode $m \in \{-1, 1\}$ and considering all the obstacles are defined as follows:

$$\mathcal{F}_m := \bigcup_{k \in \mathbb{I}} \left(\mathcal{F}_k^m \times \{k\}\right), \ \mathcal{J}_m := \bigcup_{k \in \mathbb{I}} \left(\mathcal{J}_k^m \times \{k\}\right). \tag{4.17}$$

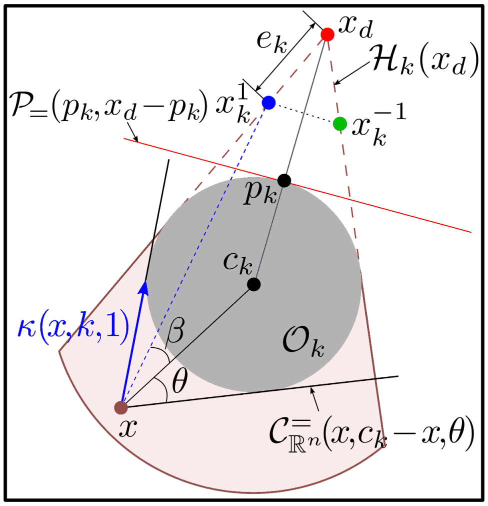


Figure 4.3: Construction of the control in the *obstacle-avoidance* mode for a 2D case.

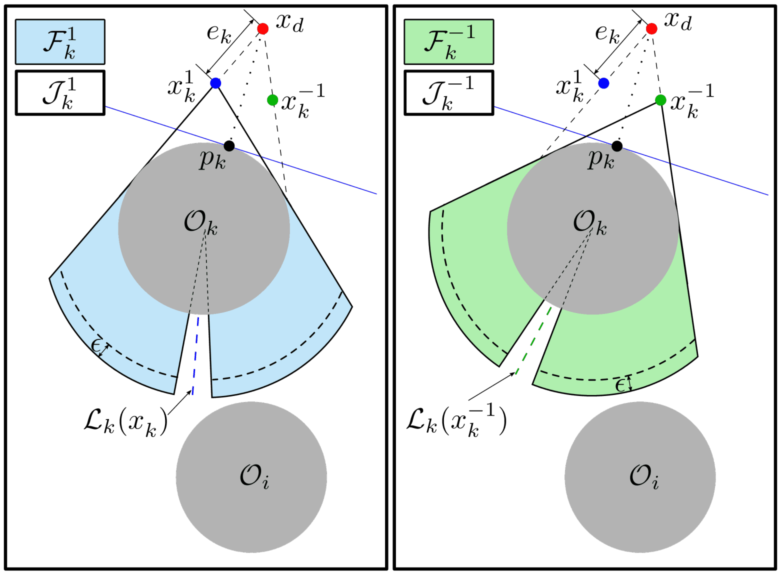


Figure 4.4: 2D illustration of the flow and jump sets for the *obstacle-avoidance* mode associated with obstacle $\mathcal{O}_k,\ k \in \mathbb{I}$.

#### 4.3.1.3 Operation mode switching scheme

The jump maps are designed to effectively switch between the operation modes to attain the following objectives:

- Avoid every encountered obstacle through local optimal maneuvers.
- Avoid obstacles one by one to ensure safety and offer the possibility of sensor-based implementation.
- Converge to the destination $x_d$ from any initial position in the obstacle-free space.

We first define the smoothing functions as follows:

$$\mu(x,k,m) := \left(1 + \frac{e_k}{\|x - x_k^m\|}\frac{\beta(x,k,m)}{\theta(x,k)}\right), \tag{4.18}$$

$$\alpha(x,k) := \begin{cases} 1 & d(x,\mathcal{O}_k) < \bar{r}_k - \epsilon, \\ \frac{\bar{r}_k - d(x,\mathcal{O}_k)}{\epsilon} & \bar{r}_k - \epsilon \leq d(x,\mathcal{O}_k) \leq \bar{r}_k, \\ 0 & d(x,\mathcal{O}_k) > \bar{r}_k, \end{cases} \tag{4.19}$$

with $0 < \epsilon \leq \bar{\epsilon}$ and $\bar{\epsilon} := \min_{k \in \mathbb{I}} \bar{r}_k$. In fact, the scalar function $\alpha(x,k)$ ensures a smooth transition from the *motion-to-destination* mode to the *obstacle-avoidance* mode. The scalar function $\mu(x,k,m)$, together with the modified switching scheme of the mode $m$ that will be designed in Subsection 4.3.3, ensures a smooth transition from the *obstacle-avoidance* mode to the *motion-to-destination* mode.
Now, we define the jump map $K(\cdot)$ of the obstacle selector as

$$K(x,k,m) := \begin{cases} k & x \in \mathcal{J}_k^m,\ m \in \{-1,1\}, \\ k' & x \in \mathcal{J}_0^{k'},\ m = 0, \end{cases} \tag{4.20}$$

and the jump map $M(\cdot)$ of the mode selector as

$$M(x,k,m) := \begin{cases} 0 & x \in \mathcal{J}_k^m,\ m \in \{-1,1\}, \\ B(x,k) & x \in \mathcal{J}_0^k,\ m = 0, \end{cases} \tag{4.21}$$

where $B(\cdot)$ is defined as:

$$B(x,k) := \begin{cases} 1 & x \in \mathcal{C}_k^1, \\ -1 & x \in \mathcal{C}_k^{-1} \\ \{-1,1\} & x \in \mathcal{C}_k, \end{cases} \tag{4.22}$$

with $\mathcal{C}_k^1 = \mathcal{C}^{\geq}_{\mathbb{R}^n}(c, v_k^1, \varphi_k^1) \setminus \mathcal{C}^{\geq}_{\mathbb{R}^n}(c, v_k^{-1}, \varphi_k^{-1})$, $\mathcal{C}_k^{-1} = \mathcal{C}^{\geq}_{\mathbb{R}^n}(c, v_k^{-1}, \varphi_k^{-1}) \setminus \mathcal{C}^{\geq}_{\mathbb{R}^n}(c, v_k^1, \varphi_k^1)$, and $\mathcal{C}_k = \mathcal{C}^{\geq}_{\mathbb{R}^n}(c, v_k^1, \varphi_k^1) \cap \mathcal{C}^{\geq}_{\mathbb{R}^n}(c, v_k^{-1}, \varphi_k^{-1})$. Note that the construction of the flow and jump sets (4.15) of the *obstacle-avoidance* mode are such that the set of undesired equilibria belongs to the jump set of the corresponding *obstacle-avoidance* mode ($m = \pm 1$).

### 4.3.2 Safety and stability analysis

In this subsection, we establish the safety and stability properties of our hybrid closed-loop system. To this end, we define the augmented state vector as

$$\xi := (x,k,m) \in \mathbb{R}^n \times \mathbb{I} \times \mathbb{M}, \tag{4.23}$$

and the overall flow and jump maps as

$$\xi \mapsto F(\xi) := (u(\xi), 0, 0), \tag{4.24}$$

$$\xi \mapsto J(\xi) := (x, K(\xi), M(\xi)). \tag{4.25}$$

Then, the resulting hybrid closed-loop system can be written as

$$\begin{cases} \dot{\xi} = F(\xi) & \xi \in \mathcal{F}, \\ \xi^+ \in J(\xi) & \xi \in \mathcal{J}, \end{cases} \tag{4.26}$$

and its representation with the hybrid data is given by $\mathcal{H} := (\mathcal{F}, \mathrm{F}, \mathcal{J}, \mathrm{J})$. To analyze our closed-loop system, we first establish its well-posedness by showing that it complies with the hybrid basic conditions (Goebel et al., 2012, Assumption 6.5), as shown in the next lemma.

**Lemma 4.2** *The hybrid closed-loop system (4.26) represented by its data $\mathcal{H}$, satisfies the following hybrid basic conditions:*

*i) The flow set $\mathcal{F}$ and the jump set $\mathcal{J}$, defined in (4.17), are closed subsets of $\mathcal{K}$.*

*ii) The flow map $\mathrm{F}$, defined in (4.24), is outer semicontinuous and locally bounded relative to $\mathcal{F}$, $\mathcal{F} \subset dom(F)$, and $\mathrm{F}(\xi)$ is convex for every $\xi \in \mathcal{F}$.*

*iii) The jump map $\mathrm{J}$, defined in (4.25), is outer semicontinuous and locally bounded relative to $\mathcal{J}$, and $\mathcal{J} \subset dom(j)$.*

**Proof** See Appendix B.1.

Now, let us define the augmented free space and the desired equilibrium set as follows:

$$\mathcal{K} := \mathcal{X} \times \mathbb{I} \times \mathbb{M}, \ \mathcal{A} := \{x_d\} \times \mathbb{I} \times \mathbb{M}. \tag{4.27}$$

Robot navigation is said to be safe if the state $x$ evolves in the obstacle-free space $\mathcal{X}$ at all times. For the hybrid closed-loop system (4.26), this is equivalent to showing forward invariance (Chai and Sanfelice, 2019) of the augmented obstacle-free space $\mathcal{K}$. The following theorem will present our results concerning safe and global navigation under the proposed hybrid controller.

**Theorem 4.1** *Consider the augmented state space $\mathcal{K}$, described in (4.27), and the hybrid closed-loop system (4.26). Then, the following statements hold:*

*i) The augmented state space $\mathcal{K}$ is forward invariant.*

*ii) The set $\mathcal{A}$ is globally asymptotically stable.*

**Proof** See Appendix B.2.

Theorem 4.1 states that the robot with position $x$ reaches the destination $x_d$ safely from any initial condition of the obstacle-free space $\mathcal{X}$. In addition to the safety and GAS of the target location, our control strategy ensures the continuity of the control input (4.8a) and optimality of the local obstacle avoidance maneuvers as demonstrated in the next subsection.

### 4.3.3 Continuity and optimality

To ensure continuity of the control input and optimality of the avoidance maneuvers when implementing the hybrid control (4.8), two properties of the proposed hybrid system will be utilized. In fact, the closed-loop hybrid system defined in (4.26) offers a flexibility in choosing the virtual destinations (see (4.12a)) as well as the operation mode $m$ (set-valued map in (4.22)). In this section, we show that, for a given obstacle $k \in \mathbb{I}$, the virtual destinations in (4.12) can be selected to guarantee a two-dimensional motion. Moreover, when the robot is in the hysteresis region (*i.e.*, $x \in \mathcal{J}_k^0 \cap \mathcal{C}_k$), the mode $m$ can be forced in (4.22) to the value that ensures the virtual destination $x_k^m$ is the closest to the robot. In the following, we present the detailed design process that guarantees continuity of the control input and local optimality of the obstacle avoidance maneuver.
The proposed control strategy generates planar trajectories during the time interval in which a given obstacle $\mathcal{O}_k$ is being avoided, as stated in the following lemma:

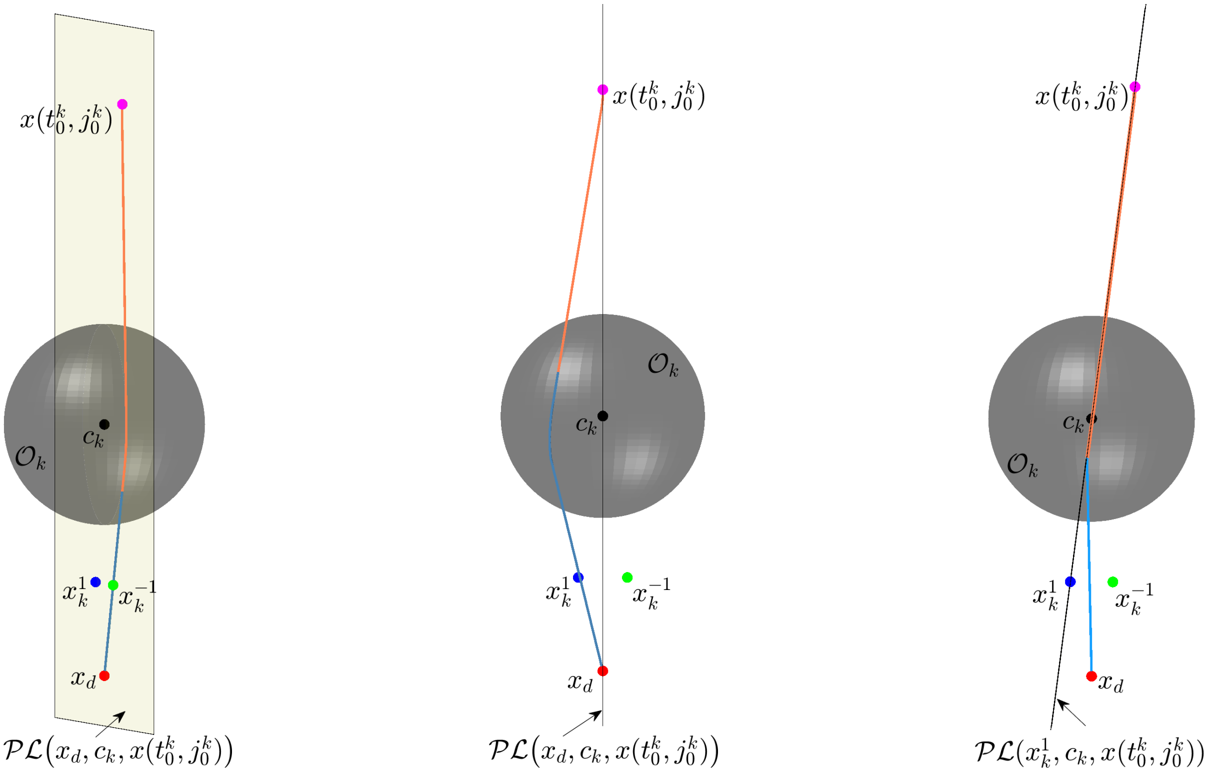
**Lemma 4.3** *Let the plane spanned by two non-colinear vectors $(q_1 - y) \in \mathbb{R}^n$ and $(q_2 - y) \in \mathbb{R}^n$ be denoted by $\mathcal{PL}(q_1, q_2, y)$. Consider the closed-loop hybrid system (4.26). For a given obstacle index $k \in \mathbb{I}$, if the virtual destinations $x_k^{-1}$ and $x_k^1$ belong to the plane $\mathcal{PL}(x_d, c_k, x(t_0^k, j_0^k))$ when the destination $x_d$, the obstacle's center $c_k$ and the position*

*$x(t_0^k, j_0^k)$ are not aligned, the motion takes place in the plane $\mathcal{PL}(x_d, c_k, x(t_0^k, j_0^k))$ where $(t_0^k, j_0^k)$ is the hybrid time at which obstacle $k$ is selected. If the points $x_d$, $c_k$ and $x(t_0^k, j_0^k)$ are aligned, the motion takes place in the plane $\mathcal{PL}(x_d, c_k, y)$ for a given $y \in \mathbb{R}^n \setminus \mathcal{L}(x_d, c_k)$ such that $x_k^m \in \mathcal{PL}(x_d, c_k, y)$ and $m \in \{-1, 1\}$.*

**Proof** See Appendix B.3.

The result of Lemma 4.3 requires selecting the virtual destinations associated with obstacle $k \in \mathbb{I}$ depending on the robot's position when it first enters the jump set $\mathcal{J}_0^k$ of obstacle $k$. Suppose the virtual destinations, defined in (4.12), belong to the two-dimensional plane $\mathcal{PL}(x_d, c_k, x(t_0^k, j_0^k))$ when $x(t_0^k, j_0^k) \in \mathcal{J}_k^0 \setminus \mathcal{L}(x_d, c_k)$. In this case, the motion generated by the closed-loop system (4.26), while obstacle $k$ is selected, is two-dimensional and takes place on the plane $\mathcal{PL}(x_d, c_k, x(t_0^k, j_0^k))$. It is clear that in the case where $x(t_0^k, j_0^k) \in \mathcal{J}_k^0 \cap \mathcal{L}(x_d, c_k)$, the plane of motion can be any plane $\mathcal{PL}(x_d, c_k, y)$ such that $y \in \mathbb{R}^n \setminus \mathcal{L}(x_d, c_k)$ and $x_d^{\pm 1} \in \mathcal{PL}(x_d, c_k, y)$. We should also mention that the case where the robot is initially in the set $\tilde{\mathcal{F}}_0$ is omitted since the robot will operate in the *motion-to-destination* mode until it enters the active region of an obstacle or converges to the destination if the line of sight to the destination is clear for the robot. The generated trajectory is then a line segment. An example illustrating the effect of selecting the virtual destinations as per Lemma 4.3 is shown in Fig. 4.5. In Fig.4.5(a), the virtual destinations belong to the two-dimensional plane $\mathcal{PL}(x_d, c_k, x(t_0^k, j_0^k))$, which results in a two-dimensional motion that takes place in the plane $\mathcal{PL}(x_d, c_k, x(t_0^k, j_0^k))$ for both modes considering obstacle $k$ (*obstacle-avoidance* mode represented by the orange curve and *motion-to-destination* mode represented by the blue curve). In Fig. 4.5(b), the virtual destinations do not belong to the plane $\mathcal{PL}(x_d, c_k, x(t_0^k, j_0^k))$. The motion in the two modes occurs in two different planes, as shown in Fig. 4.5(c), where the orange curve represents the *obstacle-avoidance* mode, and the blue curve represents the *motion-to-destination* mode.

When the robot's position belongs to the region $\left(\mathcal{J}_k^0 \cap \mathcal{C}_k\right)$, the jump maps (4.21)-(4.22) enable the mode $m$ to take the value 1 or $-1$ (indistinguishably) when switching from the *motion-to-destination* mode to the *obstacle-avoidance* mode (*i.e.,* avoid the obstacle considering, indistinguishably, the virtual destination $x_k^1$ or $x_k^{-1}$). Leveraging this property together with the fact that robot's motion, during every obstacle avoidance maneuver, is planar (as per Lemma 4.3), one can force the jump from the *motion-to-destination* mode, when $x \in \left(\mathcal{J}_k^0 \cap \mathcal{C}_k\right)$, to the *obstacle-avoidance* mode corresponding to the virtual destination closest to the robot's position. This will ensure a smooth transition from the *obstacle-avoidance* mode to the *motion-to-destination* mode, while guaranteeing locally optimal obstacle avoidance maneuvers as it will be shown later in Proposition 4.1. Figure 4.6(a) clearly shows that the green trajectory, generated by switching to the closest virtual destination, $x_k^{-1}$, to the robot's position when first entering the hysteresis region $\left(\mathcal{J}_k^0 \cap \mathcal{C}_k\right)$ (pink region), is shorter than the blue trajectory generated by selecting $x_k^1$. Figure 4.6(b) also shows that when selecting $x_k^{-1}$, the mode switches back to the *motion-to-destination* mode earlier than when selecting $x_k^1$, as $x_k^{-1}$ becomes visible to the robot before $x_k^1$ does. Another observation from Fig. 4.6(b) is that when switching back to the *motion-to-destination* mode, the robot's position, the destination $x_d$, and the virtual



(a) (b) (c)

Figure 4.5: Illustration of the property in Lemma 4.3 in a 3D space. In the left figure (a), the virtual destinations $x_k^{\pm 1}$ belong to the plane $\mathcal{PL}(x_d, c_k, x(t_0^k, j_0^k))$, resulting in a trajectory belonging to the same plane for the obstacle $\mathcal{O}_k$, where the *obstacle-avoidance* mode is represented by the orange curve and the *motion-to-destination* mode is represented by the blue curve. In the middle figure (b), the virtual destinations $x_k^{\pm 1}$ do not belong to the plane $\mathcal{PL}(x_d, c_k, x(t_0^k, j_0^k))$, resulting in a trajectory that does not belong to the plane $\mathcal{PL}(x_d, c_k, x(t_0^k, j_0^k))$. The right figure (c) shows that the trajectory generated in figure (b) does not belong to a single plane but to two different planes, as the virtual destinations are not on the plane $\mathcal{PL}(x_d, c_k, x(t_0^k, j_0^k))$.

destination $x_k^{-1}$ are aligned, which ensures the continuity of the velocity. As per the discussion above, we propose a modified version of the jump map $B(\cdot)$, defined in (4.22), as follows:

$$\hat{B}(x, k) := \begin{cases} 1 & x \in \mathcal{C}_k^1 \cup \big(\mathcal{C}_k \cap \mathcal{P}_<(c_k, x_k^{-1} - x_k^1)\big), \\ -1 & x \in \mathcal{C}_k^{-1} \cup \big(\mathcal{C}_k \cap \mathcal{P}_>(c_k, x_k^{-1} - x_k^1)\big), \\ \{-1, 1\} & x \in \mathcal{C}_k \cap \mathcal{P}_=(c_k, x_k^{-1} - x_k^1). \end{cases} \tag{4.28}$$

Note that by considering the modified jump map $\hat{B}(\cdot)$ in the mode switching scheme, the hysteresis region $\big(\mathcal{J}_k^0 \cap \mathcal{C}_k\big)$, when switching from the *motion-to-destination* mode to the *obstacle-avoidance* mode, reduces to a line segment (shown in red in Fig. 4.6) in the 2D case, thus losing the robustness of the hybrid system. The updating scheme of the mode

$m$ in the jump set of the *motion-to-destination* mode, considering the modified jump map (4.28), and the design of the virtual destinations in Lemma 4.3 are summarized in Algorithm 3, and the obtained result is stated in the following proposition.

**Proposition 4.1** *If the virtual destinations are designed as in Lemma 4.3 and the modified operation mode switching scheme* (4.21)-(4.28) *is considered, the hybrid closed-loop system* (4.26) *generates continuous velocity control inputs and locally optimal obstacle avoidance maneuvers.*

**Proof** See Appendix B.4.

**Algorithm 3** Mode selector updating scheme in the jump set of the *motion-to-destination* mode

```
Initialization: $x, x_d, k, c_k, r_k, e_k, \bar{r}$;
Ensure: $m$.
if $x \notin \mathcal{L}(x_d, c_k)$ then
    $y \to x$;
else
    Pick $y \in \mathbb{R}^n \setminus \mathcal{L}(x_d, c_k)$;
end if
Select the virtual destinations such that
$x_k^1 \in \mathcal{H}_k(x_d) \cap \mathcal{P}_{\geq}(p_k, x_d - p_k) \cap \mathcal{PL}(x_d, c_k, y)$
$\|x_d - x_k^1\| = e_k$
$x_k^{-1} \to x_d - \pi_r(\frac{c_k - x_d}{\|c_k - x_d\|})(x_k^1 - x_d)$;
if $x \in \mathcal{C}_k$ then
    if $x \in \mathcal{P}_{<}(c_k, x_k^{-1} - x_k^1)$ then
        $m \to 1$;
    else if $x \in \mathcal{P}_{>}(c_k, x_k^{-1} - x_k^1)$ then
        $m \to -1$;
    else
        $m \to \{-1, 1\}$;
    end if
else
    $m \to B(x, k)$ using (4.22);
end if
return m;
```

The implementation of the hybrid control (4.8) is summarized in Algorithm 4, where the steps colored in blue are only required for the sensor-based implementation. For compactness, we write the workspace's data and design parameters as follows:

$$c := [c_1, \ldots, c_b],\ r := [r_1, \ldots, r_b]^\top,\ \bar{r} := [\bar{r}_1, \ldots, \bar{r}_b]^\top,\ e := [e_1, \ldots, e_b]^\top.$$

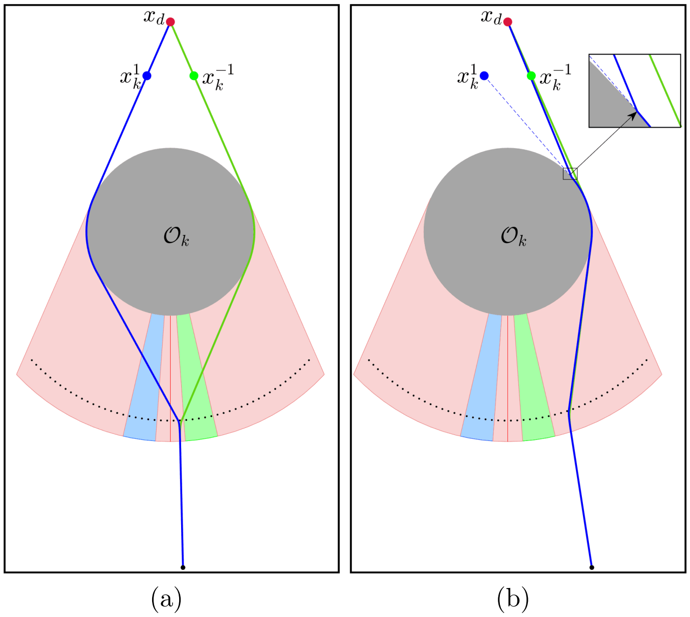


Figure 4.6: Effects of selecting the navigation mode according to Proposition 4.1. In the left figure (a), when first entering the jump set $\mathcal{J}_k^0$ through the hysteresis region (pink region), the mode $m$ switches to *obstacle-avoidance* mode where the blue trajectory is generated by selecting the virtual destination $x_k^1$ and the green trajectory is generated by selecting the closest virtual destination $x_k^{-1}$ to the robot's position. Both trajectories are smooth, but the blue trajectory is longer than the green trajectory. In the right figure (b), the blue trajectory is longer and non-smooth. The green (blue resp.) region in the jump set $\mathcal{J}_k^0$ is where, if the robot operates in the *motion-to-destination* mode, the mode $m$ can only jump to $m = -1$ ($m = 1$ resp.).

