

Motion Planning for Maintaining Landmarks Visibility with a Differential Drive Robot

Jean-Bernard Hayet^a, Hugo Carlos^a, Claudia Esteves^b, Rafael
Murrieta-Cid^a

^a*Centro de Investigación en Matemáticas (CIMAT), Guanajuato, México*

^b*Departamento de Matemáticas, Universidad de Guanajuato, Guanajuato, México*

Abstract

This work studies the interaction of non-holonomic and visibility constraints using a Differential Drive Robot (DDR) that has to keep static landmarks in sight in an environment with obstacles. The robot has a limited sensor, namely, it has a restricted field of view and bounded sensing range (e.g. a video camera). Here, we mean by *visibility* that a clear line of sight can be thrown between the landmark and the sensor mounted on the DDR. We first determine the necessary and sufficient conditions for the existence of a path such that our system is able to maintain *one* given landmark visibility in the presence of obstacles. This is done through a recursive, complete algorithm that uses motion primitives exhibiting local optimality, as they are locally shortest-lengths paths. Then, we extend this result to the problem of planning paths guaranteeing visibility among a set of landmarks, e.g. to observe a given sequence of landmarks or to observe at each point of the path *at least one* element of the landmarks set. We also provide a procedure that computes the robot controls yielding such a path¹.

Keywords: motion planning, landmark-based navigation

1. Introduction

In this work, we consider a differential drive robot (nonholonomic system) equipped with a sensor limited in field of view (e.g. a video camera).

¹A preliminary version of this work has been presented in the 2008 International Workshop on the Algorithmic Foundations of Robotics (WAFR'08) and published in Hayet et al. [2010].

The robot moves in an environment with obstacles that produce motion and visibility obstructions. The task given to the robot is to compute a collision free path between two configurations, while maintaining visibility of one or several landmark(s) during the execution of the path. More precisely, we present a *complete* planning algorithm², that given a pair of initial and final configurations and a set of N landmarks can determine whether or not there exists a path that maintains visibility of (1) one specific landmark among the N or (2) at least one of the N landmarks, during the execution of the robot path. If such a path exists, our planner provides a constructive procedure to execute it. This procedure computes the robot controls yielding such a path. Note that, in general, during the execution of the resulting path, the robot may switch from maintaining visibility of one landmark i to another landmark j and then to another landmark k , and so on. The ability of our planner to determine whether or not it is possible to maintain visibility of any pair of landmarks, one after the other, can be used to generate a robot path that maintains visibility of a given *sequence* of landmarks. Thus, our planner can be used to accomplish navigation/surveillance tasks such as: Move the robot from the red ball to the blue cube and then to the green pyramid, while not losing sight of these objects, one after the other. Also, the ability of our planner for generating robot paths that maintain landmarks visibility can be useful for localization purposes. In this work, we also present applications to robot localization and compare the evolution of the robot uncertainty when the robot moves following paths generated with our planner, to the uncertainty when moving in other paths planned without considering landmarks visibility.

1.1. Related work

Our work is related to motion planning with nonholonomic constraints Laumond et al. [1994]; Bicchi et al. [1996]; Souères and Laumond [1996]; Laumond [1998]; Balkcom and Mason [2002], motion planning with visibility constraints Isler et al. [2005]; Michel et al. [2007], and landmark based robot navigation Lazanas and Latombe [1995]; Roy and Thrun [1999]; Murrieta-Cid et al. [2002]; Ranganathan et al. [2002]; Madhavan and Durrant-Whyte [2004]; Hayet et al. [2007] and localization Sim and Dudek [1998]; Murrieta-Cid et al. [2002]; Se et al. [2005];

²Such an algorithm is guaranteed to return a correct solution when one exists, or to report failure in finite time when a solution does not exist.

Ranganathan et al. [2002]; Hayet et al. [2007]. A landmark used in robotic navigation can be defined in many different manners: from single characteristic image points Se et al. [2005], appearance-based representations Sim and Dudek [1998, 2003], up to a 3D object associated with a semantic label and 3D position accuracy. . . Hayet et al. [2007]. In all cases, this definition involves at some degree properties of saliency and invariance to viewpoint changes.

Motion planning with non-holonomic constraints has been a very active research field (a nice overview is given in Laumond [1998]). The most important results in this field have been obtained by addressing the problem with tools from Differential Geometry and Control Theory. Laumond pioneered this research and produced the result that a free path for a holonomic robot moving among obstacles in a 2D workspace can always be transformed into a feasible path for a non-holonomic car-like robot by making car maneuvers Laumond et al. [1994]. The work in Bicchi et al. [1996] uses motion primitives to build a basic path diagram, similar to a visibility graph, but for non-holonomic vehicles.

In Murray and Sastry [1993], the authors investigate methods for steering systems with nonholonomic constraints between arbitrary configurations. The authors define a class of systems, which can be steered using sinusoids. However, as pointed out by the authors, the difficulty is to determine the size of the free space allowed around the steering solution. In Laumond et al. [1994], the authors show that if the robot path metric induces the same topology as the metric used to measure distance between the robot and the obstacles, then the existence of any $\epsilon > 0$ clearance between the robot and the obstacles guarantees a feasible path. The authors addressed the case of a car-like robot moving among obstacles, and provided a constructive procedure for finding a solution path between two robot configurations. In our work, we find equivalent results as in Laumond et al. [1994], but for a differential drive robot equipped with a sensor limited in field of view, which must maintain landmarks visibility in the presence of obstacles.

In Divelbiss and Wen [1997], the authors present an algorithm for finding a feasible path for a nonholonomic system in the presence of obstacles. First, the path-planning problem without obstacles is transformed into a nonlinear least squares problem in an augmented space. Obstacle avoidance is included as inequality constraints and simulations results for the case of a tractor-trailer are presented. However, in Divelbiss and Wen [1997] the authors did not show the convergence of the proposed algorithm, neither considered visibility constraints as in the work presented in this paper.

In Michalek and Kozłowski [2011], the problem of proposing feedback controllers for car-like robots is addressed. The effectiveness and limits of the method have been illustrated by simulations. However, in Michalek and Kozłowski [2011] the authors did not take into account either an environment with obstacles or visibility constraints.

In the study of optimal paths for non-holonomic systems, Reeds and Shepp determined the shortest paths for a car-like robot that can move forward and backward Reeds and Shepp [1990]. In Souères and Laumond [1996], a complete characterization of the shortest paths for a car-like robot is given. In Balkcom and Mason [2002], Balkcom and Mason determined the time-optimal trajectories for a DDR using Pontryagin’s Maximum Principle (PMP) and geometry. In Chitsaz et al. [2009], the PMP is used to obtain the extremal trajectories to minimize the amount of wheel rotation for a DDR.

