SENSOR-BASED MOTION PLANNING VIA NONSMOOTH ANALYSIS

by

Shawn Rusaw

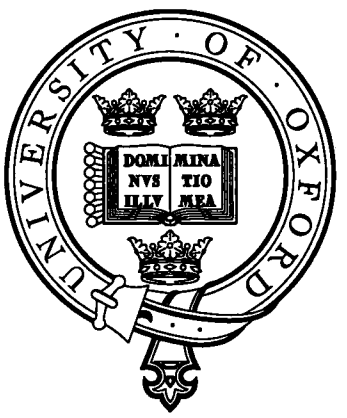
B.Sc. Engineering Physics, University of Saskatchewan, Canada, 1994

M.A.Sc. Engineering Science, Simon Fraser University, Canada, 1998

A THESIS SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY IN COMPUTING



Abstract

In this thesis we present a novel approach to sensor-based motion planning developed using the mathematical tools provided by the field of nonsmooth analysis. The work is based on a broad body of background material developed using the tools of differential topology (smooth analysis), that is limited to simple cases like a point or circular robot. Nonsmooth analysis is required to extend this background work to the case of a polygonal robot moving amidst polygonal obstacles.

We present a detailed nonsmooth analysis of the distance function for arbitrary configuration spaces and use this analysis to develop a planner for a rotating and translating polygonal mobile robot. Using the tools of nonsmooth analysis, we then describe a one-dimensional nonsmooth roadmap of the robot's freespace called the *Nonsmooth Critical Set + Nonsmooth Generalised Voronoi Graph* (NCRIT+NGVG) where the robot is equidistant to a number of obstacles, in a critical configuration or passing between two obstacles. We then use the related field of nonsmooth control theory to develop several provably stable control laws for following and exploring the nonsmooth roadmap. Finally, we implement a motion planner in simulation and for a real polygonal mobile robot, thus verifying the utility and practicality of the nonsmooth roadmap.

Acknowledgements

The submission of this thesis marks the end of a five year journey. In September 1996, after two years working as a systems administrator, I returned to University to pursue a Masters degree with the full support of my family and then girlfriend Jodi. My experiences during my two years at Simon Fraser University were overwhelmingly positive, in most part due to my two fantastic supervisors Kamal Gupta and Sharam Payandeh. Their encyclopedic knowledge of robotics, penchant for asking incisive questions, and in particular, Kamal's enthusiasm for algorithmic robotics and motion planning, inspired me to pursue a Doctorate in Computer Science at Oxford.

They say that moving to another city and getting married are two of the most stressful events life has to offer, so adding a thesis defence, and doing all three in one month was a very ill-advised move. By the time we arrived in Oxford I was completely stressed out. Compounding this pain was the fact that we had no place to live for our first three weeks in Oxford. Things were not starting out very auspiciously. Bailing us out of our initial housing problem was our good friend Claire Oldaker, a Brit we met in Vancouver, who arranged for us to stay at her lovely parents' house. Doug and Leslie Oldaker were lifesavers.

Soon after arriving in Oxford, my extraordinarily enthusiastic and supportive supervisor Stephen Cameron invited Jodi and I to a small dinner party at his home. This friendly gesture was much needed at the time, since we were already feeling a bit homesick after only a few short weeks. Here I met Joe Pitt-Francis, one of Stephen's colleagues, whose office door has been a constant source of amusement. Around the same time, I met with my 'Moral Tutor', Martin Edwards, who turned out to be a good chap to have on my side. Since he was the Dean of The Queen's College, anytime I had financial difficulty, he was first in line to plead my case to the College governing body, which was enormously appreciated.

After 12 months in a terrible office in the 'attic', I eventually settled into Room 006 with my first office-mate Craig Eldershaw. Craig happened to be very knowledgeable about many variants of Unix, and was

a constant source of useful information. Since Craig was also a foreigner (an Australian) at Oxford, we often shared amusing stories about experiences in England. After Craig was finished at Oxford, my next office-mate was Tristam Fenton-May, who helped me a great deal with Linux administration questions. Thanks also to Mary Mahabir, a friend of Jodi's, for helping us with logistic issues when moving and taking us for trips in the English countryside.

I must not forget to mention the friendly and helpful staff at the Computing Laboratory and The Queen's College. Thanks to: Christine O'Connor (Comlab) for answering all my stupid administration questions and offering weekend employment in the Lab on several occasions, Sue Baker (Comlab) for helping with all my payroll concerns, Joe Leggett (Comlab) for supplying plasters and organising the Departmental cricket games, Duncan Oliver (Comlab) for providing jaded and always comical references to the operations of the lab, Peter Turner (Comlab) for helping with all my electronics needs, Terry Brown (Comlab) for helping me with the Windows 2000 machines, Fleur Heughes (Queen's) for providing College signatures for all my student loan applications and general helpfulness, the 'Bursary Ladies' (Queen's) for being so chipper and Annette Goodey (Queen's) for being so enormously helpful with all my housing concerns when we staying in College accommodation.

Special thanks go out to Francis Clarke for sparing the time to answer several questions about non-smooth analysis. When I was completely flummoxed by a proof I was working on, his insight proved crucial to overcoming my difficulties. Extra special thanks must go out to Jane Langdale. During a somewhat informal conversation with her I mentioned the imminent arrival of our son Alexander and our less than adequate housing situation. She took it upon herself to convince the Home Bursar to allow us to rent one of the College houses, even though these are normally reserved for first year students. Having the extra space really made the adjustment to parenthood a bit smoother.

I want to thank my parents Charles and Lynda for their unconditional love and support, without which I would certainly have never have accomplished or experienced anywhere near as much as I have. And most importantly, I'd like to thank my wife Jodi for accompanying and supporting me on this journey. I love you.

This thesis was typeset using LaTeX.

Dedication

This thesis is dedicated to Jodi and Alexander.

Contents

List of Figures

List of Tables

Nomenclature

s, $\mathbf{v}$ Normal weight indicates a scalar or matrix and bold indicates a vector.

$\mathbf{v} = [v_x, v_y, v_z]$ Vector components are indicated with subscripts.

$\mathbf{v}_1\mathbf{v}_2 = \mathbf{v}_1 - \mathbf{v}_2$ Commonly used short form of vector difference.

$\mathbf{a}$, $\mathbf{a}_i$ The closest (witness) point on the robot $\mathcal{A}$ to the obstacle $\mathcal{B}$, indexed if there are multiple obstacles $\mathcal{B}_i$. $\mathbf{a}$ can also be the follow term of the control system.

$\mathcal{A}$, $\mathcal{A}(\mathbf{q})$ The set of points occupied by the robot, in configuration $\mathbf{q}$.

$\mathcal{A}_{ijk}$ An accessibility region for obstacles $\mathcal{B}_i$, $\mathcal{B}_j$ and $\mathcal{B}_k$.

$\mathbf{b}$, $\mathbf{b}_i$ The closest (witness) point on the obstacle $\mathcal{B}$ to the robot $\mathcal{A}$, indexed if there are multiple obstacles $\mathcal{B}_i$. $\mathbf{b}$ can also be the correction term of the control system.

$\mathcal{B}$, $\mathcal{B}_i$ The set of points occupied by an obstacle, indexed accordingly if there are multiple obstacles.

bndy(S) The boundary of the set S.

C Configuration space.

$C\mathcal{B}$, $C\mathcal{B}_i$ The set of points in configuration space C occupied by a configuration space obstacle, indexed accordingly if there are multiple configuration space obstacles.

C_{free} The portion of configuration space C not occupied by configuration space obstacles $C\mathcal{B}_i$—freespace.

cl(S) The closure of the set S.

co{S} The convex hull of the set S.

C_r Reachable configuration space in a bounded workspace.

$D(\mathbf{q})$ The minimum distance function. $D(\mathbf{q}) = \min_i\{d_i(\mathbf{q})\}$.

$d_i(\mathbf{q})$ The single object distance function between the convex robot $\mathcal{A}$ and the convex obstacle $\mathcal{B}_i$.

$d_{\mathcal{A}}(\theta)$ The diameter function for the convex polygon $\mathcal{A}$.

E_{ϑ} The set of minima and maxima of $d_{\mathcal{A}}(\theta)$. ϑ denotes a single minimum or maximum.

$\mathcal{D}_{ij}^{\vartheta}$, $\mathcal{D}_{ijk}^{\vartheta}$ A diameter extrema edge and a diameter extrema node. $\mathcal{D}_{ijk}^{\vartheta} = \mathcal{D}_{ij}^{\vartheta} \cap \mathcal{R}_{GVG}$.

$\dim(S)$ The dimension of the set S.

$\text{dist}(\mathbf{x}_1, \mathbf{x}_2)$ The distance between two points $\mathbf{x}_1$ and $\mathbf{x}_2$.

$f: X \to Y$ A map taking points in the domain X to the range Y.

$\mathcal{F}_{\mathcal{A}}$ The reference frame of the robot $\mathcal{A}$.

$\mathcal{F}_{\mathcal{W}}$ The world (workspace) reference frame.

GVG The Generalised Voronoi Diagram.

I Identity matrix.

$\text{int}(S)$ The interior of the set S.

K, $K_{f(\mathbf{x})}$ The set of critical points of the function $f(\mathbf{x})$.

K_c The set of critical points of the function $f(\mathbf{x})$ satisfying $f(\mathbf{x}) = c$.

K_{ij}, K_{ij}^{ϑ} The set of critical configurations ($K_{ij} \subseteq K$) where the robot is equidistant to obstacles $\mathcal{B}_i$ and $\mathcal{B}_j$ and the subset ($K_{ij}^{\vartheta} \subseteq K_{ij}$) where $\mathcal{A}$ is oriented in diameter extrema ϑ.

$\min S$, $\min_i S$ The minimum element of the set S, indexed if S has a finite number of elements.

$N_S(\mathbf{x})$ Normal cone to the set S at point $\mathbf{x}$.

$\text{nbhd}(S)$ A neighbourhood of the set S.

NCRIT+NGVG The Nonsmooth Critical Set + Nonsmooth Generalised Voronoi Diagram.

$A^{\dagger}$ The Penrose pseudo-inverse of the matrix A.

$\text{Null}(A)$ The nullspace of the matrix A.

P, $\tilde{P}$ The set of closest points between the robot $\mathcal{A}$ and the obstacle $\mathcal{B}$, and the subset $\tilde{P} \subseteq P$ of extremal points (vertices of P).

π_S The projection operator onto a set S.

$\mathbf{q}$ The configuration of the robot $\mathcal{A}$. In $SE(2)$ this corresponds to the $\mathbf{q} = (x, y, \theta)$ coordinates and in $SE(3)$ to the $\mathbf{q} = (x, y, z, \phi, \theta, \psi)$ coordinates.

$\mathbf{q}_s$, $\mathbf{q}_g$ The start and goal configurations.

$R_{\phi,\theta,\psi}$ The composite rotation matrix composed of the Euler rotations R_{ϕ},R_{θ} and R_{ψ}.

$\mathbb{R}$ The real numbers.

$\mathcal{R}$ The roadmap. $\mathcal{R} = \mathcal{R}_{GVG} + \mathcal{R}_K + \mathcal{R}_D$.

$\mathcal{R}_{GVG}$ The set of configurations where the robot $\mathcal{A}$ is equidistant to three (roadmap segment) or four (roadmap node) obstacles. $\mathcal{R}_{GVG} = \left(\cup_{i,j,k} \mathcal{IF}_{ijk}\right) \cup \left(\cup_{i,j,k,l} \mathcal{IF}_{ijkl}\right)$.

$\mathcal{R}_K$ The set of critical configurations.

$\mathcal{R}_D$ The set of diameter extrema edges. $\mathcal{R}_D = \cup_{i,j,\vartheta} \mathcal{D}_{ij}^{\vartheta}$.

S^1 The unit circle.

$SE(2)$ $SE(2)$ is equivalent to $\mathbb{R}^2 \times SO(2)$.

$SE(3)$ $SE(3)$ is equivalent to $\mathbb{R}^3 \times SO(3)$.

$SO(n)$ The Special Orthogonal Group of $n \times n$ rotation matrices in $\mathbb{R}^{n^2}$ with orthonormal columns/rows and a determinant = +1.

$\tau(t)$ A path in configuration space.

U The location of the witness point $\mathbf{a}$ in the coordinate frame $\mathcal{F}_{\mathcal{A}}$.

$\mathcal{W}$ The workspace of the robot $\mathcal{A}$.

$\mathcal{S}_{ij}, \mathcal{S}_{ijk}, \mathcal{S}_{ijkl}$ The sets of configurations where the robot $\mathcal{A}$ is equidistant to $\mathcal{B}_i$ and $\mathcal{B}_j$ (and $\mathcal{B}_k$ and $\mathcal{B}_l$ if appropriate).

$\mathcal{IS}_{ij}, \mathcal{IS}_{ijk}, \mathcal{IS}_{ijkl}$ The invertible sets of configurations where the robot $\mathcal{A}$ is equidistant to $\mathcal{B}_i$ and $\mathcal{B}_j$ (and $\mathcal{B}_k$ and $\mathcal{B}_l$ if appropriate).

$\mathcal{IF}_{ij}, \mathcal{IF}_{ijk}, \mathcal{IF}_{ijkl}$ The invertible sets of configurations where the robot $\mathcal{A}$ is equidistant and closest to $\mathcal{B}_i$ and $\mathcal{B}_j$ (and $\mathcal{B}_k$ and $\mathcal{B}_l$ if appropriate).

$\partial f(\mathbf{x})$ The generalised gradient of $f(\mathbf{x})$.

$\nabla f(\mathbf{x})$ The gradient of (smooth) $f(\mathbf{x})$.

$f^\circ(\mathbf{x}; \mathbf{e})$ The generalised directional derivative of $f(\mathbf{x})$ in direction $\mathbf{e}$.

$f'(\mathbf{x}; \mathbf{e})$ The directional derivative of $f(\mathbf{x})$ in direction $\mathbf{e}$.

Z_c The set of points of the function $f(\mathbf{x})$ satisfying $f(\mathbf{x}) = c$.

$f|_S$ The function f restricted to the set S.

$\langle \mathbf{v}_1, \mathbf{v}_2 \rangle$ The inner product of $\mathbf{v}_1$ and $\mathbf{v}_2$. Throughout the paper, the inner product is equivalent to the dot product $\mathbf{v}_1 \cdot \mathbf{v}_2$ and the multiplication $\mathbf{v}_1 \mathbf{v}_2^T$.

$\{\mathbf{x}_k\}$, $\mathbf{x}_k$ A sequence and an element of the sequence.

$A \setminus B$ Difference of set A and set B.

$A \cup B$ Union of set A and set B.

$A \cap B$ Intersection of set A and set B.

$A \Rightarrow B$ Material Implication—A implies B.

$A \Leftrightarrow B$ Mutual Implication—A implies B, B implies A.

$\forall A \exists B$ For all A there exists a B.

$A \subseteq B$ A is contained in, or equal to, B.

§1

"The beginning is the most important part of the work."

Plato

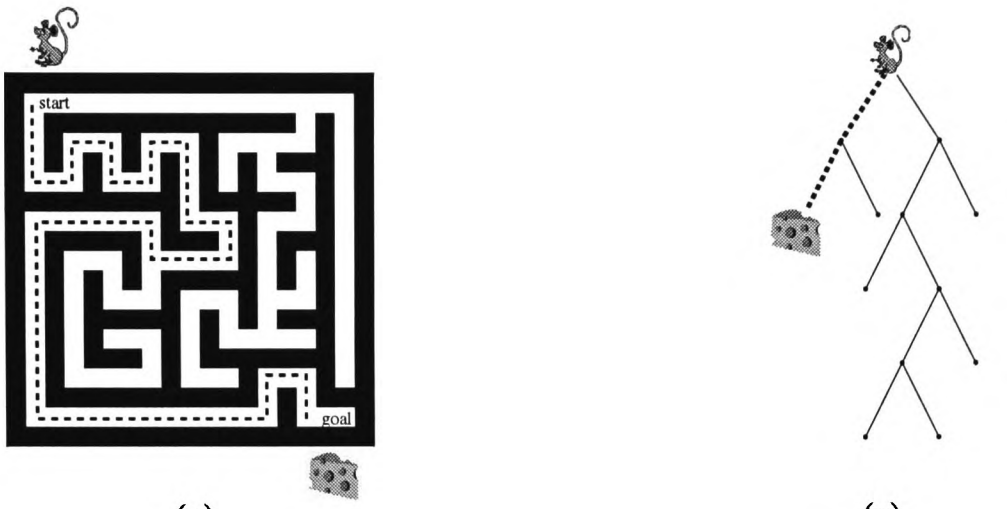
Introduction

The robot motion planning problem is to determine a suitable path for a robot that avoids collisions with obstacles in its environment. Consider the maze shown in Figure 1.1(a). In the top left corner is a robotic mouse and in the bottom right corner is a piece of cheese representing the goal of the mouse. Robot motion planning is the field that answers "How do we program the mouse to find the cheese?" More formally, the motion planning problem for a robot $\mathcal{A}$ in an environment $\mathcal{W}$ occupied by a finite number of obstacles $\mathcal{B}_i$ can be stated as:

Definition 1.1 (Latombe [38]) Given an initial position and orientation (configuration) $\mathbf{q}_s$ and a goal configuration $\mathbf{q}_g$ of the robot $\mathcal{A}$, generate a path τ specifying a continuous sequence of configurations of $\mathcal{A}$ avoiding contact with the obstacles $\mathcal{B}_i$, starting at $\mathbf{q}_s$ and terminating at $\mathbf{q}_g$. Report failure if no path exists. □

An answer to the mouse and cheese example which should be familiar to most readers is the use of a graph structure to systematically examine all pathways (see Figure 1.1(b)). However, now consider the mouse in the environment shown in Figure 1.2 in which we have a bounded area filled with several differently shaped obstacles. In this more general case, the direct use of a graph search is not immediately apparent and so more complex algorithmic tools are required.

Of course we are not always programming robotic mice, and in fact the motion planning problem is a large class of problems. First consider the case of a mobile robot moving about in a laboratory. If the



(A) (B)

FIGURE 1.1: HOW DO WE PROGRAM THE MOUSE TO FIND THE CHEESE? (A) A MAZE. (B) THE GRAPH USED TO FIND THE PATH.

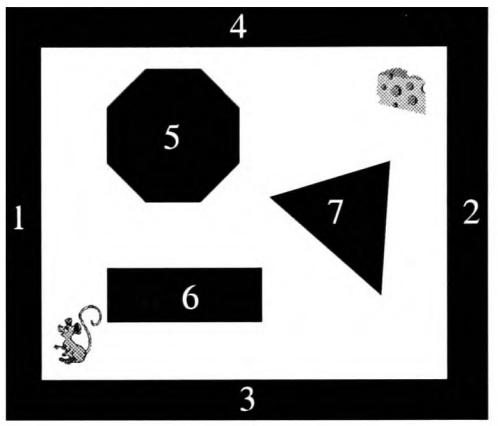
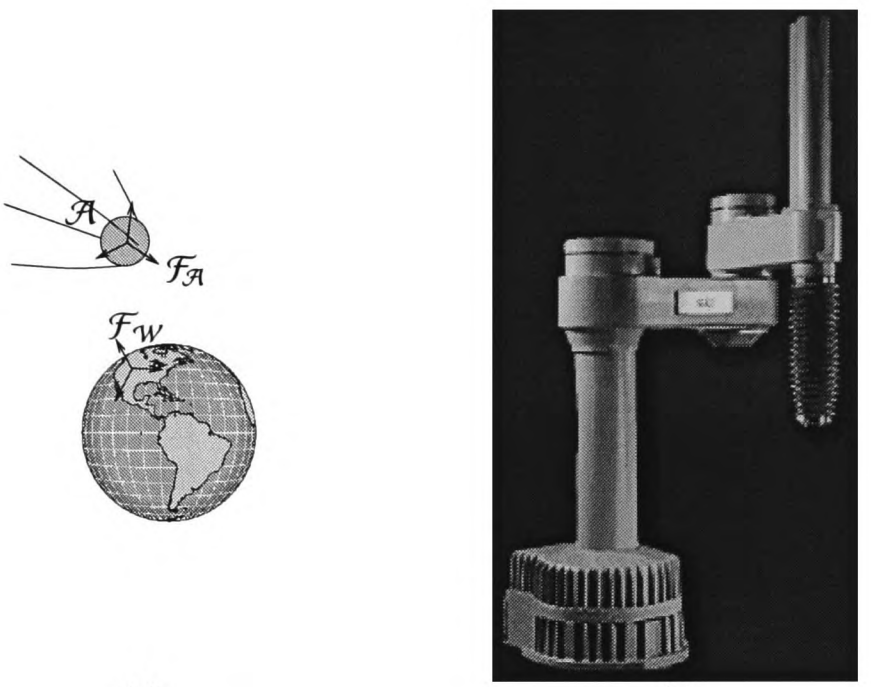


FIGURE 1.2: MORE GENERAL PATH PLANNING PROBLEM. HOW DO WE PROGRAM THE MOUSE TO FIND THE CHEESE? IT IS NOT AS OBVIOUS AS IN THE CASE OF THE MAZE IN FIGURE 1.1.

goal/target of the robot is to move across the lab, then it must be given the ability to avoid tables and chairs on its traversal of the room. One method to do this, which is referred to as *classical motion planning*, is to pre-program the location of all the obstacles and walls before planning. So, when it comes time to plan a path across the room, the planner need only concern itself with the *a priori* known location of obstacles. However, if an unexpected obstacle appears in a location the robot does not expect (e.g. a chair falls over or a person gets in the way) the planner may fail to avoid that obstacle. To enable the robot to avoid unexpected obstacles we need to equip it with sensors and adjust the planner to incorporate sensor information in the planning process. This is an example of a *sensor-based motion planning*.



(A) (B)

FIGURE 1.3: TWO EXAMPLES OF ROBOTS WITH DIFFERENT CONFIGURATION SPACES. (A) A FREE-FLOATING SATELLITE. (B) AN ADEPTONE ROBOT.

1.1 The Configuration Space

The concept of configuration space was introduced by Lozano-Pérez and Wesley [45]. The basic notion of the configuration space was to transform the problem of planning the motion of a physical robot in a physical workspace to planning the motion of a one-dimensional point representing the configuration of the robot around an n-dimensional space populated by mathematical transformations of the physical obstacles (n is the number of degrees of freedom of the robot).

Consider a free-floating body such as the satellite $\mathcal{A}$ in Figure 1.3(a). If we attach a coordinate frame $\mathcal{F}_{\mathcal{A}}$ to the satellite, then its position with respect to another fixed (world) coordinate frame $\mathcal{F}_{\mathcal{W}}$ is given by six numbers, three representing translational motion x, y, z and three representing rotational motion ϕ, θ, ψ[1]. Since its position is uniquely defined by six numbers, the satellite is said to have six *degrees of freedom* (DOF). Next consider the configuration space of a multi-link manipulator in Figure 1.3(b). The position and orientation of the end effector of a n-link manipulator is given by the angles of the joints $\theta_1, .., \theta_n$. So, the configuration space is n-dimensional and points in the space correspond to a unique orientation of the robot's end effector. A path in the configuration space corresponds to a continuous motion of the robot.

Obstacles $\mathcal{B}_i$ in the environment impose additional constraints on the motion of the robot such that $\mathcal{A} \cap \mathcal{B}_i = \varnothing$. That is, the robot cannot interpenetrate any of the obstacles. Physical obstacles $\mathcal{B}_i$ are trans-

[1]The Euler angles are a common representation of the three rotational degrees of freedom

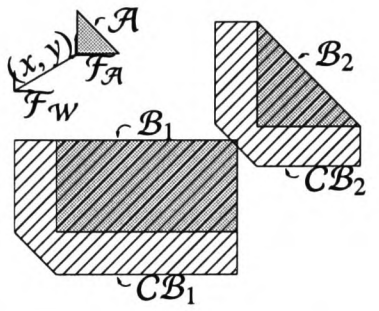


FIGURE 1.4: CONFIGURATION SPACE FOR A TRANSLATING TRIANGULAR ROBOT.

formed to n-dimensional forbidden regions in configuration space called *configuration space obstacles* and denoted $C\mathcal{B}_i$. The region of configuration space not occupied by configuration space obstacles is referred to as *free space* C_{free}. Motion planning consists of finding a continuous path $\tau(t)$ that lies entirely in free space C_{free} from the *start* configuration $\mathbf{q}_s$ to the *goal* configuration $\mathbf{q}_g$.

Two examples of configuration space are given in Figures 1.4 and 1.5. Figure 1.4 shows a two-dimensional configuration space for a triangular mobile robot. In this case, the robot can only translate with its position given by (x, y), so the configuration space is $\mathbb{R}^2$. The configuration space obstacles in this case are equal to the *Minkowski difference* of the obstacles $\mathcal{B}_i$ and the robot $\mathcal{A}$.

$$C\mathcal{B}_i = \mathcal{B}_i \ominus \mathcal{A} = \{x \in \mathcal{W} / \exists b \in \mathcal{B}_i, \exists a \in \mathcal{A} : x = b - a\}$$

The Minkowski difference operator can be thought of as "gluing" a reflected copy of the robot to every point of the obstacle. The resulting configuration space obstacles (hashed regions) represent forbidden regions for the origin of the robot's coordinate frame. Notice that the robot cannot pass between $\mathcal{B}_1$ and $\mathcal{B}_2$ since $C\mathcal{B}_1$ and $C\mathcal{B}_2$ are overlapping. Figure 1.5(b) shows a two-dimensional configuration space for the two-link planar manipulator shown in Figure 1.5(a). In this case, each joint can take a range of angles given by $S^1 = [0, 2\pi)$, using modulo 2π arithmetic to deal with wrapping joint angles. Since there are two joints, the configuration space is given by $S^1 \times S^1$—the surface of a torus. Figure 1.5(b) shows a "flattened" configuration space. In both cases, the configuration space obstacles $C\mathcal{B}_1$ and $C\mathcal{B}_2$ represent those configurations where the robot $\mathcal{A}$ is colliding with the workspace obstacles $\mathcal{B}_1$ and $\mathcal{B}_2$. Finally, note that the configuration space obstacles in these figures are of dramatically different shape, indicating the different nature of the robots' configuration spaces.

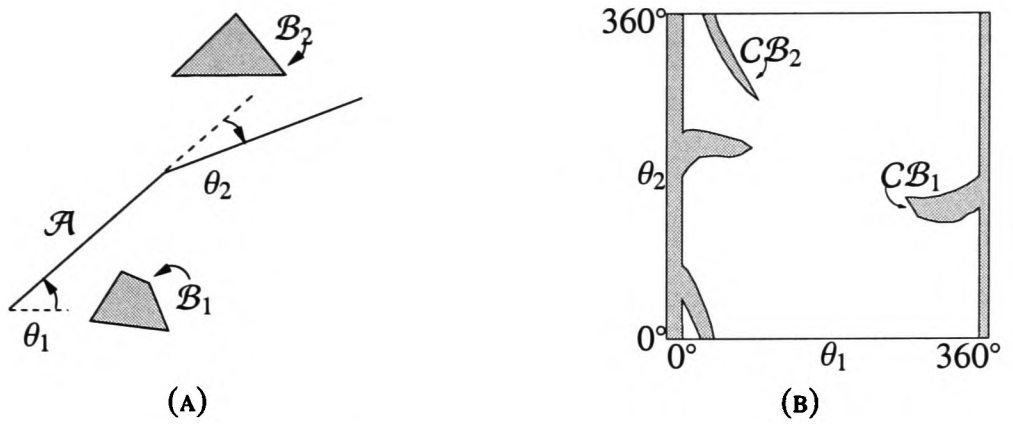


FIGURE 1.5: CONFIGURATION SPACE OF A TWO-LINK MANIPULATOR. (A) THE PLANAR MANIPULATOR $\mathcal{A}$ AND TWO OBSTACLES $\mathcal{B}_i$ AND $\mathcal{B}_j$. (B) THE 'FLATTENED' CONFIGURATION SPACE (GENERATED FROM A BITMAP REPRESENTATION OF SOME OF OUR EARLY EXPERIMENTAL RESULTS).

1.2 Introduction to Motion Planning Techniques

In this section we introduce techniques used in classical and sensor-based motion planning. Section 1.2.1 introduces the major classes of classical motion planning and Section 1.2.2 outlines various techniques that have been used for sensor-based motion planning. Section 1.2.3 discusses sensor-based motion planning in the context of distance measurements and sets the stage for the analysis and algorithm presented in this thesis. We first begin with a definition of *completeness*.

Definition 1.2 A motion planning algorithm is **complete** if it is guaranteed to find a solution if one exists and otherwise indicate that no path exists. This contrasts with **heuristic** planners which cannot be guaranteed to find a solution, even if one exists. □

This thesis deals primarily with complete motion planners since the planner developed in subsequent chapters is complete. The terms *complete*, *exact* and *non-heuristic* are used interchangeably in the motion planning literature.

1.2.1 Classical Motion Planning

The classical motion planning problem has historically been called the *Piano Mover's Problem* or the *Generalised Mover's Problem* in reference to the problem movers have when trying to move furniture from room to room. For example, how do the movers get a grand piano through a narrow doorway? In this type of problem, the location of obstacles is known *a priori*. The planner thus has full knowledge of the environment and is able to plan accordingly.

The first step of many classical motion planners is to generate the configuration space for the problem at hand, given the geometry and location of the robot and obstacles. At this point the motion planning problem is reduced to planning the motion of a point $\mathbf{q}$ amidst n-dimensional obstacles $C\mathcal{B}_i$. The most influential planners have been based on three basic methodologies: (i) *potential fields*, (ii) *cellular decompositions* of free space, and (iii) *roadmaps* of free space. Potential based planners typically operate by applying an artificial potential to the point $\mathbf{q}$ that tends to draw the robot toward the goal while at the same time pushing it away from obstacles. Cellular decomposition-based planners divide the free space into a number of cells. Since the connectivity of these cells is easily determined, motion planning is reduced to the problem of searching the connectivity graph from the cell containing the start $\mathbf{q}_s$ to the cell containing the goal $\mathbf{q}_g$. Roadmap-based planners generate a one-dimensional roadmap which represents the topology of free space, as demonstrated by the following formal definition

Definition 1.3 (Choset [10]) A **roadmap** $\mathcal{R}$ is a union of one-dimensional curves such that for all start and goal configurations $\mathbf{q}_s, \mathbf{q}_g \in C_{free}$, there exists a path between $\mathbf{q}_s$ and $\mathbf{q}_g$ if and only if

1. There exists a path from $\mathbf{q}_s \in C_{free}$ to some $\mathbf{q}'_s \in \mathcal{R}$ (*accessibility*),
2. There exists a path from $\mathbf{q}_g \in C_{free}$ to some $\mathbf{q}'_g \in \mathcal{R}$ (*departability*),
3. There exists a path in $\mathcal{R}$ from $\mathbf{q}'_s$ to $\mathbf{q}'_g$ (*connectivity*). □

This thesis is concerned with the construction of a roadmap based on sensed distance measurements.

As a general rule (counter examples always exist) potential-based planners are not complete while roadmap and cellular decomposition-based planners are. The primary advantage of the potential-based planners is they tend to be much faster than the other classes of planner for large degree-of-freedom robots, since the time complexity to generate the roadmap/decomposition is exponential in the number of degrees of freedom.

A number of subclasses of the mover's problem exist. These include cases where the obstacles are moving, where there are multiple robots, and where the robot is able to move the obstacles. These problems are usually handled by modifying the definition of configuration space to include dynamic quantities like time or velocity. This thesis is only concerned with the case of a finite number of stationary obstacles and a single robot.

1.2.2 Sensor-Based Motion Planning

If the robot does not know the location of obstacles in its workspace, then it must be augmented with sensors that allow it to determine the state of its environment. The sensor-based planner incorporates this information into the planning process. Sensor-based planners fall into two main categories: *heuristic* and *non-heuristic*. A non-heuristic planner is defined as one that is *provably* complete. Heuristic planners on the other hand are not provably complete in that they may fail to find a path, even if one exists. Heuristic planners often equip the robot with a set a behaviours (follow a wall, avoid cluttered spaces) but tend to fail when they encounter some obstacle configuration the designer didn't envision.

Complete sensor-based planners are guaranteed to find a path to the goal if one exists. Classical motion planners have this ability because they are able to look at the environment as a whole, but sensor-based planners typically only have access to those parts of the environment that they have already explored. The crux of the problem is to sense *enough* of the environment to ensure completeness. The most influential planners have been based on three basic methodologies: (i) *potential-based* planners, (ii) *bug-like* planners and (iii) *roadmap generating* planners. The potential-based planners typically generate a potential field with a single global minimum generated from sensed obstacles. As more obstacles are sensed, the potential is updated, always ensuring a single global minimum exists that drives the configuration point toward the goal. The bug-like algorithms model the robot as a point and equip it with simple behaviours, combined with a higher level of control that tells the robot when to switch between the behaviours. The roadmap generating planners use sensor input to construct a roadmap that represents the connectivity of free space. Planners of this type typically use stereo imaging cameras, range finders and/or sonar to detect obstacles and generate a roadmap based on the obstacles' locations. One possible variant in $\mathbb{R}^2$ [65] is to use the sensors to follow obstacle boundaries and detect which obstacles are within line of sight of each other. A roadmap may then be created that consists of boundary segments and segments connecting mutually visible obstacles. The resulting roadmap can then be used for motion planning. One drawback of such a technique is that portions of the roadmap corresponding to boundary segments tend to guide the robot unnecessarily close to the obstacles. In terms of safety and path robustness, it is generally favourable to maximise the distance between the robot and the obstacles. We term the sensor-based planning methods that satisfy this requirement *distance-based* roadmaps. This thesis is devoted to the development of a complete distance-based roadmap for planning the motion of a convex polygonal mobile robot with a configuration space of $SE(2)$.

1.2.3 Motion Planning and Distance

A number of algorithms exist that construct a roadmap based on workspace distance measurements [60, 59, 12, 13, 40, 15, 34]. These algorithms typically use workspace distance measurements to construct a one-dimensional roadmap that tends to maximise the distance between the robot and environmental obstacles. Rimon and Canny present one such planner [60, 59] which was the first provably correct sensor-based planner for arbitrarily large configuration spaces. This algorithm differs quite dramatically from the method used in this thesis, so a detailed description is left until Section 2.2.2. Our technique is most closely related to the work of Generalised Voronoi Diagram (GVG) work of Choset et. al. [12, 13, 15, 40], so we begin with a short introduction of this work.

The main building block of our analysis and algorithm is the *single object distance function*

$$d_i(\mathbf{q}) = \min_{\substack{\mathbf{a}\in\mathcal{A}(\mathbf{q}) \\ \mathbf{b}\in\mathcal{B}_i}} \|\mathbf{a}-\mathbf{b}\|$$

which is the minimum distance between the convex robot $\mathcal{A}$ and a single convex obstacle $\mathcal{B}_i$. In [12, 13] and in this thesis, the GVG roadmap is constructed from sets of configurations equidistant to several obstacles—the number of obstacles required to generate a one-dimensional roadmap depending on the characteristics of the configuration space of the robot. The sets of configurations equidistant to multiple obstacles and the dimension of these sets are determined using the techniques of differential topology [24], which we now discuss in the context of [12, 13] which applies to a *point robot* $\mathcal{A}$ in $\mathbb{R}^2$.

It is known [16] that the distance function between a point and a convex set is smooth, or everywhere differentiable. Because of this differentiability, the set $M_i = \{(\mathbf{q}, r) \in C \times \mathbb{R} : d_i(\mathbf{q}) = r\}$ (the graph of $d_i(\mathbf{q})$) is actually a smooth manifold of dimension $\dim(C)$ embedded in a space of dimension $\dim(C) + 1$. Figure 1.6(a) shows graphs of the seven single object distance functions $d_1(\mathbf{q}), \cdots, d_7(\mathbf{q})$ for the seven convex obstacles $\mathcal{B}_1, \cdots, \mathcal{B}_7$ in Figure 1.2 (point robot $\mathcal{A}$). These graphs are two-dimensional smooth manifolds embedded in a three-dimensional space. We can determine the set of points equidistant to two obstacles $\mathcal{B}_i$ and $\mathcal{B}_j$, and the dimension of this set, by considering the intersection of M_i and M_j. As shown in Figure 1.7(a), the intersection of two smooth manifolds M_i and M_j with dimension $\dim(M_i) = \dim(M_i) = m$ is a itself a smooth manifold with dimension $\dim(M_i \cap M_j) = m - 1$, as long as they intersect transversely [24]. The graphs of $d_i(\mathbf{q})$ and $d_j(\mathbf{q})$ can be shown to intersect transversely if $\nabla d_i(\mathbf{q})$ and $\nabla d_j(\mathbf{q})$ are linearly independent, which is the case for all the $\mathcal{B}_i$'s in Figure 1.2. The intersection $M_i \cap M_j$ is a set in a space with one more dimension than C so in order to determine the set of configurations equidistant to

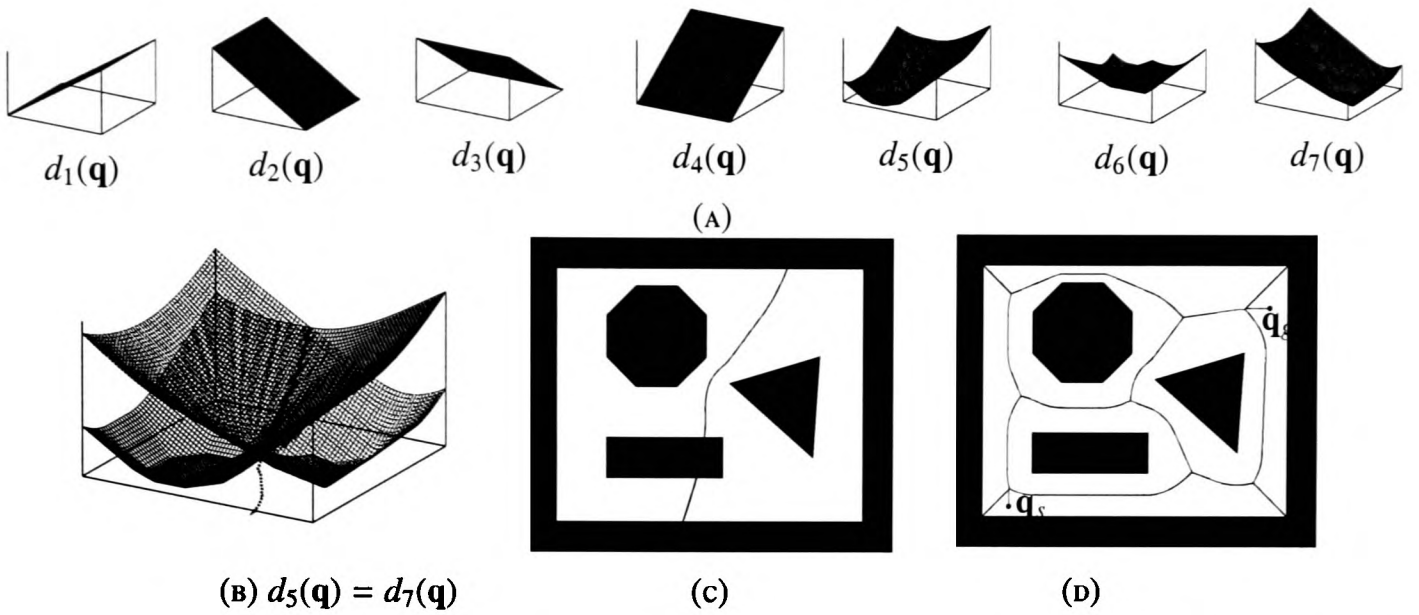


FIGURE 1.6: POINT DISTANCE $d_i(\mathbf{q})$ FROM THE SEVEN INDIVIDUAL CONVEX OBSTACLES. (A) THE DISTANCE FUNCTION FOR ALL SEVEN OBSTACLES $\mathcal{B}_1, \cdots, \mathcal{B}_7$. (B) THE SEVEN CONVEX OBSTACLES AND THE SET OF POINTS EQUIDISTANT TO $\mathcal{B}_5$ AND $\mathcal{B}_7$. (C) THE SET $\pi_C(M_5 \cap M_7)$. (D) THE GENERALISED VORONOI DIAGRAM (GVG)

$\mathcal{B}_i$ and $\mathcal{B}_j$ we need to project $M_i \cap M_j$ onto C. Let $\pi_C(M_i \cap M_j)$ denote this projection. In this case, $\dim(M_i) = \dim(M_j) = m = 2$ and thus $\dim(M_i \cap M_j) = \dim(\pi_C(M_i \cap M_j)) = m - 1 = 1$. Thus in $\mathbb{R}^2$, the GVG is constructed of one-dimensional roadmap arcs where the point robot $\mathcal{A}$ is equidistant to two obstacles and zero-dimensional roadmap nodes where $\mathcal{A}$ is equidistant to three or more obstacles.

The intersection of the graphs of $d_5(\mathbf{q})$ and $d_7(\mathbf{q})$ is shown in Figure 1.6(b) and the projection $\pi_C(M_5 \cap M_7)$ is shown in Figure 1.6(c). The projection $\pi_C(M_5 \cap M_7)$ comprises part of the entire GVG shown in Figure 1.6(d). Since the GVG is the set of points *closest* to two or more obstacles, the entire set $\pi_C(M_5 \cap M_7)$ is not a part of the GVG—only the part that is closest to $\mathcal{B}_5$ and $\mathcal{B}_7$. The intersection of each graph in Figure 1.6(a) with every other graph results in a one-dimensional roadmap that can be used for motion planning.

This elegant analysis based on the intersection of manifolds turns out to be of little use when considering the single object distance function $d_i(\mathbf{q})$ for the case of a convex obstacle $\mathcal{B}_i$ and a convex robot $\mathcal{A}$ with configuration spaces of $SE(2)$ or $SE(3)$. In these cases, $d_i(\mathbf{q})$ is in fact nonsmooth for a subset of C(see Chapter 3). The sets of points equidistant to several obstacles, and in particular the dimension of these sets, can no longer be determined with an application of differential topology. Consider the intersection of two two-dimensional sets S_i and S_j that are *not* manifolds in Figure 1.7(b). How does one determine the dimension of $S_i \cap S_j$? A primary contribution of this thesis is to show how to determine the dimension of $S_i \cap S_j$, and in turn apply this to developing a one-dimensional roadmap of configuration space.

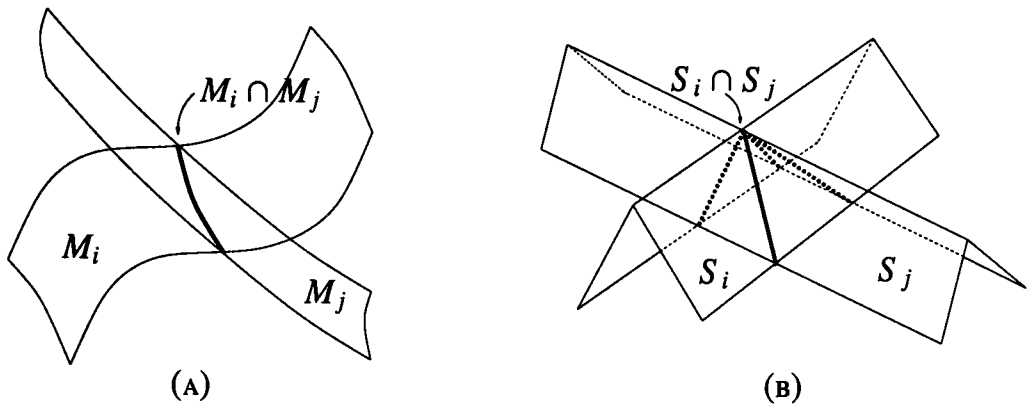


FIGURE 1.7: INTERSECTION OF SMOOTH MANIFOLDS AND NONSMOOTH SETS. (A) THE INTERSECTION OF TWO SMOOTH MANIFOLDS M_1 AND M_2 WITH DIMENSION $\dim(M_1) = \dim(M_2) = m$ IS A ITSELF A SMOOTH MANIFOLD WITH DIMENSION $\dim(M_1 \cap M_2) = m - 1$. (B) WHAT ABOUT THE INTERSECTION OF TWO NONSMOOTH SETS S_1 AND S_2 WITH DIMENSION m?

1.3 Outline of This Thesis

The remainder of the thesis is comprised of a previous work chapter (§2), four chapters that comprise the bulk of our results (§3-§6) and a chapter of conclusions and possible future work (§7). Chapters 3 through 6 have been written in such a way that they progress from purely theoretical to experimental in a smooth way.

Chapter 3 begins the foray into nonsmooth analysis with a large background section, given that most readers are likely unfamiliar with the concepts. We then proceed to develop some tools we use in subsequent chapters, and present a detailed nonsmooth analysis of the distance functions $d_i(\mathbf{q})$. Chapter 4 begins the application of nonsmooth analysis to robot motion planning by carefully describing and outlining the construction of a nonsmooth roadmap which encodes the connectivity of $SE(2)$, thus enabling the development of a complete motion planning algorithm by fully exploring this roadmap. Chapter 5 presents some nonsmooth control theory and uses it to develop stable control laws for exploring the nonsmooth roadmap presented in Chapter 4. Finally, in Chapter 6, we present the results of simulations and experimental trials using a mobile robot. The success of these trials validates the theory presented in the previous chapters, and the experiments with a mobile robot validate the practical utility of the methods outlined.

There are a number of Appendices (§A-§D) that include several important derivations and supporting material. Appendices A through C present a number of mathematical derivations utilising nonsmooth analysis and Appendix D in particular presents an overview of the experimental apparatus used for the experiments described in Chapter 6.

§2

"If I have seen further it is by standing on the shoulders of giants."

Sir Isaac Newton

Relation to Previous Work

2.1 Classical Motion Planning

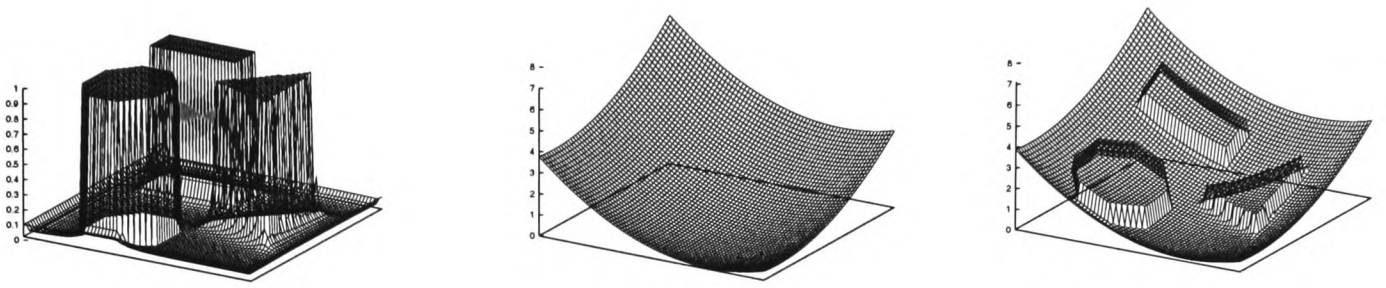
In this section we highlight important previous results when the location of the obstacles is known *a priori*. We address the potential field-based, roadmap-based and cellular decomposition-based methods discussed in Section 1.2.1.

2.1.1 Potential Field-Based Methods

Potential based planners (see Khatib [35] and Latombe [38]) model the robot as a point in configuration space under the influence of an artificial potential function. The potential function $U : C \rightarrow \mathbb{R}$ maps the configuration space to a value representative of the environment. The potential is typically comprised of two parts (i) $U_{rep} : C \rightarrow \mathbb{R}$ which acts to repel the robot from obstacles and (ii) $U_{att} : C \rightarrow \mathbb{R}$ which acts to attract the robot toward the goal. Figure 2.1 gives an example for the environment shown in Figure 1.2. U_{rep} is inversely proportional to the distance from the obstacles and U_{att} is a potential with a single global minimum at the goal. A path is found by gradient descent of $U = U_{tot}$, the sum of U_{rep} and U_{att}.

Potential based planners have a major drawback in that during the gradient descent, the planner has the potential[1] to become trapped in a local minimum. Methods for dealing with local minima have been

[1] sorry...



U_{rep} U_{att} $U_{tot} = U_{rep} + U_{att}$

FIGURE 2.1: POTENTIAL-BASED PATH PLANNER. THE DRIVING POTENTIAL U_{tot} IS A SUM OF A POTENTIAL REPELLING THE ROBOT FROM THE OBSTACLES U_{rep} AND A POTENTIAL ATTRACTING THE ROBOT TO THE GOAL U_{att}. THE PATH CAN BE FOUND WITH A GRADIENT DESCENT OF $U = U_{tot}$.

proposed including randomised escape from local minima (see Latombe [37]) and using a global learning strategy to improve performance over successive trials (see Faverjon and Tournassoud [23]). Connolly [19] presented a novel complete path planning algorithm based on potential functions that are solutions to Laplace's equation—the so called Harmonic Potential Functions. Solutions to Laplace's equation are nice in that they have no local minima, however finding a solution is computationally expensive, requiring an iterative calculation to converge to a solution.

2.1.2 Roadmap-Based Methods

A roadmap $\mathcal{R}$ in free space C_{free} is a collection of one dimensional curves in C_{free} that represent the topological connectedness of C_{free}. That is, a path exists from the start $\mathbf{q}_s \in C_{free}$ to the goal $\mathbf{q}_g \in C_{free}$ if and only if there exists a path from $\mathbf{q}_s$ to $\mathbf{q}_g$ in the roadmap of C_{free}. Once the roadmap is constructed, a standard graph search can be made from $\mathbf{q}_s$ to $\mathbf{q}_g$. If a shortest path is required, then an appropriate graph search is the well-known A* algorithm [25]. There are three primary types of roadmap that are used for motion planning: *visibility graph*, *retraction* and *silhouette*.

2.1.2.1 Visibility Graph

The visibility graph (see Latombe [38] and Lozano-Perez and Wesley [45]) is a graph used for environments with polygonal and polyhedral obstacles. The nodes of the graph consist of vertices of the obstacles as well as the start and goal positions. Arcs of the graph are edges of the obstacles and straight line segments between all vertices that do not pass through any obstacles. An analogy for $\mathbb{R}^2$ and $\mathbb{R}^3$ is that arcs are drawn between nodes that can "see" each other. Figure 2.2 gives an example for the environment shown

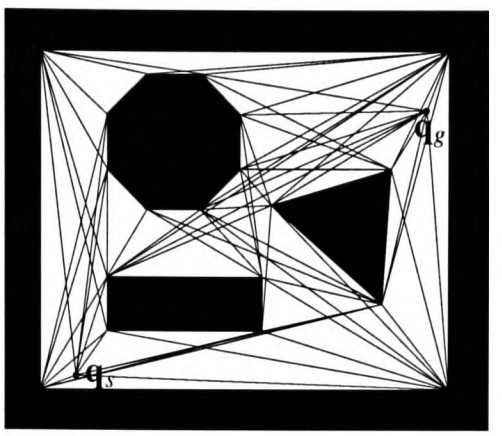


FIGURE 2.2: NODES OF THE ROADMAP CONSIST OF THE VERTICES OF THE OBSTACLES AS WELL AS THE START AND GOAL POSITIONS. ARCS ARE STRAIGHT LINE SEGMENTS CONNECTING THE NODES WHICH DO NOT GO THROUGH THE OBSTACLES.

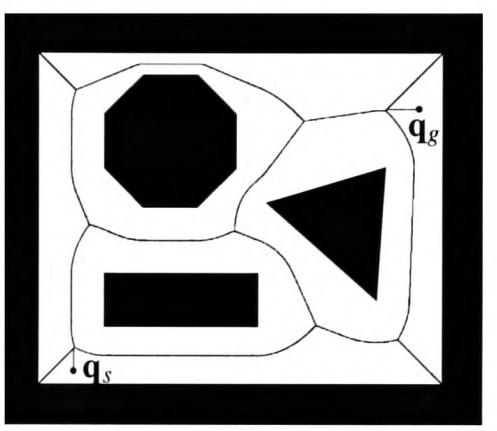


FIGURE 2.3: THE ROADMAP CONSISTS OF ALL POINTS WHOSE MINIMAL DISTANCE TO THE BOUNDARY OF FREE SPACE bndy(C_{free}) IS ACHIEVED WITH TWO OR MORE POINTS IN bndy(C_{free}).

in Figure 1.2. A path is found using a standard graph search from node $\mathbf{q}_s$ to node $\mathbf{q}_g$.

2.1.2.2 Retraction

A retraction of free space C_{free} (see Ó'Dúnlaing and Yap [54, 53]) is a continuous transformation of C_{free} onto a one-dimensional subset $\mathcal{R}$ of C_{free}. The image of the retraction $\mathcal{R}$ is the roadmap.

Figure 2.3 shows a common retraction used for motion planning—the *Voronoi graph*. The Voronoi graph is the set of points whose minimal distance to the boundary of free space bndy(C_{free}) is achieved by two or more points in bndy(C_{free}). In the plane, the graph arcs are straight line segments or parabolic arcs equidistant to two polygon edges or a polygon edge and polygon vertex respectively. Nodes of the graph are points equidistant from three or more different points in bndy(C_{free}). In the plane $\mathbb{R}^2$, the Voronoi graph is always connected.

The most relevant recent development is the Generalised Voronoi Graph (GVG) and Hierarchical

Generalised Voronoi Graph (HGVG) of Choset [12]. The GVG is a one-dimensional roadmap of an n-dimensional free space C_{free}. Defined in a similar way to the Voronoi graph, a point in the GVG is equidistant to n obstacles in a n-dimensional space. When $n = 2$, the GVG and Voronoi graph are one and the same. However when $n > 2$, the GVG is not guaranteed to be connected—it is not a true roadmap. So, the HGVG was defined to recursively connect disconnected parts of the GVG. Although developed for use in a sensor-based planning scheme, these graphs are easily constructed from an *a priori* known environment.

2.1.2.3 Silhouette

The silhouette method was developed by Canny [7]. Given a semi-algebraic formulation of a single configuration space obstacle $\mathcal{CB}$ (possibly obtained from motion constraints as in [38]), the algorithm traces out maxima and minima curves of the so-called "slice function" $h(\mathbf{q}) = \mathbf{q}_i$, where $\mathbf{q}_i$ is one of the coordinates, on the surface of $\mathcal{CB}$. At points called *critical points* these maxima and minima curves may disappear or appear. At critical points, the algorithm is called recursively on a lower dimensional *slice* of the configuration space, ensuring connectivity of the resulting map.

Lin and Canny developed the Opportunistic Path Planner [42, 43] in somewhat the same vein as the silhouette of Canny. However, instead of tracing local extrema on the surface of the configuration space obstacle itself, local maxima are traced on a potential function defined as the minimum distance $D(\mathbf{q})$ from the robot $\mathcal{A}$ to obstacles $\mathcal{B}_1, \cdots, \mathcal{B}_n$ in its environment.

$$D(\mathbf{q}) = \min_{\substack{\mathbf{a}\in\mathcal{A}(\mathbf{q}) \\ \mathbf{b}\in\mathcal{B}_1\cup\mathcal{B}_2\cup\cdots\cup\mathcal{B}_n}} \|\mathbf{a} - \mathbf{b}\| \tag{2.1}$$

The trace of a local maximum is termed a *ridge curve* and the ridge curves lie in connected parts of free space termed *freeways*. The ridge curves alone do not necessarily define a connected roadmap of free space, as they are not necessarily connected. In order to connect disconnected ridge curves, so-called *bridge curves* are added at critical points where freeways split or join. These bridge curves act to connect ridge curves in disconnected freeways. Bridge curves are also added at points where ridge curves appear or disappear—the bifurcation points of the distance potential function. Figure 2.4 gives an example roadmap generated by the Opportunistic Path Planner. This planner has subsequently been modified to enable sensor-based exploration of n-dimensional configuration space [60, 59] (discussed in detail in Section 2.2.2), based on the simplifying assumption that $D(\mathbf{q})$ is smooth, which this thesis clearly shows is not true

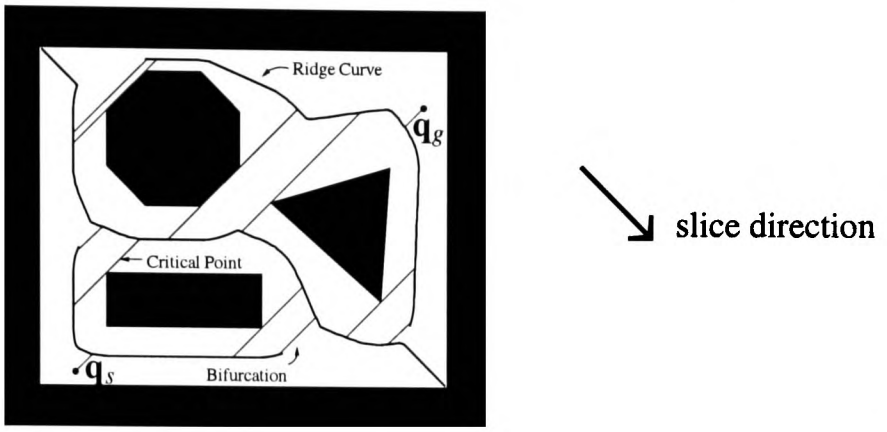


FIGURE 2.4: OPPORTUNISTIC PATH PLANNER ROADMAP. THE RIDGE CURVES ARE THOSE POINTS AT THE MAXIMUM OF THE POTENTIAL FUNCTION AND THE BRIDGE CURVES ARE USED TO LINK POSSIBLY DISCONNECTED RIDGE CURVES. THE BRIDGE CURVES ARE GENERATED AT CRITICAL AND BIFURCATION POINTS.

in general.

2.1.3 Cellular Decomposition-Based Methods

The cellular decomposition methods (see Latombe [38]) split free space C_{free} into a collection of cells. By keeping track of which cells share common boundaries, one constructs a connectivity graph for the environment whose nodes are the cells and arcs correspond to cells sharing a common boundary. A path may be found by searching this graph from the cell (node) that contains the start $\mathbf{q}_s$ to the cell that contains the goal $\mathbf{q}_g$. There are two classes of cellular decomposition method: *exact* and *approximate*.

2.1.3.1 Exact Cellular Decomposition

Exact cellular decomposition methods split the free space into cells whose union is equal to free space C_{free}. One such exact decomposition for an environment with polygonal obstacles is the trapezoidal decomposition (see Latombe [38]). In this method, vertical lines are drawn through the vertices of the obstacles resulting in trapezoidal cells. Figure 2.5(a) shows a trapezoidal decomposition and Figure 2.5(b) shows the resulting connectivity graph. Another similar decomposition (see Chazelle [8]) splits free space into a collection of convex polygonal cells.

The Voronoi diagram in Figure 2.3 can also be interpreted as an exact cellular decomposition. Each cell of the *Generalised Voronoi decomposition* is defined as the set of points nearest to a particular obstacle, and cell connectivity is determined from the edges of the Voronoi graph.

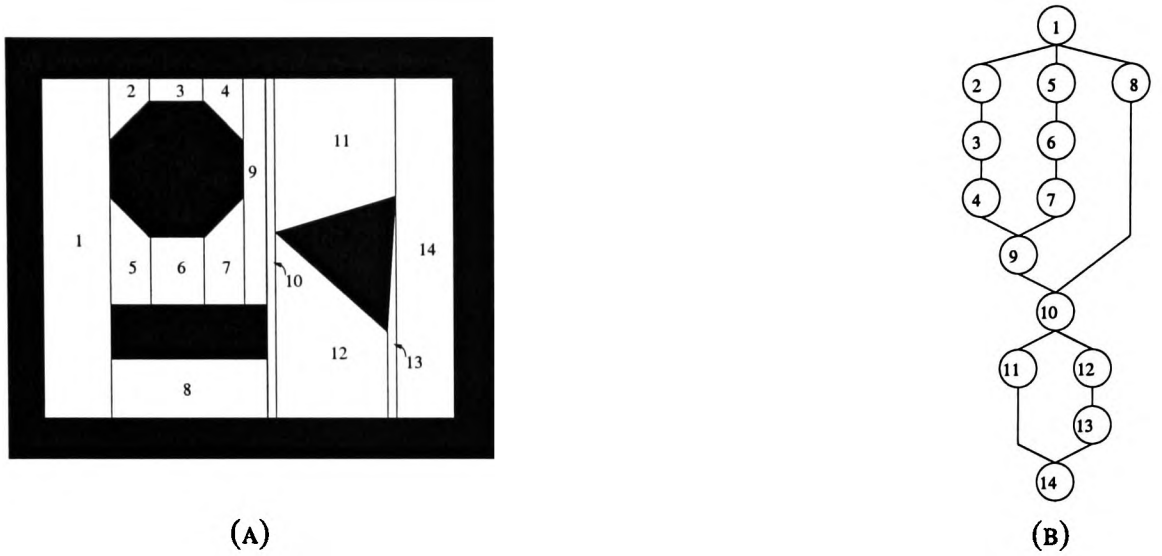


FIGURE 2.5: TRAPEZOIDAL EXACT CELLULAR DECOMPOSITION. (A) TRAPEZOIDAL EXACT CELLULAR DECOMPOSITION. THE CONFIGURATION SPACE IS SPLIT INTO ADJACENT TRAPEZOIDS. (B) PATH IN CELLULAR DECOMPOSITION IS FOUND BY SEARCHING A GRAPH OF CONNECTED CELLS. THIS GRAPH CORRESPONDS TO (A).