## 4.4 Sensor-based implementation

Since the workspace is assumed to contain spherical obstacles, one can reconstruct the obstacles from their detected portions obtained via a range scanner that covers a region $\mathcal{B}(x, R)$, $R > 0$, around the robot. As the detection region is limited to the sensor's range $R$, we redefine the range of the *active region* of obstacle $k \in \mathbb{I}$, defined in (4.4), by

$$\bar{r}_k \in (0, \tilde{r}_k), \ \tilde{r}_k := \min(\hat{r}_k, R). \tag{4.29}$$

Next, we implement our hybrid strategy in two and three-dimensional spaces using 2D and 3D LiDAR range scanners (*e.g.,* LEICA, BLK, ARC scanning modules).

### 4.4.1 Two-dimensional spaces

Consider a two-dimensional workspace, and assume that the robot is equipped with a LiDAR of resolution $d\psi > 0$, a maximum radial range $R > 0$, and an angular range of 360°. We model the measurements of the sensor, at a position $x$, by the polar curve

**Algorithm 4** General implementation of the hybrid control (4.8)

```
Initialization: $x_d$, $e_c$, $c$, $r$, $\bar{r}$, $e$, $\epsilon$, $e_s$, $x(0,0) \in \mathcal{X}$, $k(0,0) \in \mathbb{I}$, and $m(0,0) = 0$;
while true do
    Measure $x$;
    if $\|x - x_d\| \leq e_c$ then
        Break;
    else
        Measure $\mathcal{R}(x)$;
        Reconstruct obstacles using (4.32)-(4.34);
        Dilate obstacles radii: $r_i \leftarrow r_i + e_s$, $i \in \mathbb{S}_x$;
        Update $\bar{r}$;
        if $(x, m) \in \mathring{\mathcal{J}}_0 \times \{0\}$ then
            Update $k$ using (4.20);
            Update $m$ using Algorithm 3;
        else
            Update $k$ and $m$ using (4.8b)-(4.8c);
        end if
        Execute $u(x, k, m)$ using (4.8a);
    end if
end while
```

$\rho(x, \psi) : \mathcal{X} \times \mathcal{G} \to [0, R]$ defined as follows:

$$\rho(x, \psi) := \min \left( R, \min_{\substack{y \in \partial\mathcal{X} \\ \text{atan2}(y(2)-x(2), y(1)-x(1)) = \psi}} \|y - x\| \right), \tag{4.30}$$

where $\mathcal{G} := \{0, d\psi, 2d\psi, \ldots, 360 - d\psi\}$ is the set of scanned angles. The Cartesian coordinates of the scanned points are modeled by the mapping $\delta(x, \psi) : \mathcal{X} \times \mathcal{G} \to \mathcal{X}$ defined as follows:

$$\delta(x, \psi) := x + \rho(x, \psi)[\cos(\psi) \ \sin(\psi)]^\top. \tag{4.31}$$

Let $G_x(\delta)$ be the graph of the mapping $\delta$ at a position $x$. The set $\mathbb{I}_x \subset \mathbb{I}$ of the detected obstacles, at position $x$, is defined as $\mathbb{I}_x := \{i \in \mathbb{I} | d(x, \mathcal{O}_i) \leq R\}$. Assume that at each position $x$, the sensor returns a list of arcs $\mathcal{R}(x) := \left\{A_1, A_2, \ldots, A_{\iota(x)}\right\}$ from the detected obstacles corresponding to the intersection of the graph $G_x(\delta)$ and obstacles of the set $\mathbb{I}_x$, as shown in Fig. 4.7, where $\iota(x) = \mathbf{card}(\mathbb{I}_x)$. Using the arcs of the list $\mathcal{R}(x)$, at a position $x$, one can reconstruct the obstacles by determining their centers and radii. Due to the LiDAR's radial sweep, at positions where some obstacles are partially hidden by other obstacles, the detected arcs of partially hidden obstacles may be asymmetrical with respect to the projection of the robot's position on these obstacles. These asymmetrical arcs are ignored as they imply that the robot is outside the *active region* of their associated obstacles since, according to definitions (4.4) and (4.29), the active region must be free of any other obstacles (see also Fig. 4.7). The indices of obstacles associated with the symmetrical arcs, detected at position $x$, are grouped in the set $\mathbb{S}_x \subset \mathbb{I}_x$. Consider a symmetrical arc $A_i \in \mathcal{R}(x)$ associated with obstacle $\mathcal{O}_k$ where $i \in \{1, \dots, \iota(x)\}$ and $k \in \mathbb{S}_x$. The center $c_k$ and the radius $r_k$ can be obtained, as illustrated in Fig. 4.7, through the following steps:

- Determine the projection of $x$ onto the arc $A_i$ (*i.e.*, the closest point of obstacle $k$ to the robot) as follows:

$$\hat{c}_i := \arg\min_{y \in A_i} \|x - y\|. \tag{4.32}$$

- Determine the radius $r_k$ as follows:

$$r_k := \frac{b^2}{2\sqrt{b^2 - a^2}}, \tag{4.33}$$

  where $a = \|c_i^+ - c_i^-\|$, $b = \|c_i^+ - \hat{c}_i\| = \|c_i^- - \hat{c}_i\|$, and $c_i^+$, $c_i^-$ are the endpoints of arc $A_i$.

- Determine the center $c_k$ as follows:

$$c_k := \hat{c}_i + r_k \frac{\hat{c}_i - x}{\|\hat{c}_i - x\|}. \tag{4.34}$$

Since the centers and radii of the detected obstacles in the vicinity of the robot can be determined, the hybrid control (4.8) can be implemented in an unknown two-dimensional workspace with disc-shaped obstacles, as described in Algorithm 4 considering the steps colored in blue.

### 4.4.2 Three-dimensional spaces

Consider a three-dimensional workspace, and assume that the robot is equipped with a 3D-LiDAR of polar resolution $d\vartheta > 0$, polar angular range of 180°, azimuthal resolution $d\psi > 0$, azimuthal angular range of 360°, and a maximum radial range $R > 0$. We model the measurements of the sensor, at a position $x$, by the curve $\bar{\rho}(x, \vartheta, \psi) : \mathcal{X} \times \mathcal{U} \times \mathcal{G} \to [0, R]$ defined as follows:

$$\bar{\rho}(x, \vartheta, \psi) := \min\left( R, \min_{\substack{y \in \partial\mathcal{X} \\ \mathrm{atan2}(y(2)-x(2), y(1)-x(1)) = \psi \\ \arccos(\|y-x\|/(y(3)-x(3))) = \vartheta}} \|y - x\| \right), \tag{4.35}$$

where $\mathcal{U} := \{0, d\vartheta, 2d\vartheta, \dots, 360 - d\vartheta\}$ and $\mathcal{G} := \{0, d\psi, 2d\psi, \dots, 180 - d\psi\}$ are, respectively, the set of scanned polar angles and the set of scanned azimuthal angles. The Cartesian coordinates of the scanned points are modeled by the mapping $\bar{\delta}(x, \vartheta, \psi) : \mathcal{X} \times \mathcal{G} \times \mathcal{U} \to \mathcal{X}$ defined as follows:

$$\bar{\delta}(x, \vartheta, \psi) := x + \bar{\rho}(x, \vartheta, \psi)[\cos(\psi)\sin(\vartheta) \;\; \sin(\psi)\sin(\vartheta) \;\; \cos(\vartheta)]^\top. \tag{4.36}$$

Similar to the two-dimensional case, $G_x(\bar{\delta})$ represents the graph of the mapping $\bar{\delta}$ at a position $x$ and $\mathbb{I}_x \subset \mathbb{I}$ the set of detected obstacles at position $x$. Assume that at

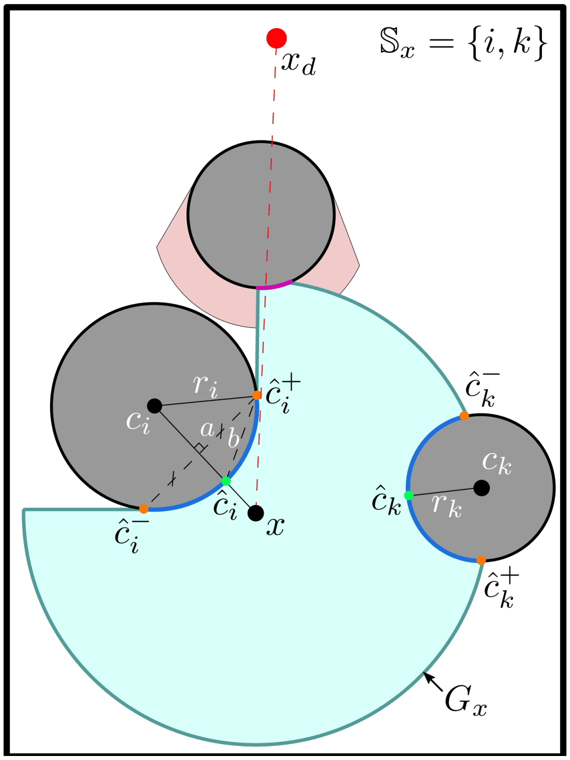


Figure 4.7: Obstacle reconstruction from sensor data.

each position $x$, the sensor returns a list of spherical caps $\bar{\mathcal{R}}(x) := \{\bar{A}_1, \bar{A}_2, \ldots, \bar{A}_{\iota(x)}\}$ from the detected obstacles corresponding to the intersection of the graph $G_x(\bar{\delta})$ and obstacles of the set $\mathbb{I}_x$, where $\iota(x) = \mathbf{card}(\mathbb{I}_x)$. The intersection of two spheres is a circle if the distance between their centers is less than the sum of their radii and if neither sphere is enclosed by the other. As a result, the detected spherical caps are formed by a bump with a circular base. At certain positions, some obstacles may be partially hidden from the sensor by other obstacles. The detected parts of the partially hidden obstacles may be asymmetrical with respect to the projection of the robot's position onto these obstacles. At positions where asymmetrical spherical caps are detected, the robot is certainly outside the *active regions* of obstacles associated with the asymmetrical caps since, as per definition (4.4)-(4.29), the *active region* of an obstacle should not contain any other obstacle. Consequently, we ignore the asymmetrical spherical caps as their associated obstacles are not required for the control. The indices of obstacles associated with the symmetrical spherical caps, detected at position $x$, are grouped in the set $\mathbb{S}_x \subset \mathbb{I}_x$. Consider a symmetrical spherical cap $\bar{A}_i \in \bar{\mathcal{R}}(x)$, detected at position $x \in \mathcal{X}$, associated with obstacle $\mathcal{O}_k$ where $i \in \{1, \ldots, \iota(x)\}$ and $k \in \mathbb{S}_x$. The reconstruction of obstacle $\mathcal{O}_k$ from its detected spherical cap $\bar{A}_i$ can be obtained as follows:

- Determine the projection of $x$ onto the spherical cap $\bar{A}_i$ (*i.e.*, the closest point of obstacle $\mathcal{O}_k$ to the robot)

$$\tilde{c}_i := \arg\min_{y \in \bar{A}_i} \|x - y\|. \tag{4.37}$$

- Determine the radius of obstacle $\mathcal{O}_k$

$$r_k := \frac{b^2}{2\sqrt{b^2 - a^2}}, \tag{4.38}$$

where $a$ is the radius of the circular basis $\bar{\mathcal{C}}_i$ of the portion $\bar{A}_i$, and $b = d(\tilde{c}_i, \bar{\mathcal{C}}_i)$.

- Determine the center of obstacle $\mathcal{O}_k$

$$c_k := \tilde{c}_i + r_k \frac{\tilde{c}_i - x}{\|\tilde{c}_i - x\|}. \tag{4.39}$$

Now, with the information of neighboring obstacles to the robot available through the sensor's output, the hybrid control (4.8) can be implemented in unknown spherical three-dimensional spaces as summarized in Algorithm 4, considering the steps colored in blue and replacing $\mathcal{R}(x)$ in step 6 of the algorithm with $\bar{\mathcal{R}}(x)$.

**Remark 4.1** *For safer navigation, the numerical errors and low resolution of the LiDAR should be considered. Therefore, a security margin $e_s > 0$ can be added to the radius of the detected obstacles where the separation between every pair of obstacles has to be larger than $2e_s$ (i.e., $\forall i, k \in \mathbb{I}$, $i \neq k$, $\|c_k - c_i\| - r_k - r_i > 2e_s$.)*

## 4.5 Numerical simulations

### 4.5.1 Implementation with global knowledge of the environment

In order to visualize the performance of our proposed hybrid approach, we compare it with another hybrid approach that considers a single integrator model and guarantees safety and GAS in $n$-dimensional Euclidean spaces, proposed in (Berkane et al., 2022). We performed simulations starting from 10 different initial conditions in two different workspaces. The first experiment is done in a two-dimensional environment, as shown in Fig. 4.8, where we plotted the trajectories obtained by our approach along with the trajectories generated by the approach proposed in (Berkane et al., 2022). We also report the relative length difference of the paths generated by the approach proposed in (Berkane et al., 2022) with respect to ours in Table 4.1. For each initial position $p_i$, $i \in \{1, \dots, 10\}$, in Fig. 4.8, we computed the relative length difference $RLD_i = 100(L_i - l_i)/l_i$, where $L_i$ is the length of the $i$th path generated by the approach proposed in (Berkane et al., 2022), and $l_i$ is the length of the path generated by our approach. The trajectories plotted in Fig. 4.8 show clearly that our approach generates a continuous control input (robot's velocity) while the approach proposed in (Berkane et al., 2022) generates a discontinuous control input. We can also observe that our trajectories are shorter than the ones generated by the approach proposed in (Berkane et al., 2022), which is confirmed by the positive relative difference reported in Table 4.1. This difference in length is mainly due to the fact that our approach starts the *obstacle-avoidance* mode, with local optimal maneuvers, once in the *active region* of an obstacle and switches back to the *motion-to-destination* once the avoided obstacle is no longer blocking the view of the destination $x_d$. In contrast, the approach proposed in (Berkane et al., 2022) starts the *obstacle-avoidance* mode in close vicinity of the obstacle, performing a boundary-following motion on a helmet covering the obstacle. It does not switch back to the *motion-to-destination* mode once the avoided obstacle stops blocking the view of the destination $x_d$, but once it exits the helmet. To ensure that the performance of our approach is preserved regardless of the dimension of the workspace, we repeated the same experiment in a three-dimensional environment and reported the results in Fig. 4.9 and Table 4.2. The same observations can be drawn from this experiment as in the 2D case, concluding the efficiency of our approach in higher dimensions. Notably, the approach proposed in (Berkane et al., 2022) considers a more general class of obstacles, namely ellipsoids, whereas our approach only considers spheres. Nevertheless, our approach can be implemented in *a priori* unknown environments using only on-board range scanners as illustrated in the simulations in Subsection 4.5.2.

We repeat the same two comparison experiments in the same environments as in

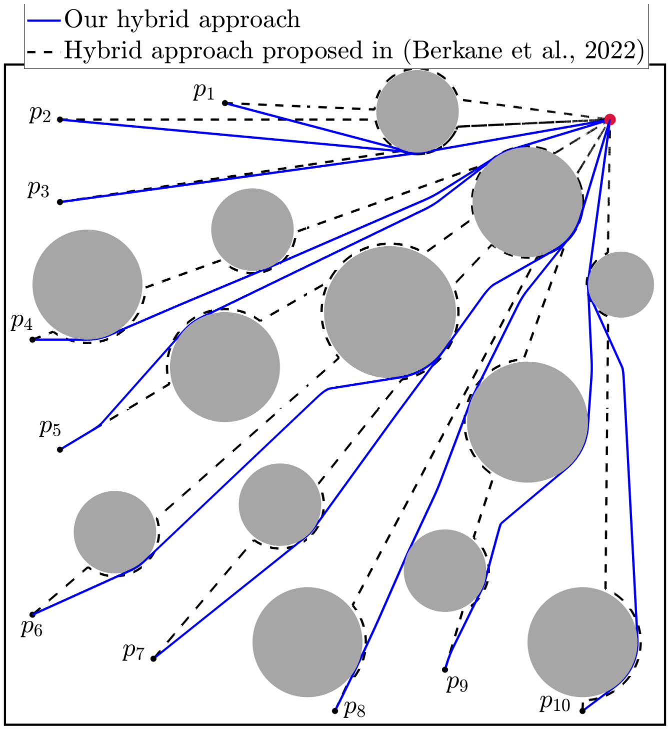


Figure 4.8: Robot navigation trajectories from ten different initial positions in a 2D workspace cluttered with circular obstacles. The blue trajectories are generated by our hybrid approach and the black trajectories are generated by the hybrid approach proposed in (Berkane et al., 2022). The target is represented by the red dot.

Fig. 4.8 and 4.9 but with the continuous quasi-optimal approach proposed in Chapter 3 to clearly discern the gains and losses in terms of performance when switching from the continuous scheme to the hybrid scheme. For the 2D case, the trajectories of our hybrid approach and the quasi-optimal approach are shown in Fig. 4.10. The relative length difference of the paths generated by our hybrid approach with respect to the quasi-optimal approach is reported in Table 4.3. For each initial position $p_i$, $i \in \{1, \ldots, 10\}$,

Table 4.1: The relative length difference of the paths, shown in Fig. 4.8, generated by the hybrid approach proposed in (Berkane et al., 2022) with respect to our hybrid approach in a 2D workspace.

| **Paths** | $\boldsymbol{p_1}$ | $\boldsymbol{p_2}$ | $\boldsymbol{p_3}$ | $\boldsymbol{p_4}$ | $\boldsymbol{p_5}$ | $\boldsymbol{p_6}$ | $\boldsymbol{p_7}$ | $\boldsymbol{p_8}$ | $\boldsymbol{p_9}$ | $\boldsymbol{p_{10}}$ |
|---|---|---|---|---|---|---|---|---|---|---|
| $RLD$ (%) | 7.36 | 4.39 | 1.76 | 7.33 | 10.68 | 9.97 | 11.2 | 3.95 | 6.08 | 5.14 |

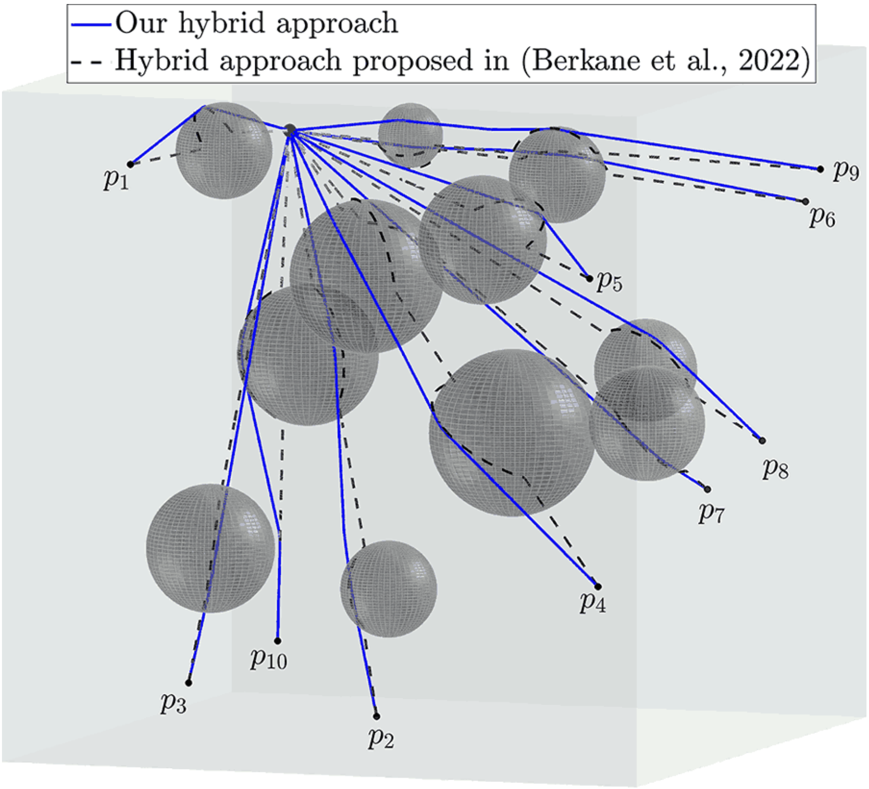


Figure 4.9: Robot navigation trajectories from ten different initial positions in a 3D workspace cluttered with spherical obstacles. The blue trajectories are generated by our hybrid approach and the black trajectories are generated by the hybrid approach proposed in (Berkane et al., 2022). The target is represented by the red dot.

Table 4.2: The relative length difference of the paths, shown in Fig. 4.9, generated by the hybrid approach proposed in (Berkane et al., 2022) with respect to our hybrid approach in a 3D workspace.

| **Paths** | $\boldsymbol{p_1}$ | $\boldsymbol{p_2}$ | $\boldsymbol{p_3}$ | $\boldsymbol{p_4}$ | $\boldsymbol{p_5}$ | $\boldsymbol{p_6}$ | $\boldsymbol{p_7}$ | $\boldsymbol{p_8}$ | $\boldsymbol{p_9}$ | $\boldsymbol{p_{10}}$ |
|---|---|---|---|---|---|---|---|---|---|---|
| $RLD$ (%) | 6.38 | 4.91 | 4.18 | 8.78 | 5.02 | 5.71 | 5.12 | 4.9 | 7.15 | 5.25 |

in Fig. 4.10, The relative length difference $RLD_i = 100(l_i^h - l_i)/l_i$ is computed for the ten initial positions $p_i$, $i \in \{1, \ldots, 10\}$, illustrated in Fig. 4.10, where $l_i^h$ is the length of the $i$th path generated by our hybrid approach and $l_i$ is the length of the path generated by the quasi-optimal approach. Paths starting from the initial conditions $p_1$ and $p_9$ are not taken into account in Table 4.3 since $p_1$ is an undesirable equilibrium point for the controller of the quasi-optimal approach, and $p_9$ lies in the attraction region (nests) of undesirable equilibria of the controller of the quasi-optimal approach, as shown in Fig. 4.10. The positive relative length differences in Table 4.3 indicate that our hybrid approach generated slightly longer paths in some cases and identical ones in others. One can observe the reduction of performance in terms of path length optimality for the hybrid approach with respect to the quasi-optimal approach, which can be justified by the fact that the hybrid approach starts the avoidance maneuver once the robot enters the active region and must pass through the *motion-to-destination* mode before it can avoid another obstacle. In contrast, the avoidance maneuvers of the quasi-optimal approach are not limited by the active regions of the obstacles and proceed consecutively until the target is in the robot's line of sight. Nevertheless, the hybrid approach solves the global navigation problem where the robot can navigate from all initial positions in the free space, whereas the quasi-optimal approach suffers from undesired equilibria with attraction regions of non-zero measure in 2D workspaces. The results for the 3D scenario are reported in Fig. 4.11 and Table 4.4. The same observations as in the 2D scenario can be made concerning the path length optimality but finding nests in this case (3D) was not feasible as was conjectured in Chapter 3 that the quasi-optimal approach guarantees AGAS of the target location when the dimension of the space $n = 3$.