One of the earliest works in landmark based robot navigation is the one of Lazanas et al. Lazanas and Latombe [1995]. Geometric visibility is not explicitly integrated; instead, each landmark defines a circular “safe zone” in which the robot is supposed to sense and move without uncertainty. The complete, polynomial algorithm uses back-chaining of omnidirectional back-projections to compute plans in the presence of uncertainty. In Briggs et al. [2004], the landmarks are seen as sub-goals to reach and the planning uses a probabilistic framework to compute expected shortest paths, in the landmarks graph. In Roy and Thrun [1999], a landmark-based navigation approach has been proposed, that generates “coastal navigation” paths. In a probabilistic robot localization task, an important advantage of a landmark-based approach is that if landmarks are detected, they stop the incremental growth of the robot position uncertainty Roy and Thrun [1999]; Sim and Dudek [1998]; Murrieta-Cid et al. [2002]; Hayet et al. [2007]; Ranganathan et al. [2002].

On the specific problem of maintaining landmarks visibility, similar problems as the one tackled in this paper have been addressed from quite different viewpoints. In Bhattacharya et al. [2007], it has been shown that the shortest distance paths, in the absence of obstacles, for a DDR under *angular and visibility constraints only*, are composed of three motion primitives: straight-line segments, rotations in place without translation and logarithmic spirals. In that same work, a characterization of the shortest paths for the system based on a partition of the plane into disjoint regions was also provided. This synthesis attempted to obtain the globally optimal paths *in the absence of obstacles*. Later in Salaris et al. [2008], it has been shown that the synthesis

presented in Bhattacharya et al. [2007] was incomplete and in Salaris et al. [2010], the complete partition of the plane and the corresponding globally optimal paths in the absence of obstacles that connect any two configurations of $SE(2)$, while keeping one particular landmark in sight, has been proposed.

In visual servo-control, the problem of moving while keeping an object in-sight has been usually dealt within a stack-of-tasks formalism Mansard and Chaumette [2007]. High priority tasks may include collision avoidance and visibility maintenance of a certain visual feature. An interesting related work is López-Nicolás et al. [2010] that also uses a visual servoing strategy (based on homographies) to control a DDR to follow some of the optimal paths proposed by Bhattacharya et al. [2007] and commented above.

More recently, the (soft) visibility constraints have been integrated quite naturally in a sampling-based scheme, by filtering out sampled configurations that do not give minimal satisfaction w.r.t. the landmark visibility Michel et al. [2007]. However, the use of a sample-based strategy, a realistic alternative for complex articulated systems, is done at the cost of losing deterministic completeness.

As shown in the aforementioned works, the use of landmarks for robot navigation and localization is widespread in robotics Lazanas and Latombe [1995]; Roy and Thrun [1999]; Murrieta-Cid et al. [2002]; Se et al. [2005]; Michel et al. [2007]; Hayet et al. [2007]; Bhattacharya et al. [2007]; López-Nicolás et al. [2010]. In any of these contexts (navigation, localization...), the first basic requirement to use landmarks is to be able to perceive them, especially during the execution of robot motion, in spite of the sensors field of view limitations. It is to this end that our current research efforts are focused on: to integrate traditional kinematic constraints and obstacles altogether with visibility constraints in the formulation of the motion planning problem.

2. Contributions and paper organization

In this work, we propose an approach that incorporates to the planning the additional constraint that the robot has to be able to observe *at least one landmark at each point of its path*. Obstacles in the environment generate both, motion and visibility obstructions. An example of an output of this algorithm is depicted in Figure 1. This Figure shows that the computed plan (1) leads from a starting point to an end point through the execution of several motion primitives that, as we will see, exhibit local optimality properties and (2) is formed exclusively by robot configurations from which

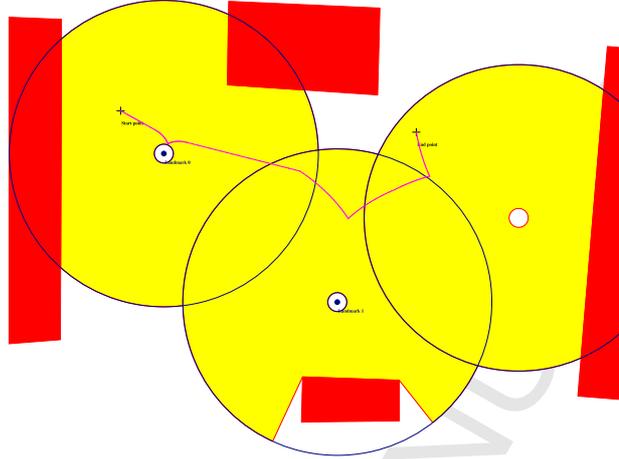


Figure 1: An example of input/output of our algorithm. Obstacles in the 2D environment are depicted as polygons, and the visibility areas of each landmark are shaded. The black dots correspond to the landmarks that populate the environment, and the two circles centered on them are the minimal and maximal range of the sensor. The planner outputs paths connecting a start point to an end point.

at least one of the three landmarks are visible with a camera-like sensor (i.e. with angular limits and distance limits, the latter ones being represented by the light gray discs). This is to our knowledge *the first complete algorithm* for determining whether or not a path for a holonomic robot in the presence of obstacles that has to maintain visibility of landmarks with a limited sensor can be transformed into a feasible path for a DDR. We believe that our research is very pertinent given that a lot of mobile robots are DDRs equipped with limited field of view sensors (e.g., lasers or cameras) and to the wide range of applications such as those mentioned in above. For the sake of clarity, we will explain first the approach for a DDR that has to maintain one single landmark visible (Section 6); then, we extend it to the case of a DDR that has to maintain at least one landmark (among N) visible (Section 8). Obviously, an algorithm capable of solving the second problem will also be capable of solving the first.

As it was mentioned in the section of previous work, motion planning with nonholonomic constraints as well as motion planning under visibility

constraints have been two very active topics in the robotics community. But, to our knowledge, the works most closely related to our approach are Laumond et al. [1994]; Bhattacharya et al. [2007]; Salaris et al. [2010]. In Laumond et al. [1994], a complete algorithm for moving a car-like robot in an environment with obstacles from an initial to a final configuration has been presented. One of the main results presented in Laumond et al. [1994] states that a free path for a holonomic robot moving among obstacles in a 2D workspace can always be transformed into a feasible path for a car-like robot by making car maneuvers Laumond et al. [1994]. In this work, we follow a methodology similar to the one proposed in Laumond et al. [1994]. However, there are two important differences between our work and Laumond et al. [1994]: 1) Our system is a differential drive robot and not a car-like system; 2) We consider the sensing constraint of maintaining visibility of landmarks. Visibility restrictions have not been considered in Laumond et al. [1994]. In Bhattacharya et al. [2007], the problem of planning optimal Euclidean distance paths for a differential-drive robot with limited sensing, that must maintain visibility of a single landmark as it navigates in an environment without obstacles, has been formulated. In Bhattacharya et al. [2007], the three locally optimal motion primitives for the problem have been found. In Salaris et al. [2010], the complete partition of the plane and the corresponding globally optimal paths for the problem stated in Bhattacharya et al. [2007] have been determined. Although we use results from Bhattacharya et al. [2007]; Salaris et al. [2010], there are new significant contributions in this work compared with the results obtained in Bhattacharya et al. [2007] and Salaris et al. [2010]: 1) We consider an environment populated with obstacles; 2) We extend the work for maintaining visibility of a sequence of landmarks in contrast to a single one. It is also important to notice that while in Salaris et al. [2010] (for an environment *without* obstacles and a single landmark) the resulting paths are globally *optimal*, in this work, in general, the resulting paths are only *feasible*.