2.1.3.2 Approximate Cellular Decomposition

In approximate cellular decomposition methods the free space C_{free} is decomposed into cells of pre-defined shape. For example pixels in two dimensions, or voxels in three dimensions. A very common decomposition is the 2^m-tree for an m dimensional configuration space (see Jackins and Tanimoto [28] and Jung and Gupta [29]). In two dimensions and three dimensions, the 2^m-tree decomposition has the more familiar names quad-tree and oct-tree respectively. Figure 2.6 shows a quad-tree decomposition of the environment shown in Figure 1.2. In the figure, C is recursively subdivided into rectangles. If a rectangle is free of obstacles (white cells in figure) or entirely within an obstacle (black cells in figure) then the recursion on that cell ends. Cells that lie partially within an obstacle (grey cells in figure) are further subdivided. At pre-defined resolution the recursion ends. Path planning is done with the connectivity graph of the white cells. Because the decomposition is only approximate, this decomposition may obscure paths that are finer than the resolution of the quad-tree—very narrow channels being commonly affected.

2.1.4 Planners That Don't Use Configuration Space

Recently, a great deal of attention has been directed toward path planners that don't explicitly use configuration space. Since the computation of configuration space obstacles is exponential in the number of degrees of freedom [38], if this computation can be avoided then there may be substantial time speedup in the resulting algorithms. The crux of these algorithms is that in order to determine if a configuration $\mathbf{q}$ is within a forbidden region $\mathcal{CB}$, it is often faster to perform some type of software collision detection

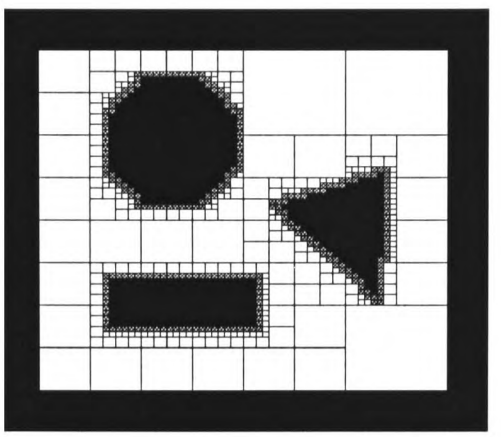


FIGURE 2.6: QUAD-TREE APPROXIMATE CELLULAR DECOMPOSITION. CELLS THAT AREA WHITE (RESP. BLACK) ARE ENTIRELY OUTSIDE (RESP. INSIDE) THE OBSTACLES. GREY CELLS ARE PARTIALLY INSIDE AND OUTSIDE THE OBSTACLES.

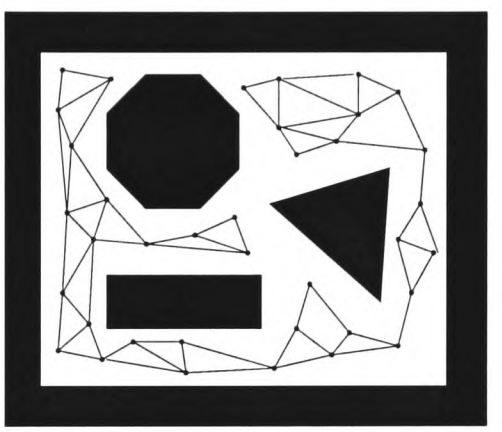


FIGURE 2.7: PROBABILISTIC ROADMAP OF CONFIGURATION SPACE

(i.e. with the geometric models of robot and obstacles) than to explicitly generate $\mathcal{CB}$. Assuming the models of the robot and obstacles are polyhedral, several collision detection algorithms may be suitable, Cohen et.al.'s I-COLLIDE algorithm [18], Mirtich's V-Clip algorithm [51] and Cameron's Enhanced-GJK algorithm [5] all being popular.

Although collision detection-based planners have existed for some time (see Dupont and Derby[22]), the first widely accepted planner in this genre was the *Probabilistic Roadmap* (PRM) by Kavraki and Latombe [33]. This planner first generates many random nodes (configurations) that are evenly distributed throughout C. Each node is tested to see if it is a valid, non-colliding configuration with a software collision detection, and colliding nodes are removed. The next stage of the algorithm is to connect the remaining nodes together that have a valid path between them. This may be done with a straight line test to see if a path exists between the two. The resulting graph can then be searched using a standard graph search algorithm. Figure 2.7 gives an example probabilistic roadmap of the environment shown in Figure 1.2. The nodes are spread randomly in free space and connections are made with collision detection tests. Since nodes are spread randomly, narrow channels in configuration space will often have few nodes, and these

nodes may be difficult to connect to each other. Another planner in this vein is *Ariadne's Clew Algorithm* by Bessiereand et. al. [3].

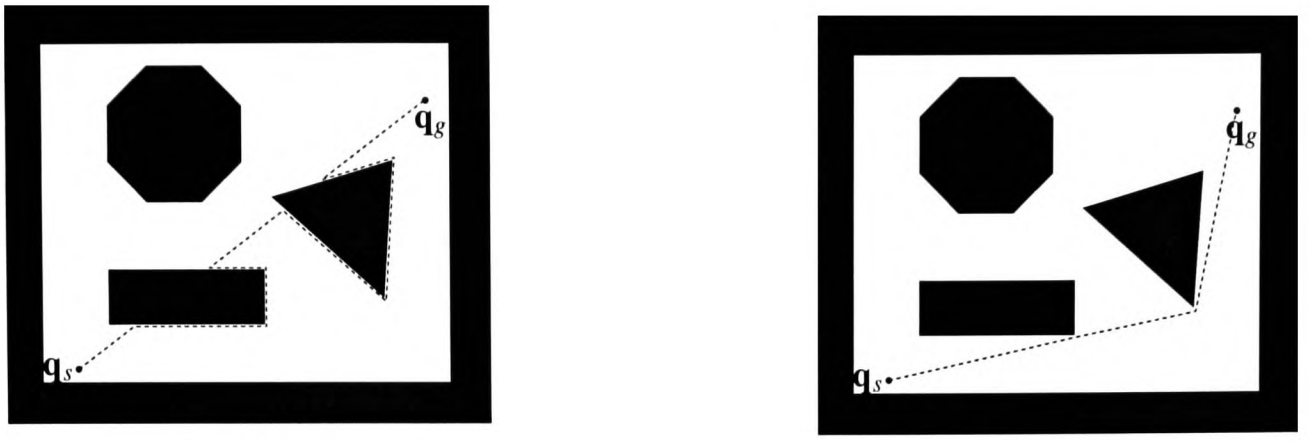
2.2 Sensor-Based Motion Planning

This section addresses previous results in sensor-based planning. The emphasis is on complete algorithms, but notable heuristic algorithms are mentioned at the end as well. A great overview of non-heuristic mobile robot path planners is given by Rao, Kareti, Shi and Iyengar [57]. Although a few years old now, it does outline many important techniques.

2.2.1 The Bug-Algorithms

The algorithms BUG1 and BUG2 by Lumelsky and Stepanov [49] were the first complete sensor-based algorithms that relied only on local sensor data—no world model is generated. In these algorithms the robot operates in a planar environment and initially has no knowledge of its environment. It then moves directly toward the target until it contacts an obstacle. Once it hits an obstacle, it follows its boundary until it is able to move toward the goal again. These algorithms are globally complete and based only on local sensor information (position and contact sensors). Figure 2.8(a) gives an example of a path generated by BUG2 for the environment of Figure 1.2. The robot has two modes of behaviour, motion toward the target along the line joining the start $\mathbf{q}_s$ to the goal $\mathbf{q}_g$ and boundary following, along with a higher level of control that determines when to switch between the two modes. Whenever it is not following a boundary it is travelling on the line between $\mathbf{q}_s$ and $\mathbf{q}_g$.

BUG1 and BUG2 have been used by Lumelsky [47] for path planning of two-link robots in completely unknown environments. Assuming the robot is covered in a contact sensitive skin, the manipulator is able to follow the boundary of configuration space obstacles just like a mobile robot moves around workspace obstacles. Lumelsky and Sun [50] present a general framework for two-DOF planar manipulators based on touch sensitive manipulators following the boundary of virtual obstacles on a parameterisation of the joint space (i.e a torus ($S^1 \times S^1$) for the case of a robot with two revolute joints). Sun and Lumelsky examine the problem for the case of a three link planar manipulator [64]. In this case however, the final link is a sliding link, and the problem is reduced to a simpler case. The BUG algorithms were also applied to the case of a two-link manipulator by Li, Ma and Tso [41], although this is a somewhat incomplete, partial result. Lumelsky and Cheung [46] examine the automation of obstacle avoidance in tele-operation.



(A) (B)

FIGURE 2.8: EXAMPLE PATHS OF THE BUG ALGORITHMS. (A) EXAMPLE OF THE BUG2 ALGORITHM. ROBOT MOVES TOWARD TARGET AND AROUND OBSTACLES WHEN THEY ARE DISCOVERED. (B) EXAMPLE OF THE VISBUG ALGORITHM. ROBOT MOVES TOWARD TARGET AND AROUND OBSTACLES WHEN THEY ARE DISCOVERED.

The operator uses a remote manipulator to move the robot, however when the robot is near an obstacle, it modifies its path so as to avoid collision. The modified path is generated with a bug like algorithm for a two-link manipulator [47]. The robot is covered in an infra-red detecting skin that allows it to follow the surface of obstacles.

The contact sensor-based bug algorithms were further modified to take advantage of more complex range sensor data. VISBUG21 and VISBUG22 by Lumelsky and Skewis [48] use a range finder with variable range. In VISBUG21 the range finder is used to find shortcuts to the paths that would have been generated by BUG2. Figure 2.8(b) gives an example of a path generated by VISBUG21 for the environment of Figure 1.2.

Kamon and Rivlin [31] and Kamon, Rimon and Rivlin [30] developed DISTBUG and TANGENTBUG respectively to "specifically exploit range data". These algorithms use the basic behaviour of motion toward target and boundary following like the other BUG algorithms but use more optimal decisions based on the range sensor data. In DISTBUG the boundary traversing direction is chosen based on sensor data (this direction is fixed in the BUG algorithms) and the robot is able to reverse the local direction when it has made a poor initial choice. In TANGENTBUG they introduce the notion of *Local Tangent Graph (LTG)* which is related to the Tangent Graph of Liu and Arimoto [44][2]. They use the LTG to determine the locally optimal direction to follow a boundary and decide when they've followed a boundary in the wrong direction. TANGENTBUG has been used for a planetary rover by Laubach, Burdick and Matthies [39]. They address a lot of practical issues like the requirement for an omni-directional range sensor and limited accuracy of a dead reckoning systems.

[2] the Tangent Graph has been proved to contain the shortest path from start to goal in the plane $\mathbb{R}^2$

3DBug by Kamon, Rivlin and Rimon [32] is a bug-like algorithm for a *point robot* moving amidst 3-dimensional polyhedral obstacles. Again, the algorithm has basic motion toward goal and boundary following behaviour, but the boundary following behaviour is complicated by the infinite number of direction choices when an obstacle is detected. The algorithm uses a range finder to incrementally construct the obstacle's boundary segments, and determine if a point exists where motion toward the goal can continue.

2.2.2 Distance-Based Roadmap Building Algorithms

2.2.2.1 The Generalised Voronoi Graph

Recent important developments in sensor-based planning are the Generalised Voronoi Graph (GVG) and Hierarchical Generalised Voronoi Graph (HGVG) of Choset [12]. The GVG and HGVG are retract-like sets in C (whose elements are configurations equidistant to several obstacles) which represent the connectivity of the workspace—like any roadmap should. Choset has shown how the GVG is easily constructed with a mobile robot and realistic distance sensors (sonar ring) [13]. In this work, distances are measured in configuration space, so the application of these techniques is limited to circular mobile robots. This work was meant as a first step in a systematic sensor-based framework for multi-link manipulators. The progression was to be as follows:

1. point robot in $\mathbb{R}^n$—completed in [12, 13].
2. rod shaped robot in $\mathbb{R}^2$ (configuration space of $SE(2)$)—completed in [15].
3. rod shaped robot in $\mathbb{R}^3$ (configuration space of $\mathbb{R}^3 \times S^1 \times S^1$)—completed in [40].
4. polyhedral robot in $\mathbb{R}^3$ (configuration space of $SE(3)$)—not completed.
5. a chain of polyhedra in $\mathbb{R}^3$—not completed.

The cases of rod-shaped robots in $\mathbb{R}^2$ and $\mathbb{R}^3$ were examined by Choset [15] and Lee, Choset and Rizzi [40] respectively, however these works use the simplifying assumption that single object distance function $d_i(\mathbf{q})$ for a rod-shaped robot is smooth. This thesis shows that $d_i(\mathbf{q})$ for a convex robot $\mathcal{A}$ and convex obstacle $\mathcal{B}_i$ (the rod is convex) is in fact nonsmooth, so the analysis based on differential topology used in these works is not strictly correct.

Figure 2.9 shows the incremental construction of the GVG in $\mathbb{R}^2$ at some point before fully exploring the workspace. Tick marks represent the direction to the closest obstacle (since points on the GVG are equidistant to two obstacles, there are two ticks at each point). The incremental construction is based on two properties: (i) the robot can follow edges equidistant to two obstacles based on sensor measurements

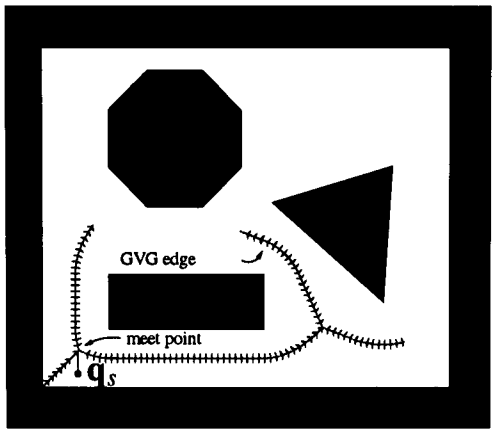


FIGURE 2.9: CONSTRUCTION OF THE GENERALISED VORONOI GRAPH (GVG). TICK MARKS REPRESENT THE DIRECTION TO THE CLOSEST OBSTACLE.

and (ii) the robot is able to detect points where GVG edges meet. These *meet points* occur when the robot is equidistant to more than two obstacles (or more than $n+1$ obstacles in $\mathbb{R}^n$). This second requirement (ii) is accomplished with a *meet point detector*, which is a primary difference between Choset's incremental GVG construction and the sensor-based planner of Rimon and Canny described below.

2.2.2.2 The Distance Silhouette

Rimon and Canny [60, 59] have also presented a complete sensor-based planner based on incremental construction of a roadmap. This algorithm was indeed the first provably correct sensor-based planner for arbitrarily large configuration spaces. This planner is a sensor-based version of the Opportunistic Path Planner (OPP) [42]. OPP was based upon the construction of *ridge curves* that maximise the distance to obstacles in the environment. Since these ridge curves are not necessarily connected, in order to have a topology representing roadmap, we connect ridge curves with *bridge curves* through critical points on configuration space obstacles and bifurcations of the ridge curves.

The sensor-based version of OPP is based on two properties: (i) the robot can follow ridge curves that are the maximum of some distance potential and (ii) the robot is able to detect critical points. (ii) is accomplished through the use of a *critical point detector* and *minimum passage detector* that signify when to start generating a bridge curve. The implementation of these sensors is briefly discussed for the highly simplified case of an elliptical mobile robot, however, a general implementation of the critical point sensor is not addressed, and is a major drawback of the algorithm. Another major drawback of this technique is that it assumes the distance function $D(\mathbf{q})$ is a smooth function, which it is not [11, 16]. This assumption leads to some very elegant proofs of completeness based on the bifurcation theory of smooth functions, but hinders practical application of the algorithm. A main contribution of this thesis is a detailed analysis

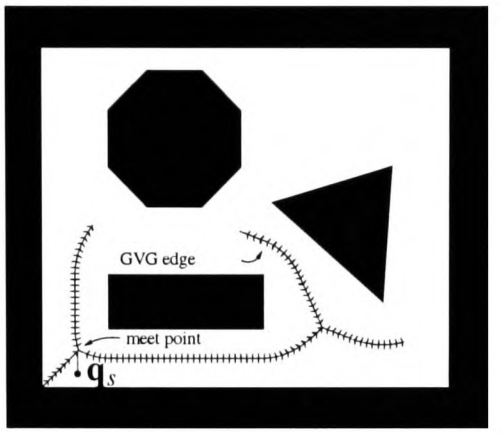


FIGURE 2.10: CONSTRUCTION OF ROADMAP BASED ON SILHOUETTE OF DISTANCE FUNCTION.

of the nonsmooth nature of $D(\mathbf{q})$.

Figure 2.10 shows the incremental construction of this roadmap at some point before completion. From the start point, the first encountered ridge curve was fully explored and a detected critical point was used to generate a bridge curve to a second partially explored ridge curve.

2.2.2.3 Other Distance-Based Algorithms

Another very recent related development in distance based roadmap generation is the EquiDistance Diagram (EDD) of Keerthi et. al. [34]. Developed in the context of a sensorless algorithm, the EDD algorithm seems to be the first that can generate a one-dimensional roadmap for *arbitrary* configuration spaces. However, the authors again use the simplifying assumption that $d_i(\mathbf{q})$ is smooth, and use an analysis based on differential topology. The authors allude to the fact that the algorithm may be "sensorised", but the issue of connecting disconnected portions of the EDD in a sensor-based manner is not addressed.

2.2.3 Other Map Building Algorithms

Taylor and Kriegman [65] presented a complete planner for a mobile robot in the plane equipped with a vision system. In their method, the robot circumnavigates the obstacles and then creates an adjacency graph between obstacles within line of sight of each other. Another similar method is due to Kortenkamp and Weymouth [36]. In their technique, a mobile robot in the plane equipped with a vision and a sonar sensor is able to generate a *topological map* of the environment, representing the connectivity of C_{free}—thus being a complete planner. The planner's major drawback is that it relies on a very simplified world model of rectangular obstacles.

Yu and Gupta [66] present an incremental construction of the environment for a multi-link manipulator with a range sensor. Sensor measurements are made in the workspace with the range sensor. At each iteration, the range finder takes a snapshot, landmarks are placed randomly in the free space and surrounding area and the robot finally moves to a new accessible landmark. Connected landmarks in free-space may eventually represent the connectivity of free-space. Jung and Gupta developed an hierarchical oct-tree-based representation for storing distance data collected with the range finder [29]. This data structure is used in [66] to represent the environment.

Zeller and Schulten [67] use an external camera which views the robot and adjacent obstacles. Images are processed using a neural network which generates a *topology representing network.* This network is used to plan the motion of the end effector in order to avoid obstacles. The planner is based on a diffusion process.

Another recent development is the D* (DYNAMIC A*) algorithm by Stentz [63]. This algorithm assumes a small amount of initially sensed information and generates a optimal path based on this information (using A*). As the robot proceeds it will detect new obstacles, at which point the graph must be modified, weights adjusted etc. The D* algorithm generates the optimal path at each step, based on its knowledge at that step. This likely doesn't correspond to the globally optimal path (i.e. using A* when all obstacles are known).

2.2.4 Potential Functions and Sensors

Connolly and Grupen adapted a harmonic potential based planner [19] for use with a mobile robot with the ability to detect obstacles [20]. As obstacles are discovered, the harmonic potential is updated to drive the robot away from newly discovered obstacles. Because the new potential is still a solution to Laplace's equation, the robot is always guaranteed to find the goal if a path exists. This method has the drawback that at each iteration where new obstacles are discovered, there is significant computational cost to re-calculating the potential function as new obstacles are detected.

2.2.5 Snake Like Robots

Motion planning for snake-like robots must almost be sensor-based by default. Snake-like robots are referred to as hyper-redundant, meaning they have a very large number of degrees of freedom. This large number of degrees of freedom makes classical planners explode in term of time complexity (generating

configuration space obstacles has exponential time complexity in the number of degrees of freedom), so sensor-based methods are the only choice. The first serious study of snake-like robots was done by Chirikjian [9]. Reznik and Lumelsky [58] developed a planner for a snake like manipulator (i.e. one end is fixed). They assume the robot is able to sense obstacles on any point of its body (with a sensitive skin), and present a planner to move the head from start to goal in an initially unknown environment. Their method hinges on moving the head according to VisBug21 algorithm [48] and moving the body with a tractrix motion[3] while avoiding collisions with environment. Henning has also applied the GVG to snake robot motion planning [26]. In this algorithm, the head of the robot generates the GVG and the remainder of the robot follows the GVG.

2.2.6 Heuristic Algorithms

The *Vector Field Histogram* method by Borenstein and Koren [4] is another practical heuristic method for motion planning of mobile robots. This approach is based on a discretisation of configuration space where each pixel is assigned a value based on the probability that it is occupied by an obstacle. So, the more sensor readings taken, the more accurate these pixel probabilities become. Since this is a discrete method, the memory requirements increase polynomially with discretisation resolution (x^2 in 2D and x^3 in 3D, where x is the inverse of the pixel width) .

[3] A tractrix motion is such that a large motion of the head leads to progressively smaller motion in preceding links.

§3

"The world, indeed, is like a dream and the treasures of the world are an alluring mirage! Like the apparent distances in a picture, things have no reality in themselves, but they are like heat haze."

Buddha

Nonsmooth Analysis of the Distance Function

In this chapter we present a detailed nonsmooth analysis of the distance function $D(\mathbf{q})$

$$D(\mathbf{q}) = \min_i \{d_i(\mathbf{q})\}$$

defined as the pointwise minimum of a finite number single object distance functions $d_i(\mathbf{q})$ (i is the obstacle $\mathcal{B}_i$ index). Those $d_i(\mathbf{q})$s that attain the minimum distance $D(\mathbf{q})$ can be indexed by the set $I(\mathbf{q}) = \{i : d_i(\mathbf{q}) = D(\mathbf{q})\}$, a notation used throughout this thesis.

$$d_i(\mathbf{q}) = \min_{\substack{\mathbf{a}\in\mathcal{A}(\mathbf{q}) \\ \mathbf{b}\in\mathcal{B}_i}} \|\mathbf{a} - \mathbf{b}\|$$

where $\mathbf{q} \in C$ and $\mathcal{A}(\mathbf{q})$ is the set of points occupied by the robot at configuration $\mathbf{q}$. Nearly all previous work that deals with realistic robot models has failed to appreciate that the distance function $d_i(\mathbf{q})$ between convex bodies is *not* everywhere differentiable. In fact, $d_i(\mathbf{q})$ is both nonsmooth and non-convex (in $C = SE(2)$ and $C = SE(3)$), so the usual tools of differential topology are not strictly applicable. Nonetheless, a detailed understanding of the distance function is vital to the robotics community, and this chapter presents the most realistic analysis to date. Our analysis is based on the nonsmooth notions of *weak slope* and

generalised gradient, the former never having been used in robotics. These are powerful tools, especially when working in finite dimensional spaces, and we believe are of such general purpose that smoothness assumptions may no longer be needed for a wide range of robotics problems

3.1 Preliminaries

Since most readers are probably not familiar with the tools of nonsmooth analysis it is worth devoting some time to introducing the most important (to this thesis that is) concepts. We begin with the *generalised gradient.*

3.1.1 Clarke's Generalised Gradient

Definition 3.1 Let X be an open subset of a Banach space E, and $\phi : X \to \mathbb{R}$ be a locally Lipschitz continuous function where $\mathbf{x} \in X$ and $\mathbf{w} \in X$. The **generalised directional derivative**, denoted $\phi^\circ(\mathbf{x}; \mathbf{e})$, is defined as

$$\text{for all } \mathbf{e} \in E : \phi^\circ(\mathbf{x}; \mathbf{e}) = \limsup_{\substack{\mathbf{w} \to \mathbf{x} \\ t \to 0^+}} \frac{\phi(\mathbf{w} + t\mathbf{e}) - \phi(\mathbf{w})}{t}. \qquad \square$$

The most important attribute of $\phi^\circ(\mathbf{x}; \mathbf{e})$ with regards to our analysis is that it is *upper semicontinuous* (u.s.c.) in $\mathbf{x}$,$\mathbf{e}$ [16]. So, when $\phi^\circ(\mathbf{x}; \mathbf{e}) < 0$, $\phi(\mathbf{x} + t\mathbf{e}) < \phi(\mathbf{x})$ for small enough t (ϕ tends "downhill" in direction $\mathbf{e}$). In fact, ϕ tends "downhill" in direction $\mathbf{e}$ for all $\mathbf{y} \in \text{nbhd}(\mathbf{x})$. "uphill" in direction $\mathbf{e}$ implies $\phi^\circ(\mathbf{x}; \mathbf{e}) \geq 0$, but the converse is not true. We use this property extensively when exploring the critical points of $D(\mathbf{q})$ in Section 3.2.3. The generalised directional derivative is used to define the generalised gradient.

Definition 3.2 The **generalised gradient** of $\phi(\mathbf{x})$ defined above, denoted $\partial\phi(\mathbf{x})$, is the set

$$\partial\phi(\mathbf{x}) = \{\alpha \in E^{'} : \phi^\circ(\mathbf{x}; \mathbf{e}) \geq \langle \alpha, \mathbf{e} \rangle \quad \text{for all} \quad \mathbf{e} \in E\}.$$

where $E^{'}$ is the **dual space** of E. $\square$

Comment 3.3 The reader should at this point realise that $SE(2)$ (and $SE(3)$) are not Banach spaces since they are not linear. However, since S^1 is a manifold, we can operate through the tangent space $T_{\mathbf{q}}SE(2)$ of $SE(2)$ and the dual of the tangent space (tangent bundle/tangent dual globally). For example, we calculate

$\phi^\circ(\mathbf{x};\mathbf{e})$ and $\partial\phi(\mathbf{x})$ in $\mathbb{R}^n$, and through the existence of a diffeomorphic map $\Psi : SE(2) \rightarrow T_{\mathbf{q}}SE(2)$, we preserve the critical aspects of $\partial\phi(\mathbf{x})$ (i.e compactness etc.) in $SE(2)$. □

In motion planning, the domain C of $d_i(\mathbf{q})$ and $D(\mathbf{q})$ has dimension equal to the number of degrees of freedom of the robot. A physically realisable robot must certainly have a finite number of degrees of freedom n, so the domain has a finite dimension of $\dim(C) = n$. Let's consider finite dimensional spaces in general. If X is finite dimensional, we have the following useful property, based on Rademacher's Theorem, which states that a Lipschitz function on an open subset of $\mathbb{R}^n$ is differentiable almost everywhere (non-differentiable set has measure zero). In the following, co{} denotes convex hull.

Corollary 3.4 Let X be finite dimensional, let S be any set of Lebesque measure 0 in X and let $\Omega_\phi \subseteq X$ be the set of points where ϕ fails to be differentiable. Then the generalised gradient is given by

$$\partial\phi(\mathbf{x}) = \text{co}\{\lim \nabla\phi(\mathbf{x}_i) : \mathbf{x}_i \rightarrow \mathbf{x}, \mathbf{x}_i \notin S, \mathbf{x}_i \notin \Omega_\phi\}.$$ ■

where $\{\mathbf{x}_i\}$ is *any* sequence with accumulation point $\lim \mathbf{x}_i = \mathbf{x}$.

This alternative definition of the generalised gradient says that $\partial\phi(\mathbf{x})$ is given by the convex hull of all the gradient limits approaching $\mathbf{x}$. Consider this example

Example 3.5 Consider the function graph $\phi(\mathbf{x}) = -|x_1| - |x_2| + 1$ shown in Figure 3.1(a). An arbitrary sequence $\{\mathbf{x}_k\}$ with $\mathbf{x}_k \rightarrow \mathbf{x}$ is shown in Figure 3.1(b). The limit of the gradient vectors of this sequence of points $\lim \nabla\phi(\mathbf{x}_i)$ is shown to be $(1,-1)$. The convex hull of the limits of all possible sequences gives the generalised gradient. In this case $\partial\phi(\mathbf{x}) = \text{co}\{(1,1),(-1,1),(-1,-1),(1,-1)\}$, which is a square region.

The well established results of convexity theory [1], in particular the *separation theorem*, are used in a number of critical proofs since $\partial\phi(\mathbf{x})$ is compact and convex [16]. We now present a third, and final, formulation of the generalised gradient based on the *normal cone* of the *epigraph*.

Definition 3.6 The **normal cone** of a set $C \subseteq \mathbb{R}^n$ at a point $\mathbf{x} \in C$, denoted $N_C(\mathbf{x})$ is given by

$$N_C(\mathbf{x}) = \text{cl}(\text{co}\{\lambda \lim \frac{\mathbf{v}_i}{|\mathbf{v}_i|} : \lambda \geq 0, \mathbf{v}_i \perp C \text{ at } \mathbf{x}_i, \mathbf{x}_i \rightarrow \mathbf{x}, \mathbf{v}_i \rightarrow \mathbf{0}\}) \tag{3.1}$$

where $\mathbf{v}_i = \mathbf{x}_i - \mathbf{x}$. □

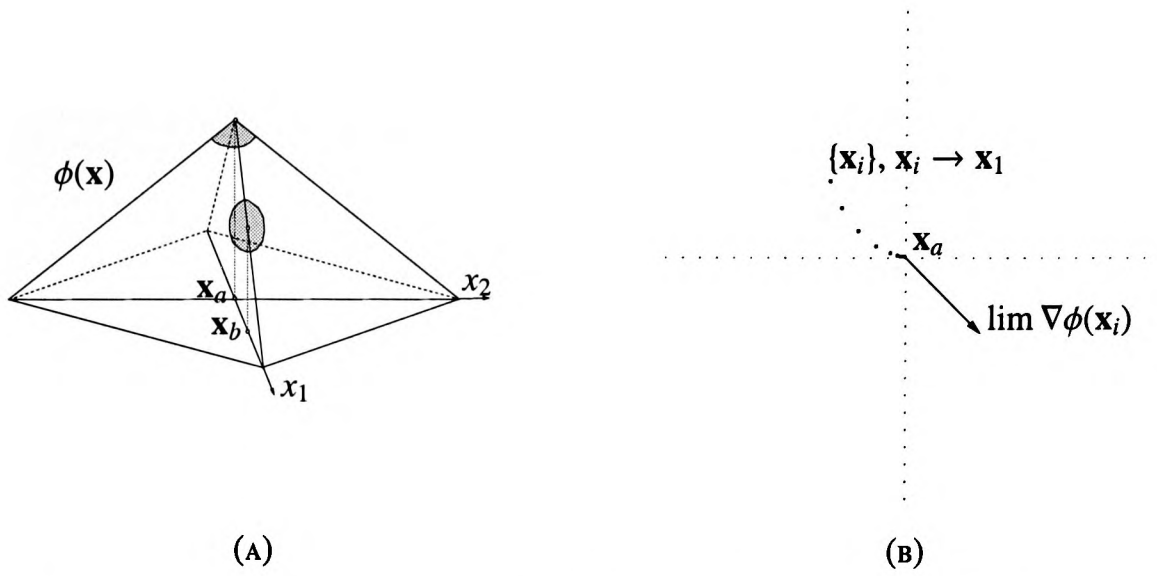


FIGURE 3.1: (A) WEAK SLOPE IS VISUALISED USING "RUBBER SHEETS". (B) $\partial\phi(\mathbf{x}_a)$ IS CONVEX HULL OF GRADIENT VECTOR LIMITS. $\partial\phi(\mathbf{x}_a) = \text{co}\{(1,1),(-1,1),(-1,-1),(1,-1)\}$

Definition 3.7 The **epigraph** of a function $\phi : \mathbb{R}^n \rightarrow \mathbb{R}$, denoted epi ϕ, is the set

$$\text{epi}\,\phi = \{(\mathbf{x}, r) \in X \times \mathbb{R} : \phi(\mathbf{x}) \leq r\}$$

i.e. the set of points above the graph of ϕ. □

We now have the following alternate definition of the generalised gradient based on the geometric notions of the normal cone and epigraph [16]. We use this formulation exclusively in Chapter 4.

$$\partial\phi(\mathbf{x}) = \{\zeta : (\zeta, -1) \in N_{\text{epi}\,\phi}(\mathbf{x}, \phi(\mathbf{x}))\} \tag{3.2}$$

which is the set of ζs where $(\zeta, -1)$ is a member of the normal cone to the epigraph $N_{\text{epi}\,\phi}$. Consider the following example.

Example 3.8 Figure 3.2 shows the epigraph of the function

$$\phi(x) = \begin{cases} \frac{2}{3}x & x \leq 3 \\ \frac{3}{2}x - \frac{5}{2} & x > 3 \end{cases} \tag{3.3}$$

and the normal cone $N_{\text{epi}\,\phi}(x)$ at $x = 3$. The generalised gradient is the set $\partial\phi(x) = \{\zeta : (\zeta, -1) \in N_{\text{epi}\,\phi}(x, \phi(x))\}$, which can be determined by subtracting $x = 3$ from the set of x values where hyperplane $y = \phi(3) - 1$ (dashed line) coincides with $N_{\text{epi}\,\phi}(3)$. In this example $\partial\phi(3) = [\frac{2}{3}, \frac{3}{2}]$.

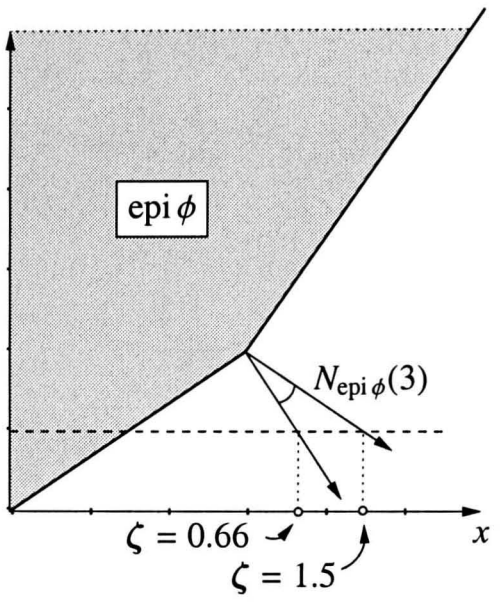


FIGURE 3.2: EPIGRAPH OF $\phi(x)$ GIVEN BY EQUATION 3.3 AND THE NORMAL CONE $N_{\text{epi}\,\phi}(x)$ AT $x = 3$. THE GENERALISED GRADIENT IS THE SET $\partial\phi(3) = [\frac{2}{3}, \frac{3}{2}]$.

The generalised gradient is an extremely powerful tool given that a Calculus exists to determine the generalised gradient of function sums, multiplications, compositions etc. We next present several basic rules of the generalised gradient calculus that we use in this thesis. The interested reader can refer to [16] for more examples.

Corollary 3.9 (Scalar multiplication) For any scalar λ, one has

$$\partial(\lambda f)(\mathbf{x}) = \lambda \partial f(\mathbf{x})$$ ■

Corollary 3.10 (Finite Sum) If $\phi(\mathbf{x})$ is expressed as a finite sum of functions $\phi_1(\mathbf{x}) + \cdots + \phi_n(\mathbf{x})$ then the generalised gradient satisfies

$$\partial(\sum_{i=1}^{n} \phi_i)(\mathbf{x}) \subseteq \sum_{i=1}^{n} \partial f_i(\mathbf{x})$$

where the right hand side denotes the set of all points ζ obtainable as a sum $\sum_{i=1}^{n} \zeta_i$, where each ζ_i belongs to $\partial\phi_i(\mathbf{x})$. ■

Corollary 3.11 (Products) If $\phi(\mathbf{x})$ is expressed as a product of two functions $\phi_1(\mathbf{x})\phi_2(\mathbf{x})$ then the generalised gradient satisfies

$$\partial(\phi_1\phi_2)(\mathbf{x}) \subseteq \phi_1(\mathbf{x})\partial\phi_2(\mathbf{x}) + \phi_2(\mathbf{x})\partial\phi_1(\mathbf{x}) \tag{3.4}$$

where $\phi_i(\mathbf{x})\partial\phi_j(\mathbf{x})$ denotes a multiplication of every element of $\partial\phi_j(\mathbf{x})$ by $\phi_i(\mathbf{x})$ and sum has the same

meaning as in Corollary 3.10. ■

Corollary 3.12 (Chain rule) If $\phi(\mathbf{x})$ is expressed as a composition of two functions $\phi = g \circ h$ with $h: X \rightarrow \mathbb{R}^n$ and $g: \mathbb{R}^n \rightarrow \mathbb{R}$ then the generalised gradient satisfies

$$\partial\phi(\mathbf{x}) \subseteq \text{cl}(\text{co}\{\sum \alpha\zeta_i : \zeta_i \in \partial h_i(\mathbf{x}), \alpha \in \partial g(h(\mathbf{x}))\}) \tag{3.5}$$

where the sum has the same meaning as in Corollary 3.10 and co denotes convex hull. ■

The distance function $D(\mathbf{q})$ is a pointwise minimum function, so it's useful to have a 'calculus' rule for functions of this type as well. Theorem 3.13 presents a very useful rule for determining $\partial\phi(\mathbf{x})$ of pointwise minimum functions.

Theorem 3.13 (Pointwise minimum) If $\phi(\mathbf{x})$ is expressed as the pointwise minimum of a finite number of functions $\phi(\mathbf{x}) = \min_i\{f_i(\mathbf{x})\}$ then the generalised gradient satisfies

$$\partial\phi(\mathbf{x}) \subseteq \text{co}\{\partial f_i(\mathbf{x}) : i \in I(\mathbf{x})\} = \text{co}\{\bigcup_{i \in I(\mathbf{x})} \partial\phi_i(\mathbf{x})\} \tag{3.6}$$

where $I(\mathbf{x})$ is the set of indices where $f_i(\mathbf{x}) = \phi(\mathbf{x})$ and co{} denotes the convex hull of a set of vectors.

Proof We have the following for the pointwise maximum function $\phi_{max}(\mathbf{x}) = \max_i\{f_i(\mathbf{x})\}$ [16]

$$\partial\phi_{max}(\mathbf{x}) \subseteq \text{co}\{\partial f_i(\mathbf{x}) : i \in I(\mathbf{x})\}$$

Since $\min_i\{f_i(\mathbf{x})\} = -\max_i\{-f_i(\mathbf{x})\}$ we have

$$\begin{aligned}
\partial\phi(\mathbf{x}) &= \partial(-\max_i\{-f_i(\mathbf{x})\}) \\
&= -\partial(\max_i\{-f_i(\mathbf{x})\}) \text{ (by Corollary 3.9)} \\
&\subseteq -\text{co}\{-\partial f_i(\mathbf{x}) : i \in I(\mathbf{x})\} \text{ (by Equation 3.6)} \\
&= \text{co}\{\partial f_i(\mathbf{x}) : i \in I(\mathbf{x})\} \text{ (property of convex hull)}
\end{aligned}$$

■

3.1.2 The Weak Slope

In subsequent sections, $\phi : X \to \mathbb{R}$ is an everywhere continuous function taking points in its domain X to the real numbers $\mathbb{R}$. We assume X is a complete metric space with metric d.

Definition 3.14 (Canino and Degiovanni, [6]) Let $\phi : X \to \mathbb{R}$ be a continuous function. For every $\mathbf{x} \in X$ we denote by $|d\phi|(\mathbf{x})$ the supremum of the σs in $[0, \infty)$ such that there exists $\delta > 0$ and a continuous map $\mathcal{H} : B_\delta(\mathbf{x}) \times [0, \delta] \to X$ such that

$$\begin{aligned} &\forall \mathbf{v} \in B_\delta(\mathbf{x}), \forall t \in [0, \delta] : \text{dist}(\mathcal{H}(\mathbf{v}, t), \mathbf{v}) \leq t \\ &\forall \mathbf{v} \in B_\delta(\mathbf{x}), \forall t \in [0, \delta] : \phi(\mathcal{H}(\mathbf{v}, t)) - \phi(\mathbf{v}) \leq -\sigma t \end{aligned} \tag{3.7}$$

The extended real number $|d\phi|(\mathbf{x})$ is called the **weak slope** of ϕ at $\mathbf{x}$. □

The weak slope is directly related to the gradient, when the gradient exists. In the smooth case, we have Corollary 3.15.

Corollary 3.15 (Canino and Degiovanni, [6]) If X is a C^1 differentiable manifold and ϕ is a C^1 function, $|d\phi|(\mathbf{x}) = \| \nabla \phi(\mathbf{x}) \|$. ■

Heuristically, if $|d\phi|(\mathbf{x}) > 0$, we can map a nbhd($\mathbf{x}$) (nbhd() denotes neighbourhood) "downhill" continuously. Consider the following two examples, the first being less formal than the second:

Exaınple 3.16 Consider the function plot $\phi(\mathbf{x}) = -|x_1| - |x_2| + 1$ shown in Figure 3.1(a). The figure shows "rubber sheets" centred at two points $\phi(\mathbf{x}_a)$ and $\phi(\mathbf{x}_b)$. We have $|d\phi|(\mathbf{x}_a) = 0$, since every point of the sheet cannot be stretched downhill without "poking a hole" in it (changing the topology). $|d\phi|(\mathbf{x}_b) > 0$ since we can stretch all points "downhill" without "tearing" the rubber sheet. □

Example 3.17 Consider the function graph of $\phi(x) = -|x - 1| + 1$ in Figure 3.3(a) in which we desire to determine $|d\phi|(1)$, where ϕ is not differentiable in the classical sense. We need to determine the supremum of the σs in $[0, \infty)$ such that Equations 3.7 are satisfied. If $\sigma = 0$, the continuous function $\mathcal{H}(v, t) = v$ (no motion) satisfies the requirements. However, if $\sigma > 0$, no continuous function can be found. To see this, consider without loss of generality the case where $x = 1$ maps to the right ($\mathcal{H}(1, t) - 1 > 0$) in Figure 3.3(a). For any t, we can construct an ϵ-neighbourhood around $\mathcal{H}(1, t)$ that does not contain points in any δ-neighbourhood around $x = 1$, thus no continuous transformation exists. The same applies if $x = 1$ maps left. The weak slope is thus $|d\phi|(1) = 0$. □

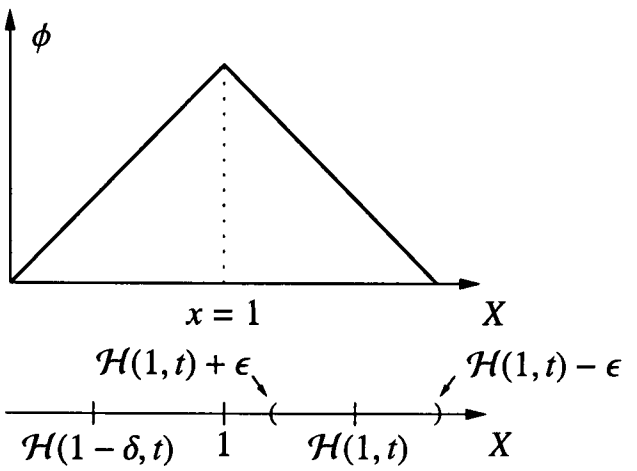


FIGURE 3.3: EXAMPLE OF THE WEAK SLOPE AT NON-DIFFERENTIABLE POINT $x = 1$ OF THE FUNCTION $\phi = -|x-1|+1$. (TOP) THE GRAPH OF ϕ AND (BOTTOM) SHOWS THAT NO CONTINUOUS 'DOWNHILL' MAPPING $\mathcal{H}(\mathbf{v},t)$ EXISTS SINCE IT IS IMPOSSIBLE TO MAP POINTS IN A δ-NEIGHBOURHOOD OF $x = 1$ TO POINTS IN AN ϵ-NEIGHBOURHOOD OF $\mathcal{H}(1,t)$. IN THIS EXAMPLE, THE WEAK SLOPE IS $|d\phi|(1) = 0$.

An important note about the weak slope is that it satisfies our intuitive notions of "downhill", which is important for determining the *critical points* of $D(\mathbf{q})$.

Definition 3.18 If $|d\phi|(\mathbf{x}) = 0$, then $\mathbf{x}$ is considered a **critical point** of ϕ. Other points are called **regular points** of ϕ. □

Recall that a *critical point* $p \in X$ of a smooth function $\phi : X \to \mathbb{R}$ occurs when the partial derivatives are all zero

$$\frac{\partial \phi}{\partial x_1}(p) = \cdots = \frac{\partial \phi}{\partial x_n}(p) = 0$$

Most readers will be familiar with the smooth concepts of local maximum, local minimum and saddle-points—types of smooth critical points. It turns out that these concepts all translate intuitively to non-smooth examples, and the reader should verify to himself/herself that the weak slope is zero in all these cases. Figure 3.4 should help with this exercise.

Definition 3.19 A real number c is called a **critical value** of ϕ if the set $\phi^{-1}(c) = \{\mathbf{x} \in X\,;\, \phi(\mathbf{x}) = c\}$ contains at least one critical point. All other real numbers, even those $c \notin \text{Range}(\phi)$, are referred to as **regular values**. □

We now present some notation used to represent various subsets of X.

$$K_c = \{\mathbf{x} \in X\,;\, \phi(\mathbf{x}) = c, |d\phi|(\mathbf{x}) = 0\}$$

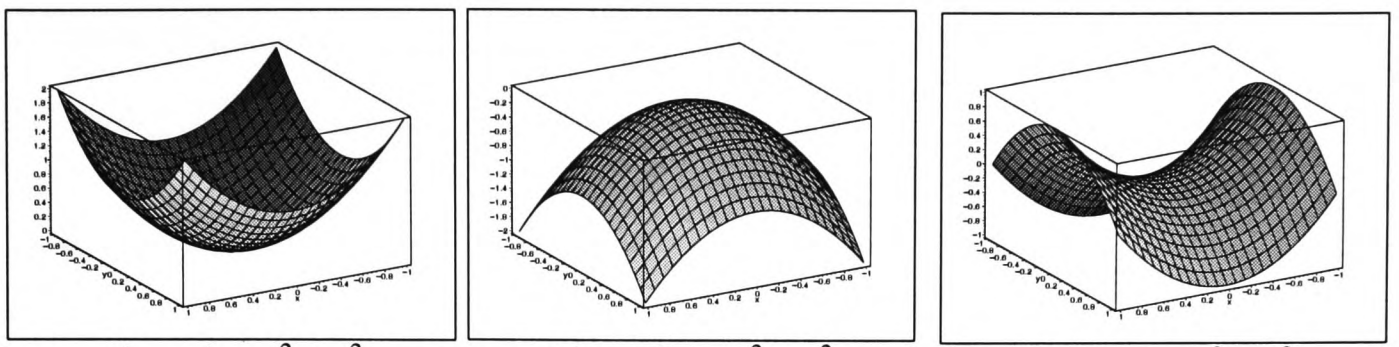


FIGURE 3.4: (A) SMOOTH AND NONSMOOTH LOCAL MINIMUM. (B) SMOOTH AND NONSMOOTH LOCAL MAXIMUM. (C) SMOOTH AND NONSMOOTH SADDLE-POINTS (OR 'MOUNTAIN-PASS' POINTS). IN ALL THESE FIGURES, $|d\phi|(\mathbf{0}) = 0$.

K_c is referred to a **critical set at level** c. The set of all critical points is denoted K, where $K = \cup K_c$. Sometimes $K_{\phi(\mathbf{x})}$ is subscripted to identify the function. The set of all points in X at level c is denoted by

$$\phi^{-1}(c) = \{\mathbf{x} \in X\,;\, \phi(\mathbf{x}) = c\}$$

and is referred to as a **preimage** or **level set** of ϕ. Note that $K_c \subseteq \phi^{-1}(c)$.

The Palais-Smale condition (referred to as condition-C in some literature) denoted (PS), is vital to establishing the compactness of K_c, since it ensures no critical points occur at infinity and that $\phi(K)$ is closed. Nearly all the critical point literature, both smooth and nonsmooth, assumes that ϕ obeys (PS). From a planning point of view, ensuring critical points don't occur at infinity is important, since a physically realisable robot cannot possibly explore to infinity.

Definition 3.20 Say that ϕ verifies the **Palais-Smale condition** (PS), if any sequence $\{\mathbf{x}_n\}$ satisfying $\lim_n \phi(\mathbf{x}_n) = c$ and $\lim_n |\phi|(\mathbf{x}_n) = 0$ has a convergent subsequence. □

The (PS) condition provides a way to build some "compactness" into the functional ϕ. Indeed, the (PS) condition implies that each K_c is a compact set, and that each disjoint connected component of each K_c is

also a compact set [55].

It was noted in [21] that a critical point theory based on the generalised gradient cannot be developed. The function $\phi : \mathbb{R} \rightarrow \mathbb{R}$ defined by $\phi(x) = x - \sqrt[3]{x^2}$ was used as an example. This function obeys (PS) and has a critical point at the origin. However, the generalised gradient in this case evaluates to the empty set $\partial\phi(0) = \varnothing$. For Lipschitz functions with a finite dimensional domain, the notions of weak slope and the generalised gradient are closely related, with each suitable for tackling different issues in nonsmooth analysis.

We now present a critical theorem[1] that links the notions of weak slope and generalised gradient for the case of Lipschitz functions defined on a finite dimensional domain $\mathbb{R}^n$.

Theorem 3.21 $|d\phi|(\mathbf{x}) = 0$ ($\mathbf{x}$ is a critical point) if and only if $\mathbf{0} \in \partial\phi(\mathbf{x})$.

Proof This is not true in general (i.e. infinite dimensional domains), since by Theorem 2.17 [21] we have

$$|d\phi|(\mathbf{x}) \geq \min\{\|\alpha\| : \alpha \in \partial\phi(\mathbf{x})\}$$

which only gives one implication direction. However, if the domain X is finite dimensional, we have equality in the above relation and the theorem follows. Recall from Corollary 3.4 the alternate definition of the generalised gradient

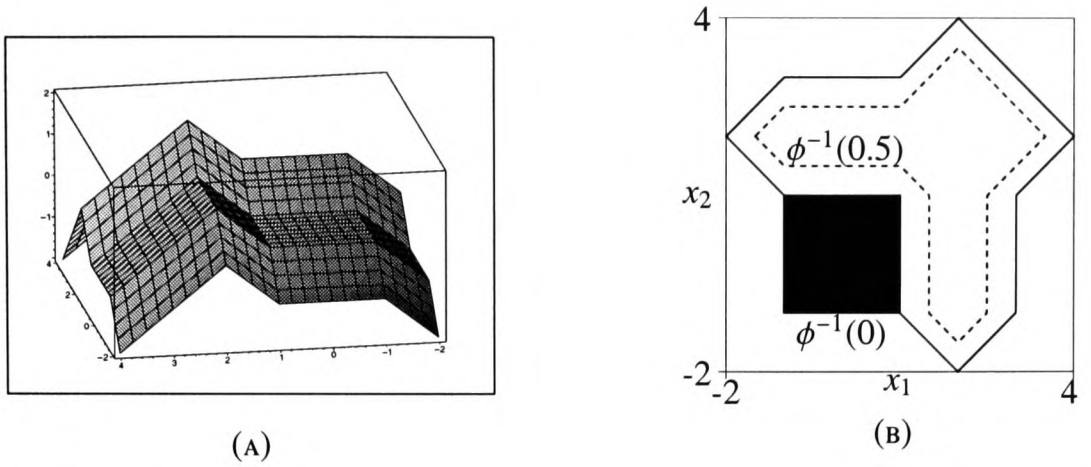
$$\partial\phi(\mathbf{x}) = \text{co}\{\lim \nabla\phi(\mathbf{x}_i) : \mathbf{x}_i \rightarrow \mathbf{x}, \mathbf{x}_i \notin S, \mathbf{x}_i \notin \Omega_\phi\}$$

Let the minimum norm element of $\partial\phi(\mathbf{x})$ be ζ. Now, there must exist a sequence of points $\{\mathbf{x}_k\}$ with $\mathbf{x}_k \rightarrow \mathbf{x}$ such that $\lim_{\mathbf{x}_k \rightarrow \mathbf{x}} \nabla\phi(\mathbf{x}_k) = \zeta$. By the continuity of the norm operator (in a finite dimensional space) $\lim_{\mathbf{x}_k \rightarrow \mathbf{x}} \| \nabla\phi(\mathbf{x}_k) \| = \| \zeta \|$. Within any delta neighbourhood $B_\delta(\mathbf{x})$, by Corollary 3.15, there exist points $\mathbf{x}_k$ where $|d\phi|(\mathbf{x}_k) = \| \nabla\phi(\mathbf{x}_k) \|$ so by definition of the weak slope for any $\epsilon > 0$, there exists $\delta > 0$ where $|d\phi|(\mathbf{x}) \leq \| \nabla\phi(\mathbf{x}_k) \| + \epsilon$ for all $\mathbf{x}_k \in B_\delta(\mathbf{x})$. Since $\lim_{\mathbf{x}_k \rightarrow \mathbf{x}} \| \nabla\phi(\mathbf{x}_k) \| = \| \zeta \|$ for at least one sequence we have

$$|d\phi|(\mathbf{x}) \leq \min\{\|\alpha\| : \alpha \in \partial\phi(\mathbf{x})\} = \| \zeta \|$$

so the theorem follows. ■

[1] $|d\phi|(\mathbf{x})$ and $\partial\phi(\mathbf{x})$ are actually defined in complete metric spaces and Banach spaces respectively, so Theorem 3.21 requires $\mathbb{R}^n$ as the domain. We assume C is charted to $\mathbb{R}^n$



(A) (B)

FIGURE 3.5: THE PREIMAGE THEOREM ALLOWS US TO DETERMINE THE DIMENSION OF A PREIMAGE $\phi^{-1}(c)$ FOR REGULAR VALUES OF c. (A) GRAPH OF $\phi(\mathbf{x})$ IN EQUATION 3.8. $\phi(\mathbf{x})$ HAS A CRITICAL VALUE $c = 0$ AND A REGULAR VALUE $c = 0.5$. (B) PREIMAGES $\phi^{-1}(0)$ (SOLID BLACK) AND $\phi^{-1}(0.5)$ (DASHED LINE). THEOREM 3.24 TELLS US THAT $\dim(\phi^{-1}(0.5)) = 1$, BUT TELLS US NOTHING ABOUT $\dim(\phi^{-1}(0))$.

3.1.3 Nonsmooth Preimage Theorems

We are now going to develop a nonsmooth preimage theorem that we will use extensively in the remainder of this thesis. For a function $\phi : X \to \mathbb{R}$, recall that the preimage of a function value $c \in \mathbb{R}$ is the set $\phi^{-1}(c) = \{\mathbf{x} \in X : \phi(\mathbf{x}) = c\}$. The preimage theorem provides a way to determine the dimension of $\phi^{-1}(c)$ based on whether or not c is a critical or regular value. Consider Figure 3.5(a) which shows a graph of the function

$$\phi(\mathbf{x}) = \begin{cases} x_1 + 1 & x_1 < -1 \\ 0 & -1 \le x_1 \le 1 \\ x_1 - 1 & 1 < x_1 < 2 \\ -x_1 + 3 & x_1 \ge 2 \end{cases} + \begin{cases} x_2 + 1 & x_2 < -1 \\ 0 & -1 \le x_2 \le 1 \\ x_2 - 1 & 1 < x_2 < 2 \\ -x_2 + 3 & x_2 \ge 2 \end{cases} \tag{3.8}$$

For Equation 3.8, the value $c = 0$ is a critical value since the set $\phi^{-1}(0)$ contains critical values (a square plateau). Figure 3.5(b) shows $\phi^{-1}(0)$ as a square region and a solid black looping segment. Notice in particular that $\dim(\phi^{-1}(0)) = \dim(X)$. On the other hand, the value $c = 0.5$ is a regular value since the set $\phi^{-1}(0.5)$ doesn't contain any critical points. Figure 3.5(b) shows that $\phi^{-1}(0.5)$ (dashed line) has dimension $\dim(\phi^{-1}(0.5)) = \dim(X) - 1$. The following preimage Theorem 3.24 says that whenever c is a regular value, the dimension of $\phi^{-1}(c)$ is one less than the dimension of its domain. We start by presenting Theorem 3.22 which allows us to develop a nonsmooth preimage theorem, similar to the smooth variants in [24].

Theorem 3.22 (Local Inversion Theorem [56]) If $\phi : \mathbb{R}^n \to \mathbb{R}^n$ is locally Lipschitz continuous and each

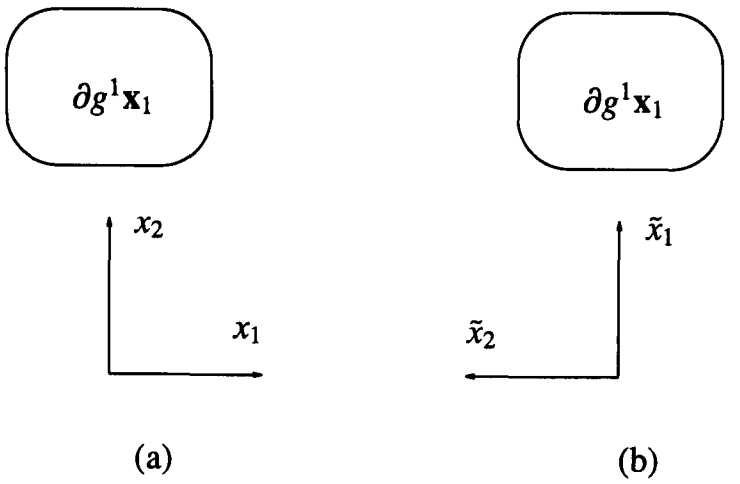


FIGURE 3.6: LINEAR CHANGE OF COORDINATES RESULT IN FULL-RANK $\partial G(\mathbf{x})$. WITH THE ORIGINAL SET OF COORDINATES (x_1, x_2) IN (A), SOME ELEMENTS OF $g^1(\mathbf{x})_1$ HAVE A NON-ZERO x_1 COORDINATE, LEADING TO ELEMENTS OF $\partial G(\mathbf{x})$ THAT ARE NOT FULL-RANK. BY APPLYING A CHANGE OF COORDINATES, LEADING TO NEW COORDINATES $(\tilde{x}_1, \tilde{x}_2)$ IN (B), WE ENSURE THIS IS NO LONGER THE CASE.

element of the generalised Jacobian[2] $\partial\phi(\mathbf{x})$ is invertible, then ϕ maps a neighbourhood of $\mathbf{x} \in \mathbb{R}^n$ homeomorphically onto a neighbourhood of $y = \phi(\mathbf{x})$. ■

In the smooth case [24], ϕ can be shown to map a neighbourhood of $\mathbf{x} \in \mathbb{R}^n$ *diffeomorphically* (a smooth homeomorphism) onto a neighbourhood of $y = \phi(\mathbf{x})$. The proof of Theorem 3.24 is thus quite similar to the proof using diffeomorphisms[24]—we just lack smoothness.

Definition 3.23 The **canonical submersion** $\pi : \mathbb{R}^k \to \mathbb{R}^l$ is the standard projection of $\mathbb{R}^k$ onto $\mathbb{R}^l$ for $k \geq l$, in which $(x_1, \cdots, x_k) \to (x_1, \cdots, x_l)$. □

Theorem 3.24 If $y \in \mathbb{R}$ is a regular value of $\phi : \mathbb{R}^n \to \mathbb{R}$, then the preimage $\phi^{-1}(y)$ is a set with dimension $\dim(\phi^{-1}(y)) = n - 1$.

Proof By Theorem 3.21, if y is a regular value, then $|d\phi|(\mathbf{x}) > 0$ for all $x \in Z_y$ and so $\mathbf{0} \notin \partial\phi(\mathbf{x})$ for all $\mathbf{x} \in Z_y$. Consider the trivial local parameterisation diagram where $g = \phi$

$$\begin{array}{ccc} \mathbb{R}^n & \xrightarrow{\phi} & \mathbb{R} \\ {\scriptstyle I}\uparrow & & \uparrow{\scriptstyle I} \\ \mathbb{R}^n & \xrightarrow{g} & \mathbb{R} \end{array}$$

We seek to modify g and apply the Local Inversion Theorem. Define $G : \mathbb{R}^n \to \mathbb{R}^n$ by $G(\mathbf{x}) = (g(\mathbf{x}), x_2,$

[2]The generalised Jacobian is an analogue of the generalised derivative for vector valued functions. i.e. for $F : \mathbb{R}^n \to \mathbb{R}^m$, $\partial F(\mathbf{x}) = \text{co}\{\lim JF(\mathbf{x_i}) : \mathbf{x}_i \to \mathbf{x}, \mathbf{x}_i \notin \Omega_F\}$, where Ω_F is the set where F is not differentiable. Compare this to Equation ■.

$\cdots, x_n)$, where $\mathbf{x} = (x_1, \cdots, x_n)$.