Table 4.3: The relative length difference of the paths, shown in Fig. 4.10, generated by the hybrid approach with respect to the quasi-optimal approach in a 2D workspace.

| **Paths** | $\boldsymbol{p_2}$ | $\boldsymbol{p_3}$ | $\boldsymbol{p_4}$ | $\boldsymbol{p_5}$ | $\boldsymbol{p_6}$ | $\boldsymbol{p_7}$ | $\boldsymbol{p_8}$ | $\boldsymbol{p_{10}}$ |
|---|---|---|---|---|---|---|---|---|
| *RLD* (%) | 0 | 0 | 0.09 | 0.41 | 1.42 | 0.74 | 0.17 | 0.67 |

Table 4.4: The relative length difference of the paths, shown in Fig. 4.11, generated by the hybrid approach with respect to the quasi-optimal approach in a 3D workspace.

| **Paths** | $\boldsymbol{p_1}$ | $\boldsymbol{p_2}$ | $\boldsymbol{p_3}$ | $\boldsymbol{p_4}$ | $\boldsymbol{p_5}$ | $\boldsymbol{p_6}$ | $\boldsymbol{p_7}$ | $\boldsymbol{p_8}$ | $\boldsymbol{p_9}$ | $\boldsymbol{p_{10}}$ |
|---|---|---|---|---|---|---|---|---|---|---|
| *RLD* (%) | 0 | 0.02 | 0 | 0.04 | 0 | 0 | 0 | 0 | 0.05 | 0.03 |

### 4.5.2 Sensor-based implementation

To test the practicality of our approach, we simulated the sensor-based implementation in the same spaces as in the general implementation section 4.5.1. For the 2D case, we used a 360°-LiDAR model with 0.5° resolution and $2m$ radial range. For the 3D

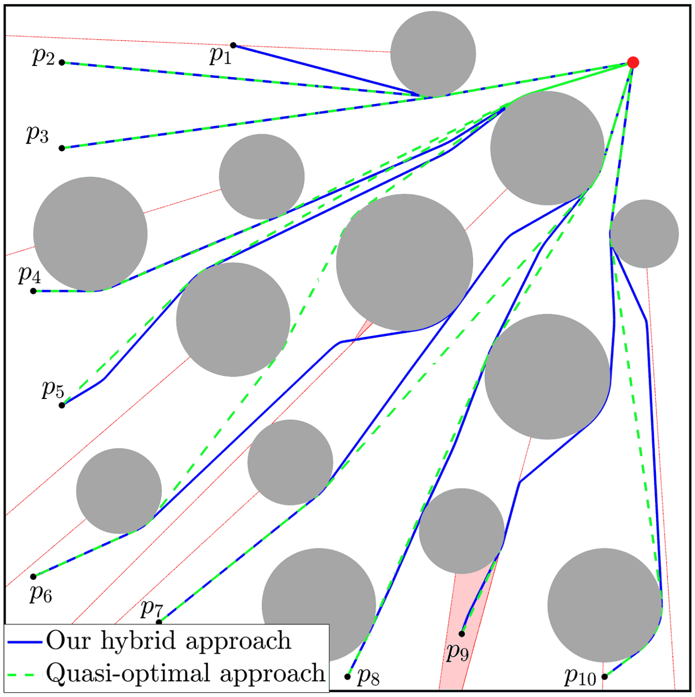


Figure 4.10: Robot navigation trajectories from ten different initial positions in a 2D workspace cluttered with circular obstacles. The trajectories generated by our hybrid approach are represented by the blue trajectories, the trajectories generated by the quasi-optimal approach are represented by the green dashed curves, and the target is represented by the red dot. The red dash-dotted line segments represent the sets of undesired equilibria of the quasi-optimal control and the red areas represent the attraction regions of the undesired equilibria (nests).

case, we used a 3D-LiDAR with 1° polar and azimuthal resolutions, 180° polar angular range, 360° azimuthal angular range, and $2m$ radial range. We considered a security margin $e_s = 0.1m$ for the obstacles radii. We plotted the trajectories obtained through the sensor-based implementation along with the ones generated when the environment is *a priori known* in Fig. 4.12 (resp. Fig. 4.13) for the 2D case (resp. for the 3D case). As the sensor's range is limited, the *active regions* have been redefined in (4.29), which explains the fact that the sensor-based implementation generates paths that are, in some cases, longer than the general implementation. Nevertheless, the sensor-based implementation generally reproduces the same trajectories as the implementation with a global knowledge of the environment. Simulation videos for the 2D case and 3D case showing the sensor-based navigation can be found online.[1].

[1] [Online]. Available: `https://youtube.com/shorts/bL0d0l7W9Ms?feature=share` and `https://youtu.be/oJqpUW8Blb4`

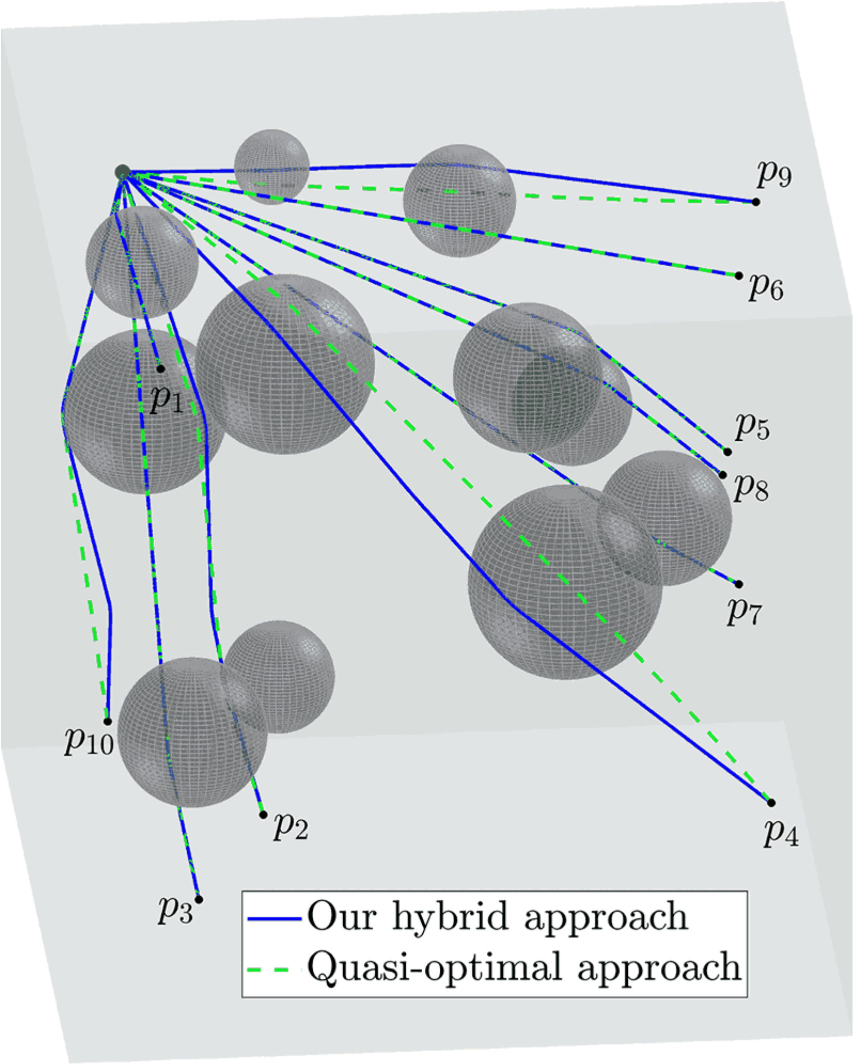


Figure 4.11: Robot navigation trajectories from ten different initial positions in a 3D workspace cluttered with spherical obstacles. The trajectories generated by our hybrid approach are represented by the blue trajectories, the trajectories generated by the quasi-optimal approach are represented by the green dashed trajectories, and the target is represented by the red dot.

## 4.6 Experimental validation

Under the same experimental settings as in Section 3.8.1 and using the parameters in Table 3.4, we set up a $6.65m \times 4.2m$ workspace with four obstacles. The robot is initially at the origin with its heading aligned with the $x$-axis of the workspace ($\Phi = 0$), and the target is set at the position $x_d = [6.1\ 3.6]^\top$. The experimental results are shown in Fig. 4.14 and in a video that can be found online[2]. The top figure of Fig. 4.14 shows the workspace configuration with the initial and final positions. In the bottom figure, the

[2] https://youtu.be/rQc062EDYts

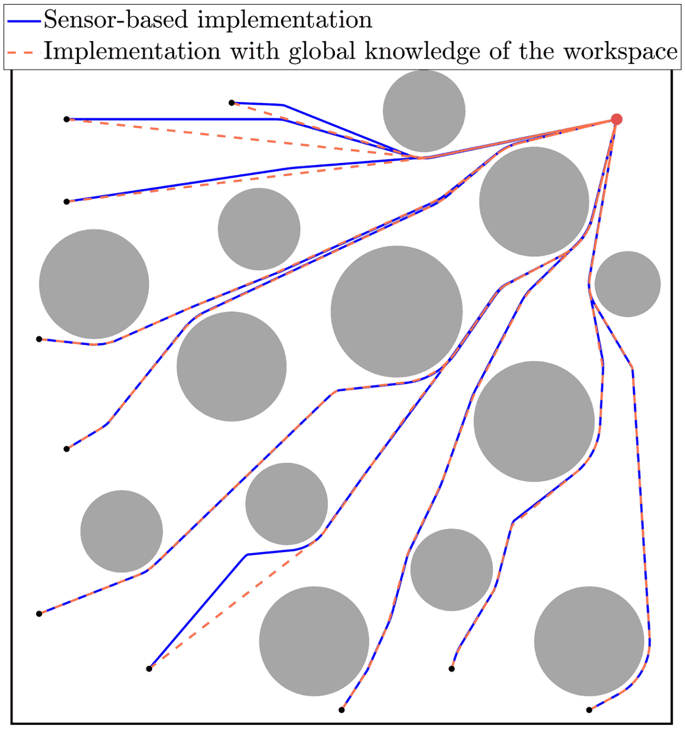


Figure 4.12: Robot navigation trajectories from ten different initial positions in a 2D workspace cluttered with circular obstacles. The blue trajectories are generated by the sensor-based implementation of our hybrid approach, and the orange trajectories are generated by our hybrid approach when the environment is *a priori* known. The red dot represents the target.

trajectory of the robot is plotted in an orthographic projection top view of the workspace. The obtained results illustrate the safe navigation of the robot from the initial position to the final destination. We also carried out a comparative experimental study with respect to some popular reactive feedback-based autonomous navigation algorithms, namely, the separating hyperplane approach (SH) (Arslan and Koditschek, 2019), the vector field histogram approach (VFH) (Borenstein and Koren, 1991), and the quasi-optimal navigation approach (QO) (Cheniouni et al., 2024c). Under the same settings as in the previous experiment, the robot starts from the origin of the workspace with its heading aligned with the $x$-axis and navigates towards the target $x_d = [6.6 - 3]$ as shown in Fig. 4.15a. The generated trajectories with the four different algorithms are shown in Fig. 4.15a and the video of the experiment can be found online[3]. The execution time and path length of each algorithm are reported in Table 4.15b. One can notice that the quasi-optimal approach developed in our previous work (Cheniouni et al., 2023a) generates the shortest path and has the lowest execution time. Although the performance of our previously proposed quasi-optimal navigation approach is slightly better than that of the proposed hybrid feedback navigation approach, the former, unlike the latter, does not guarantee global asymptotic stability of the target location. Indeed, if the robot starts

[3] `https://youtu.be/KzUNLwQ5lMo`

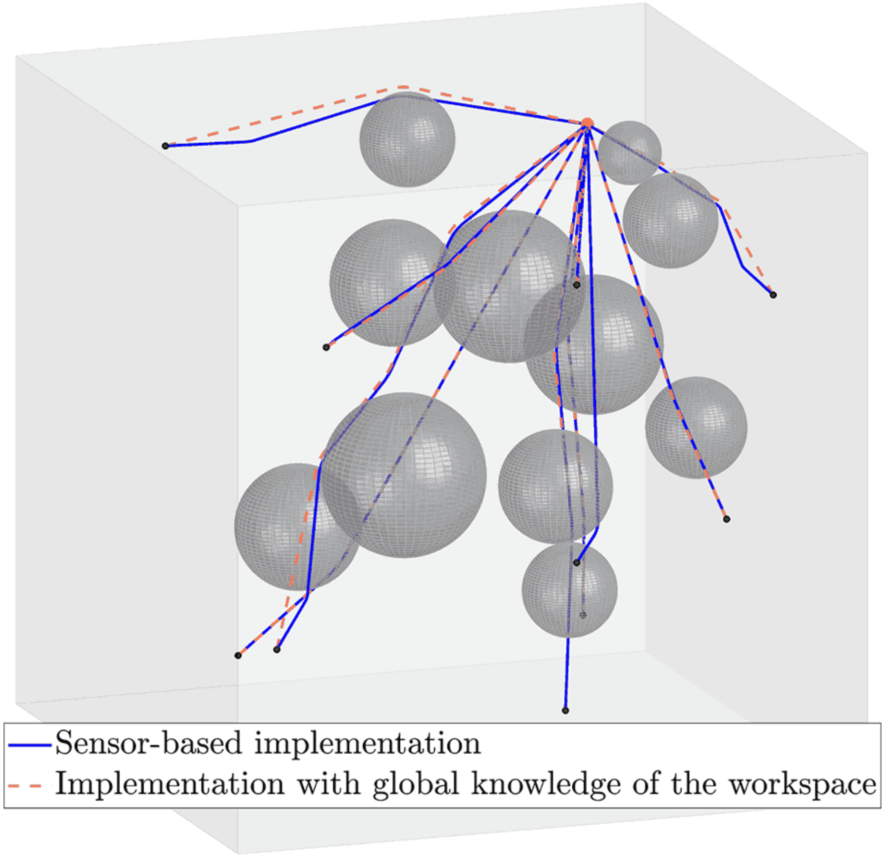


Figure 4.13: Robot navigation trajectories from ten different initial positions in a 3D workspace cluttered with spherical obstacles. The blue trajectories are generated by the sensor-based implementation of our hybrid approach, and the orange trajectories are generated by our hybrid approach when the environment is *a priori* known. The red dot represents the target.

from the set of undesired equilibria, it can get stuck and have difficulty getting out, as shown in the video available online[4]. For the VFH approach, selecting an appropriate threshold[5] depends on the workspace, which is crucial for safety and performance. In our experiment, the threshold was taken as 70000. The four implemented algorithms can be found online[6].

[4] `https://youtu.be/1gDqVkkAU0Y`

[5] The lower bound of the polar histogram values, indicating the presence of obstacles.

[6] `https://github.com/IshakChen9/navigate_TBT4_pkg.git`

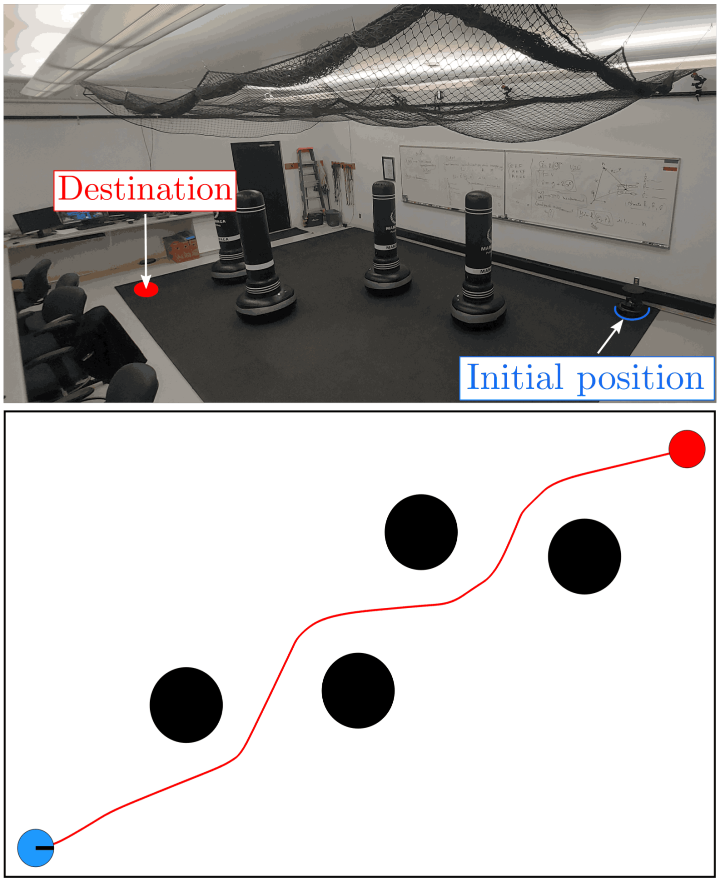
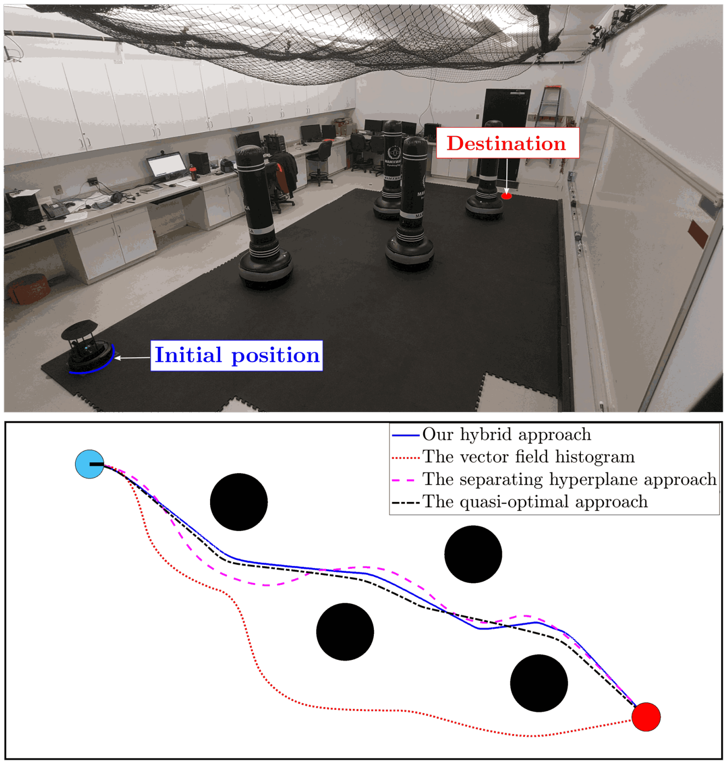


Figure 4.14: The top figure shows the workspace configuration with initial and final positions highlighted. The bottom figure is a plot of the robot's trajectory in an orthographic projection of the workspace's top view.



(a)

| Algorithm | SH | VFH | QO | Hybrid approach |
|---|---|---|---|---|
| **Execution time** | $13.21ms$ | $1.69ms$ | $0.43ms$ | $0.88ms$ |
| **Path length** | $7.8405m$ | $8.1877m$ | $7.4541m$ | $7.5584m$ |

(b)

Figure 4.15: a) Example navigation scenario in a priori unknown environment, showing the robot's trajectory generated by the proposed hybrid feedback control approach (blue) compared to alternative methods. (b) Performance comparison highlighting the path length and computational efficiency of the proposed approach. The proposed approach generates paths similar to our previously proposed quasi-optimal (QO) approach (Cheniouni et al., 2023a) while avoiding the issue of undesired equilibria in QO approach. The details of this experiment are reported in Section 4.6. The complete experiment can be visualized in the video available online `https://youtu.be/KzUNLwQ5lMo`.

# Chapter 5

# Conclusions

## 5.1 Summary

In this thesis, we proposed two solutions for autonomous navigation in $n$-dimensional environments with convex obstacles. The first solution is a continuous feedback controller that guarantees safe and *quasi-optimal* navigation with AGAS guarantees. A sensor-based version of this solution is implemented in two-dimensional convex worlds under certain curvature conditions. The second solution is a hybrid feedback controller ensuring safe navigation with locally optimal obstacle avoidance and GAS guarantees.

In Chapter 3, a continuous feedback control strategy for the autonomous navigation problem in an $n$-dimensional sphere world has been proposed, with safety guarantees and *quasi-optimal* trajectory generation. The proposed strategy consists in steering the robot tangentially to the blocking obstacles through successive projections of the nominal control onto the obstacles' enclosing cones. Consequently, the deviations from the nominal direction to the target are minimized with respect to each blocking obstacle, resulting in a *quasi-optimal* overall collision-free trajectory. The price to pay for the almost global asymptotic stability result, in two-dimensional sphere worlds, is a somewhat restrictive assumption on the configuration of the obstacles (Assumption 3.3) that has been lifted in the sensor-based version, where the robot can navigate to the target location from almost everywhere in the free space without prior knowledge of environment containing sufficiently curved convex obstacles. An experimental validation is also conducted, showing the effectiveness of the sensor-based approach.

In Chapter 4, an autonomous robot navigation scheme is proposed for navigation in $n$-dimensional Euclidean spaces with an arbitrary number of ball-shaped obstacles. The target location is proved to be globally asymptotically stable for the closed-loop system under the proposed hybrid feedback controller that switches between two modes of operation, namely, the *obstacle-avoidance* mode and the *motion-to-destination* mode. The proposed control scheme, besides guaranteeing global asymptotic stability of the target location, generates continuous control inputs and locally optimal obstacle avoidance maneuvers. Notably, the proposed scheme is implementable using only local sensor information, such as from LiDAR or vision systems. Experimental validation using the TurtleBot4 platform confirmed the effectiveness and practicality of the proposed approach. This approach can also be implemented in environments cluttered with obstacles of arbitrary shape by enclosing them in spheres. However, this could be a conservative approach, as some valid obstacle-free regions would not be navigable.

## 5.2 Perspectives

The contributions presented in this thesis offer a preliminary glimpse into the potential of integrating path optimality with feedback-based navigation approaches. Some possible extensions of our work and future directions are provided below.

The *quasi-optimal* approach of Chapter 3 guarantees almost global asymptotic stability in 2D spaces under Assumption 3. Although no proof of AGAS was provided for higher dimensions ($n \geq 3$), extensive simulations in 3D spaces have led us to conjecture that AGAS holds at least in the 3D case. This conjecture is reinforced by the observation that nests are only formed when all obstacles involved in successive projections within a specific region have their centers in the same plane as the target. In this case, the problem reduces to a local 2D configuration, and nests are generated when the conditions discussed in Section 3.5 are met. Figure 5.1 shows an example of two obstacles whose centers and the target are coplanar, where the green region represents the nest formed by the equilibria associated with these two obstacles. Since this nest is planar, it has zero Lebesgue measure. The worst case observed is similar to the 2D case shown in Fig. 3.8, where the generated nest forms a 2D barrier with a Lebesgue measure zero, as illustrated in Fig. 5.2 by the alternating green and red regions. A rigorous proof establishing AGAS of the target location for $n \geq 3$ would be a valuable contribution, further strengthening our *quasi-optimal* navigation approach.