A preliminary version of portions of this work appeared in Hayet et al. [2010]. The main distinguishing features of our current work compared with our previous research in Hayet et al. [2010] are: 1) We extend our work for maintaining visibility of a sequence of landmarks in contrast to a single one. 2) We provide a complexity characterization of our algorithm. 3) We apply our approach to probabilistic robot localization, and we compare the robot localization uncertainty while the robot moves following paths generated by our planner with other paths that do not necessarily sense landmarks during

their execution. 4) We implement our approach for landmark-based navigation on a real robot.

The remainder of this paper is organized as follows: In Section 3, we present in detail our problem formulation and the notations that we will use throughout the paper. In Section 4, we provide the controls for the execution of the locally optimal motion primitives. In Section 5, we present a brief summary of the complete synthesis (presented in Hayet and Murrieta-Cid [2009]; Salaris et al. [2010]), yielding the globally optimal paths for maintaining visibility of a single landmark, in an environment without obstacles. This synthesis is used, as one of the components, in the generation of a path in the presence of obstacles. In Section 6, we present a complete motion planner for a *holonomic* disc, which generates collision-free paths while maintaining a single landmark visibility. The path generated by this holonomic planner is used to obtain connectivity in the configuration space. In Section 7, we propose a motion planner that integrates the non-holonomic constraints executing robot maneuvers, which correspond to the execution of the locally optimal motion primitives. In Section 7.3, we study the properties of completeness and complexity of this planner. Section 8 extends the planner to obtain paths maintaining visibility of at least one of the N landmarks. Section 9 presents experiments done in order to prove the effectiveness and pertinence of our algorithm. Finally in Section 10, we present the conclusion and future work.

3. Problem settings and approach overview

The DDR is described in Figure 2. It is controlled through the angular velocities of its wheels w_l and w_r . We make the usual assignment of a $x'y'$ frame attached to the robot. The origin is at the midpoint between the two wheels, y' -axis parallel to the axle, and the x' -axis pointing forward. The angle θ is formed by the world x -axis and the robot x' -axis. The robot can move forward and backward. The heading is the direction in which the robot moves, so it is either zero (forward move) or π (backward move) with respect to the robot x -axis. The position of the robot w.r.t some global frame \mathcal{R} will be defined either in Cartesian coordinates (x, y) or in polar coordinates (r, α) : $r = \sqrt{x^2 + y^2}$, $\alpha = \arctan \frac{y}{x}$ ³. Figure 2 sums up these conventions.

³It will always be assumed in the remainder that the polar angle is computed in such a way that no orientations are lost, i.e. as $\arctan 2(y, x)$

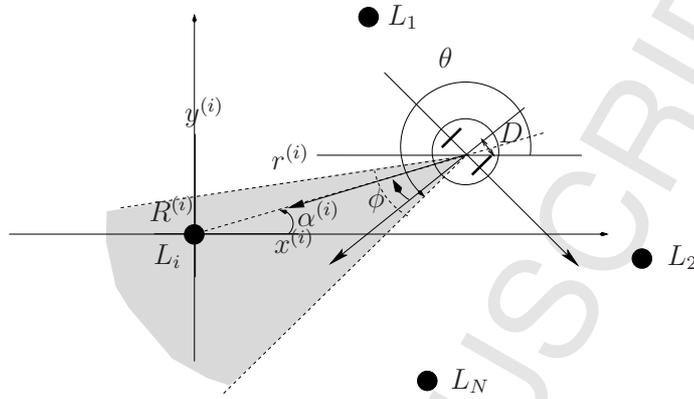


Figure 2: DDR with sensor restrictions in angle (i.e. on the sensor angle ϕ) and target visibility. The robot visibility region is the shaded region.

The robot is equipped with a pan-controllable sensor with limited field of view (e.g., a camera), that can move w.r.t. the robot basis (e.g., a pan-tilt-zoom camera). We assume that this sensor is placed on the robot so that the optical center always lies directly above the origin of the robot's local coordinate frame, i.e., the center of rotation of the sensor is the same as the one of the robot. Its pan angle ϕ is the angle from the robot x' -axis to its optical axis. The sensor is limited, both in angle and in range: $\phi \in [\phi_-, \phi_+]$.

We suppose that the workspace is populated with N landmarks \mathcal{L}_i located at positions $L_i = (x_i, y_i)$ relative to \mathcal{R} . Associated to each landmark \mathcal{L}_i is a local frame \mathcal{R}_i with axis parallel to the ones of \mathcal{R} and in which we can locate the robot by its cartesian coordinates $(x^{(i)}, y^{(i)})$ or polar coordinates $(r^{(i)}, \alpha^{(i)})$. The constraints we handle for this system are three-fold: kinematic constraints, landmark visibility constraints and sensor constraints.

Assume first that the robot moves in the absence of physical obstacles. We will remove this assumption later. Moreover, *let us cope with just one landmark*, say \mathcal{L}_1 . We will then extend the problem to observe at least one of the \mathcal{L}_i . Hence, we will use $\mathcal{R} = \mathcal{R}_1$ up to Section 8.

The *non-holonomy* constraint arises from the robot mechanism (i.e., the DDR instantaneous velocity is in the robot direction), and translates into

$$\dot{y} \cos \theta - \dot{x} \sin \theta = 0. \quad (1)$$

The *visibility* constraint results from constraining the landmark \mathcal{L}_1 to be

at the center of the robot gaze, which can be written (see Figure 2),

$$\theta = \alpha - \phi + (2k + 1)\pi, k \in \mathbb{Z}. \quad (2)$$

Sensor constraints can be defined classically, e.g. in terms of physical bounds on the sensor range,

$$\phi_- \leq \phi \leq \phi_+. \quad (3)$$

The values $\phi_- < 0$ and $\phi_+ > 0$ are the angle sensor limits. From now, we simplify the problem by setting $\phi_- = -\phi_+$ (i.e., symmetric sensor).

We also add constraints on the sensor range, e.g. the locations p from which \mathcal{L}_1 is visible are such that the Euclidean distance from (x_1, y_1) to p ,

$$d^{(1)} \leq r^{(1)} \leq D^{(1)}. \quad (4)$$

The parameters $d^{(i)}$ and $D^{(i)}$ are minimal and maximal ranges for the sensor with respect to \mathcal{L}_i . They may vary with the landmark size. Notice that these limitations on the sensor induce landmark-dependent virtual obstacles in the configuration space even without physical obstacles. Also, note that the two sensor constraints presented here will be treated differently: the angular constraints will be managed at the motion level, whereas the range constraints will be managed as “regular” (but virtual) obstacles.