We now show that all elements of $\partial G(\mathbf{x})$ are invertible. We know from [16] that $\partial G(\mathbf{x}) \subseteq \partial g^1(\mathbf{x}) \times \cdots \times \partial g^k(\mathbf{x})$, where the latter denotes the set of all matrices whose i^{th} row belongs to the generalised gradient of the coordinate function $\partial g^i(\mathbf{x})$ ($g^1(\mathbf{x}) = g(\mathbf{x})$ and $g^i(\mathbf{x}) = x_i, i \in \{2, \cdots, k\}$). Since $\partial g^1(\mathbf{x})$ is invertible (invertible means $\mathbf{0} \notin \partial g^1(\mathbf{x})$), we can have a linear change of coordinates so that $\partial g^1(\mathbf{x})$ always has a nonzero x_1 coordinate $g^1(\mathbf{x})_1$ (see Figure 3.6), and the elements of $\partial G(\mathbf{x})$ have the full-rank form

$$\begin{bmatrix} \partial g^1(\mathbf{x})_1 & \partial g^1(\mathbf{x})_2 & \cdots & \partial g^1(\mathbf{x})_k \\ 0 & 1 & \cdots & 0 \\ & & \ddots & \\ 0 & 0 & \cdots & 1 \end{bmatrix}$$

By construction $g = \pi \circ G$, so $g \circ G^{-1}$ is the canonical submersion π. The following local parameterisation diagram commutes.

$$\begin{array}{ccc} \mathbb{R}^n & \xrightarrow{\phi} & \mathbb{R} \\ {\scriptstyle G^{-1}}\uparrow & & \uparrow{\scriptstyle I} \\ \mathbb{R}^n & \xrightarrow{\pi} & \mathbb{R} \end{array}$$

Locally, $\phi(\mathbf{x})$ is equivalent to the canonical submersion π (the projection of $\mathbb{R}^n$ onto its first component), so select local coordinates around $\mathbf{x} \in \mathbb{R}^n$ and $y \in \mathbb{R}$ where y corresponds to 0 and $\phi(x_1, \cdots, x_n) = x_1$. Let V denote the neighbourhood of $\mathbf{x}$ in coordinate system $(x_1, \cdots, x_n)$. Then $(\phi \circ G^{-1})^{-1}(y) \cap V$ is the set of points where $x_1 = 0$, a relatively open subset of ϕ^{-1} having dimension $n - 1$. By the arbitrariness of $\mathbf{x}$, $\dim(\phi^{-1}(y)) = n - 1$. ■

The previous Theorem applies to functions defined on $\mathbb{R}^n$, and has a global flavour to it, in that $y = \phi(\mathbf{x})$ is a regular value. We actually desire a more local Theorem since our distance measurements are just that—local. Just because the robot is in a regular configuration $\mathbf{q}$ with $D(\mathbf{q}) = d$ doesn't mean all configurations with $D(\mathbf{q}) = d$ are regular. We have the following for any set S locally homeomorphic to $\mathbb{R}^n$.

Corollary 3.25 If $\mathbf{x} \in S$ is a regular point of $\phi : S \to \mathbb{R}$, with $\phi(\mathbf{x}) = y$, then the preimage $\phi^{-1}(y)$ has dimension $\dim(\phi^{-1}(y)) = n - 1$ in a neighbourhood of $\mathbf{x}$.

Proof If $\mathbf{x}$ is a regular point of S, then there exists a continuous transformation $\mathcal{H}(\mathbf{v}, t)$ satisfying Equations 3.7 in a neighbourhood $B_\delta(\mathbf{x})$ ($X = S$ in the notation of Equation 3.7). Let $G : S \to \mathbb{R}^n$ be a local homeomorphism between S and $\mathbb{R}^n$. Since $\mathcal{H}(\mathbf{v}, t)$ is continuous, so is $G \circ \mathcal{H}$ and thus $G(\mathbf{x}) \in \mathbb{R}^n$ is a regular

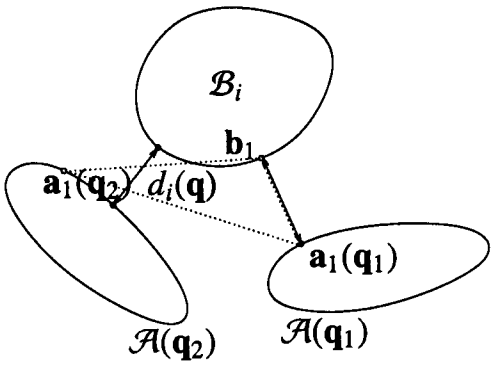


FIGURE 3.7: THE DISTANCE FUNCTION $d_i(\mathbf{q})$ IS LIPSCHITZ IN $SE(2)$ AND $SE(3)$. THE FIGURE SHOWS A CONVEX OBSTACLE $\mathcal{B}_i$ AND THE ROBOT IN TWO CONFIGURATIONS $\mathcal{A}(\mathbf{q}_1)$ AND $\mathcal{A}(\mathbf{q}_2)$. THE LIPSCHITZ CONDITION IMPLIES THE DISTANCE CHANGES LESS THAN THE MAXIMUM DISTANCE MOVED BY ANY PARTICULAR POINT ON THE ROBOT. THE TRIANGLE INEQUALITY IN EQUATION 3.10 IS SHOWN IN DOTTED LINES.

point of $\phi_G = G^{-1} \circ \phi$ where $\phi_G(G(\mathbf{x})) = y$. By Theorem 3.24 $\phi_G^{-1}(y)$ has dimension $n-1$ in a neighbourhood of $G(\mathbf{x})$, and since homeomorphism preserves dimension, $G^{-1}(\phi_G^{-1}(y)) \cap B_\delta(\mathbf{x}) = \phi^{-1}(y) \cap B_\delta(\mathbf{x})$ has dimension $n-1$. ■

3.2 Nonsmooth Analysis of the Distance Function

The robot $\mathcal{A}$ will usually operate in an environment with a finite number of convex obstacles $\mathcal{B}_i$. Assume $\mathcal{A}$ and the $\mathcal{B}_i$s are all convex and compact. Furthermore, assume that the $\mathcal{B}_i$s enclose, or bound, a subset of the workspace. This ensures that there are 'enough' obstacles to construct a roadmap in the following chapter. The 'reachable' configuration space $C_r \subseteq SE(2)$ or $SE(3)$ is then also bounded. For the remainder of this thesis, assume $\mathbf{q} \in C_r$. We start with Lemma 3.26 which shows that $D(\mathbf{q})$ can never reach $\pm\infty$.

Lemma 3.26 $D(\mathbf{q})$ is a function bounded from above and below (totally bounded).

Proof By convention, the individual $d_i(\mathbf{q})$s (and thus $D(\mathbf{q})$) are bounded below by the value zero. Since the robot operates in a bounded workspace, the individual $d_i(\mathbf{q})$s (and thus $D(\mathbf{q})$) are bounded above by a finite distance. ■

The generalised gradient is defined only for Lipschitz functions, so we evidently must show that $d_i(\mathbf{q})$ and $D(\mathbf{q})$ satisfy the Lipschitz condition. The Lipschitz condition requires that the domain X be a metric space. From now on, assume that the metric $d(\mathbf{q}_1, \mathbf{q}_2)$ on C is the so called "maximum distance metric"

$$d(\mathbf{q}_1, \mathbf{q}_2) = \max_{\mathbf{a} \in \mathcal{A}} \|\mathbf{a}(\mathbf{q}_1) - \mathbf{a}(\mathbf{q}_2)\|$$

which corresponds to the maximum distance moved (in the ambient Euclidean workspace $\mathcal{W} = \mathbb{R}^2$ or $\mathbb{R}^3$) by any of the points in $\mathcal{A}$.

Lemma 3.27 $d_i(\mathbf{q})$ satisfies the global Lipschitz condition $|d_i(\mathbf{q}_1) - d_i(\mathbf{q}_2)| \leq d(\mathbf{q}_1 - \mathbf{q}_2)$.

Proof By definition, $d_i(\mathbf{q}_1) \geq \| \mathbf{a}_1(\mathbf{q}_1) - \mathbf{b}_1 \|$. As well,

$$\begin{aligned} d_i(\mathbf{q}_2) &\leq \| \mathbf{a}_1(\mathbf{q}_2) - \mathbf{b}_1 \| \\ &\leq \| \mathbf{a}_1(\mathbf{q}_2) - \mathbf{a}_1(\mathbf{q}_1) \| + \| \mathbf{a}_1(\mathbf{q}_1) - \mathbf{b}_1 \| \text{ (triangle inequality)} \\ &\leq d(\mathbf{q}_1 - \mathbf{q}_2) + d_i(\mathbf{q}_1) \end{aligned} \tag{3.9}$$

where $\mathbf{a}_1(\mathbf{q}_1)$ is the closest point on the robot in configuration $\mathbf{q}_1$, and $\mathbf{a}_1(\mathbf{q}_2)$ is the location of that same point in configuration $\mathbf{q}_2$ (it may no longer be the closest point). Since we can switch roles of $\mathbf{q}_1$ and $\mathbf{q}_2$ we get the condition $|d_i(\mathbf{q}_1) - d_i(\mathbf{q}_2)| \leq d(\mathbf{q}_1 - \mathbf{q}_2)$. ■

Lemma 3.28 $D(\mathbf{q})$ is a locally Lipschitz function.

Proof $D(\mathbf{q})$ is locally Lipschitz since a function $g(\mathbf{x}) = \min_i\{f_i(\mathbf{x})\}$ expressed as the maximum (or minimum) of a finite number of locally Lipschitz functions $f_1(\mathbf{x}), \cdots, f_n(\mathbf{x})$ is itself locally Lipschitz ([16] p. 47). ■

The Lipschitz condition places an upper bound on the nearest distance change $|D(\mathbf{q}_1) - D(\mathbf{q}_2)|$ when the robot moves from configuration $\mathbf{q}_1$ to $\mathbf{q}_2$. More importantly for our use, when given a desired change in the minimum distance $|D(\mathbf{q}_1) - D(\mathbf{q}_2)|$, the Lipschitz condition implies a lower bound on the motion required in configuration space $d(\mathbf{q}_1 - \mathbf{q}_2)$ required to effect the distance change. It is also important that $d_i(\mathbf{q})$ and $D(\mathbf{q})$ satisfy the Palais-Smale condition (PS) because nearly all the critical point literature, both smooth and nonsmooth, is developed in the context of functions that satisfy (PS). Analysing critical points that occur at infinity is a non-trivial thing.

Theorem 3.29 $D(\mathbf{q})$ satisfies (PS).

Proof Since C_r is compact, any sequence $\mathbf{q}_n \subseteq C_r$ has a convergent subsequence and $D(\mathbf{q})$ thus satisfies (PS). ■

The (PS) condition also ensures that $D(\mathbf{q})$ is proper (see the Glossary for a definition of a *proper map*) and that the set $D(K)$ is closed in the reals. This immediately gives Lemma 3.30.

Lemma 3.30 The set of critical points K of $D(\mathbf{q})$ is a compact set in C_r.

Proof The (PS) condition immediately implies that each critical level K_c is compact and that $D(K)$ is closed in $\mathbb{R}$, but says nothing about the set of all critical points K. However, since $D(\mathbf{q})$ is bounded and $D(K)$ is closed, $D(K)$ is also compact. C_r is closed, so by the continuity of $D(\mathbf{q})$, $D^{-1}(K) = K$ is a closed set. K is compact by the compactness of a closed subset of a compact metric space C_r, so the lemma follows. ■

The planner developed in subsequent chapters operates by constructing a roadmap that connects all components of the critical set K. Now, if a planner based on searching for interesting areas of configuration space (the set K) is to terminate, we must know *a priori* that K has a finite number of components. Consider what we know about our function $D(\mathbf{q})$; it is Lipschitz, totally bounded, the critical set K is compact, as are the individual sets K_c. These characteristics however are not sufficient to guarantee a finite number of components in K. Consider the function $\phi : [-1, 1] \rightarrow \mathbb{R}$

$$\phi(x) = \begin{cases} x^2 sin(1/x), & -1 \leq x \leq 1 \text{ and } x \neq 0 \\ 0, & x = 0 \end{cases}$$

where $K_{\phi(x)} = \{0\} \cup \{2/\pi, 2/3\pi, 2/5\pi, \cdots\}$, which has an infinite number of components. Since the individual K_cs are disjoint compact sets, we know the distance between each set must be positive. However, in order to establish a finite number of critical sets in configuration space, we need to show that all the K_cs are in fact *disconnected*. We use the notation $\check{K}_i$ to represent a single disjoint connected component of K. The individual $\check{K}_i$s may or may not be subsets of the same K_cs. We present the following as a proposition, and leave the proof until the end of Section 3.2.3.

Proposition 3.31 Each $\check{K}_i$ is disconnected from $\bigcup_{j \neq i} \check{K}_j$ (union of all $\check{K}_j$s not equal to $\check{K}_i$). ■

This Proposition leads immediately to a proof that K consists of a finite number of disconnected components.

Lemma 3.32 There are a finite number of critical level set components $\check{K}_1, \cdots, \check{K}_n$.

Proof We use proof by contradiction. By definition of the critical set, $K = \bigcup_i \check{K}_i$, and by Proposition 3.31, each $\check{K}_i$ is disconnected from $\bigcup_{j \neq i} \check{K}_j$. Assume there are an infinite number of $\check{K}_i$s. Since each critical level set is disconnected, we can construct an open cover of K using the sets C_i, such that each $\check{K}_i$ is contained in one and only one C_i. Since each $\check{K}_i$ is disconnected from the rest of K, C_i is simply any epsilon neighbourhood nbhd($\check{K}_i$) such that nbhd($\check{K}_i$) $\cap \bigcup_{j \neq i} \check{K}_j = \varnothing$. By construction, each C_i contains

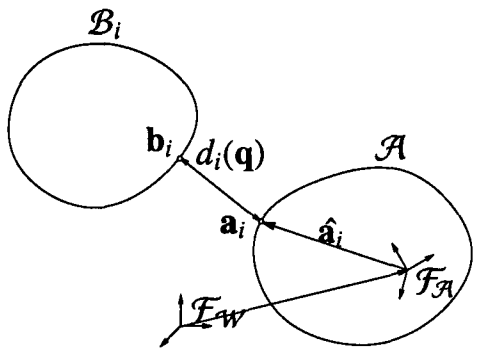


FIGURE 3.8: GRADIENT OF DISTANCE IN $SE(3)$. $\mathbf{a}$ AND $\mathbf{b}$ ARE THE CLOSEST POINTS ON THE ROBOT $\mathcal{A}$ AND OBSTACLE $\mathcal{B}_i$ RESPECTIVELY. $\hat{\mathbf{a}}$ ARE THE COORDINATES OF $\mathbf{a}$ IN $\mathcal{A}$'S COORDINATE FRAME $\mathcal{F}_{\mathcal{A}}$. $\mathcal{F}_{\mathcal{W}}$ IS THE WORLD COORDINATE FRAME.

only one $\check{K}_i$, so any finite collection of C_is necessarily misses some $\check{K}_i$s. This contradicts the fact that K is compact, in that every infinite open cover must have a finite open cover. ■

3.2.1 Generalised Gradients of the Distance Functions $\partial \mathbf{d_i(q)}$ and $\partial \mathbf{D(q)}$

In this section we develop expressions for the gradient of the distance function of a free floating convex body with the usual six degrees of freedom. The gradient $\nabla d_i(\mathbf{q})$ derived in this section plays a critical role in calculation of the generalised gradients $\partial d_i(\mathbf{q})$ and $\partial D(\mathbf{q})$. These results and proof are constructed in $SE(3)$, but the results are also applicable to $SE(2)$ by discarding the z, θ and ψ terms.

Figure 3.8 shows the environment used to determine the generalised gradient. The (single) points on the robot $\mathcal{A}$ and obstacle $\mathcal{B}_i$ closest to each other are $\mathbf{a} = [a_x, a_y, a_z]^T$ and $\mathbf{b} = [b_x, b_y, b_z]^T$ respectively. We use the additional notation $\hat{\mathbf{a}} = [\hat{a}_x, \hat{a}_y, \hat{a}_z]^T$ to represent the coordinates of $\mathbf{a}$ in $\mathcal{A}$'s coordinate frame $\mathcal{F}_{\mathcal{A}}$. The minimum distance then is given by

$$d_i(\mathbf{q}) = \sqrt{(a_x - b_x)^2 + (a_y - b_y)^2 + (a_z - b_z)^2} \tag{3.10}$$

Let $\mathbf{r} = [x, y, z]^T$ be the location of the origin of $\mathcal{F}_{\mathcal{A}}$ with respect to $\mathcal{F}_{\mathcal{W}}$. To parameterise the rotational degrees of freedom, we choose the Euler angle parameterisation of $SO(3)$ with the three angles ϕ, θ and ψ (see [38] for an excellent description). ϕ represents a rotation about the z-axis of $\mathcal{F}_{\mathcal{A}}$ resulting in a new coordinate frame $\mathcal{F}'_{\mathcal{A}}$, θ represents a rotation about the y-axis of $\mathcal{F}'_{\mathcal{A}}$ leading to another new coordinate frame $\mathcal{F}''_{\mathcal{A}}$ and ψ represents a rotation about the z-axis of $\mathcal{F}''_{\mathcal{A}}$ resulting in the final configuration of $\mathcal{A}$'s coordinate

frame. Using the shorthand notation $c = \cos$ and $s = \sin$, the composite rotation matrix is given by

$$\begin{aligned}
\mathbf{R}_{\phi,\theta,\psi} &= \mathbf{R}_{\phi}\mathbf{R}_{\theta}\mathbf{R}_{\psi} \\
&= \begin{bmatrix} c\phi & -s\phi & 0 \\ s\phi & c\phi & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} c\theta & 0 & s\theta \\ 0 & 1 & 0 \\ -s\theta & 0 & c\theta \end{bmatrix}\begin{bmatrix} c\psi & -s\psi & 0 \\ s\psi & c\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \\
&= \begin{bmatrix} c\phi c\theta c\psi - s\phi s\psi & -c\phi c\theta s\psi - s\phi c\psi & c\phi s\theta \\ s\phi c\theta c\psi + c\phi s\psi & -s\phi c\theta s\psi + c\phi c\psi & s\phi s\theta \\ -s\theta c\psi & s\theta s\psi & c\theta \end{bmatrix}
\end{aligned}$$

We use the notation R_{ij} to represent the nine entries of the composite rotation matrix. This helps keep the subsequent matrix formulae to a reasonable size. The world coordinates of $\hat{\mathbf{a}}$ are denoted $\mathbf{U} = R_{\phi,\theta,\psi}\hat{\mathbf{a}}$ and the world coordinates of $\mathbf{a}$ are

$$\begin{aligned}
\mathbf{a} &= \mathbf{r} + \mathbf{U} \\
&= \begin{bmatrix} x + \hat{a}_x R_{11} + \hat{a}_y R_{12} + \hat{a}_z R_{13} \\ y + \hat{a}_x R_{21} + \hat{a}_y R_{22} + \hat{a}_z R_{23} \\ z + \hat{a}_x R_{31} + \hat{a}_y R_{32} + \hat{a}_z R_{33} \end{bmatrix}
\end{aligned}$$

The main result of this section is the following theorem. This theorem has in fact been verified for the case of a point robot in [16], and although rather intuitive is not a trivial matter. In the follow Theorem, the notation $(\mathbf{a} - \mathbf{b})_x$ indicates the x-component on the vector $(\mathbf{a} - \mathbf{b})$.

Theorem 3.33 $\nabla d_i(\mathbf{q})$ exists and is equal to the 6×1 vector

$$\nabla d_i(\mathbf{q}) = \frac{1}{d_i(\mathbf{q})}\begin{bmatrix} (\mathbf{a} - \mathbf{b})_x \\ (\mathbf{a} - \mathbf{b})_y \\ (\mathbf{a} - \mathbf{b})_z \\ (\mathbf{U} \times (\mathbf{a} - \mathbf{b}))_z \\ (\mathbf{R}_{\phi}^{-1}(\mathbf{U} \times (\mathbf{a} - \mathbf{b})))_y \\ (\mathbf{R}_{\theta}^{-1}\mathbf{R}_{\phi}^{-1}(\mathbf{U} \times (\mathbf{a} - \mathbf{b})))_z \end{bmatrix} \tag{3.11}$$

if and only if there are single unique points $\mathbf{a}$ and $\mathbf{b}$ on the robot $\mathcal{A}$ and obstacle $\mathcal{B}_i$ respectively, such that $\|\mathbf{a} - \mathbf{b}\| = d_i(\mathbf{q}) > 0$

Proof (**IF**) Appendices A and B establish that $\nabla d_i(\mathbf{q})$ exists and is equal to Equation 3.11 if there are single unique points $\mathbf{a}$ and $\mathbf{b}$ on the robot $\mathcal{A}$ and obstacle $\mathcal{B}_i$ respectively.

(**ONLY IF**) Without loss of generality, assume there are multiple closest points on the robot. Figure 3.10 and Lemma 3.34 in the following section show that $\partial d_i(\mathbf{q})$ contains multiple points, so $\nabla d_i(\mathbf{q})$ does not exist. ■

It is worth noting that the same gradient result can be obtained using standard calculus differentials if $(\mathbf{a} - \mathbf{b})$ is assumed perpendicular at $\mathbf{a}$ (resp. $\mathbf{b}$) to the tangent space $T_{\mathbf{a}}\text{bndy}(\mathcal{A})$ (resp. $T_{\mathbf{b}}\text{bndy}(\mathcal{B}_i)$) of the boundary of $\mathcal{A}$ (resp $\mathcal{B}_i$) at all configurations $\mathbf{q}$. This is the approach taken in [15] for $SE(2)$, in which the single object distance function $d_i(\mathbf{q})$ is incorrectly assumed everywhere differentiable.

3.2.2 Multiple Closest Points

By Theorem 3.33, when there are multiple closest points on the robot $\mathcal{A}$, the generalised gradient $\partial d_i(\mathbf{q})$ must not be a singleton. When determining the $\partial d_i(\mathbf{q})$ we must remember to address the fact that, by convexity of $\mathcal{A}$ and the $\mathcal{B}_i$s , there will be an infinite number of closest points if more than one closest point exists. We denote these closest points $\mathbf{a}_t(\mathbf{q})$ indexed by $t \in T$ where T is an abstract indexing space (with an infinite number of points by necessity). We denote by $\text{dist}(\mathbf{a}_t(\mathbf{q}), \mathcal{B}_i)$ the minimum distance between $\mathbf{a}_t(\mathbf{q})$ and $\mathcal{B}_i$ and note that by definition $\text{dist}(\mathbf{a}_t(\mathbf{q}), \mathcal{B}_i) = d_i(\mathbf{q})$ for all $t \in T$. We then have the following alternate definition of $d_i(\mathbf{q})$

$$d_i(\mathbf{q}) = \min\{\text{dist}(\mathbf{a}_t(\mathbf{q}), \mathcal{B}_i) : \mathbf{a}_t(\mathbf{q}) \in \mathcal{A}, \text{dist}(\mathbf{a}_t(\mathbf{q}), \mathcal{B}_i) = d_i(\mathbf{q})\} \tag{3.12}$$

Lemma 3.34 The generalised gradient when there are multiple closest points is given by

$$\partial d_i(\mathbf{q}) = \text{co}\{\nabla\text{dist}(\mathbf{a}_t(\mathbf{q}), \mathcal{B}_i) : \mathbf{a}_t(\mathbf{q}) \in \mathcal{A}, \text{dist}(\mathbf{a}_t(\mathbf{q}), \mathcal{B}_i) = d_i(\mathbf{q})\}$$

Proof Since $\mathcal{B}_i$ is convex, the point distance function is everywhere differentiable, so $\text{dist}(\mathbf{a}_t(\mathbf{q}), \mathcal{B}_i)$ is everywhere differentiable and $\nabla\text{dist}(\mathbf{a}_t(\mathbf{q}), \mathcal{B}_i)$ exists. We also note

$$\min_i\{f_i\} = -\max_i\{-f_i\}$$

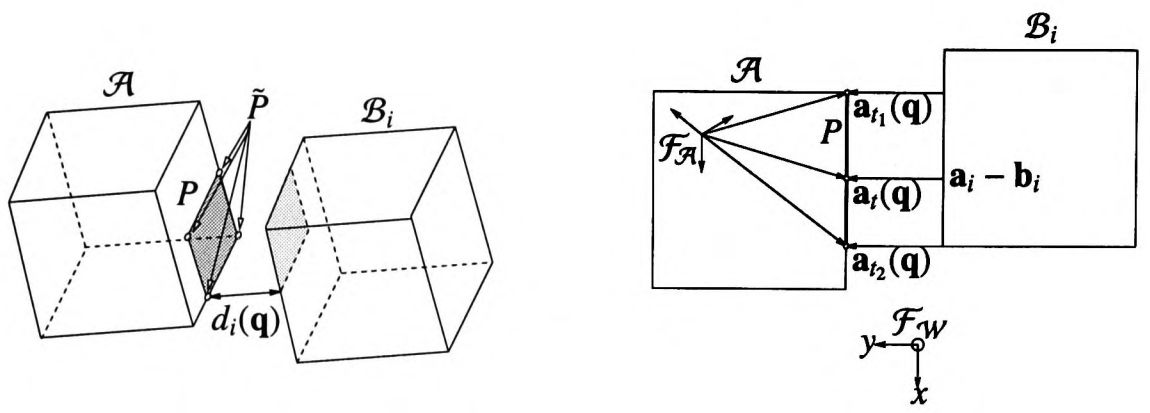


FIGURE 3.9: THE ROBOT $\mathcal{A}$ MAY HAVE MULTIPLE CLOSEST POINTS TO THE OBSTACLES $\mathcal{B}_i$. (A) A ROBOT IN $SE(3)$ WITH A SET OF POINTS P EQUIDISTANT FROM THE OBSTACLE $\mathcal{B}_i$ AND THE SET $\tilde{P}$ OF EXTREMAL POINTS OF P. (B) THE SAME CONFIGURATION FROM A DIFFERENT VIEWPOINT. THE FOURTH GRADIENT TERM $U_x ab_y - U_y ab_x$ ACHIEVES A MINIMUM OVER P AT POINT $\mathbf{a}_{j_1}(\mathbf{q})$ AND A MAXIMUM AT POINT $\mathbf{a}_{j_1}(\mathbf{q})$. IT IS SUFFICIENT TO CONSIDER ONLY THE *extremal points* OF P WHEN DETERMINING THE GENERALISED GRADIENT.

By the smoothness of $f_i(\mathbf{q}) = \text{dist}(\mathbf{a}_t(\mathbf{q}), \mathcal{B}_i)$ and Corollary 3.9, $\partial \min_i\{f_i(\mathbf{q})\} = \partial \max_i\{f_i(\mathbf{q})\}$. Since C is finite dimensional, the Lemma follows from the differentiability of $\text{dist}(\mathbf{a}_t(\mathbf{q}), \mathcal{B}_i)$ and application of Theorem 2.8.6 in [16] ■

Lemma 3.34 states that the generalised gradient is determined by taking the convex hull of an infinite number of single point gradient vectors. One may ask if it is possible to determine the generalised gradient $d_i(\mathbf{q})$ using a finite number of points, since it is impossible to determine the convex hull of an infinite number of points in a reasonable time. Figure 3.9(a) shows a typical problem in $SE(3)$ with a cuboid robot $\mathcal{A}$ and a cuboid obstacle $\mathcal{B}_i$. The shaded area P on the face of $\mathcal{A}$ is the set of points $\mathbf{a}_t(\mathbf{q}) \in P$ that are distance $d_i(\mathbf{q})$ from the obstacle. We now show that the generalised gradient can be calculated using only the points $\mathbf{a}_t(\mathbf{q})$ that are *extremal points* of P (a subset of the support points of P). These extremal points are either the vertices of the polygonal area P or endpoints of the line segment P. We use the notation $\tilde{P}$ to represent this finite subset of P. For each $\mathbf{a}_t(\mathbf{q})$, there is a corresponding $\mathbf{b}_t(\mathbf{q})$ representing the corresponding closest point on the obstacle.

Lemma 3.35 $\partial d_i(\mathbf{q}) = \text{co}\{\nabla \text{dist}(\mathbf{a}_t(\mathbf{q}), \mathcal{B}_i); \mathbf{a}_t(\mathbf{q}) \in \tilde{P}\}$

Proof Equation 3.11 is used to determine $\nabla \text{dist}(\mathbf{a}_t(\mathbf{q}), \mathcal{B}_i)$ by substituting $\mathbf{a}_i = \mathbf{a}_t(\mathbf{q})$ and $\mathbf{b}_i = \mathbf{b}_t(\mathbf{q})$. Without loss of generality we can apply a linear change of coordinates to the world frame $\mathcal{F}_{\mathcal{W}}$ such that one of the primary axes is parallel to $\mathbf{a} - \mathbf{b}$ and the other two are perpendicular, so that all but one of ab_x, ab_y and ab_z are zero (recall $ab_x = (a_x - b_x)$ etc.).

The first three terms (ab_x, ab_y and ab_z) of Equation 3.11 are equal for all $\mathbf{a}_t(\mathbf{q})$, so certainly the finite

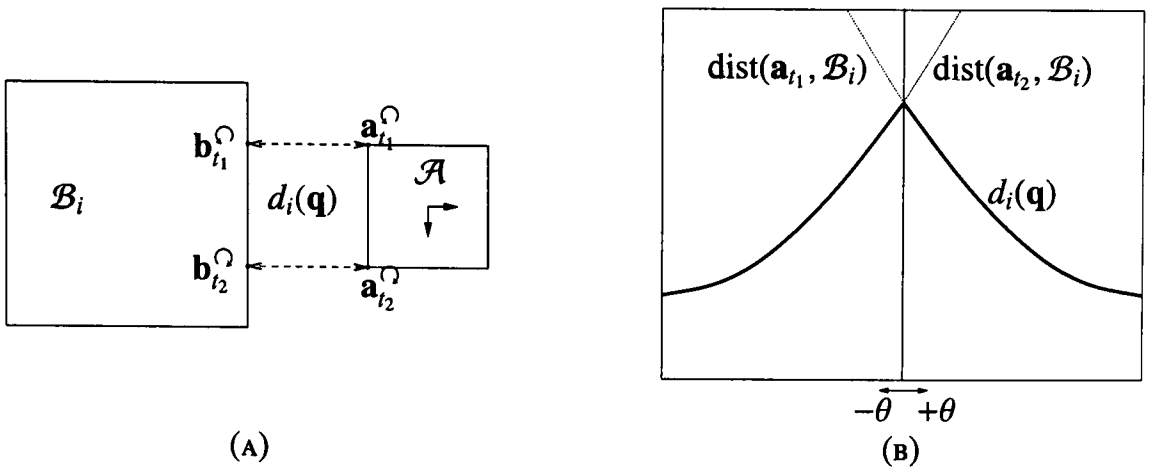


FIGURE 3.10: $d_i(\mathbf{q})$ CAN HAVE NONSMOOTH POINTS. (A) A ROBOT $\mathcal{A}$ AT CONFIGURATION $\mathbf{q} = [x, y, \theta]^T$ WITH A NUMBER OF CLOSEST POINTS TO THE OBSTACLE $\mathcal{B}_i$. IF $\mathcal{A}$ ROTATES CLOCKWISE (RESP. COUNTERCLOCKWISE) THE CLOSEST POINT BECOMES $\mathbf{a}_{t_2}$ (RESP. $\mathbf{a}_{t_1}$) AND THE DISTANCE TO THE CLOSEST POINT ON THE OBSTACLE $\mathbf{b}_{t_2}$ (RESP. $\mathbf{b}_{t_1}$) IS $\text{dist}(\mathbf{a}_{t_2}(\mathbf{q}), \mathcal{B}_i)$ (RESP. $\text{dist}(\mathbf{a}_{t_1}(\mathbf{q}), \mathcal{B}_i)$). (B) A GRAPH OF $\text{dist}(\mathbf{a}_{t_2}(\mathbf{q}), \mathcal{B}_i)$ AND $\text{dist}(\mathbf{a}_{t_1}(\mathbf{q}), \mathcal{B}_i)$ ALONG WITH $d_i(\mathbf{q})$ FOR θ WITH x, y HELD CONSTANT. WE CAN SEE THAT $d_i(\mathbf{q})$ IS NONSMOOTH AND NON-CONVEX AT CONFIGURATION $\mathbf{q}$.

subset $\tilde{P}$ is sufficient to determine the first three terms. The fourth term $(U_x ab_y - U_y ab_x)$ is handled with the assistance of Figure 3.9(b) which shows a different view of Figure 3.9(a). Without loss of generality assume $ab_x = 0$, and since ab_y is a constant we need only address the term $U_x^t = \mathbf{a}_t(\mathbf{q}) - [x, y, z]^T$. It is easily seen that U_x^t achieves a minimum at extremal point $\mathbf{a}_{t_1}(\mathbf{q})$ and maximum at the extremal point $\mathbf{a}_{t_2}(\mathbf{q})$ such that $U_x^{t_1} \leq U_x^t \leq U_x^{t_2}$. Thus it is sufficient to consider only the extremal points for the fourth term. The fifth and sixth terms are handled similarly. Since the linear change of coordinates of $\mathcal{F}_W$ is arbitrary, the lemma follows. ■

Lemma 3.35 does not apply if P is not a convex polygon (or line segment), so we assume from now on that $\mathcal{A}$ and all $\mathcal{B}_i$s are convex polytopes or polygons.

3.2.3 Detection of Critical Points

Recall that a *critical point* $p \in X$ of a smooth function $\phi : X \to \mathbb{R}$ occurs when the partial derivatives are all zero

$$\frac{\partial \phi}{\partial x_1}(p) = \cdots = \frac{\partial \phi}{\partial x_n}(p) = 0$$

Since both the $d_i(\mathbf{q})$s and $D(\mathbf{q})$ may have nonsmooth points, we need to address where nonsmooth critical points can occur. Although $d_i(\mathbf{q})$ is in general nonsmooth, it does not have any nonsmooth (or smooth) critical points in C_{free} since Theorem 3.33 establishes that the generalised gradient $\partial d_i(\mathbf{q})$ never contains the point $\mathbf{0}$, since $\mathbf{a} - \mathbf{b}$ is always a single nonzero value. By Theorem 3.21, all of the points of $d_i(\mathbf{q})$ are

regular. On the other hand, the distance function $D(\mathbf{q})$ is a pointwise minimum function, so the generalised gradient satisfies

$$\partial D(\mathbf{q}) \subseteq \text{co}\{\partial d_i(\mathbf{q}) : i \in I(\mathbf{q})\} \tag{3.13}$$

In [16], the inclusions in the generalised gradient calculus are shown to be equalities if the functions $d_i(\mathbf{q})$ are "regular" in the sense that the generalised derivative is equivalent to the standard directional derivative (see [16]). By Lemma 3.36, $d_i(\mathbf{q})$ is not "regular", so $\partial D(\mathbf{q})$ may not be strictly equal to the convex hull.

Lemma 3.36 In general, $d_i(\mathbf{q})$ is nonsmooth and non-convex.

Proof Figure 3.10 shows an example of a nonsmooth point of $d_i(\mathbf{q})$ with respect to the angle coordinate θ. Figure 3.10(a) shows the robot in configuration $(x, y, \theta = 0)$. If $\mathcal{A}$ rotates clockwise (resp. counter-clockwise) the closest point becomes $\mathbf{a}_{t_2}$ (resp. $\mathbf{a}_{t_1}$) and the distance to the closest point on the obstacle $\mathbf{b}_{t_2}$ (resp. $\mathbf{b}_{t_1}$) is $\text{dist}(\mathbf{a}_{t_2}(\mathbf{q}), \mathcal{B}_i)$ (resp. $\text{dist}(\mathbf{a}_{t_1}(\mathbf{q}), \mathcal{B}_i)$). We can see $d_i(\mathbf{q})$ is nonsmooth and non-convex in Figure 3.10(b), which shows a "slice" of $d_i(\mathbf{q})$ through the θ coordinate. ■

We can now present one of the more important results in this chapter. Theorem 3.37 provides a useful way to detect critical points of $D(\mathbf{q})$ based on local single object distance measurements.

Theorem 3.37 $\mathbf{0} \in \partial D(\mathbf{q})$ if and only if $\mathbf{0} \in \text{co}\{\partial d_i(\mathbf{q}) : i \in I(\mathbf{q})\}$

Proof We show that equality holds in Equation 3.13 despite the non-regularity of $d_i(\mathbf{q})$—the theorem immediately follows. Consider the following alternate definition of $D(\mathbf{q})$

$$D(\mathbf{q}) = \min\{\text{dist}(\mathbf{a}_t(\mathbf{q}), \mathcal{B}_i) : \mathbf{a}_t(\mathbf{q}) \in \mathcal{A}, \text{dist}(\mathbf{a}_t(\mathbf{q}), \mathcal{B}_i) = D(\mathbf{q})\}$$

Since $\mathcal{B}_i$ is convex, the point distance function is everywhere differentiable, so $\text{dist}(\mathbf{a}_t(\mathbf{q}), \mathcal{B}_i)$ is everywhere differentiable and $\nabla\text{dist}(\mathbf{a}_t(\mathbf{q}), \mathcal{B}_i)$ exists. We also note

$$\min_i\{f_i\} = -\max_i\{-f_i\}$$

By the smoothness of $f_i(\mathbf{q}) = \text{dist}(\mathbf{a}_t(\mathbf{q}), \mathcal{B}_i)$ and Corollary 3.9, $\partial \min_i\{f_i(\mathbf{q})\} = \partial \max_i\{f_i(\mathbf{q})\}$. Since C is finite dimensional, and $\text{dist}(\mathbf{a}_t(\mathbf{q}), \mathcal{B}_i)$ is differentiable, by application of Theorem 2.8.6 in [16] we have

$$\begin{aligned}\partial D(\mathbf{q}) &= \text{co}\{\nabla\text{dist}(\mathbf{a}_t(\mathbf{q}), \mathcal{B}_i) : \mathbf{a}_t(\mathbf{q}) \in \mathcal{A}, \text{dist}(\mathbf{a}_t(\mathbf{q}), \mathcal{B}_i) = D(\mathbf{q}), i \in I(\mathbf{q})\} \\ &= \text{co}\{\partial d_i(\mathbf{q}) : i \in I(\mathbf{q})\} \text{ by Lemma 3.34}\end{aligned}$$

and the theorem follows. ■

To detect if the current configuration $\mathbf{q}$ is critical, we simply measure the distance $d_i(\mathbf{q})$ to each of the obstacles, determine the n nearest obstacles, determine the individual generalised gradients $\partial d_i(\mathbf{q})$ and take the convex hull. If the convex hull contains $\mathbf{0}$, Theorem 3.21 and Theorem 3.37 tell us $\mathbf{q}$ is critical.

Let's finally return to showing that $K_{D(\mathbf{q})}$ has a finite number of disconnected components. The proof relies on the fact that $\mathbf{0} \notin \partial d_i(\mathbf{q})$, which we hadn't established yet in Section 3.2 and the structure of $\nabla d_i(\mathbf{q})$, when it exists. The proof is again based on the notion of 'downhill'.

Lemma 3.38 The standard directional derivative $d_i'(\mathbf{q}, \mathbf{e}) = 0$ only if $d_i^\circ(\mathbf{q}, -\mathbf{e}) = 0$.

Proof We recall the alternate definition of $d_i(\mathbf{q})$ in Equation 3.12, and recall Lemma 3.34 which established that $\partial d_i(\mathbf{q})$ is equal to the convex hull of the gradient vectors $\nabla \text{dist}(\mathbf{a}_t(\mathbf{q}), \mathcal{B}_i)$. Now, by definition of the pointwise minimum function, if one $\text{dist}(\mathbf{a}_t(\mathbf{q}), \mathcal{B}_i)$ goes 'downhill' in direction $\mathbf{e}$, then $d_i(\mathbf{q})$ also goes 'downhill' ($d_i'(\mathbf{q}, \mathbf{e}) < 0$) in direction $\mathbf{e}$. So, $d_i'(\mathbf{q}, \mathbf{e}) < 0$ whenever there exists a $\zeta \in \partial d_i(\mathbf{q})$ such that $\langle \mathbf{e}, \zeta \rangle < 0$. As well, by the upper-semicontinuity of $d_i^\circ(\mathbf{q}, \mathbf{e})$, $d_i(\mathbf{q})$ also goes 'uphill' ($d_i'(\mathbf{q}, \mathbf{e}) > 0$) in direction $\mathbf{e}$ whenever $d_i^\circ(\mathbf{q}, -\mathbf{e}) < 0$. This only leaves those directions $\mathbf{e}$ satisfying $d_i^\circ(\mathbf{q}, -\mathbf{e}) = 0$ that can possibly have $d_i'(\mathbf{q}, \mathbf{e}) = 0$. ■

Proof of Proposition 3.31: The most straightforward method to show a set $\check{K}_i$ is disconnected is to construct a neighbourhood such that $\text{nbhd}(\check{K}_i) \cap K = \check{K}_i$. We use this approach. The configuration $\mathbf{q}$ is an arbitrary point in $\check{K}_i$. There are two cases which are proved in similar ways:

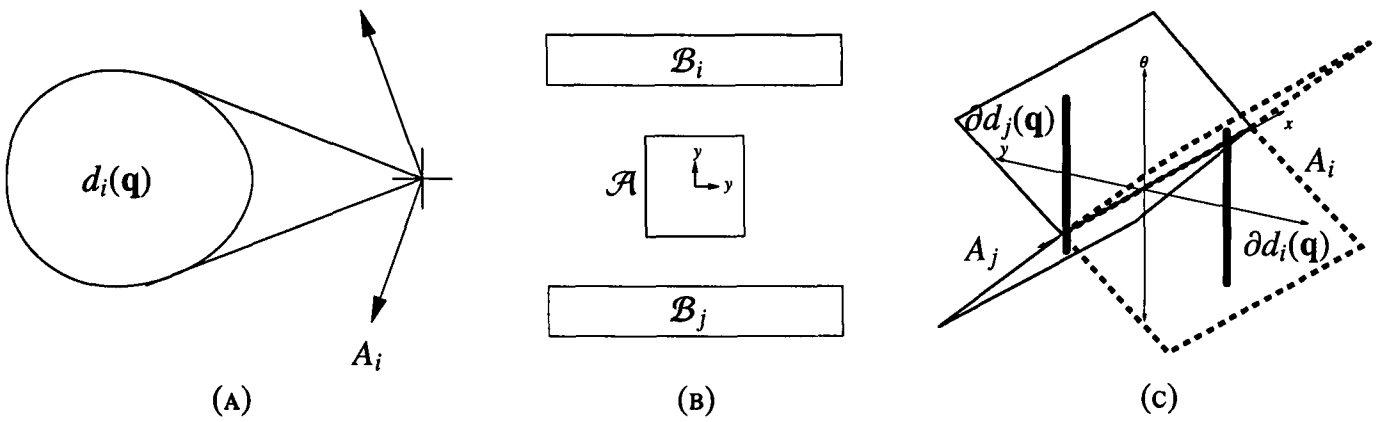
Case 1. $\mathbf{q} \in \text{int}(\text{co}\{\partial d_i(\mathbf{q}) : i \in I(\mathbf{q})\})$

When $\mathbf{q} \in \check{K}_i$ is in the interior of the convex hull, we know that for every direction $\mathbf{e}$, there exists an element $\zeta \in \partial d_i(\mathbf{q})$ for some i that satisfies $\langle \zeta, \mathbf{e} \rangle < 0$. This means that at least one $\text{dist}(\mathbf{a}_t(\mathbf{q}), \mathcal{B}_i)$ goes 'downhill' in direction $\mathbf{e}$ for every $\mathbf{e} \in \mathbb{R}^n$, and does so for a finite distance ϵ by the upper semicontinuity of the generalised directional derivative. By definition of the pointwise minimum function, $D(\mathbf{q})$ must also then go downhill in every direction $\mathbf{e}$ for a finite distance. If this is the case, $\check{K}_i = \mathbf{q}$ is simply a point, and setting $\text{nbhd}(\check{K}_i)$ to the standard ϵ-nbhd of $\mathbf{q}$ will suffice for some small ϵ since $\text{nbhd}(\mathbf{q}) \cap K = \check{K}_i$.

Case 2. $\mathbf{q} \in \text{bndy}(\text{co}\{\partial d_i(\mathbf{q}) : i \in I(\mathbf{q})\})$

For each $d_i(\mathbf{q})$, let the set A_i be defined as the set

$$A_i = \{\mathbf{e} : d_i^\circ(\mathbf{q}; -\mathbf{e}) = 0\} \tag{3.14}$$



(A) (B) (C)

FIGURE 3.11: THE SET $\check{K}_i$ HAS A NEIGHBOURHOOD SATISFYING $\text{nbhd}(\check{K}_i) \cap K = \check{K}_i$. (A) GENERALISED DERIVATIVE $\partial d_i(\mathbf{x})$ ALONG WITH THE SET OF DIRECTION A_i $f_i^\circ(\mathbf{x}; -\mathbf{e}) = 0$. $f_i'(\mathbf{x}; \mathbf{e}) = 0$ ONLY IF $\mathbf{e} \in A_i$. (B) ROBOT $\mathcal{A}$ IN A CRITICAL CONFIGURATION. (C) GENERALISED DERIVATIVES $\partial d_i(\mathbf{q})$ AND $\partial d_j(\mathbf{q})$ (CORRESPONDING TO (B)) ALONG WITH SETS OF DIRECTIONS A_i AND A_j WHERE $d_i^\circ(\mathbf{q}; -\mathbf{e}) = 0$ AND $d_j^\circ(\mathbf{q}; -\mathbf{e}) = 0$.

Lemma 3.38 established that $d_i'(\mathbf{q}; \mathbf{e}) = 0$ only if $\mathbf{e} \in A_i$. This allows us to conclude that the only possible directions $\mathbf{e}$ that $d_i(\mathbf{q})$ can remain at the same value are in A_i. The function $D(\mathbf{q})$ can then only remain at the same value in direction $\mathbf{e}$ only if $\mathbf{e} \in \bigcap_{i \in I(\mathbf{q})} A_i$. Furthermore, the existence of the sin and cos terms in $\nabla d_i(\mathbf{q})$ tell us that $d_i(\mathbf{q})$ will not stay at the same level if the ϕ, θ or ψ terms of $\mathbf{e}$ are non-zero. This means that those directions $\mathbf{e}$ where (i) $\mathbf{e} \notin \bigcap_{i \in I(\mathbf{q})} A_i$ and (ii) $\mathbf{e}_\phi, \mathbf{e}_\theta, \mathbf{e}_\psi \neq 0$ result in $D(\mathbf{q} + \epsilon\mathbf{e}) \neq D(\mathbf{q})$ for all $0 < \epsilon < \delta$, where δ is a small number whose existence is ensured by the upper semicontinuity of $d_i(\mathbf{q})$. Only purely translation motions where $\langle \mathbf{e}, \mathbf{ab}_i \rangle = 0$ for all $i \in I(\mathbf{q})$ can stay in $\check{K}_i$. If motion in a purely translational direction $\mathbf{e}$ satisfying $\langle \mathbf{e}, \mathbf{ab}_i \rangle = 0$ causes one of the $\mathbf{ab}_i$s to no longer satisfy $\langle \mathbf{e}, \mathbf{ab}_i \rangle = 0$, then we know by the upper semicontinuity of $d_i(\mathbf{q})$ that it does so for some small distance. We can thus construct a standard ϵ-neighbourhood of $\mathbf{q}$ that satisfies $\text{nbhd}(\mathbf{q}) \cap K = \text{nbhd}(\mathbf{q}) \cap \check{K}_i$. We can then construct a neighbourhood of the whole critical set component $\text{nbhd}(\check{K}_i) = \bigcup_{\mathbf{q} \in \check{K}_i} \text{nbhd}(\mathbf{q})$ which satisfies $\text{nbhd}(\check{K}_i) \cap K = \check{K}_i$. ■

Figure 3.11(a) shows the set A_i (see Equation 3.14) of possible directions where $d_i(\mathbf{q})$ does not change value. Figure 3.11(b) shows $\mathcal{A}$ in a critical configuration. Figure 3.11(c) shows the generalised gradients $\partial d_i(\mathbf{q})$ and $\partial d_j(\mathbf{q})$ corresponding to Figure 3.11(b) and the two sets of directions A_i and A_j where $d_i^\circ(\mathbf{q}; -\mathbf{e}) = 0$ and $d_j^\circ(\mathbf{q}; -\mathbf{e}) = 0$. Notice the critical set extends only in directions where $A_i \cap A_j \neq \emptyset$ and the motion is purely translational. The neighbourhood $\text{nbhd}(\check{K}_i)$ in this case is an ϵ-neighbourhood of the line segment $\check{K}_i$.

3.3 Summary

In this chapter we developed a detailed nonsmooth analysis of the distance functions $d_(\mathbf{q})$ and $D(\mathbf{q})$. Our approach was to provide information useful to the development of sensor-based motion planning algorithms, but to not dwell on the application of our results—that is left for subsequent chapters.

After our initial introduction of some of the techniques and definitions used in nonsmooth analysis, we proceeded to present a nonsmooth preimage theorem similar in vein to smooth variants used in differential topology. These preimage theorems will be used in subsequent chapters to help define a nonsmooth roadmap used for sensor-based motion planning. A nonsmooth analysis of the distance functions $d_i(\mathbf{q})$ and $D(\mathbf{q})$ was then presented which among other things gave an expression for $\nabla d_i(\mathbf{q})$ when it exists, gave a method to calculate $\partial d_i(\mathbf{q})$ using a finite subset of the closest points P, and most importantly gave a method to calculate $\partial D(\mathbf{q})$ and determine if $\mathbf{q}$ is critical, despite the non-'regularity' of $d_i(\mathbf{q})$. Finally, it was shown that the set of critical points K has a finite number of disjoint components, so a planning algorithm developed based on searching for these components can be guaranteed to terminate in finite time.

Throughout this chapter we have assumed that our distance measurements are exact and without error. In a real robotic system, distance measurements will of course be affected by sensor accuracy, noise, resolution and other real-world issues. However, in order to tackle the problem of dealing with a nonsmooth distance function, we are constrained to use nonsmooth analysis, which does not at the moment accommodate such issues. From a practical standpoint these issues are simply accommodated by the development of a control system robust to these issues, as in Chapter 5.

§4

"The road goes ever on and on down from the door where it began. Now far ahead the road has gone and I must follow if I can. Pursuing it with weary feet until it joins some larger way, where many paths and errands meet -and whither then, I cannot say."

J.R.R. Tolkien

Sensor Based Roadmap for Convex Polygonal Robot in SE(2) Moving Amidst Convex Polygonal Obstacles

Figure 4.1(a) shows a typical problem. A mobile convex robot $\mathcal{A}$ must search for a path through an environment cluttered with some finite number of unknown convex obstacles $\mathcal{B}_i$. $\mathcal{A}$ is equipped with sensors that allow it to measure the minimum distance $d_i(\mathbf{q})$ to each of the $\mathcal{B}_i$s and determine the minimum distance to all obstacles in its environment $D(\mathbf{q})$. In this chapter, we describe a roadmap of C_{free}, which we have termed a *Nonsmooth Critical Set Generalised Voronoi Graph* (NCRIT+NGVG), based on a nonsmooth analysis of the distance function $D(\mathbf{q})$. Figure 4.1(b) illustrates the general idea. The Figure shows a two-dimensional bounded, and thus compact, configuration space C with several configuration space obstacles $C\mathcal{B}_i$. We have shown in Chapter 3 that there are a finite number of *critical sets* $\breve{K}_i$ of the workspace distance function $D(\mathbf{q})$ scattered throughout C_{free}. A system of one-dimensional segments $\mathcal{R}_{\neg K}$ (solid lines) is constructed to connect the critical sets. Notice that $\mathcal{R}_{\neg K}$ does not form a connected roadmap in this example. To establish a fully connected roadmap of C_{free} we construct a second system of one-dimensional segments $\mathcal{R}_K$ (dashed lines) lying within the critical sets. The set $\mathcal{R} = \mathcal{R}_{\neg K} \cup \mathcal{R}_K$ comprises the NCRIT+NGVG.

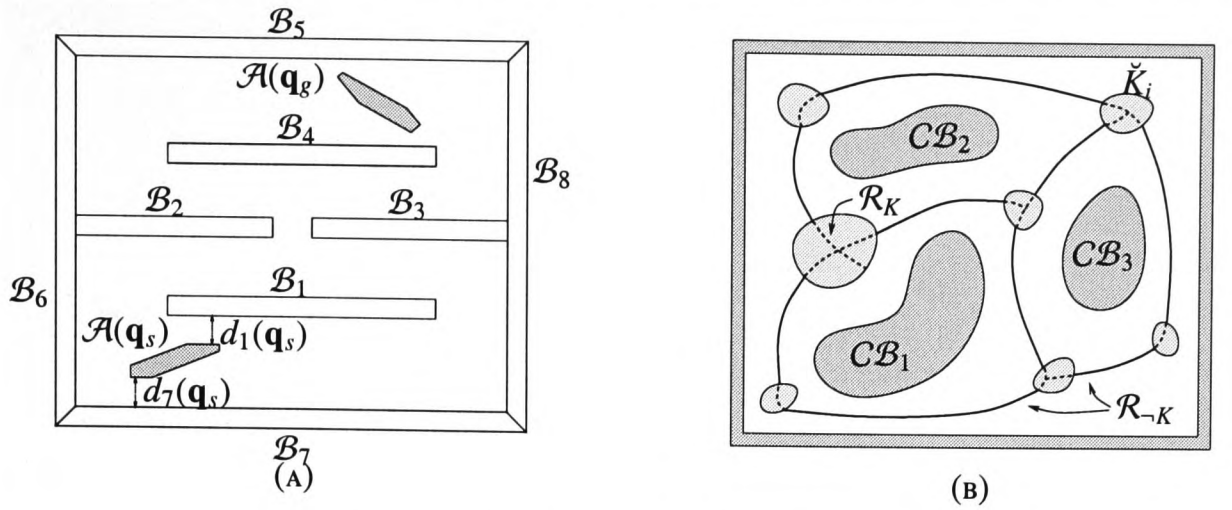


FIGURE 4.1: (A) TYPICAL EXAMPLE IN $SE(2)$. (B) THE NCRIT+NGVG IN A 2D CONFIGURATION SPACE (DOESN'T MATCH (A))

4.1 The $\mathcal{R}_{GVG}$ is Equidistant to Several Obstacles

In the previous chapter, we built up a powerful set of mathematical tools that allow us to begin developing a sensor-based planner that considers the nonsmooth aspects of the single-object distance function $d_i(\mathbf{q})$. In this thesis we will consider only $SE(2)$, and leave $SE(3)$ for further study.

This section outlines our approach to constructing the roadmap $\mathcal{R}$ consisting of regular configurations $\mathcal{R}_{\neg K}$ and critical configurations $\mathcal{R}_K$ as shown in Figure 4.1(b). In line with previous work with distance based roadmaps [12, 34], our approach is to first present a subset of $\mathcal{R}_{\neg K}$ that consists of one-dimensional sets where the robot $\mathcal{A}$ is equidistant to some number of obstacles. We denote this set by $\mathcal{R}_{GVG}$. In Section 4.1.1 we examine the case when $\mathcal{A}$ is equidistant to two obstacles $\mathcal{B}_i$ and $\mathcal{B}_j$ using the concept of the generalised gradient. In Section 4.1.2 we examine the case when $\mathcal{A}$ is equidistant to more than two obstacles using the concept of the weak slope.

4.1.1 Robot Equidistant to Two Obstacles

Like the Generalised Voronoi Graph [12] and EquiDistance Diagram [34], part of our roadmap is constructed from the union of one-dimensional sets in C_{free} where the robot is equidistant to some number of obstacles. However, since the single object distance function $d_i(\mathbf{q})$ is in general nonsmooth, our analysis is drastically different.

In Section 3.2.3 we noted that no nonsmooth (or smooth) critical points of the single object distance function $d_i(\mathbf{q})$ occur in C_{free}. This indicates that all nonsmooth (or smooth) critical points must occur at configurations where the robot is equidistant to at least two obstacles. We start with the definition of a

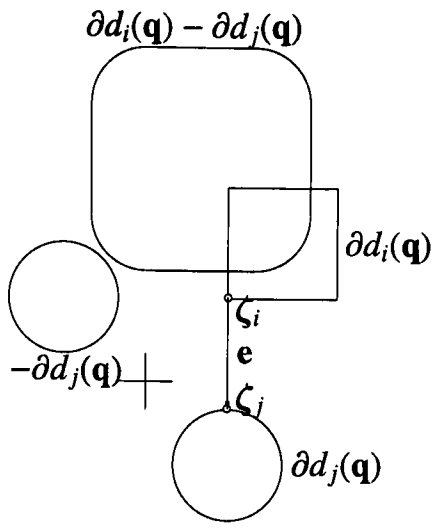


FIGURE 4.2: $(d_i - d_j)(\mathbf{q})$ IS REGULAR IF $\partial d_i(\mathbf{q}) \cap \partial d_j(\mathbf{q}) = \emptyset$. IN THIS EXAMPLE, $\partial d_i(\mathbf{q})$ AND $\partial d_j(\mathbf{q})$ DO NOT INTERSECT, SO WE CAN FIND TWO CLOSEST POINTS ζ_i AND ζ_j ON THE RESPECTIVE SETS. LEMMA 4.1 SHOWS THAT THE SUM $\partial d_i(\mathbf{q}) + (-\partial d_j(\mathbf{q}))$ THEN DOES NOT CONTAIN THE ORIGIN, AND THUS $\mathbf{q}$ IS REGULAR.

two-equidistant set (similar to the two-equidistant surface of [12])

$$\mathcal{S}_{ij} = \{\mathbf{q} \in SE(2) : d_i(\mathbf{q}) = d_j(\mathbf{q}) > 0\} \tag{4.1}$$

which is equivalent to the preimage $(d_i - d_j)^{-1}(\mathbf{0})$ of the function $(d_i - d_j)(\mathbf{q}) : SE(2) \to \mathbb{R}$.

In Section 4.1.2 we will be using the Nonsmooth Preimage Theorem 3.24 to show that the segments of $\mathcal{R}_{GVG}$ are one dimensional. In order the use Theorem 3.24, we need to know when $(d_i - d_j)(\mathbf{q})$ is regular.

Lemma 4.1 $\mathbf{q}$ is a regular point of $(d_i - d_j)(\mathbf{q})$ if $\partial d_i(\mathbf{q}) \cap \partial d_j(\mathbf{q}) = \emptyset$.

Proof Recall that $\mathbf{0} \in \partial(d_i - d_j)(\mathbf{q})$ ($\mathbf{q}$ is critical) if and only if $(d_i - d_j)^\circ(\mathbf{q}; \mathbf{e}) \geq 0$ for all $\mathbf{e}$. So, if $\mathbf{q}$ is regular, there must exist a direction $\mathbf{e}$ such that $(d_i - d_j)^\circ(\mathbf{q}; \mathbf{e}) < 0$ and $(d_i - d_j)^\circ(\mathbf{q}; -\mathbf{e}) > 0$. By definition of generalised directional derivative

$$\begin{aligned} (d_i - d_j)^\circ(\mathbf{q}; \mathbf{e}) &\leq d_i^\circ(\mathbf{q}; \mathbf{e}) + (-d_j)^\circ(\mathbf{q}; \mathbf{e})^1 \\ &= d_i^\circ(\mathbf{q}; \mathbf{e}) + d_j^\circ(\mathbf{q}; -\mathbf{e}) \end{aligned} \tag{4.2}$$

$$\begin{aligned} (d_i - d_j)^\circ(\mathbf{q}; -\mathbf{e}) &= (d_j - d_i)^\circ(\mathbf{q}; \mathbf{e}) \\ &\geq d_j^\circ(\mathbf{q}; \mathbf{e}) + \liminf_{\substack{\mathbf{y} \to \mathbf{q} \\ t \to 0^+}} \frac{-d_i(\mathbf{y} + t\mathbf{e}) + d_i(\mathbf{y})}{t}{}_2 \\ &= d_j^\circ(\mathbf{q}; \mathbf{e}) - d_i^\circ(\mathbf{q}; \mathbf{e})^3 \end{aligned} \tag{4.3}$$

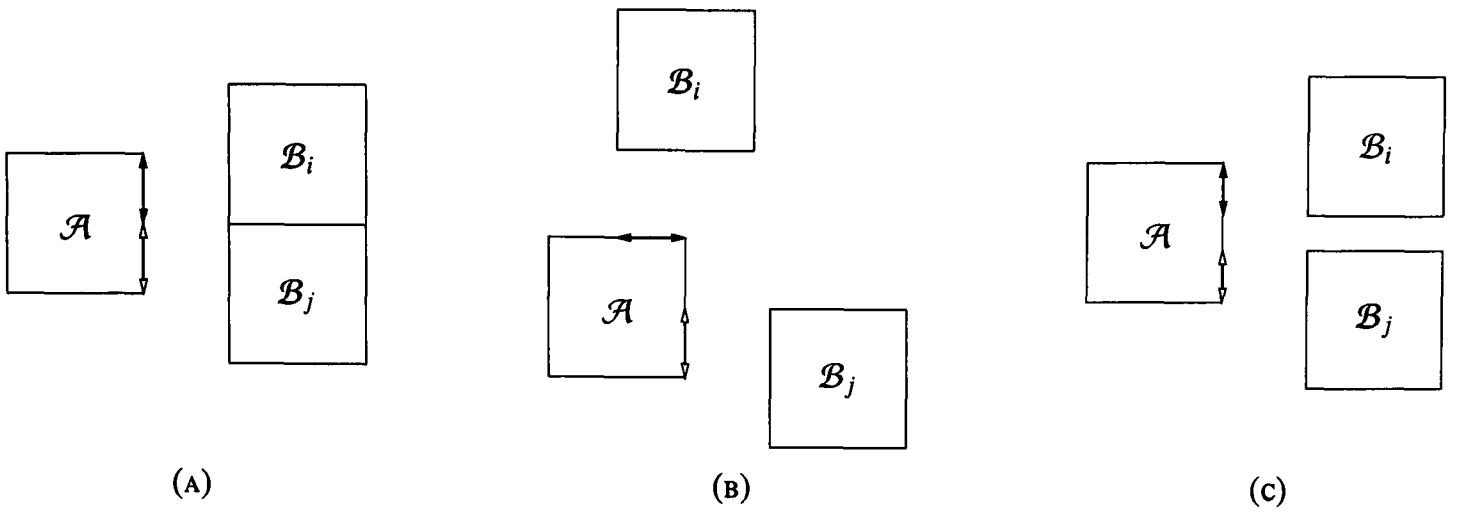


FIGURE 4.3: (A) SHOWS A CONFIGURATION WHERE $\partial d_i(\mathbf{q}) \cap \partial d_j(\mathbf{q}) \neq \emptyset$, SO WE CAN SAY NOTHING ABOUT REGULARITY OF $\mathbf{q}$. THE ALGORITHM IGNORES THESE CONFIGURATIONS. (B) SHOWS A CONFIGURATION WHERE x AND y COORDINATES OF $\partial d_i(\mathbf{q})$ AND $\partial d_j(\mathbf{q})$ DIFFER, SO $\mathbf{q}$ IS REGULAR. (C) SHOWS ANOTHER REGULAR CONFIGURATION WHERE θ COORDINATES OF $\partial d_i(\mathbf{q})$ AND $\partial d_j(\mathbf{q})$ DIFFER.