Another promising avenue for future work is extending the proposed approach to handle arbitrarily shaped obstacles. This would involve adopting the concept of a conic hull instead of a regular enclosing cone. However, defining an appropriate projection that minimizes the velocity deviation (robot's heading) from the nominal direction becomes a challenging task when the projection cone is not circularly symmetric around an axis.

Improving the recursive minimization process is another key area for enhancement. In the current framework, successive projections depend on the previous ones, and consequently, the final projection, which defines the robot's velocity, may not provide the optimal direction due to its dependence on earlier projections and the obstacles considered at each step. This issue is illustrated in Fig. 3.16, where the generated path do not necessarily reflect the shortest possible trajectory.

Moreover, the proposed navigation strategies in this dissertation are designed for velocity-controlled vehicles (first-order kinematic system). These navigation strategies could be extended to second-order dynamics through the reference governor approach (Arslan and Koditschek, 2017; İşleyen et al., 2022). Although our proposed hybrid approach solves the autonomous navigation problem with safety and global asymptotic stability guarantees, it only provides local optimal avoidance maneuvers and does not generate *quasi-optimal* trajectories. Designing autonomous navigation schemes with safety, global asymptotic stability, and path-length optimality guarantees is a challenging open problem that would be an interesting future work.

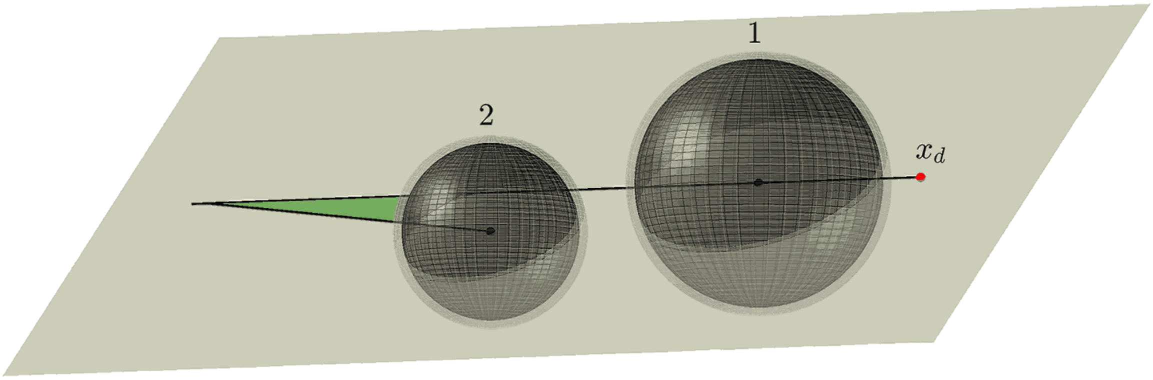


Figure 5.1: Nest generation (green region) in a 3D sphere world with two obstacles.

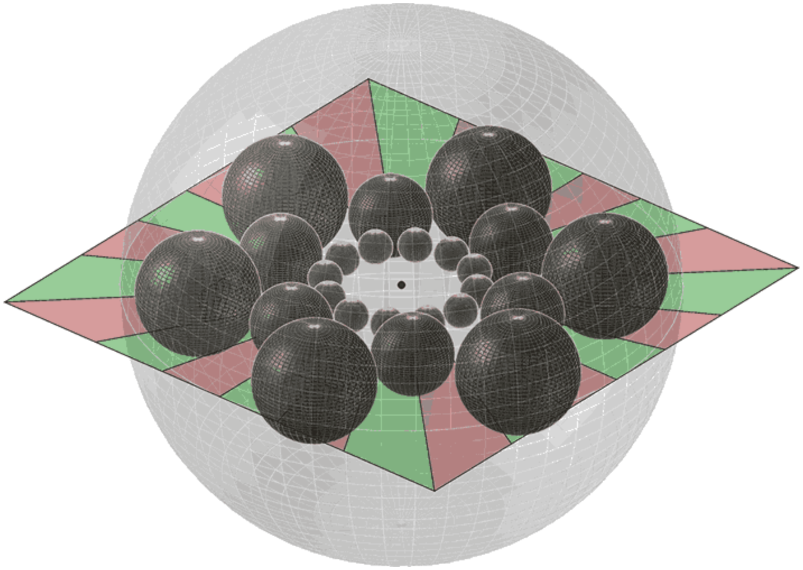


Figure 5.2: Generation of the largest nest (union of the colored planar regions representing cells) in a 3D sphere world with multiple obstacles.

# Bibliography

Alterovitz, R., Patil, S., and Derbakova, A. (2011). Rapidly-exploring roadmaps: Weighing exploration vs. refinement in optimal motion planning. In *2011 IEEE International Conference on Robotics and Automation*, pages 3706–3712.

Ames, A. D., Grizzle, J. W., and Tabuada, P. (2014). Control barrier function based quadratic programs with application to adaptive cruise control. In *53rd IEEE Conference on Decision and Control*, pages 6271–6278.

Ames, A. D., Xu, X., Grizzle, J. W., and Tabuada, P. (2017). Control barrier function based quadratic programs for safety critical systems. *IEEE Transactions on Automatic Control*, 62(8):3861–3876.

Arslan, O. and Koditschek, D. E. (2016). Exact robot navigation using power diagrams. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1–8.

Arslan, O. and Koditschek, D. E. (2017). Smooth extensions of feedback motion planners via reference governors. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4414–4421.

Arslan, O. and Koditschek, D. E. (2019). Sensor-based reactive navigation in unknown convex sphere worlds. *The International Journal of Robotics Research*, 38(2-3):196–223.

Arslan, O. and Tsiotras, P. (2013). Use of relaxation methods in sampling-based algorithms for optimal motion planning. In *2013 IEEE International Conference on Robotics and Automation*, pages 2421–2428.

Barraquand, J., Kavraki, L., Latombe, J.-C., Li, T.-Y., Motwani, R., and Raghavan, P. (1996). A random sampling scheme for path planning. In Giralt, G. and Hirzinger, G., editors, *Robotics Research*, pages 249–264, London. Springer London.

Berkane, S. (2021). Navigation in unknown environments using safety velocity cones. In *2021 American Control Conference (ACC)*, pages 2336–2341.

Berkane, S., Bisoffi, A., and Dimarogonas, D. V. (2019). A hybrid controller for obstacle avoidance in an $n$-dimensional euclidean space. In *2019 18th European Control Conference (ECC)*, pages 764–769.

Berkane, S., Bisoffi, A., and Dimarogonas, D. V. (2022). Obstacle avoidance via hybrid feedback. *IEEE Transactions on Automatic Control*, 67(1):512–519.

Blanchini, F. and Miani, S. (2007). *Set-Theoretic Methods in Control*. Birkhäuser Basel, 1st edition.

Bohlin, R. and Kavraki, L. (2000). Path planning using lazy prm. In *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, volume 1, pages 521–528 vol.1.

Borenstein, J. and Koren, Y. (1991). The vector field histogram-fast obstacle avoidance for mobile robots. *IEEE Transactions on Robotics and Automation*, 7(3):278–288.

Braun, P., Kellett, C. M., and Zaccarian, L. (2018). Unsafe point avoidance in linear state feedback. In *2018 IEEE Conference on Decision and Control (CDC)*, pages 2372–2377.

Braun, P., Kellett, C. M., and Zaccarian, L. (2021). Explicit construction of stabilizing robust avoidance controllers for linear systems with drift. *IEEE Transactions on Automatic Control*, 66(2):595–610.

Canny, J. and Reif, J. (1987). New lower bound techniques for robot motion planning problems. In *28th Annual Symposium on Foundations of Computer Science (sfcs 1987)*, pages 49–60.

Chai, J. and Sanfelice, R. G. (2019). Forward invariance of sets for hybrid dynamical systems (part i). *IEEE Transactions on Automatic Control*, 64(6):2426–2441.

Cheniouni, I., Berkane, S., and Tayebi, A. (2023a). Safe and quasi-optimal autonomous navigation in environments with convex obstacles. *arXiv:2308.13425*.

Cheniouni, I., Berkane, S., and Tayebi, A. (2024a). Hybrid feedback control for global and optimal safe navigation. In *2024 American Control Conference (ACC)*, pages 1131–1136.

Cheniouni, I., Berkane, S., and Tayebi, A. (2024b). Hybrid feedback control for global navigation with locally optimal obstacle avoidance in $n$-dimensional spaces. *IEEE Transactions on Robotics (submitted)*.

Cheniouni, I., Berkane, S., and Tayebi, A. (2024c). Safe and quasi-optimal autonomous navigation in environments with convex obstacles. *IEEE Transactions on Automatic Control*, pages 1–16.

Cheniouni, I., Tayebi, A., and Berkane, S. (2023b). Safe and quasi-optimal autonomous navigation in sphere worlds. In *2023 American Control Conference (ACC)*, pages 2678–2683.

Connolly, C. (1992). Applications of harmonic functions to robotics. In *Proceedings of the 1992 IEEE International Symposium on Intelligent Control*, pages 498–502.

Connolly, C., Burns, J., and Weiss, R. (1990). Path planning using laplace's equation. In *Proceedings., IEEE International Conference on Robotics and Automation*, pages 2102–2106 vol.3.

Constantinou, N. and Loizou, S. G. (2020). Robot navigation on star worlds using a single-step navigation transformation. In *2020 59th IEEE Conference on Decision and Control (CDC)*, pages 1537–1542.

Cortes, J. (2008). Discontinuous dynamical systems. *IEEE Control Systems Magazine*, 28(3):36–73.

Czimmermann, T., Chiurazzi, M., Milazzo, M., Roccella, S., Barbieri, M., Dario, P., Oddo, C. M., and Ciuti, G. (2022). An autonomous robotic platform for manipulation and inspection of metallic surfaces in industry 4.0. *IEEE Transactions on Automation Science and Engineering*, 19(3):1691–1706.

Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:269–271.

Dimarogonas, D. V. and Kyriakopoulos, K. J. (2007). Decentralized navigation functions for multiple robotic agents with limited sensing capabilities. *Journal of Intelligent and Robotic Systems*, 48(3):411–433.

Dimarogonas, D. V., Loizou, S. G., and Kyriakopoulos, K. J. (2006). *Multirobot Navigation Functions II: Towards Decentralization*, pages 209–253. Springer Berlin Heidelberg, Berlin, Heidelberg.

Feder, H. and Slotine, J.-J. (1997). Real-time path planning using harmonic potentials in dynamic environments. In *Proceedings of International Conference on Robotics and Automation*, volume 1, pages 874–881 vol.1.

Filippidis, I. and Kyriakopoulos, K. J. (2011). Adjustable navigation functions for unknown sphere worlds. In *2011 50th IEEE Conference on Decision and Control and European Control Conference*, pages 4276–4281.

Filippidis, I. and Kyriakopoulos, K. J. (2013). Navigation functions for focally admissible surfaces. In *2013 American Control Conference*, pages 994–999.

Filippidis, I. F. and Kyriakopoulos, K. J. (2012). Navigation functions for everywhere partially sufficiently curved worlds. In *2012 IEEE International Conference on Robotics and Automation*, pages 2115–2120.

Georges, B. (1932). *Introduction à la géométrie infinitésimale directe*. Librairie Vuibert, Paris.

Goebel, R., Sanfelice, R. G., and Teel, A. R. (2009). Hybrid dynamical systems. *IEEE Control Systems Magazine*, 29(2):28–93.

Goebel, R., Sanfelice, R. G., and Teel, A. R. (2012). *Hybrid Dynamical Systems: Modeling, Stability, and Robustness*. Princeton University Press.

Guzzetti, D. and Baoyin, H. (2019). Human path-planning for autonomous spacecraft guidance at binary asteroids. *IEEE Transactions on Aerospace and Electronic Systems*, 55(6):3126–3138.

Haidegger, T. (2019). Autonomy for surgical robots: Concepts and paradigms. *IEEE Transactions on Medical Robotics and Bionics*, 1(2):65–76.

Hart, P. E., Nilsson, N. J., and Raphael, B. (1968). A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2):100–107.

Hsu, D., Latombe, J.-C., and Motwani, R. (1997). Path planning in expansive configuration spaces. In *Proceedings of International Conference on Robotics and Automation*, volume 3, pages 2719–2726 vol.3.

Jang, D., Jeong, J., Song, H., and Chung, S. K. (2019). Targeted drug delivery technology using untethered microrobots: a review. *Journal of Micromechanics and Microengineering*, 29(5):053002.

Janson, L., Ichter, B., and Pavone, M. (2018). Deterministic sampling-based motion planning: Optimality, complexity, and performance. *The International Journal of Robotics Research*, 37(1):46–61.

Janson, L., Schmerling, E., Clark, A., and Pavone, M. (2015). Fast marching tree: A fast marching sampling-based method for optimal motion planning in many dimensions. *The International Journal of Robotics Research*, 34(7):883–921. PMID: 27003958.

Jiang, C., Hu, Z., Mourelatos, Z. P., Gorsich, D., Jayakumar, P., Fu, Y., and Majcher, M. (2022). R2-rrt*: Reliability-based robust mission planning of off-road autonomous ground vehicle under uncertain terrain environment. *IEEE Transactions on Automation Science and Engineering*, 19(2):1030–1046.

Kamon, I., Rimon, E. D., and Rivlin, E. (1998). Tangentbug: A range-sensor-based navigation algorithm. *The International Journal of Robotics Research*, 17:934 – 953.

Kamon, I. and Rivlin, E. (1997). Sensory-based motion planning with global proofs. *IEEE Transactions on Robotics and Automation*, 13(6):814–822.

Karaman, S. and Frazzoli, E. (2011). Sampling-based algorithms for optimal motion planning. *The International Journal of Robotics Research*, 30(7):846–894.

Kavraki, L., Svestka, P., Latombe, J.-C., and Overmars, M. (1996). Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Transactions on Robotics and Automation*, 12(4):566–580.

Khatib, O. (1986). Real time obstacle avoidance for manipulators and mobile robots. *The International Journal of Robotics Research*, 5(1):90–99.

Kim, J.-O. and Khosla, P. (1992). Real-time obstacle avoidance using harmonic potential functions. *IEEE Transactions on Robotics and Automation*, 8(3):338–349.

Koditchek, D. E. and Rimon, E. D. (1990). Robot Navigation Functions on Manifolds with Boundary. *ADVANCES IN APPLIED MATHEMATICS*, 11:412–442.

Kumar, H., Paternain, S., and Ribeiro, A. (2020). Navigation of a quadratic potential with star obstacles. In *2020 American Control Conference (ACC)*, pages 2043–2048.

Kumar, H., Paternain, S., and Ribeiro, A. (2022). Navigation of a quadratic potential with ellipsoidal obstacles. *Automatica*, 146:110643.

Latombe, J. (2012). *Robot Motion Planning*. The Springer International Series in Engineering and Computer Science. Springer US.

Laumond, J.-P. (1987). Obstacle growing in a nonpolygonal world. *Information Processing Letters*, 25(1):41–50.

LaValle, S. M. (1998). Rapidly-exploring random trees : a new tool for path planning. *The annual research report*.

LaValle, S. M. (2006). *Planning Algorithms*. Cambridge University Press.

Lionis, G., Papageorgiou, X., and Kyriakopoulos, K. J. (2007). Locally computable navigation functions for sphere worlds. In *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pages 1998–2003.

Lionis, G., Papageorgiou, X., and Kyriakopoulos, K. J. (2008). Towards locally computable polynomial navigation functions for convex obstacle workspaces. In *2008 IEEE International Conference on Robotics and Automation*, pages 3725–3730.

Liu, Y.-H. and Arimoto, S. (1992). Path planning using a tangent graph for mobile robots among polygonal and curved obstacles: Communication. *The International Journal of Robotics Research*, 11(4):376–382.

Loizou, S. and Kyriakopoulos, K. (2006). Multirobot navigation functions i. *Lecture Notes in Control and Information Sciences*, 337:171–207.

Loizou, S. G. (2011a). Closed form navigation functions based on harmonic potentials. In *2011 50th IEEE Conference on Decision and Control and European Control Conference*, pages 6361–6366.

Loizou, S. G. (2011b). The navigation transformation: Point worlds, time abstractions and towards tuning-free navigation. In *2011 19th Mediterranean Conference on Control Automation (MED)*, pages 303–308.

Loizou, S. G. (2012). Navigation functions in topologically complex 3-d workspaces. In *2012 American Control Conference (ACC)*, pages 4861–4866.

Loizou, S. G. (2017). The navigation transformation. *IEEE Transactions on Robotics*, 33(6):1516–1523.

Loizou, S. G. and Kyriakopoulos, K. (2002). Closed loop navigation for multiple holonomic vehicles. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 3, pages 2861–2866 vol.3.

Loizou, S. G. and Rimon, E. D. (2021). Correct-by-construction navigation functions with application to sensor based robot navigation.

Lumelsky, V. and Skewis, T. (1990). Incorporating range sensing in the robot navigation function. *IEEE Transactions on Systems, Man, and Cybernetics*, 20(5):1058–1069.

Lumelsky, V. and Stepanov, A. (1986). Dynamic path planning for a mobile automaton with limited information on the environment. *IEEE Transactions on Automatic Control*, 31(11):1058–1063.

Lyes, S. and Soulaimane, B. (2024). Real-time sensor-based feedback control for obstacle avoidance in unknown environments. In *2024 American Control Conference (ACC)*, pages ?–?

Meyer, C. D. (2000). *Matrix Analysis and Applied Linear Algebra*. Society for Industrial and Applied Mathematics, USA.

Nagumo, M. (1942). Über die lage der integralkurven gewöhnlicher differentialgleichungen. *Proceedings of the Physico-Mathematical Society of Japan. 3rd Series*, 24:551–559.

Ng, J. and Bräunl, T. (2007). Performance comparison of bug navigation algorithms. *Journal of Intelligent and Robotic Systems*, 50:73–84.

Nilsson, N. J. (1969). A mobile automaton: An application of artificial intelligence techniques. In *International Joint Conference on Artificial Intelligence*.

Noh, D., Lim, H., Eoh, G., Choi, D., Choi, J., Lim, H., Baek, S., and Myung, H. (2024). Cloi-mapper: Consistent, lightweight, robust, and incremental mapper with embedded systems for commercial robot services. *IEEE Robotics and Automation Letters*, 9(9):7541–7548.

Paternain, S. and Ribeiro, A. (2020). Stochastic artificial potentials for online safe navigation. *IEEE Transactions on Automatic Control*, 65(5):1985–2000.

Phillips, J., Bedrossian, N., and Kavraki, L. (2004). Guided expansive spaces trees: a search strategy for motion- and cost-constrained state spaces. In *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004*, volume 4, pages 3968–3973 Vol.4.

Pimenta, L. C. A., Michael, N., Mesquita, R. C., Pereira, G. A. S., and Kumar, V. (2008). Control of swarms based on hydrodynamic models. In *2008 IEEE International Conference on Robotics and Automation*, pages 1948–1953.

Plaku, E., Bekris, K., Chen, B., Ladd, A., and Kavraki, L. (2005). Sampling-based roadmap of trees for parallel motion planning. *IEEE Transactions on Robotics*, 21(4):597–608.

Poveda, J. I., Benosman, M., Teel, A. R., and Sanfelice, R. G. (2018). A hybrid adaptive feedback law for robust obstacle avoidance and coordination in multiple vehicle systems. In *2018 Annual American Control Conference (ACC)*, pages 616–621.

Prajna, S. (2003). Barrier certificates for nonlinear model validation. In *42nd IEEE International Conference on Decision and Control (IEEE Cat. No.03CH37475)*, volume 3, pages 2884–2889 Vol.3.

Prajna, S. and Jadbabaie, A. (2004). Safety verification of hybrid systems using barrier certificates. In *HSCC*.

Reis, M. F., Aguiar, A. P., and Tabuada, P. (2021). Control barrier function-based quadratic programs introduce undesirable asymptotically stable equilibria. *IEEE Control Systems Letters*, 5(2):731–736.

Rimon, E. D. and Koditchek, D. E. (1992). Exact Robot Navigation Using Artificial Potential Functions. *IEEE Transactions on Robotics and Automation*, 8(5):501–518.

Rohnert, H. (1986). Shortest paths in the plane with convex polygonal obstacles. *Information Processing Letters*, 23(2):71–76.

S. Paternain, D. E. K. and Ribeiro, A. (2018). Navigation Functions for Convex Potentials in a Space With Convex Obstacles. *IEEE Transactions on Robotics and Automation*, 63(9):2944–2959.

Sanfelice, R., Messina, M., Emre Tuna, S., and Teel, A. (2006). Robust hybrid controllers for continuous-time systems with applications to obstacle avoidance and regulation to disconnected set of points. In *2006 American Control Conference*, page pp. 3352–3357.

Savkin, A. V. and Huang, H. (2022). Navigation of a uav network for optimal surveillance of a group of ground targets moving along a road. *IEEE Transactions on Intelligent Transportation Systems*, 23(7):9281–9285.

Sawant, M., Berkane, S., Polushin, I., and Tayebi, A. (2023). Hybrid feedback for autonomous navigation in planar environments with convex obstacles. *IEEE Transactions on Automatic Control*, 68(12):7342–7357.

Sawant, M., Polushin, I., and Tayebi, A. (2024a). Hybrid feedback control design for non-convex obstacle avoidance. *IEEE Transactions on Automatic Control*, pages 1–16.

Sawant, M., Polushin, I., and Tayebi, A. (2024b). Hybrid feedback control design for non-convex obstacle avoidance. *arXiv:2304.10598*.

Sawant, M., Polushin, I., and Tayebi, A. (2024c). Hybrid feedback for three-dimensional convex obstacle avoidance. *arXiv:2403.11279*.

Tan, X. and Dimarogonas, D. V. (2024). On the undesired equilibria induced by control barrier function based quadratic programs. *Automatica*, 159:111359.

Vasilopoulos, V. G. and Koditschek, D. E. (2018). Reactive Navigation in Partially Known Non-Convex Environments. In *13th International Workshop on the Algorithmic Foundations of Robotics (WAFR)*.

Vasilopoulos, V. G., Pavlakos, G., Bowman, S. L., Caporale, J. D., Daniilidis, K., Pappas, G. J., and Koditschek, D. E. (2021). Reactive Semantic Planning in Unexplored Semantic Environments Using Deep Perceptual Feedback. *IEEE Robotics and Automation Letters (RA-L)*, 5(3):4455–4462.

Vasilopoulos, V. G., Pavlakos, G., Schmeckpeper, K., Daniilidis, K., and Koditschek, D. E. (2020). Reactive navigation in partially familiar planar environments using semantic perceptual feedback. *ArXiv*, abs/2002.08946.

Vidal, E., Hernández, J. D., Istenič, K., and Carreras, M. (2017). Online view planning for inspecting unexplored underwater structures. *IEEE Robotics and Automation Letters*, 2(3):1436–1443.

Vlantis, P., Vrohidis, C., Bechlioulis, C. P., and Kyriakopoulos, K. J. (2018). Robot navigation in complex workspaces using harmonic maps. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1726–1731.

Wieland, P. and Allgöwer, F. (2007). Constructive safety using control barrier functions. *IFAC Proceedings Volumes*, 40(12):462–467. 7th IFAC Symposium on Nonlinear Control Systems.