Approach overview. As mentioned above, we follow an incremental methodology, similar to the one in Laumond et al. [1994] for a DDR equipped with a sensor with a limited field of view. First, we propose a complete motion planner for a *holonomic* disc, which generates collision-free paths while maintaining landmark visibility. Notice that if a path does not exist for a holonomic robot then a path neither exists for a nonholonomic system with the same shape and size. The work reported in Laumond et al. [1994] uses the complete planner proposed in Avnaim and Boissonat [1988], for a holonomic polygonal robot moving amid polygonal obstacles, to establish whether or not a path for a holonomic robot exists. In this paper, we propose a planner for a holonomic disc. On one hand, the assumption about the shape of the robot (i.e. a disc) facilitates the task, but on the other hand we also consider visibility constraints in our holonomic planner. Second, we use the shortest path synthesis Salaris et al. [2010] (partition of the plane)

for a DDR with visibility constraints to generate a candidate robot path; if the former is collision free, we use it to move the robot and take the optimal path in terms of the Euclidean distance for maintaining visibility of a single landmark. Third, if this path is in collision with an obstacle then it is recursively divided until a collision free path is obtained. Each element of the modified path corresponds to one locally optimal motion primitives from Bhattacharya et al. [2007]. Based on the equivalence between the metric induced by the shortest path of our system and the Euclidean metric used to compute the distance between the robots and the obstacles, we show that the recursive division does converge. Finally, we present an algorithm to maintain visibility of a sequence of landmarks, which keeps the completeness of the approach for a single landmark.

4. Controls for executing the locally optimal primitives

Our robot state transition equation has two controls, as, in this scheme, we suppose that the sensor is pointing to the landmark by adjusting its angle value according to equation 2. Hence, the sensor control is not independent in this motion analysis. The state transition equation is:

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix}, \quad (5)$$

which corresponds to the classical DRBalkcom and Mason [2002]; LaValle [2006], where $u_1 = w_r + w_l$ is the linear velocity and $u_2 = w_r - w_l$ the angular one.

As recalled hereafter, shortest lengths paths satisfying non-holonomic, visibility and sensor angle constraints can be locally only one of *three motion primitives*, straight lines, rotation in place and logarithmic spirals (see Section 5). The vector field associated to the straight line is $\vec{X}_1 = (\cos \theta, \sin \theta, 0)^T$, the one associated to the rotation in place is simply $\vec{X}_2 = (0, 0, 1)^T$.

Let us express the vector field associated to logarithmic spirals. In other words: What are the open-loop controls to follow the logarithmic, saturating sensor pan angle? The equations of these curves are, in polar coordinates:

$$r = r_0 e^{(\alpha_0 - \alpha) / \tan \phi}, \quad (6)$$

where (r_0, α_0) is one arbitrary point on the spiral and the camera pan angle ϕ remains constant along the spiral. When the robot moves drawing a sector of a logarithmic spiral, the camera pan angle is saturated Bhattacharya et al. [2007] and hence the landmark is in the limit of the sensor field of view. As a consequence, $\dot{\phi} = 0$. Now, by taking the derivative of Equation 2, we obtain a relation between u_1 and u_2 after some algebraic developments:

$$(-y \cos \theta + x \sin \theta)u_1 = (x^2 + y^2)u_2,$$

which can be re-written in its polar form (that does not depend on θ)

$$u_2 = \frac{1}{r}u_1 \sin(\phi). \quad (7)$$

In terms of the wheels controls, we deduce from Equation 7, for $r > 0$,

$$\begin{cases} w_r = 1 \\ w_l = \frac{r - \sin(\phi)}{r + \sin(\phi)}. \end{cases}$$

Again in terms of u_1 and u_2 , and by setting $u_1 = 1$,

$$\begin{cases} u_1 = 1 \\ u_2 = \frac{\sin(\phi)}{r}. \end{cases}$$

From the previous equation, we can derive the corresponding vector field,

$$\vec{X}_3 = \begin{pmatrix} \cos \theta \\ \sin \theta \\ \frac{\sin(\phi)}{r} \end{pmatrix}. \quad (8)$$

Thus, to follow the optimal motion primitives, it is sufficient to consider three cases of controls (u_1, u_2) that lead the robot to trace these primitives: Straight lines, rotation in place and logarithmic spirals. They are respectively

$$\begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix} \text{ and } \begin{pmatrix} 1 \\ \frac{\sin(\phi)}{r} \end{pmatrix}. \quad (9)$$

Note that the first two controls are constant and therefore bounded, and the third one is also bounded since we consider $r > d^{(1)} > 0$.

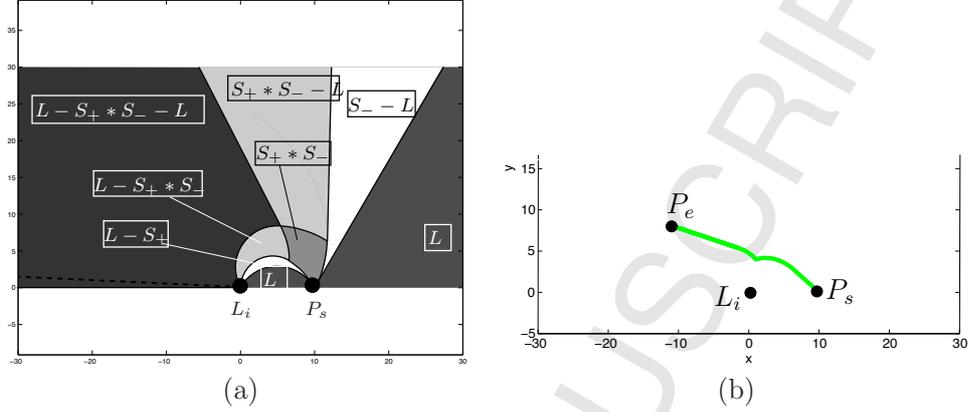


Figure 3: (a) Distribution of shortest paths, revised from Bhattacharya et al. [2007] into Hayet and Murrieta-Cid [2009]; Salaris et al. [2010]. The partition is relative to a given landmark \mathcal{L}_i (black dot) and a given starting point P_s (the other black dot), and is symmetric w.r.t. to the line $(\mathcal{L}_i P_s)$. (b) An example of optimal primitive $(L-S_+*S_-*L)$ for a pair (P_s, P_e) of starting and ending configurations.