Since $\partial d_i(\mathbf{q})$ and $\partial d_j(\mathbf{q})$ do not intersect, and are both compact [16], there exist two points $\boldsymbol{\zeta}_i \in \partial d_i(\mathbf{q})$ and $\boldsymbol{\zeta}_j \in \partial d_j(\mathbf{q})$ that attain the minimum distance between $\partial d_i(\mathbf{q})$ and $\partial d_j(\mathbf{q})$. That is

$$\|\boldsymbol{\zeta}_i, \boldsymbol{\zeta}_j\| = \min_{\substack{\mathbf{a} \in \partial d_i(\mathbf{q}) \\ \mathbf{b} \in \partial d_j(\mathbf{q})}} \|\mathbf{a}, \mathbf{b}\|$$

Take the direction $\mathbf{e} = \boldsymbol{\zeta}_j - \boldsymbol{\zeta}_i$, so by the Separation Theorem [1] $\langle \boldsymbol{\zeta}_i, \mathbf{e} \rangle < \langle \boldsymbol{\zeta}_j, \mathbf{e} \rangle$. We also have (by convexity of $\partial d_i(\mathbf{q})$ and $\partial d_j(\mathbf{q})$)

$$\begin{aligned} d_i^\circ(\mathbf{q}; \mathbf{e}) &= \langle \boldsymbol{\zeta}_i, \mathbf{e} \rangle \\ d_j^\circ(\mathbf{q}; \mathbf{e}) &\geq \langle \boldsymbol{\zeta}_j, \mathbf{e} \rangle \\ d_j^\circ(\mathbf{q}; -\mathbf{e}) &= -\langle \boldsymbol{\zeta}_j, \mathbf{e} \rangle \end{aligned}$$

so the right hand side (RHS) of Equation 4.2 is strictly negative and the RHS of Equation 4.3 is strictly positive, so $(d_i - d_j)(\mathbf{q})$ is regular at $\mathbf{q}$. ■

We next introduce the *invertible two-equidistant set* $\mathcal{IS}_{ij}$ which is the subset of $\mathcal{S}_{ij}$ where $(d_i - d_j)(\mathbf{q})$ is regular.

$$\mathcal{IS}_{ij} = \{\mathbf{q} \in \mathcal{S}_{ij} : \partial d_i(\mathbf{q}) \cap \partial d_j(\mathbf{q}) = \emptyset\}$$

---

[1] $\limsup_{y \to x}(f(y) + g(y)) \leq \limsup_{y \to x} f(y) + \limsup_{y \to x} g(y)$

[2] $\limsup_{y \to x} f(y) + \liminf_{y \to x} g(y) \leq \limsup_{y \to x}(f(y) + g(y))$

[3] $\limsup_{y \to x} -f(y) = -\liminf_{y \to x} f(y)$

This is the set of points equidistant to two obstacles $\mathcal{B}_i$ and $\mathcal{B}_j$, such that the generalised gradient $\partial(d_i - d_j)(\mathbf{q})$ is invertible. Theorem 3.24 tells us that the dimension of $\mathcal{IS}_{ij}$ is $\dim(\mathcal{IS}_{ij}) = \dim(C) - 1 = 2$. Consider the following example which highlights the fact that $\dim(\mathcal{IS}_{ij}) = 2$.

Example 4.2 Figure 4.3 shows various cases for a square robot $\mathcal{A}$ and two square obstacles $\mathcal{B}_i$ and $\mathcal{B}_j$. Figure 4.3(a) shows the case where $\partial d_i(\mathbf{q}) \cap \partial d_j(\mathbf{q}) \neq \emptyset$, and we can thus say nothing about criticality/regularity of $\mathbf{q}$. We can see this corresponds to those cases where a non-convex obstacle is decomposed into several convex obstacles. There is no point following roadmap segments equidistant to two such obstacles (if they even exist) since there is surely no gap between them for the robot to move through. Figure 4.3(b) shows a regular configuration since the x and y coordinates of the generalised gradients $\partial d_i(\mathbf{q})$ and $\partial d_j(\mathbf{q})$ differ (θ coordinates are the same). Likewise, Figure 4.3(b) shows a regular configuration where the θ coordinates of the generalised gradients differ (x and y coordinates are the same). The reader should verify in their own mind that $\dim(\mathcal{IS}_{ij}) = 2$ by considering the possible motions of $\mathcal{A}$ in Figures 4.3(b) and (c) such that equidistance is maintained with $\mathcal{B}_i$ and $\mathcal{B}_j$.

4.1.2 Robot Equidistant to Multiple Obstacles

To construct the NCRIT+NGVG, we further define lower dimensional subsets of $\mathcal{S}_{ij}$ and $\mathcal{IS}_{ij}$. We start with the *three-equidistant set*

$$\begin{aligned} \mathcal{S}_{ijk} &= \{\mathbf{q} \in \mathcal{S}_{ij} : d_i(\mathbf{q}) = d_j(\mathbf{q}) = d_k(\mathbf{q})\} \\ &= \mathcal{S}_{ij} \cap \mathcal{S}_{ik} \cap \mathcal{S}_{jk} \end{aligned}$$

which is equivalent to the preimage $(d_i - d_k)^{-1}(\mathbf{0})$ of the function $(d_i - d_k)(\mathbf{q}) : \mathcal{S}_{ij} \rightarrow \mathbb{R}$. To determine the dimension of the roadmap segments, we need to determine at which points $(d_i - d_k)(\mathbf{q})$ is regular. In Section 4.1.1 we determined the regularity of $(d_i - d_j)(\mathbf{q})$ based on the definition of the generalised gradient and the intersection of convex sets $\partial d_i(\mathbf{q})$ and $\partial d_i(\mathbf{q})$. As noted in Chapter 3, the generalised gradient is defined for a domain X that is a *Banach space*, which $\mathcal{S}_{ij}$ certainly is not ($\mathcal{S}_{ij}$ is not a complete linear space at nonsmooth points). In order to establish regularity we need to use the definition of weak slope, which is valid for complete *metric spaces*[4].

Consider the function $\phi : \mathbb{R}^n \rightarrow \mathbb{R}$ defined on a n-dimensional metric space $\mathbb{R}^n$. Take the m-dimensional

[4] Any subspace Y of a metric space X with metric $d : X \times X \rightarrow \mathbb{R}$ is also a metric space with the *induced metric* $d_Y(\mathbf{x}, \mathbf{y}) = d(\mathbf{x}, \mathbf{y})$ for all $\mathbf{x}, \mathbf{y} \in Y$.

subset Y of $\mathbb{R}^n$ ($m \leq n$) and denote the function ϕ *restricted to* Y with $\phi|_Y$ and the function ϕ restricted to the boundary bndy(Y) with $\phi|_{\text{bndy}(Y)}$

Theorem 4.3 If $\mathbf{x}$ is a regular point of $\phi(\mathbf{x})$ then $\mathbf{x} \in \text{bndy}(Y)$ is a regular point of $\phi|_{\text{bndy}(Y)}$ if $\pm\zeta + \nu \neq \mathbf{0}$ for all $\zeta \in \partial\phi(\mathbf{x})$ and $\nu \in N_Y(\mathbf{x}) \setminus \mathbf{0}$ [5].

Proof Assume $\pm\zeta + \nu \neq \mathbf{0}$ for all $\zeta \in \partial\phi(\mathbf{x})$ and $\nu \in N_Y(\mathbf{x}) \setminus \mathbf{0}$. We know that $\mathbf{0} \notin \partial\phi(\mathbf{x})$ since $\mathbf{x}$ is a regular point of $\phi(\mathbf{x})$ so there must exist a direction $\mathbf{e} \in X$ such that $\phi^\circ(\mathbf{x}; \mathbf{e}) < 0$ and so ϕ tends "downhill" in direction $\mathbf{e}$. To show regularity of the point $\mathbf{x} \in Y$ it is sufficient to show the existence of a continuous mapping $\mathcal{H}(\mathbf{v}, t)$ that maps points in $B_\delta(\mathbf{x}) \cap \text{bndy}(Y)$ "downhill" (see Definition 3.14).

Assume first that $N_Y(\mathbf{x})$ contains only one direction, and thus bndy(Y) is locally like a manifold of dimension $k = m$ or a manifold boundary of a $k = m + 1$ dimensional manifold (see Figure 4.4(a)). In either case, a local diffeomorphism (a **chart** in the terminology of [24]) $\Psi : Y \to \mathbb{R}^k$ exists between a neighbourhood of $\mathbf{x}$ and the tangent space $T_\mathbf{x}\text{bndy}(Y)$. Now, there exists a direction $\mathbf{e}$ in $T_\mathbf{x}\text{bndy}(Y)$ such that $\phi^\circ(\mathbf{x}; \mathbf{e}) < 0$, for if there wasn't, we could find elements $\pm\zeta + \nu = \mathbf{0}$, a contradiction. By upper semicontinuity of the generalised directional derivative, $\phi^\circ(\mathbf{y}; \mathbf{e}) < 0$ for all $y \in \text{nbhd}(\mathbf{x})$. We define the continuous transformation $\tilde{\mathcal{H}} : \tilde{B}_\delta(\mathbf{0}) \times [0, \delta] \to T_\mathbf{x}\text{bndy}(Y)$ (maps points $\tilde{\mathbf{v}} \in \tilde{B}_\delta(\mathbf{0}) \subseteq T_\mathbf{x}\text{bndy}(Y)$ to $T_\mathbf{x}\text{bndy}(Y)$)

$$\tilde{\mathcal{H}}(\tilde{\mathbf{v}}, t) = t\mathbf{e}$$

and set $\mathcal{H}(\mathbf{v}, t) = \Psi^{-1} \circ \tilde{\mathcal{H}}(\tilde{\mathbf{v}}, t) \circ \Psi$ in a neighbourhood of $\mathbf{x}$. By the continuity of Ψ, Ψ^{-1} and $\tilde{\mathcal{H}}$, $\mathcal{H}$ is a continuous transform satisfying Equations 3.7.

Now we need to address those points $\mathbf{x} \in Y$ where the normal cone contains more than one direction. We first note that by Rademacher's Theorem[6], the set $\Omega \subseteq \text{bndy}(Y)$ where $N_Y(\mathbf{x})$ contains more than one direction has Lebesque measure zero, and so the previous paragraph applies almost everywhere in bndy(Y). Define a hyperplane H such that $H \cap N_{\text{bndy}(Y)} = H \cap -N_{\text{bndy}(Y)} = \mathbf{0}$ which contains a direction $\mathbf{e}$ such that $\phi^\circ(\mathbf{x}; \mathbf{e}) < 0$. Now by definition of the normal cone, the projection operator $\pi_H : \text{bndy}(Y) \to H$ is a homeomorphism in a neighbourhood of $\mathbf{x}$. Define $\tilde{\mathcal{H}} : \tilde{B}_\delta(\mathbf{0}) \times [0, \delta] \to H$ as in the smooth case and let $\mathcal{H}(\mathbf{v}, t) = \pi_H^{-1} \circ \tilde{\mathcal{H}}(\tilde{\mathbf{v}}, t) \circ \pi_H$ in a neighbourhood of $\mathbf{x}$. By the continuity of π_H, π_H^{-1} and $\tilde{\mathcal{H}}$, $\mathcal{H}$ is a continuous transform satisfying Equations 3.7. ■

Consider the following example.

[5] The element $\mathbf{0} \in N_Y(\mathbf{x})$ tells us nothing about the local structure of bndy(Y)

[6] This is not actually a direct consequence of Rademacher's Theorem, but rather uses the relationship between the normal cone and the distance function, to which Rademacher's Theorem applies. See [16] for details.

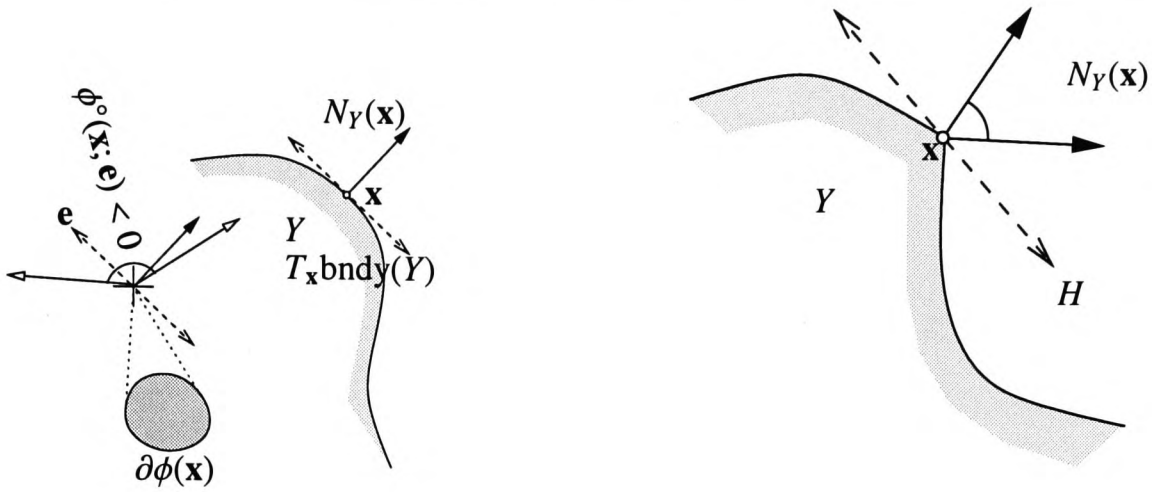


FIGURE 4.4: $\mathbf{x} \in \text{bndy}(Y)$ IS A REGULAR POINT OF $\phi|_{\text{bndy}(Y)}$ IF $\pm\zeta + \nu \neq \mathbf{0}$ FOR ALL $\zeta \in \partial\phi(\mathbf{x})$ AND $\nu \in N_Y(\mathbf{x})\backslash\mathbf{0}$. (A) CASE WHERE $N_{\text{bndy}(Y)}$ CONTAINS ONLY ONE DIRECTION, THE THUS $\text{bndy}(Y)$ IS A SMOOTH MANIFOLD BOUNDARY. WITH THE $\partial\phi(\mathbf{x})$ GIVEN, IT IS EASY TO SEE THAT POINTS MAP "DOWNHILL" IN A DIRECTION THAT IS A MEMBER OF THE TANGENT SPACE. (B) CASE WHERE $\text{bndy}(Y)$ IS NONSMOOTH AND $N_{\text{bndy}(Y)}$ CONTAINS A RANGE OF DIRECTIONS.

Example 4.4 Figure 4.5(a) shows an example for the function $\phi(\mathbf{x}) = 0.5\sqrt{x^2 + y^2} - x_1$ given by a cone rotated about the y-axis restricted to the boundary of the set $Y = \{(x, y); y - |x| \geq 0\}$. In this case, $N_Y(\mathbf{0})\backslash\mathbf{0}$ (arrowheads) and $\partial\phi(\mathbf{0})$ (shaded oval) have elements that sum to $\mathbf{0}$ (are linearly dependent). By Theorem 4.3 we cannot say if $\mathbf{x} = \mathbf{0}$ is regular or critical. In this case it turns out to be a critical point, since $\phi|_{\text{bndy}(Y)}(\mathbf{x})$ (solid line on the cone) attains a global minimum at $\mathbf{x} = \mathbf{0}$. Figure 4.5(b) shows an example for the function $\phi(\mathbf{x}) = 0.5\sqrt{x^2 + y^2} - y$ given by the same cone rotated about the x-axis. Since $N_Y(\mathbf{0})\backslash\mathbf{0}$ and $\pm\partial\phi(\mathbf{0})$ have no elements that sum to $\mathbf{0}$ (are linearly independent), $\mathbf{x} = \mathbf{0}$ is a regular point of $\phi|_{\text{bndy}(Y)}(\mathbf{x})$, and hence is not a local minimum/maximum. □

In Theorem 4.3 it is important to note the use of the normal cone $N_Y(\mathbf{x})$, rather than $N_{\text{bndy}(Y)}(\mathbf{x})$, when determining if points in $\phi|_{\text{bndy}(Y)}$ are regular. The normal cone $N_{\text{bndy}(Y)}(\mathbf{x})$ in Figure 4.4(b) is equal to $\mathbb{R}^n$, which is less than informative about the local structure of $\text{bndy}(Y)$. It turns out that there is no elegant if and only if proof regarding the criticality/regularity of $d_i(\mathbf{q})$ at points $\mathbf{q} \in C$. This is due to the lack of "regularity" (in Clarke's sense) of the single object distance functions $d_i(\mathbf{q})$. In the following, $Y = \{\mathbf{x} : \phi^{-1}(\mathbf{x}) \leq c\} \cap B_\delta(\mathbf{x})$ for some $c \in \mathbb{R}$ and some regular neighbourhood $B_\delta(\mathbf{x})$. In order to use Theorem 4.3, we need the following Lemma which relates the normal cone and the generalised gradient for a function $\phi(\mathbf{x})$ at a regular point $\mathbf{x}$[7].

Lemma 4.5 $N_Y(\mathbf{x}) \subseteq \{\lambda\partial\phi(\mathbf{x}) : \lambda \geq 0\}$

[7]Although this appears to be a classical result based on implicit references to it in [16], the proof is the authors own interpretation.

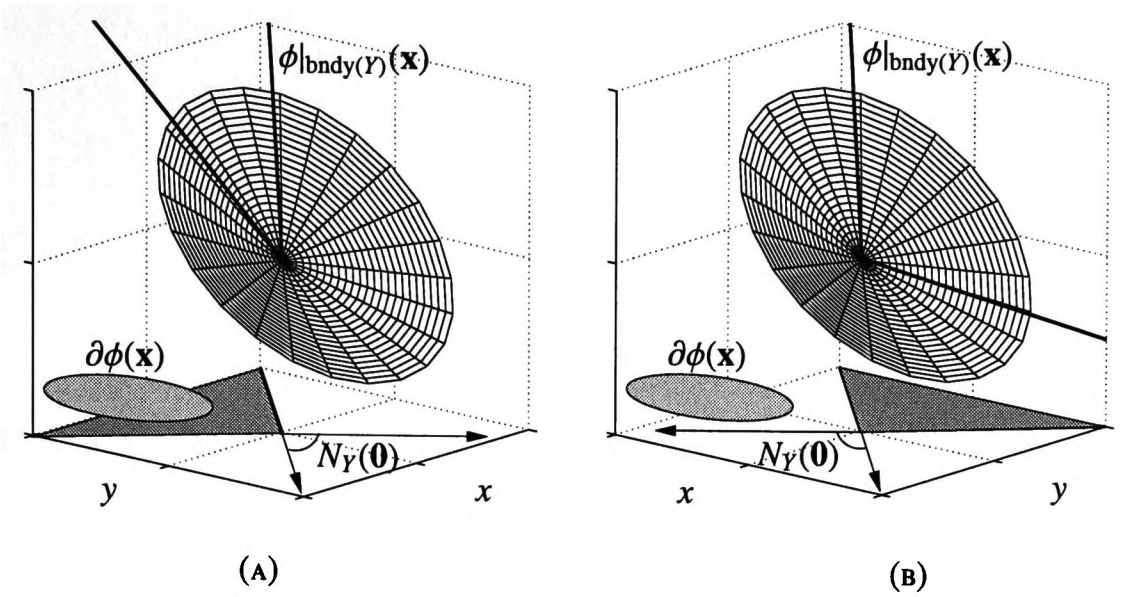


FIGURE 4.5: DETERMINING IF A POINT $\mathbf{x}$ IS REGULAR. (A) GRAPH OF $\phi(\mathbf{x}) = 0.5\sqrt{x^2+y^2} - x$ RESTRICTED TO THE BOUNDARY OF THE SET $Y = \{(x,y); y - |x| \geq 0\}$. THE NORMAL CONE $N_Y(\mathbf{0})$ (ARROWHEADS) AND $\partial\phi(\mathbf{0})$ (SHADED OVAL) HAVE PROPORTIONAL POINTS, SO WE CANNOT SAY IF $\mathbf{x}$ IS REGULAR (IT HAPPENS TO BE CRITICAL). (B) GRAPH OF $\phi(\mathbf{x}) = 0.5\sqrt{x^2+y^2} - y$ IN WITH $N_Y(\mathbf{0})$ AND $\partial\phi(\mathbf{0})$ HAVE NO PROPORTIONAL POINTS AND $\mathbf{x}$ IS THUS A REGULAR POINT.

Proof Recall that the generalised gradient $\partial\phi(\mathbf{x})$ is related to the normal cone of the epigraph $N_{\text{epi}\,\phi}(\mathbf{x}, \phi(\mathbf{x}))$ by Equation 3.2. Let $\hat{N}_{\text{epi}\,\phi}(\mathbf{x}, \phi(\mathbf{x}))$ be the set constructed as per Equation 3.1 using only sequences $\{\mathbf{x}_i, \phi(\mathbf{x}_i)\}$ such that $\mathbf{x}_i \notin Y$. We then have $\hat{N}_{\text{epi}\,\phi}(\mathbf{x}, \phi(\mathbf{x})) \subseteq N_{\text{epi}\,\phi}(\mathbf{x}, \phi(\mathbf{x}))$. Letting $\pi_{\mathbb{R}^n}$ be the projection operator onto $\mathbb{R}^n$ (recalling $N_{\text{epi}\,\phi} \subseteq \mathbb{R}^n \times \mathbb{R}$) we have $\pi_{\mathbb{R}^n}\hat{N}_{\text{epi}\,\phi}(\mathbf{x}, \phi(\mathbf{x})) \subseteq \pi_{\mathbb{R}^n}N_{\text{epi}\,\phi}(\mathbf{x}, \phi(\mathbf{x}))$. As well, since $\pi_{\mathbb{R}^n}\{\mathbf{x}_i, \phi(\mathbf{x}_i)\} = \{\mathbf{x}_i\}$ we have $N_Y(\mathbf{x}) = \pi_{\mathbb{R}^n}\hat{N}_{\text{epi}\,\phi}(\mathbf{x}, \phi(\mathbf{x}))$. Finally, by direct consequence of Equation 3.2 we have $\pi_{\mathbb{R}^n}N_{\text{epi}\,\phi}(\mathbf{x}, \phi(\mathbf{x})) = \{\lambda\partial\phi(\mathbf{x})) \,:\, \lambda \geq 0\}$, from which the lemma follows. ■

Lemma 4.5 does not apply if $\mathbf{x}$ is critical. To see this, consider the functions $\phi_1(x,y) = 0$ and $\phi_2(x,y) = x^2+y^2$. Both functions have $(0,0)$ as a critical point, but have level set normal cones $\{\varnothing\}$ and $\mathbb{R}^2$ respectively.

We can now use Theorem 4.3 and Lemma 4.5 to determine when $(d_i - d_k)(\mathbf{q})$ is regular. Set $\phi(\mathbf{x}) = (d_i - d_j)(\mathbf{q})$, $c = d_i(\mathbf{q})$ and $Y = \{\mathbf{q} \,:\, (d_i - d_j)(\mathbf{q}) \leq c\}$. Applying Theorem 4.3, we then have a subset of the three-equidistant set that is not critical is termed the *invertible three-equidistant set*

$$\mathcal{IS}_{ijk} = \{\mathbf{q} \in \mathcal{IS}_{ij} : \pm\boldsymbol{\zeta} + \boldsymbol{\nu} \neq \mathbf{0} \text{ for all } \boldsymbol{\zeta} \in \partial(d_i - d_k)(\mathbf{q}), \\ \boldsymbol{\nu} \in \{\lambda\partial(d_i - d_j)(\mathbf{q}) \,:\, \lambda > 0\}\}$$

which is those points equidistant to $\mathcal{B}_i$, $\mathcal{B}_j$ and $\mathcal{B}_k$ having $\partial(d_i - d_j)(\mathbf{q})$ and $\partial(d_i - d_k)(\mathbf{q})$ with non-zero linearly independent elements. The set of points closest to $\mathcal{B}_i$, $\mathcal{B}_j$ and $\mathcal{B}_k$ is the *invertible three-equidistant*

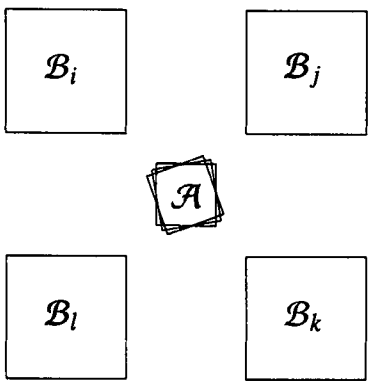


FIGURE 4.6: CASE WHEN $\dim(\mathcal{IF}_{ijkl}) \neq 0$. IN THIS CASE, THE SET $\mathcal{IF}_{ijkl}$ IS ONE DIMENSIONAL, SINCE $d_i(\mathbf{q}) = d_j(\mathbf{q}) = d_k(\mathbf{q}) = d_k(\mathbf{q})$ FOR A RANGE OF ORIENTATIONS.

face

$$\mathcal{IF}_{ijk} = \{\mathbf{q} \in \mathrm{cl}(\mathcal{IS}_{ijk}) : d_i(\mathbf{q}) \leq d_l(\mathbf{q}) \ \forall\, l \neq i, j, k\}$$

By Corollary 3.24, $\dim(\mathcal{IS}_{ijk}) = 1$, so $\mathcal{IF}_{ijk}$ are edges of NCRIT+NGVG.

In the smooth case [14], zero-dimensional subsets of $\mathcal{IS}_{ijk}$ equidistant to $\mathcal{B}_i$, $\mathcal{B}_j$, $\mathcal{B}_k$ and $\mathcal{B}_l$ are defined when $\nabla(d_i - d_l)(\mathbf{q})$ is not a member of the subspace spanned by $\nabla(d_i - d_j)(\mathbf{q})$ and $\nabla(d_i - d_k)(\mathbf{q})$. Analogues to transversality etc. exist in nonsmooth analysis, but to avoid choking this thesis with definitions, we simply define the *invertible four-equidistant face* $\mathcal{IF}_{ijkl}$ as the intersection of the appropriate $\mathcal{IF}_{ijk}$s

$$\mathcal{IF}_{ijkl} = \mathcal{IF}_{ijk} \cap \mathcal{IF}_{ijl} \cap \mathcal{IF}_{jkl}$$

and assume $\dim(\mathcal{IF}_{ijkl}) = 0$ so that $\mathcal{IF}_{ijkl}$ are vertices of NCRIT+NGVG. This assumption is equivalent to the "no four points are co-circular" assumption used in the Voronoi diagram literature [12]. A set of four obstacles that violate this assumption are shown in Figure 4.6 in which $\mathcal{IF}_{ijk}$, $\mathcal{IF}_{ijl}$, $\mathcal{IF}_{jkl}$ intersect in such a way that $\dim(\mathcal{IF}_{ijkl}) = 1$. In practise it is quite easy to modify a search algorithm to recognise this case and follow only the appropriate invertible three-equidistant face $\mathcal{IF}_{ijk}$.

We can now fully define the $\mathcal{R}_{GVG}$—a very critical component of our NCRIT+NGVG roadmap.

$$\mathcal{R}_{GVG} = \left(\cup_{i,j,k} \mathcal{IF}_{ijk}\right) \cup \left(\cup_{i,j,k,l} \mathcal{IF}_{ijkl}\right)$$

$\mathcal{R}_{GVG}$ is the set of one-dimensional edges equidistant to three $\mathcal{B}_i$s and the zero-dimensional vertices equidistant to four $\mathcal{B}_i$s (see Figure 4.9(a)). In the next section we address how $\mathcal{A}$ initially moves onto the $\mathcal{R}_{GVG}$ if it is not equidistant to several obstacles to begin with.

4.1.3 An Accessibility Retraction onto $\mathcal{R}_{GVG}$

In this section we show it is possible to access $\mathcal{R}_{GVG}$ from any $\mathbf{q} \in C_{free}$, using a retraction-like transformation. Retractions are useful in that they preserve the connectivity of their domain [38], allowing planning to be performed in the image of the retraction.

Lemma 4.6 For all configurations $\mathbf{q}$ in C_{free}, there exists a continuous path $\tau(t)$ onto the graph/set of equidistant configurations $\mathcal{R}_{GVG}$.

Proof Assume the robot $\mathcal{A}$'s orientation θ remains constant so that all $d_i(\mathbf{q})$s can be considered smooth with configuration space $C = \mathbb{R}^2$. Firstly, from start configuration $\mathbf{q}_i$, find the closest obstacle $\mathcal{B}_i$ (assume without loss of generality that $d_i(\mathbf{q}) < d_j(\mathbf{q})$ for all $j \neq i$) and move away in direction $\nabla d_i(\mathbf{q}) = \mathbf{a}_i - \mathbf{b}_i$, until equidistant to two obstacles $\mathcal{B}_i$ and $\mathcal{B}_j$ at a configuration denoted by $\mathbf{q}_{ij}$. We are assured that an obstacle $\mathcal{B}_j$ will be found since $\nabla d_i(\mathbf{q}) \neq \mathbf{0}$ for all $\mathbf{q} \in C$ as $d_i(\mathbf{q})$ has no critical points ($\mathcal{A}$ will not get stuck in a local minimum of $d_i(\mathbf{q})$ before finding $\mathcal{B}_j$). Secondly, find the next closest obstacle $\mathcal{B}_k$ and move toward it while maintaining equidistance with obstacles $\mathcal{B}_i$ and $\mathcal{B}_j$, until equidistant to all three obstacles $\mathcal{B}_i$, $\mathcal{B}_j$ and $\mathcal{B}_k$ at a final configuration denoted $\mathbf{q}_{ijk}$. This is done by projecting $\nabla d_i(\mathbf{q}) - \nabla d_k(\mathbf{q})$ onto the tangent space of $\mathcal{IS}_{ij}$ (move in the direction $\pi_{T_\mathbf{q}\mathcal{IS}_{ij}}(\nabla d_i(\mathbf{q}) - \nabla d_k(\mathbf{q}))$), which exists because the single object distance functions are smooth when rotation is restricted.

If upon following the second step any of the following occurs

1. The robot becomes "jammed"—$d_i(\mathbf{q}) = 0$
2. $\mathcal{A}$ arrives at a local minimum while moving toward $\mathcal{B}_k$—$\pi_{T_\mathbf{q}\mathcal{IS}_{ij}}(\nabla d_i(\mathbf{q}) - \nabla d_k(\mathbf{q})) = \mathbf{0}$
3. We find some $\mathcal{B}_n$ closer than $\mathcal{B}_k$—$d_i(\mathbf{q}) = d_j(\mathbf{q}) = d_n(\mathbf{q}) < d_k(\mathbf{q})$
4. The obstacle $\mathcal{B}_k$ is no longer visible

then return to $\mathbf{q}_{ij}$ and find the next closest $\mathcal{B}_n \neq \mathcal{B}_k$ and repeat. In $SE(2)$, a bounded environment contains enough $\mathcal{B}_i$s that this process terminates. $\tau(t)$ is the path generated by the above process. ■

Our definition of accessibility differs from that in [14] which incorrectly assumes $\mathcal{A}$ can access a *specific* segment $\mathcal{IF}_{ijk}$ when $\mathcal{A}(\mathbf{q}_i)$ is initially closest to $\mathcal{B}_i$, $\mathcal{B}_j$ and $\mathcal{B}_k$. In Figure 4.7(a), we show a case where $\mathcal{A}$ became "jammed" whilst moving toward $\mathcal{B}_k$. The robot then moved back to $\mathbf{q}_{ij}$ and moved toward $\mathcal{B}_n$, eventually terminating at $\mathbf{q}_{ijn} \in \mathcal{IF}_{ijn}$. The path described in Lemma 4.6 will form the basis of the required retraction H defined on an *accessibility region*.

Definition 4.7 An **accessibility region** $\mathcal{A}_{ijk}$ is the set of initial configurations $\mathbf{q}_i$ that result in a final

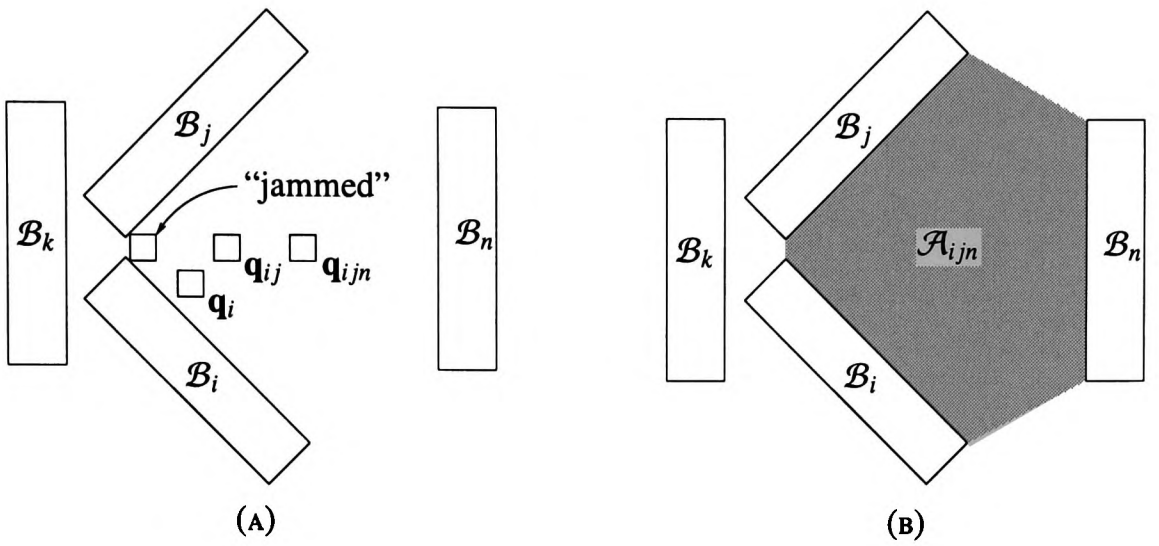


FIGURE 4.7: (A) ACCESSING THE $\mathcal{R}_{GVG}$. THE ROBOT $\mathcal{A}$ BECAME "JAMMED" WHEN MOVING TOWARD $\mathcal{B}_k$ AND SO MOVED TOWARD $\mathcal{B}_n$ INSTEAD. (B) ACCESSIBILITY REGION $\mathcal{A}_{ijn}$ FOR THE THREE OBSTACLES $\mathcal{B}_i$, $\mathcal{B}_j$ AND $\mathcal{B}_n$.

configuration $\mathbf{q}_{ijk}$ in the invertible three-equidistant face $\mathcal{IF}_{ijk}$ when following the path $\tau(t)$ described in Lemma 4.6.

$$\mathcal{A}_{ijk} = \{\mathbf{q}_i \in C_{free} \,:\, \mathbf{q}_{ijk} \in \mathcal{IF}_{ijk}\}$$ ■

Figure 4.7(b) shows accessibility region $\mathcal{A}_{ijn}$. The union of all the $\mathcal{A}_{ijk}$s is equal to freespace $\cup_{i=1}^{i=n-2} \cup_{j=i+i}^{j=n-1} \cup_{k=j+1}^{k=n} \mathcal{A}_{ijk} = C_{free}$. Lemma 4.8 presents the desired continuous retraction $H : \mathcal{A}_{ijk} \times [0,1] \rightarrow \mathcal{IF}_{ijk}$

Lemma 4.8 There exists a continuous map $H : \mathcal{A}_{ijk} \times [0,1] \rightarrow \mathcal{IF}_{ijk}$ describing path from an initial configuration $\mathbf{q}_i$ in the accessibility region $\mathcal{A}_{ijk}$ to a final configuration $\mathbf{q}_{ijk}$ in the invertible three-equidistant face $\mathcal{IF}_{ijk}$.

Proof It is known[8] that the transformation $H(\mathbf{q}, t)$ generated by gradient descent is continuous. For this reason, the transformation $H_1 : \mathcal{A}_{ijk} \times [0,1] \rightarrow \mathcal{IS}_{ij}$ that takes $\mathbf{q}_i \in \mathcal{A}_{ijk}$ $(t = 0)$ to $\mathbf{q}_{ij} \in \mathcal{IS}_{ij}$ $(t = 1)$ is continuous. Likewise, by the continuity of the projection operator $\pi_{T_{\mathbf{q}}\mathcal{IS}_{ij}}$, the transformation $H_2 : \mathcal{IS}_{ij} \times [0,1] \rightarrow \mathcal{IF}_{ij}$ that takes $\mathbf{q}_{ij} \in \mathcal{IS}_{ij}$ $(t = 0)$ to $\mathbf{q}_{ijk} \in \mathcal{IF}_{ijk}$ $(t = 1)$ is also continuous. The composite transformation $H = H_2 \circ H_1$ is the continuous transform that satisfies the lemma. ■

It is important to note that H is not a global retraction, but still satisfies the "retraction-like" properties in [38]. The fact that a retraction exists of the accessibility region $\mathcal{A}_{ijk}$ onto the invertible three-equidistant face $\mathcal{IF}_{ijk}$ allows us to reduce planning within $\mathcal{A}_{ijk}$ to planning within $\mathcal{IF}_{ijk}$.

[8] Define a smooth function $f : X \rightarrow \mathbb{R}$ that takes points $\mathbf{x}$ in a manifold X to the reals. The gradient $\nabla f(\mathbf{x})$ points in the direction of most rapid increase of $f(\mathbf{x})$, when it is nonzero. We denote by $\phi_t(\mathbf{x})$ the maximal flow generated by $-\nabla f(\mathbf{x})$ (what we follow in Lemma 4.6) and note that $\frac{d}{dt}\phi_t(\mathbf{x}) = -\nabla f(\phi_t(\mathbf{x}))$. It is a well known fact that the one parameter flow generated by gradient descent is continuous.

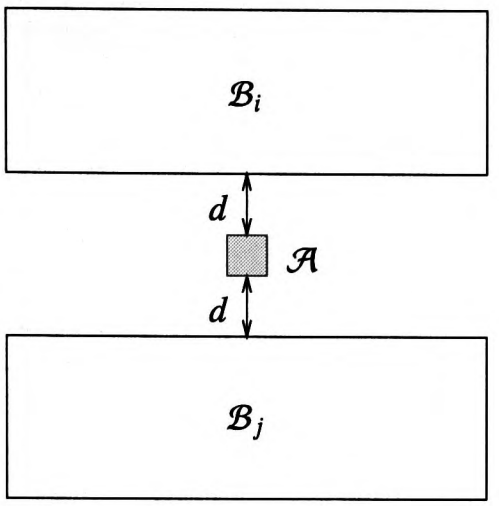


FIGURE 4.8: A ROBOT $\mathcal{A}$ IN A CRITICAL CONFIGURATION $\mathbf{q} = [x_c, y_c, \theta_c]^T$. IT IS EASY TO SEE THE ROBOT CAN ONLY REMAIN IN A CRITICAL CONFIGURATION *and* REMAIN DISTANCE $d_j(\mathbf{q}) = d$ FROM THE OBSTACLES $\mathcal{B}_i$ AND $\mathcal{B}_j$ IF IT MOVES ALONG THE ONE-DIMENSIONAL SEGMENT GIVEN BY $y = y_c$, $\theta = \theta_c$. THE CRITICAL SET COMPONENT $\check{K}_i$ HAS DIMENSION LESS THAN OR EQUAL TO ONE.

Theorem 4.9 Two configurations $\mathbf{q}_1$ and $\mathbf{q}_2$ in an accessibility region $\mathcal{A}_{ijk}$ are path connected in the accessibility region $\mathcal{A}_{ijk}$ if and only if $H(\mathbf{q}_1, 1)$ and $H(\mathbf{q}_2, 1)$ are path connected in the invertible three-equidistant face $\mathcal{IF}_{ijk}$.

Proof **(IF)** By Lemma 4.8, $\mathbf{q}_1$ and $H(\mathbf{q}_1, 1)$ are path connected in $\mathcal{A}_{ijk}$ and $\mathbf{q}_2$ and $H(\mathbf{q}_2, 1)$ are also path connected in $\mathcal{A}_{ijk}$. If as well $H(\mathbf{q}_1, 1)$ and $H(\mathbf{q}_2, 1)$ are path connected in the $\mathcal{IF}_{ijk} \subseteq \mathcal{A}_{ijk}$, then certainly $\mathbf{q}_1$ and $\mathbf{q}_2$ are path connected in $\mathcal{A}_{ijk}$.

(ONLY IF) Let $\tau(t)$ describe a continuous (connected) path lying entirely in $\mathcal{A}_{ijk}$ with $\tau(0) = \mathbf{q}_1$ and $\tau(1) = \mathbf{q}_2$. We know the transformation of this path $H(\tau(t), 1)$ lies in $\mathcal{IF}_{ijk}$. The image $H(\tau(t), 1)$ is also connected since the image of a connected set ($\tau(t)$) under a continuous transformation ($H(\cdot, t)$) is also a connected set, so $H(\mathbf{q}_1, 1)$ and $H(\mathbf{q}_2, 1)$ are certainly path connected in $\mathcal{IF}_{ijk}$. ■

Theorem 4.9 tells us we can plan within $\mathcal{A}_{ijk}$ using only $\mathcal{IF}_{ijk}$, but we still require proof that paths exists between adjacent accessibility regions. The *critical set graph* $\mathcal{R}_K$ connects some cases.

4.2 Critical Set Graph $\mathcal{R}_K$

We now define the graph $\mathcal{R}_K$ consisting of critical configurations. First of all, we know the set of critical configurations K is a subset of $\mathcal{IS}_{ij}$ since the robot $\mathcal{A}$ must be equidistant to at least two obstacles $\mathcal{B}_i$ and $\mathcal{B}_j$ to be critical. As well, in addition to the condition $d_i(\mathbf{q}) - d_j(\mathbf{q}) = 0$ for all $\mathbf{q} \in \mathcal{IS}_{ij}$, we have the

necessary condition that $d_j(\mathbf{q}) = d$ for all $\mathbf{q} \in K_d$. Define the function $f_j : \mathcal{IS}_{ij} \rightarrow \mathbb{R}$ as

$$f_j(\mathbf{q}) = d_j(\mathbf{q}) - d$$

Level sets of $f_j(\mathbf{q})$ represent motion that doesn't change $d_j(\mathbf{q})$. Since $d_j(\mathbf{q}) = d$ is only a necessary condition for $\mathbf{q} \in K$, we also know that $K_c \subseteq f_j^{-1}(\mathbf{0})$. We bound the dimension of K_d by using the Nonsmooth Preimage Theorem 4.3 to determine the dimension of the preimage $f_j^{-1}(\mathbf{0})$ of which K_d is a subset.

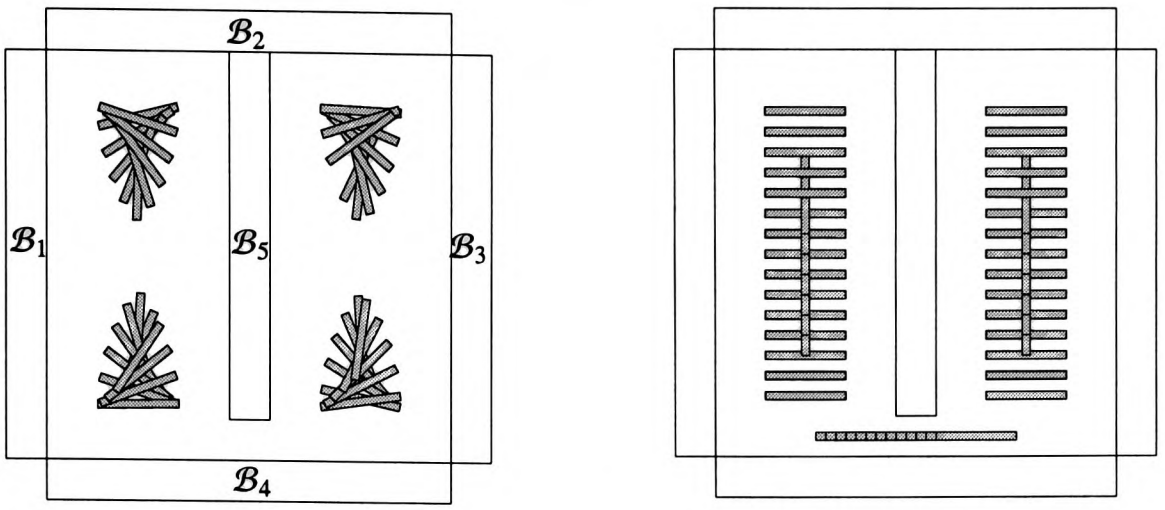
Lemma 4.10 The dimension of the critical set K is less than or equal to one ($\dim(K) \leq 1$).

Proof Recall that the dimension of the invertible two-equidistant set $\dim(\mathcal{IS}_{ij}) = 2$. By the calculus of the generalised gradient $\partial f_j(\mathbf{q}) = \partial d_j(\mathbf{q})$, so $\pm\partial f_j(\mathbf{q}) \cap N_{\mathcal{IS}_{ij}}(\mathbf{q}) \setminus \mathbf{0} = \varnothing$. Applying Theorem 4.3, we see that $\mathbf{q}$ is regular and so by Theorem 3.24, we know the dimension of the preimage $\dim(f_j^{-1}(\mathbf{0})) = 1$. Since $d_j(\mathbf{q}) = c$ is only a necessary condition for $\mathbf{q} \in K$, we know that $K_c \subseteq f_j^{-1}(\mathbf{0})$, and thus the dimension of the critical set $\dim(K) \leq 1$ as well. ■

In $SE(2)$ we have the unique result that $\mathcal{R}_K = K$. Since the dimension of the critical set $\dim(K) \leq 1$, we know the dimension of any disconnected component $\check{K}_i$ of K_d (and K) is also $\dim(\check{K}_i) \leq 1$. Figure 4.8 shows an example of a critical configuration and shows that the critical set component $\check{K}_i$ is a one dimensional line segment.

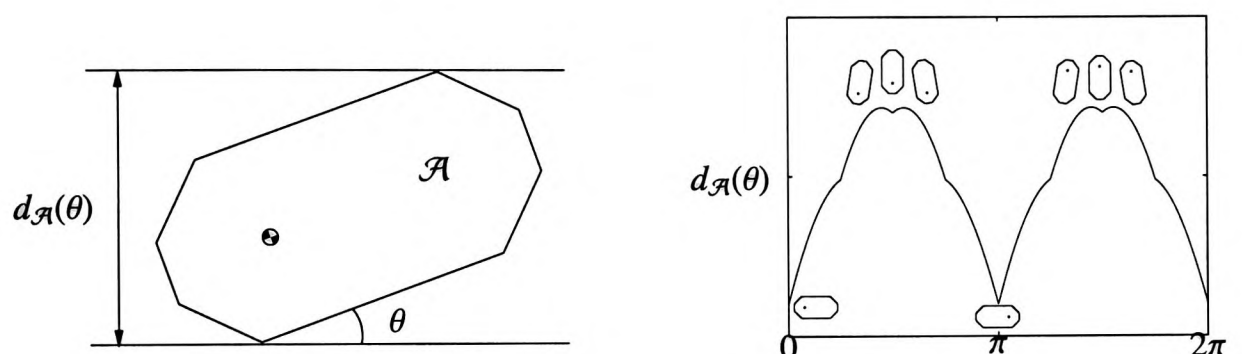
4.3 Connecting it all Together

To this point, we have fully defined the subset $\mathcal{R}_K \subseteq \mathcal{R}$ consisting of critical configurations ($\mathcal{R}_K = K$), and the subset $\mathcal{R}_{GVG} \subseteq \mathcal{R}$ consisting of configurations where the robot is equidistant to some number of obstacles. It turns out these two sets do not define a fully connected roadmap. Figure 4.9(a) shows the roadmap segments corresponding to several invertible three-equidistant sets $\mathcal{IS}_{ijk}$. In other equidistance based roadmaps, the GVG in particular, three-equidistant sets (roadmap segments) meet at four-equidistant sets (roadmap vertices). In this case however, there is no configuration where the robot is equidistant to four obstacles while at the same time having those four obstacles within line-of-sight or not being in collision with an obstacle. Figure 4.9(b) shows a few configurations sampled from the five disjoint components $\check{K}_i$ of the critical set K. In this case all $\check{K}_i$s have dimension one. Since $\mathcal{R}_{GVG}$ and $\mathcal{R}_K$ are not necessarily connected, there are still elements of $\mathcal{R}_{\neg K}$ that need to be defined. In this section we present our approach.



(A)

FIGURE 4.9: PORTIONS OF THE ROADMAP $\mathcal{R}$. (A) ROADMAP SEGMENTS CORRESPONDING TO THREE-EQUIDISTANT SETS $\mathcal{R}_{GVG}$. (B) CRITICAL SET K WHICH CORRESPONDS $\mathcal{R}_K$. $\mathcal{R}_{GVG}$ AND $\mathcal{R}_K$ DO NOT COMPRISE A CONNECTED ROADMAP OF FREESPACE C_{free}.



(A) (B)

FIGURE 4.10: (A) $d_{\mathcal{A}}(\theta)$ IS GIVEN BY THE DISTANCE BETWEEN TWO PARALLEL SUPPORT LINES OF THE POLYGON $\mathcal{A}$ IN ORIENTATION θ. (B) $d_{\mathcal{A}}(\theta)$ HAS A FINITE NUMBER OF LOCAL MINIMA AND MAXIMA.

Let us first show that $\mathcal{R}_{GVG}$ is sufficient to connect critical sets K_{ijk} equidistant to three obstacles $\mathcal{B}_i$, $\mathcal{B}_j$ and $\mathcal{B}_k$.

Lemma 4.11 The set of critical configurations equidistant to three obstacles K_{ijk} is a subset of $\mathcal{R}_{GVG} \cap K$ for all $i, j, k \in \{1, .., n\}$.

Proof If the robot $\mathcal{A}$ is in a critical configuration $\mathbf{q} \in K$ equidistant to three or more obstacles $\mathcal{B}_i$, $\mathcal{B}_j$, $\mathcal{B}_k$, $\cdots$, $\mathcal{B}_n$, it must certainly be equidistant to three obstacles $\mathcal{B}_i$, $\mathcal{B}_j$ and $\mathcal{B}_k$. Since $\mathcal{A}$ is equidistant to three obstacles, $\mathbf{q}$ must be a member of $\mathcal{R}_{GVG}$, and the lemma follows. ■

More importantly is the existence of a path from sets of critical configurations K_{ij} equidistant to two obstacles $\mathcal{B}_i$ and $\mathcal{B}_j$ onto the $\mathcal{R}_{GVG}$ graph. This path is based on $\mathcal{A}$'s *diameter*

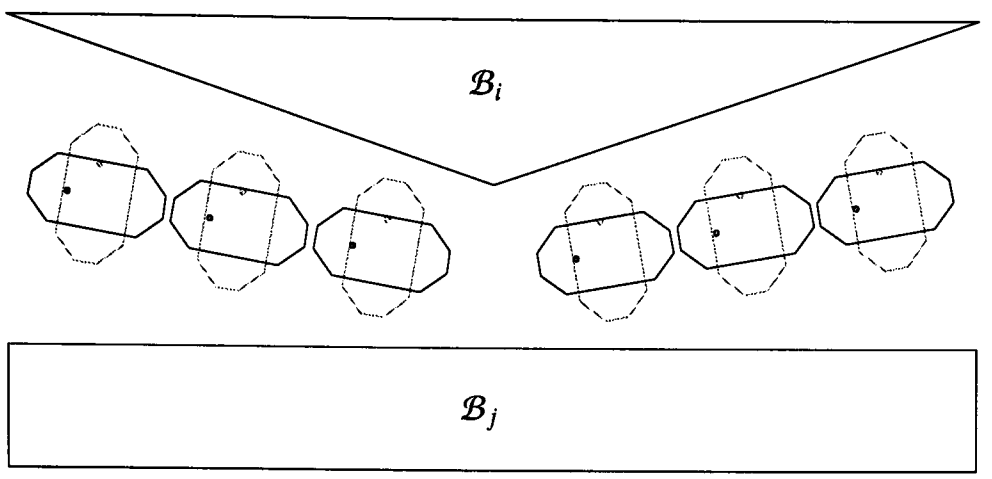


FIGURE 4.11: TWO DIAMETER EXTREMA EDGES $\mathcal{D}_{ij}^{\vartheta}$ FOR TWO EXTREMAL ORIENTATIONS $\vartheta = 0°$ AND $\vartheta = 270°$.

Definition 4.12 The **diameter function** of a convex polygon $\mathcal{A}$ is a mapping $d_{\mathcal{A}} : S^1 \to \mathbb{R}$ that maps the orientation θ of the polygon to distance $d_{\mathcal{A}}(\theta)$ between two parallel supporting lines of constant orientation. (See Figure 4.10(a)) □

There are a finite number of local minima/local maxima of $d_{\mathcal{A}}(\theta)$—Figure 4.10(b) shows the finite number of local extrema for the polygon $\mathcal{A}$ in Figure 4.10(a). We denote this finite set of local extrema with $E_{\vartheta} = \{\vartheta_1 \cdots, \vartheta_n\}$.

The structure connecting K_{ij} to $\mathcal{R}_{GVG}$ is termed a *diameter extrema edge*, denoted $\mathcal{D}_{ij}^{\vartheta}$. Let the angles ϕ_i (resp. ϕ_j) between the vector $\mathbf{ab}_i$ (resp. $\mathbf{ab}_j$) and the orientation vector $[\cos(\theta+\vartheta), \sin(\theta+\vartheta)]^T$ be given by

$$\phi_i(\mathbf{q}) = \tan^{-1}\left(\frac{ab_{iy}(\mathbf{q})}{ab_{ix}(\mathbf{q})}\right) - \theta - \vartheta$$
$$\phi_j(\mathbf{q}) = \theta + \vartheta - \tan^{-1}\left(\frac{ab_{jy}(\mathbf{q})}{ab_{jx}(\mathbf{q})}\right)$$

The diameter extrema edge is defined to be the preimage[9] $\mathcal{D}_{ij}^{\vartheta} = f_{ij}^{-1}(0)$ of $f_{ij} : \mathcal{IS}_{ij} \to \mathbb{R}$ given by

$$\begin{aligned} f_{ij}(\mathbf{q}) &= \phi_i(\mathbf{q}) - \phi_j(\mathbf{q}) \\ &= \tan^{-1}\left(\frac{ab_{iy}(\mathbf{q})}{ab_{ix}(\mathbf{q})}\right) + \tan^{-1}\left(\frac{ab_{jy}(\mathbf{q})}{ab_{jx}(\mathbf{q})}\right) - 2\theta - 2\vartheta \end{aligned} \tag{4.4}$$

[9]We should note that the diameter extrema edge $\mathcal{D}_{ij}^{\vartheta}$ is equivalently described by the zero preimage $f_{ij}^{-1}(0)$ of several different functions—$f_{ij} = \tan^{-1}\left(\frac{ab_{iy}-ab_{jy}}{ab_{ix}-ab_{jx}}\right) - \theta - \vartheta + \frac{\pi}{2}$ and $f_{ij} = \frac{ab_{ix}\cos\theta + ab_{iy}\sin\theta}{\|\mathbf{ab}_i\|} - \frac{ab_{jx}\cos\theta + ab_{jy}\sin\theta}{\|\mathbf{ab}_j\|}$ for example. We've decided to concentrate on Equation 4.4 because it lends itself most easily to a clear and informative analysis. In particular, it is easiest to show the set $f_{ij}^{-1}(0)$ has dimension $\dim(f_{ij}^{-1}(0)) = 1$ using Equation 4.4, even though $f_{ij}^{-1}(0)$ is the same for all examples

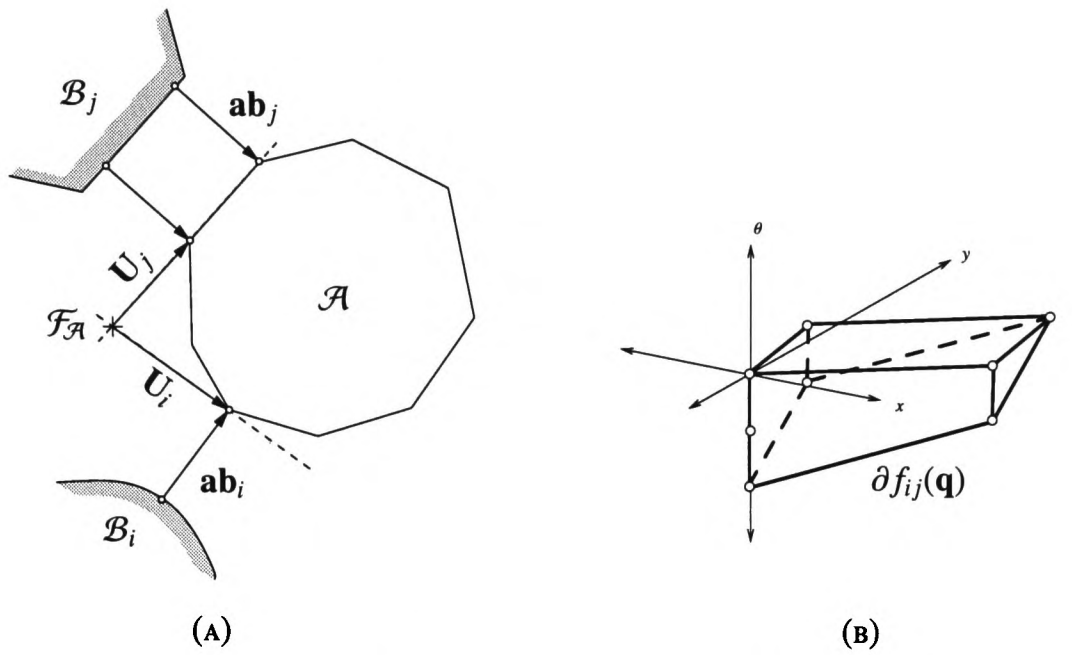


FIGURE 4.12: THE DIAMETER EXTREMA EDGE $\mathcal{D}_{ij}^{\vartheta}$ IS ONE DIMENSIONAL. (A) IT IS POSSIBLE TO PLACE $\mathcal{A}$'S REFERENCE FRAME $\mathcal{F}_\mathcal{A}$ SUCH THAT $\mathbf{U}_i \cdot \mathbf{ab}_i = \mathbf{U}_j \cdot \mathbf{ab}_j = 0$ AND $(\mathbf{U}_i \times \mathbf{ab}_i)_z - (\mathbf{U}_j \times \mathbf{ab}_j)_z \geq 0$. (B) AN EXAMPLE THAT SHOWS EVERY ELEMENT OF $\partial f_{ij}(\mathbf{q})$ IS LINEARLY INDEPENDENT FROM EVERY ELEMENT OF $\partial(d_i - d_j)(\mathbf{q})$ (DOESN'T MATCH (A)).

Figure 4.11(a) shows two $\mathcal{D}_{ij}^{\vartheta}$s for obstacles $\mathcal{B}_i$ and $\mathcal{B}_j$. The diameter extrema edges act to take $\mathcal{A}$ between $\mathcal{B}_i$ and $\mathcal{B}_j$ such that it remains equidistant to them ($d_i(\mathbf{q}) = d_j(\mathbf{q})$) *and* tends to keep the angles between the vectors $\mathbf{ab}_i$ and $\mathbf{ab}_j$ and the support lines equal. We now show the dimension of the diameter minima edges $\dim(\mathcal{D}_{ij}^{\vartheta}) = 1$, so $\mathcal{D}_{ij}^{\vartheta}$ does indeed form an edge of our nonsmooth roadmap NCRIT+NGVG.

Lemma 4.13 The dimension of the diameter extrema edge $\mathcal{D}_{ij}^{\vartheta}$ is $\dim(\mathcal{D}_{ij}^{\vartheta}) = 1$.