Yamazaki, K., Ueda, R., Nozawa, S., Kojima, M., Okada, K., Matsumoto, K., Ishikawa, M., Shimoyama, I., and Inaba, M. (2012). Home-assistant robot for an aging society. *Proceedings of the IEEE*, 100(8):2429–2441.

İşleyen, A., van de Wouw, N., and Arslan, O. (2022). From low to high order motion planners: Safe robot navigation using motion prediction and reference governor. *IEEE Robotics and Automation Letters*, 7(4):9715–9722.

# Appendix A

# Proofs of Chapter 3

## A.1 Proof of Lemma 3.1

Minimizing the angle $\angle(x_d - x, v_i)$ is equivalent to minimizing the cost function $g(v_i) = 1 - V_d^\top \frac{v_i}{\|v_i\|}$ with $V_d = (x_d - x)/\|x_d - x\|$ under the constraint $\Gamma(v_i) = \frac{v_i^\top V_{ci}}{\|v_i\|} - \cos(\theta_i) = 0$ with $V_{ci} = (c_i - x)/\|c_i - x\|$. Define the Lagrangian associated to the optimization problem (3.16) by $L_\lambda(v_i) = g(v_i) - \lambda\Gamma(v_i)$ where $\lambda$ is the Lagrange multiplier. The optimum is the solution of

$$\nabla_{v_i} L_\lambda(v_i) = 0, \nabla_\lambda L_\lambda(v_i) = 0,$$

which gives

$$\pi^\perp(v_i)(V_d + \lambda V_{ci}) = 0, \quad \frac{v_i^\top V_{ci}}{\|v_i\|} - \cos(\theta_i) = 0. \tag{A.1}$$

From the first equation, one has $v_i = \alpha(V_d + \lambda V_{ci})$ for some $\alpha \in \mathbb{R}$. Substituting this into the second equation, one gets

$$\alpha(\cos(\beta_i) + \lambda) = \cos(\theta_i)\|\alpha(V_d + \lambda V_{ci})\|. \tag{A.2}$$

Squaring (A.2) and substituting $\|\alpha(V_d + \lambda V_{ci})\|^2 = \alpha^2(\lambda^2 + 2\lambda\cos(\beta_i) + 1)$, one can solve for $\lambda$

$$\lambda_{1,2} = -\frac{\sin(\theta_i \pm \beta_i)}{\sin(\theta_i)}. \tag{A.3}$$

Consequently, one can obtain $v_i^1$ and $v_i^2$ as follows:

$$v_i^{1,2} = \pm|\alpha| \left( V_d - \frac{\sin(\theta_i \pm \beta_i)}{\sin(\theta_i)} V_{ci} \right). \tag{A.4}$$

The value of $g$ at the two solutions is as follows:

$$g(v_i^1) = 1 + \cos(\theta_i + \beta_i), \; g(v_i^2) = 1 - \cos(\theta_i - \beta_i),$$

and $g(v_i^1) - g(v_i^2) = 2\cos(\theta_i)\cos(\beta_i) \geq 0$ which implies that

$$\mathcal{U}(x) = \{\bar{\alpha}(V_d - \sin^{-1}(\theta_i)\sin(\theta_i - \beta_i)V_{ci})|\ \bar{\alpha} \geq 0\}. \tag{A.5}$$

When $x \in \mathcal{S}(x_d, c_i)$, $u_d(x) \in \mathcal{V}(c_i - x, \theta_i)$ which implies that $\theta_i = \beta_i$, and $u_d(x) \in \mathcal{U}$. Therefore, for all $x \in \mathcal{S}(x_d, c_i)$, $u(x) \in \mathcal{U}$ implies that $u(x) = \bar{\alpha}V_d$, and if in addition $u(x) = u_d(x)$, then $\bar{\alpha} = \gamma\|x_d - x\|$. One can conclude that the solution is unique and is given by

$$u(x) = \gamma\|x_d - x\|\left(V_d - \frac{\sin(\theta_i - \beta_i)}{\sin(\theta_i)}V_{ci}\right) = \xi(u_d(x), x, i),$$

where the last equation is obtained after some straightforward manipulations.

## A.2 Proof of Lemma 3.2

Let $x(0) \in \mathcal{F} \setminus \mathcal{L}_d(x_d, c_i)$. Then, one has two situations. First, when $x(0) \in \mathcal{VI}$, the trajectory $x(t)$ is a line segment which is the closest path. Now, when $x(0) \in \mathcal{D}(x_d, c_i)$, there are two types of possible trajectories: trajectories inside the enclosing cone $\mathcal{C}^{\leqq}_{\mathcal{F}}(x(0), c_i - x(0), \theta_i)$ and trajectories outside this cone. One can show that the trajectory generated by the closed-loop system (3.7)-(3.20), on the enclosing cone $\mathcal{C}^{\equiv}_{\mathcal{F}}(x(0), c_i - x(0), \theta_i)$, has minimum length. For the first type of trajectory, one only considers the ones between the line segment $\mathcal{L}_s(x(0), x_d)$ and the closest tangent to it (blue segment in Fig. A.1) among the cone enclosing the obstacle (the red trajectory in Fig. A.1 is an example). All these trajectories will merge with our trajectory, which is on the closest tangent (as shown in Lemma 3.1), at the intersection point of the tangent with the obstacle. Since, before the intersection point, our trajectory is a line segment, one can conclude that it is the shortest path. The best that can be achieved outside the cone for a smooth trajectory is a dilated version of our trajectory (larger radius of curvature) which is longer than ours (black path in Fig. A.1).

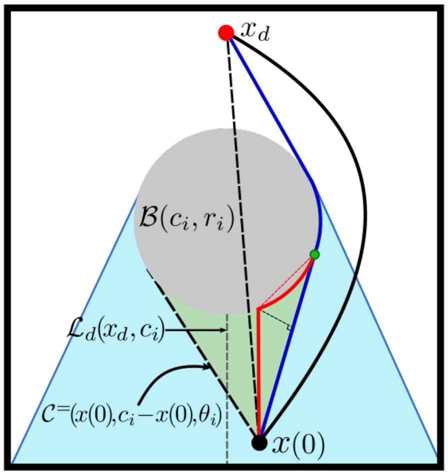

$x_d$
$\mathcal{B}(c_i, r_i)$
$\mathcal{L}_d(x_d, c_i)$
$\mathcal{C}^{=}(x(0), c_i - x(0), \theta_i)$
$x(0)$

Figure A.1: Shortest path in a single-obstacle sphere world.

## A.3 Proof of Lemma 3.3

First, we prove that the closed-loop system admits a unique solution. The control is Lipschitz on $\mathcal{VI}$ since $u(x) = u_d(x)$ is continuously differentiable. When $x \in \mathcal{BL}$, for simplicity, $\sin(\theta_{\iota_x(p)}(q) - \beta_{\iota_x(p)}(u_{p-1}(x), q))$ is denoted by $s^s_{\iota_x(p)}(q)$ and $\frac{\sin(\beta_{\iota_x(p)}(u_{p-1}(x),q))}{\sin(\theta_{\iota_x(p)}(q))}$ by $s^d_{\iota_x(p)}(q)$, where $p \in \{1, \dots, h(x)\}$. After manipulation, the control (3.22) can be expressed as $u(x) = u_d(x) - \gamma\|x - x_d\| \sum\limits_{p=1}^{h(x)} \prod\limits_{k=1}^{p-1} s^d_{\iota_x(k)}(x) \frac{s^s_{\iota_x(p)}(x)}{r_{\iota_x(p)}}(c_{\iota_x(p)} - x)$, which is shown to be one-sided Lipschitz as follows:

$$
\begin{aligned}
(u(x) - u(y))^\top (x - y) = -\gamma\|x - y\|^2 & \\
& - \gamma\|x_d - x\| \sum_{p=1}^{h(x)} \prod_{k=1}^{p-1} s^d_{\iota_x(k)}(x) \frac{s^s_{\iota_x(p)}(x)}{r_{\iota_x(p)}} (c_{\iota_x(p)} - x)^\top (x - y) \\
& + \gamma\|x_d - y\| \sum_{p=1}^{h(y)} \prod_{k=1}^{p-1} s^d_{\iota_x(k)}(y) \frac{s^s_{\iota_x(p)}(y)}{r_{\iota_x(p)}} (c_{\iota_x(p)} - y)^\top (x - y), \\
\leq -\gamma\|x - y\|^2 & \\
& \gamma + \|x_d - x\|\|x - y\| \sum_{p=1}^{h(x)} \prod_{k=1}^{p-1} s^d_{\iota_x(k)}(x) \frac{s^s_{\iota_x(p)}(x)}{r_{\iota_x(p)}} \|c_{\iota_x(p)} - x\| \\
& + \gamma\|x_d - y\|\|x - y\| \sum_{p=1}^{h(y)} \prod_{k=1}^{p-1} s^d_{\iota_x(k)}(y) \frac{s^s_{\iota_x(p)}(y)}{r_{\iota_x(p)}} \|c_{\iota_x(p)} - y\|.
\end{aligned}
$$

Note that $\forall x \in \mathcal{BL}$ and $\forall p \in \{1, \dots, h(x)\}$, $0 \leq s^d_{\iota_x(p)}(x) \leq 1$, $0 \leq s^s_{\iota_x(p)}(x) \leq 1$, $\|c_{\iota_x(p)} - x\| \leq 2r_0 - r_{\iota_x(p)}$ and $\|x_d - x\| \leq 2r_0$, which implies that there exists $M > 0$ such that $\|x_d - x\| \sum_{p=1}^{h(x)} \frac{\|c_{\iota_x(p)} - x\|}{r_{\iota_x(p)}} \leq M\|x - y\|$. Therefore,

$$
\begin{aligned}
(u(x) - u(y))^\top (x - y) &\leq -\gamma\|x - y\|^2 + \gamma M_1\|x - y\|^2 + \gamma M_2\|x - y\|^2 \\
&\leq \gamma(-1 + M_1 + M_2)\|x - y\|^2 \\
&\leq L\|x - y\|^2.
\end{aligned}
$$

One can take $L = \gamma(-1 + M_1 + M_2)$ where $M_1 > 0$, $M_2 > 0$ and $M_1 + M_2 > 1$. The control (3.22) is one-sided Lipschitz (Cortes, 2008) when $x \in \mathcal{BL}$, and is Lipschitz when $x \in \mathcal{VI}$. Thus, according to (Cortes, 2008, Proposition 2), the closed-loop system (3.7)-(3.22) has a unique solution for all $x(0) \in \mathcal{F}$. Now, we prove forward invariance using Nagumo's theorem. We only need to verify Nagumo's condition at the free space boundary as it is trivially met when $x \in \mathring{\mathcal{F}}$ where $\mathrm{T}_{\mathcal{F}}(x) = \mathbb{R}^n$. Since the free space is a sphere world, the tangent cone on its boundary is the half-space $\mathcal{C}^{\leq}_{\mathbb{R}^n}(x, -x, \frac{\pi}{2})$ when $x \in \partial\mathcal{W}$ and $\mathcal{C}^{\geq}_{\mathbb{R}^n}(x, c_i - x, \frac{\pi}{2})$ when $x \in \partial\mathcal{O}_i$ (see Fig. 3.5). Let us consider an obstacle $\mathcal{O}_i$ and verify Nagumo's condition in three regions of the free space.
In the first region, When $x \in \partial\mathcal{W}$, $\mathrm{T}_{\mathcal{F}}(x) = \mathcal{C}^{\leq}_{\mathbb{R}^n}(x, -x, \frac{\pi}{2})$ and two sub-regions must be considered.

- $x \in \partial\mathcal{W} \cap \partial\mathcal{BL}$ (brown arc in Fig. 3.5): Since $u(x) \in \mathcal{C}^{=}_{\mathcal{F}}(x, c_i - x, \theta_i)$ and $\mathcal{C}^{=}_{\mathcal{F}}(x, c_i - x, \theta) \subseteq \mathcal{C}^{\leq}_{\mathbb{R}^n}(x, -x, \frac{\pi}{2})$, one concludes that $u(x) \in \mathrm{T}_{\mathcal{F}}(x)$.

- $x \in \partial\mathcal{W} \setminus \partial\mathcal{BL}$ (grey arc in Fig. 3.5): Since $u_d(x) \in \mathcal{F}$ and $\mathcal{F} \subseteq \mathcal{C}^{\leq}_{\mathbb{R}^n}(x, -x, \frac{\pi}{2})$, one concludes that $u(x) = u_d(x) \in \mathrm{T}_{\mathcal{F}}(x)$.

In the second region, $x \in \partial\mathcal{O}_i \cap \mathcal{AR}_i$ (green arc in Fig. 3.5) and $\mathrm{T}_{\mathcal{F}}(x) = \mathcal{C}^{\geq}_{\mathbb{R}^n}(x, c_i - x, \frac{\pi}{2})$. Since $u(x) \in \mathcal{C}^{=}_{\mathbb{R}^n}(x, c_i - x, \frac{\pi}{2}) \subset \mathcal{C}^{\geq}_{\mathbb{R}^n}(x, c_i - x, \frac{\pi}{2})$, one concludes that $u(x) \in \mathrm{T}_{\mathcal{F}}(x)$. Finally, in the last region, $x \in \partial\mathcal{O}_i \setminus \mathcal{AR}_i$ (blue arc in Fig. 3.5) and $\mathrm{T}_{\mathcal{F}}(x) = \mathcal{C}^{\geq}_{\mathbb{R}^n}(x, c_i - x, \frac{\pi}{2})$. Since $x \notin \mathcal{AR}_i$, $\forall p \in \{0, \cdots, h(x)\}$, obstacle $\mathcal{O}_i$ is not selected in the successive projections ($\iota_x(p) \neq i$) and $u_p(x) \notin \mathcal{C}^{\leq}_{\mathbb{R}^n}(x, c_i - x, \frac{\pi}{2})$. Therefore, $u(x)$ must be in the complement of the enclosing cone to the obstacle $\mathcal{O}_i$. Thus, one can conclude that $u(x) \in \mathcal{C}^{\geq}_{\mathbb{R}^n}(x, c_i - x, \frac{\pi}{2}) = \mathrm{T}_{\mathcal{F}}(x)$. Since $\forall x \in \mathcal{F}$, $u(x) \in \mathrm{T}_{\mathcal{F}}(x)$ and the solution of the closed-loop system (3.7)-(3.22) is unique, it follows that the free space $\mathcal{F}$ is positively invariant and the closed-loop system (3.7)-(3.22) is safe.

## A.4 Proof of Lemma 3.5

Let $\mathcal{L}^e_i$ be the central half-line associated to obstacle $\mathcal{O}_i$, and let $y \in \mathcal{L}^e_i \setminus \mathcal{O}_i$. Since the control is tangent to the obstacle, for any $x \in \mathcal{AR}_i \setminus \mathring{\mathcal{H}}(y, c_i)$ the control, at a step $p$, can never point inside the hat $\mathcal{H}(y, c_i)$, *i.e.*, there is no position $q \in \mathcal{AR}_i \cap \mathring{\mathcal{H}}(y, c_i)$ such that $\angle(q - x, u_p(x)) = 0$, where $p = \iota_x^{-1}(i)$. Assume that $\mathcal{M}_i \neq \varnothing$ and consider an obstacle $k \in \mathcal{M}_i$ such that $k = \kappa_i^{-1}(1)$, $c_k \in \mathring{\mathcal{H}}(y, c_i)$, and $x^*_{k,i} = y$. Assume that $\mathring{\mathcal{H}}(x^*_{k,i}, c_i) \cap \mathcal{O}_l = \varnothing$ for all $l \in \mathbb{I} \setminus \{i, k\}$, which ensures that no other obstacle interferes and changes the direction of the control $u_p$ towards the hat $\mathring{\mathcal{H}}(x^*_{k,i}, c_i)$. Consequently, there is no $x \in \mathcal{AR}_k \cap \mathcal{AR}_i$ such that $\angle(c_k - x, u_p) = \beta(c_k - x, u_p) = 0$ where $p = \iota_x^{-1}(i)$. Thus, obstacle $k$ does not generate a set of undesired equilibria $\mathcal{L}_k$ (see Fig. A.2). Following the same reasoning, one can show that obstacle $j = \kappa_i^{-1}(2)$ will not generate a set of undesired equilibria by considering the obstacles $\mathcal{O}_i$ and $\mathcal{O}_k$ as single obstacle whose hat is the union $\mathring{\mathcal{H}}(x^*_{j,i}, c_i) \cup \mathring{\mathcal{H}}(x^*_{j,i}, c_k)$ and $\mathcal{L}^e_i$ as its single central half-line since obstacle $k = \kappa_i^{-1}(1)$ is free of undesired equilibria. These considerations reduce to the first case where if $c_j \in \mathring{\mathcal{H}}(x^*_{j,i}, c_i) \cup \mathring{\mathcal{H}}(x^*_{j,i}, c_k)$ and $(\mathring{\mathcal{H}}(x^*_{j,i}, c_i) \cup \mathring{\mathcal{H}}(x^*_{j,i}, c_k)) \cap \mathcal{O}_l = \varnothing$ for all $l \in \mathbb{I} \setminus \{i, k\}$, obstacle $j$ does not generate undesired equilibria. The same can be applied up to obstacle $\kappa_i^{-1}(p)$ by considering the union of the hat of obstacle $i$ and obstacles $\mathcal{M}_i^{p-1}$, and the central half-line $\mathcal{L}^e_i$. Therefore, the obstacles of indices in the set $\mathcal{M}_i^p$ are free of undesired equilibria if, for all $j \in \mathcal{M}_i^p$ with $p \in \{1, \ldots, N_i\}$, 1) $c_j \in \cup_{l \in \mathcal{M}_i^{p-1}} \mathring{\mathcal{H}}(x^*_{j,i}, c_l)$ and 2) $\cup_{l \in \mathcal{M}_i^{p-1}} \mathring{\mathcal{H}}(x^*_{j,i}, c_l) \cap \mathcal{O}_z = \varnothing$ for all $z \in \mathbb{I} \setminus (\mathcal{M}_i^j \cup \{j\})$. If, in addition, $p = N_i$, or obstacle $\kappa_i^{-1}(p+1)$, with $p < N_i = \mathbf{card}(M_i)$, does not satisfy conditions 1) and 2), we say that $\bar{N}_i = p$ is the order of the set $\mathcal{M}_i$ and the number of obstacles free of undesired equilibria with indices grouped in the set $\mathcal{M}_i^{\bar{N}_i}$, which concludes the proof.

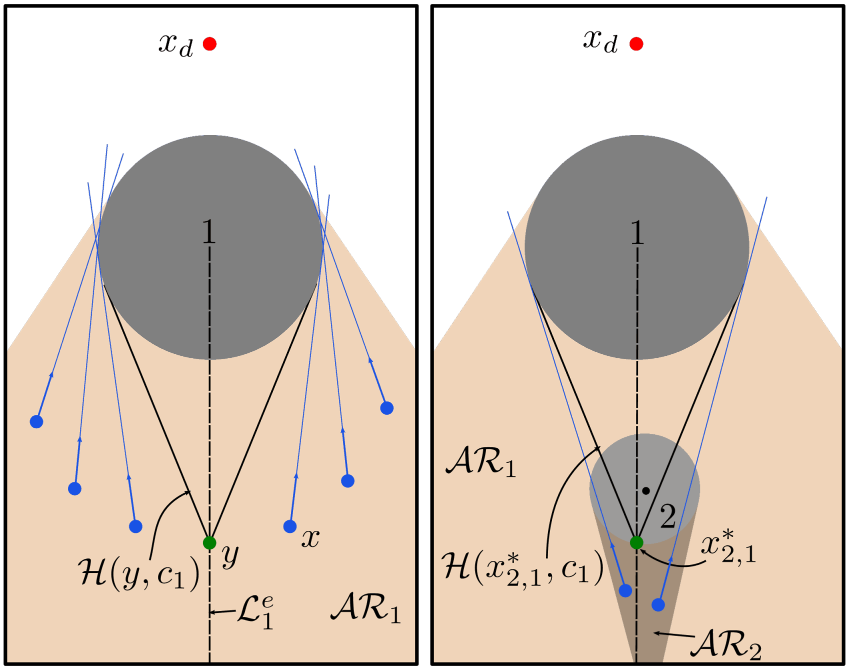


Figure A.2: The left figure shows that for all position $x \in \mathcal{AR}_1 \setminus \mathring{\mathcal{H}}(y, c_1)$, the control cannot point inside $\mathcal{H}(y, c_1)$. In the right figure, obstacle 2 is placed such that $c_2 \in \mathring{\mathcal{H}}(y, c_1)$, the intermediary control $u_1$ cannot point inside $\mathcal{H}(y, c_1)$ and $y = x^*_{2,1}$. Then, the intermediary control $u_1$ cannot point into the center $c_2$ at any position $x \in \mathcal{AR}_2$ which implies that $\mathcal{L}_2$ is an empty set.

## A.5 Proof of Theorem 3.1

Item i) and item ii) follow directly from lemma 3.3 and lemma 3.4 respectively. Now let us prove item iii). Consider the set of undesired equilibria $\mathcal{L}_i$, $i \in \mathcal{Z}$. Define the tube surrounding $\mathcal{L}_i$ inside the free space $\mathcal{TU}_i := \{x \in \mathcal{F} | d(x, \mathcal{L}_i) \leq e_i\}$ where $e_i$ is small such that $\mathcal{TU}_i \subset \mathcal{AR}_i$, and $e_i < r_i$. Let $V(x) = \frac{1}{2} d^2(\mathcal{L}_i, x) = \frac{1}{2}(x - c_i)^\top \pi^\perp(\bar{v}_i)(x - c_i)$ where $\bar{v}_i = (\bar{x}_i - c_i)/\|\bar{x}_i - c_i\|$, $\bar{x}_i \in \mathcal{L}_i$, $V(x) = 0$ for all $x \in \mathcal{L}_i$, and $V(x) > 0$ for all $x \in \mathcal{TU}_i \setminus \mathcal{L}_i$. Let $U := \mathcal{TU}_i \cap \mathcal{P}_{\leq}(\bar{x}_i, \bar{v}_i) \setminus \mathcal{L}_i$ where $\bar{x}_i \in \mathcal{L}_i$ is such that $U \cap \mathcal{L}_k = \varnothing$ for all $k \in \mathcal{Z} \setminus \{i\}$, and $U \subset \mathcal{AR}_i^h$ with $\mathcal{AR}_i^h := \{q \in \mathcal{AR}_i | \iota_q^{-1}(i) = h(x)\}$. Note that $V(x) > 0$ for all $x \in \mathcal{U}$. The time-derivative of $V(x)$ is given by $\dot{V}(x) = \frac{\partial V(x)}{\partial x}^\top \dot{x} = (x - c_i)^\top \pi^\perp(\bar{v}_i) u(x)$. Since $e_i < r_i$ and for all $x \in U$, $u(x)$ lies on the cone enclosing obstacle $\mathcal{O}_i \subset \mathcal{P}_{\leq}(\bar{x}_i, \bar{v}_i)$, $0 < \angle(\bar{v}_i, x - c_i) < \pi/2$ and $\pi/2 < \angle(\bar{v}_i, u(x)) < \pi$. Therefore, $\dot{V}(x) > 0$ for all $x \in U$. As $U$ is a compact set, $V(x)$ is increasing on $U$, and $V(x) = 0$ on $\mathcal{L}_i$ (the axis of the tube), $x(t)$ must leave the set $U$. The set $U$ is bounded on top by obstacle $i$, its lateral boundary is the surface of the tube, and is bounded from the bottom by the hyperplane $\mathcal{P}_=(\bar{x}_i, \bar{v}_i)$. Due to the safety of the system, as per Lemma 3.3, $x(t)$ can not leave $U$ from the upper boundary (the boundary of obstacle $i$), and since $\pi/2 < \angle(\bar{v}_i, u(x)) < \pi$, $x(t)$ can only leave $U$ from the surface of the tube. Now, let us prove item iv). Since $x_d \in \mathcal{VI}$, there exists $r_d > 0$ such that $\mathcal{B}(x_d, r_d) \subset \mathcal{VI}$. As the closed-loop system (3.7)-(3.22) reduces to $\dot{x} = -\gamma(x - x_d)$

on the visible set $\mathcal{VI}$, the equilibrium $x = x_d$ is locally exponentially stable. Finally, let us prove item v). Consider a trajectory starting from $x(0) \in \mathcal{VL}$ generated by the closed-loop system (3.7)-(3.22). Since the control on the visible set $\mathcal{VL}$ is the nominal one $u_d(x) = \gamma(x_d - x)$, the generated trajectory is the line segment connecting $x(0)$ to $x_d$, which has the shortest length. Now consider a trajectory with initial condition $x(0) \in \mathcal{BL}$, generated by the closed-loop system (3.7)-(3.22). The velocity of a vehicle moving along the considered trajectory at an instant $t \in [0, \infty)$ is given by $\dot{x}(t) = u_{h(x(t))}(x(t))$ where $h(x(t)) = \mathbf{card}(\mathcal{I}(x(t)))$. Since the virtual destination at position $x(t)$ is the point given by $P(x(t)) = x(t) + u_{h(x(t))}(x(t))$, then, the direction from $x(t)$ to the virtual destination is the vehicle's velocity $P(x(t)) - x(t) = u_{h(x(t))}(x(t)) = \dot{x}(t)$. Therefore, one can conclude that for $x(0) \in \mathcal{BL}$, the tangent to the trajectory generated by the closed-loop system (3.7)-(3.22), at any position $x(t)$, points to the virtual destination $P(x(t))$.