5. Globally optimal paths in the free space

As stated above, the locally optimal motion primitives yielding a globally optimal path are either line segments, rotations in place or *logarithmic spirals*. Spirals are paths *saturating the sensor angle* ϕ at ϕ_- or ϕ_+ , i.e. satisfying in the polar coordinates of the local frame \mathcal{R}_i related to the landmark \mathcal{L}_i ,

$$r^{(i)}(t) = r_0^{(i)} e^{-\frac{\alpha_0^{(i)} - \alpha^{(i)}(t)}{\tan \phi_k}}, \text{ where } k = -, +,$$

and where $P_0 = (r_0^{(i)}, \alpha_0^{(i)})$ denotes any point of the spiral. In the remaining, logarithmic spirals trajectory parts will be referred to as a letter “ S_- ” or “ S_+ ” (according to whether the sensor is saturated at ϕ_- or ϕ_+), or simply “ S ”; line segments trajectory parts will be referred to as “ L ” and rotations in place trajectory parts as an “ $*$ ” between two other parts (non-differentiable point), or will be omitted if they appear at the beginning or at the end of the path. Similarly, the “ $-$ ” will refer to differentiable transitions between primitives. From this local characterization and by using geometric arguments Hayet and Murrieta-Cid [2009]; Salaris et al. [2010], it has been shown that the resulting optimal trajectories are to be found among the following types: L , $L-S_+$, S_+-L , S_+*S_- , $L-S_+*S_-$, $S_+*S_- - L$,

$L - S_+ * S_- - L$, $L * L$ and 6 other types obtained by swapping S_- and S_+ .

It might be debated whether the $L * L$ type should be considered, as it implies passing *through* the landmark. Moreover, paths going through the origin induce a local topology for shortest paths that *a priori* could lead our recursive planning algorithm to fail; however, we will see in Section 7.2 that it is not the case. In the same aforementioned article, a plane partition according to the relative position of initial and final points P_s and P_e is given. Figure 3(a) illustrates this partition, for a value of $\phi_+ = \frac{\pi}{3}$. This partition is centered in the landmark \mathcal{L}_i and is symmetric with respect to a landmark-starting position line (the horizontal axis, in this case). As an example, the dark region including the figure upper right corner is the one where $L - S_+ * S_- - L$ trajectories are the shortest, and an example of path following this primitive is shown in Figure 3(b). We will use this synthesis in the planning algorithms and the motion primitives from Bhattacharya et al. [2007] are used to replace parts of a holonomic path.

6. One-visible-landmark paths for a holonomic robot

6.1. Configuration space without obstacles

As mentioned above, our robot must maintain visibility of the landmark \mathcal{L}_1 . By visibility we mean that a clear line of sight, lying within the minimal and maximal bounds of the sensor rotation angle and range, can join the landmark and the sensor. The landmark \mathcal{L}_1 is static and, for the moment, will be supposed to be coincident with the origin O of the coordinate system (i.e. $\mathcal{R}_1 = \mathcal{R}$). For the sake of clarity, we will omit the superscripts “(1)” referring to the landmark. In this frame, the constraints can be re-written as

$$\theta = \alpha - \phi + (2k + 1)\pi, \quad k \in \mathbb{Z}, \quad (10)$$

$$\phi_- \leq \phi \leq \phi_+, \quad (11)$$

$$d^{(1)} \leq r \leq D^{(1)}. \quad (12)$$

We can now describe the robot admissible configuration space, referred to as \mathcal{C}_{adm} , a subspace of $SE(2)$, as from Equation 2, ϕ is not a degree of freedom. Moreover, Equation 2 adds a constraint that can be rewritten

$$\phi_- \leq -\theta + \arctan\left(\frac{y}{x}\right) + (2k + 1)\pi \leq \phi_+ \text{ for some } k \in \mathbb{Z}. \quad (13)$$

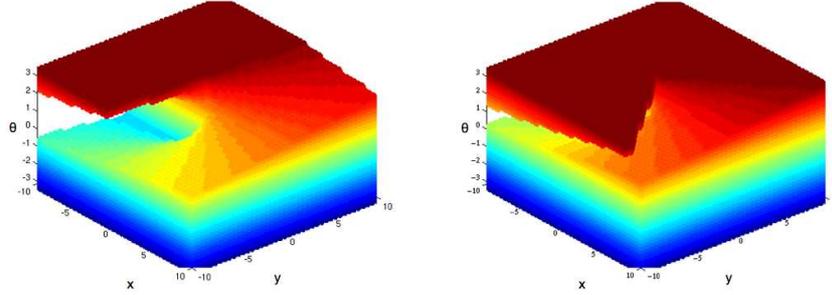


Figure 4: Admissible configuration space \mathcal{C}_{adm}^ϕ in the case of a (x, y, θ) configuration space: visibility acts as a virtual obstacle in $SE(2)$. The obstacle is depicted for $(\phi_-, \phi_+) = (-\frac{\pi}{2}, \frac{\pi}{2})$ (left) and $(\phi_-, \phi_+) = (-\frac{\pi}{3}, \frac{\pi}{3})$ (right) .

This means that the visibility and sensor angle limits constraints can be translated into virtual obstacles in $SE(2)$. From Equation 13, it is straightforward to deduce the admissible configuration space, which is $SE(2)$ minus these obstacles. Figure 4 depicts the virtual obstacle (actually, only one obstacle) in $SE(2)$ for $\phi_+ = -\phi_- = \frac{\pi}{2}$ (a) and $\phi_+ = -\phi_- = \frac{\pi}{3}$ (b), as the hollowed volume. It is worth noting that the free space resulting from this visibility obstacle is a single, helical-shaped component of $SE(2)$, which is smaller while the authorized pan range is smaller. We call \mathcal{C}_{adm}^ϕ the admissible configuration space resulting from the constraints 3 and 2.

As far as the range constraints of the inequalities in Equation 4 are concerned, they introduce two other virtual cylindrical obstacles which reduce the admissible configuration space into \mathcal{C}_{adm}^r . Finally the combination of these constraints gives rise to the admissible configuration space:

$$\mathcal{C}_{adm} = \mathcal{C}_{adm}^\phi \cap \mathcal{C}_{adm}^r.$$

A simpler characterization can be made in (x, y, ϕ) , instead of (x, y, θ) . These representations are equivalent, since ϕ and θ are related by Equation 2, but the admissible configuration space, as depicted on Figure 5, is easier to handle, as the constraints over ϕ (inequalities in Equation 3) do not depend on x or y . As a result, \mathcal{C}_{adm}^ϕ is simply the space between the two planes $\phi = \phi_-$ and $\phi = \phi_+$, and \mathcal{C}_{adm} is the intersection of this volume with \mathcal{C}_{adm}^r . The advantage of this representation is that it makes the task of determining a complete algorithm for the holonomic version of the DDR easier.

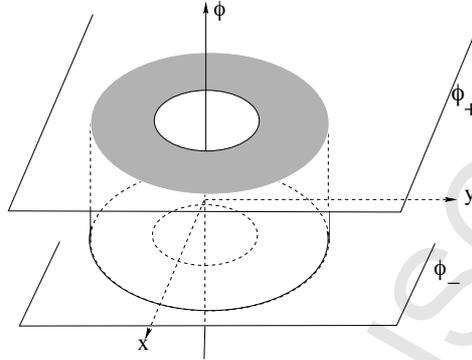


Figure 5: \mathcal{C}_{adm} for a (x, y, ϕ) configuration space, delimited by two horizontal planes on ϕ and two vertical cylinders.