Proof We use Lemma 4.3 to determine that $\mathbf{q} \in \mathcal{IS}_{ij}$ is regular when

$$\pm\boldsymbol{\zeta} + \boldsymbol{\nu} \neq \mathbf{0} \text{ for all } \boldsymbol{\zeta} \in \partial f_{ij}(\mathbf{q}), \boldsymbol{\nu} \in \{\lambda\partial(d_i - d_j)(\mathbf{q}) \ : \ \lambda > 0\} \tag{4.5}$$

In Appendix C we give details of the calculation of $\partial f_{ij}(\mathbf{q})$, and show that $\partial f_{ij}(\mathbf{q})$ is constructed from the convex hull of the limit vectors $\lim \nabla f_{ij}(\mathbf{q}_k)$ for nine cases (see Table C.2). In all cases, for each $\mathbf{q} \in \mathcal{IS}_{ij}$, $\|\mathbf{ab}_i\| = \|\mathbf{ab}_j\|$ so we can simply replace all occurrences of this term with $\|\mathbf{ab}\|$. We should also note that in general, the convex hull of all nine terms simultaneously does not satisfy Equation 4.5. However, by placing the origin of $\mathcal{F}_\mathcal{A}$ such that $\mathbf{U}_i \cdot \mathbf{ab}_i = 0$ and $\mathbf{U}_j \cdot \mathbf{ab}_j = 0$ (see Figure 4.12(a)) we can guarantee that the θ coordinate of the limit vectors $\lim \nabla f_{ij}(\mathbf{q}_k)$ is strictly negative for cases 1-8 and zero for case 9. This placement of $\mathcal{F}_\mathcal{A}$ also guarantees that $(\mathbf{U}_i \times \mathbf{ab}_i)_z - (\mathbf{U}_j \times \mathbf{ab}_j)_z$ (the θ coordinate of $\partial(d_i - d_j)(\mathbf{q})$) is greater

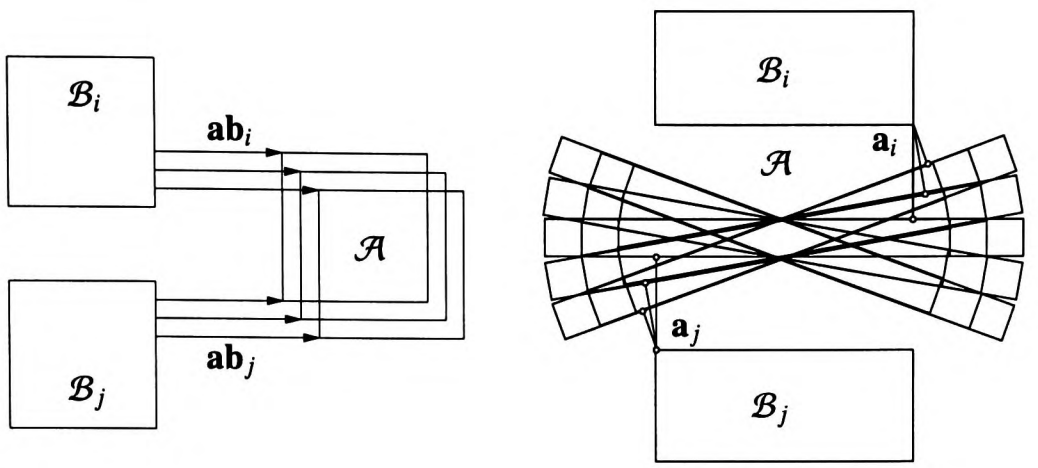


FIGURE 4.13: THE DIAMETER EXTREMA EDGE $\mathcal{D}_{ij}^{\vartheta}$ MAY BE TWO DIMENSIONAL IF $\mathbf{ab}_i = \pm\mathbf{ab}_j$. (A) WHEN $\mathbf{ab}_i = \mathbf{ab}_j$ $\mathcal{A}$ WILL NOT FIT BETWEEN $\mathcal{B}_i$ AND $\mathcal{B}_j$ ANYWAY. (B) WHEN $\mathbf{ab}_i = -\mathbf{ab}_j$ THE ROBOT MAY BE ABLE TO ROTATE WITHOUT CHANGING THE ANGLE BETWEEN $\mathbf{ab}_i/\mathbf{ab}_j$ AND THE SUPPORT LINES.

than or equal to zero. In addition, the x and y coordinates of the limit vectors (denoted f_x, f_y) satisfy

$$f_x(ab_{ix} - ab_{jx}) + f_y(ab_{ix} - ab_{jx}) > 0$$

for all nine case as long as $\mathbf{ab}_i \neq \pm\mathbf{ab}_j$. Thus, for any $\lambda > 0$, there are no elements $\zeta \in \pm\partial f_{ij}(\mathbf{q})$ and $\nu \in \{\lambda\partial(d_i - d_j)(\mathbf{q})$ that satisfy $\zeta = -\nu$ (the θ coordinates always sum to a nonzero value), and the lemma follows. Figure 4.12(b) shows a typical example, gives an idea of the shape of $\partial f_{ij}(\mathbf{q})$, and shows that every element of $\partial f_{ij}(\mathbf{q})$ is linearly independent from every element of $\partial(d_i - d_j)(\mathbf{q})$. ■

An important thing to notice is that the proof of Lemma 4.13 is invalid if $\mathbf{ab}_i = \pm\mathbf{ab}_j$. If $\mathbf{ab}_i = \mathbf{ab}_j$ (see Figure 4.13(a)), $\mathcal{D}_{ij}^{\vartheta}$ cannot be used to take $\mathcal{A}$ between $\mathcal{B}_i$ and $\mathcal{B}_j$ since $\mathcal{D}_{ij}^{\vartheta}$ may be two dimensional. However, when $\mathbf{ab}_i = \mathbf{ab}_j$, $\mathcal{A}$ will not fit between $\mathcal{B}_i$ and $\mathcal{B}_j$ anyway. If $\mathbf{ab}_i = -\mathbf{ab}_j$ (Figure 4.13(b)), as $\mathcal{A}$ rotates and translates, the angle between $[\cos\theta, \sin\theta]^T$ and $\mathbf{ab}_i/\mathbf{ab}_j$ does not change, and $\mathcal{D}_{ij}^{\vartheta}$ is two-dimensional.

4.3.1 Connecting the Critical Set K and the Diameter Extrema Edges $\mathcal{D}_{ij}^{\vartheta}$

Notice in Figure 4.11 that the diameter extrema edge $\mathcal{D}_{ij}^{\vartheta}$ does not form a connected roadmap segment between the obstacles $\mathcal{B}_i$ and $\mathcal{B}_j$, since in the narrow gap $\mathbf{ab}_i = -\mathbf{ab}_j$. In this section we finally show how $\mathcal{D}_{ij}^{\vartheta}$ connects with the critical set K, and finally fully define the roadmap $\mathcal{R}$. We first introduce some

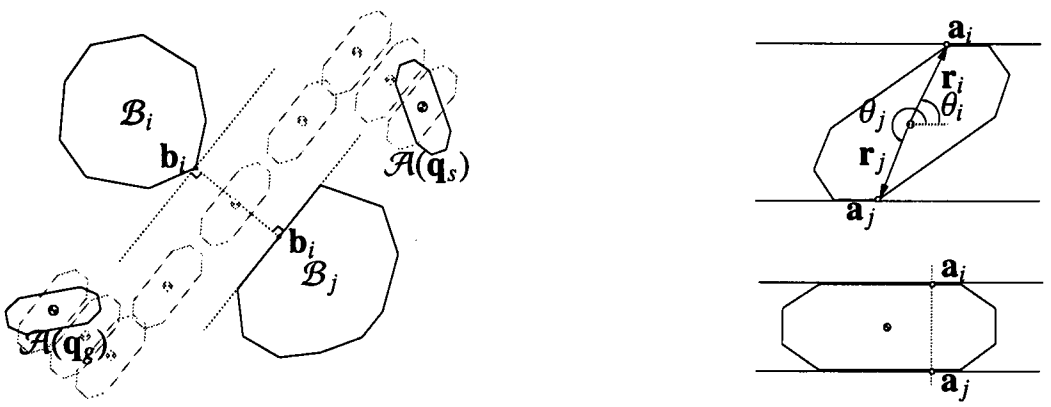


FIGURE 4.14: A PATH EXISTS BETWEEN $\mathcal{B}_i$ AND $\mathcal{B}_j$ WHERE $\mathcal{A}$ HAS AN ORIENTATION OF ϑ WITH RESPECT TO THE SUPPORT LINES. (A) A PATH BETWEEN $\mathcal{B}_i$ AND $\mathcal{B}_j$ WITH AN ORIENTATION OF MINIMUM DIAMETER WITH RESPECT TO THE SUPPORT LINES. (B) IF $\mathcal{A}$ IS IN AN ORIENTATION ϑ OF EXTREMAL DIAMETER, THERE ARE TWO POINTS $\mathbf{a}_i$ AND $\mathbf{a}_i$ SUCH THAT $\mathbf{a}_i\mathbf{a}_j$ IS PERPENDICULAR TO THE SUPPORT LINES.

notation to represent the minimum distance between two obstacles $\mathcal{B}_i$ and $\mathcal{B}_j$

$$\text{dist}(\mathcal{B}_i, \mathcal{B}_j) = \min_{\substack{\mathbf{b}_i \in \mathcal{B}_i \\ \mathbf{b}_j \in \mathcal{B}_j}} \|\mathbf{b}_i - \mathbf{b}_j\|$$

For each pair of points $\mathbf{b}_i$ and $\mathbf{b}_j$ that attain the minimum distance, there are two points $\mathbf{a}_i$ and $\mathbf{a}_j$ on $\mathcal{A}$. Let $\mathbf{b}_i\mathbf{b}_j$ (resp. $\mathbf{a}_i\mathbf{a}_j$) be the vector from $\mathbf{b}_j$ to $\mathbf{b}_i$ (resp. $\mathbf{a}_j$ to $\mathbf{a}_i$).

Lemma 4.14 A collision free path $\tau(t)$ exists between obstacles $\mathcal{B}_i$ and $\mathcal{B}_j$ if and only if

$$\text{dist}(\mathcal{B}_i, \mathcal{B}_j) > d_{\mathcal{A}}(\vartheta) \text{ for some } \vartheta \in E_\vartheta$$

In other words, a path exists if and only if the distance between the obstacles is greater than some diameter extrema value.

Proof **(IF)** If $\text{dist}(\mathcal{B}_i, \mathcal{B}_j) > d_{\mathcal{A}}(\vartheta)$ for some $\vartheta \in E_\vartheta$, we can construct a path as follows (see Figure 4.14(a)): Let $\mathbf{b}_i$ and $\mathbf{b}_j$ be the closest points on $\mathcal{B}_i$ and $\mathcal{B}_j$ respectively. By the Separation Theorem, we can define two supporting lines perpendicular to $\mathbf{b}_i\mathbf{b}_j$. (i) From the start configuration $\mathbf{q}_s$ well clear of the obstacles, rotate the robot until $d_{\mathcal{A}}(\theta) = d_{\mathcal{A}}(\vartheta) < \text{dist}(\mathcal{B}_i, \mathcal{B}_j)$ with respect to the support lines. (ii) Translate the robot to a position midway between the supporting lines. (iii) Translate the robot parallel to the supporting lines. (iv) Once clear of the obstacles, translate and rotate to the goal configuration $\mathbf{q}_g$.

(ONLY IF) If $\text{dist}(\mathcal{B}_i, \mathcal{B}_j) \leq d_{\mathcal{A}}(\vartheta)$ for all $\vartheta \in E_\vartheta$, then all possible paths result in a configuration $\mathbf{q}$ where $\mathcal{A}(\mathbf{q}) \cap \mathcal{B}_i \neq \emptyset$ or $\mathcal{A}(\mathbf{q}) \cap \mathcal{B}_i \neq \emptyset$—a collision. ■

Lemma 4.15 If the robot $\mathcal{A}$ is in an orientation ϑ of extremal diameter $\vartheta \in E_\vartheta$, then there exists two points $\mathbf{a}_i$ and $\mathbf{a}_j$ on the boundary bndy($\mathcal{A}$) such that $\| \mathbf{a}_i - \mathbf{a}_j \| = d_{\mathcal{A}}(\vartheta)$ and $\mathbf{a}_i\mathbf{a}_j$ is perpendicular to the supporting lines.

Proof Assume on the contrary that no such points exist (see Figure 4.14(b)). Then we have a point $\tilde{\mathbf{a}}_i$ at the left (resp. right) of the contact point(s) on one support line, and a point $\tilde{\mathbf{a}}_j$ at the right (resp. left) of the contact point(s) on the other support line. Using a coordinate system with the x-axis parallel to the support lines, we know the rate of change of the y-coordinate with respect to orientation of each point to be

$$\frac{d\tilde{\mathbf{a}}_i^y}{\theta_i} = \|\mathbf{r}_i\| \cos\theta_i, \quad \frac{d\tilde{\mathbf{a}}_j^y}{\theta_j} = \|\mathbf{r}_j\| \cos\theta_j$$

Now, both terms are never simultaneously zero, otherwise $\tilde{\mathbf{a}}_i$ and $\tilde{\mathbf{a}}_j$ are vertically opposite each other, a contradiction. For all locations of P's reference frame, $\frac{d\mathbf{a}_i^y}{\theta_i}$ and $\frac{d\mathbf{a}_j^y}{\theta_j}$ are different, so we can pick a rotation direction leading to an orientation $\tilde{\vartheta}$ near ϑ such that $d_P(\tilde{\vartheta}) < d_P^{min}(\theta)$ or $d_P(\tilde{\vartheta}) > d_P^{max}(\theta)$, a contradiction. ■

Denote by K_{ij}^{ϑ} the subset of critical points K_{ij} where the orientation coordinate $q_\theta = \vartheta$.

Lemma 4.16 If a collision free path $\tau(t)$ exists that takes the robot $\mathcal{A}$ between the obstacles $\mathcal{B}_i$ and $\mathcal{B}_j$, then there exists a collision free path $\tau_\vartheta(t)$ between $\mathcal{B}_i$ and $\mathcal{B}_j$ containing the set $K_{ij}^{\vartheta} \subseteq \tau(t)$ for each $\vartheta \in E_\vartheta$ satisfying $d_{\mathcal{A}}(\vartheta) < \text{dist}(\mathcal{B}_i, \mathcal{B}_j)$.

Proof If there exists a path $\tau(t)$, then we must have $\text{dist}(\mathcal{B}_i, \mathcal{B}_j) > 0$. Since $\text{dist}(\mathcal{B}_i, \mathcal{B}_j) > 0$, by the Separation Theorem, we can construct two supporting lines perpendicular to $\mathbf{b}_i - \mathbf{b}_j$ as in Lemma 4.14. We also know by Lemma 4.14 that a path $\tau'(t)$ exists such that $\mathcal{A}$ passes between $\mathcal{B}_i$ and $\mathcal{B}_j$ with an extremal diameter with respect to the support lines. By Lemma 4.15 there are two points $\mathbf{a}_i$ and $\mathbf{a}_j$ such that $\mathbf{a}_i\mathbf{a}_j$ is perpendicular to the supporting lines as well, hence there is at least one configuration $\mathbf{q}_c$ where $\mathbf{a}_i\mathbf{b}_i$ is parallel to $\mathbf{a}_j\mathbf{b}_i$ and $\| \mathbf{b}_i - \mathbf{a}_i \| = \| \mathbf{b}_j - \mathbf{a}_j \| = d_i(\mathbf{q}) = d_j(\mathbf{q})$.

We now show $\mathbf{q}_c$ is in fact critical, and thus a member of K_{ij}^{ϑ}. In order to be critical, there must exist points $\zeta_i \in \partial d_i(\mathbf{q}_c)$ and $\zeta_j \in \partial d_j(\mathbf{q}_c)$ such that $\zeta_i = -\zeta_j$, otherwise the origin will not be contained in the convex hull $\text{co}\{\partial d_i(\mathbf{q}_c), \partial d_j(\mathbf{q}_c)\}$. Recalling Equation 3.11 for the gradient $\nabla d_i(\mathbf{q})$ (we need only consider terms 1,2,4), we need only to find two points on bndy($\mathcal{A}$) where $\nabla d_i(\mathbf{q}_c) = -\nabla d_j(\mathbf{q}_c)$. For points $\mathbf{a}_i$ and $\mathbf{a}_j$

we immediately have

$$(\mathbf{a}_i - \mathbf{b}_i)_x = -(\mathbf{a}_j - \mathbf{b}_j)_x$$

$$(\mathbf{a}_i - \mathbf{b}_i)_y = -(\mathbf{a}_j - \mathbf{b}_j)_y$$

With a suitable change of world coordinates, such that either the x or y axis is aligned with the supporting lines (assume x-axis is aligned so that $(\mathbf{a}_i - \mathbf{b}_i)_x = 0$), we immediately have $U_x^i = U_x^j$ and

$$U_x^i(\mathbf{a}_i - \mathbf{b}_i)_y = -U_x^j(\mathbf{a}_j - \mathbf{b}_j)_y$$

and thus $\mathbf{q}_c$ is a critical configuration. Since the selection of $\mathbf{q}_c \in K_{ij}^{\vartheta}$ is arbitrary, the path $\tau_{\vartheta}(t) = \tau'(t)$ must contain the set K_{ij}^{ϑ}, so the lemma follows. ■

We now show that the set $K_{ij}^{\vartheta} \cup \mathcal{D}_{ij}^{\vartheta}$ forms a connected roadmap segment between $\mathcal{B}_i$ and $\mathcal{B}_j$.

Lemma 4.17 If a collision free path $\tau(t)$ exists that takes the robot $\mathcal{A}$ between the obstacles $\mathcal{B}_i$ and $\mathcal{B}_j$, then the roadmap segment constructed by the union of the diameter extrema edge $\mathcal{D}_{ij}^{\vartheta}$ and the set of critical configurations K_{ij}^{ϑ} forms a connected path $\tau(t)$ between obstacles $\mathcal{B}_i$ and $\mathcal{B}_j$ for all $\vartheta \in E_{\vartheta}$ satisfying $d_{\mathcal{A}}(\vartheta) < \text{dist}(\mathcal{B}_i, \mathcal{B}_j)$.

Proof The set $\mathcal{D}_{ij}^{\vartheta} \cup K_{ij}^{\vartheta}$ is connected. By Lemma 4.16, the existence of the path $\tau(t)$ establishes the existence of a path $\tau_{\vartheta}(t)$ through the set K_{ij}^{ϑ} for each $\vartheta \in E_{\vartheta}$ satisfying $d_{\mathcal{A}}(\vartheta) < \text{dist}(\mathcal{B}_i, \mathcal{B}_j)$. The set $\mathcal{D}_{ij}^{\vartheta} \cup K_{ij}^{\vartheta}$ thus forms a connected path between $\mathcal{B}_i$ and $\mathcal{B}_j$ for each $\vartheta \in E_{\vartheta}$ satisfying $d_{\mathcal{A}}(\vartheta) < \text{dist}(\mathcal{B}_i, \mathcal{B}_j)$. ■

Finally, we show that some $K_{ij}^{\vartheta} \cup \mathcal{D}_{ij}^{\vartheta}$ is sufficient to connect two adjacent accessibility regions $\mathcal{A}_{ijk}$, $\mathcal{A}_{ijl}$.

Theorem 4.18 Let two configurations $\mathbf{q}_1$ and $\mathbf{q}_2$ in the invertible three-equidistant faces $\mathcal{IF}_{ijk}$ and $\mathcal{IF}_{ijl}$ respectively also be members of the diameter extrema edge $\mathcal{D}_{ij}^{\vartheta}$ (or K_{ij}^{ϑ}), and implicitly members of adjacent accessibility regions $\mathcal{A}_{ijk}$ and $\mathcal{A}_{ijl}$. $\mathbf{q}_1$ and $\mathbf{q}_2$ are path connected if and only if the union of the diameter extrema edge $\mathcal{D}_{ij}^{\vartheta}$ and the critical set K_{ij}^{ϑ} is connected for some $\vartheta \in E_{\vartheta}$.

Proof **(IF)** If the union $\mathcal{D}_{ij}^{\vartheta} \cup K_{ij}^{\vartheta}$ is connected then there exists a path $\tau(t) = \mathcal{D}_{ij}^{\vartheta} \cup K_{ij}^{\vartheta}$ between the configurations $\mathbf{q}_1$ and $\mathbf{q}_2$.

(ONLY IF) Let $\tau(t)$ be a continuous path between configurations $\mathbf{q}_1$ and $\mathbf{q}_2$. By Lemma 4.17, the existence

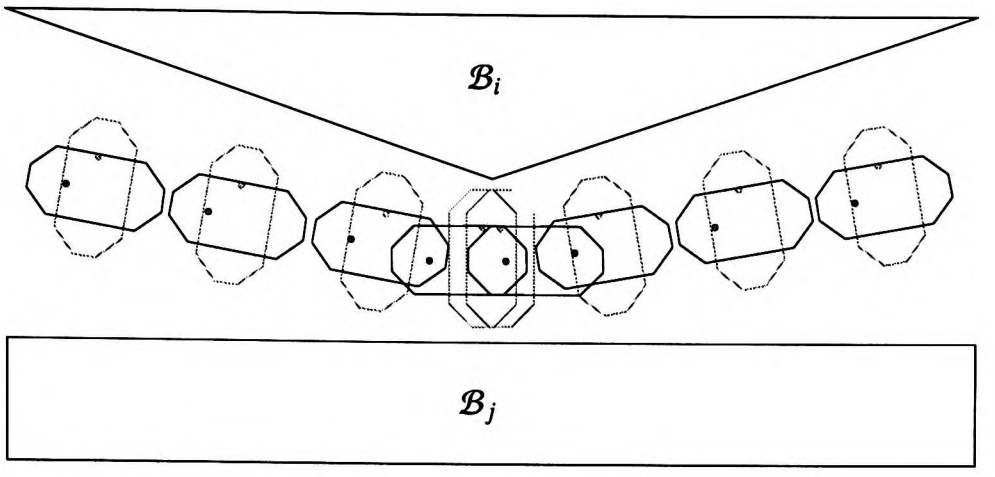


FIGURE 4.15: DIAMETER EXTREMA EDGE $\mathcal{D}_{ij}^{\vartheta}$ PLUS CRITICAL SET K_{ij}^{ϑ}. THE UNION OF THE TWO SETS CREATES A CONNECTED ROADMAP SEGMENT BETWEEN THE OBSTACLES $\mathcal{B}_i$ AND $\mathcal{B}_j$.

of a path $\tau(t)$ establishes the existence of a set $K_{ij}^{\vartheta} \cup \mathcal{D}_{ij}^{\vartheta}$ between $\mathcal{B}_i$ and $\mathcal{B}_j$ for each $\vartheta \in E_{\vartheta}$ satisfying $d_{\mathcal{A}}(\vartheta) < \text{dist}(\mathcal{B}_i, \mathcal{B}_j)$. ■

We can now finally describe the entire roadmap NCRIT+NGVG. Denote the set of all diameter extrema edges $\mathcal{R}_D$

$$\mathcal{R}_D = \cup_{i,j,\vartheta} \mathcal{D}_{ij}^{\vartheta}$$

and let NCRIT+NGVG $= \mathcal{R}_{GVG} \cup \mathcal{R}_K \cup \mathcal{R}_D$. By Lemmas 4.11 and 4.17, NCRIT+NGVG connects and contains all $\check{K}_i$s. By Theorems 4.9 and 4.18, NCRIT+NGVG represents the connectivity of C_{free} and can thus be used for complete sensor-based planning.

4.4 Summary

In this Chapter we used the mathematical tools presented in Chapter 3 to develop a mathematical graph structure called termed the Nonsmooth Critical Set Generalised Voronoi Graph and denoted NCRIT+NGVG. The NCRIT+NGVG consists of the union of three components $\mathcal{R}_{GVG}$, $\mathcal{R}_K$ and $\mathcal{R}_D$.

$\mathcal{R}_{GVG}$ was defined as the union of one dimensional roadmap segments and zero dimensional roadmap nodes consisting of configurations where the robot $\mathcal{A}$ is equidistant to three and four obstacles respectively. We constructed $\mathcal{R}_{GVG}$ by first defining an invertible two-equidistant face $\mathcal{IF}_{ij}$ where $\mathcal{A}$ is equidistant to two obstacles using the concept of the generalised gradient. We then proceeded to define the invertible three-equidistant face $\mathcal{IF}_{ijk}$ where $\mathcal{A}$ is equidistant to three obstacles using the notion of weak slope. The invertible four-equidistant face $\mathcal{IS}_{ijkl}$ was then defined to be the zero dimensional intersection of $\mathcal{IF}_{ijk}$,

$\mathcal{IF}_{ijl}$ and $\mathcal{IF}_{jkl}$. It was shown that $\mathcal{R}_{GVG}$ is not sufficient for complete motion planning, so we next defined the critical set graph $\mathcal{R}_K$.

$\mathcal{R}_K$ is the subset of the NCRIT+NGVG where $D(\mathbf{q})$ is critical. The critical set graph $\mathcal{R}_K$ was shown to be equal to the entire critical set K by showing that the dimension of K is less than or equal to one. It was also shown that $\mathcal{R}_{GVG} \cup \mathcal{R}_K$ is not sufficient for complete motion planning, so we next defined the set of diameter extrema edges $\mathcal{R}_D$.

The diameter extrema edges $\mathcal{D}_{ij}^{\vartheta}$ were defined as the zero preimage of a function $f_{ij}(\mathbf{q})$ that tends to keep the angle between the witness vectors $\mathbf{ab}_i/\mathbf{ab}_j$ and the support lines of orientation ϑ equal. Utilising a detailed analysis of the generalised derivative $\partial f_{ij}(\mathbf{q})$ in Appendix C, we showed that $\mathcal{D}_{ij}^{\vartheta}$ is in fact one-dimensional and forms a component of NCRIT+NGVG. The union of all the $\mathcal{D}_{ij}^{\vartheta}$s is the graph $\mathcal{R}_D$.

Finally, it was shown that the set $\mathcal{R} = \mathcal{R}_{GVG} \cup \mathcal{R}_K \cup \mathcal{R}_D$ is sufficient for the complete sensor-based motion planning problem. This was accomplished by first showing $\mathcal{R}_{GVG}$ to be sufficient for planning within the accessibility regions $\mathcal{A}_{ijk}$ by showing a retraction exists that takes $\mathcal{A}_{ijk}$ onto an invertible three-equidistant face $\mathcal{IF}_{ijk}$. Secondly, the set $\mathcal{R}_K \cup \mathcal{R}_D$ was then shown to connect adjacent accessibility regions (Lemma 4.17), thus ensuring $\mathcal{R}$ forms a connected roadmap of freespace.

§5

"Of all men's miseries the bitterest is this: to know so much and to have control over nothing."

Herodotus

Following Roadmap Segments using Nonsmooth Control Laws

In this chapter we present a number of nonsmooth control laws that allow the robot $\mathcal{A}$ to explore the nonsmooth roadmap NCRIT+NGVG defined in Chapter 4. After briefly reviewing some relevant concepts from control theory in Section 5.1 and reviewing a smooth control law on which our technique is based in Section 5.2 we begin examining the four control laws. In Section 5.3 we present a nonsmooth control law that allows $\mathcal{A}$ to follow an invertible three-equidistant face $\mathcal{IF}_{ijk}$, in Section 5.4 we present a nonsmooth control law that allows $\mathcal{A}$ to localise an invertible four-equidistant face $\mathcal{IF}_{ijkl}$, in Section 5.5 we present a nonsmooth control law that allows $\mathcal{A}$ to follow a diameter extrema edge $\mathcal{D}_{ij}^{\vartheta}$, in Section 5.6 we present a nonsmooth control law that allows $\mathcal{A}$ to follow a critical set edge K_{ij}^{ϑ}. Finally, in Section 5.7 we examine the stability of the four control laws in previous sections. Let us start with the promised review of control theory.

5.1 Short Overview of Applicable Control Theory

In this section we briefly describe the concepts of a standard control system and stability, and provide a few useful definitions that enable us to understand the nonsmooth control laws presented in Sections 5.3,

5.4 and 5.5.

The standard control system is given by

$$\dot{\mathbf{x}}(t) = f(t, \mathbf{x}(t), \mathbf{u}(t))$$

where $f : \mathbb{R} \times \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$, t is time, $\mathbf{x}(t)$ is position, $\dot{\mathbf{x}}(t)$ is $\mathbf{x}(t)$'s derivative with respect to time, and $\mathbf{u}(t)$ is the control function. When there are no control inputs $\mathbf{u}(t)$ and $\dot{\mathbf{x}}(t)$ has no explicit time dependency (as is the case in our work), then the system

$$\dot{\mathbf{x}}(t) = f(\mathbf{x}(t)) \tag{5.1}$$

is referred to as an *autonomous differential equation*. A *trajectory* $\mathbf{x}(\cdot)$ of the differential equation is taken to mean an absolutely continuous $\mathbf{x} : [a, b] \rightarrow \mathbb{R}^n$ which, together with $\dot{\mathbf{x}}(t)$, satisfies 5.1. In the classical theory of differential equations, $f(\mathbf{x}(t))$ must be continuous, however the nonsmooth control theory outlined in [17] makes no such requirement.

A point $\tilde{\mathbf{x}}$ is an *equilibrium point* of the system 5.1 if $\mathbf{x}(t) = \tilde{\mathbf{x}}$ for all $t \geq a$. In other words, if at time a the system is in configuration $\tilde{\mathbf{x}}$, and $\tilde{\mathbf{x}}$ is an equilibrium point, then the system will remain in configuration $\tilde{\mathbf{x}}$ indefinitely. The equilibrium point $\tilde{\mathbf{x}} = \mathbf{0}$ is said to be *stable* if, for any $\epsilon > 0$, there exists $\delta > 0$, such that $\| \mathbf{x}(0) \| < \delta$ implies $\| \mathbf{x}(t) \| < \epsilon$ for all $t \geq 0$. In other words, the system trajectory can be kept arbitrarily close to the equilibrium point by starting sufficiently close to it. Stability can be verified by using the simple and *very* commonplace notion of Lyapunov functions. Suppose that smooth $V(\mathbf{x})$ exists such that:

- Positive definiteness: $V(\mathbf{x}) \geq 0$ for all $\mathbf{x} \in \mathbb{R}^n$
- Infinitesimal decrease: $\langle \nabla V(\mathbf{x}), f(\mathbf{x}) \rangle \leq 0$ for all $\mathbf{x} \in \mathbb{R}^n$

It follows that $\tilde{\mathbf{x}} = \mathbf{0}$ is stable [17].

In [17], the notion of Lyapunov stability is extended to include nonsmooth $V(\mathbf{x})$, which we use in Section 5.7 to verify the stability of our nonsmooth control laws. This is accomplished by modifying the infinitesimal decrease property to

$$\langle \zeta, f(\mathbf{x}) \rangle \leq 0 \text{ for all } \zeta \in \partial V(\mathbf{x}) \text{ and for all } \mathbf{x} \in \mathbb{R}^n. \tag{5.2}$$

5.2 Overview of Existing Smooth Control Law

In this section we give a brief overview of the control law used in [13] to allow a point robot $\mathcal{A}$ to follow Generalised Voronoi Diagram (GVG) [12] segments and localise GVG nodes, which our control laws are based upon. The general approach is based on *numerical root following*, which is a technique used to trace the (one-dimensional) root set of a vector-valued function. The function of interest is

$$G(\mathbf{q}) = \begin{bmatrix} (d_1 - d_2)(\mathbf{q}) \\ \vdots \\ (d_1 - d_n)(\mathbf{q}) \end{bmatrix} \tag{5.3}$$

which has roots that coincide with the GVG—i.e. $G(\mathbf{q}) = \mathbf{0}$ if and only if $d_i(\mathbf{q}) = d_j(\mathbf{q}) = \cdots = d_n(\mathbf{q})$. The smooth control law states that a point robot $\mathcal{A}$ at configuration $\mathbf{q}$ should make a step in direction $\dot{\mathbf{q}}$

$$\dot{\mathbf{q}} = \alpha \text{Null}\left(\nabla G(\mathbf{q})\right) + \beta \nabla G(\mathbf{q})^{\dagger} G(\mathbf{q}) \tag{5.4}$$

where $\nabla G(\mathbf{q})^{\dagger}$ is the *Penrose pseudo-inverse* [13] of $\nabla G(\mathbf{q})$ defined as

$$\nabla G(\mathbf{q})^{\dagger} = \nabla G(\mathbf{q})^{T} (\nabla G(\mathbf{q}) \nabla G(\mathbf{q})^{T})^{-1}$$

The term $\beta \nabla G(\mathbf{q})^{\dagger} G(\mathbf{q})$ acts to draw the robot onto the GVG (correction term) and the term αNull $(\nabla G(\mathbf{q}))$ tends to draw the robot along the GVG (follow term). For general n, $\nabla G(\mathbf{q})^{\dagger} G(\mathbf{q})$ has the property

$$\nabla G(\mathbf{q}) \nabla G(\mathbf{q})^{\dagger} = \mathbf{I} \text{ (identity matrix)}$$

so the correction term acts to simultaneously take $(d_1 - d_2)(\mathbf{q}), \cdots, (d_1 - d_n)(\mathbf{q})$ closer to zero as long as β is negative and $(\nabla G(\mathbf{q}) \nabla G(\mathbf{q})^{T})^{-1}$ exists—which is the case when all rows of $\nabla G(\mathbf{q})$ are nonzero and linearly independent. The follow term is a member of the nullspace of $G(\mathbf{q})$ and so acts to simultaneously keep $(d_1 - d_2)(\mathbf{q}), \cdots, (d_1 - d_n)(\mathbf{q})$ at the same value.

In Figure 5.1 we illustrate the case when $n = 2$ and $\mathcal{A}$ is following a GVG segment generated by $\mathcal{B}_i$ and $\mathcal{B}_j$. When $n = 2$, the control law becomes

$$\dot{\mathbf{q}} = \alpha \nabla (d_i - d_j)(\mathbf{q})^{\perp} + \beta \frac{(d_i - d_j)(\mathbf{q})}{\nabla (d_1 - d_2)(\mathbf{q})}$$

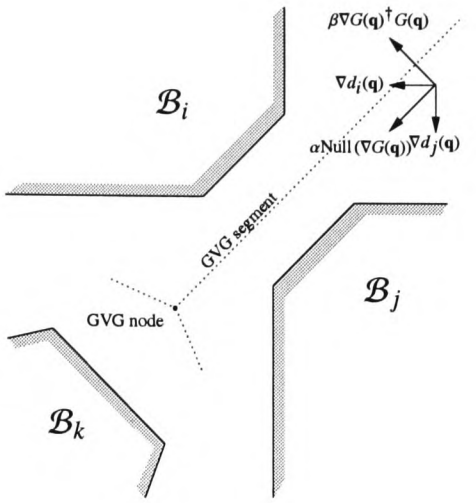


FIGURE 5.1: ILLUSTRATION OF THE 'FOLLOW' TERM $\alpha\text{Null}(\nabla G(\mathbf{q}))$ AND THE 'CORRECTION' TERM $\beta\nabla G(\mathbf{q})^\dagger G(\mathbf{q})$ IN THE SMOOTH CONTROL LAW. $\alpha\text{Null}(\nabla G(\mathbf{q}))$ POINTS ALONG THE GVG SEGMENT AND $\beta\nabla G(\mathbf{q})^\dagger G(\mathbf{q})$ POINTS TOWARD THE GVG (IF $\beta < 0$).

where the follow term is simply a vector $\alpha\nabla(d_1 - d_2)(\mathbf{q})^\perp$ perpendicular to $\nabla(d_1 - d_2)(\mathbf{q})$ and the correction term is the vector $\beta\frac{(d_1-d_2)(\mathbf{q})}{\nabla(d_1-d_2)(\mathbf{q})}$. When $n = 3$, a GVG segment is defined by equidistance to three obstacles $\mathcal{B}_i$, $\mathcal{B}_j$ and $\mathcal{B}_k$, and the follow term is a vector perpendicular to both $\nabla(d_i - d_j)(\mathbf{q})$ and $\nabla(d_i - d_k)(\mathbf{q})$—the follow term is effectively the cross product $\nabla(d_i - d_j)(\mathbf{q}) \times \nabla(d_i - d_k)(\mathbf{q})$.

When in the vicinity of a GVG node, a similar control law is used to localise the node. Since a GVG node when $n = 2$ is equidistant to three obstacles $\mathcal{B}_i$, $\mathcal{B}_j$ and $\mathcal{B}_k$, the function $G(\mathbf{q})$ used for localisation is given by

$$G(\mathbf{q}) = \begin{bmatrix} (d_i - d_j)(\mathbf{q}) \\ (d_i - d_k)(\mathbf{q}) \end{bmatrix}$$

Since a GVG node is zero-dimensional, the follow term is of no use, and we have only a correction term in the control law

$$\dot{\mathbf{q}} = \beta\nabla G(\mathbf{q})^\dagger G(\mathbf{q})$$

Notice that in the smooth case, both the nullspace $\nabla G(\mathbf{q})^T$ and Penrose pseudo-inverse $\nabla G(\mathbf{q})^\dagger$ exist if and only if the rows of $\nabla G(\mathbf{q})$ are nonzero and linearly independent. In [13] this was shown to be true for the smooth case, and in the following section we extend to the nonsmooth case. Effectively, we show that all elements of $\partial G(\mathbf{q})$ have rows that are nonzero and linearly independent – a somewhat expected result. Let us begin our examination of the nonsmooth control laws in $SE(3)$ based on the smooth law just presented.

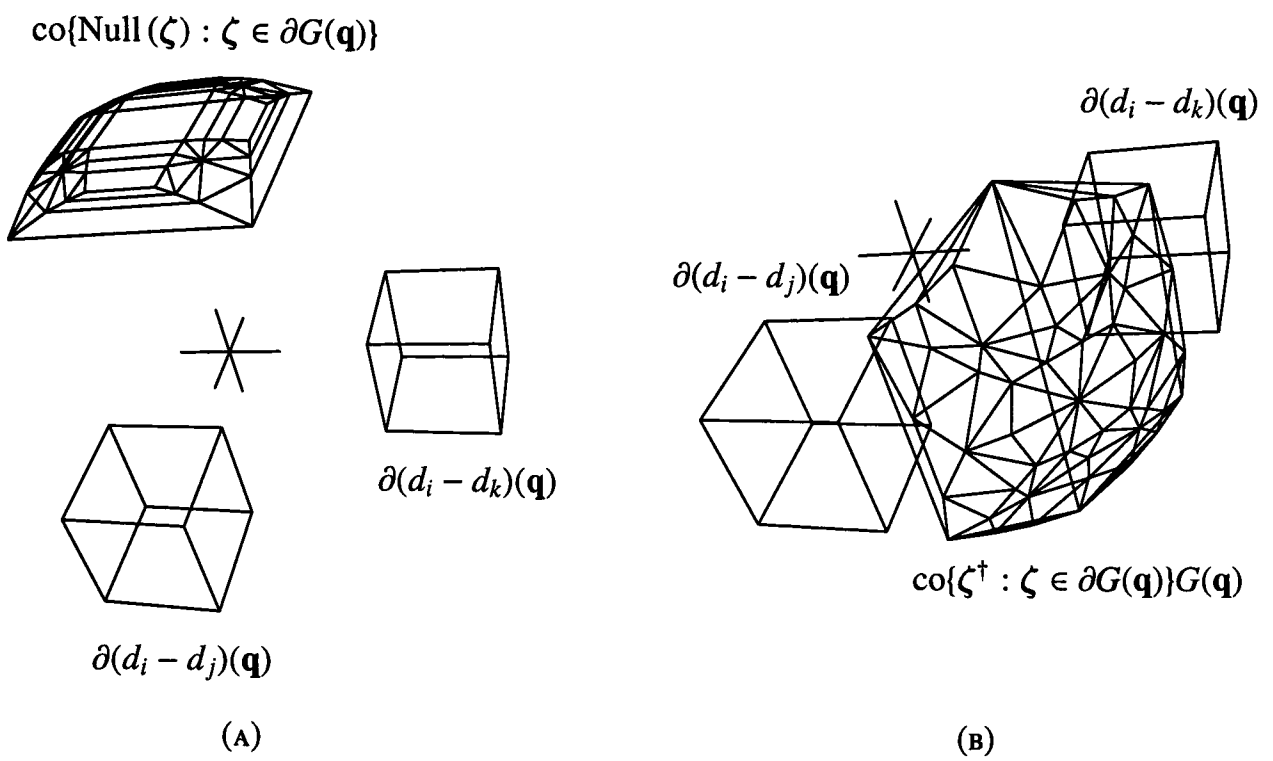


FIGURE 5.2: EXPLANATION OF NONSMOOTH CONTROL TERMS $\alpha\mathbf{a}$ AND $\beta\mathbf{b}$. IN THIS EXAMPLE, ALL ELEMENTS OF $\partial(d_i - d_j)(\mathbf{q})$ AND $\partial(d_i - d_k)(\mathbf{q})$ ARE NONZERO AND LINEARLY INDEPENDENT. (A) THE SET $A = \text{co}\{\text{Null}(\zeta) : \zeta \in \partial G(\mathbf{q})\}$ (SCALED TO FIT IN FIGURE). (B) THE SET $B = \text{co}\{\zeta^\dagger : \zeta \in \partial G(\mathbf{q})\}G(\mathbf{q})$ (SCALED TO FIT IN FIGURE) WHEN $(d_i - d_j)(\mathbf{q})$ AND $(d_i - d_k)(\mathbf{q})$ ARE POSITIVE.

5.3 Following Three-Equidistant Edges $\mathcal{IF}_{ijk}$

In this section we present a control law that allows $\mathcal{A}$ to follow an invertible three-equidistant face $\mathcal{IF}_{ijk}$. We recall that $\mathcal{IF}_{ijk}$ is defined as the set of configurations where $d_i(\mathbf{q}) = d_j(\mathbf{q}) = d_k(\mathbf{q}) < d_l(\mathbf{q})$ for all $l \neq i, j, k$. This is also equivalently described as the zeros of the vector valued function $G(\mathbf{q})$

$$G(\mathbf{q}) = \begin{bmatrix} (d_i - d_j)(\mathbf{q}) \\ (d_i - d_k)(\mathbf{q}) \end{bmatrix}$$

In contrast to the point GVG, this $G(\mathbf{q})$ is composed of nonsmooth terms $(d_i - d_j)(\mathbf{q})$ and $(d_i - d_j)(\mathbf{q})$, so $G(\mathbf{q})$ is itself a nonsmooth function. The control law developed for following $\mathcal{IF}_{ijk}$ is as follows: at configuration $\mathbf{q}$, $\mathcal{A}$ makes a step in the direction

$$\dot{\mathbf{q}} = \alpha\mathbf{a} + \beta\mathbf{b} \tag{5.5}$$

where the vector **a** satisfies

$$\begin{aligned} &1.\ \mathbf{a} \in \text{co}\{\text{Null}\,(\zeta) : \zeta \in \partial G(\mathbf{q})\} \\ &2.\ \langle \mathbf{a}, \sigma_z \rangle \leq 0 \text{ for all } \sigma_z \in (d_i - d_z)(\mathbf{q})\partial(d_i - d_z)(\mathbf{q}),\ \text{for all } z \in \{j, \cdots, k\} \end{aligned} \tag{5.6}$$

and the vector **b** satisfies

$$\begin{aligned} &1.\ \mathbf{b} \in \text{co}\{\zeta^\dagger : \zeta \in \partial G(\mathbf{q})\}G(\mathbf{q}) \\ &2.\ \langle \mathbf{b}, \sigma_z \rangle \geq 0 \text{ for all } \sigma_z \in (d_i - d_z)(\mathbf{q})\partial(d_i - d_z)(\mathbf{q}),\ \text{for all } z \in \{j, \cdots, k\} \end{aligned} \tag{5.7}$$

As in the smooth case, the α term is a follow term and the β term is a correction term.

Recall that in the smooth case (see Figure 5.1), the action of the follow term in Equation 5.4 is to move the robot so that, *locally, there is no change in the value of* the components of $G(\mathbf{q})$. Similarly, the follow term in Equation 5.5 acts to *minimise* the change in the components of $G(\mathbf{q})$ at nonsmooth points. Also recall that in the smooth case the correction term in Equation 5.4 acts to move the robot so that *the components of* $G(\mathbf{q})$ *simultaneously get closer to zero.* Similarly, the correction term in Equation 5.5 acts to move the robot so that *as many components as possible* of $G(\mathbf{q})$ simultaneously get closer to zero. When $D(\mathbf{q})$ is smooth, Equation 5.5 is equivalent to Equation 5.4 (the hulls are singletons).

The existence of vectors **a** and **b** satisfying Equations 5.6 and 5.7 is not immediately obvious, so we must explore this a bit further. Let us first examine the vector **a**.

Lemma 5.1 There exists a vector **a** that satisfies (1) and (2) in Equation 5.6.

Proof Let us note in general that for an arbitrary vector-valued function $\tilde{G}(\mathbf{x}) = [g_1(\mathbf{x}), \cdots, g_{n-1}(\mathbf{x})]^T$, there does not exist a vector $\tilde{\mathbf{a}}$ satisfying

$$\begin{aligned} &1.\ \tilde{\mathbf{a}} \in \text{co}\{\text{Null}\,(\zeta) : \zeta \in \partial \tilde{G}(\mathbf{q})\} \\ &2.\ \langle \tilde{\mathbf{a}}, \sigma_z \rangle \leq 0 \text{ for all } \sigma_z \in g_h(\mathbf{x})\partial g_h(\mathbf{x}),\ \text{for all } z \in \{1, \cdots, n-1\} \end{aligned}$$

To see this, consider a function $\tilde{G}(\mathbf{x})$ with $\mathbf{0} \in \text{int}(\partial g_1(\mathbf{x}))$. For any $\tilde{\mathbf{a}} \in \mathbb{R}^n$, $\langle \tilde{\mathbf{a}}, \sigma_1 \rangle > 0$ for all $\sigma_1 \in g_1(\mathbf{x})\partial g_1(\mathbf{x})$. This proof thus relies critically on the nature of the $d_i(\mathbf{q})$s in $SE(2)$.

Firstly, the set $A = \text{co}\{\text{Null}\,(\zeta) : \zeta \in \partial G(\mathbf{q})\}$ exists if and only if $\text{Null}\,(\zeta)$ exists for all $\zeta \in \partial G(\mathbf{q})$. $\text{Null}\,(\zeta)$ exists if and only if the rows of ζ are non-zero and linearly independent. The first (resp. second) row of ζ is an element of $\partial(d_i - d_j)(\mathbf{q})$ (resp. $\partial(d_i - d_k)(\mathbf{q})$). Since $\partial d_i(\mathbf{q}) \cap \partial d_j(\mathbf{q}) = \varnothing$ for all $i \neq j$, we know

$\mathbf{0} \notin \partial(d_i - d_j)(\mathbf{q})$ for all $i \neq j$ (recall Lemma 4.1). By definition of $\mathcal{IF}_{ijk}$, all elements of $\partial(d_i - d_j)(\mathbf{q})$ and $\partial(d_i - d_k)(\mathbf{q})$ linearly independent. Since elements of $\partial(d_i - d_j)(\mathbf{q})$ and $\partial(d_i - d_k)(\mathbf{q})$ are linearly independent nonzero vectors, the existence of A is assured and (1) is satisfied.

Secondly, a component $\check{\mathbf{a}} \in A$ is by construction equal to the nullspace of two vectors $\boldsymbol{\sigma}_j \in \partial(d_i - d_j)(\mathbf{q})$ and $\boldsymbol{\sigma}_k \in \partial(d_i - d_k)(\mathbf{q})$ and thus $\check{\mathbf{a}}$ is perpendicular to $\pm\boldsymbol{\sigma}_j$ and $\pm\boldsymbol{\sigma}_k$. Now, since elements of $\partial(d_i - d_j)(\mathbf{q})$ and $\partial(d_i - d_k)(\mathbf{q})$ are linearly independent and $\mathbf{0} \notin \pm\partial(d_i - d_j)(\mathbf{q}), \pm\partial(d_i - d_k)(\mathbf{q})$ we can choose points $\boldsymbol{\sigma}_j \in \text{bndy}((d_i - d_j)(\mathbf{q})\partial(d_i - d_j)(\mathbf{q}))$ and $\boldsymbol{\sigma}_k \in \text{bndy}((d_i - d_k)(\mathbf{q})\partial(d_i - d_k)(\mathbf{q}))$ which, along with the origin, define a hyperplane with normal vector $\check{\mathbf{a}}$ that satisfies Equation 5.6. If this wasn't the case, then we could find linearly dependent elements of $\partial(d_i - d_j)(\mathbf{q})$ and $\partial(d_i - d_k)(\mathbf{q})$—a contradiction. Letting $\mathbf{a} = \check{\mathbf{a}}$, (2) is thus satisfied and the lemma follows. Figure 5.3(a) illustrates a vector $\mathbf{a}$ that satisfies Equations 5.6. ■

Now we present a similar proof for vector $\mathbf{b}$

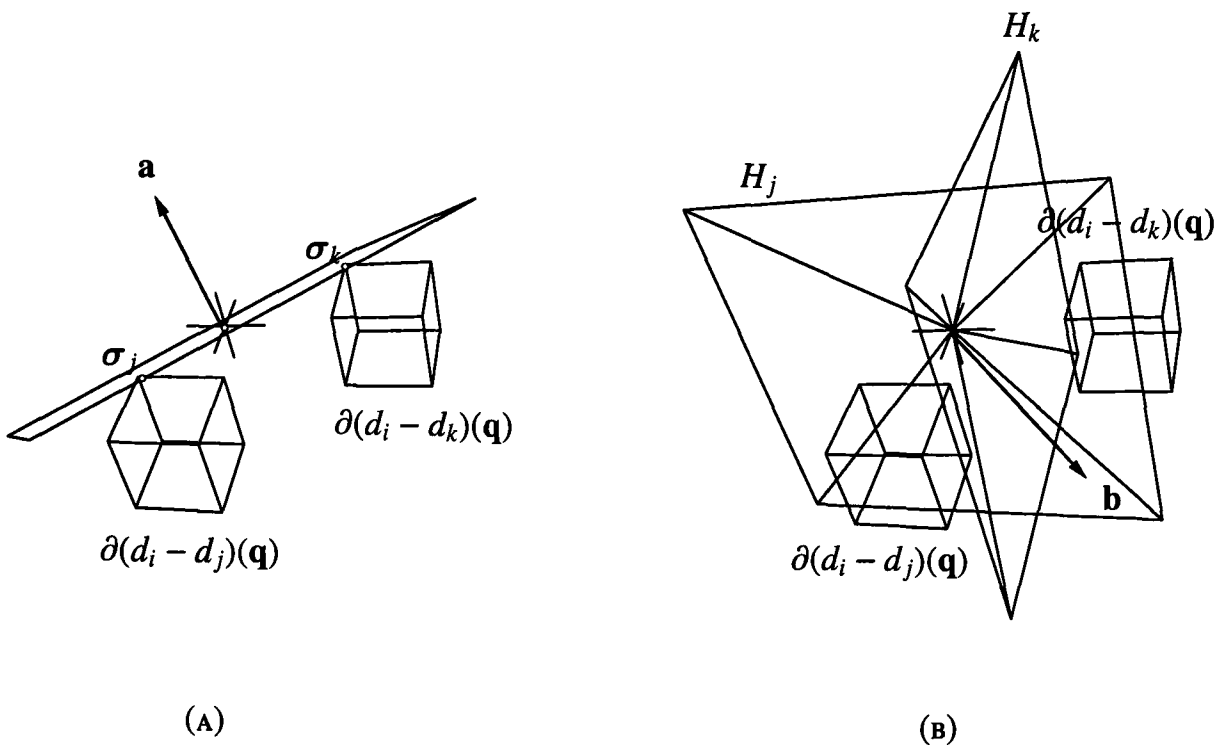
Lemma 5.2 There exists a vector $\mathbf{b}$ that satisfies (1) and (2) in Equation 5.7.

Proof Firstly, the set $B = \text{co}\{\boldsymbol{\zeta}^\dagger : \boldsymbol{\zeta} \in \partial G(\mathbf{q})\}G(\mathbf{q})$ exists if and only if $\boldsymbol{\zeta}^\dagger$ exists for all $\boldsymbol{\zeta} \in \partial G(\mathbf{q})$. $\boldsymbol{\zeta}^\dagger$ exists if and only if rows of $\boldsymbol{\zeta}$ are non-zero and linearly independent. Using the exact same argument as in Lemma 5.1, the existence of B is assured and (1) is satisfied.

Secondly, because $\partial(d_i - d_j)(\mathbf{q})$ is convex and compact, there is a cone of directions H_j where $\left\langle \boldsymbol{\zeta}_j, \boldsymbol{\sigma}_j \right\rangle > 0$ for all $\boldsymbol{\sigma}_j \in H_j$ and $\boldsymbol{\zeta}_j \in \partial(d_i - d_j)(\mathbf{q})$. Similarly, there is a cone of directions H_k where $\langle \boldsymbol{\zeta}_k, \boldsymbol{\sigma}_k \rangle > 0$ for all $\boldsymbol{\sigma}_k \in H_k$ and $\boldsymbol{\zeta}_k \in \partial(d_i - d_k)(\mathbf{q})$. Now, we must have $H_j \cap H_k \neq \varnothing$, since otherwise we could find linearly dependent elements of $\partial(d_i - d_j)(\mathbf{q})$ and $\partial(d_i - d_k)(\mathbf{q})$—a contradiction. By definition, the set B is contained in $H_j \cap H_k$, so letting $\mathbf{b}$ be an arbitrary member of B satisfies (2) and the lemma follows. Figure 5.3(b) illustrates a vector $\mathbf{b}$ that satisfies Equations 5.7. ■

Lemmas 5.1 and 5.2 ensure the existence of vectors $\mathbf{a}$ and $\mathbf{b}$, ensuring that the control law of Equation 5.5 is valid. However, existence is not enough to ensure that the control law can be implemented. The sets $\partial(d_i - d_j)(\mathbf{q})$ etc. may have an infinite number of points, so we need to ensure that vectors $\mathbf{a}$ and $\mathbf{b}$ can be found by considering finite subsets of $\partial(d_i - d_j)(\mathbf{q})$ and $\partial(d_i - d_k)(\mathbf{q})$, ensuring a computer can find solutions quickly and efficiently. The proofs of Lemmas 5.1 and 5.2 provide the answer.

In Lemma 5.1, the vector $\mathbf{a}$ is the normal to a hyperplane through the origin and points $\boldsymbol{\sigma}_j$ and $\boldsymbol{\sigma}_k$ in the boundary of the sets $(d_i - d_j)(\mathbf{q})\partial(d_i - d_j)(\mathbf{q})$ and $(d_i - d_k)(\mathbf{q})\partial(d_i - d_k)(\mathbf{q})$ respectively such that Equation 5.6 is satisfied. Now, we can also see that $\boldsymbol{\sigma}_j$ and $\boldsymbol{\sigma}_k$ need only be members of the support points (vertices) of their corresponding sets in order to satisfy Equation 5.6. Considering the subset of A constructed from



(A) (B)

FIGURE 5.3: ILLUSTRATION OF THE PROOFS OF LEMMAS 5.1 AND 5.2. (A) THE VECTOR $\mathbf{a} \in A$ DESCRIBES A HYPERPLANE THROUGH THE ORIGIN AND TWO POINTS $\boldsymbol{\sigma}_j \in \partial(d_i - d_j)(\mathbf{q})$ AND $\boldsymbol{\sigma}_k \in \partial(d_i - d_k)(\mathbf{q})$ SUCH THAT $\partial(d_i - d_j)(\mathbf{q})$ AND $\partial(d_i - d_j)(\mathbf{q})$ LIE ENTIRELY ON ONE SIDE OF THE PLANE. (B) THE CONE H_j (RESP. H_k) WHERE $\left\langle \boldsymbol{\zeta}_j, \boldsymbol{\sigma}_j \right\rangle > 0$ FOR ALL $\boldsymbol{\sigma}_j \in H_j$ AND $\boldsymbol{\zeta}_j \in \partial(d_i - d_j)(\mathbf{q})$ (RESP. $\langle \boldsymbol{\zeta}_k, \boldsymbol{\sigma}_k \rangle > 0$ FOR ALL $\boldsymbol{\sigma}_k \in H_k$ AND $\boldsymbol{\zeta}_k \in \partial(d_i - d_k)(\mathbf{q})$) AND A VECTOR $\mathbf{b} \in B \subseteq H_j \cap H_k$. REFER TO FIGURE 5.2 TO SEE THAT $\mathbf{a} \in A$ AND $\mathbf{b} \in B$.

the vertices of $\partial(d_i - d_j)(\mathbf{q})$ of $\partial(d_i - d_k)(\mathbf{q})$ is sufficient to find a suitable vector $\mathbf{a}$. Similarly, since the vector $\mathbf{b}$ can be any member of the set B, it is sufficient to consider only those elements of B constructed from vertices of $\partial(d_i - d_j)(\mathbf{q})$ and $\partial(d_i - d_k)(\mathbf{q})$. In order maximise the amount of correction, it is suitable to select the vector $\mathbf{b}$ that results in the each element of $G(\mathbf{q})$ decreasing the fastest.

5.4 Localisation of NCRIT+NGVG Nodes

In this section we present a control law that allows $\mathcal{A}$ to localise an invertible four-equidistant face $\mathcal{IF}_{ijkl}$. We recall that $\mathcal{IF}_{ijkl}$ is defined as the set of configurations $\mathcal{IF}_{ijk} \cap \mathcal{IF}_{ijl} \cap \mathcal{IF}_{jkl}$ and is assumed to be one-dimensional. This is also equivalently described as the zeros of the vector valued function $G(\mathbf{q})$

$$G(\mathbf{q}) = \begin{bmatrix} (d_i - d_j)(\mathbf{q}) \\ (d_i - d_k)(\mathbf{q}) \\ (d_i - d_l)(\mathbf{q}) \end{bmatrix}$$

Since the $\mathcal{IF}_{ijkl}$ is one-dimensional, the follow term in Equation 5.5 is not needed. The control law used to follow localise $\mathcal{IF}_{ijkl}$ is thus

$$\dot{\mathbf{q}} = \beta \mathbf{b} \tag{5.8}$$

where the vector $\mathbf{b}$ satisfies

$$\begin{aligned} &1.\ \mathbf{b} \in \text{co}\{\zeta^{\dagger} : \zeta \in \partial G(\mathbf{q})\} G(\mathbf{q}) \\ &2.\ \langle \mathbf{b}, \sigma_z \rangle \geq 0 \text{ for all } \sigma_z \in (d_i - d_z)(\mathbf{q}) \partial (d_i - d_z)(\mathbf{q}), \text{ for all } z \in \{j, \cdots, l\} \end{aligned} \tag{5.9}$$

The assumption that $\mathcal{IF}_{ijkl}$ is one dimensional is equivalent to assuming all elements of $(d_i - d_l)(\mathbf{q})$ are linearly independent from $(d_i - d_j)(\mathbf{q}) + (d_i - d_k)(\mathbf{q})$ (recall from Chapter 4). Since $(d_i - d_j)(\mathbf{q})$ and $(d_i - d_k)(\mathbf{q})$ are linearly independent, we know $\mathbf{0} \notin (d_i - d_j)(\mathbf{q}) + (d_i - d_k)(\mathbf{q})$. We can thus use exactly the same argument as in Lemma 5.2 to show that a vector $\mathbf{b}$ exists that satisfies Equations 5.9.

5.5 Following Diameter Extrema Edges

In this section we present a control law that allows $\mathcal{A}$ to follow a diameter extrema edge $\mathcal{D}^{\vartheta}_{ij}$. We recall that $\mathcal{D}^{\vartheta}_{ij}$ is defined as the one-dimensional set of configurations where the witness vectors $\mathbf{ab}_i$ and $\mathbf{ab}_j$ have the same angle between themselves and the orientation vector $[\cos(\theta + \vartheta), \sin(\theta + \vartheta)]^T$. This is also equivalently described as the zeros of the vector valued function $G(\mathbf{q})$

$$G(\mathbf{q}) = \begin{bmatrix} (d_i - d_j)(\mathbf{q}) \\ f_{ij}(\mathbf{q}) \end{bmatrix}$$

Where $f_{ij}(\mathbf{q}) = \tan^{-1}\left(\frac{ab_{iy}(\mathbf{q})}{ab_{ix}(\mathbf{q})}\right) + \tan^{-1}\left(\frac{ab_{jy}(\mathbf{q})}{ab_{jx}(\mathbf{q})}\right) - 2\theta - 2\vartheta$ as described in Chapter 4. The control law developed for following $\mathcal{D}^{\vartheta}_{ij}$ is the same as for following $\mathcal{IF}_{ijk}$

$$\dot{x} = \alpha \mathbf{a} + \beta \mathbf{b} \tag{5.10}$$

but with a special caveat. In the case for following $\mathcal{IF}_{ijk}$, the proofs of Lemmas 5.1 and 5.2 were based partially on the fact that $\mathbf{0} \notin (d_i - d_z)(\mathbf{q})$ for all $n \in \{j, \cdots, n\}$, thus ensuring that the sets A and B exist. Recalling the proof of Lemma 4.13 that establishes $\mathcal{D}^{\vartheta}_{ij}$ is one-dimensional—which is based on the calculation of $\partial f_{ij}(\mathbf{q})$ in Appendix C—we know that $\mathbf{0}$ can in fact be an element of $\partial f_{ij}(\mathbf{q})$ (see Case 9

in Appendix C). To accommodate this, we simply remove the element **0** from consideration. In Section 5.7.2, we will see this has no effect on the stability of the control law of Equation 5.10.