## A.6 Proof of Lemma 3.6

Let $i \in \mathbb{L}$. Since $\mathbf{Cell}_i$ is bounded by line segments of undesired equilibria $(\cup_{k \in \mathcal{Z}} \mathcal{L}_k)$ and the free space boundary, $u(x)$ points inside the cell when $x \in \partial \mathbf{Cell}_i \cap \partial \mathcal{F}$, as per Lemma 3.3, and $u(x) = 0$ when $x \in \partial \mathbf{Cell}_i \cap (\cup_{k \in \mathcal{Z}} \mathcal{L}_k)$. Consequently, $\mathbf{Cell}_i$ is forward invariant for the closed-loop system (3.7)-(3.22).

## A.7 Proof of Lemma 3.7

Since the nests are invariant, as per Lemma 3.6, and all the undesired equilibria are contained inside the nests, it remains to show that if we start outside nests, we will never get back in. We begin with the special nest ($\mathbf{Nest}_0 = \cup_{i \in \mathcal{Z}} \mathcal{L}_i \setminus \cup_{j \in \mathbb{L}} \mathbf{Cell}_j$) formed by segments of undesired equilibria and show their repellency. These segments can be defined as $\mathcal{CL}_i := \{q \in \mathcal{L}_i | q \notin \cup_{k \in \mathbb{L}} \mathbf{Cell}_k\} \subset \mathbf{Nest}_0$ for $i \in \mathcal{Z}$.
Consider obstacle $i \in \mathcal{Z}$ and segment $\mathcal{CL}_i$ in the following three possible cases illustrated in (Fig. A.3):
**Case 1:** Consider the region $\mathcal{AR}_i^h := \left\{q \in \mathcal{AR}_i | \iota_q^{-1}(i) = h(q)\right\}$ where obstacle $i$ is the last on the list of projections. Define the tube $\mathcal{TU}_i := \{x \in \mathcal{F} | d(x, \mathcal{CL}_i) \leq e_i\}$ where $e_i$ is small enough to have $\mathcal{TU}_i \cap \mathcal{L}_j = \varnothing$ for all $j \in \mathcal{Z} \setminus \{i\}$ and $\mathcal{TU}_i \subset \mathcal{AR}_i$. Let $V(x) = 1 - \frac{(\bar{x}_i - c_i)^\top}{\|\bar{x}_i - c_i\|} \frac{(x - c_i)}{\|x - c_i\|}$ where $\bar{x}_i \in \mathcal{CL}_i \cap \mathcal{AR}_i^h$. Note that $V(\bar{x}_i) = 0$, and $V(x) > 0$ for all $x \in \mathcal{TU}_i \setminus \mathcal{CL}_i$. Define the set $U := (\mathcal{TU}_i \cap \mathcal{AR}_i^h) \setminus \mathcal{CL}_i$. The time-derivative of $V(x)$ is given by

$$\begin{aligned}
\dot{V}(x) &= \frac{\partial V(x)}{\partial x}^\top \dot{x}, \\
&= -\frac{(\bar{x}_i - c_i)^\top}{\|\bar{x}_i - c_i\|} J_x \left( \frac{(x - c_i)}{\|x - c_i\|} \right) u(x), \\
&= -K \bar{V}_{ci}^\top \pi^\perp(V_{ci}) \bar{\xi}_i,
\end{aligned}$$

where $K = \frac{\|u(x)\|}{\|x - c_i\|} > 0$, $V_{ci} = \frac{(c_i - x)}{\|c_i - x\|}$, $\bar{V}_{ci} = \frac{(\bar{x}_i - c_i)}{\|\bar{x}_i - c_i\|}$ and $\bar{\xi}_i = \frac{\sin(\theta_i) u_{h(x)-1}}{\sin(\beta_i) \|u_{h(x)-1}\|} - \frac{\sin(\theta_i - \beta_i)}{\sin(\beta_i)} V_{ci}$.

The segment $\mathcal{CL}_i$ divides the set $\mathcal{AR}_i^h$ into two symmetric regions, a left-side $\mathcal{P}_<(c_i, R_1\bar{V}_{ci}) \cap \mathcal{AR}_i^h$, and a right-side $\mathcal{P}_>(c_i, R_1\bar{V}_{ci}) \cap \mathcal{AR}_i^h$. On the right side, the control is on the right tangent to obstacle $i$, while on the left, the control is on the right tangent to obstacle $i$. Therefore, the control can not point inside the region enclosed by the vectors $\bar{V}_{ci}$, $V_{ci}$ (*i.e.*, $\forall x \in \mathcal{AR}_i^h \setminus \mathcal{CL}_i$, $u(x) \notin \{v \in \mathbb{R}^n | v = a\bar{V}_{ci} + bV_{ci},\ a > 0, b > 0\}$). Thus, for all $x \in U$, $\bar{V}_{ci}^\top \bar{\xi}_i = \cos(\sigma_i + \theta_i)$ where $0 < \sigma_i = \angle(\bar{V}_{ci}, V_{ci}) < \pi$ and $0 < \theta_i = \angle(V_{ci}, \bar{\xi}_i) \leq \frac{\pi}{2}$. Then,

$$\begin{aligned}\dot{V}(x) &= -K\left(\cos(\sigma_i + \theta_i) - \cos(\sigma_i)\cos(\theta_i)\right),\\ &= K\sin(\sigma_i)\sin(\theta_i).\end{aligned}$$

Therefore, $\dot{V}(\bar{x}_i) = 0$ and $\dot{V}(x) > 0$ for all $x \in U$.

**Case 2:** Consider the region $\mathcal{AR}_k^h$ where $k \in \mathcal{M}_i^{\bar{N}_i}$. Define the tube $\mathcal{TU}_i := \{x \in \mathcal{F} | d(x, \mathcal{CL}_i) \leq e_i\}$ where $e_i$ is small such that $\mathcal{TU}_i \cap \mathcal{L}_j = \varnothing$ for all $j \in \mathcal{Z} \setminus \{i\}$, and $\mathcal{TU}_i \subset \mathcal{AR}_k$. Let $V(x) = 1 - \frac{(\bar{x}_i - x_{k,i}^*)^\top}{\|\bar{x}_i - x_{k,i}^*\|}\frac{(x - x_{k,i}^*)}{\|x - x_{k,i}^*\|}$ where $\bar{x}_i \in \mathcal{CL}_i \cap \mathcal{AR}_k^h$ and $x_{k,i}^* = \arg \max_{q \in \mathcal{L}_i^e \cap \partial\mathcal{O}_k} ||c_i - q||$. Note that $V(\bar{x}_i) = 0$, and $V(x) > 0$ for all $x \in (\mathcal{TU}_i \cap \mathcal{AR}_k^h) \setminus \mathcal{CL}_i$. Define the set $U := (\mathcal{TU}_i \cap \mathcal{AR}_h^k) \setminus \mathcal{CL}_i$. The time-derivative of $V(x)$ is given by

$$\begin{aligned}\dot{V}(x) &= \frac{\partial V(x)}{\partial x}^\top \dot{x},\\ &= -\frac{(\bar{x}_i - x_{k,i}^*)^\top}{\|\bar{x}_i - x_{k,i}^*\|} J_x\left(\frac{(x - x_{k,i}^*)}{\|x - x_{k,i}^*\|}\right) u(x),\\ &= -K\bar{V}_{k,i}^\top \pi^\perp(V_{k,i})\bar{\xi}_k,\end{aligned}$$

where $K = \frac{\|u(x)\|}{\|x - x_{k,i}^*\|} > 0$, $V_{k,i} = \frac{(x_{k,i}^* - x)}{\|x_{k,i}^* - x\|}$, $\bar{V}_{k,i} = \frac{(\bar{x}_i - x_{k,i}^*)}{\|\bar{x}_i - x_{k,i}^*\|}$ and $\bar{\xi}_k = \frac{\sin(\theta_k)u_{h(x)-1}}{\sin(\beta_k)\|u_{h(x)-1}\|} - \frac{\sin(\theta_k - \beta_k)}{\sin(\beta_k)}V_{ck}$. Similar to the previous case, segment $\mathcal{CL}_i$ divides set $\mathcal{AR}_k^h$ into two regions, a left-side $\mathcal{P}_<(c_i, R_1\bar{V}_{k,i}) \cap \mathcal{AR}_k^h$ and a right-side $\mathcal{P}_>(c_i, R_1\bar{V}_{k,i}) \cap \mathcal{AR}_k^h$. On the right side, the control is on the right tangent to obstacle $k$, while on the left, the control is on the left tangent to obstacle $k$. Therefore, by considering $V_{k,i}$ instead of $V_{ck}$ where $V_{k,i} = a\bar{V}_{k,i} + bV_{ck}$, $a > 0$, $b > 0$, $0 < \sigma_{k,i} = \angle(\bar{V}_{k,i}, V_{k,i}) < \pi$, and $\theta_k < \theta_k^* = \angle(\bar{V}_{k,i}, \bar{\xi}_k) < \pi$, we can write $\bar{V}_{k,i}^\top \bar{\xi}_k = \cos(\sigma_{k,i} + \theta_k^*)$ for all $x \in U$. Then,

$$\begin{aligned}\dot{V}(x) &= -K\left(\cos(\sigma_{k,i} + \theta_k^*) - \cos(\sigma_{k,i})\cos(\theta_k^*)\right),\\ &= K\sin(\sigma_{k,i})\sin(\theta_k^*),\end{aligned}$$

Therefore, $\dot{V}(\bar{x}_i) = 0$ and $\dot{V}(x) > 0$ for all $x \in U$.

**Case 3:** Consider the region $\mathcal{AR}_k^h$ where $\mathcal{L}_i^e \cap \mathcal{O}_k = \varnothing$ and $\mathcal{CL}_i \cap \mathcal{AR}_k^h \neq \varnothing$. Let $\bar{\Omega}_i = R_b(c_i - \bar{x}_i)$ where $\bar{x}_i \in \mathcal{CL}_i \cap \mathcal{AR}_k^h$, $R_b = \left(\begin{smallmatrix} 0 & b \\ -b & 0 \end{smallmatrix}\right)$, and $b \in \{-1, 1\}$ is chosen such that $\bar{\Omega}_i^\top(c_k - \bar{x}_i) > 0$. Since $c_k \in \mathcal{P}_>(\bar{x}_i, \bar{\Omega}_i)$ and $\mathcal{L}_i^e \cap \mathcal{O}_k = \varnothing$, $\mathcal{O}_k \subset \mathcal{P}_>(\bar{x}_i, \bar{\Omega}_i)$. Define the tube $\mathcal{TU}_i := \{x \in \mathcal{F} | d(x, \mathcal{CL}_i) \leq e_i\}$ where $e_i$ is small such that $\mathcal{TU}_i \cap \mathcal{P}_>(\bar{x}_i, \bar{\Omega}_i) \cap \mathcal{L}_j = \varnothing$ for all $j \in \mathcal{Z} \setminus \{i\}$, and $\bar{\Omega}_i^\top u(x) > 0$ for all $x \in \mathcal{TU}_i \cap \mathcal{P}_>(\bar{x}_i, \bar{\Omega}_i)$. Let $V(x) = \bar{\Omega}_i^\top(x - \bar{x}_i)$

where $V(\bar{x}_i) = 0$ and $V(x) > 0$ for all $x \in \mathcal{P}_>(\bar{x}_i, \bar{\Omega}_i)$. Define the set $U := \mathcal{TU}_i \cap \mathcal{P}_>(\bar{x}_i, \bar{\Omega}_i) \cap \mathcal{AR}_k^h$.

$$\dot{V}(x) = \frac{\partial V(x)}{\partial x}^\top \dot{x} = \bar{\Omega}_i^\top u(x),$$

where $\dot{V}(x) > 0$ for all $x \in U$ and $\dot{V}(x) = 0$ for $x \in \mathcal{CL}_i$.
In the three treated cases, $U$ is compact, $V(x) = 0$ on the undesired equilibria $\mathcal{CL}_i$, and $V$ is increasing on $U$. Therefore, $x(t)$ must leave $U$.

Now, we will show that if we start outside a given nest $\mathbf{Nest}_k$, $k > 0$, we will never get inside any nest.

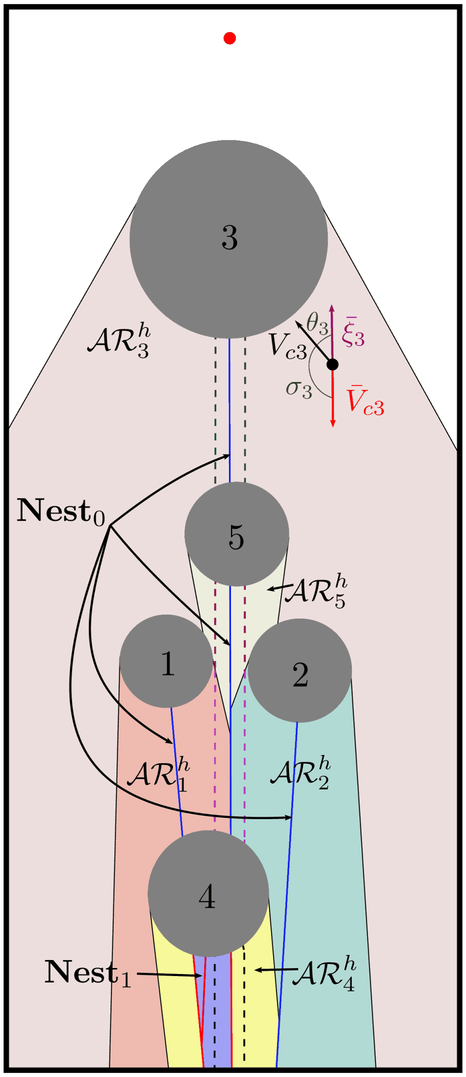


Figure A.3: Illustration of the nests (attraction regions of the undesired equilibria.)

Consider the nest $\mathbf{Nest}_k$, obstacle $j \in \mathbb{L}$, and the set of undesired equilibria $\mathcal{L}_i$ such that $k > 0$, $i \in \mathcal{R}_j$, and $\partial\mathbf{Nest}_k \cap \mathcal{L}_i \cap \mathcal{AR}_j^h \neq \varnothing$ (*i.e.*, segment (or segments) of $\mathcal{L}_i$ forms a portion of the boundary of the nest $\mathbf{Nest}_k$ when the last projection involves obstacle

$j$). Define the tube $\mathcal{TU}_i := \{x \in \mathcal{F} | d(x, \mathcal{L}_i \cap \mathcal{AR}_j^h) \leq e_i\}$ where $e_i$ is small such that $(\mathcal{TU}_i \setminus \mathbf{Nest}_k) \cap \mathcal{L}_p = \varnothing$ for all $p \in \mathcal{Z} \setminus \{i\}$, and $\mathcal{TU}_i \subset \mathcal{AR}_j$. This case amounts to case 2) with $U := (\mathcal{TU}_i \setminus \mathbf{Nest}_k) \cap \mathcal{AR}_j^h$. Since the nests are invariants, $\cup_{i \in \mathcal{Z}} \mathcal{L}_i \subset \cup_k \mathbf{Nest}_k$, and for all $x(0) \notin \cup_k \mathbf{Nest}_k$, $\lim_{t \to +\infty} x(t) \notin \mathbf{Nest}_k$, the set of nests $\cup_k \mathbf{Nest}_k$, is the attraction region of the undesired equilibria.

## A.8 Proof of Theorem 3.2

Item i) follow directly from Lemma 3.7. According to item v) of Theorem 3.1, all trajectories converging to $x_d$ are generated by a *quasi-optimal* obstacle avoidance maneuver, and item i) states that $x_d$ is attractive from all $x(0) \in \mathcal{F} \setminus \cup_k \mathbf{Nest}_k$, which proves item ii). Since Assumption 3.3 imposes that $\mathbb{L} = \varnothing$, no cell will be created, which implies that only the special nest exists. Therefore, $\cup_k \mathbf{Nest}_k = \mathbf{Nest}_0 = \cup_{i \in \mathcal{Z}} \mathcal{L}_i$ is the region of attraction of the undesired equilibria $\cup_{i \in \mathcal{Z}} \mathcal{L}_i$, as per Lemma 3.7, and has Lebesgue measure zero, which shows the almost global asymptotic stability of $x_d$.

## A.9 Proof of Lemma 3.8

Let us look for the equilibria of the closed-loop system (3.7)-(3.33) by setting $u(x) = 0$. Then, from the first equation of (3.33), the equilibrium point is $x_d$. From the second equation of (3.33), one gets $u_d(x) = \|u_d(x)\| \frac{\sin(\tilde{\theta} - \tilde{\beta})}{\sin(\tilde{\theta})} \frac{\tilde{c} - x}{\|\tilde{c} - x\|}$ which is true if and only if $\tilde{\beta} = 0$ ($i.e.$,$\angle(u_d, (\tilde{c} - x)) = 0$). As $\widetilde{\mathcal{BL}}$ is the union of the disjoints practical shadow regions, there exists a unique $i \in \mathbb{I}$ such that if $x \in \widetilde{\mathcal{BL}}$, $x \in \tilde{\mathcal{D}}(x_d, c_i, R)$, and since $\tilde{c}$ is the projection of $x$ onto obstacle $i$, $\frac{\tilde{c} - x}{\|\tilde{c} - x\|} = \frac{c_i - x}{\|c_i - x\|}$. Therefore, $u(x) = 0$ for all $x \in \tilde{\mathcal{L}}_d(x_d, c_i, R) := \mathcal{L}_d(x_d, c_i) \cap \tilde{\mathcal{D}}(x_d, c_i, R)$ where $\mathcal{L}_d(x_d, c_i)$ is defined in Lemma 3.2. Finally, one can conclude that the set of undesired equilibria for the closed-loop system (3.7)-(3.33) is $\zeta = \{x_d\} \cup \Big( \cup_{i \in \mathbb{I}} \tilde{\mathcal{L}}_d(x_d, c_i, R) \Big)$.

## A.10 Proof of Theorem 3.3

Since the sensor-based case is a special case of the map-based scenario when each obstacle is considered as a unique obstacle in the workspace, item i) follows from item i) of Theorem 3.1. Item ii) follows directly from Lemma 3.8. Now, let us prove item iii). Consider obstacle $i \in \mathbb{I}$ and the set of equilibrium points $\tilde{\mathcal{L}}_d(x_d, c_i, R)$. Define the tube $\mathcal{TU}_i := \{x \in \tilde{\mathcal{D}}(x_d, c_i, R) | d(x, \tilde{\mathcal{L}}_d(x_d, c_i, R)) \leq e_i\}$ surrounding $\tilde{\mathcal{L}}_d(x_d, c_i, R)$ inside the practical shadow region where $e_i$ is small such that $\tilde{c} = \arg\min\limits_{y \in \mathcal{O}_i} \|x - y\|$. Let $V(x) = 1 - \frac{(\tilde{x}_i - c_i)^\top}{\|\tilde{x}_i - c_i\|} \frac{(x - \tilde{c})}{\|x - \tilde{c}\|}$ where $\tilde{x}_i \in \tilde{\mathcal{L}}_d(x_d, c_i, R)$. Note that $V(\tilde{x}_i) = 0$ and $V(x) > 0$ for all $x \in \mathcal{TU}_i \setminus \tilde{\mathcal{L}}_d(x_d, c_i, R)$. Define the set $U := \{x \in \mathcal{TU}_i | V(x) > 0\}$. The time-derivative of $V(x)$ on $\mathcal{TU}_i$ is given by

$$\begin{aligned}
\dot{V}(x) &= \frac{\partial V(x)}{\partial x}^\top \dot{x}, \\
&= -\frac{(\tilde{x}_i - c_i)^\top}{\|\tilde{x}_i - c_i\|} J_x\left(\frac{(x-\tilde{c})}{\|x-\tilde{c}\|}\right) u(x), \\
&= \frac{-1}{\|\tilde{x}_i - c_i\|\|x-\tilde{c}\|}(\tilde{x}_i - c_i)^\top \pi^\perp\left(\frac{(x-\tilde{c})}{\|x-\tilde{c}\|}\right) u(x), \\
&= \frac{-\gamma}{\|\tilde{x}_i - c_i\|\|x-\tilde{c}\|}(\tilde{x}_i - c_i)^\top \pi^\perp\left(\frac{(x-\tilde{c})}{\|x-\tilde{c}\|}\right) (x_d - x), \\
&= -K(\tilde{x}_i - c_i)^\top \pi^\perp(V_{ci})(x_d - x),
\end{aligned}$$

where $K = \frac{\gamma}{\|\tilde{x}_i - c_i\|\|x-\tilde{c}\|}$, and as $\tilde{c}$ is the projection of $x$ onto obstacle $i$ for all $x \in \mathcal{TU}_i$, $V_{ci} = \frac{(c_i - x)}{\|c_i - x\|} = \frac{(\tilde{c}-x)}{\|\tilde{c}-x\|}$. Since $\tilde{x}_i = c_i + \delta\frac{c_i - x_d}{\|c_i - x_d\|}$ with $\delta \geq r_i$, one gets

$$\begin{aligned}
\dot{V}(x) &= -\frac{\delta K}{\|c_i - x_d\|}(c_i - x_d)^\top \pi^\perp(V_{ci})(x_d - x), \\
&= -\frac{\delta K}{\|c_i - x_d\|}\big((c_i - x) + (x - x_d)\big)^\top \pi^\perp(V_{ci})(x_d - x), \\
&= \frac{\delta K}{\|c_i - x_d\|}(x_d - x)^\top \pi^\perp(V_{ci})(x_d - x).
\end{aligned}$$

where we used the fact that $(c_i - x)^\top \pi^\perp(V_{c_i})(x_d - x) = 0$. It is clear that $\dot{V}(x) > 0$ for all $x \in U$, and $\dot{V}(x) = 0$ for all $x \in \tilde{\mathcal{L}}_d(x_d, c_i, R)$ over the set $\mathcal{TU}_i$. Since $U$ is a compact set, $V(x)$ is increasing on $U$, and $V(x) = 0$ on $\tilde{\mathcal{L}}_d(x_d, c_i, R)$ (the tube axis), $x(t)$ must leave the set $U$. Note that the set $U$ is bounded by the free space boundary and the lateral surface of tube $\mathcal{TU}_i$. Due to the safety of the system, as per item i), $x(t)$ can not leave $U$ from the free space boundary and can only leave it from the surface of the tube for all $x(0) \in U$. Therefore, the set of equilibria $\tilde{\mathcal{L}}_d(x_d, c_i, R)$ is unstable. Lastly, we prove item iv). Consider the equilibrium point $x_d$ and the positive definite function $V_1(x) = \frac{1}{2}||x - x_d||^2$ whose time-derivative is given by

$$\begin{aligned}
\dot{V}_1(x) &= \frac{\partial V_1(x)}{\partial x}^\top \dot{x}, \\
&= (x - x_d)^\top u(x), \\
&= \begin{cases} -\gamma\|x - x_d\|^2, & x \in \widetilde{\mathcal{VI}} \\ -\gamma\|x - x_d\|^2 + \gamma\|x - x_d\|\frac{\sin(\tilde{\theta}-\tilde{\beta})}{\sin(\tilde{\theta})}\frac{(x_d - x)^\top(\tilde{c}-x)}{\|\tilde{c}-x\|}, & x \in \widetilde{\mathcal{BL}} \end{cases} \\
&= \begin{cases} -\gamma\|x - x_d\|^2, & x \in \widetilde{\mathcal{VI}} \\ -\gamma\|x - x_d\|^2 + \gamma\|x - x_d\|^2\frac{\sin(\tilde{\theta}-\tilde{\beta})}{\sin(\tilde{\theta})}\cos(\tilde{\beta}), & x \in \widetilde{\mathcal{BL}} \end{cases} \\
&= \begin{cases} -\gamma\|x - x_d\|^2, & x \in \widetilde{\mathcal{VI}} \\ -\gamma\|x - x_d\|^2\frac{\sin(\tilde{\beta})}{\sin(\tilde{\theta})}\cos(\tilde{\theta} - \tilde{\beta}), & x \in \widetilde{\mathcal{BL}} \end{cases}
\end{aligned}$$

where we used the fact that $\sin(\tilde{\theta}) - \sin(\tilde{\theta} - \tilde{\beta})\cos(\tilde{\beta}) = \sin(\tilde{\beta})\cos(\tilde{\theta} - \tilde{\beta})$, $0 < \tilde{\theta} \leq \frac{\pi}{2}$ and $0 \leq \tilde{\beta} \leq \tilde{\theta}$. Therefore, $\dot{V}_1(x) = 0$ only for $x \in \zeta$ and $\dot{V}_1(x) < 0$ for all $x \in \mathcal{F} \setminus \zeta$. Since the practical shadow regions are disjoint by construction, and the undesired equilibria $\tilde{\mathcal{L}}_d(x_d, c_i, R)$ are limited to the shadow regions as per Lemma 3.8, $\mathbb{L} = \varnothing$ and $\mathbf{Cell}_i = \varnothing$ for all $i \in \mathbb{I}$. Thus, for all $i \in \mathbb{I}$, the attraction region of the set of undesired equilibria $\tilde{\mathcal{L}}_d(x_d, c_i, R)$ reduces to the line segment $\mathcal{L}_d(x_d, c_i) \cap \mathcal{D}^t(x_d, c_i)$. Since the attraction region of the undesired equilibria is a set of measure zero, the equilibrium point $x_d$ is almost globally asymptotically stable in $\mathcal{F}$.