6.2. Finding a path for a holonomic robot among obstacles

Lets assume that our DDR is disc-shaped and that we are given a (virtual) holonomic robot with the same circular shape, i.e. we will first ignore the non-holonomy constraints. The holonomic robot evolves in a plane filled with obstacles and has to respect the visibility constraint.

The free space \mathcal{C}_{free} is the set of configurations inside \mathcal{C}_{adm} (1) not in collision with the physical obstacles and (2) not in the shadow areas created by these same obstacles w.r.t. a light located in \mathcal{L}_1 . We build it explicitly on top of \mathcal{C}_{adm} as depicted in Figure 6. From now we also introduce the projection on \mathbb{R}^2 of the free and admissible spaces, denoted with a superscript g . Hence, let us consider the projection \mathcal{C}_{adm}^g of \mathcal{C}_{adm} on the xy plane, without considering ϕ , as the obstacles shape remain identical while ϕ is changing.

The obstacles have a two-fold effect, as they prevent the robot presence (1) by a physical collision with them and (2) by hiding the landmark \mathcal{L}_i from the current position. First, all the physical obstacles, dilated by the circular robot, are subtracted from \mathcal{C}_{adm}^g (Figure 6, upper right part). We get, in white, the *collision obstacles*. In a second step, we remove the obstacles shadows w.r.t. the origin. We get, in pink, the *visibility (virtual) obstacles* (Figure 6, lower left part). Finally, we integrate the sensor range constraints by defining interior and exterior cylinders. The resulting projection of \mathcal{C}_{free} (Figure 6, lower right part) is delimited by arcs of circles and straight-line segments. Note that the obstacles do not depend on the values of ϕ (but

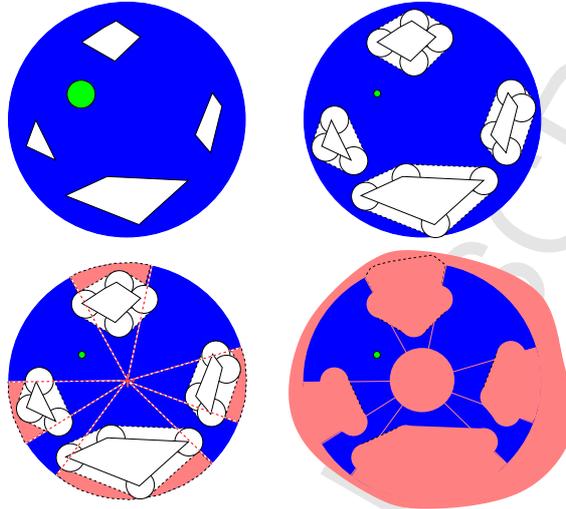


Figure 6: Construction of \mathcal{C}_{free}^g for the (x, y, ϕ) representation. By dilating physical obstacles to define collision obstacles (polygons, top left), the circular robot (disc, top left) can be reduced to a point. Shadows and visibility constraints then define visibility obstacles (lower right).

on θ), so that, in the x, y, ϕ space, \mathcal{C}_{obst} is made of cylinders in $SE(2)$, by translating the projection of Figure 6, right along the ϕ axis.

Also note that \mathcal{C}_{obst} is not always made of one singly-connected component as in Figure 6: the dilation of two obstacles may be connected and create a portion of the free space not connected with the central area, e.g. in Figure 7, \mathcal{C}_{free} is made of three different connected components.

Now, let \mathcal{C}_{free}^g be the domain in the xy plane that induces \mathcal{C}_{free} . Several complete algorithms can generate a path for a 2D point in \mathcal{C}_{free}^g , e.g. by building a roadmap capturing the domain connectivity Latombe [1991]. Among them, we chose the Generalized Voronoi Graph (GVG) approach, a classical tool for the construction of a roadmap in the environment. Obstacles here are made of arcs of circles and line segments, hence the GVG is made of arcs of parabola (circle-line), hyperbola (circle-circle) and line segments (line-line).

The GVG approach gives a complete algorithm for finding a path for a 2D point in \mathcal{C}_{free}^g by connecting the desired start and end points to the generated

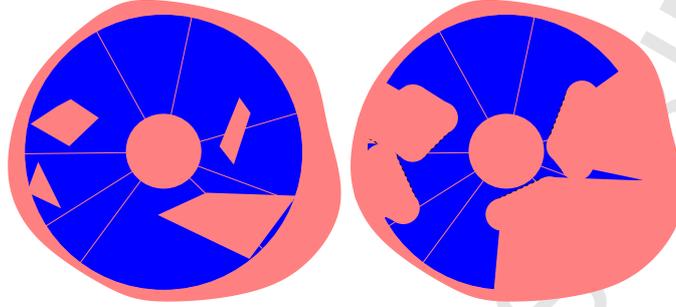


Figure 7: A configuration of physical obstacles (left) such that \mathcal{C}_{free}^g is made of three connected components (right).

graph Latombe [1991]. By using this classical result, we can now make plans in the projection \mathcal{C}_{free}^g and deduce plans for \mathcal{C}_{free} , with the following result.

Theorem 1. *The problem of planning a path in $\mathcal{C}_{free} \subset SE(2)$ for a holonomic, circular robot with visibility constraints on both range and angular displacement of its sensor is reducible to the problem of finding a path for a single point in $\mathcal{C}_{free}^g \subset \mathbb{R}^2$.*

Proof. Let $P_s = (x_s, y_s, \theta_s)^T$ and $P_e = (x_e, y_e, \theta_e)^T$ be two free initial and final configurations. By construction, the 2D points $(x_s, y_s)^T$ and $(x_e, y_e)^T$ belong to \mathcal{C}_{free}^g . Suppose that we can find a path s^g connecting them in \mathcal{C}_{free}^g ,

$$\begin{aligned} s^g : [0, 1] &\rightarrow \mathcal{C}_{free}^g \\ s^g(0) &= (x_s, y_s)^T, \quad s^g(1) = (x_e, y_e)^T, \quad s^g(t) = (x(t), y(t))^T. \end{aligned}$$

Then, if ϕ_s, ϕ_e are the sensor angle (given by Equation 2) at the initial and final configurations, we will define a path for the angle, that can be combined with s^g to provide an acceptable path in $SE(2)$:

$$\begin{aligned} s_\theta : [0, 1] &\rightarrow SO(2) \\ s_\theta(t) &= \arctan 2(y(t), x(t)) - (1-t)\phi_s - t\phi_e + \pi. \end{aligned}$$

The function s_θ is continuous on $[0, 1]$ since s^g is also continuous and $(x(t), y(t)) \neq (0, 0)$. We can define the following continuous path in $SE(2)$:

$$\begin{aligned} s : [0, 1] &\rightarrow SE(2) \\ s(t) &= (x(t), y(t), s_\theta(t))^T. \end{aligned}$$

It satisfies the initial and final constraints, and, for all $t \in [0, 1]$, $(1 - t)\phi_s + t\phi_e \in [\phi^-, \phi^+]$, i.e. the visibility constraint is satisfied at every point.