Denote the two rows of an element $\zeta \in \partial G(\mathbf{q})$ with $\zeta_1 \in \partial(d_i - d_j)(\mathbf{q})$ and $\zeta_2 \in \partial f_{ij}(\mathbf{q})$. In Equation 5.10, the vector **a** satisfies

$$\begin{aligned} &1.\ \mathbf{a} \in \text{co}\{\text{Null}\,(\zeta) : \zeta \in \partial G(\mathbf{q}), \zeta_2 \neq \mathbf{0}\} \\ &2.\ \langle \mathbf{a}, \sigma_1 \rangle \leq 0 \text{ for all } \sigma_1 \in (d_i - d_j)(\mathbf{q})\partial(d_i - d_j)(\mathbf{q}) \\ &3.\ \langle \mathbf{a}, \sigma_2 \rangle \leq 0 \text{ for all } \sigma_2 \in f_{ij}(\mathbf{q})\partial f_{ij}(\mathbf{q}) \end{aligned} \tag{5.11}$$

and the vector **b** satisfies

$$\begin{aligned} &1.\ \mathbf{b} \in \text{co}\{\zeta^\dagger : \zeta \in \partial G(\mathbf{q}), \zeta_2 \neq \mathbf{0}\}G(\mathbf{q}) \\ &2.\ \langle \mathbf{b}, \sigma_1 \rangle \geq 0 \text{ for all } \sigma_1 \in (d_i - d_j)(\mathbf{q})\partial(d_i - d_j)(\mathbf{q}) \\ &3.\ \langle \mathbf{b}, \sigma_2 \rangle \geq 0 \text{ for all } \sigma_2 \in f_{ij}(\mathbf{q})\partial f_{ij}(\mathbf{q}) \end{aligned} \tag{5.12}$$

Notice that we only use elements ζ with a second row that is nonzero $\zeta_2 \neq \mathbf{0}$. With this restriction, elements of $\partial(d_i - d_j)(\mathbf{q})$ and $\partial f_{ij}(\mathbf{q}) \setminus \mathbf{0}$ are nonzero and linearly independent so A and B exist, and the vectors **a** and **b** exist by the exact same arguments as in Lemmas 5.1 and 5.2.

5.6 Following Critical Set Edges

In this section we present a control law that allows $\mathcal{A}$ to follow a critical edge K_{ij}^{ϑ}. We recall that K_{ij}^{ϑ} is defined as the subset of critical configurations where the distance $D(\mathbf{q}) = d$ remains constant. This is also equivalently described as the zeros of the vector valued function $G(\mathbf{q})$

$$G(\mathbf{q}) = \begin{bmatrix} (d_i - d_j)(\mathbf{q}) \\ f_j(\mathbf{q}) \end{bmatrix}$$

Where $f_{ij}(\mathbf{q}) = d_j(\mathbf{q}) - d$ as presented in Chapter 4. The control law developed for following K_{ij}^{ϑ} is the same as for following $\mathcal{D}_{ij}^{\vartheta}$

$$\dot{x} = \alpha\mathbf{a} + \beta\mathbf{b} \tag{5.13}$$

where the vector **a** satisfies

$$\begin{aligned}
&1.\ \mathbf{a} \in \text{co}\{\text{Null}\,(\zeta) : \zeta \in \partial G(\mathbf{q})\} \\
&2.\ \langle \mathbf{a}, \sigma_1 \rangle \leq 0 \text{ for all } \sigma_1 \in (d_i - d_j)(\mathbf{q})\partial(d_i - d_j)(\mathbf{q}) \\
&3.\ \langle \mathbf{a}, \sigma_2 \rangle \leq 0 \text{ for all } \sigma_2 \in f_j(\mathbf{q})\partial f_j(\mathbf{q})
\end{aligned} \tag{5.14}$$

and the vector **b** satisfies

$$\begin{aligned}
&1.\ \mathbf{b} \in \text{co}\{\zeta^\dagger : \zeta \in \partial G(\mathbf{q})\}G(\mathbf{q}) \\
&2.\ \langle \mathbf{b}, \sigma_1 \rangle \geq 0 \text{ for all } \sigma_1 \in (d_i - d_j)(\mathbf{q})\partial(d_i - d_j)(\mathbf{q}) \\
&3.\ \langle \mathbf{b}, \sigma_2 \rangle \geq 0 \text{ for all } \sigma_2 \in f_j(\mathbf{q})\partial f_j(\mathbf{q})
\end{aligned} \tag{5.15}$$

In this case, $\mathbf{0} \notin \partial f_j(\mathbf{q})$ so elements of $\partial(d_i - d_j)(\mathbf{q})$ and ∂f_j are nonzero and linearly independent so A and B exist, and the vectors **a** and **b** exist by the exact same arguments as in Lemmas 5.1 and 5.2.

5.7 Stability

In this section, we examine the stability of the four control laws developed in Sections 5.3, 5.4, 5.5, and 5.6. Section 5.7.1 deals with the first two cases, and Sections 5.7.2 and 5.7.3 deal with the remaining cases respectively.

5.7.1 Stability of Following $\mathcal{IF}_{ijk}$ and Localising $\mathcal{IF}_{ijkl}$

In this section, we examine the stability of following three-equidistant edges $\mathcal{IF}_{ijk}$ and localising three-equidistant nodes $\mathcal{IF}_{ijkl}$, both of which are zeros of a function $G(\mathbf{q})$ of the form

$$G(\mathbf{q}) = \begin{bmatrix} (d_i - d_j)(\mathbf{q}) \\ \vdots \\ (d_i - d_n)(\mathbf{q}) \end{bmatrix} \tag{5.16}$$

Let us consider the following positive definite Lyapunov candidate function $V(\mathbf{q})$

$$\begin{aligned} V(\mathbf{q}) &= \frac{1}{2}G(\mathbf{q})^T G(\mathbf{q}) \\ &= \frac{1}{2}((d_i - d_j)(\mathbf{q})^2 + \cdots + (d_i - d_n)(\mathbf{q})^2) \end{aligned}$$

which is a measure of the 'distance' from the NCRIT+NGVG. The generalised gradient of $V(\mathbf{q})$ is calculated using the calculus rules in Chapter 3 and is given by

$$\partial V(\mathbf{q}) = (d_i - d_j)(\mathbf{q})\partial(d_i - d_j)(\mathbf{q}) + \cdots + (d_i - d_n)(\mathbf{q})\partial(d_i - d_n)(\mathbf{q})$$

We recall equation 5.2 which tells us in order to show stability, we need to show that $\langle \zeta, \dot{\mathbf{q}} \rangle \leq 0$ for all $\zeta \in \partial V(\mathbf{q})$ and for all $\mathbf{q} \in C$.

Theorem 5.3 The control law $\dot{\mathbf{q}} = \alpha\mathbf{a} + \beta\mathbf{b}$ is stable for $G(\mathbf{q})$ of the form of Equation 5.16

Proof Take an arbitrary element $G^T\zeta \in \partial V(\mathbf{q})$. $G^T\zeta$ can be decomposed into a vector sum of the form $(d_i - d_j)(\mathbf{q})\sigma_j + \cdots + (d_i - d_n)(\mathbf{q})\sigma_n$, where $\sigma_z \in \partial(d_i - d_z)(\mathbf{q})$ for all $z \in \{j, \cdots, n\}$. We consider the terms $\alpha\mathbf{a}$ and $\beta\mathbf{b}$ separately.

($\alpha\mathbf{a}$). According to Equation 5.6, $\langle \mathbf{a}, \sigma_z \rangle \leq 0$ for all $\sigma_z \in (d_i - d_z)(\mathbf{q})\partial(d_i - d_z)(\mathbf{q})$, so as long as $\alpha \geq 0$, $\langle \zeta, \alpha\mathbf{a} \rangle \leq 0$ for all $\zeta \in \partial V(\mathbf{q})$.

($\beta\mathbf{b}$). According to Equation 5.7, $\langle \mathbf{b}, \sigma_z \rangle \geq 0$ for all $\sigma_z \in (d_i - d_z)(\mathbf{q})\partial(d_i - d_z)(\mathbf{q})$, so as long as $\beta \leq 0$, $\langle \zeta, \beta\mathbf{b} \rangle \leq 0$ for all $\zeta \in \partial V(\mathbf{q})$.

Since $\langle \zeta, \alpha\mathbf{a} \rangle \leq 0$ and $\langle \zeta, \beta\mathbf{b} \rangle \leq 0$, we have $\langle \zeta, \alpha\mathbf{a} + \beta\mathbf{b} \rangle \leq 0$ for all $\zeta \in \partial V(\mathbf{q})$ and for all $\mathbf{q} \in C$, so the lemma follows. ■

5.7.2 Stability of Following $\mathcal{D}_{ij}^{\vartheta}$

In this section, we examine the stability of following a diameter extrema edge $\mathcal{D}_{ij}^{\vartheta}$, which is equivalent to zeros of

$$G(\mathbf{q}) = \begin{bmatrix} (d_i - d_j)(\mathbf{q}) \\ f_{ij}(\mathbf{q}) \end{bmatrix} \tag{5.17}$$

Let us consider the following positive definite Lyapunov candidate function $V(\mathbf{q})$

$$\begin{aligned} V(\mathbf{q}) &= \frac{1}{2}G(\mathbf{q})^T G(\mathbf{q}) \\ &= \frac{1}{2}((d_i - d_j)(\mathbf{q})^2 + f_{ij}(\mathbf{q})^2) \end{aligned}$$

which is a measure of the 'distance' from the diameter extrema edge $\mathcal{D}_{ij}^{\vartheta}$. The generalised gradient of $V(\mathbf{q})$ is calculated using the calculus rules in Chapter 3 and is given by

$$\partial V(\mathbf{q}) = (d_i - d_j)(\mathbf{q})\partial(d_i - d_j)(\mathbf{q}) + f_{ij}(\mathbf{q})\partial f_{ij}(\mathbf{q})$$

To show stability, we need to show that $\langle \zeta, \dot{\mathbf{q}} \rangle \leq 0$ for all $\zeta \in \partial V(\mathbf{q})$ and for all $\mathbf{q} \in C$. We recall that in control law of Equation 5.10 we ignored the element $\mathbf{0} \in \partial f_{ij}(\mathbf{q})$. However, the proof of stability requires that we consider all elements of $\partial f_{ij}(\mathbf{q})$, so this is highlighted in the proof as a 'special case'.

Theorem 5.4 The control law $\dot{\mathbf{q}} = \alpha\mathbf{a} + \beta\mathbf{b}$ is stable for $G(\mathbf{q})$ of the form of Equation 5.17.

Proof Take an arbitrary element $G^T\zeta \in \partial V(\mathbf{q})$. $G^T\zeta$ can be decomposed into a vector sum of the form $(d_i - d_j)(\mathbf{q})\sigma_1 + f_{ij}(\mathbf{q})\sigma_2$ or $(d_i - d_j)(\mathbf{q})\sigma_1 + \mathbf{0}$ (special case), where $\sigma_1 \in \partial(d_i - d_j)(\mathbf{q})$ and $\sigma_2 \in \partial f_{ij}(\mathbf{q})\backslash\mathbf{0}$. We consider the terms $\alpha\mathbf{a}$ and $\beta\mathbf{b}$ separately.

($\alpha\mathbf{a}$). According to Equations 5.11, $\langle \mathbf{a}, \sigma_1 \rangle \leq 0$ for all $\sigma_1 \in (d_i - d_j)(\mathbf{q})\partial(d_i - d_j)(\mathbf{q})$ and $\langle \mathbf{a}, \sigma_2 \rangle \leq 0$ for all $\sigma_2 \in f_{ij}(\mathbf{q})\partial f_{ij}(\mathbf{q})\backslash\mathbf{0}$ or $\sigma_2 = \mathbf{0}$ (special case). So, as long as $\alpha \geq 0$, $\langle \zeta, \alpha\mathbf{a} \rangle \leq 0$ for all $\zeta \in \partial V(\mathbf{q})$.

($\beta\mathbf{b}$). According to Equations 5.12, $\langle \mathbf{b}, \sigma_1 \rangle \geq 0$ for all $\sigma_1 \in (d_i - d_j)(\mathbf{q})\partial(d_i - d_j)(\mathbf{q})$ and $\langle \mathbf{b}, \sigma_2 \rangle \geq 0$ for all $\sigma_2 \in f_{ij}(\mathbf{q})\partial f_{ij}(\mathbf{q})\backslash\mathbf{0}$ or $\sigma_2 = \mathbf{0}$ (special case). So, as long as $\beta \leq 0$, $\langle \zeta, \beta\mathbf{b} \rangle \leq 0$ for all $\zeta \in \partial V(\mathbf{q})$.

Since $\langle \zeta, \alpha\mathbf{a} \rangle \leq 0$ and $\langle \zeta, \beta\mathbf{b} \rangle \leq 0$, we have $\langle \zeta, \alpha\mathbf{a} + \beta\mathbf{b} \rangle \leq 0$ for all $\zeta \in \partial V(\mathbf{q})$, for all $\mathbf{q} \in C$. ■

5.7.3 Stability of Following K_{ij}^{ϑ}

In this section, we examine the stability of following a diameter extrema edge K_{ij}^{ϑ}, which is equivalent to zeros of

$$G(\mathbf{q}) = \begin{bmatrix} (d_i - d_j)(\mathbf{q}) \\ f_j(\mathbf{q}) \end{bmatrix} \tag{5.18}$$

Let us consider the following positive definite Lyapunov candidate function $V(\mathbf{q})$

$$\begin{aligned} V(\mathbf{q}) &= \frac{1}{2} G(\mathbf{q})^T G(\mathbf{q}) \\ &= \frac{1}{2}((d_i - d_j)(\mathbf{q})^2 + f_j(\mathbf{q})^2) \end{aligned}$$

which is a measure of the 'distance' from the diameter extrema edge $\mathcal{D}_{ij}^{\vartheta}$. The generalised gradient of $V(\mathbf{q})$ is calculated using the calculus rules in Chapter 3 and is given by

$$\partial V(\mathbf{q}) = (d_i - d_j)(\mathbf{q})\partial(d_i - d_j)(\mathbf{q}) + f_j(\mathbf{q})\partial f_j(\mathbf{q})$$

To show stability, we need to show that $\langle \zeta, \dot{\mathbf{q}} \rangle \leq 0$ for all $\zeta \in \partial V(\mathbf{q})$ and for all $\mathbf{q} \in C$.

Theorem 5.5 The control law $\dot{\mathbf{q}} = \alpha\mathbf{a} + \beta\mathbf{b}$ is stable for $G(\mathbf{q})$ the form of Equation 5.18.

Proof Take an arbitrary element $G^T\zeta \in \partial V(\mathbf{q})$. $G^T\zeta$ can be decomposed into a vector sum of the form $(d_i - d_j)(\mathbf{q})\sigma_1 + f_j(\mathbf{q})\sigma_2$ where $\sigma_1 \in \partial(d_i - d_j)(\mathbf{q})$ and $\sigma_2 \in \partial f_j(\mathbf{q})$. We consider the terms $\alpha\mathbf{a}$ and $\beta\mathbf{b}$ separately.

($\alpha\mathbf{a}$). According to Equations 5.14, $\langle \mathbf{a}, \sigma_1 \rangle \leq 0$ for all $\sigma_1 \in (d_i - d_j)(\mathbf{q})\partial(d_i - d_j)(\mathbf{q})$ and $\langle \mathbf{a}, \sigma_2 \rangle \leq 0$ for all $\sigma_2 \in f_j(\mathbf{q})\partial f_j(\mathbf{q})$. So, as long as $\alpha \geq 0$, $\langle \zeta, \alpha\mathbf{a} \rangle \leq 0$ for all $\zeta \in \partial V(\mathbf{q})$.

($\beta\mathbf{b}$). According to Equations 5.15, $\langle \mathbf{b}, \sigma_1 \rangle \geq 0$ for all $\sigma_1 \in (d_i - d_j)(\mathbf{q})\partial(d_i - d_j)(\mathbf{q})$ and $\langle \mathbf{b}, \sigma_2 \rangle \geq 0$ for all $\sigma_2 \in f_j(\mathbf{q})\partial f_j(\mathbf{q})$. So, as long as $\beta \leq 0$, $\langle \zeta, \beta\mathbf{b} \rangle \leq 0$ for all $\zeta \in \partial V(\mathbf{q})$.

Since $\langle \zeta, \alpha\mathbf{a} \rangle \leq 0$ and $\langle \zeta, \beta\mathbf{b} \rangle \leq 0$, we have $\langle \zeta, \alpha\mathbf{a} + \beta\mathbf{b} \rangle \leq 0$ for all $\zeta \in \partial V(\mathbf{q})$, for all $\mathbf{q} \in C$. ■

5.8 Summary

In this chapter, we presented novel nonsmooth control laws for following the invertible three-equidistant faces $\mathcal{IF}_{ijk}$, localising the invertible four-equidistant faces $\mathcal{IF}_{ijkl}$, following the diameter extrema edges $\mathcal{D}_{ij}^{\vartheta}$, and for following a critical set edge K_{ij}^{ϑ}. These control laws were based loosely on smooth control laws developed in the context of the point GVG, in that the control term $\dot{\mathbf{q}}$ had a 'follow' term $\alpha\mathbf{a}$ and a 'correction' term $\beta\mathbf{b}$, but the analysis was notably different. In the smooth case, the terms $\alpha\mathbf{a}$ and $\beta\mathbf{b}$ are immediately available directly from the definitions of nullspace and Penrose pseudo-inverse. In the nonsmooth case, we must check all vertex pairs of the appropriate generalised gradients in order to select the vectors $\mathbf{a}$ and $\mathbf{b}$.

Equations 5.6 and 5.7 for selecting appropriate vectors **a** and **b** to follow $\mathcal{IF}_{ijk}$ (and the associated equations for localising $\mathcal{IF}_{ijkl}$ and following $\mathcal{D}_{ij}^{\vartheta}$ and K_{ij}^{ϑ}) come directly from the definition of stability in Equation 5.2. Given the Lyapunov function candidate $V(\mathbf{q})$, the controls laws were developed so that Equation 5.2 was satisfied. Using this technique, we successfully developed three stable control laws that allow a real robot to fully and completely explore the NCRIT+NGVG developed in Chapter 4.

§6

"It is often said that experiments must be made without preconceived ideas. That is impossible. Not only would it make all experiments barren, but that would be attempted which could not be done."

Henri Poincare

Simulations and Experiments

In this chapter we describe the simulations and experiments used to verify the utility of the techniques developed in the previous chapters. In Section 6.1 we describe a graph search algorithm developed to explore the NCRIT+NGVG and in Section 6.2 we describe a simulator developed to test the algorithm—and the results of those tests. Finally, in Section 6.3 we describe the results of some experimental trials with a real robot.

6.1 Graph Search Algorithm

In Chapter 4, we described a mathematical structure NCRIT+NGVG that represents the connectivity of free space, but have not yet described how to explore this structure. Since the NCRIT+NGVG is equivalently expressed as a graph ($\mathcal{IF}_{ijkl}$ and $\mathcal{D}^{\vartheta}_{ijk}$ are nodes and $\mathcal{IF}_{ijk}$ and $\mathcal{D}^{\vartheta}_{ij}$ are arcs), any number of graph search algorithms could be used. The notable A* algorithm [25], which is guaranteed to return a shortest length path if one exists, is commonly used in robotics, since shortest length paths tend to minimise energy consumption and execution time. However, when used for sensor-based exploration, A* has several drawbacks. First of all, A* is based on two weights $g(\mathbf{q})$ and $h(\mathbf{q})$ [38] assigned to each node $\mathbf{q}$, which are used to decide which arc to examine next. $g(\mathbf{q})$ represents the length of the path from the start node $\mathbf{q}_s$ to $\mathbf{q}$ in the current search and $h(\mathbf{q})$ represents a *heuristic* estimate of the path length between $\mathbf{q}$ and the goal node $\mathbf{q}_g$. If $h(\mathbf{q})$ is based on a poor heuristic, the algorithm tends to explore more of the graph than necessary. In

```
EQUISEARCH()
 1  from q_s, climb onto the nearest IF_ijk
 2  PUSH(q, equiOPEN)
 3  while SIZE(equiOPEN) > 0
 4  do node ← HEAD(equiOPEN)
 5     move to node
 6     EXPLORENODE(node)
 7     while SIZE(diamOPEN) > 0
 8     do node ← TAIL(diamOPEN)
 9        move to node
10        EXPLORENODE(node)
```

method	method operation
PUSH(*node*,*list*)	push *node* onto tail of *list*.
REMOVE(*node*,*list*)	remove *node* from *list*.
HEAD(*list*)	remove and return the head of *list*.
TAIL(*list*)	remove and return the tail of *list*.

FIGURE 6.1: PSEUDOCODE FOR EQUISEARCH() AND DESCRIPTIONS OF BASIC METHODS USED IN THE PSEUDOCODE.

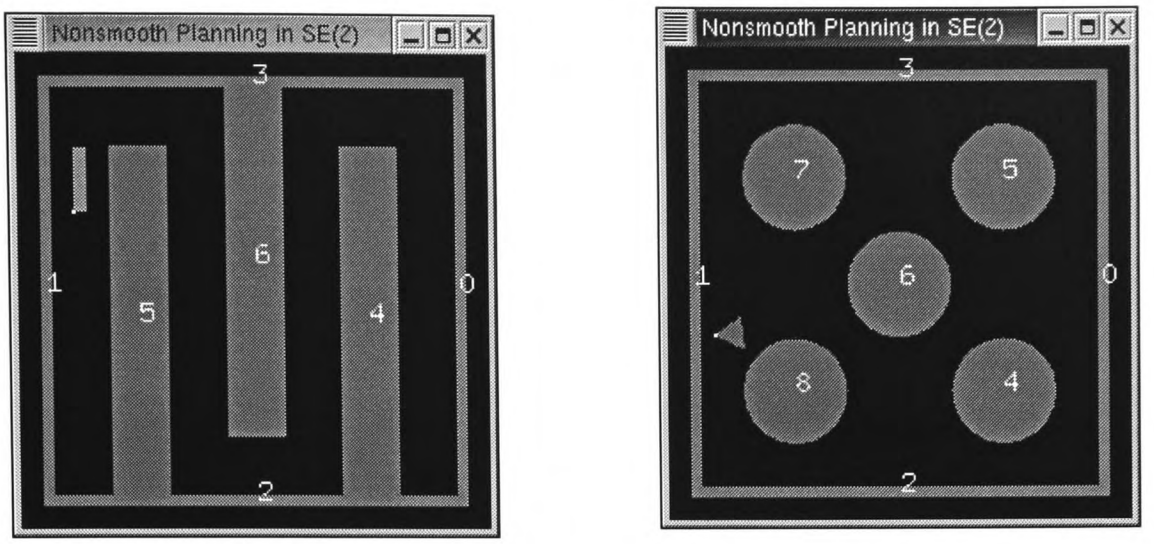
sensor-based motion planning, where the graph is unknown at the start of exploration, very few cues can be used to generate a sensible heuristic weight $h(\mathbf{q})$, so the robot tends to explore inefficiently. Secondly, the length-optimality of A* refers only to the length of the *final* path generated. While searching for this length-optimal path, A* spends a great deal of time returning to nodes to look for a shorter path. When implemented by a real robot $\mathcal{A}$, this adds to the total length of the path $\mathcal{A}$ must follow while exploring. So, in addition to the A* solution, we have developed an algorithm geared toward the unique graph structure of the NCRIT+NGVG that generates sensible, if not length-optimal, paths.

Our algorithm, which we have called EQUISEARCH(), searches the graph in a modified depth-first manner. Starting at the start node $\mathbf{q}_s$, the robot first climbs onto an invertible three-equidistant face $\mathcal{IF}_{ijk}$ and then proceeds to *fully* explore it. As it does so, it keeps track of newly discovered four-equidistant nodes $\mathcal{IF}_{ijkl}$ and diameter extrema nodes $\mathcal{D}^{\vartheta}_{ijk}$ that have unexplored three-equidistant faces $\mathcal{IF}_{ijk}$ emanating from them in the list *equiOPEN* and new $\mathcal{IF}_{ijkl}$s and $\mathcal{D}^{\vartheta}_{ijk}$s that have unexplored diameter extrema edges $\mathcal{D}^{\vartheta}_{ij}$ emanating from them in the list *diamOPEN*. After each three-equidistant face $\mathcal{IF}_{ijk}$ is explored, we proceed to pop and explore *every* $\mathcal{D}^{\vartheta}_{ijk}$ in *diamOPEN*, in turn, before starting to explore another node in *equiOPEN*. In a sense, we are examining the three-equidistant faces $\mathcal{IF}_{ijk}$ in succession. The search is finished when *equiOPEN* and *diamOPEN* are both empty. The completeness of EQUISEARCH() is assured since it is simply an exhaustive depth first search. Figure 6.1 gives pseudocode for EQUISEARCH() and Figure 6.2 gives pseudocode for two helper functions EXPLORENODE() and EXPLOREARC(). The table in Figure 6.1 gives a few elementary list operations used in our pseudocode.

```
EXPLORENODE(node)
 1  for each of node's unexplored IF_ijks
 2  do for each direction dir of IF_ijk
 3      do EXPLOREARC(IF_ijk, dir)
 4          return to node
 5  PUSH(node, equiCLOSED)
 6  for each of node's unexplored D^ϑ_ijs
 7  do for each direction dir of D^ϑ_ij
 8      do EXPLOREARC(D^ϑ_ij, dir)
 9          return to node
10  PUSH(node, diamCLOSED)

EXPLOREARC(arc, dir)
 1  while no collision
 2  do move along arc in direction dir
 3     if a IF_ijkl or D^ϑ_ijk is discovered
 4       then node ← IF_ijkl or D^ϑ_ijk
 5            update node's visited arc information
 6            if all node's IF_ijks have been explored
 7              then REMOVE(node, equiOPEN)
 8                   PUSH(node, equiCLOSED)
 9              else PUSH(node, equiOPEN)
10            if all node's D^ϑ_ijs have been explored
11              then REMOVE(node, diamOPEN)
12                   PUSH(node, diamCLOSED)
13              else PUSH(node, diamOPEN)
```

FIGURE 6.2: PSEUDOCODE FOR HELPER FUNCTIONS EXPLORENODE() AND EXPLOREARC().



(A) (B)

FIGURE 6.3: TWO SNAPSHOTS OF OUR SIMULATOR IN ACTION. (A) EXPERIMENTAL RESULTS USING THIS ENVIRONMENT ARE IN FIGURE 6.4. (B) EXPERIMENTAL RESULTS USING THIS ENVIRONMENT ARE IN FIGURE 6.5.

6.2 Simulations

A simulator was written to explore the NCRIT+NGVG using the nonsmooth control laws of Chapter 5 and to examine the behaviour EQUISEARCH() and A* in a highly controlled manner. Using idealised models of the robot, environment and control system, the simulator allowed us to work the kinks out of code that would eventually be used to control a real robot. The simulator and planning code were written as distinct modules so that when the time came, the simulator module could be replaced by the control code for a real robot with a minimum of effort. Two snapshots of the simulator exploring two different environments shown are shown in 6.3.

6.2.1 Results

The results of several representative experiments are shown in Figures 6.4 and 6.5. In Figure 6.4, the environment is the same as in Figure 6.3(a) but each of the four cases uses a different convex polygonal $\mathcal{A}$. In all cases, the three-equidistant faces $\mathcal{IF}_{ijk}$ are in the corners and ends of the channel and are all similarly shaped. One key difference between Figure 6.4(b) with short rectangular $\mathcal{A}$ and Figure 6.4(c) with a longer rectangular $\mathcal{A}$ is that, in Figure 6.4(c), $\mathcal{A}$ is unable to turn a full 360° in the corners of the channel. In Figure 6.5 the environment is the same as in Figure 6.3(b), but again the two cases use different $\mathcal{A}$s. In this case, notice that the NCRIT+NGVG is more 'net-like' since many more $\mathcal{D}_{ij}^{\vartheta}$s meet up with a given $\mathcal{IF}_{ijk}$.

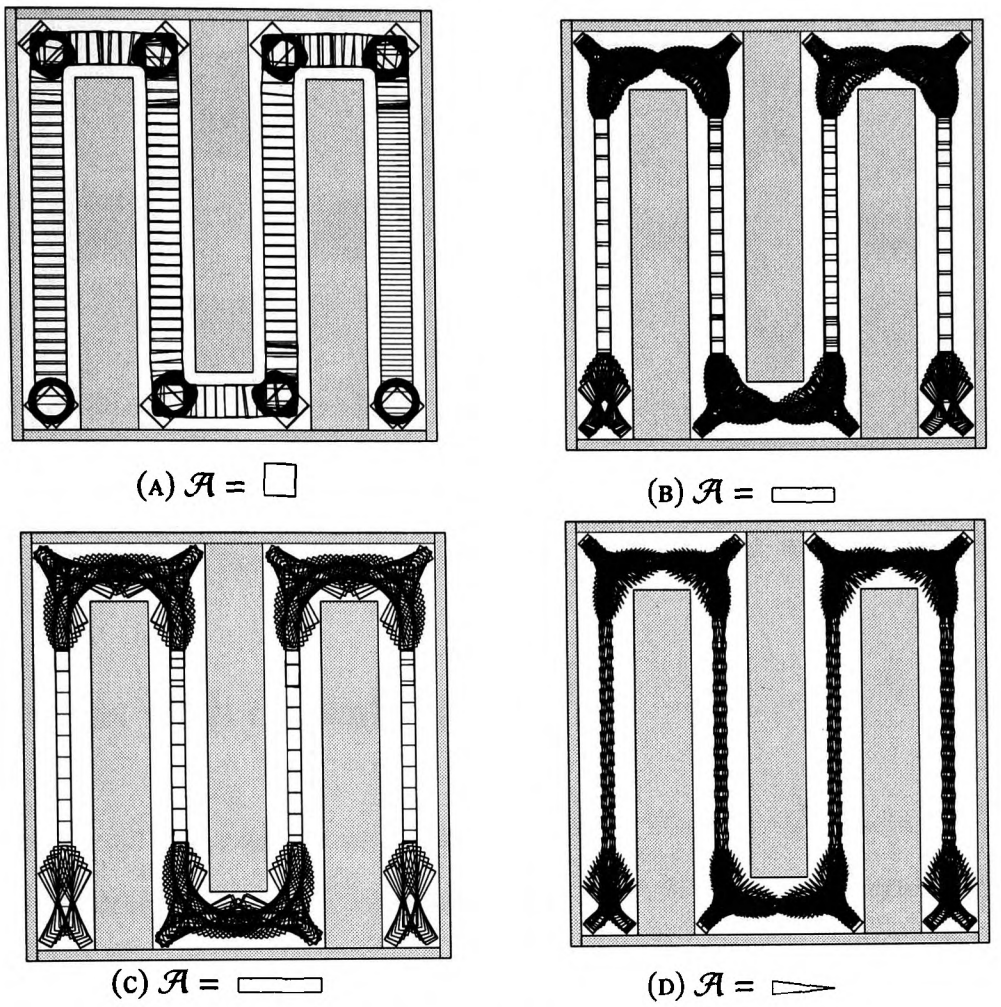
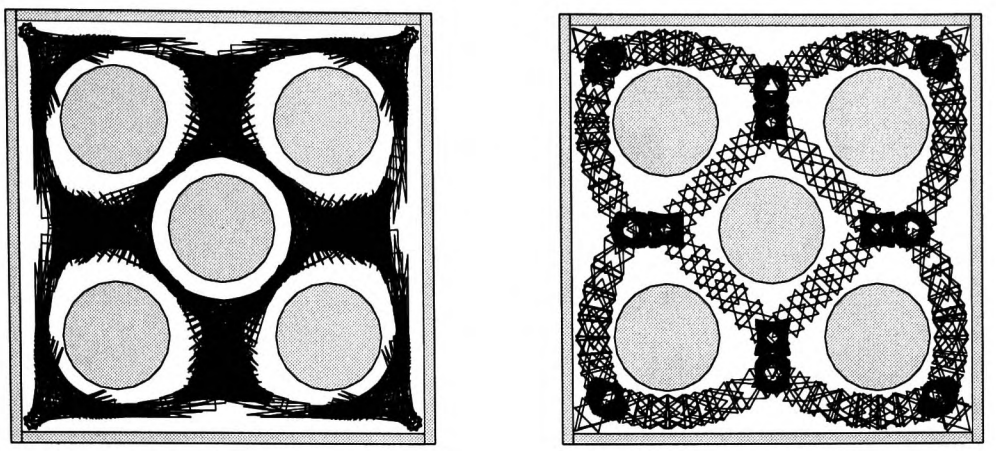


FIGURE 6.4: SEVERAL EXAMPLES OF EXPERIMENTAL TRIALS. IN ALL CASES THE ENVIRONMENT IS THAT IN FIGURE 6.3(A).

6.2.2 Comparison EQUISEARCH to A* Algorithm

We compared EQUISEARCH() to A* one the basis of *total* distance travelled by the robot (both translational and rotational) while exploring NCRIT+NGVG. We first ran EQUISEARCH() to generate a complete roadmap NCRIT+NGVG for several combinations of environment and robot shape, some of which are shown in Figures 6.4 and 6.5. During these searches, as each node ($\mathcal{IF}_{ijkl}$ or $\mathcal{D}^{\vartheta}_{ijk}$) was first created, a note was made of the distance the robot had travelled up to that point. Then, for every node $\mathbf{q} \in$ NCRIT+NGVG, A* was run to see what distance the robot would have travelled while searching for the shortest path from $\mathbf{q}_s$ to $\mathbf{q}$. In all of our trials, A* generated total motions at least three times the length of those generated by EQUISEARCH(). This is substantially longer than paths generated by EQUISEARCH(), and verifies the utility of EQUISEARCH() for exploring the NCRIT+NGVG.



(A) $\mathcal{A} =$ (B) $\mathcal{A} =$

FIGURE 6.5: SEVERAL EXAMPLES OF EXPERIMENTAL TRIALS. IN ALL CASES THE ENVIRONMENTS IS THAT IN FIGURE 6.3(B).

6.3 Experiments with a Real Robot

In order to fully demonstrate that exploration of the NCRIT+NGVG is in fact practically feasible, we need to fully implement our techniques in a *real* robot. Verification by simulation is useful for working the kinks out of the techniques, but is just not sufficient to show conclusively that the contributions of this thesis are useful. This section does this.

The mobile robot used for our experiments is the Acroname PPRK (Palm Pilot Robot Kit) developed and licensed by Carnegie Mellon University [52]. A photo of the PPRK is shown in Figure 6.6(a). The PPRK is a simple robot that accepts RS232 (serial) data to control three omni-wheels, which allow motion transverse to the rolling direction, in an open-loop manner. By controlling the velocity of each wheel individually, the PPRK is able to move holonomically. To obtain closed loop control of the PPRK's velocity, and thus enabling the PPRK to implement the control laws of Chapter 5, we close the control loop with a camera and off-board control computer (see Figure 6.6(b)). The visual-servo loop created is described in Appendix D, so we don't go into any detail in this chapter. Suffice to say we can implement the control laws in Chapter 5 by controlling the PPRK's velocity $\dot{\mathbf{q}}$ at a given moment. The camera system is also used to determine the distance $d_i(\mathbf{q})$ between the PPRK and obstacles in its environment. The fine details of this procedure are also detailed in Appendix D.

6.3.1 Results

The results of three representative experiments are shown in Figures 6.7. In both figures, a snapshot of the environment is shown in (a), a diagram showing the results of a simulation is shown in (b), and a diagram

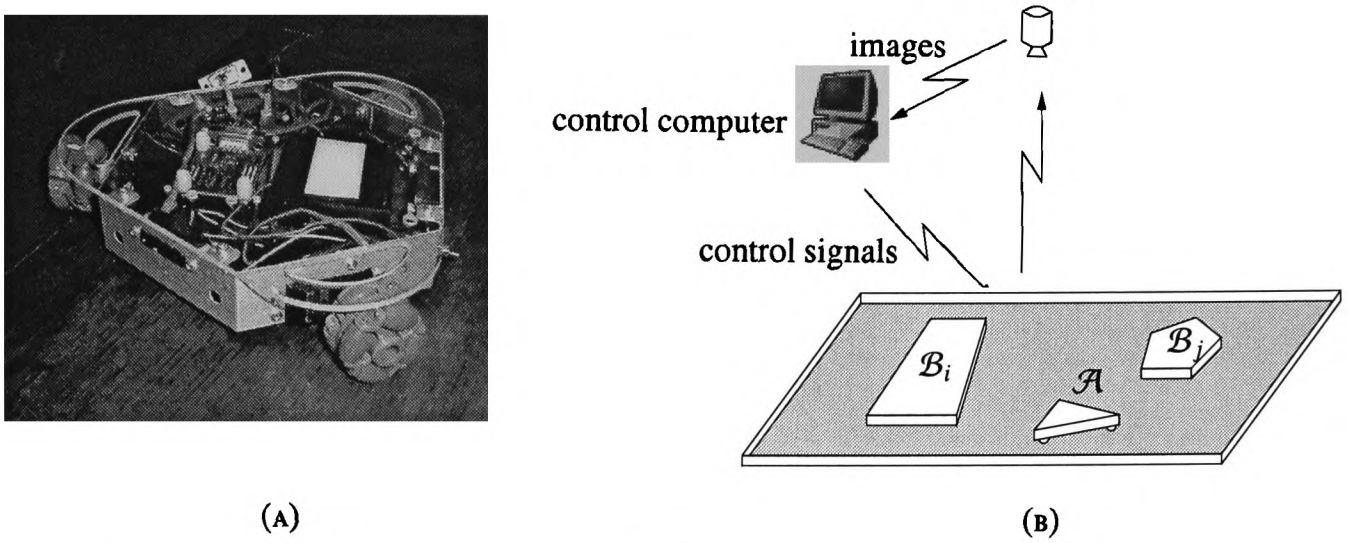


FIGURE 6.6: SCHEMATIC OF THE EXPERIMENTAL APPARATUS. (A) THE ACRONAME PPRK (PALM PILOT ROBOT KIT) ROBOT. (B) THE CLOSED-LOOP CONTROL RELATIONSHIP BETWEEN THE ROBOT $\mathcal{A}$, CAMERA AND CONTROL COMPUTER.

representing the motion of the PPRK during an actual trial is shown in (c).

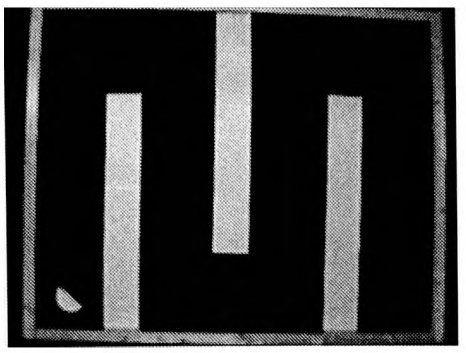
A key observation made during the actual experiments, and one well understood by anyone with even a basic knowledge of digital control systems, is that a system that is stable under a continuous control law (as in Chapter 5) may not be stable under the same law when control inputs and outputs are generated at discrete time intervals—with the performance being worse the longer the time interval. In our system, the sample rate is constrained to the relatively slow framerate of the video system—25 samples per second[1]. In addition, the control signals sent to the PPRK are subjected to a significant time lag due to a rather slow 9600 baud serial connection. Since the field of nonsmooth *digital* control systems does not yet exist, determining suitable values for the follow gain α and correction gain β is subject to an element of trial and error. By trial and error, we determined that to obtain satisfactory curve following, the gains must satisfy $|\beta| > |\alpha|$, although if $|\beta|/|\alpha|$ is too large, the system tends to oscillate badly.

Another factor that may have lead to the marginal stability of the control system was the use of a first order numerical integrator to solve $\dot{\mathbf{q}} = \alpha\mathbf{a} + \beta\mathbf{b} = f(\mathbf{q}(t))$. In our discrete system, used the simplistic approximation $\dot{\mathbf{q}} = \frac{\mathbf{q}_{i+1}-\mathbf{q}_i}{t}$ for velocity, and our solver was effectively

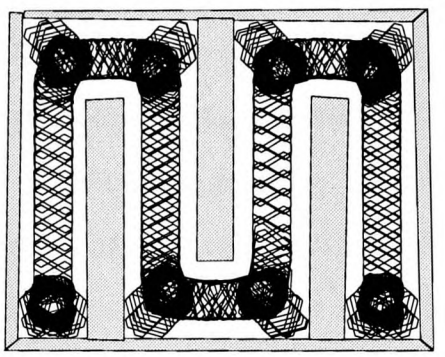
$$\mathbf{q}_{i+1} = \mathbf{q}_i + tf(\mathbf{q}(t))$$

which is the well known Euler method. Some improvement in the stability may have been realised by

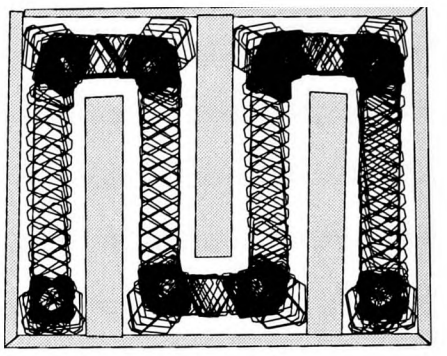
[1] We were unable to compensate for this by slowing the robot down since the PPRK's servo motors had a fixed gearing which was incompatible with very slow speeds.



RAW IMAGE

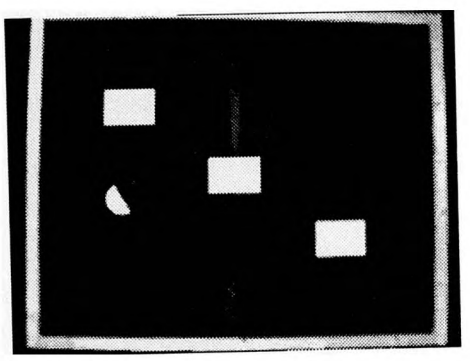


SIMULATION

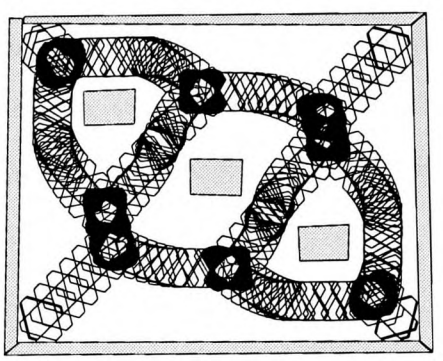


EXPERIMENT

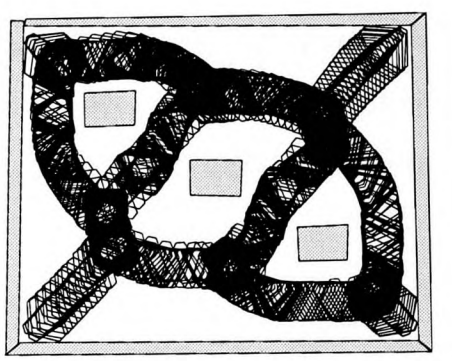
(A) – EXPERIMENT #1 ($\mathcal{A} = \bigtriangledown$)



RAW IMAGE

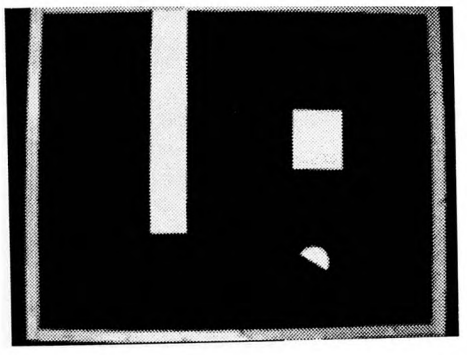


SIMULATION

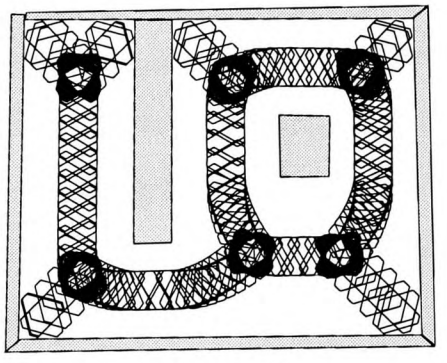


EXPERIMENT

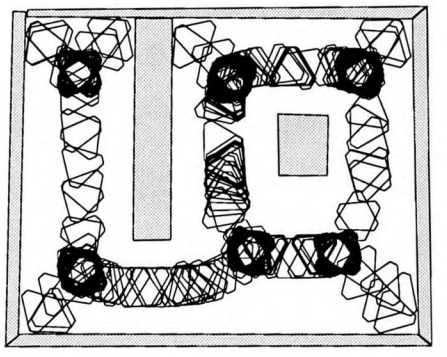
(B) – EXPERIMENT #2



RAW IMAGE



SIMULATION



EXPERIMENT

(C) – EXPERIMENT #3

FIGURE 6.7: THREE EXPERIMENTAL TRIALS.

using a higher order terms (i.e. position information from a few timesteps back/forward).

It is useful to contrast the results of the real trials with the simulations. Because the stepsize in a simulation can be made arbitrarily small, the simulated PPRK can be made to follow the three-equidistant faces $\mathcal{IF}_{ijk}$ and diameter extrema edges $\mathcal{D}^{\vartheta}_{ij}$ arbitrarily closely, thus more closely representing a continuous control system. Notice in the experimental trials that the PPRK does not follow $\mathcal{IF}_{ijk}$ and $\mathcal{D}^{\vartheta}_{ij}$ as closely as the simulations, so in practise the PPRK is unable to follow $\mathcal{IF}_{ijk}$ and $\mathcal{D}^{\vartheta}_{ij}$ when the distances $d_i(\mathbf{q})$, $d_j(\mathbf{q})$ etc. are below a certain threshold. In our experiments, this threshold was about 5 cms—about 10 pixels in a grabbed image of 640×480 pixels. As well, the existence of fish-eye distortion in the camera system causes the true distance between the robot and the obstacles to differ somewhat from the distance in the grabbed image. This also affects the ability of the PPRK to follow roadmap segments where $d_i(\mathbf{q})$, $d_j(\mathbf{q})$ are below the threshold. The threshold mentioned above takes into account this distortion as well. Finally, the control system had a tendency to oscillate a great deal more when using the PPRK due to the nature of selecting α and β. These oscillations appear in Figure 6.7 as places where the PPRK tended to 'linger'—a high density of polygons in a particular area. It is quite evident nonetheless that the NCRIT+NGVG can be followed by a real robot and that the theory developed in the preceding chapters is of practical importance.

6.4 Summary

In this chapter we have implemented simulated and real robotic systems that explore the NCRIT+NGVG described in Chapter 4 using the nonsmooth control laws developed in Chapter 5. Firstly, a specialised depth-first search called EQUISEARCH() was developed for exploration of the NCRIT+NGVG, and was subsequently shown in simulation to be more suitable for sensor-based exploration than the A* algorithm. This is due to A*'s requirement for a good heuristic estimate $h(\mathbf{q})$ of the distance to the goal and it's general backtracking behaviour when looking for shorter paths. Secondly, a software system was developed to simulate a polygonal shaped robot $\mathcal{A}$ in an environment cluttered with polygonal obstacles $\mathcal{B}_i$. The simulations were used to fine-tune behaviour of the control algorithms (determine suitable α (following) and β (correction) terms) and to determine suitable tolerances and thresholds required to detect NCRIT+NGVG nodes. Finally, once satisfied with our implementation of EQUISEARCH() and the control algorithms, we implemented a real robotic system based upon the Palm Pilot Robot Kit (PPRK). Using a closed-loop visual servoing control system, the PPRK was able to explore the NCRIT+NGVG for a number of test cases, verifying the practicality of the theory developed in the preceding chapters.

§7

"The strongest arguments prove nothing so long as the conclusions are not verified by experience. Experimental science is the queen of sciences and the goal of all speculation."

Roger Bacon

Conclusions

The theory of sensor-based motion planning, although examined in previous works, has always been hindered by the simplifying smoothness assumptions on the distance functions $d_i(\mathbf{q})$ and $D(\mathbf{q})$. This thesis' main contribution is an examination of the problem from the point of view that we simply should not use an assumption that is obviously incorrect, and that we need to approach the problem from a different angle. We have shown that nonsmooth analysis provides a new approach to study the problem and provides the necessary tools to construct a topology representing network of freespace—the main goal of all motion planning algorithms.

The key to the usefulness of nonsmooth analysis, and in particular the generalised gradient, being so useful is its highly geometric nature in finite dimensional configuration spaces. The convex hull is such a widely used concept in computational geometry, solid modelling and robotics, that practitioners have little difficulty grasping the basics. The calculation of the generalised gradient can be tedious (see Appendices A and C), but is not particularly difficult to accomplish. As well, nonsmooth analysis allows the development of theorems analogous to the smooth variants, in our case the tremendously powerful preimage theorems, since one needs only to extend the notion of a normal vector of a manifold to the normal cone of a set. It is for these reasons that we believe nonsmooth analysis should find many applications in robotics.

Our implementation of a real robotic system that implements the theory developed in this thesis further illustrates the practicality and utility of the underlying analysis.

7.1 Future Work and Possible Improvements

In this thesis, we have presented a detailed nonsmooth analysis of the distance function $D(\mathbf{q})$ for arbitrary configurations spaces, but have only presented a planning algorithm for $SE(2)$. Therefore, a key topic of future research must be to extend these techniques to higher dimensional configuration spaces in much the same way that [40] extended [15] using the smoothness assumptions. This is obviously a very difficult problem.

The key improvement we envision is to equip the robot with *on-board* distance sensors, so that it is able to explore beyond the confines of the ceiling camera's field-of-view. Two possibilities for on-board distance sensor are sonar or a laser range finder. Given our algorithm's requirement for reasonably accurate distance measurements, a sonar setup is probably less useful that the range finder, given it's poor angular resolution. Given that commercially available range finders are of generally the same size as our Palm Pilot robot, a new, larger mobile robot will need to be purchased. In addition, the new mobile robot must be equipped with a proper on-board processing system so that it can deal with the large quantity of sensor data supplied by the range finder.

§A

Generalised Gradient of the Single Object Distance Function $\partial d_i(q)$ in SE(3)

In this section we develop expressions for the generalised gradient $\partial d_i(\mathbf{q})$ of the single object distance function $d_i(\mathbf{q})$between a convex *polyhedral* robot $\mathcal{A}$ and a convex *polyhedral* obstacle $\mathcal{B}_i$. These derivations are new to the literature and are essential tools used to climb onto and follow $\mathcal{R}_{GVG}$ segments in configuration space C. Figure A.1 shows the environment used to determine $\partial d_i(\mathbf{q})$. The distance function used is the minimum distance between all points on the robot $\mathcal{A}$ and obstacle $\mathcal{B}_i$

$$d_i(\mathbf{q}) = \min_{\substack{\mathbf{a}\in\mathcal{A}(\mathbf{q}) \\ \mathbf{b}\in\mathcal{B}_i}} \|\mathbf{a}-\mathbf{b}\|$$

where $\mathbf{q} \in SE(3)$ and $\mathcal{A}(\mathbf{q})$ is the set of points occupied by the robot at configuration $\mathbf{q}$. The points on the robot $\mathcal{A}$ and obstacle $\mathcal{B}_i$ closest to each other are $\mathbf{a} = [a_x, a_y, a_z]^T$ and $\mathbf{b} = [b_x, b_y, b_z]^T$ respectively. We use the additional notation $\hat{\mathbf{a}} = [\hat{a}_x, \hat{a}_y, \hat{a}_z]^T$ to represent the coordinates of $\mathbf{a}$ in the robot's coordinate

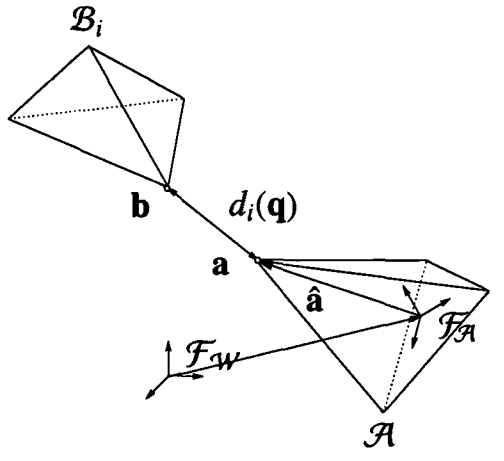


FIGURE A.1: GRADIENT OF DISTANCE IN $SE(3)$. **a** AND **b** ARE THE CLOSEST POINTS ON THE ROBOT $\mathcal{A}$ AND OBSTACLE $\mathcal{B}_i$ RESPECTIVELY. **â** REPRESENTS **a** IN $\mathcal{A}$'S COORDINATE FRAME $\mathcal{F}_{\mathcal{A}}$. $\mathcal{F}_{\mathcal{W}}$ IS THE WORLD COORDINATE FRAME.

frame $\mathcal{F}_{\mathcal{A}}$. The minimum distance then is given by

$$d_i(\mathbf{q}) = \sqrt{(b_x - a_x)^2 + (b_y - a_y)^2 + (b_z - a_z)^2}$$

Let $[x, y, z]^T$ be the location of the origin of $\mathcal{F}_{\mathcal{A}}$ with respect to $\mathcal{F}_{\mathcal{W}}$'s origin. To parameterise the rotational degrees of freedom, we choose the Euler angle parameterisation of $SO(3)$ with the three angles ϕ, θ and ψ (see [38] for an excellent description). ϕ represents a rotation about the z-axis of $\mathcal{F}_{\mathcal{A}}$ resulting in a new coordinate frame $\mathcal{F}'_{\mathcal{A}}$, θ represents a rotation about the y-axis of $\mathcal{F}'_{\mathcal{A}}$ leading to another new coordinate frame $\mathcal{F}''_{\mathcal{A}}$ and ψ represents a rotation about the z-axis of $\mathcal{F}''_{\mathcal{A}}$ resulting in the final configuration of the robot's coordinate frame. Using the shorthand notation $c = \cos$ and $s = \sin$, the composite rotation matrix is given by

$$\begin{aligned}
\mathbf{R}_{\phi,\theta,\psi} &= \mathbf{R}_\phi \mathbf{R}_\theta \mathbf{R}_\psi \\
&= \begin{bmatrix} c\phi c\theta c\psi - s\phi s\psi & -c\phi c\theta s\psi - s\phi c\psi & c\phi s\theta \\ s\phi c\theta c\psi + c\phi s\psi & -s\phi c\theta s\psi + c\phi c\psi & s\phi s\theta \\ -s\theta c\psi & s\theta s\psi & c\theta \end{bmatrix} \\
&= \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix}
\end{aligned}$$

where

$$\mathbf{R}_\phi = \begin{bmatrix} c\phi & -s\phi & 0 \\ s\phi & c\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad \mathbf{R}_\theta = \begin{bmatrix} c\theta & 0 & s\theta \\ 0 & 1 & 0 \\ -s\theta & 0 & c\theta \end{bmatrix} \qquad \mathbf{R}_\psi = \begin{bmatrix} c\psi & -s\psi & 0 \\ s\psi & c\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

We use the notation R_{ij} to represent the nine entries of the composite rotation matrix. This helps keep the subsequent matrix formulae to a reasonable size. The world coordinates of **a** are

$$\begin{aligned} \mathbf{a} &= \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} \begin{bmatrix} \hat{a}_x \\ \hat{a}_y \\ \hat{a}_z \end{bmatrix} \\ &= \begin{bmatrix} x + \hat{a}_x R_{11} + \hat{a}_y R_{12} + \hat{a}_z R_{13} \\ y + \hat{a}_x R_{21} + \hat{a}_y R_{22} + \hat{a}_z R_{23} \\ z + \hat{a}_x R_{31} + \hat{a}_y R_{32} + \hat{a}_z R_{33} \end{bmatrix} \end{aligned} \tag{A.1}$$

Using the calculus of the generalised gradient presented in Chapter 3, we know $\partial d_i(\mathbf{q})$ is a subset of the right hand side of

$$\partial d_i(\mathbf{q}) \subseteq \frac{1}{d_i(\mathbf{q})}((b_x - a_x)(\partial b_x - \partial a_x) + (b_y - a_y)(\partial b_y - \partial a_y) + (b_z - a_z)(\partial b_z - \partial a_z))$$

The generalised gradient terms ∂a_x, ∂b_x etc. can themselves be found to be subsets of sets using the calculus of Chapter 3. Since elements of the rotation matrices R_{ij} are smooth, we know $\partial R_{ij} = \nabla R_{ij}$. By a similar argument $\partial x = \nabla x$, $\partial y = \nabla y$ and $\partial z = \nabla z$ and so we have

$$\begin{aligned} \partial a_x &\subseteq \nabla x + \partial \hat{a}_x R_{11} + \hat{a}_x \nabla R_{11} + \partial \hat{a}_y R_{12} + \hat{a}_y \nabla R_{12} + \partial \hat{a}_z R_{13} + \hat{a}_x \nabla R_{13} \\ \partial a_y &\subseteq \nabla y + \partial \hat{a}_x R_{21} + \hat{a}_x \nabla R_{21} + \partial \hat{a}_y R_{22} + \hat{a}_y \nabla R_{22} + \partial \hat{a}_z R_{23} + \hat{a}_x \nabla R_{23} \\ \partial a_z &\subseteq \nabla z + \partial \hat{a}_x R_{31} + \hat{a}_x \nabla R_{31} + \partial \hat{a}_y R_{32} + \hat{a}_y \nabla R_{32} + \partial \hat{a}_z R_{33} + \hat{a}_x \nabla R_{33} \end{aligned} \tag{A.2}$$

It turns out we cannot simply determine the individual generalised gradients ($\partial \hat{a}_x$, ∂b_x etc.) and plug these into Equations A.2 because the terms $\hat{a}_x$, b_x etc. are in fact nonsmooth and nonconvex in general. To see these terms are nonsmooth and nonconvex, consider Figure A.2 which shows the terms a_x, b_x and ab_x as a vertex of $\mathcal{A}$ moves to the right (increasing x). A nonsmooth point occurs at the x-coordinate of $\mathcal{B}_i$'s vertex. At those points where these terms are nonsmooth, the calculus of the generalised gradient

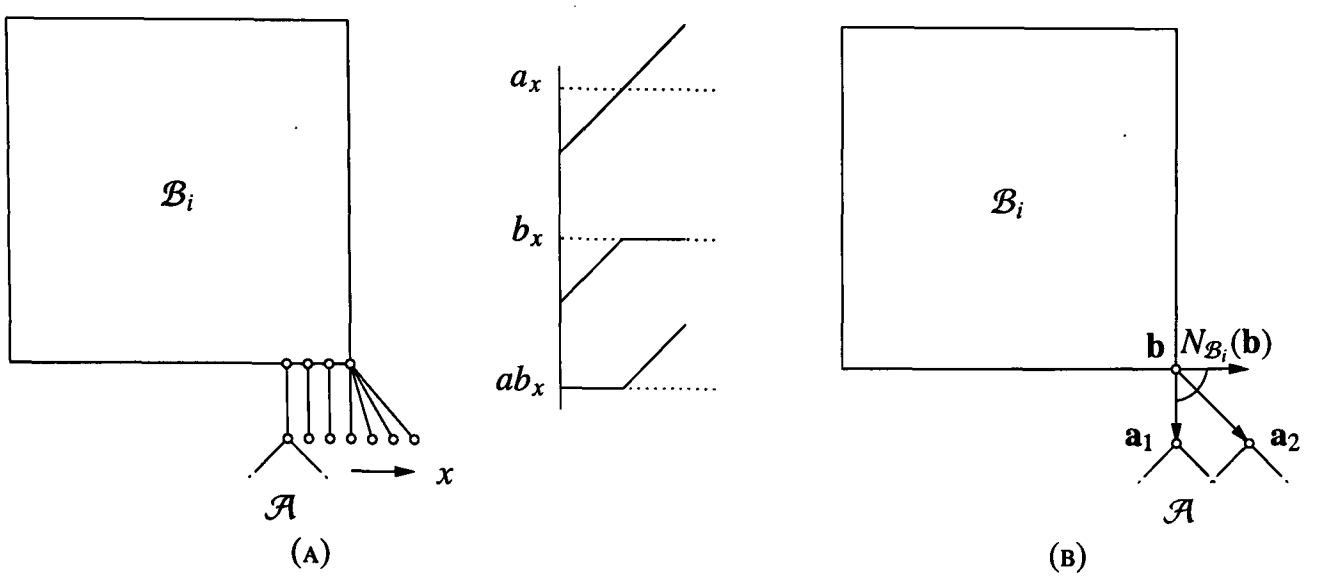


FIGURE A.2: a_x, b_x AND ab_x ARE NONSMOOTH IN GENERAL. (A) A VERTEX OF THE ROBOT $\mathcal{A}$ IS THE CLOSEST POINT TO THE OBSTACLE $\mathcal{B}_i$ (THE FIGURE SHOWS A POLYGONAL $\mathcal{A}/\mathcal{B}_i$ IN $SE(2)$). AS THE VERTEX MOVES RIGHT (INCREASING x), A NONSMOOTH POINT OF b_x AND ab_x OCCURS AT THE x-COORDINATE OF THE VERTEX. GRAPHS OF a_x, b_x AND ab_x ARE GIVEN TO FURTHER ILLUSTRATE THE NONSMOOTH POINT. (B) b_x IS NONSMOOTH WHEN $\mathbf{a} - \mathbf{b}$ IS NOT IN THE INTERIOR OF $N_{\mathcal{B}_i}(\mathbf{b})$, AS IS THE CASE FOR $\mathbf{a}_1 - \mathbf{b}$.

will result in a set of points for the right hand side of Equations A.2, so $\partial d_i(\mathbf{q})$ cannot be said to contain a single point when there is a single closest point on $\mathcal{A}$ and $\mathcal{B}_i$. In order to show $\partial d_i(\mathbf{q})$ contains a single point when there is a single closest point $\mathbf{a}$ on $\mathcal{A}$ and a single closest point $\mathbf{b}$ on $\mathcal{B}_i$, we need a different approach. To determine $\partial d_i(\mathbf{q})$ we recall Corollary 3.4 which gives the following expression for $\partial\phi(\mathbf{x})$ for a finite dimensional domain.

$$\partial\phi(\mathbf{x}) = \text{co}\{\lim \nabla\phi(\mathbf{x}_k) : \mathbf{x}_k \to \mathbf{x}, \mathbf{x}_k \notin S, \mathbf{x}_k \notin \Omega_\phi\}$$

and present Lemma A.1.