## A.11 Proof of Lemma 3.9

Following the same procedure as in A.9, $u(x) = 0$ if and only if $x = x_d$ or $x \in \tilde{\mathcal{L}}_d(x_d, \mathrm{x}_i, R)$, where for all $x \in \tilde{\mathcal{L}}_d(x_d, \mathrm{x}_i, R)$, $\tilde{c} = \mathrm{x}_i$ and $\frac{(\tilde{c}-x)^\top}{\|\tilde{c}-x\|}\frac{(x_d-x)^\top}{\|x_d-x\|} = 1$. Therefore, one can conclude that the set of equilibria is $\tilde{\zeta} = \{x_d\} \cup \Big(\cup_{i \in \mathbb{I}} \tilde{\mathcal{L}}_d(x_d, \mathrm{x}_i, R)\Big)$.

## A.12 Proof of Theorem 3.4

Since the considered convex obstacles have smooth boundaries, the tangent cone on the boundaries of the obstacles are half-planes, which is similar to the spherical obstacles case. Therefore, item i) follows from item i) of Theorem 3.3. Item ii) follows directly from Lemma 3.9. Now, let us prove item iii). Consider obstacle $i \in \mathbb{I}$ and the equilibrium point $\bar{x}_i \in \tilde{\mathcal{L}}_d(x_d, \mathrm{x}_i, R)$. Define the tube $\mathcal{TU}_i := \{x \in \tilde{\mathcal{D}}(x_d, i, R) | d(x, \tilde{\mathcal{L}}_d(x_d, \mathrm{x}_i, R)) \leq e_i\}$ surrounding $\tilde{\mathcal{L}}_d(x_d, \mathrm{x}_i, R)$ inside the practical shadow region where $e_i$ is small such that $\tilde{c} = \arg\min\limits_{y \in \tilde{\mathcal{O}}_i} \|x - y\|$. Let $V(x) = 1 - \frac{(\bar{x}_i - x_d)^\top}{\|\bar{x}_i - x_d\|}\frac{(x-\tilde{c})}{\|x-\tilde{c}\|}$ where $\bar{x}_i \in \tilde{\mathcal{L}}_d(x_d, \mathrm{x}_i, R)$. Note that $V(\bar{x}_i) = 0$ and $V(x) > 0$ for all $x \in \mathcal{TU}_i \setminus \tilde{\mathcal{L}}_d(x_d, \mathrm{x}_i, R)$. Define the set $U := \{x \in \mathcal{TU}_i | V(x) > 0\}$. The time-derivative of $V(x)$ on $\mathcal{TU}_i$ is given by

$$
\begin{aligned}
\dot{V}(x) = \frac{\partial V(x)}{\partial x}^\top \dot{x} &= -\frac{(\bar{x}_i - x_d)^\top}{\|\bar{x}_i - x_d\|} J_x \left(\frac{(x-\tilde{c})}{\|x-\tilde{c}\|}\right) u(x), \\
&= \frac{-1}{\|x-\tilde{c}\|}\frac{(\bar{x}_i - x_d)^\top}{\|\bar{x}_i - x_d\|} \pi^\perp \left(\frac{(x-\tilde{c})}{\|x-\tilde{c}\|}\right) u(x), \\
&= \frac{\gamma}{\|x-\tilde{c}\|}\frac{(\bar{x}_i - x_d)^\top}{\|\bar{x}_i - x_d\|} \pi^\perp \left(\frac{(x-\tilde{c})}{\|x-\tilde{c}\|}\right) (x - x_d), \\
&= K(\cos(\varphi_d) - \cos(\tilde{\varphi})\cos(\tilde{\beta})),
\end{aligned}
$$

where $K = \gamma\frac{\|x-x_d\|}{\|x-\tilde{c}\|}$, $\tilde{\varphi} = \angle(\bar{x}_i - x_d, x - \tilde{c})$, $0 < \varphi_d = \angle(\bar{x}_i - x_d, x - x_d) \leq \varphi_d^{max}$, and $\varphi_d^{max} = \arcsin(e_i/\|x - x_d\|) \in (0, \pi/2]$. Since obstacle $i$ satisfies the curvature condition, as per Assumption 3.4, and $\tilde{c}$ is the projection of $x$ onto obstacle $i$, $0 < \tilde{\varphi} = \varphi_d + \tilde{\beta} < \pi$, where $0 \leq \tilde{\beta} = \angle(\tilde{c} - x, x_d - x) \leq \frac{\pi}{2}$. Thus, $\dot{V}(x) = K\sin(\tilde{\varphi})\sin(\tilde{\beta})$. It is clear that, over the set $\mathcal{TU}_i$, $\dot{V}(x) = 0$ for $x \in \tilde{\mathcal{L}}_d(x_d, \mathrm{x}_i, R)$(*i.e.*, $\tilde{\beta} = 0$), and $\dot{V}(x) > 0$ for all $x \in U$. Since $U$ is a compact set, $V(x)$ is increasing on $U$, and $V(x) = 0$ on $\tilde{\mathcal{L}}_d(x_d, \mathrm{x}_i, R)$ (the tube axis), $x(t)$ must leave the set $U$. Note that the set $U$ is bounded by the free space boundary and the lateral surface of tube $\mathcal{TU}_i$. Due to the safety of the system, as per item i), $x(t)$ can not leave $U$ from the free space boundary and can only leave it from the surface of the tube for all $x(0) \in U$. Therefore, the set of equilibria $\tilde{\mathcal{L}}_d(x_d, \mathrm{x}_i, R)$ is unstable. Finally, proof of item iv) is similar to that of item iv) in Appendix A.10.

# Appendix B

# Proofs of Chapter 4

## B.1 Proof of Lemma 4.2

The sets $\mathcal{F}$ and $\mathcal{J}$, defined in (4.9), are by construction closed subsets of $\mathbb{R}^n \times \mathbb{I} \times \mathbb{M}$, which shows that condition i) is satisfied.

Since the flow map F, given in (4.24), is defined for all $\xi \in \mathcal{F}$, single-valued and continuous on $\mathcal{F}$, then $\mathcal{F} \subset \text{dom}(\text{F})$, F is outer semicontinuous and bounded relative to $\mathcal{F}$, and convex for every $\xi \in \mathcal{F}$. Therefore, condition ii) is fulfilled.

The jump map J, given in (4.25), is single-valued on $\mathcal{J}_m \times \{m\}$, $m \in \{-1, 1\}$. Since the angles $\varphi_k^1, \varphi_k^{-1}$ are selected according to Lemma 2.1, we have $\mathcal{C}^{\leq}_{\mathbb{R}^n}(c_k, v_k^1, \varphi_k^1) \cap \mathcal{C}^{\leq}_{\mathbb{R}^n}(c_k, v_k^{-1}, \varphi_k^{-1}) = \{c_k\}$, which implies that $\mathcal{C}^{\geq}_{\mathbb{R}^n}(c_k, v_k^1, \varphi_k^1) \cap \mathcal{C}^{\geq}_{\mathbb{R}^n}(c_k, v_k^{-1}, \varphi_k^{-1}) = \mathbb{R}^n$. Therefore, given (4.22), $B(x, k) \neq \varnothing$ for all $(x, k) \in \mathbb{R}^n \times \mathbb{I}$, then $M(\xi) \neq \varnothing$ for all $\xi \in \mathcal{J}_0 \times \{0\}$, and hence $\text{J}(\xi) \neq \varnothing$ for all $\xi \in \mathcal{J}_0 \times \{0\}$. Thus, $\mathcal{J} \in \text{dom}(\text{J})$. Moreover, J has a closed graph relative to $\mathcal{J}_0 \times \{0\}$ as $B$ is allowed to be set-valued whenever $x \in \cap_{m=-1,1}\mathcal{C}^{\leq}_{\mathbb{R}^n}(c_k, v_k^m, \varphi_k^m)$. Then, according to (Goebel et al., 2012, Lemma 5.10), J is outer semicontinuous relative to $\mathcal{J}$. Furthermore, the jump map J is locally bounded relative to $\mathcal{J}$ since $M$ and $K$ take values over finite discrete sets $\mathbb{I}$ and $\mathbb{M}$, and the remaining component of J is a single-valued continuous function on $\mathcal{J}$, which shows the satisfaction of condition iii) and completes the proof.

## B.2 Proof of Theorem 4.1

**Item i):** First, we prove that $\mathcal{F} \cup \mathcal{J} = \mathcal{K}$, which boils down to show that for each $k \in \mathbb{I}$, and $m \in \{-1, 1\}$,

$$\tilde{\mathcal{F}}_0 \cup \tilde{\mathcal{J}}_0 = \mathcal{F}_k^m \cup \mathcal{J}_k^m = \mathcal{X}, \tag{B.1}$$

since the satisfaction of (B.1), along with (4.11), (4.17), and (4.9), implies that $\mathcal{F} \cup \mathcal{J} = \mathcal{X} \times \mathbb{I} \times \mathbb{M} = \mathcal{K}$. We start by showing $\tilde{\mathcal{F}}_0 \cup \tilde{\mathcal{J}}_0 = \mathcal{X}$ in (B.1). Recall that $\tilde{\mathcal{F}}_0 := \cap_{k\in\mathbb{I}}\mathcal{F}_k^0$, $\tilde{\mathcal{J}}_0 := \cup_{k\in\mathbb{I}}\mathcal{J}_k^0$, and for each $k \in \mathbb{I}$, $\mathcal{F}_k^0 := \overline{\mathcal{X} \setminus \mathcal{A}_k(x_d)}$, $\mathcal{J}_k^0 := \mathcal{A}_k(x_d)$, as defined in (4.10), and hence,

$$\begin{aligned}\left(\bigcap_{k\in\mathbb{I}}\mathcal{F}_k^0\right)\cup\left(\bigcup_{k\in\mathbb{I}}\mathcal{J}_k^0\right)&=\left(\bigcap_{k\in\mathbb{I}}\overline{\mathcal{X}\setminus\mathcal{A}_k(x_d)}\right)\cup\left(\bigcup_{k\in\mathbb{I}}\mathcal{A}_k(x_d)\right)\\&=\bigcap_{k\in\mathbb{I}}\left\{\overline{\mathcal{X}\setminus\mathcal{A}_k(x_d)}\cup\left(\bigcup_{i\in\mathbb{I}}\mathcal{A}_i(x_d)\right)\right\}\\&=\bigcap_{k\in\mathbb{I}}\overline{\mathcal{X}}=\mathcal{X},\end{aligned}$$

where $\mathcal{X}$ is a closed set as defined in (4.1). Now we prove that for each $k \in \mathbb{I}$, and $m \in \{-1, 1\}$, $\mathcal{F}_k^m \cup \mathcal{J}_k^m = \mathcal{X}$. Note that from (4.15), $\mathcal{F}_k^m := \mathcal{A}_k(x_k^m) \setminus \mathcal{C}_{\mathcal{X}}^{<}(c_k, v_k^m, \varphi_k^m)$, $\mathcal{J}_k^m := \overline{\mathcal{X} \setminus \mathcal{F}_k^m}$, and thus, $\mathcal{F}_k^m \cup \mathcal{J}_k^m = \mathcal{F}_k^m \cup (\overline{\mathcal{X} \setminus \mathcal{F}_k^m}) = \mathcal{X}$.

Let us define the set $\mathcal{M}_{\mathcal{H}}(\mathcal{K})$ of all maximal solutions to the hybrid system (4.26) represented by its data $\mathcal{H}$ with $\xi(0,0) \in \mathcal{K}$. Each solution $\xi \in \mathcal{M}_{\mathcal{H}}(\mathcal{K})$ has range $\operatorname{rge}\xi \subset \mathcal{K} = \mathcal{F} \cup \mathcal{J}$. The augmented state space $\mathcal{K}$ is forward invariant for $\mathcal{H}$ if, for each $\xi(0,0) \in \mathcal{K}$, there exists one solution, and every $\xi \in \mathcal{M}_{\mathcal{H}}(\mathcal{K})$ is complete and has range $\operatorname{rge}\xi \subset \mathcal{K}$ as per (Chai and Sanfelice, 2019, Definition 3.3). The forward invariance of $\mathcal{K}$ is then shown by the completeness of the solutions $\xi \in \mathcal{M}_{\mathcal{H}}(\mathcal{K})$ which we prove using (Goebel et al., 2012, Proposition 6.10). We start by showing the following viability condition

$$\mathrm{F}(\xi) \cap \mathrm{T}_{\mathcal{F}}(\xi) \neq \varnothing, \forall \xi \in \mathcal{F} \setminus \mathcal{J}, \tag{B.2}$$

where $\mathrm{T}_{\mathcal{F}}(\xi)$ is Bouligand's tangent cone of the set $\mathcal{F}$ at $\xi$ as defined in (2.15). Inspired by (Berkane et al., 2022, Appendix 1), we proceed as follows. Let $\xi = (x, k, m) \in \mathcal{F} \setminus \mathcal{J}$, which implies by (4.9) that $(x, k) \in \mathcal{F}_m \setminus \mathcal{J}_m$ for some $m \in \mathbb{M}$. Consider the two cases (modes) $m = 0$ and $m \in \{-1, 1\}$. For $m = 0$, as per definition (4.11), there exists $k \in \mathbb{I}$ such that $x \in \tilde{\mathcal{F}}_0 \setminus \tilde{\mathcal{J}}_0$. When $x \in \mathring{\tilde{\mathcal{F}}}_0 \setminus \tilde{\mathcal{J}}_0$, then $x$ is in the interior of the set $\tilde{\mathcal{F}}_0$, and hence, $\mathrm{T}_{\mathcal{F}}(\xi) = \mathbb{R}^n \times \{0\} \times \{0\}$ and (B.2) holds. When $x \in \partial\tilde{\mathcal{F}}_0 \setminus \tilde{\mathcal{J}}_0$, it is clear, according to (4.4) and (4.10), that $x$ must be on the boundary of one of the obstacles and does not belong to the *active region* (*i.e.*, $\partial\tilde{\mathcal{F}}_0 \setminus \tilde{\mathcal{J}}_0 \subseteq \bigcup_{k\in\mathbb{I}}(\partial\mathcal{O}_k \setminus \mathcal{A}_k(x_d))$. Then, $\mathrm{T}_{\mathcal{F}}(x, k, 0) = \mathcal{P}_{\geq}(x, x - c_k) \times \{0\} \times \{0\}$. Since $u(x, k, 0) = -\gamma(x - x_d)$, $u(x, k, 0)^\top(x - c_k) > 0$ and (B.2) holds. Now, when $m \in \{-1, 1\}$, according to (4.15), there exists $k \in \mathbb{I}$ such that $x \in \mathring{\mathcal{F}}_m^k \setminus \mathcal{J}_k^m$. For $x \in \mathring{\mathcal{F}}_m^k \setminus \mathcal{J}_k^m$, $\mathrm{T}_{\mathcal{F}}(x, k, m) = \mathbb{R}^n \times \{0\} \times \{0\}$ and (B.2) holds. When $x \in \partial\mathcal{F}_k^m \setminus \mathcal{J}_k^m$, it is clear, according to (4.15), that $x$ must be on the boundary of obstacle $\mathcal{O}_k$ and belongs to the *active region* (*i.e.*, $\partial\mathcal{F}_k^m \setminus \mathcal{J}_k^m \subseteq \partial\mathcal{O}_k \cap \mathcal{A}_k(x_k^m)$). Thus, $\mathrm{T}_{\mathcal{F}}(x, k, m) = \mathcal{P}_{\geq}(x, x - c_k) \times \{0\} \times \{0\}$. From (4.13) and (4.8a), and since $x \in \partial\mathcal{O}_k$, then $\theta(x, k) = \frac{\pi}{2}$ and $u(x, k, m) = -\gamma\mu(x, k, m)\left(x_k^m - x - \|x_k^m - x\| \cos(\beta(x, k, m))\frac{c_k - x}{\|c_k - x\|}\right)$ where $\beta(x, k, m) = \angle(c_k - x, x_k^m - x)$ and $\mu(x, k, m) = \left(1 + \frac{e_k}{\|x - x_k^m\|}\frac{\beta(x,k,m)}{\theta(x,k)}\right) > 0$. Hence,

$$\begin{aligned}u(x, k, m)^\top(x - c_k) = \gamma\mu(x, k, m)((x_k^m - x)^\top(x - c_k)&\\-\frac{\|x_k^m - x\|}{\|c_k - x\|}\cos(\beta(x, k, m))(c_k - x)^\top(x - c_k))&\end{aligned}$$

$$
\begin{aligned}
&= \gamma\mu(x,k,m)((x_k^m - x)^\top (x - c_k) + \|x_k^m - x\|\|c_k - x\| \cos(\beta(x,k,m)) \\
&= \gamma\mu(x,k,m)((x_k^m - x)^\top (x - c_k) + (x_k^m - x)^\top (c_k - x)) \\
&= 0,
\end{aligned}
$$

and (B.2) holds for $m \in \{-1, 1\}$. Therefore, according to (Goebel et al., 2012, Proposition 6.10), since (B.2) holds for all $\xi \in \mathcal{F} \setminus \mathcal{J}$, there exists a nontrivial solution to the hybrid system $\mathcal{H}$ for every initial condition in $\mathcal{K}$. Finite escape times can only occur via the flow. They cannot occur for $x \in \mathcal{F}_k^{-1} \cup \mathcal{F}_k^1$, as the sets $\mathcal{F}_k^{-1}$ and $\mathcal{F}_k^1$ are bounded by their definition 4.15, nor for $x \in \tilde{\mathcal{F}}_0$ since they would make $(x - x_d)\top(x - x_d)$ grow unbounded, which would contradict the fact that $\frac{d}{dt}((x - x_d)\top(x - x_d)) \leq 0$ by the definition of $u(x, k, 0)$. Therefore, all maximal solutions do not have finite escape times. Moreover, according to (4.26), $x^+ = x$, and from the definitions (4.21), (4.22), and (4.25), it follows that $\mathrm{J}(\mathcal{J}) \subset \mathcal{K}$. Thus, solutions of the hybrid system (4.26) cannot leave $\mathcal{K}$ through jumps and, as per (Goebel et al., 2012, Proposition 6.10), all maximal solutions are complete.

**Item ii):** Using (Goebel et al., 2012, Definition 7.1), we first show the stability of $\mathcal{A}$, and then its global attractivity. Since $x_d \in \mathring{\mathcal{X}}$, there exists $\bar{\varepsilon} > 0$ such that $\mathcal{B}(x_d, \bar{\varepsilon}) \cap \mathcal{O}_k = \varnothing$, $\forall k \in \mathbb{I}$. As per the sets definitions in (4.10) and (4.11), $\mathcal{B}(x_d, \varepsilon) \subset \tilde{\mathcal{F}}_0$ for all $\varepsilon \in [0, \bar{\varepsilon}]$. Thus, $\mathcal{B}(x_d, \varepsilon) \cap \tilde{\mathcal{J}}_0 = \varnothing$, and $x$ evolves under $\dot{x} = -\gamma(x - x_d)$, which implies forward invariance of the set $\mathcal{B} := \mathcal{B}(x_d, \varepsilon) \times \mathbb{I} \times \mathbb{M}$. Therefore, according to (Goebel et al., 2012, Definition 7.1), the set $\mathcal{A}$ is stable for the hybrid system (4.26). Now, let us prove the global attractivity of $\mathcal{A}$, but first, we need the following lemma.