Conversely, if we are not able to find any free path in \mathcal{C}_{free}^g , we cannot have any free path in $SE(2)$: if there were, its projection on \mathbb{R}^2 would also be free, which contradicts our initial assumption. Hence, an algorithm that solves the planning problem in \mathcal{C}_{free}^g also solves the problem in \mathcal{C}_{free} . \square

7. Planning paths for an non-holonomic robot

In this section, the previous results are used to propose a complete planner for a circular-shaped DDR navigating among obstacles and having to maintain a landmark in sight, whereas its sensor is under range and angular constraints. Inspired from the classical roadmap-based approach, we implemented a simple planning algorithm according to the following steps:

1. Build $\mathcal{C}_{obst}^g = \overline{\mathcal{C}_{free}^g}$ by taking the union of the dilated obstacles with the shadows induced by the landmark visibility, as explained in 6.2;
2. Build the GVG on \mathcal{C}_{free}^g ; as \mathcal{C}_{free}^g is made of line segments and arcs of circle (from the obstacles dilation), the resulting GVG is made of line segments (line-line equidistance) and arcs of parabola (line-circle equidistance) or hyperbola (circle-circle equidistance);
3. Given starting and goal configurations, connect them to the GVG and compute a path \hat{s} for the *holonomic* system associated to the robot (see 6.2); if not possible, a non-holonomic path does not exist;
4. Recursively connect the starting and ending points with the globally optimal path (obtained from the synthesis in Salaris et al. [2010]); if the optimal path is in collision, then use as a sub-goal the point at middle-path in \hat{s} and apply the *recursive* procedure to the two resulting pairs of points.

Figure 8 (left) shows a path connecting an initial and final configurations, composed of two spirals and rotations in place. The only obstacles present in this case are the visibility obstacles and the cylinders at $d^{(1)}$ and $D^{(1)}$. The Figure 8 (center) shows the first part of step 1, i.e. the construction of the shadows induced by the landmark, and Figure 8 (right) shows the dilated obstacles (dashed red areas) and the GVG build in step 2.

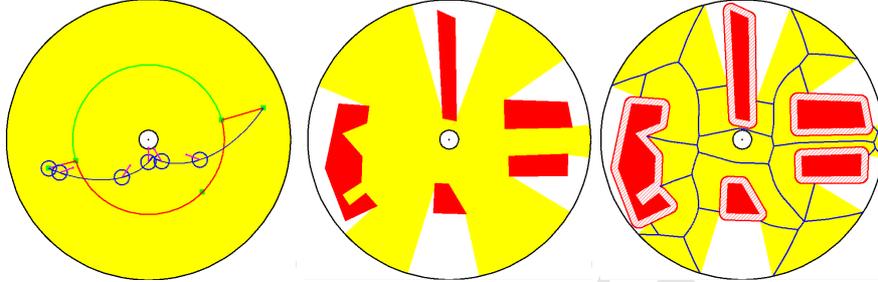


Figure 8: (Left) Path for a DDR under visibility constraints alone (angular and distance ranges, represented by the inner and outer circles). The robot shape, heading, and gaze are drawn here, and then omitted for the sake of readability. The recursion has only a one level depth: a direct connection of the initial and final configurations with optimal paths is possible) (Center) Obstacles (red) and regions visible from the landmark (yellow). (Right) Construction of \mathcal{C}_{free}^g : dilated obstacles (dashed red) are removed from the visible area and underlying GVG.

7.1. Choice of the underlying holonomic planner

As mentioned above, there are several choices in the literature as for the holonomic planner. The optimal trajectories (in terms of Euclidean distance) would be given by the visibility graph Latombe [1991] but the resulting paths would be in contact with the real obstacles, which is not always desirable, in particular not in our case, since, for proving the algorithm convergence, we need the path to be included in the open of \mathcal{C}_{free}^g . The Voronoi has the advantage of maintaining the robot as far as possible from the obstacles and its resulting path are in the open of \mathcal{C}_{free}^g .

Another advantage of a Voronoi-based holonomic planner, is that we can tune the graph to prefer some trajectories to other, by annotating the Voronoi graph edges. The edges can be weighted with their lengths, to favor shorter paths, with some decreasing function of the minimal clearance along the edge, to favor safer paths, or with a combination of both.

7.2. Analysis of the metric induced by shortest paths

The lengths along the shortest paths defined by Bhattacharya curves (spirals, lines) clearly establish a metric in \mathcal{C}_{free}^g . Let us prove that this metric is locally equivalent to the Euclidean metric in \mathbb{R}^2 . Let $P_s = (x_s, y_s, \theta_s)^T$ and $P_e = (x_e, y_e, \theta_e)^T$ be a pair of initial and final points in the free space, and $P_s^g = (x_s, y_s)$, $P_e^g = (x_e, y_e)$ their projections on \mathcal{C}_{free}^g . As recorded in 5, there are just a few kinds of shortest paths in a close neighborhood of point

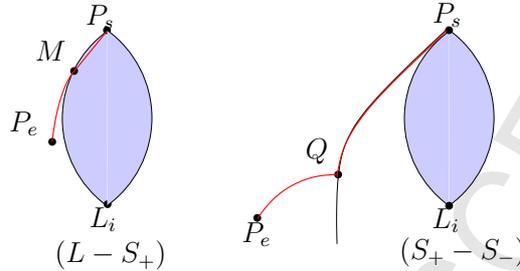


Figure 9: Shortest paths including logarithmic spirals: concatenation of a line segment and a ϕ_+ spiral ($L - S_+$, left), and concatenation of a ϕ_+ and a ϕ_- spiral ($S_+ * S_-$, right).

P_s : line segments (its length obviously equal to the Euclidean distance in \mathbb{R}^2), concatenations of a line segment and a logarithmic spiral at saturated ϕ (Figure 9, left) and concatenations of two spirals at saturated opposite values of ϕ (Figure 9, right). The other possible primitives (e.g. $L * L$) concern points “far” from point P_s (we will see in the following paragraphs how far they are). Here, we will use exclusively the norm $\|\cdot\|$ in \mathbb{R}^2 .

Our approach is the following: (1) We show that we can always choose a neighborhood of the initial configuration such that no $L - L$ primitives should be executed; (2) We show that $L - S$ and $S - S$ primitives induce a metric equivalent to the Euclidean one; (3) Since the remaining possible primitives are by essence shorter than the $S - S$ ones, we conclude that in a sufficiently small neighborhood of P_s^g , our optimal motion primitives induce paths of lengths equivalent to the Euclidean metric.