Lemma A.1 The terms ∂a_x, ∂a_y, ∂a_z, ∂b_x, ∂b_y and ∂b_z all contain a single point (a_x etc. are smooth) if and only if there is a single closest point $\mathbf{a}$ on $\mathcal{A}$ and a single closest point $\mathbf{b}$ on $\mathcal{B}_i$ such that

1. $\mathbf{a} - \mathbf{b}$ is in the interior of $N_{\mathcal{B}_i}(\mathbf{b})$; and
2. $\mathbf{b} - \mathbf{a}$ is in the interior of $N_{\mathcal{A}}(\mathbf{a})$. (A.3)

Proof (**IF**) Figure A.2(b) shows the meaning of Conditions A.3. The witness vector $\mathbf{a}_2 - \mathbf{b}$ is in the interior of $N_{\mathcal{B}_i}(\mathbf{b})$ but $\mathbf{a}_1 - \mathbf{b}$ is not. Let there be two single closest points $\mathbf{a}$ on $\mathcal{A}$ and $\mathbf{b}$ on $\mathcal{B}_i$ satisfying Conditions A.3. For two convex polytopes $\mathcal{A}$ and $\mathcal{B}_i$, there are a number of ways these single points can

exist—vertex-vertex, vertex-edge, vertex-face and edge-edge. The other combinations edge-face and face face always lead to multiple closest points, by the convexity of $\mathcal{A}$ and $\mathcal{B}_i$. Let us consider each case in turn.

a is a Vertex of $\mathcal{A}$ and b is a Vertex of $\mathcal{B}_i$

This is the simplest of the cases. Because the closest points **a** and **b** satisfy Conditions A.3, there exists a neighbourhood nbhd(**q**) where the closest point **b** on $\mathcal{B}_i$ does not change so

$$\partial b_x = \partial b_y = \partial b_z = \mathbf{0}$$

which all contain a single point, so b_x, b_y and b_z are smooth. Likewise, the closest point $\hat{\mathbf{a}}$ on $\mathcal{A}$ does not change in $\mathcal{A}$'s frame so

$$\partial \hat{a}_x = \partial \hat{a}_y = \partial \hat{a}_z = \mathbf{0}$$

Substituting into Equations A.2

$$\partial a_x = \nabla a_x = \nabla x + \hat{a}_x \nabla R_{11} + \hat{a}_y \nabla R_{12} + \hat{a}_x \nabla R_{13}$$
$$\partial a_y = \nabla a_y = \nabla y + \hat{a}_x \nabla R_{21} + \hat{a}_y \nabla R_{22} + \hat{a}_x \nabla R_{23}$$
$$\partial a_z = \nabla a_z = \nabla z + \hat{a}_x \nabla R_{31} + \hat{a}_y \nabla R_{32} + \hat{a}_x \nabla R_{33}$$

Since the gradients ∇a_x, ∇a_y and ∇a_z exist, a_x, a_y and a_z are smooth.

a is on an Edge or Face of $\mathcal{A}$ and b is a Vertex of $\mathcal{B}_i$

Again, because the closest points **a** and **b** satisfy Conditions A.3, there exists a neighbourhood nbhd(**q**) where the closest point **b** on $\mathcal{B}_i$ does not change so

$$\partial b_x = \partial b_y = \partial b_z = \mathbf{0}$$

which all contain a single point, so b_x, b_y and b_z are smooth. However, the closest point $\hat{\mathbf{a}}$ may change, so we need to show it does so smoothly. Because the closest points **a** and **b** satisfy Conditions A.3, there exists a neighbourhood nbhd(**q**) where the closest point **a** remains on an edge or face. We show smoothness by showing the directional derivative $\hat{\mathbf{a}}'(\mathbf{x};\mathbf{e}) = \frac{\hat{\mathbf{a}}(\mathbf{q}+t\mathbf{e})-\hat{\mathbf{a}}(\mathbf{q})}{t} = \hat{\mathbf{a}}^{\circ}(\mathbf{x};\mathbf{e})$ for every direction $\mathbf{e} \in C$ [16]. The

location of **b** in $\mathcal{A}$'s frame is given by

$$\hat{\mathbf{b}} = R^{-1}_{\theta,\phi,\psi}(\mathbf{b} - [x, y, z]^T)$$

Define by $H_{\mathcal{A}} : \mathbb{R}^3 \to \mathbb{R}^3$ the map that takes a point $\hat{\mathbf{b}}$ to the closest point $\hat{\mathbf{a}}$ on the convex set $\mathcal{A}$. It is known [16] that this a continuous and smooth function. The location of $\hat{\mathbf{a}}$ is then

$$\hat{\mathbf{a}} = H_{\mathcal{A}}(R^{-1}_{\theta,\phi,\psi}(\mathbf{b} - [x, y, z]^T))$$

Since $H_{\mathcal{A}}$, $R^{-1}_{\theta,\phi,\psi}$, **b** and $[x, y, z]^T$ are all smooth, we have $\hat{\mathbf{a}}'(\mathbf{x}; \mathbf{e}) = \hat{\mathbf{a}}^\circ(\mathbf{x}; \mathbf{e})$ is equal for all $\mathbf{e} \in \mathbb{R}^6$ [16], so $\partial\hat{a}_x$, $\partial\hat{a}_y$ and $\partial\hat{a}_z$ all contain one element, and $\hat{a}_x$, $\hat{a}_y$ and $\hat{a}_z$ are smooth. Substituting into Equations A.2

$$\begin{aligned}
\partial a_x = \nabla a_x &= \nabla x + \nabla\hat{a}_x R_{11} + \hat{a}_x \nabla R_{11} + \nabla\hat{a}_y R_{12} + \hat{a}_y \nabla R_{12} + \nabla\hat{a}_z R_{13} + \hat{a}_x \nabla R_{13} \\
\partial a_y = \nabla a_x &= \nabla y + \nabla\hat{a}_x R_{21} + \hat{a}_x \nabla R_{21} + \nabla\hat{a}_y R_{22} + \hat{a}_y \nabla R_{22} + \nabla\hat{a}_z R_{23} + \hat{a}_x \nabla R_{23} \\
\partial a_z = \nabla a_x &= \nabla z + \nabla\hat{a}_x R_{31} + \hat{a}_x \nabla R_{31} + \nabla\hat{a}_y R_{32} + \hat{a}_y \nabla R_{32} + \nabla\hat{a}_z R_{33} + \hat{a}_x \nabla R_{33}
\end{aligned}$$

Since the gradients ∇a_x, ∇a_y and ∇a_z exist, a_x, a_y and a_z are smooth. We also make the observation that since the vector **ab** is perpendicular to an face/edge of $\mathcal{A}$ and the vector $R_{\phi,\theta,\psi}\left[\frac{\partial\hat{a}_x}{\partial x}, \frac{\partial\hat{a}_y}{\partial x}, \frac{\partial\hat{a}_z}{\partial x}\right]^T$ is constrained to lie in the face/edge of $\mathcal{A}$ (in fact all partial vectors of all 6 coordinates are constrained to lie in the face/edge of $\mathcal{A}$) we know

$$\left\langle R_{\phi,\theta,\psi}\left[\frac{\partial\hat{a}_x}{\partial i}, \frac{\partial\hat{a}_y}{\partial i}, \frac{\partial\hat{a}_z}{\partial i}\right]^T, \mathbf{a} - \mathbf{b} \right\rangle = 0 \text{ for all } i \in \{x, y, z, \theta, \phi, \psi\}$$

a is a Vertex of $\mathcal{A}$ and b is on an Edge or Face of $\mathcal{B}_i$

Using the same arguments as the previous case, we know

$$\partial\hat{a}_x = \partial\hat{a}_y = \partial\hat{a}_z = \mathbf{0}$$

and

$$\partial b_x = \nabla b_x$$
$$\partial b_y = \nabla b_y$$
$$\partial b_z = \nabla b_z$$

and

$$\left\langle \left[\frac{\partial b_x}{\partial i}, \frac{\partial b_y}{\partial i}, \frac{\partial b_z}{\partial i} \right]^T, \mathbf{a} - \mathbf{b} \right\rangle = 0 \text{ for all } i \in \{x, y, z, \theta, \phi, \psi\}$$

a is on an Edge of $\mathcal{A}$ and b is an Edge of $\mathcal{B}_i$

This is the trickiest of the cases, since both of the closest points $\hat{\mathbf{a}}$ and $\mathbf{b}$ may change during any infinitesimal motion. However, because the closest points $\mathbf{a}$ and $\mathbf{b}$ satisfy Conditions A.3, there exists a neighbourhood nbhd($\mathbf{q}$) where the closest points remain on the edges of $\mathcal{A}$ and $\mathcal{B}_i$. There are only two directions $\mathbf{v}_1$ and $\mathbf{v}_2$ that $\mathbf{b}$ can move (both a parallel to the edge of $\mathcal{B}_i$), with $\mathbf{v}_1 = -\mathbf{v}_2$. We know then that $\mathbf{b}(\mathbf{w} + t\mathbf{e}) - \mathbf{b}(\mathbf{w}) = \mathbf{b}(\mathbf{q} + t\mathbf{e}) - \mathbf{b}(\mathbf{q})$ for $\mathbf{w} \in$ nbhd($\mathbf{q}$) and t sufficiently small. This leads immediately to:

$$\begin{aligned}
\mathbf{b}^\circ(\mathbf{q}; \mathbf{e}) &= \limsup_{\substack{\mathbf{w} \to \mathbf{q} \\ t \to 0^+}} \frac{\mathbf{b}(\mathbf{w} + t\mathbf{e}) - \mathbf{b}(\mathbf{w})}{t} \\
&= \lim_{t \to 0^+} \frac{\mathbf{b}(\mathbf{q} + t\mathbf{e}) - \mathbf{b}(\mathbf{q})}{t} \\
&= \mathbf{b}'(\mathbf{q}; \mathbf{e})
\end{aligned}$$

and so $\mathbf{b}(\mathbf{q})$ is smooth. Using the same argument as the previous case, $\hat{\mathbf{a}} = H_{\mathcal{A}}(R^{-1}_{\theta,\phi,\psi}(\mathbf{b}(\mathbf{q}) - [x, y, z]^T))$ is thus smooth as well.

$$\partial b_x = \nabla b_x$$
$$\partial b_y = \nabla b_y$$
$$\partial b_z = \nabla b_z$$
$$\partial a_x = \nabla a_x = \nabla x + \nabla \hat{a}_x R_{11} + \hat{a}_x \nabla R_{11} + \nabla \hat{a}_y R_{12} + \hat{a}_y \nabla R_{12} + \nabla \hat{a}_z R_{13} + \hat{a}_x \nabla R_{13}$$
$$\partial a_y = \nabla a_y = \nabla y + \nabla \hat{a}_x R_{21} + \hat{a}_x \nabla R_{21} + \nabla \hat{a}_y R_{22} + \hat{a}_y \nabla R_{22} + \nabla \hat{a}_z R_{23} + \hat{a}_x \nabla R_{23}$$
$$\partial a_z = \nabla a_z = \nabla z + \nabla \hat{a}_x R_{31} + \hat{a}_x \nabla R_{31} + \nabla \hat{a}_y R_{32} + \hat{a}_y \nabla R_{32} + \nabla \hat{a}_z R_{33} + \hat{a}_x \nabla R_{33}$$

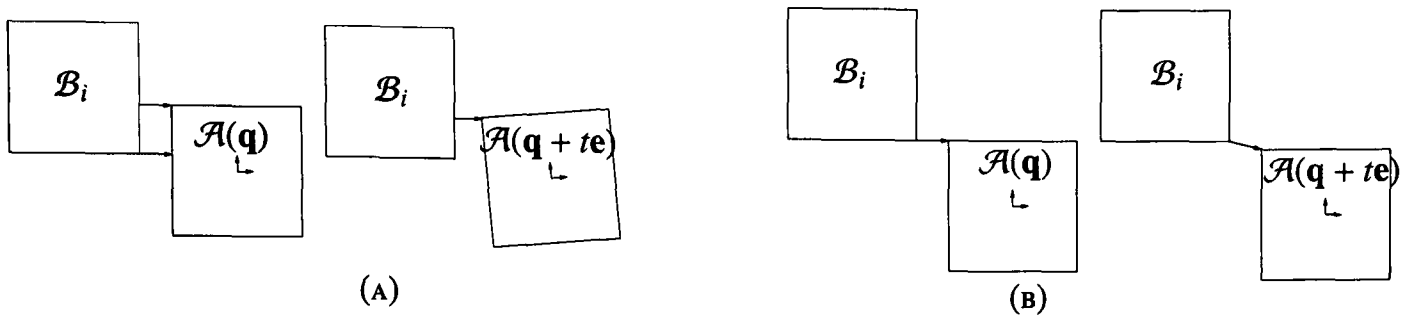


FIGURE A.3: ILLUSTRATION OF MEASURE ZERO SETS S_1 AND S_2. (A) IT IS POSSIBLE TO GET OUT OF THE SET S_1 OF MULTIPLE CLOSEST POINTS WITH AN INFINITESIMAL ROTATION. (B) IT IS POSSIBLE TO GET OUT OF THE SET S_2 WHERE THE SINGLE CLOSEST POINTS VIOLATE CONDITIONS A.3 WITH AN INFINITESIMAL TRANSLATION.

Again, since the gradients ∇a_x, ∇a_y and ∇a_z exist, a_x, a_y and a_z are smooth.

(ONLY IF) Let there be two single closest points $\mathbf{a}$ on $\mathcal{A}$ and $\mathbf{b}$ on $\mathcal{B}_i$ that do not satisfy Conditions A.3. For some small motion of $\mathcal{A}$, a closest point $\mathbf{a}/\mathbf{b}$ will switch to an edge/vertex/face, and the velocity (direction of point motion) of the point will necessarily stop or change direction, leading to a nonsmooth $\partial\mathbf{a}$ or $\partial\mathbf{b}$ term. To illustrate, Figure A.2(b) shows that b_x is nonsmooth when $\mathbf{a} - \mathbf{b}$ is not in the interior of $N_{\mathcal{B}_i}(\mathbf{b})$ ■

A.1 The Final Step

Recalling Equation A.3 from earlier in this appendix, we can see that $\partial d_i(\mathbf{q})$ can be determined by

$$\partial d_i(\mathbf{q}) = \text{co}\{\lim \nabla d_i(\mathbf{q}_k) : \mathbf{q}_k \to \mathbf{q}, \mathbf{q}_k \notin S, \mathbf{q}_k \notin \Omega_{d_i}\}$$

By definition, the sequence points $\mathbf{q}_k$ cannot be elements of the set of nondifferentiable points Ω_{d_i}, otherwise $\nabla d_i(\mathbf{q}_k)$ would not exist to take the limit of. More importantly for our problem is the fact we can also exclude $\mathbf{q}_k$ from *any* set S of measure zero when taking the limits $\lim \nabla d_i(\mathbf{q}_k)$. Lemmas A.2 and A.3 are used to construct a set S appropriate to our needs.

Lemma A.2 The set of configurations S_1 where there exists a set of multiple closest points P has measure zero in C_r.

Proof If there are multiple closest points, then those closest points must lie on edges a of $\mathcal{A}$ and b of $\mathcal{B}_i$. For a particular edge-edge combination a,b, the set of configurations $S_{1_{a,b}}$ where the exist multiple closest points corresponds to a constant orientation hyperplane (if $\mathcal{A}$ rotates, there will no longer be multiple closest points), which has measure zero. Since there are a finite number of edge-edge combinations, the

set $S_1 = \cup_{a,b} S_{1_{a,b}}$ has measure zero as well. ■

Lemma A.3 The set S_2 of configurations where there exists a single closest point **a** on $\mathcal{A}$ and a single closest point **b** on $\mathcal{B}_i$ such that Conditions A.3 are violated has measure zero in C_r.

Proof For a particular vertex-vertex combination a,b, the set of configurations $S_{2_{a,b}}$ where the exists a single closest point **a** on $\mathcal{A}$ and a single closest point **b** on $\mathcal{B}_i$ such that Conditions A.3 are violated, correspond to motion where **a** is constrained to a line segment. Configurations resulting from motion satisfying this constraint are known [38] to be a measure-zero submanifold of $SE(2)$. Since there are a finite number of vertex-vertex combinations, the set $S_1 = \cup_{a,b} S_{1_{a,b}}$ has measure zero as well. ■

Setting $S = S_1 \cup S_2$, we can now show $d_i(\mathbf{q})$ is smooth whenever a single closest point **a** on $\mathcal{A}$ and a single closest point **b** on $\mathcal{B}_i$ exist by showing that all gradient limits approach a single point ζ

$$\lim \nabla d_i(\mathbf{q}_k) = \zeta \text{ for all } \mathbf{q}_k \to \mathbf{q}, \mathbf{q}_k \notin S, \mathbf{q}_k \notin \Omega_{d_i}$$

This single point is then the gradient $\zeta = \nabla d_i(\mathbf{q})$. The gradient term $\lim \nabla d_i(\mathbf{q}_k)$ (in long form) is

$$\begin{aligned}
\nabla d_i(\mathbf{q}_k) &= d_i(\mathbf{q}_k)^{-1}((b_x - a_x)(\partial b_x - \partial a_x) + (b_y - a_y)(\partial b_y - \partial a_y) + (b_z - a_z)(\partial b_z - \partial a_z)) \\
&= d_i(\mathbf{q}_k)^{-1}((b_x - a_x)(\nabla b_x - \nabla a_x) + (b_y - a_y)(\nabla b_y - \nabla a_y) + (b_z - a_z)(\nabla b_z - \nabla a_z)) \\
&= d_i(\mathbf{q}_k)^{-1}((b_x - a_x)(\nabla b_x - (\nabla x + \nabla \hat{a}_x R_{11} + \hat{a}_x \nabla R_{11} + \nabla \hat{a}_y R_{12} + \hat{a}_y \nabla R_{12} + \nabla \hat{a}_z R_{13} + \hat{a}_x \nabla R_{13}))+ \\
&\qquad (b_y - a_y)(\nabla b_y - (\nabla y + \nabla \hat{a}_x R_{21} + \hat{a}_x \nabla R_{21} + \nabla \hat{a}_y R_{22} + \hat{a}_y \nabla R_{22} + \nabla \hat{a}_z R_{23} + \hat{a}_x \nabla R_{23}))+ \\
&\qquad (b_z - a_z)(\nabla b_z - (\nabla z + \nabla \hat{a}_x R_{31} + \hat{a}_x \nabla R_{31} + \nabla \hat{a}_y R_{32} + \hat{a}_y \nabla R_{32} + \nabla \hat{a}_z R_{33} + \hat{a}_x \nabla R_{33})))
\end{aligned}$$

It the above equation we must note that all the terms are calculated at $\mathbf{q}_k$, since $\mathbf{q}$ is the accumulation point of the limit. A number of cancellations occur in the above equation, the first being

$$(b_x - a_x)\nabla b_x + (b_y - a_y)\nabla b_y + (b_z - a_z)\nabla b_z = \mathbf{0}$$

since either $\nabla b_x = \nabla b_y = \nabla b_z = \mathbf{0}$, or the components $\left\langle [\frac{\partial b_x}{\partial i}, \frac{\partial b_y}{\partial i}, \frac{\partial b_z}{\partial i}]^T, \mathbf{a} - \mathbf{b} \right\rangle = 0$ for all $i \in \{x, y, z, \theta, \phi, \psi\}$. Similarly,

$$(b_x - a_x)\nabla \hat{a}_x R_{11} + (b_y - a_y)\nabla \hat{a}_y R_{12} + (b_z - a_z)\nabla \hat{a}_z R_{13} +$$
$$(b_x - a_x)\nabla \hat{a}_x R_{21} + (b_y - a_y)\nabla \hat{a}_y R_{22} + (b_z - a_z)\nabla \hat{a}_z R_{23} +$$
$$(b_x - a_x)\nabla \hat{a}_x R_{31} + (b_y - a_y)\nabla \hat{a}_y R_{32} + (b_z - a_z)\nabla \hat{a}_z R_{33} = \mathbf{0}$$

since either $\nabla \hat{a}_x = \nabla \hat{a}_y = \nabla \hat{a}_z = \mathbf{0}$, or the components $\left\langle R_{\phi,\theta,\psi} [\frac{\partial \hat{a}_x}{\partial i}, \frac{\partial \hat{a}_y}{\partial i}, \frac{\partial \hat{a}_z}{\partial i}]^T, \mathbf{a} - \mathbf{b} \right\rangle = 0$ for all $i \in \{x, y, z, \theta, \phi, \psi\}$. The remaining term

$$(a_x - b_x)(\nabla x + \hat{a}_x \nabla R_{11} + \hat{a}_y \nabla R_{12} + \hat{a}_x \nabla R_{13}) +$$
$$(a_y - b_y)(\nabla y + \hat{a}_x \nabla R_{21} + \hat{a}_y \nabla R_{22} + \hat{a}_x \nabla R_{23}) +$$
$$(a_z - b_z)(\nabla z + \hat{a}_x \nabla R_{31} + \hat{a}_y \nabla R_{32} + \hat{a}_x \nabla R_{33})$$

is a smooth function, and so $\lim \nabla d_i(\mathbf{q}_k) = \boldsymbol{\zeta}$ for all $\mathbf{q}_k \to \mathbf{q}$. Appendix B gives details of the calculation of $\partial d_i(\mathbf{q}) = \nabla d_i(\mathbf{q})$ for this smooth case.

§B

Gradient of the Single Object Distance Function $\nabla d_i(\mathbf{q})$ in SE(3) (smooth case)

In this section we develop expressions for the gradient of the distance function $\nabla d_i(\mathbf{q})$. Figure B.1 shows the environment used to determine the $\nabla d_i(\mathbf{q})$. The gradient $\nabla d_i(\mathbf{q})$ has components $\frac{\partial d_i(\mathbf{q})}{\partial x}$, $\frac{\partial d_i(\mathbf{q})}{\partial y}$, $\frac{\partial d_i(\mathbf{q})}{\partial z}$, $\frac{\partial d_i(\mathbf{q})}{\partial \phi}$, $\frac{\partial d_i(\mathbf{q})}{\partial \theta}$, $\frac{\partial d_i(\mathbf{q})}{\partial \psi}$. The x, y and z components are all determined a similar manner, while the ϕ, θ and ψ component require a bit more mathematical acrobatics. We start with the x component of the gradient.

$$\frac{\partial d_i(\mathbf{q})}{\partial x} = \frac{1}{d_i(\mathbf{q})}\left((b_x - a_x)\left(\frac{\partial b_x}{\partial x} - \frac{\partial a_x}{\partial x}\right) + (b_y - a_y)\left(\frac{\partial b_y}{\partial x} - \frac{\partial a_y}{\partial x}\right) + (b_z - a_z)\left(\frac{\partial b_z}{\partial x} - \frac{\partial a_z}{\partial x}\right)\right)$$

which can be considered a inner product of the form

$$\frac{\partial d_i(\mathbf{q})}{\partial x} = \frac{1}{d_i(\mathbf{q})}\left\langle (\mathbf{b} - \mathbf{a}), \frac{\partial}{\partial x}(\mathbf{b} - \mathbf{a})\right\rangle \tag{B.1}$$

The partial derivatives of **a** are determined from Equations A.1

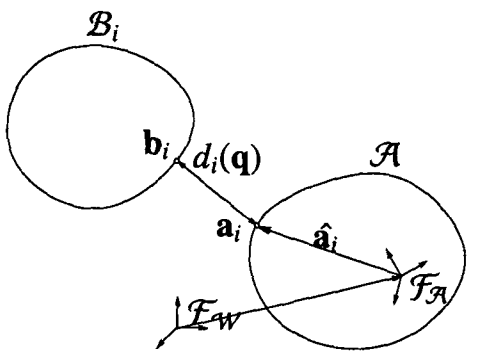


FIGURE B.1: GRADIENT OF DISTANCE IN $SE(3)$. $\mathbf{a}$ AND $\mathbf{b}$ ARE THE CLOSEST POINTS ON THE ROBOT $\mathcal{A}$ AND OBSTACLE $\mathcal{B}_i$ RESPECTIVELY. $\hat{\mathbf{a}}$ REPRESENTS $\mathbf{a}$ IN $\mathcal{A}$'S COORDINATE FRAME $\mathcal{F}_{\mathcal{A}}$. $\mathcal{F}_{\mathcal{W}}$ IS THE WORLD COORDINATE FRAME.

$$\begin{aligned}
\frac{\partial a_x}{\partial x} &= 1 + \frac{\partial \hat{a}_x}{\partial x} R_{11} + \frac{\partial \hat{a}_y}{\partial x} R_{12} + \frac{\partial \hat{a}_z}{\partial x} R_{13} \\
\frac{\partial a_y}{\partial x} &= \frac{\partial \hat{a}_x}{\partial x} R_{21} + \frac{\partial \hat{a}_y}{\partial x} R_{22} + \frac{\partial \hat{a}_z}{\partial x} R_{23} \\
\frac{\partial a_z}{\partial x} &= \frac{\partial \hat{a}_x}{\partial x} R_{31} + \frac{\partial \hat{a}_y}{\partial x} R_{32} + \frac{\partial \hat{a}_z}{\partial x} R_{33}
\end{aligned} \tag{B.2}$$

Substituting into Equation B.1

$$\frac{\partial d_i(\mathbf{q})}{\partial x} = \frac{1}{d_i(\mathbf{q})} \left\langle \begin{bmatrix} b_x - a_x \\ b_y - a_y \\ b_z - a_z \end{bmatrix}, \begin{bmatrix} \frac{\partial b_x}{\partial x} \\ \frac{\partial b_y}{\partial x} \\ \frac{\partial b_z}{\partial x} \end{bmatrix} - \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} - \begin{bmatrix} \frac{\partial \hat{a}_x}{\partial x} R_{11} + \frac{\partial \hat{a}_y}{\partial x} R_{12} + \frac{\partial \hat{a}_z}{\partial x} R_{13} \\ \frac{\partial \hat{a}_x}{\partial x} R_{21} + \frac{\partial \hat{a}_y}{\partial x} R_{22} + \frac{\partial \hat{a}_z}{\partial x} R_{23} \\ \frac{\partial \hat{a}_x}{\partial x} R_{31} + \frac{\partial \hat{a}_y}{\partial x} R_{32} + \frac{\partial \hat{a}_z}{\partial x} R_{33} \end{bmatrix} \right\rangle \tag{B.3}$$

It is readily apparent that the vector $(\mathbf{b} - \mathbf{a})$ is orthogonal to the tangent spaces of the boundary of the obstacle $\mathcal{B}_i$ at $\mathbf{b}$ as well as the robot $\mathcal{A}$ at $\mathbf{a}$. The vector $[\frac{\partial b_x}{\partial x}, \frac{\partial b_y}{\partial x}, \frac{\partial b_z}{\partial x}]^T$, or the vector of partial fractions with respect to any of the degrees of freedom for that matter, is an element of the tangent space of the boundary of $\mathcal{B}_i$. As well, $\mathbf{R}_{\phi,\theta,\psi}[\frac{\partial \hat{a}_x}{\partial x}, \frac{\partial \hat{a}_y}{\partial x}, \frac{\partial \hat{a}_z}{\partial x}]$ is an element of the tangent space of the boundary of $\mathcal{A}$. So, these terms disappear from inner product of Equation B.3. The resulting inner product of $(\mathbf{b} - \mathbf{a})$ and $[-1, 0, 0]^T$ is

$$\frac{\partial d_i(\mathbf{q})}{\partial x} = \frac{1}{d_i(\mathbf{q})}(a_x - b_x)$$

Using the same approach, we determine

$$\frac{\partial d_i(\mathbf{q})}{\partial y} = \frac{1}{d_i(\mathbf{q})}(a_y - b_y)$$
$$\frac{\partial d_i(\mathbf{q})}{\partial z} = \frac{1}{d_i(\mathbf{q})}(a_z - b_z)$$

The gradient components corresponding to the rotational degrees of freedom require a bit more work. Let us first consider $\frac{\partial d_i(\mathbf{q})}{\partial \phi}$.

$$\frac{\partial d_i(\mathbf{q})}{\partial \phi} = \frac{1}{d_i(\mathbf{q})}\left((b_x - a_x)\left(\frac{\partial b_x}{\partial \phi} - \frac{\partial a_x}{\partial \phi}\right) + (b_y - a_y)\left(\frac{\partial b_y}{\partial \phi} - \frac{\partial a_y}{\partial \phi}\right) + (b_z - a_z)\left(\frac{\partial b_z}{\partial \phi} - \frac{\partial a_z}{\partial \phi}\right)\right)$$

Since the R_{ij}'s are dependent on ϕ, we need to apply the multiplication rule to the partial derivatives with respect to ϕ. The resulting terms are more complicated than the x components in Equation B.2.

$$\begin{aligned}
\frac{\partial a_x}{\partial \phi} &= \frac{\partial \hat{a}_x}{\partial \phi}R_{11} + \hat{a}_x\frac{\partial R_{11}}{\partial \phi} + \frac{\partial \hat{a}_y}{\partial \phi}R_{12} + \hat{a}_y\frac{\partial R_{12}}{\partial \phi} + \frac{\partial \hat{a}_z}{\partial \phi}R_{13} + \hat{a}_z\frac{\partial R_{13}}{\partial \phi} \\
\frac{\partial a_y}{\partial \phi} &= \frac{\partial \hat{a}_x}{\partial \phi}R_{21} + \hat{a}_x\frac{\partial R_{21}}{\partial \phi} + \frac{\partial \hat{a}_y}{\partial \phi}R_{22} + \hat{a}_y\frac{\partial R_{22}}{\partial \phi} + \frac{\partial \hat{a}_z}{\partial \phi}R_{23} + \hat{a}_z\frac{\partial R_{23}}{\partial \phi} \\
\frac{\partial a_z}{\partial \phi} &= \frac{\partial \hat{a}_x}{\partial \phi}R_{31} + \hat{a}_x\frac{\partial R_{31}}{\partial \phi} + \frac{\partial \hat{a}_y}{\partial \phi}R_{32} + \hat{a}_y\frac{\partial R_{32}}{\partial \phi} + \frac{\partial \hat{a}_z}{\partial \phi}R_{33} + \hat{a}_z\frac{\partial R_{33}}{\partial \phi}
\end{aligned}$$

Once again, the inner product of Equation B.1 can be split into a number of components.

$$\frac{\partial d_i(\mathbf{q})}{\partial \phi} = \frac{1}{d_i(\mathbf{q})}\left\langle \begin{bmatrix} b_x - a_x \\ b_y - a_y \\ b_z - a_z \end{bmatrix}, \begin{bmatrix} \frac{\partial b_x}{\partial \phi} \\ \frac{\partial b_y}{\partial \phi} \\ \frac{\partial b_z}{\partial \phi} \end{bmatrix} - \begin{bmatrix} \frac{\partial \hat{a}_x}{\partial \phi}R_{11} + \frac{\partial \hat{a}_y}{\partial \phi}R_{12} + \frac{\partial \hat{a}_z}{\partial \phi}R_{13} \\ \frac{\partial \hat{a}_x}{\partial \phi}R_{21} + \frac{\partial \hat{a}_y}{\partial \phi}R_{22} + \frac{\partial \hat{a}_z}{\partial \phi}R_{23} \\ \frac{\partial \hat{a}_x}{\partial \phi}R_{31} + \frac{\partial \hat{a}_y}{\partial \phi}R_{32} + \frac{\partial \hat{a}_z}{\partial \phi}R_{33} \end{bmatrix} - \begin{bmatrix} \hat{a}_x\frac{\partial R_{11}}{\partial \phi} + \hat{a}_y\frac{\partial R_{12}}{\partial \phi} + \hat{a}_z\frac{\partial R_{13}}{\partial \phi} \\ \hat{a}_x\frac{\partial R_{21}}{\partial \phi} + \hat{a}_y\frac{\partial R_{22}}{\partial \phi} + \hat{a}_z\frac{\partial R_{23}}{\partial \phi} \\ \hat{a}_x\frac{\partial R_{31}}{\partial \phi} + \hat{a}_y\frac{\partial R_{32}}{\partial \phi} + \hat{a}_z\frac{\partial R_{33}}{\partial \phi} \end{bmatrix} \right\rangle \tag{B.4}$$

Using the same reasoning regarding $(\mathbf{b} - \mathbf{a})$ being orthogonal to the tangent spaces of the boundaries of both $\mathcal{A}$ and $\mathcal{B}_i$, the first two components of the right hand side of the inner product in Equation B.4 are nullified. Expanding the partial derivatives of the R_{ij}'s, we are left with the following

$$\frac{\partial d_i(\mathbf{q})}{\partial \phi} = \frac{1}{d_i(\mathbf{q})}\left\langle \begin{bmatrix} b_x - a_x \\ b_y - a_y \\ b_z - a_z \end{bmatrix}, \begin{bmatrix} \hat{a}_x(s\phi c\theta c\psi + c\phi s\psi) - \hat{a}_y(s\phi c\theta s\psi - c\phi c\psi) + \hat{a}_z(s\phi s\theta) \\ -\hat{a}_x(c\phi c\theta c\psi - s\phi s\psi) + \hat{a}_y(c\phi c\theta s\psi + s\phi c\psi) - \hat{a}_z(c\phi s\theta) \\ 0 \end{bmatrix} \right\rangle \tag{B.5}$$

The second term of the inner product in Equation B.5 looks very similar to $\mathbf{R}_{\phi,\theta,\psi}\hat{\mathbf{a}}$. In fact, if we set $\mathbf{U} = \mathbf{R}_{\phi,\theta,\psi}\hat{\mathbf{a}}$, then it can be represented as a transformation $\mathbf{QU}$, where $\mathbf{Q}$ is some 3×3 transformation matrix. In the case of the ϕ component of the gradient, the inner product of Equation B.5 is equal to

$$\begin{aligned}
\frac{\partial d_i(\mathbf{q})}{\partial \phi} &= \frac{1}{d_i(\mathbf{q})} \begin{bmatrix} b_x - a_x \\ b_y - a_y \\ b_z - a_z \end{bmatrix}^T \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} \begin{bmatrix} \hat{a}_x \\ \hat{a}_y \\ \hat{a}_z \end{bmatrix} \\
&= \frac{1}{d_i(\mathbf{q})} \begin{bmatrix} b_x - a_x \\ b_y - a_y \\ b_z - a_z \end{bmatrix}^T \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} U_x \\ U_y \\ U_z \end{bmatrix} \\
&= \frac{1}{d_i(\mathbf{q})}((\mathbf{b} - \mathbf{a}) \times \mathbf{U})_z \\
&= \frac{1}{d_i(\mathbf{q})}(\mathbf{U} \times (\mathbf{a} - \mathbf{b}))_z
\end{aligned}$$

The ϕ component of the gradient is equal to the scalar z component of the cross product of $\mathbf{U}$ and $(\mathbf{a} - \mathbf{b})$. The θ and ψ components of the gradient are determined in similar manners. However, in these cases the transformation matrix $\mathbf{Q}$ is dependent of ϕ and (ϕ,θ) respectively.

$$\frac{\partial d_i(\mathbf{q})}{\partial \theta} = \frac{1}{d_i(\mathbf{q})} \begin{bmatrix} b_x - a_x \\ b_y - a_y \\ b_z - a_z \end{bmatrix}^T \begin{bmatrix} 0 & 0 & -c\phi \\ 0 & 0 & -s\phi \\ c\phi & s\phi & 0 \end{bmatrix} \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} \begin{bmatrix} \hat{a}_x \\ \hat{a}_y \\ \hat{a}_z \end{bmatrix}$$

$$= \frac{1}{d_i(\mathbf{q})} \left(\begin{bmatrix} b_x - a_x \\ b_y - a_y \\ b_z - a_z \end{bmatrix}^T \begin{bmatrix} 0 & 0 & -c\phi \\ 0 & 0 & -s\phi \\ c\phi & s\phi & 0 \end{bmatrix} \right) \begin{bmatrix} U_x \\ U_y \\ U_z \end{bmatrix}$$

$$= \frac{1}{d_i(\mathbf{q})} (\mathbf{R}_\phi^{-1}((\mathbf{b} - \mathbf{a}) \times \mathbf{U}))_y$$

$$= \frac{1}{d_i(\mathbf{q})} (\mathbf{R}_\phi^{-1}(\mathbf{U} \times (\mathbf{a} - \mathbf{b})))_y$$

The θ component of the gradient is equal to the y component of the vector $\mathbf{U} \times (\mathbf{a} - \mathbf{b})$ transformed by the matrix $\mathbf{R}_\phi^{-1}$.

$$\frac{\partial d_i(\mathbf{q})}{\partial \psi} = \frac{1}{d_i(\mathbf{q})} \begin{bmatrix} b_x - a_x \\ b_y - a_y \\ b_z - a_z \end{bmatrix}^T \begin{bmatrix} 0 & c\theta & -s\phi s\theta \\ -c\theta & 0 & c\phi s\theta \\ s\phi s\theta & -c\phi s\theta & 0 \end{bmatrix} \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} \begin{bmatrix} \hat{a}_x \\ \hat{a}_y \\ \hat{a}_z \end{bmatrix}$$

$$= \frac{1}{d_i(\mathbf{q})} \left(\begin{bmatrix} b_x - a_x \\ b_y - a_y \\ b_z - a_z \end{bmatrix}^T \begin{bmatrix} 0 & c\theta & -s\phi s\theta \\ -c\theta & 0 & c\phi s\theta \\ s\phi s\theta & -c\phi s\theta & 0 \end{bmatrix} \right) \begin{bmatrix} U_x \\ U_y \\ U_z \end{bmatrix}$$

$$= \frac{1}{d_i(\mathbf{q})} (\mathbf{R}_\theta^{-1}\mathbf{R}_\phi^{-1}((\mathbf{b} - \mathbf{a}) \times \mathbf{U}))_z$$

$$= \frac{1}{d_i(\mathbf{q})} (\mathbf{R}_\theta^{-1}\mathbf{R}_\phi^{-1}(\mathbf{U} \times (\mathbf{a} - \mathbf{b})))_z$$

The ψ component of the gradient is equal to the z component of the vector $\mathbf{U} \times (\mathbf{a} - \mathbf{b})$ transformed by the matrix $\mathbf{R}_\theta^{-1}\mathbf{R}_\phi^{-1}$.

The gradient of the distance function $d_i(\mathbf{q})$ is the 6×1 vector

$$\nabla d_i(\mathbf{q}) = \frac{1}{d_i(\mathbf{q})} \begin{bmatrix} (\mathbf{a}-\mathbf{b}) \\ (\mathbf{U}\times(\mathbf{a}-\mathbf{b}))_z \\ (\mathbf{R}_\phi^{-1}(\mathbf{U}\times(\mathbf{a}-\mathbf{b})))_y \\ (\mathbf{R}_\theta^{-1}\mathbf{R}_\phi^{-1}(\mathbf{U}\times(\mathbf{a}-\mathbf{b})))_z \end{bmatrix}$$

As noted in [15], the distance gradient reflects the lack of bi-invariance of *all* metrics in $SE(2)$ and $SE(3)$. A *left-invariant* metric in $SE(3)$ is one for which given any two points $p_1, p_2 \in SE(3)$, the distance between the points $d(p_1, p_2)$ is the same as $d(Tp_1, Tp_2)$ for all $T \in SE(3)$. This means changing the location of the world coordinate frame $\mathcal{F}_\mathcal{W}$ does not affect the metric. A *right-invariant* metric in $SE(3)$ is one for which given any two points $p_1, p_2 \in SE(3)$, the distance between the points $d(p_1, p_2)$ is the same as $d(p_1T, p_2T)$ for all $T \in SE(3)$. This means changing the location of the robots local coordinate frame $\mathcal{F}_\mathcal{A}$ does not affect the metric. It can be shown that no metric in $SE(3)$ can be left and right-invariant, or bi-invariant.

§C

Calculation of $\partial \mathbf{f_{ij}(q)}$ in SE(2)

In this appendix we are going to calculate the generalised gradient $\partial f_{ij}(\mathbf{q})$ of $f_{ij}(\mathbf{q}) = \tan^{-1}\left(\frac{ab_{iy}(\mathbf{q})}{ab_{ix}(\mathbf{q})}\right) + \tan^{-1}\left(\frac{ab_{jy}(\mathbf{q})}{ab_{jx}(\mathbf{q})}\right) - 2\theta - 2\vartheta$ for a convex *polygonal* obstacle $\mathcal{B}_i$ and a convex *polygonal* robot $\mathcal{A}$. Let's first review the notation used in this Appendix. The closest points between the robot $\mathcal{A}$ and the obstacle $\mathcal{B}_i$ are **a** and **b** respectively. When addressing the closest points between $\mathcal{A}$ and multiple obstacles $\mathcal{B}_i$ and $\mathcal{B}_j$, the points will be subscripted accordingly (i.e. $\mathbf{a}_i$, $\mathbf{a}_j$, $\mathbf{b}_i$, $\mathbf{b}_j$). The components of these closest points are denoted $\mathbf{a} = [a_x, a_y]^T$ and $\mathbf{b} = [b_x, b_y]^T$ ($\mathbf{a}_i = [a_{ix}, a_{iy}]^T$ and $\mathbf{b}_i = [b_{ix}, b_{iy}]^T$ if specifying a particular obstacle). Finally, the witness vector is $\mathbf{ab} = \mathbf{a} - \mathbf{b}$ ($\mathbf{ab}_i = \mathbf{a}_i - \mathbf{b}_i$ and $\mathbf{ab}_j = \mathbf{a}_j - \mathbf{b}_j$ if specifying a particular obstacle).

Recall Lemma A.1 from Appendix A which states that $\frac{\partial \mathbf{a}}{\partial x}$, $\frac{\partial \mathbf{a}}{\partial y}$, $\frac{\partial \mathbf{a}}{\partial \theta}$, $\frac{\partial \mathbf{b}}{\partial x}$, $\frac{\partial \mathbf{b}}{\partial y}$ and $\frac{\partial \mathbf{b}}{\partial \theta}$ exist if the vectors **ab** and **ba** defined by the single closest points **a** and **b** are in the interior of the normal cones to $\mathcal{A}$ and $\mathcal{B}_i$. In Sections C.1 and C.2 we determine the partials of **a** and **b** when the conditions of Lemma A.1 are satisfied. This information is summarised in Table C.1.

C.1 $\frac{\partial \mathbf{a}}{\partial x}$, $\frac{\partial \mathbf{a}}{\partial y}$ and $\frac{\partial \mathbf{a}}{\partial \theta}$

We consider the two cases when **a** is a vertex of $\mathcal{A}$ (**b** is either a vertex of or on the edge of $\mathcal{B}_i$) and **a** is on and edge of $\mathcal{A}$ (**b** is implicitly a vertex of $\mathcal{B}_i$).

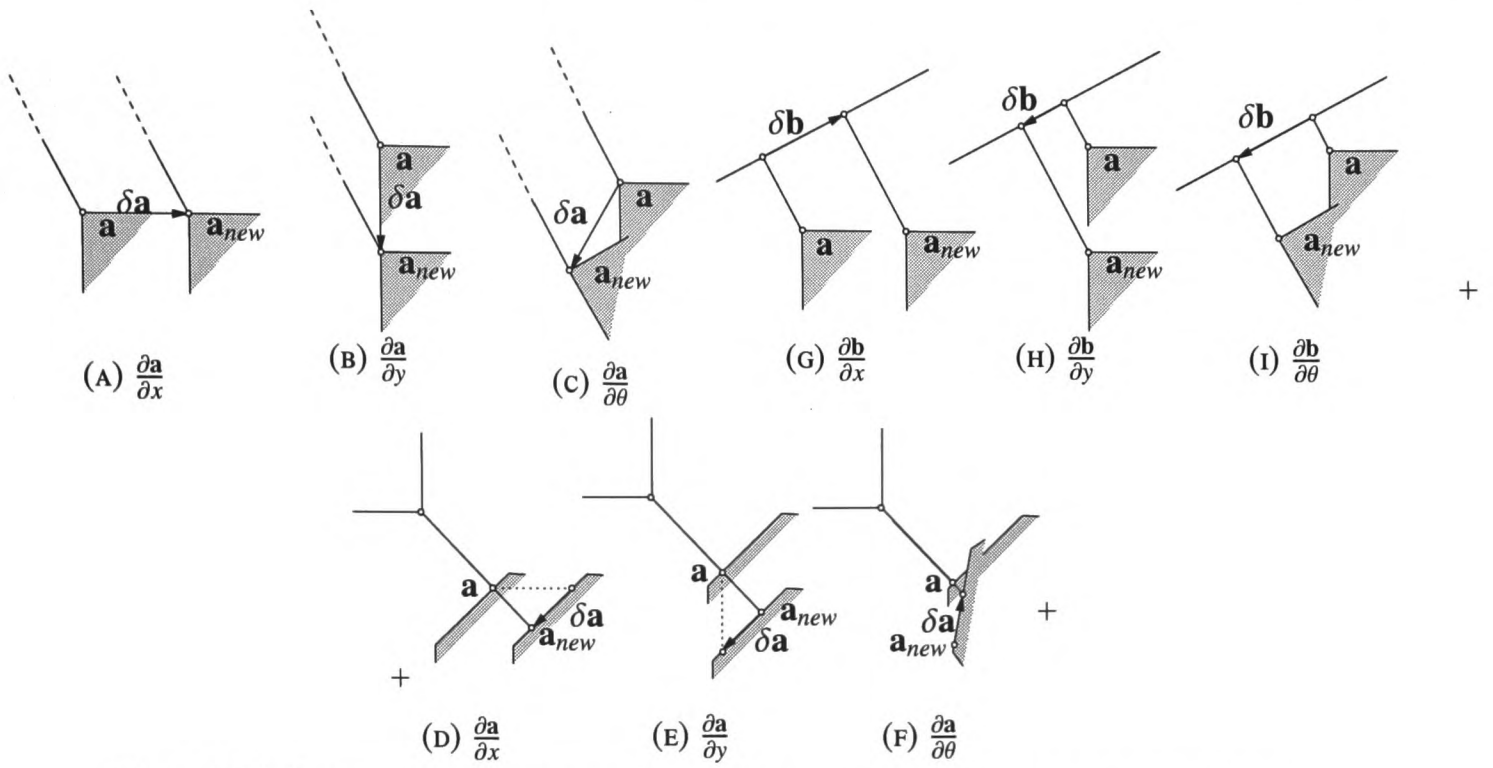


FIGURE C.1: DETERMINING PARTIAL DERIVATIVE OF **a** WHEN A SINGLE CLOSEST POINT EXISTS.

C.1.1 a is a Vertex of $\mathcal{A}$

Figures C.1(a), (b) and (c) show the case where **a** is a vertex of $\mathcal{A}$. We determine the various partial derivatives by adding a small component δx, δy or $\delta\theta$ to the configuration of $\mathcal{A}$ to arrive at a new location for the closest point $\mathbf{a}_{new}$. The resulting change in the location of the closest point $\delta\mathbf{a} = \mathbf{a}_{new} - \mathbf{a}$ is used to determine the partial derivative in the limit as δx, δy or $\delta\theta$ goes to zero. Let us first consider the partials with respect to x

$$\begin{aligned}\left[\frac{\partial a_x}{\partial x}, \frac{\partial a_y}{\partial x}\right]^T &= \lim_{\delta x \to 0} \frac{\mathbf{a}_{new} - \mathbf{a}}{\delta x} \\ &= \lim_{\delta x \to 0} \frac{(\mathbf{a} + [\delta x, 0]^T) - \mathbf{a}}{\delta x} \\ &= \lim_{\delta x \to 0} \frac{[\delta x, 0]^T}{\delta x} \\ &= [1, 0]^T\end{aligned}$$

Likewise for the partials with respect to y

$$\begin{aligned}\left[\frac{\partial a_x}{\partial y}, \frac{\partial a_y}{\partial y}\right]^T &= \lim_{\delta y \to 0} \frac{[0, \delta y]^T}{\delta y} \\ &= [0, 1]^T\end{aligned}$$

For the partials with respect to θ, we recall (from Appendices A and B) the notation $\hat{\mathbf{a}}$ which represents the coordinates of $\mathbf{a}$ in the robot's coordinate frame $\mathcal{F}_{\mathcal{A}}$ and $R_\theta\hat{\mathbf{a}} = [U_x, U_y]^T$ which is the vector from the origin of $\mathcal{F}_{\mathcal{A}}$ to $\hat{\mathbf{a}}$ in the world coordinate frame $\mathcal{F}_{\mathcal{W}}$.

$$\begin{aligned}
\left[\frac{\partial a_x}{\partial\theta}, \frac{\partial a_y}{\partial\theta}\right]^T &= \lim_{\delta\theta\to 0} \frac{([x,y]^T + R_{\theta+\delta\theta}[\hat{a}_x, \hat{a}_y]^T) - ([x,y]^T + R_\theta[\hat{a}_x, \hat{a}_y]^T)}{\delta\theta} \\
&= \lim_{\delta\theta\to 0} \frac{(R_{\theta+\delta\theta} - R_\theta)[\hat{a}_x, \hat{a}_y]^T}{\delta\theta} \\
&= \begin{bmatrix} -\sin\theta & -\cos\theta \\ \cos\theta & -\sin\theta \end{bmatrix} [\hat{a}_x, \hat{a}_y]^T \\
&= \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} [\hat{a}_x, \hat{a}_y]^T \\
&= [-U_y, U_x]^T
\end{aligned}$$

C.1.2 a is on an Edge of $\mathcal{A}$

To determine the partials when $\mathbf{a}$ is on and edge of $\mathcal{A}$, we recall the transformation between $\mathbf{a}$ in the world frame $\mathcal{F}_{\mathcal{W}}$ and $\hat{\mathbf{a}}$ in the robot's frame $\mathcal{F}_{\mathcal{A}}$

$$\mathbf{a} = R_\theta\hat{\mathbf{a}} + [x, y]^T$$

which allows us to determine the following transform between the partials in frames $\mathcal{F}_{\mathcal{W}}$ and $\mathcal{F}_{\mathcal{A}}$

$$\frac{\partial \mathbf{a}}{\partial i} = \frac{\partial R_\theta}{\partial i}\hat{\mathbf{a}} + R_\theta\frac{\partial \hat{\mathbf{a}}}{\partial i} + \frac{\partial}{\partial i}[x, y]^T \text{ for all } i \in \{x, y, \theta\} \tag{C.1}$$

We will take limits of $\delta\hat{\mathbf{a}}$ to determine the partials $\frac{\partial\hat{\mathbf{a}}}{\partial x}$ etc. and use Equation C.1 to determine $\frac{\partial\mathbf{a}}{\partial x}$ etc. We use the notation $\mathbf{e} = [-ab_y, ab_x]^T$ to represent a vector parallel to the edge of $\mathcal{A}$ and thus perpendicular to $\mathbf{ab}$ (i.e. $\langle \mathbf{e}, \mathbf{ab} \rangle = 0$). We use the notation $\hat{\mathbf{b}}$ (resp. $\hat{\mathbf{e}}$) to represent the location of $\mathbf{b}$ (resp. $\mathbf{e}$) in frame $\mathcal{F}_{\mathcal{A}}$.

Let us first address the partials with respect to x

$$\begin{aligned}
\left[\frac{\partial \hat{a}_x}{\partial x}, \frac{\partial \hat{a}_y}{\partial x}\right]^T &= \lim_{\delta x \to 0} \frac{\hat{\mathbf{a}}_{new} - \hat{\mathbf{a}}}{\delta x} \\
&= \lim_{\delta x \to 0} \frac{\frac{(\hat{\mathbf{b}}_{new} - \hat{\mathbf{b}}) \cdot \hat{\mathbf{e}}}{\|\hat{\mathbf{e}}\|}\hat{\mathbf{e}}}{\delta x} \text{ (project } \delta\hat{\mathbf{a}} \text{ onto } \hat{\mathbf{e}}) \\
&= \lim_{\delta x \to 0} \frac{(R_\theta^{-1}(\mathbf{b} - [x+\delta x, y]^T) - R_\theta^{-1}(\mathbf{b} - [x, y]^T)) \cdot R_\theta^{-1}\mathbf{e}}{\|\mathbf{e}\|\,\delta x} R_\theta^{-1}\mathbf{e} \\
&= \lim_{\delta x \to 0} \frac{R_\theta^{-1}(-[\delta x, 0]^T) \cdot R_\theta^{-1}\mathbf{e}}{\|\mathbf{e}\|\,\delta x} R_\theta^{-1}\mathbf{e} \\
&= \lim_{\delta x \to 0} \frac{\delta x a b_y}{\delta x \,\|\mathbf{ab}\|^2} R_\theta^{-1}[-ab_y, ab_x]^T \\
&= R_\theta^{-1}\left[\frac{-ab_y^2}{\|\mathbf{ab}\|^2}, \frac{ab_x ab_y}{\|\mathbf{ab}\|^2}\right]^T
\end{aligned}$$

and after transforming according to Equation C.1

$$\left[\frac{\partial a_x}{\partial y}, \frac{\partial a_y}{\partial y}\right]^T = \left[1 - \frac{ab_y^2}{\|\mathbf{ab}\|^2}, \frac{ab_x ab_y}{\|\mathbf{ab}\|^2}\right]^T$$

Likewise for the partials with respect to y

$$\begin{aligned}
\left[\frac{\partial \hat{a}_x}{\partial y}, \frac{\partial \hat{a}_y}{\partial y}\right]^T &= \lim_{\delta y \to 0} \frac{-\delta y a b_x}{\delta y \,\|\mathbf{ab}\|^2} R_\theta^{-1}[-ab_y, ab_x]^T \\
&= R_\theta^{-1}\left[\frac{ab_x ab_y}{\|\mathbf{ab}\|^2}, \frac{-ab_x^2}{\|\mathbf{ab}\|^2}\right]^T
\end{aligned}$$

and after transforming according to Equation C.1

$$\left[\frac{\partial a_x}{\partial y}, \frac{\partial a_y}{\partial y}\right]^T = \left[\frac{ab_x ab_y}{\|\mathbf{ab}\|^2}, 1 - \frac{ab_x^2}{\|\mathbf{ab}\|^2}\right]^T$$

Likewise for the partials with respect to θ

$$\left[\frac{\partial \hat{a}_x}{\partial \theta}, \frac{\partial \hat{a}_y}{\partial \theta}\right]^T = \lim_{\delta\theta \to 0} \frac{(R_{\theta+\delta\theta}^{-1}(\mathbf{U}-\mathbf{ab}) - R_\theta^{-1}(\mathbf{U}-\mathbf{ab})) \cdot R_\theta^{-1}\mathbf{e}}{\|\mathbf{e}\|\,\delta\theta} R_\theta^{-1}\mathbf{e}$$

$$= \frac{R_\theta^{-1}\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}(\mathbf{U}-\mathbf{ab}) \cdot R_\theta^{-1}[-ab_y, ab_x]^T}{\|\mathbf{ab}\|^2} R_\theta^{-1}[-ab_y, ab_x]^T$$

$$= \frac{-\mathbf{ab} \cdot (\mathbf{U}-\mathbf{ab})}{\|\mathbf{ab}\|^2} R_\theta^{-1}[-ab_y, ab_x]^T$$

and after transforming according to Equation C.1

$$\left[\frac{\partial a_x}{\partial y}, \frac{\partial a_y}{\partial y}\right]^T = \left[ab_y \frac{\mathbf{ab} \cdot (\mathbf{U}-\mathbf{ab})}{\|\mathbf{ab}\|^2}, -ab_x \frac{\mathbf{ab} \cdot (\mathbf{U}-\mathbf{ab})}{\|\mathbf{ab}\|^2}\right]^T$$

C.2 $\frac{\partial \mathbf{b}}{\partial x}$, $\frac{\partial \mathbf{b}}{\partial y}$ and $\frac{\partial \mathbf{b}}{\partial \theta}$

We consider the two cases when **b** is a vertex of $\mathcal{B}_i$ (**a** is either a vertex of or on the edge of $\mathcal{A}$) and **b** is on and edge of $\mathcal{B}_i$ (**a** is implicitly a vertex of $\mathcal{A}$).