**Lemma B.1** *For a given obstacle index $k \in \mathbb{I}$, the obstacle-avoidance mode $m \in \{-1, 1\}$ is active for a finite hybrid time interval $[t_0^{k,m}, t_f^{k,m}] \times [j_0^{k,m}, j_f^{k,m}]$ where $(t_0^{k,m}, j_0^{k,m})$ and $(t_f^{k,m}, j_f^{k,m})$ are, respectively, the activation and deactivation hybrid times of mode $m$ for the obstacle with index $k$.*

*Proof:* Consider an obstacle index $k \in \mathbb{I}$ and the positive definite function $V_m(x) = \frac{1}{2}||x - x_k^m||^2$, for $m \in \{-1, 1\}$ and $x \in \mathcal{F}_k^m$. The time derivative of $V_m(x)$ is given by

$$
\begin{aligned}
&\dot{V}_m(x) = \frac{\partial V_m(x)}{\partial x}^\top \dot{x} = (x - x_k^m)^\top \big(\alpha(x,k)\mu(x,k,m)\kappa(x,k,m) + (1 - \alpha(x,k))u_d(x)\big), \\
&= -(x_k^m - x)^\top \left\{ K_1 \left( \gamma(x_k^m - x) - \tau(x,k,m) \frac{c_k - x}{\|c_k - x\|} \right) + \gamma K_2 (x_d - x) \right\}, \\
&= -\left\{ K_1 \left( \gamma \|x_k^m - x\|^2 - \tau(x,k,m) \frac{(x_k^m - x)^\top (c_k - x)}{\|c_k - x\|} \right) + \gamma K_2 (x_k^m - x)^\top (x_d - x) \right\}, \\
&= -\gamma \left\{ K_1 \left( \|x_k^m - x\|^2 - \frac{\|x_k^m - x\|}{\|c_k - x\|} \frac{\sin(\theta(x,k) - \beta(x,k,m))}{\sin(\theta(x,k))} (x_k^m - x)^\top (c_k - x) \right) \right. \\
&\quad \left. + K_2 (x_k^m - x)^\top (x_d - x) \right\}, \\
&= -\gamma \left\{ K_1 \left( \|x_k^m - x\|^2 - \|x_k^m - x\|^2 \frac{\sin(\theta(x,k) - \beta(x,k,m))}{\sin(\theta(x,k))} \cos(\beta(x,k,m)) \right) \right. \\
&\quad \left. + K_2 (x_k^m - x)^\top (x_d - x) \right\},
\end{aligned}
$$

$$= -\gamma\bigg(K_1\|x_k^m - x\|^2\frac{\sin(\beta(x,k,m))}{\sin(\theta(x,k))}\cos(\theta(x,k)-\beta(x,k,m)) + K_2(x_k^m - x)^\top(x_d - x)\bigg),$$

where $K_1 = \alpha(x,k)\mu(x,k,m) \geq 0$, $K_2 = 1-\alpha(x,k) \geq 0$, and we used the fact that $\sin(\theta(x,k)) - \sin(\theta(x,k)-\beta(x,k,m))\cos(\beta(x,k,m)) = \sin(\beta(x,k,m))\cos(\theta(x,k)-\beta(x,k,m))$. Since $0 < \theta(x,k) \leq \frac{\pi}{2}$ and $0 \leq \beta(x,k,m) \leq \theta(x,k)$, one can deduce that

$$K_1\|x_k^m - x\|^2\frac{\sin(\beta(x,k,m))}{\sin(\theta(x,k)})\cos(\theta(x,k)-\beta(x,k,m)) \geq 0, \tag{B.3}$$

for all $(x,k,m) \in \mathcal{F}_m \times \{-1,1\}$. Next, we prove that $(x_k^m - x)^\top(x_d - x) \geq 0$. The following fact will be needed in the rest of the proof.

**Fact 1** $\forall(k,m) \in \mathbb{I}\times\{-1,1\}$, $\mathcal{F}_k^m \subset \mathcal{P}_{\leq}(p_k, x_d - p_k)$.

*Proof:* Since $p_k \in \partial\mathcal{O}_k$ and $x_k^m \in \mathcal{P}_{\leq}(p_k, c_k - p_k)$ $\big($*i.e.*, $(c_k - p_k)^\top(x_k^m - p_k) \leq 0\big)$, then $p_k \in \mathcal{C}^{\leq}_{\mathcal{X}}(x_k^m, c_k - x_k^m, \theta_k(x_k^m)) \setminus \mathcal{S}_k(x_k^m)$. Therefore, $\mathcal{S}_k(x_k^m) \subset \mathcal{P}_{\geq}(p_k, c_k - p_k)$. Moreover, since $\mathcal{F}_k^m \subset \mathcal{S}_k(x_k^m)$ (see definitions (4.9), (4.4)) and $p_k = c_k + r_k(x_d - c_k)/\|x_d - c_k\|$, one can deduce that $\mathcal{F}_k^m \subset \mathcal{P}_{\leq}(p_k, x_d - p_k)$ for $(k,m) \in \mathbb{I}\times\{-1,1\}$, which concludes the proof of Fact 1.
According to (4.17), $(x,k) \in \mathcal{F}_m$ implies that $x \in \mathcal{F}_k^m$, and hence, according to (4.15),

$$(c_k - x)^\top(x_k^m - x) \geq 0. \tag{B.4}$$

Moreover, since $x_k^m \in \mathcal{H}_k(x_d) \cap \mathcal{P}_{\geq}(p_k, x_d - p_k)$, one has

$$(x_d - p_k)^\top(x_k^m - p_k) \geq 0, \tag{B.5a}$$

$$\left(x_d - c_k - r_k\frac{x_d - c_k}{\|x_d - c_k\|}\right)^\top\left(x_k^m - c_k - r_k\frac{x_d - c_k}{\|x_d - c_k\|}\right) \geq 0, \tag{B.5b}$$

$$\left(1 - \frac{r_k}{\|x_d - c_k\|}\right)(x_d - c_k)^\top\left(x_k^m - c_k - r_k\frac{x_d - c_k}{\|x_d - c_k\|}\right) \geq 0, \tag{B.5c}$$

$$(x_d - c_k)^\top(x_k^m - c_k) - r_k\|x_d - c_k\| \geq 0, \tag{B.5d}$$

$$(x_d - c_k)^\top(x_k^m - c_k) \geq r_k\|x_d - c_k\|. \tag{B.5e}$$

In addition, as per Fact 1, for all $x \in \mathcal{F}_k^m$, one has

$$(x_d - p_k)^\top(x - p_k) \leq 0, \tag{B.6a}$$

$$\left(x_d - c_k - r_k\frac{x_d - c_k}{\|x_d - c_k\|}\right)^\top\left(x - c_k - r_k\frac{x_d - c_k}{\|x_d - c_k\|}\right) \leq 0, \tag{B.6b}$$

$$\left(1 - \frac{r_k}{\|x_d - c_k\|}\right)(x_d - c_k)^\top\left(x - c_k - r_k\frac{x_d - c_k}{\|x_d - c_k\|}\right) \leq 0, \tag{B.6c}$$

$$(x_d - c_k)^\top(x - c_k) - r_k\|x_d - c_k\| \leq 0, \tag{B.6d}$$

$$(x_d - c_k)^\top(x - c_k) \leq r_k\|x_d - c_k\|. \tag{B.6e}$$

From (B.5e) and (B.6e), one gets

$$(x_d - c_k)^\top(x_k^m - c_k) - (x_d - c_k)^\top(x - c_k) \geq 0,$$

$$
\begin{aligned}
(x_d - c_k)^\top (x_k^m - x) &\geq 0,\\
(x_d - x)^\top (x_k^m - x) + (x - c_k)^\top (x_k^m - x) &\geq 0,\\
(x_d - x)^\top (x_k^m - x) &\geq (c_k - x)^\top (x_k^m - x),
\end{aligned}
$$

and according to (B.4), one can show that

$$(x_d - x)^\top (x_k^m - x) \geq 0. \tag{B.7}$$

Therefore, $\dot{V}_m(x) \leq 0$ for all $(x, k, m) \in \mathcal{F}_k^m \times \mathbb{I} \times \{-1, 1\}$. Moreover, $\dot{V}_m(x) = 0$ only if $x \in \mathcal{L}_k(x_k^m)$, which is excluded from the set $\mathcal{F}_k^m$ for $m \in \{-1, 1\}$, then one can conclude that $V_m(x) < 0$ for all $(x, k, m) \in \mathcal{F}_k^m \times \mathbb{I} \times \{-1, 1\}$. Thus, as $x_k^m \notin \mathcal{F}_k^m$, $k \in \mathbb{I}$, there exists a hybrid time $t_f^{k,m} > 0$, $j_f^{k,m} \in \mathbb{N} \setminus \{0\}$, such that $x(t_f^{k,m}, j_f^{k,m})$ leaves $\mathcal{F}_k^m$ and the mode $m(j_f^{k,m}, j_f^{k,m})$ jumps to the *motion-to-destination* mode. Therefore, if the *obstacle-avoidance* mode ($m \in \{-1, 1\}$) is activated at $(t_0^{k,m}, j_0^{k,m})$, the hybrid time interval for this mode $m$ is given by $[t_0^{k,m}, t_f^{k,m}] \times [j_0^{k,m}, j_f^{k,m}]$, which concludes the proof of Lemma B.1.

Lemma B.1 shows that during the avoidance of any obstacle $\mathcal{O}_k$, $k \in \mathbb{I}$, the *obstacle-avoidance* mode remains active for a finite hybrid time interval $[t_0^{k,m}, t_f^{k,m}] \times [j_0^{k,m}, j_f^{k,m}]$, and the mode jumps to the *motion-to-destination* mode (*i.e.*, $m \in \{-1, 1\}$) and $x$ leaves the flow set $\mathcal{F}_k^m$ to the flow set $\mathcal{F}_k^0$ of the *motion-to-destination* mode. Therefore, since the number of obstacles is finite, there exists a finite sequence of obstacles to avoid for every $x(0, 0) \in \mathcal{X}$, and after the last avoidance in this sequence, the mode $m$ jumps to the *motion-to-destination* mode where the flow $\dot{x} = -\gamma(x - x_d)$ guarantees the global attractivity of the equilibrium set $\mathcal{A}$.

## B.3 Proof of Lemma 4.3

For the hybrid time interval $[t_0^k, t_f^k] \times [j_0^k, j_f^k]$ where obstacle $k$ is selected for avoidance, the robot, according to the jump maps $K(\cdot)$ and $M(\cdot)$ defined in (4.20)-(4.21), has to first operate in the *obstacle-avoidance* mode ($m = \pm 1$) and then in the *motion-to-destination* mode ($m = 0$). Let $(t_s^k, j_s^k)$ be the hybrid time at which the mode selector jumps from $m = \pm 1$ to $m = 0$. Let us consider the first case where $x(t_0^k, j_0^k) \in \mathcal{J}_0^k \setminus \mathcal{L}(x_d, c_k)$. In the first mode ($m \pm 1$), the velocity vector $u(x, k, m)$, defined in (4.13), is a function of three vectors $(c_k - x)$, $(x_d - x)$, and $(x_k^m - x)$. Thus, if $x_k^m \in \mathcal{H}_k(x_d) \cap \mathcal{P}_{\geq}(p_k, x_d - p_k) \cap \mathcal{PL}\left(x_d, c_k, x(t_0^k, j_0^k)\right)$, then the points $x_d, c_k, x(t_0^k, j_0^k), x_k^m$ are contained in the two-dimensional plane $\mathcal{PL}\left(x_d, c_k, x(t_0^k, j_0^k)\right)$, and the vectors $(c_k - x(t_0^k, j_0^k))$ and $(x_d - x(t_0^k, j_0^k))$ are linearly independent. Therefore, $x(t, j) \in \mathcal{PL}\left(x_d, c_k, x(t_0^k, j_0^k)\right)$ for all $(t, j) \in [t_0^k, t_s^k] \times [j_0^k, j_s^k]$. In the second mode ($m = 0$), the velocity vector is given by $u(x, k, 0) = \gamma(x_d - x)$, which implies that the resultant trajectory is the line segment joining $x(t_s^k, j_s^k)$ and $x(t_f^k, j_f^k)$ where $\left(x(t_f^k, j_f^k) - x(t_s^k, j_s^k)\right)$ and $\left(x_d - x(t_s^k, j_s^k)\right)$ are collinear. Therefore, since $x_d, x(t_s^k, j_s^k) \in \mathcal{PL}\left(x_d, c_k, x(t_0^k, j_0^k)\right)$, as shown in the previous mode, then $x(t, j) \in \mathcal{PL}\left(x_d, c_k, x(t_0^k, j_0^k)\right)$ for all $(t, j) \in [t_s^k, t_f^k] \times [j_s^k, j_f^k]$. Finally, one can conclude that $x(t, j) \in \mathcal{PL}\left(x_d, c_k, x(t_0^k, j_0^k)\right)$ for all $(t, j) \in [t_0^k, t_f^k] \times [j_0^k, j_f^k]$. In the second case where $x(t_0^k, j_0^k) \in \mathcal{J}_0^k \cap \mathcal{L}(x_d, c_k)$, the points $x_d, c_k, x(t_0^k, j_0^k)$ are aligned. Therefore, we take any point $y$ of the free space not aligned with points $x_d, c_k$ $\big(i.e.,\ y \in \mathbb{R}^n \setminus \mathcal{L}(x_d, c_k)\big)$ so that we can consider the two-dimensional plane $\mathcal{PL}(x_d, c_k, y)$ and select the virtual destination $x_k^m \in \mathcal{H}_k(x_d) \cap \mathcal{P}_{\geq}(p_k, x_d - p_k) \cap \mathcal{PL}(x_d, c_k, y)$. The proof becomes similar to that of the first case when considering the plane $\mathcal{PL}(x_d, c_k, y)$.

## B.4 Proof of Proposition 4.1

We first start by proving the continuity of the velocity control law $u(x, k, m)$. Since $u(x, k, m)$ is continuous during the flow, we only need to verify its continuity at the switching instances, which corresponds to jumps between the two modes, the *motion-to-destination* mode ($m = 0$), and the *obstacle-avoidance* mode ($m = \pm 1$).
**Case 1** ($m = 0 \to \pm 1$): The jump from the *motion-to-destination* mode to the *obstacle-avoidance* mode occurs when the robot enters the jump set $\mathcal{J}_k^0$ through the boundary $\big(\partial \mathcal{J}_k^0 \cap \partial \mathcal{B}(c_k, \bar{r}_k)\big)$. The control law for both modes is given by $u(x, k, 0) = \gamma(x_d - x)$ and $u(x, k, m) = \alpha(x, k)\mu(x, k, m)u_m(x, k) + \gamma(1 - \alpha(x, k))(x_d - x)$ for $m = \pm 1$. Since, for all $x \in \partial \mathcal{J}_k^0 \cap \partial \mathcal{B}(c_k, \bar{r}_k)$, $d(x, \mathcal{O}_k) = \bar{r}_k$, then $\alpha(x, k) = 0$ and $u(x, k, m) = u(x, k, 0)$.
**Case 2** ($m = \pm 1 \to 0$): Let $\mathcal{O}_k$, $k \in \mathbb{I}$, be the obstacle selected for the avoidance. Since the virtual destinations are selected as in Lemma 4.3, the motion during *obstacle-avoidance* and *motion-to-destination* modes, while obstacle $\mathcal{O}_k$ is selected, is two-dimensional. The motion takes place on the plane $\mathcal{PL}\left(x_d, c_k, x(t_0^k, j_0^k)\right)$ if $x(t_0^k, j_0^k) \notin \mathcal{L}(x_d, c_k)$ $\big($resp. $\mathcal{PL}(x_d, c_k, y)$, where $y \in \mathbb{R}^n \setminus \mathcal{L}(x_d, c_k)$, if $x(t_0^k, j_0^k) \in \mathcal{L}(x_d, c_k)\big)$ where $(t_0, j_0)$ is the hybrid time at which the mode variable $m$ jumps to the *obstacle-avoidance* mode. Therefore, the navigation problem, while obstacle $\mathcal{O}_k$ is selected, is reduced to the two-dimensional case. According to the jump maps defined in (4.21)-(4.28), the mode variable $m$ jumps to 1 when $x \in \left(\mathcal{J}_k^0 \cap \mathcal{C}_k^1\right) \cup \left(\mathcal{J}_k^0 \cap \mathcal{C}_k \cap \mathcal{P}_<(c_k, x_k^{-1} - x_k^1)\right) = \mathcal{J}_k^0 \cap \mathcal{P}_<(c_k, x_k^{-1} - x_k^1)$. Similarly, $m$ jumps to $-1$ when $x \in \mathcal{J}_k^0 \cap \mathcal{P}_>(c_k, x_k^{-1} - x_k^1)$. When $x \in \mathcal{J}_k^0 \cap \mathcal{P}_=(c_k, x_k^{-1} - x_k^1)$, $m$ can jump to 1 or $-1$. Thus, one can deduce that the variable $m$ always jumps to the *obstacle-avoidance* mode in which the associated virtual destination is the closest to $x$. Furthermore, as the virtual destinations belong to the cone *hat* $\mathcal{H}_k(x_d)$ of vertex $x_d$ enclosing obstacle $\mathcal{O}_k$, the jump will occur when $x \in \mathcal{E}(x_d) \cap \mathcal{E}(x_k^m) \cap \mathcal{PL}\left(x_d, c_k, x(t_0^k, j_0^k)\right)$ if $x(t_0^k, j_0^k) \notin \mathcal{L}(x_d, c_k)$ $\big($resp. $x \in \mathcal{E}(x_d) \cap \mathcal{E}(x_k^m) \cap \mathcal{PL}(x_d, c_k, y)$, where $y \in \mathbb{R}^n \setminus \mathcal{L}(x_d, c_k)$, if $x(t_0^k, j_0^k) \in \mathcal{L}(x_d, c_k)\big)$ for $m = \pm 1$. Thus, $\frac{(x_d - x)^\top}{\|x_d - x\|}\frac{(x_k^m - x)}{\|x_k^m - x\|} = 1$ and $\|x_d - x\| = \|x_k^m - x\| + \|x_d - x_k^m\|$ (*i.e.*, the points $x, x_k^m$, and $x_d$ are aligned). In addition, $\beta(x, k, m) = \angle(c_k - x, x_k^m - x) = \theta(x, k)$ when $x \in \mathcal{E}(x_k^m)$. Therefore, $\tau(x, k, m) = 0$ and $\mu(x, k, m) = \frac{\|x - x_k^m\| + e_k}{\|x - x_k^m\|}$. Hence, $u(x, k, m) = \gamma\alpha(x, k)\frac{\|x - x_k^m\| + e_k}{\|x - x_k^m\|}(x_k^m - x) + \gamma(1 - \alpha(x, k))(x_d - x)$, and since $e_k = \|x_d - x_k^m\|$, one has $u(x, k, m) = \gamma\alpha(x, k)\|x_d - x\|\frac{x_k^m - x}{\|x_k^m - x\|} + \gamma(1 - \alpha(x, k))(x_d - x) = \gamma\alpha(x, k)(x_d - x) + \gamma(1 - \alpha(x, k))(x_d - x) = \gamma(x_d - x) = u(x, k, 0)$.
Now we prove that the obstacles are avoided through local optimal obstacle-avoidance maneuvers. We use the result of (Cheniouni et al., 2023a, Lemma 1), stating that, in the case of a single spherical obstacle, the shortest path is obtained if the obstacle-avoidance maneuver is optimal (*i.e.*, the velocity vector is tangent to the obstacle and minimizes the

deviation with respect to the nominal control $u_d(x)$ in the *shadow region*) and the motion-to-destination is performed under the nominal control $u_d(x)$. According to Lemma 4.3, the control $\kappa(x,k,m)$ of the *obstacle-avoidance* mode, defined in (4.13), satisfies the optimality conditions of the obstacle-avoidance maneuver with respect to a given virtual destination $x_k^m$ and a given obstacle $\mathcal{O}_k$ in the *active region* $\mathcal{A}_k$ where $m \in \{-1,1\}$ and $k \in \mathbb{I}$. Since $u(x,k,m) = \alpha(x,k)\mu(x,k,m)\kappa(x,k,m) + (1-\alpha(x,k))u_d(x)$ for $m = \pm 1$, and $\alpha = 1$ when $d(x,\mathcal{O}_k) \leq \bar{r}_k - \epsilon$, $u(x,k,m) = \mu(x,k,m)\kappa(x,k,m)$ where $\mu(x,k,m)$ is a positive scalar function. Therefore, the hybrid control law $u(x,k,m)$ satisfies the optimality conditions of the obstacle-avoidance maneuver with respect to a given virtual destination $x_k^m$ and a given obstacle $\mathcal{O}_k$ for all $x \in \mathcal{F}_k^m \cap \mathcal{B}(c_k, r_k + \bar{r}_k - \epsilon)$, $m \in \{-1,1\}$ and $k \in \mathbb{I}$. Now we show that the optimality conditions of the obstacle-avoidance maneuver are also satisfied with respect to the destination $x_d$ by the hybrid control law $u(x,k,m)$ in the *obstacle-avoidance mode*. The first condition (*i.e.,* the velocity vector is tangent to the considered obstacle) is satisfied by construction for $m = \pm 1$ since $\kappa(x,k,m) \in \mathcal{V}(c_k - x, \theta(x,k))$, as defined in Lemma 4.3. Let us show that the second condition is met by the hybrid control law $u(x,k,m)$ with respect to $x_d$ for $m \in \{-1,1\}$ and $k \in \mathbb{I}$ $\big($*i.e.,* $u(x,k,m)$ minimizes the deviation with respect to the nominal direction $(x_d - x)\big)$. Since the velocity vector $u(x,k,m)$ ensures a minimum angle with the nominal direction to the virtual destination $(x_k^m - x)$ given by $\angle(x_k^m - x, u(x,k,m)) = \arccos(\frac{\kappa(x,k,m)^\top (c_k - x)}{\|\kappa(x,k,m)\|\|c_k - x\|}) = \theta(x,k) - \beta(x,k,m)$, then one has to show that $\angle(x_d - x, u(x,k,m)) = \theta(x,k) - \beta(x,k,0)$ $\big($or equivalently $\frac{\kappa(x,k,m)^\top (x_d - x)}{\|\kappa(x,k,m)\|\|x_d - x\|} = \cos(\theta(x,k) - \beta(x,k,0))\big)$ where $\beta(x,k,0) = \angle(x_d - x, c_k - x)$. Hence, $\frac{\kappa(x,k,m)^\top (x_d - x)}{\|\kappa(x,k,m)\|\|x_d - x\|} = \frac{\sin(\theta(x,k))}{\sin(\beta(x,k,m))}\big(\frac{(x_k^m - x)^\top (x_d - x)}{\|x_k^m - x\|\|x_d - x\|} - \frac{\sin(\theta(x,k) - \beta(x,k,m))}{\sin(\theta(x,k))}\frac{(c_k - x)^\top (x_d - x)}{\|c_k - x\|\|x_d - x\|}\big)$. Since $x_k^m \in \mathcal{PL}(x_d, c_k, x(t_0^k, j_0^k))$ if $x(t_0^k, j_0^k) \notin \mathcal{L}(x_d, c_k)$ $\big($resp. $x_k^m \in \mathcal{PL}(x_d, c_k, y)$, where $y \in \mathbb{R}^n \setminus \mathcal{L}(x_d, c_k)$, if $x(t_0^k, j_0^k) \in \mathcal{L}(x_d, c_k)\big)$, and the *obstacle-avoidance* mode is selected such that the associated virtual destination is the closest to the position $x$, then $\angle(x_k^m - x, x_d - x) = \beta(x,k,m) - \beta(x,k,0) \geq 0$. Thus, $\frac{\kappa(x,k,m)^\top (x_d - x)}{\|\kappa(x,k,m)\|\|x_d - x\|} = \frac{\sin(\theta(x,k))}{\sin(\beta(x,k,m))}\big(\cos(\beta(x,k,m) - \beta(x,k,0)) - \frac{\sin(\theta(x,k) - \beta(x,k,m))}{\sin(\theta(x,k))}\cos(\beta(x,k,0)\big) = \cos(\theta(x,k) - \beta(x,k,0))$. The velocity vector $u(x,k,m)$ is tangent obstacle $\mathcal{O}_k$ and ensures a minimum angle with the nominal direction $(x_d - x)$ for all $x \in \mathcal{F}_k^m \cap \mathcal{B}(c_k, r_k + \bar{r}_k - \epsilon)$, $m \in \{-1,1\}$ and $k \in \mathbb{I}$. Therefore, one can conclude that the control $u(x,k,m)$ generates local optimal obstacle avoidance maneuvers.