C.2.1 b is a Vertex of $\mathcal{B}_i$

This is the most trivial case where all partials evaluate to zero.

$$\left[\frac{\partial b_x}{\partial x}, \frac{\partial b_y}{\partial x}\right]^T = \left[\frac{\partial b_x}{\partial y}, \frac{\partial b_y}{\partial y}\right]^T = \left[\frac{\partial b_x}{\partial \theta}, \frac{\partial b_y}{\partial \theta}\right]^T = [0,0]^T$$

C.2.2 b is on an Edge of $\mathcal{B}_i$

To determine the partials with respect to x when **b** is on and edge of $\mathcal{B}_i$, we simply project $\left[\frac{\partial a_x}{\partial x}, \frac{\partial a_y}{\partial x}\right]^T$ for the case when **a** is a vertex of $\mathcal{A}$ onto the edge of $\mathcal{B}_i$.

$$\begin{aligned}\left[\frac{\partial b_x}{\partial x}, \frac{\partial b_y}{\partial x}\right]^T &= \frac{\left[\frac{\partial a_x}{\partial x}, \frac{\partial a_y}{\partial x}\right]^T \cdot \mathbf{e}}{\|\mathbf{e}\|^2}\mathbf{e} \\ &= \frac{[1,0]^T \cdot [-ab_y, ab_x]^T}{\|\mathbf{ab}\|^2}[-ab_y, ab_x]^T \\ &= [\frac{ab_y^2}{\|\mathbf{ab}\|^2}, \frac{-ab_x ab_y}{\|\mathbf{ab}\|^2}]^T\end{aligned}$$

Likewise for the partials with respect to y

$$\begin{aligned}\left[\frac{\partial b_x}{\partial y}, \frac{\partial b_y}{\partial y}\right]^T &= \frac{[0,1]^T \cdot [-ab_y, ab_x]^T}{\|\mathbf{ab}\|^2}[-ab_y, ab_x]^T \\ &= [\frac{-ab_x ab_y}{\|\mathbf{ab}\|^2}, \frac{ab_x^2}{\|\mathbf{ab}\|^2}]^T\end{aligned}$$

Likewise for the partials with respect to θ

$$\begin{aligned}\left[\frac{\partial b_x}{\partial \theta}, \frac{\partial b_y}{\partial \theta}\right]^T &= \frac{\left[-U_y, U_x\right]^T \cdot [-ab_y, ab_x]^T}{\|\mathbf{ab}\|^2}[-ab_y, ab_x]^T \\ &= [\frac{-U_y ab_y^2 - U_x ab_x ab_y}{\|\mathbf{ab}\|^2}, \frac{U_y ab_x ab_y + U_x ab_x^2}{\|\mathbf{ab}\|^2}]^T\end{aligned}$$

C.3 Calculation of $\partial(\tan^{-1}\left(\frac{ab_{iy}(\mathbf{q})}{ab_{ix}(\mathbf{q})}\right) + \tan^{-1}\left(\frac{ab_{jy}(\mathbf{q})}{ab_{jx}(\mathbf{q})}\right) - 2\theta - 2\vartheta)$

Recall from Appendix A that the terms a_x, b_x etc. can in general be nonsmooth, even when a single closest points **a** on $\mathcal{A}$ and **b** on $\mathcal{B}_i$ exist. We show in this section that $\tan^{-1}\left(\frac{ab_{iy}(\mathbf{q})}{ab_{ix}(\mathbf{q})}\right) + \tan^{-1}\left(\frac{ab_{jy}(\mathbf{q})}{ab_{jx}(\mathbf{q})}\right) - 2\theta - 2\vartheta$ is also nonsmooth, and describe its structure. We first note that the partials of the constant $\partial(2\vartheta) = \mathbf{0}$ and the partial of the orientation $\partial(2\theta) = [0,0,2]^T$ are both singletons, so the nonsmoothness is a consequence of

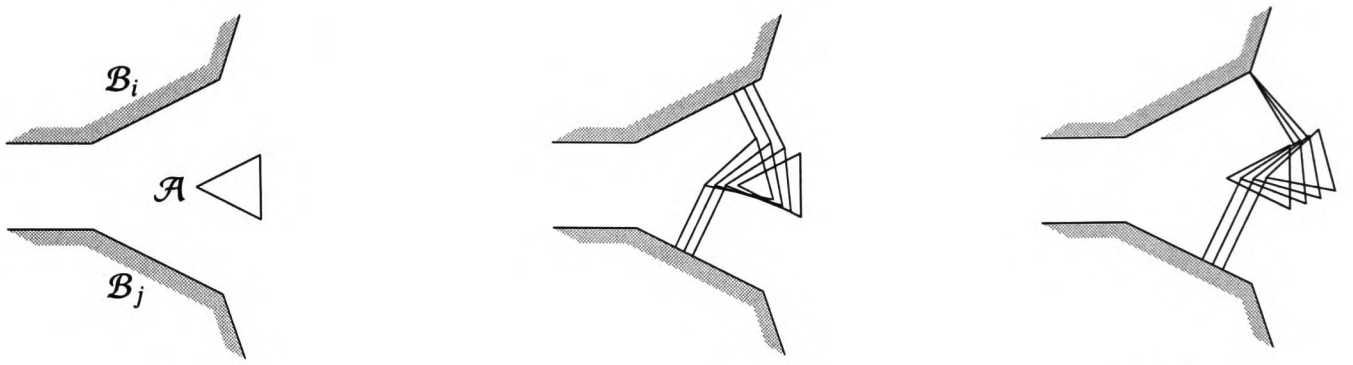
TABLE C.1: VALUES OF VARIOUS PARTIAL DERIVATIVES FOR VERTEX AND EDGE WITNESS POINTS.

a is a vertex of $\mathcal{A}$			
$\frac{\partial a_x}{\partial x}$	1	$\frac{\partial a_y}{\partial x}$	0
$\frac{\partial a_x}{\partial y}$	0	$\frac{\partial a_y}{\partial y}$	1
$\frac{\partial a_x}{\partial \theta}$	$-U_y$	$\frac{\partial a_y}{\partial \theta}$	U_x
a is on an edge of $\mathcal{A}$			
$\frac{\partial a_x}{\partial x}$	$1 - \frac{ab_y^2}{\|\mathbf{ab}\|^2}$	$\frac{\partial a_y}{\partial x}$	$\frac{ab_x ab_y}{\|\mathbf{ab}\|^2}$
$\frac{\partial a_x}{\partial y}$	$\frac{ab_x ab_y}{\|\mathbf{ab}\|^2}$	$\frac{\partial a_y}{\partial y}$	$1 - \frac{ab_x^2}{\|\mathbf{ab}\|^2}$
$\frac{\partial a_x}{\partial \theta}$	$ab_y \frac{\mathbf{ab} \cdot (\mathbf{U} - \mathbf{ab})}{\|\mathbf{ab}\|^2}$	$\frac{\partial a_y}{\partial \theta}$	$-ab_x \frac{\mathbf{ab} \cdot (\mathbf{U} - \mathbf{ab})}{\|\mathbf{ab}\|^2}$
b is a vertex of $\mathcal{B}_i$			
$\frac{\partial b_x}{\partial x}$	0	$\frac{\partial b_y}{\partial x}$	0
$\frac{\partial b_x}{\partial y}$	0	$\frac{\partial b_y}{\partial y}$	0
$\frac{\partial b_x}{\partial \theta}$	0	$\frac{\partial b_y}{\partial \theta}$	0
b is on an edge of $\mathcal{B}_i$			
$\frac{\partial b_x}{\partial x}$	$\frac{ab_y^2}{\|\mathbf{ab}\|^2}$	$\frac{\partial b_y}{\partial x}$	$\frac{-ab_x ab_y}{\|\mathbf{ab}\|^2}$
$\frac{\partial b_x}{\partial y}$	$\frac{-ab_x ab_y}{\|\mathbf{ab}\|^2}$	$\frac{\partial b_y}{\partial y}$	$\frac{ab_x^2}{\|\mathbf{ab}\|^2}$
$\frac{\partial b_x}{\partial \theta}$	$\frac{-U_y ab_y^2 - U_x ab_x ab_y}{\|\mathbf{ab}\|^2}$	$\frac{\partial b_y}{\partial \theta}$	$\frac{U_y ab_x ab_y + U_x ab_x^2}{\|\mathbf{ab}\|^2}$

the nonsmoothness of a_x, b_x etc. By the calculus of the generalised gradient we know

$$\partial f_{ij}(\mathbf{q}) \subseteq \left(1 + \left(\frac{ab_{iy}}{ab_{ix}}\right)^2\right)^{-1} \partial\left(\frac{ab_{iy}}{ab_{ix}}\right) + \left(1 + \left(\frac{ab_{jy}}{ab_{jx}}\right)^2\right)^{-1} \partial\left(\frac{ab_{jy}}{ab_{jx}}\right) - \begin{bmatrix} 0 \\ 0 \\ 2 \end{bmatrix}$$

$$\subseteq \frac{ab_{ix}\partial ab_{iy} - ab_{iy}\partial ab_{ix}}{ab_{ix}^2 + ab_{iy}^2} + \frac{ab_{jx}\partial ab_{jy} - ab_{jy}\partial ab_{jx}}{ab_{jx}^2 + ab_{jy}^2} - \begin{bmatrix} 0 \\ 0 \\ 2 \end{bmatrix}$$



(A) - NONSMOOTH $f_{ij}(\mathbf{q})$ (B) - CASE 5 (C) - CASE 8

FIGURE C.2: EXAMPLES OF $\lim f_{ij}(\mathbf{q}_k)$ FOR SEQUENCES $\mathbf{q}_k \to \mathbf{q}$. (A) $f_{ij}(\mathbf{q})$ IS NONSMOOTH. (B) $\mathcal{A}(\mathbf{q}_k)$ APPROACHES WITH VERTEX-EDGE AND VERTEX-EDGE WITNESS POINTS (CASE 5). (C) $\mathcal{A}(\mathbf{q}_k)$ APPROACHES WITH EDGE-VERTEX AND VERTEX-EDGE WITNESS POINTS (CASE 8).

Since this is a bit intractable, we return to the definition of the generalised gradient $\partial\phi(\mathbf{x})$ as the convex hull of the gradient vector limits

$$\partial\phi(\mathbf{x}) = \text{co}\{\lim \nabla\phi(\mathbf{x}_k) : \mathbf{x}_k \to \mathbf{x}, \mathbf{x}_k \notin S, \mathbf{x}_k \notin \Omega_\phi\}$$

Recalling the approach used in Appendix A we let the set $S = S_1 \cup S_2$ be the union of the sets of points S_1 where the witness vectors **ab**/**ba** violate Conditions A.3 (i.e. are not in the interior of the normal cone) and S_2 where $\mathcal{A}$ and $\mathcal{B}_i$ share multiple closest points.

Sequences $\mathbf{q}_k$ where $\nabla a_x(\mathbf{q}_k)$, $\nabla b_x(\mathbf{q}_k)$ etc. exist can be broken into a nine subcases where the witness points $\mathbf{a}_i$ and $\mathbf{b}_i$ (resp. $\mathbf{a}_j$ and $\mathbf{b}_j$) are either vertex-vertex or vertex-edge combinations. The nine cases are enumerated in Table C.2. In the following sections we calculate $\lim f_{ij}(\mathbf{q}_k)$ for the nine case using the entries in Table C.1. Figure C.2(a) shows a case where $f_{ij}(\mathbf{q})$ is nonsmooth, and Figures C.2(b) and (c) show a few $\mathcal{A}(\mathbf{q}_k)$ for sequences $\mathbf{q}_k \to \mathbf{q}$. (b) shows an example of Case 5 and (c) shows an example of Case 8. The most interesting aspect of this analysis is that $f_{ij}(\mathbf{q})$ can in fact be nonsmooth in all coordinates at the same time, unlike $d_i(\mathbf{q})$ which is smooth in x and y. The generalised gradient $\partial f_{ij}(\mathbf{q})$ is then determined by taking the convex hull of the appropriate limit terms.

C.3.1 Case 1

In all cases, all the terms a_x, b_x etc. are smooth and thus all the gradient limits $\lim \nabla a_x(\mathbf{q}_k)$, $\lim \nabla b_x(\mathbf{q}_k)$ converge to single points $\nabla a_x(\mathbf{q})$, $\nabla b_x(\mathbf{q})$ for all sequences $\mathbf{q}_k$. $\lim f_{ij}(\mathbf{q}_k)$ is thus easily determined by using

TABLE C.2: THE NINE CASES USED TO DETERMINE $\partial f_{ij}(\mathbf{q})$.

case	$\mathbf{a}_i$	$\mathbf{b}_i$	$\mathbf{a}_j$	$\mathbf{b}_j$
1	vertex	vertex	vertex	vertex
2	vertex	vertex	vertex	edge
3	vertex	vertex	edge	vertex
4	vertex	edge	vertex	vertex
5	vertex	edge	vertex	edge
6	vertex	edge	edge	vertex
7	edge	vertex	vertex	vertex
8	edge	vertex	vertex	edge
9	edge	vertex	edge	vertex

the values in Table C.1.

$$\lim f_{ij}(\mathbf{q}_k) = \frac{ab_{ix}\nabla ab_{iy} - ab_{iy}\nabla ab_{ix}}{\|\mathbf{ab}_i\|^2} + \frac{ab_{jx}\nabla ab_{jy} - ab_{jy}\nabla ab_{jx}}{\|\mathbf{ab}_j\|^2} - \begin{bmatrix}0\\0\\2\end{bmatrix}$$

$$= \frac{ab_{ix}\left([0,1,U_{ix}]^T - [\mathbf{0}]^T\right) + ab_{iy}\left(\left[1,0,-U_{iy}\right]^T - [\mathbf{0}]^T\right)}{\|\mathbf{ab}_i\|^2}$$

$$- \frac{ab_{jy}\left(\left[0,1,U_{jx}\right]^T - [\mathbf{0}]^T\right) + ab_{jy}\left(\left[1,0,-U_{jy}\right]^T - [\mathbf{0}]^T\right)}{\|\mathbf{ab}_j\|^2} - \begin{bmatrix}0\\0\\2\end{bmatrix}$$

$$= \frac{1}{\|\mathbf{ab}_i\|^2}\begin{bmatrix}-ab_{iy}\\ab_{ix}\\\mathbf{U}_i\cdot\mathbf{ab}_i\end{bmatrix} + \frac{1}{\|\mathbf{ab}_j\|^2}\begin{bmatrix}-ab_{jy}\\ab_{jx}\\\mathbf{U}_j\cdot\mathbf{ab}_j\end{bmatrix} - \begin{bmatrix}0\\0\\2\end{bmatrix}$$

C.3.2 Case 2

$$\lim f_{ij}(\mathbf{q}_k) = \frac{1}{\|\mathbf{ab}_i\|^2}\begin{bmatrix}-ab_{iy}\\ab_{ix}\\\mathbf{U}_i\cdot\mathbf{ab}_i\end{bmatrix} - \begin{bmatrix}0\\0\\2\end{bmatrix}$$

C.3.3 Case 3

$$\lim f_{ij}(\mathbf{q}_k) = \frac{1}{\|\mathbf{ab}_i\|^2}\begin{bmatrix} -ab_{iy} \\ ab_{ix} \\ \mathbf{U}_i \cdot \mathbf{ab}_i \end{bmatrix} + \frac{1}{\|\mathbf{ab}_j\|^2}\begin{bmatrix} 0 \\ 0 \\ -\mathbf{U}_j \cdot \mathbf{ab}_j + \|\mathbf{ab}_j\|^2 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ 2 \end{bmatrix}$$

C.3.4 Case 4

$$\lim f_{ij}(\mathbf{q}_k) = \frac{1}{\|\mathbf{ab}_j\|^2}\begin{bmatrix} -ab_{jy} \\ ab_{jx} \\ \mathbf{U}_j \cdot \mathbf{ab}_j \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ 2 \end{bmatrix}$$

C.3.5 Case 5

$$\lim f_{ij}(\mathbf{q}_k) = -\begin{bmatrix} 0 \\ 0 \\ 2 \end{bmatrix}$$

C.3.6 Case 6

$$\lim f_{ij}(\mathbf{q}_k) = \frac{1}{\|\mathbf{ab}_j\|^2}\begin{bmatrix} 0 \\ 0 \\ -\mathbf{U}_j \cdot \mathbf{ab}_j + \|\mathbf{ab}_j\|^2 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ 2 \end{bmatrix}$$

C.3.7 Case 7

$$\lim f_{ij}(\mathbf{q}_k) = \frac{1}{\|\mathbf{ab}_i\|^2}\begin{bmatrix} 0 \\ 0 \\ -\mathbf{U}_i \cdot \mathbf{ab}_i + \|\mathbf{ab}_i\|^2 \end{bmatrix} + \frac{1}{\|\mathbf{ab}_j\|^2}\begin{bmatrix} -ab_{jy} \\ ab_{jx} \\ \mathbf{U}_j \cdot \mathbf{ab}_j \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ 2 \end{bmatrix}$$

C.3.8 Case 8

$$\lim f_{ij}(\mathbf{q}_k) = \frac{1}{\|\mathbf{ab}_i\|^2}\begin{bmatrix} 0 \\ 0 \\ -\mathbf{U}_i \cdot \mathbf{ab}_i + \|\mathbf{ab}_i\|^2 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ 2 \end{bmatrix}$$

C.3.9 Case 9

$$\lim f_{ij}(\mathbf{q}_k) = \frac{1}{\|\mathbf{ab}_i\|^2}\begin{bmatrix} 0 \\ 0 \\ -\mathbf{U}_i \cdot \mathbf{ab}_i + \|\mathbf{ab}_i\|^2 \end{bmatrix} + \frac{1}{\|\mathbf{ab}_j\|^2}\begin{bmatrix} 0 \\ 0 \\ -\mathbf{U}_j \cdot \mathbf{ab}_j + \|\mathbf{ab}_j\|^2 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ 2 \end{bmatrix}$$

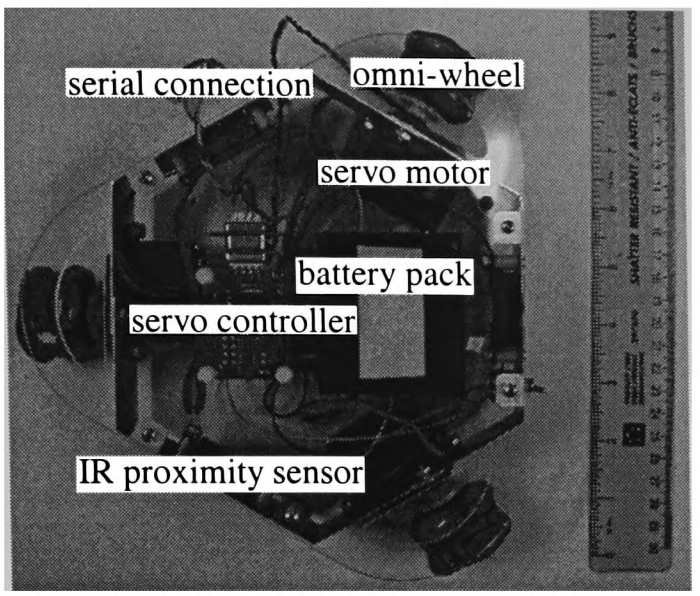
§D

Visual Servoing of the Palm Pilot Robot

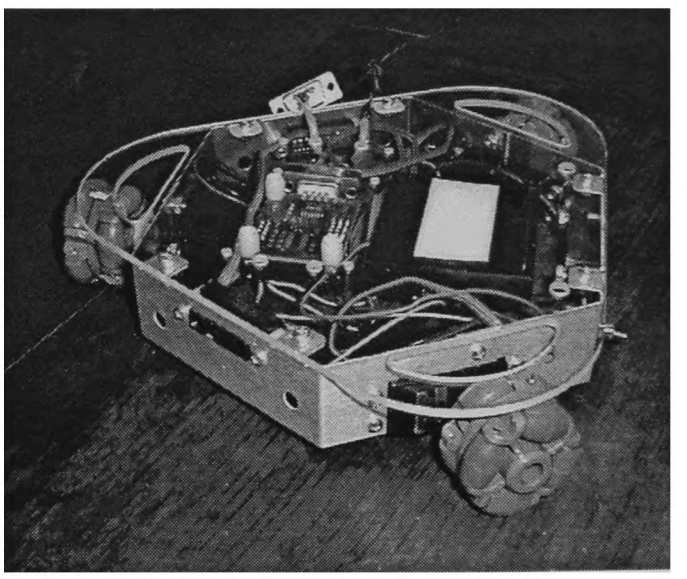
In this chapter we describe the experimental apparatus designed and constructed by the author, from scratch, to verify the theory in the bulk of the thesis.

D.1 The Robot

The mobile robot used for our experiments is the Acroname Palm Pilot Robot Kit (PPRK) developed and licensed by Carnegie Mellon University [52]. A photo of the PPRK is shown in Figure D.1. Figure D.1(a) shows a top-down view of the PPRK and Figure D.1(b) shows a profile view. In Figure D.1(a) we can see the PPRK's innards through the clear acrylic top panel. Spaced equally around the circumference of the PPRK are three omni-wheels which allow rolling motion transverse to the rotation direction. The wheels are attached to readily available servo motors which have been modified to allow a full 360° degrees of rotation. The velocity of each wheel is under the open-loop control of a small, programmable interface controller-based servo controller. This servo controller is meant to receive signals via a serial link with a Palm Pilot PDA (purchased separately), but is equally able to communicate serially through any RS-232 link. Since each wheel is controlled individually and can roll transverse to the rotation direction, the PPRK is capable of holonomic motion.



(A)



(B)

FIGURE D.1: THE ACRONAME PALM PILOT ROBOT KIT (PPRK) ROBOT. (A) LABELLED PHOTOGRAPH SHOWING THE MAIN PARTS OF THE PPRK. (B) PHOTOGRAPH OF THE PPRK FROM AN ALTERNATE ANGLE.

D.1.1 Calibrating the PPRK

Out of the box, the PPRK's control software comes with default servo velocity control parameters that are equal for each servo. These parameters effectively constitute a simple model of the servo

$$s_i = f(v_i)$$

where v_i is the resultant velocity achieved when a control signal s_i in the range integer range $[0, 255]$ is sent to the servo controller board. Upon establishing a communication link with the PPRK for the first time, it was immediately apparent that the default model parameters were not accurate enough. Near $v_i = 0$, for two of the servos, a control signal that was meant to cause a positive velocity in fact generated a negative velocity, and the maximum v_i for each servo varied ±20% from the velocity given by the default model—we needed to generate our own models.

To generate models for each servo individually, we needed the ability to measure the servo velocity v_i for a given control signal s_i. To obtain this velocity measurement, we attached a HEDS-5140 optical code wheel with 500 cycles/revolution to the shaft of the servo. When the code wheel rotates in conjunction with a HEDS-9140 optical encoder, the outputs of the HEDS-9140 pulse with a frequency proportional to rotation rate of the HEDS-5140. This frequency was measured using a frequency counter, and logged via a serial link with the frequency counter. The frequency was then converted to a velocity using the 500 cycles/revolution factor of the optical code wheel. This system allowed us to send various control

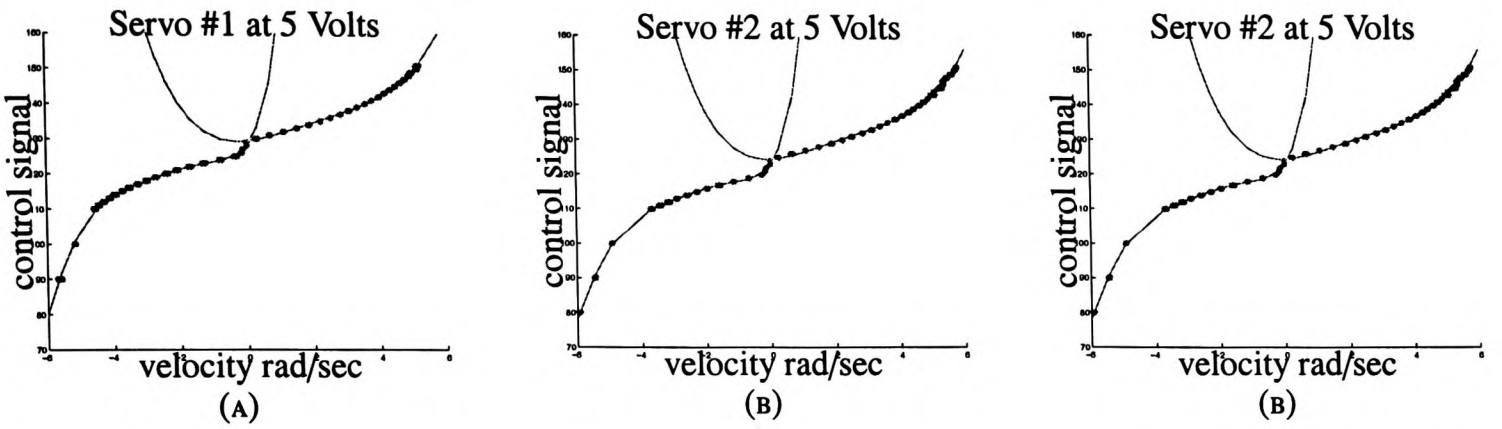


FIGURE D.2: EXPERIMENTALLY DETERMINED SERVO VELOCITY MODELS. A VERY GOOD FIT TO THE DATA WAS FOUND USING A 6-DEGREE POLYNOMIAL $s_i^{vi}(v)$ AND A 4-DEGREE POLYNOMIAL $s_i^{iv}(v)$. THE VELOCITY COORDINATE OF THE INTERSECTION OF THE TWO CURVES IS USED TO SEPARATE THE DOMAIN OF VELOCITIES.

signals to the servo controller board and measure the resulting velocity in an automated manner. The data collected during these tests is shown in Figures D.2(a) through D.2(c). A very good fit to the data (•) was found using the 6-degree polynomials $s_i^{vi}(v)$ and 4-degree polynomials $s_i^{iv}(v)$ given in Equations D.1 through D.3, which are also plotted in Figures D.2(a) through D.2(c).

$$
\begin{aligned}
s_1^{vi}(v) &= 0.017v^6 + 0.333v^5 + 2.39v^4 + 8.17v^3 + 13.5v^2 + 12.1v + 128 \\
s_1^{iv}(v) &= 0.0683v^4 - 0.545v^3 + 1.62v^2 + 1.01v + 129 \\
s_1^{vi}(v) &= s_1^{iv}(v) \text{ at } (0.105, 129)
\end{aligned} \tag{D.1}
$$

$$
\begin{aligned}
s_2^{vi}(v) &= 0.0185v^6 + 0.357v^5 + 2.56v^4 + 8.79v^3 + 14.7v^2 + 13.1v + 123 \\
s_2^{iv}(v) &= 0.0719v^4 - 0.639v^3 + 2.04v^2 + 0.419v + 124 \\
s_2^{vi}(v) &= s_2^{iv}(v) \text{ at } (0.099, 125)
\end{aligned} \tag{D.2}
$$

$$
\begin{aligned}
s_3^{vi}(v) &= 0.0203v^6 + 0.381v^5 + 2.63v^4 + 8.54v^3 + 13.3v^2 + 11.3v + 125 \\
s_3^{iv}(v) &= 0.0906v^4 - 0.869v^3 + 2.84v^2 - 0.445v + 127 \\
s_3^{vi}(v) &= s_3^{iv}(v) \text{ at } (0.109, 127)
\end{aligned} \tag{D.3}
$$

The intersection point of the two polynomials $s_i^{vi}(v) = s_i^{iv}(v)$ is used to determine which polynomial to use when calculating control signal s_i to drive the wheels at velocity v_i. Using these modified models, we were able to precisely control the instantaneous velocity of the PPRK.

D.2 The Visual Servoing System

The control laws developed in Chapter 5 are of the form $\dot{\mathbf{q}} = \alpha\mathbf{a} + \beta\mathbf{b}$, where $\dot{\mathbf{q}}$ is the desired velocity. In order to implement this law, we need to know the configuration $\mathbf{q}$ of the PPRK, the distance $d_i(\mathbf{q})$ to each obstacle $\mathcal{B}_i$, and the witness points $\mathbf{a}_i$ and $\mathbf{b}_i$ for each $\mathcal{B}_i$. The apparatus used to facilitate this is a PC framegrabber-based visual servoing system. The system comprises (I) a ceiling-mounted CCD camera, (II) a control PC running Linux which is equipped with a framegrabber, (III) a serial connection with the PPRK and (IV) appropriate cabling and connections. (I) and (IV) are self-explanatory, but (II) and (III) bear some further explanation.

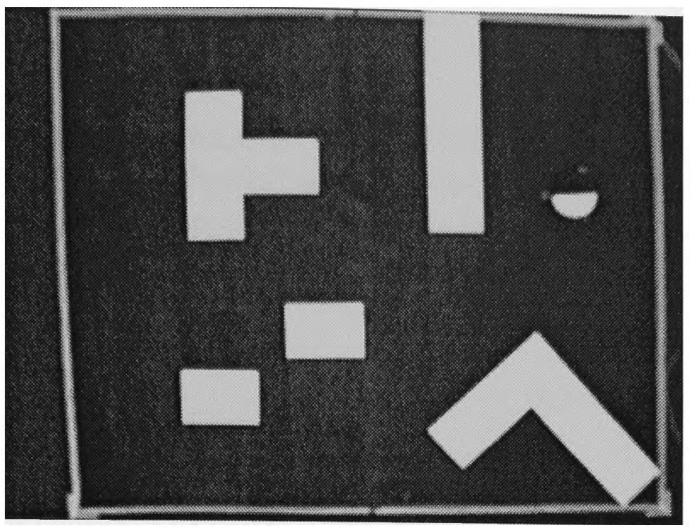
D.2.1 (I) – Control PC

The control PC has four main tasks: (i) determine the configuration $\mathbf{q}$ of the PPRK, (ii) determine the distances $d_i(\mathbf{q})$, $d_j(\mathbf{q})$ etc. to obstacles in the environment, (iii) implement our planning algorithm based on this configuration and distance information, and (iv) control the PPRK's motion via the serial connection discussed in Section D.2.2. (iii) is self-explanatory, but the rest bear some further explanation.

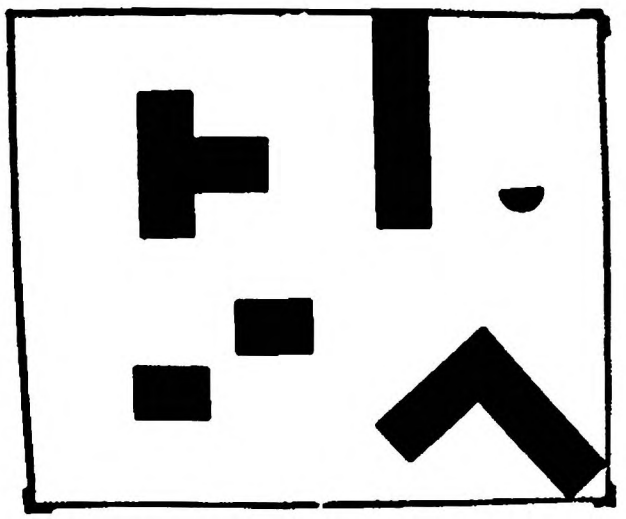
D.2.1.1 (i) – Determining the PPRK's Configuration

To determine the configuration of the PPRK, we placed a black and white pattern on top of the PPRK and determined the configuration of the pattern from the image snapped by the framegrabber. The pattern selected was a white semi-circle, which contrasts nicely the dark surface of the workspace. For each 640×480 bitmap image supplied by the framegrabber (see Figure D.3(a)) we first threshold the image into a black and white image (see Figure D.3(b)) and then use a "cross" transform [61] to detect edges in the thresholded image (see Figure D.3(c)). In the resulting edge image, the white semi-circle pattern appears as a black semi-circle outline, consisting of an arc feature and a straight line feature. The centre of the arc feature corresponds to the position (x, y) of the PPRK and is detected using the Hough circle transform. The orientation of the line feature corresponds to the orientation θ of the PPRK and is detected using the Hough line transform. The Hough transform warrants a brief explanation.

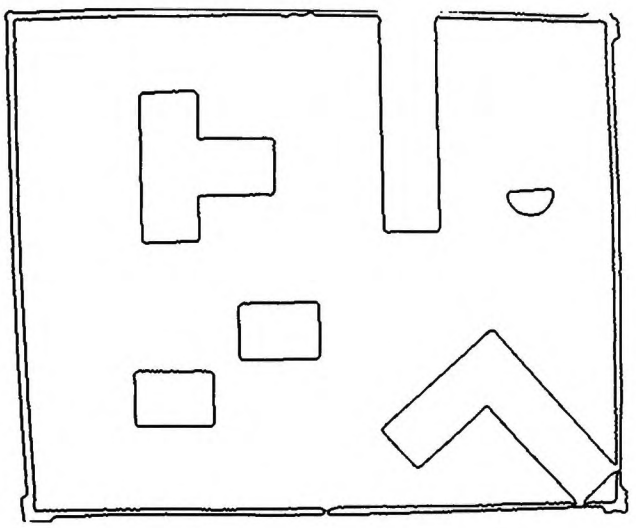
The Hough transform is a "point-to-curve" transformation from the image space to a suitable parameter space. Take this standard parameterisation of a circle with fixed radius r (we know the radius of the



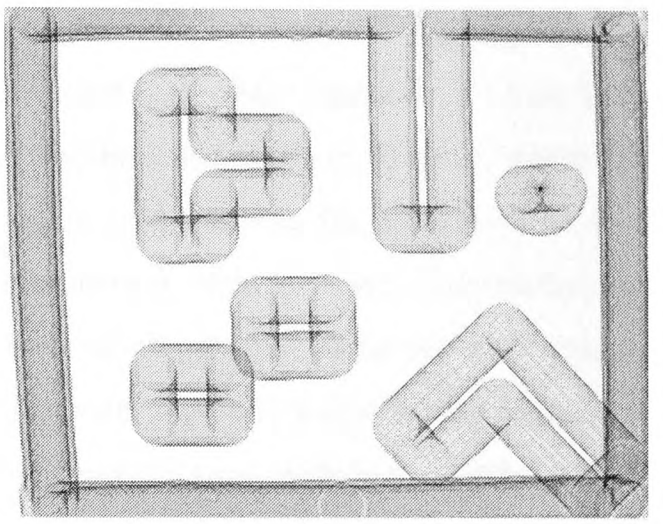
(A)



(B)



(C)



(D)

FIGURE D.3: FINDING THE PPRK POSITION VIA THE HOUGH CIRCLE TRANSFORM. (A) RAW GRAYSCALE IMAGE CAPTURED BY THE FRAMEGRABBER. (B) BLACK AND WHITE THRESHOLDED IMAGE. (C) EDGES DETECTED IN THE THRESHOLDED IMAGE USING A "CROSS" TRANSFORM. (D) HOUGH CIRCLE TRANSFORM (ACCUMULATOR REPRESENTATION) OF THE EDGES IMAGE (WHITE→BLACK = EMPTY→FULL).

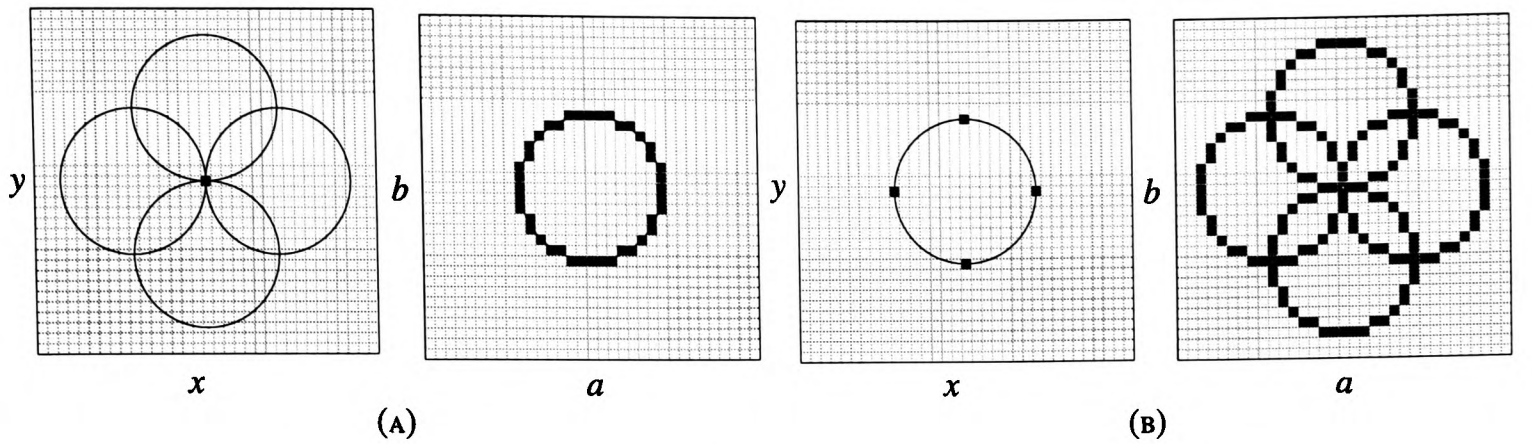


FIGURE D.4: ILLUSTRATION OF THE HOUGH CIRCLE TRANSFORM. (A) HOUGH CIRCLE TRANSFORM OF A SINGLE PIXEL. (B) HOUGH CIRCLE TRANSFORM OF MULTIPLE PIXELS. THE MAXIMUM IN THE (a, b) SPACE CORRESPONDS TO THE CENTRE OF THE DETECTED CIRCLE FEATURE.

semicircle).

$$(x-a)^2 + (y-b)^2 = r^2$$

For each black pixel at (x, y), there is a set of circles of radius r that contain (x, y). This set of circles is represented by a curve (in this case another circle) in the (a, b) parameter space (see Figure D.4(a)). When the image contains several black pixels, each pixel has a set of circles that contain it, and may be a member of several such sets of circles, in which case the curves in the (a, b) space intersect. A circle feature is detected by creating a suitably discretised *accumulator* representation of the (a, b) space, where all cells are initialised to zero. For each black pixel in the image, cells intersecting the curve in the (a, b) space are incremented, and the maximum over the entire accumulator is taken to be the centre of the detected circle feature. In Figure D.4(b), the four black image pixels are all members of the detected circle feature indicated by the intersection of four curves in the accumulator. Figure D.3(d) shows an example of the Hough circle transform of an actual image, where the centre of the semi-circle is the darkest point in the image (white→black = empty→full).

The orientation θ of the line feature is detected using the standard parameterisation

$$x\cos\theta + y\sin\theta = \rho$$

where the meanings of ρ and θ are illustrated in Figure D.5(a). As in the case of the Hough circle transform, we use an accumulator to tally up the number of curves that intersect (x, y), but in the case of the Hough line transform, the optimal dimensions of the accumulator cells are not intuitively obvious. If the cells are too small, the transform is inefficient, but if they are too large, the transform is inaccurate. We used the techniques in [27] to select an appropriate discretisation of the (ρ, θ) space.

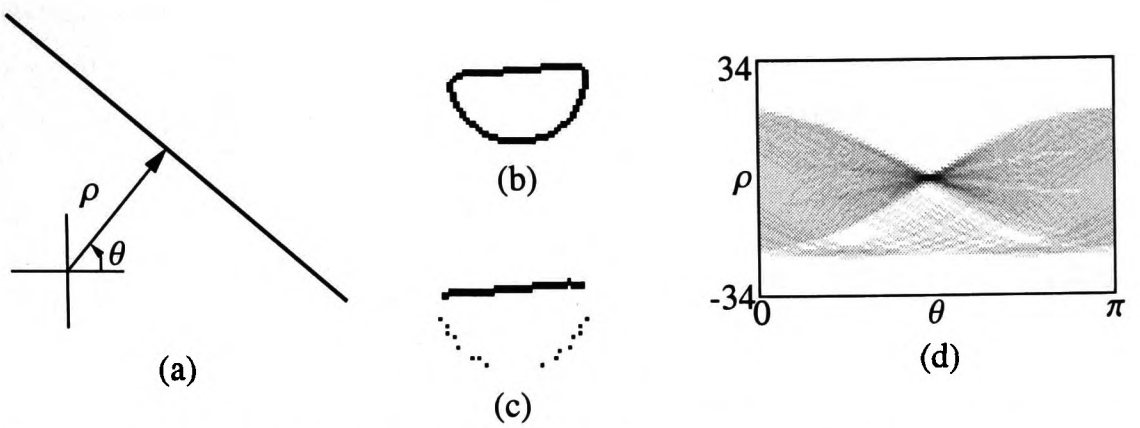


FIGURE D.5: FINDING THE PPRK ORIENTATION VIA THE HOUGH LINE TRANSFORM. (A) THE PARAMETERISATION USED IN THE HOUGH LINE TRANSFORM. (B) BOUNDED PORTION OF IMAGE FIGURE D.3(C) CENTRED AT THE MAXIMUM OF THE HOUGH CIRCLE TRANSFORM. (C) SAME BOUNDED PORTION, BUT WITH THE ARC FEATURE REMOVED. (D) HOUGH LINE TRANSFORM OF THE BOUNDED IMAGE WITH NO ARC FEATURE. THE MAXIMUM OF THE HOUGH LINE TRANSFORM OCCURS AT $(\rho, \theta) = (0, 1.5)$

Assuming we have already detected the circle feature, we first crop the portion of the edge-detected image that bounds the PPRK to reduce the amount of image that must be analysed (see Figure D.5(b)). We then remove the arc feature by 'painting' a white circle of radius r on this cropped image (see Figure D.5(c)) and determine the Hough line transformation of the cropped image. Figure D.5(d) shows the accumulator of the Hough line transform of the cropped image, with a maximum at $(\rho, \theta) = (0, 1.5)$.

D.2.1.2 (ii) – Determining the Distance $d_i(\mathbf{q})$ to Each Obstacle and the Generalised Gradient $\partial d_i(\mathbf{q})$

Since our algorithm depends on both the PPRK and obstacles having convex polygonal boundaries, we need to determine the location of the vertices of the convex PPRK and obstacles from the grabbed image. Now, it is possible to determine this information in an automated manner, however, since this is not a thesis on image analysis, we opted to simply input the location of the PPRK and obstacle vertices manually. This is accommodated through the user interface we developed for experimental trials, which is shown in Figure D.6. The interface allows the user to input the approximate location of the vertices using the mouse.

Given the location of the vertices entered by the user and the detected configuration of the PPRK, we can calculate the distance $d_i(\mathbf{q})$ between the PPRK and the obstacles $\mathcal{B}_i$ using a software collision detection. We opted to used Cameron's Enhanced-GJK [5] algorithm and software to accomplish this, since it also provides the location of the witness points (closest points) $\mathbf{a}_i \in \mathcal{A}$ and $\mathbf{b}_i \in \mathcal{B}_i$ on the PPRK and obstacle being tested. It is worth noting that fish-eye distortion causes the true distance to differ somewhat from the software detected distance.

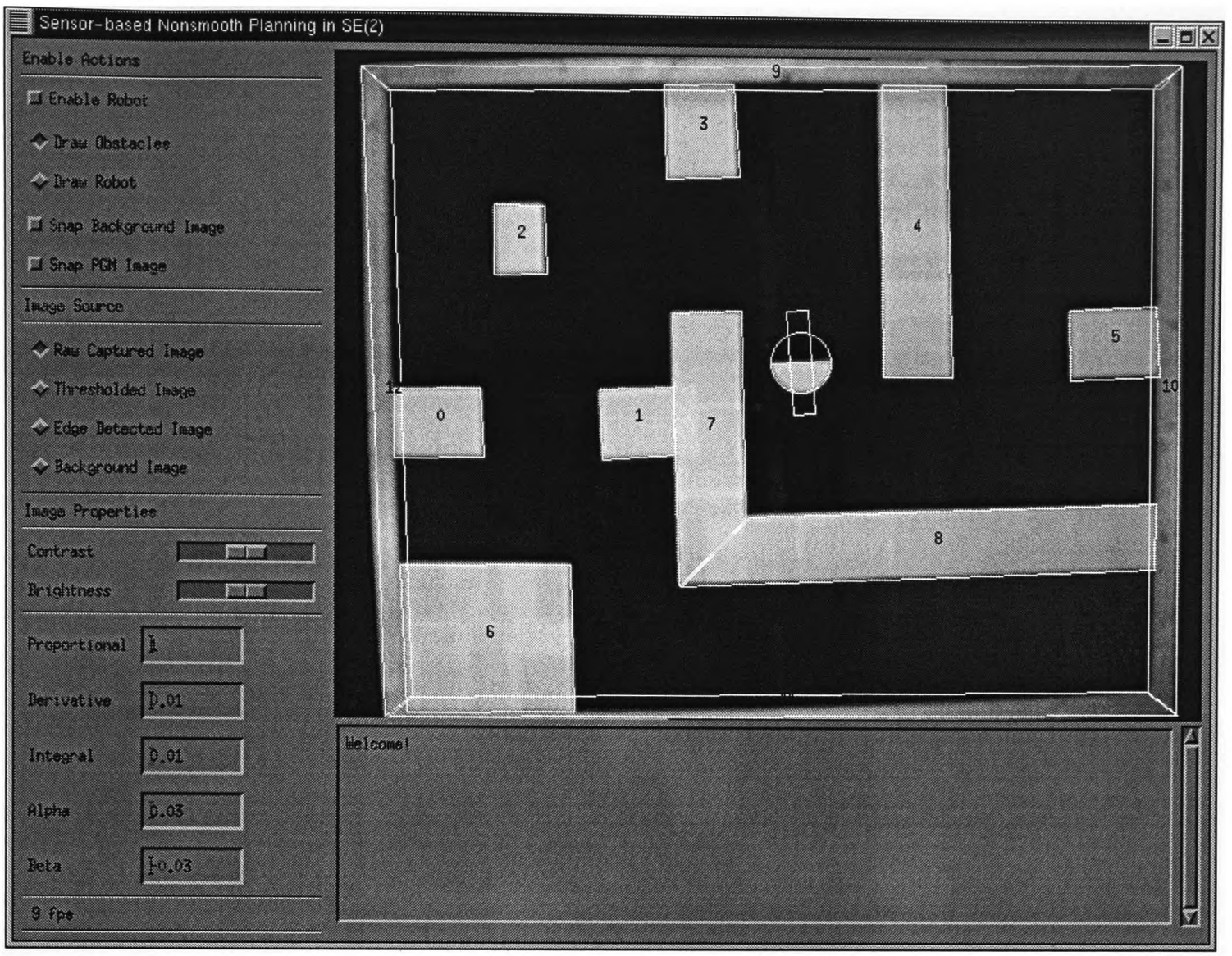


FIGURE D.6: SNAPSHOT OF THE USER INTERFACE USED IN EXPERIMENTAL TRIALS.

We determine the generalised gradient $\partial d_i(\mathbf{q})$ by determining if any edges of the PPRK and $\mathcal{B}_i$ are parallel, and if they are, taking the convex hull of the gradients of the points $\mathbf{a}_t(\mathbf{q})$ in the set $\tilde{P} \subseteq P$

$$\partial d_i(\mathbf{q}) = \text{co}\{\nabla \text{dist}(\mathbf{a}_t(\mathbf{q}), \mathcal{B}_i) : \mathbf{a}_t(\mathbf{q}) \in \tilde{P}, \text{dist}(\mathbf{a}_t(\mathbf{q}), \mathcal{B}_i) = d_i(\mathbf{q})\} \tag{D.4}$$

as per Lemma 3.35. We use the software packages qhull [2] to calculate the convex hull in Equation D.4. To deal with non-simplex sets (convex hulls with no volume) we add a few extra points a small ϵ distance from the points in $\tilde{P}$.

D.2.1.3 (ii) – Controlling the PPRK

The velocity of the PPRK was measured by observing the relative change in position during subsequent frames of captured video. A PID controller was used to control the velocity, and the proportional, derivative

and integral gains were tuned by trial and error until satisfactory curve following was achieved.

D.2.2 (III) – Serial Connection with PPRK

Communication with the PPRK is effected via a 9600 baud serial link with the control computer. Since the PPRK cannot reasonably drag a serial cable along as it explores its environment, we opted to use a wireless serial link of some description. By far the least expensive of the available options is to use infrared (IR) diodes, like those used in a remote control. Several kits exist for implementing an IR serial link, and each has little practical difference, barring baud rate, from the others. Since the PPRK is only capable of communicating at relatively low rates ($\leq$ 19,200) baud, we chose the HSDL-8000 IR evaluation kit, which consists of a pair a IR transceiver modules. Since a single transmitter is unable to cover the entire workspace, an array of IR diodes was attached to the ceiling and driven by the amplified output of the transmitter module. This array is able to cover the entire workspace and allow the receiver-equipped PPRK to move about over the entire field of view of the camera (the effective workspace of the PPRK).

Glossary

B

Banach space A normed linear space which is complete in the norm-determined metric.

C

convex function A function $f(x)$ is convex on an interval $[a, b]$ if for any two points $x_1, x_2 \in [a, b]$, $f(\frac{1}{2}(x_1 + x_2)) \leq \frac{1}{2}(f(x_1) + f(x_2))$.

convex hull The convex hull of a set S of points is the intersection of all convex sets containing S.

convex set A set S in $\mathbb{R}^n$ is convex if it contains all the line segments connecting any pair of its points.

D

diffeomorphism A differentiable (smooth) homeomorphism.

dual space If X is a normed linear space, then the dual space X^* is the space of all bounded linear functionals that can operate on elements of X.

H

holonomic motion A robot capable of holonomic motion if its instantaneous velocity has no non-holonomic constraints. A non-holonomic constraint is such that instantaneous velocities are constrained to a subset of velocity space. i.e. a car can move backwards and forwards, but not sideways.

homeomorphism Continuous, one-to-one, onto, and having a continuous inverse.

I

infimum The infimum of a set S is the greatest lower bound of S.

L

Lebesque measure Very imprecisely, a mathematical analogue of volume.

lim inf $\liminf f(\mathbf{x}) = \sup_{\delta>0} \inf_{0<|\mathbf{x}-\mathbf{y}||<\delta} f(\mathbf{x})$.

lim sup $\limsup f(\mathbf{x}) = \inf_{\delta>0} \sup_{0<|\mathbf{x}-\mathbf{y}||<\delta} f(\mathbf{x})$.

Lipschitz function A function $f : X \to Y$ is Lipschitz (of order k) if $|f(x_1) - f(x_2)| \leq k|x_1 - x_2|$ for all $x_1, x_2 \in X$.

P

PID controller A PID controller is a controller with **P**roportinal, **I**ntegral and **D**erivative terms. If **e** is the error in the system, the action of the controller is $K_P e + K_I \int e dt + K_D \frac{de}{dt}$.

positive definite A function $f(\mathbf{x})$ is positive definite at the origin if $f(\mathbf{x}) > 0$ for all $\mathbf{x} \neq \mathbf{0}$ and $f(\mathbf{0}) = 0$.

R

Rademacher's theorem If a function $f : \mathbb{R}^n \to \mathbb{R}$ is Lipschitz on an open subset U, then it is differentiable almost everywhere (in the sense of Lebesque measure) on U.

regular function A function $f(\mathbf{x})$ is said to be regular if the generalised directional derivative $f^\circ(\mathbf{x}; \mathbf{e})$ and the directional derivative $f'(\mathbf{x}; \mathbf{e})$ are equivalent.

S

separation theorem If A and B are disjoint, compact convex subsets of $\mathbb{R}^n$, then there exists an $\mathbf{x} \in \mathbb{R}^n$ such that $\max\{\langle \mathbf{a}, \mathbf{x} \rangle : \mathbf{a} \in \mathbb{R}^n\} < \min\{\langle \mathbf{b}, \mathbf{x} \rangle : \mathbb{R}^n\}$.

support points A support point $\mathbf{a} \in A$ of a convex, compact set $A \subseteq \mathbb{R}^n$ satifies $\langle \mathbf{a}, \mathbf{e} \rangle \geq \langle \mathbf{b}, \mathbf{e} \rangle$ for all $\mathbf{b} \in A$ for some $\mathbf{e} \in \mathbb{R}^n$.

supremum The supremum of a set S is the least upper bound of S.

T

triangle inequality If $x_1, x_2 \in X$ be vectors, then the triangle inequality states that $\| x_1 \| - \| x_2 \| \leq \| x_1 + x_2 \| \leq \| x_1 \| + \| x_2 \|$.

U

upper semicontinuous A function $f(x)$ is upper-semicontinuous if and only if $f(x) \geq \limsup_{n \to \infty} f(x_n)$.

Bibliography

[1] L. Asimow and A. J. Ellis. *Convexity Theorey and its Applications in Functional Analysis*. Number 16 in London Mathematical Society monographs. New series. Academic Press, London, 1980.

[2] C. B. Barber, D. P. Dobkin, and H. Huhdanpaa. The quickhull algorithm for convex hulls. *ACM Transactions on Mathematical Software*, 22(4):469–483, December 1996.

[3] P. Bessiereand, J. M. Ahuactzinand, E. G. Talbi, and E. Mazer. The Ariadne's clew algorithm: Global planning with local methods. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, pages 1373–1380, Yokohama, Japan, July 1993.

[4] J. Borenstein and Y. Koren. Real-time obstacle avoidance for fast mobile robots in cluttered environments. In *Proceedings of 1990 IEEE International Conference on Robotics and Automation*, volume 1, pages 573–577, 1990.

[5] S. Cameron. Enhancing GJK: Computing minimum and penetration distances between convex polyhedra. In *Proceedings of 1997 IEEE International Conference on Robotics and Automation*, April 1997.

[6] A. Canino and M. Degiovanni. Nonsmooth critical point theoray and quasilinear elliptic equations. In A. Granas and M. Frignon, editors, *Topological Methods in Differential Equations and Inclusions*, pages 1–50. Kluwer Academic Publishers, 1995.

[7] J. F. Canny. *The complexity of robot motion planning*. The MIT Press, London, 1988.

[8] B. Chazelle. Convex partition of polyhedra: A lower bound and worst case optimal algorithm. *SIAM Journal on Computing*, 13(3):488–507, 1984.

[9] G. S. Chirikjian. *Theory and Application of Hyper-Redundant Robotic Manipulators*. PhD thesis, California Institute of Technology, Pasadena, CA, 1992.

[10] H. Choset. Nonsmooth analysis, convex analysis, and their applications to motion planning. *International Journal of Computational Geometry and Applications*, 9(4–5):447–69, August–October 1999.

[11] H. Choset and J. Burdick. Sensor based planning and nonsmooth analysis. In *Proceedings of 1994 IEEE International Conference on Robotics and Automation*, volume 4, pages 3034–3041, 1994.

[12] H. Choset and J. Burdick. Sensor based planning, part I: The generalized voronoi graph. In *Proceedings of 1995 IEEE International Conference on Robotics and Automation*, volume 2, pages 1649–1655, 1995.

[13] H. Choset and J. Burdick. Sensor based planning, part II: Incremental construction of the generalized voronoi graph. In *Proceedings of 1995 IEEE International Conference on Robotics and Automation*, volume 2, pages 1643–1648, 1995.

[14] H. Choset and J. Burdick. Sensor based planning for a planar rod robot. In *Proceedings of 1996 IEEE International Conference on Robotics and Automation*, volume 4, pages 3584–3591, 1996.

[15] H. Choset, B. Mirtich, and J. Burdick. Sensor based planning for a planar rod robot: Incremental construction of the planar rod - HGVG. In *Proceedings of 1997 IEEE International Conference on Robotics and Automation*, volume 4, pages 3427–3434, 1997.

[16] F. H. Clarke. *Optimization and Nonsmooth Analysis*. Society of Industrial and Applied Mathematics, Philadelphia, PA, 1990.

[17] F. H. Clarke, Yu. S. Ledyaev, R. J. Stern, and P. R. Wolenski. *Nonsmooth Optimization and Control Theory*. Springer-Verlag, New York, 1998.

[18] J. Cohen, M. Lin, D. Manocha, and K. Ponamgi. I-collide: An interactive and exact collision detection system for large-scaled environments. In *Proceedings of ACM International 3D Graphics Conference*, pages 189–196, 1995.

[19] C. I. Connolly, J. B. Burns, and R. Weiss. Path planning using Laplace's equation. In *Proceedings of 1990 IEEE International Conference on Robotics and Automation*, volume 3, pages 2102–2106, 1990.

[20] C. I. Connolly and R. A. Grupen. The applications of harmonic functions to robotics. *Journal of Robotic Systems*, 10(7):931–46, 1993.

[21] M. Degiovanni and M. Marzocchi. A critical point theory for nonsmooth functionals. *Annali di Matematica pura ed applicata (IV)*, 167:73–100, 1994.

[22] P. E. Dupont and S. Derby. An algorithm for CAD-based generation of collision-free robot paths. In B. Ravani, editor, *CAD Based Programming for Sensory Robots*, volume 50 of *F*. Springer-Verlag, 1988.

[23] B. Faverjon and P. Tournassoud. A practical approach to motion planning for manipulators with many degrees of freedom. In *Preprints of the Fifth International Symposium on Robotics Research*, pages 65–73, Tokyo, 1989.

[24] V. Guilleman and A. Pollack. *Differential Topology*. Prentice Hall Inc., Englewood Cliffs, New Jersy, 1974.

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

[26] W. Henning, F. Hickman, and H. Choset. Motion planning for serpentine robots. In *ACSE Space and Robotics*, Albuquerque, New Mexico, 1998.

[27] J. Immerkaer. Some remarks on the straight line hough transform. *Pattern Recognition Letters*, 19:1133–1135, 1998.

[28] C. L. Jackins and S. L. Tanimoto. Octtrees and their use in representing three-dimensional objects. *Computer Graphics and Information Processing*, 14(3):249–270, 1980.

[29] D. Jung and K. Gupta. Octree-based hierarchical distance maps for collision detection. In *Proceedings of 1996 IEEE International Conference on Robotics and Automation*, volume 1, pages 454–459, 1996.

[30] I. Kamon, E. Rimon, and E. Rivlin. A new range sensor based globally convergent navigation algorithm for mobile robots. In *Proceedings of 1996 IEEE International Conference on Robotics and Automation*, volume 1, pages 429–435, 1996.

[31] I. Kamon and E. Rivlin. Sensory based motion planning with global proofs. In *Proceedings of 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 2, pages 435–440, 1995.

[32] I. Kamon, E. Rivlin, and E. Rimon. 3dbug: A three-dimensional range sensor based globally convergent navigation algorithm. Technical Report CIS9618, Technion, Isreali Institute of Technology, 1996.

[33] L. Kavraki and J. C. Latombe. Randomized preprocessing of configuration space for fast path planning. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 3, pages 2138–2145, San Diego, CA, USA, May 1994.

[34] S. S. Keerthi, C. J. Ong, E. Huang, and E. G. Gilbert. EquiDistance Diagram: A new roadmap for path planning. In *Proceedings of the 1999 International Conference on Robotics and Automation*, volume 1, pages 682–687, Piscataway, NJ, USA, 1999. IEEE.

[35] O. Khatib. Real-time obstacle avoidance for manipulators and mobile robots. *International Journal of Robotics Research*, 5:90–98, 1986.

[36] D. Kortenkamp and T. Weymouth. Topological mapping for mobile robots using a combination of sonar and vision sensing. In *Proceedings of the Twelfth National Conference on Artificial Intelligence*, 1994.

[37] J. C. Latombe. *Randomized Planning*, chapter 7, pages 340–350. In [38], 1991.

[38] J. C. Latombe. *Robot motion planning*. Kluwer Academic publishers, London, 1991.

[39] S. L. Laubach, J. Burdick, and L. Matthies. An autonomous path planner implemented on the Rocky7 prototype microrover. In *Proceedings of 1998 IEEE International Conference on Robotics and Automation*, volume 1, pages 292–297, 1998.

[40] J. Y. Lee, H. Choset, and A. A. Rizzi. Sensor-based planning for rod shaped robots in three dimensions: Piecewise retracts of $\mathbb{R}^3 \times s^2$. In *Proceedings of 2001 IEEE International Conference on Robotics and Automation*, pages 991–999, May 2001.

[41] W. Li, C. Ma, and S. K. Tso. Sensor-based obstacle modeling in configuration space for manipulator motion planning. In *Proceedings of the 1996 IEEE/SICE/RSJ International Conference on Multisensor Fusion and Integration for Intelligent Systems*, pages 265–272, 1996.

[42] M. C. Lin and J. F. Canny. An opportunistic global path planner. In *Proceedings of 1990 IEEE International Conference on Robotics and Automation*, volume 3, pages 1554–1559, 1990.

[43] M. C. Lin and J. F. Canny. An opportunistic global path planner. *Algorithmica*, 10:102–120, 1993.

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

[45] T. Lozano-Pérez and M. A. Wesley. An algorithm for planning collision free paths among polyhedral obstacles. *Communications of the ACM*, 22(10):21–29, October 1979.

[46] V. Lumelsky and E. Cheung. Real-time collision avoidance in teleoperated whole-sensitive robot arm manipulators. *IEEE Transactions on Systems, Man and Cybernetics*, 23(1):194–203, 1993.

[47] V. J. Lumelsky. Dynamic path planning for a planar articulated robot arm moving amonst unknown obstacles. *Automatica*, 23(5):551–570, 1987.

[48] V. J. Lumelsky and T. Skewis. Incorporating range sensing in the robot navigation function. *IEEE Transactions on Systems, Man and Cybernetics*, 20(5):1058–1068, 1990.

[49] V. J. Lumelsky and A. A. Stepanov. Path planning strategies for a point mobile automaton moving amidst obstacles of arbitrary shape. *Algorithmica*, 1(2):403–430, 1987.

[50] V. J. Lumelsky and K. Sun. A unified methodology for motion planning with uncertainty for 2D and 3D two-link robot arm manipulators. *International Journal of Robotics Research*, 9(5):89–104, 1990.

[51] B. Mirtich. V-clip: Fast and robust polyhedral collision detection. Technical Report TR-97-05, Mitsubishi Electric Research Laboratory, July 1997.

[52] G. Reshko M. T. Mason I. Nourbakhsh. Really mobile computing: Build your own palm powered robot. In *Palm Power Magazine*. 2000.

[53] C. Ó'Dúnlaing, M. Sharir, and C. K. Yap. *Retraction: a new approach to motion planning*, chapter 7. In Schwartz et al. [62], 1987.

[54] C. Ó'Dúnlaing and C. K. Yap. *A retraction method for planning the motion of a disk*, chapter 6. In Schwartz et al. [62], 1987.

[55] R. S. Palais and C. L. Terng. *Critical Point Thery and Submanifold Geometry*. Springer Verlag, 1988.

[56] B. Pourciau. Global invertibility of nonsmooth mappings. *Journal of Mathematical Analysis and Applications*, 131(1):170–179, 1988.

[57] N. S. V. Rao, S. Kareti, W. Shi, and S. S. Iyengar. Robot navigation in unknown terrains: Introductory survey of non-heuristic algorithms. Technical Report ORNL/TM-12410, Oak Ridge National Laboratory, 1993.

[58] D. Reznik and V. Lumelsky. Sensor based motion planning for highly redundant kinematic stuctures: II the case of a snake robot manipulator. In *Proceedings of 1993 IEEE International Conference on Robotics and Automation*, volume 3, pages 889–94, 1993.

[59] E. Rimon. Construction of C-space roadmaps from local sensory data - what should sensors look for. *Algorithmica*, 17:357–379, 1997.

[60] E. Rimon and J. F. Canny. Construction of C-space roadmaps from local sensory data - what should sensors look for. In *Proceedings of 1994 IEEE International Conference on Robotics and Automation*, volume 1, pages 117–23, 1994.

[61] L. G. Roberts. Machine perception of three-dimensional solids. In J. Tippett et al., editor, *Optical and Electro-Optical Information Processing*, pages 159–197. MIT Press, 1965.

[62] J. T. Schwartz, M. Sharir, and J. E. Hopcroft, editors. *Planning, geometry and complexity*. Ablex series in artificial intelligence. Ablex Publishing Corporation, New Jersey, 1987.

[63] A. Stentz. Optimal and efficient path planning for partially known environments. In *Proceedings of 1994 IEEE International Conference on Robotics and Automation*, volume 1, pages 3310–3317, 1994.

[64] K. Sun and V. J. Lumelsky. Path planning among unknown obstacles: The case of a three-dimensional cartesian arm. *IEEE Transactions on Robotics and Automation*, 8(6):776–786, 1992.

[65] C. J. Taylor and D. J. Kriegman. Vision-based motion planning and exploration algorithms for mobile robots. In *Proceedings of Workshop on the Algorithmic Foundations of Robotics*, February 1994.

[66] Y. Yu and K. Gupta. A practical approach to sensor based motion planning for manipulator arms. obtained by personal correspondence.

[67] M. Zeller, K. Schulten, and R. Sharma. Motion planning of a pneumatic robot using a neural network. *IEEE Control Systems Magazine*, 127(3):89–98, June 1997.