First, note that the length of a logarithmic spiral arc keeping ϕ constant, starting from $P_0 = (r_0, \alpha_0)$ and reaching a point $P_1 = (r_1, \alpha_1)$ is

$$l_\phi(P_0, P_1) = \frac{r_0}{\cos \phi} \left| 1 - e^{\frac{\alpha_0 - \alpha_1}{\tan \phi}} \right|. \quad (14)$$

Optimal paths in a neighborhood of P_s^g . Let us examine the nature of shortest paths for goal configurations located in a small neighborhood of the starting configuration P_s^g . Recall that in Figure 3, and keeping in mind the minimal range constraint, one can see that it is straightforward to choose a neighborhood around P_s^g in order to ensure that the paths to reach the points of this neighborhood are only lines, concatenations of one spiral and a line, and concatenations of two spirals. Notice that the choice of this neighborhood depends on ϕ^+ . Indeed, the Figure 10 represents the distribution of optimal

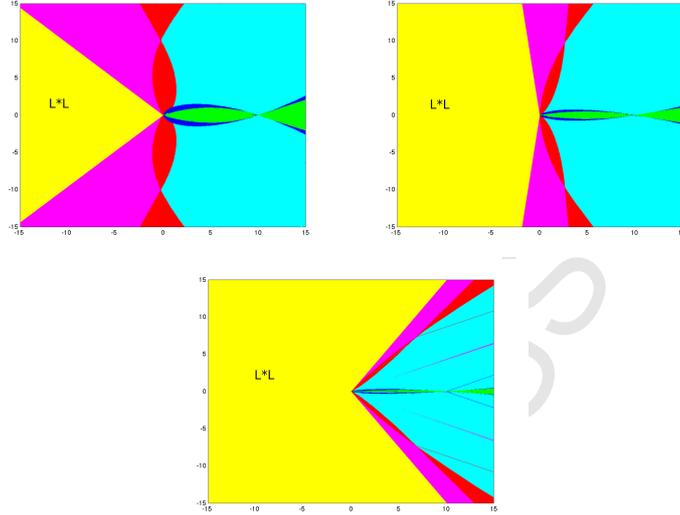


Figure 10: Structure of the optimal paths around the segment $[L_i P_s]$ for varying values of ϕ_+ (respectively $\frac{\pi}{8}$, $\frac{\pi}{16}$, $\frac{\pi}{40}$).

paths in a given portion of the plane, for three “small” values of the field of view, $\frac{\pi}{8}$, $\frac{\pi}{16}$, and $\frac{\pi}{40}$. One can observe that in a small neighborhood of P_s , the optimal paths are still to be found among the L , $L - S$, $S - L$ and $S * S$ primitives. However, when ϕ_+ tends to 0 (e.g. for $\frac{\pi}{40}$), the limits of the region corresponding to $L * L$ primitives tend to the x -axis. As a consequence, for a given fixed neighborhood of P_s^g , it is possible to choose ϕ_+ so that all primitives could be done in this neighborhood.

This fact is without consequence for the optimal paths $L - S * S$, $S * S - L$, $L - S * S - L$, since they are by construction Hayet and Murrieta-Cid [2009] shorter than the $S * S$ ones for a given pair of starting and ending configurations. Hence, the inequalities that we will establish for $S * S$ and $L - S$ also hold. Nevertheless, this is not the case of $L * L$ primitives. They differ from the other paths, since the metric they induce is not equivalent to the Euclidean one.

Fortunately, we showed in Hayet and Murrieta-Cid [2009] that the $L * L$ region boundaries (the brightest regions in Figure 10) are given by a straight line making an angle of $2\phi_+ - 4 \tan \phi_+ \log \sin \phi_+$ with the horizontal and by its symmetric w.r.t. the x axis. This means that it tends to collapse to the

x -axis at a convergence rate of $-4\phi_+ \log \phi_+$. As a consequence, if we suppose that P_s is separated from the origin L_1 by at least $d^{(1)}$, it is always possible to choose a neighborhood of P_s with a size in $O(d^{(1)}\phi_+)$ (which converges faster to 0 than $\phi_+ \log \phi_+$), that will guarantee that no $L * L$ primitive needs to be done in this neighborhood. As a conclusion, it is possible to describe the metrics of the optimal primitives around P_s by *only considering the $S * S$ and $L - S$ cases*.

Furthermore, notice that, in the recursion, these $L * L$ paths necessarily collide with the minimal range constraint virtual obstacle and induce new recursive calls, hence *they never need to be executed*.

Case of a line segment and a spiral ($L-S$). Here, we consider the case of a line followed by a S_+ spiral, the $L - S_-$ case being symmetric. We have (see Figure 9, left) the distance between P_s and P_e through

$$d_B(P_s, P_e) = \|P_s^g M\| + l_{\phi_+}(M, P_e^g).$$

One can refer to Hayet and Murrieta-Cid [2009] to make this distance more explicit, and we develop this in Appendix A, to get the following inequality:

$$r_{P_s} |\alpha_{P_e} - \alpha_{P_s}| \leq \|P_s^g P_e^g\|, \quad (15)$$

so that,

$$\|P_s^g P_e^g\| \leq d_B(P_s, P_e) \leq \frac{1}{\sin \phi_+} \|P_s^g P_e^g\|. \quad (16)$$

*Case of two spirals ($S * S$).* Consider a S_+ spiral followed by a S_- spiral (Figure 9, right). See Appendix A for a detailed version of the calculus.

By using the Equation 14 twice, and with $t_+ \stackrel{\text{def}}{=} \tan \phi_+$,

$$\begin{aligned} d_B(P_s, P_e) &= l_{\phi_+}(P_s, Q) + l_{\phi_-}(Q, P_e) \\ &= \frac{r_{P_s}}{\cos \phi_+} (1 - e^{-\frac{\alpha_{P_s} - \alpha_Q}{t_+}}) + \frac{r_{P_e}}{\cos \phi_+} (1 - e^{-\frac{\alpha_Q - \alpha_{P_e}}{t_+}}). \end{aligned}$$

After developments (see Appendix A), we finally get the same inequality as in the $L - S$ case, i.e.

$$\|P_s^g P_e^g\| \leq d_B(P_s, P_e) \leq \frac{1}{\sin \phi_+} \|P_s^g P_e^g\|.$$

The previous inequalities allow us to write the following theorem,

Theorem 2. For any constant $K > 1$, there exists a neighborhood size $\epsilon > 0$ such that for all the optimal motion primitives in the absence of obstacles, the length of the path in \mathbb{R}^2 between two configurations $P_s = (x_s, y_s, \theta_s)^T$ and $P_e = (x_e, y_e, \theta_e)$ closer than ϵ , satisfies

$$\|P_s^g P_e^g\| \leq d_B(P_s, P_e) \leq \frac{K}{\sin \phi_+} \|P_s^g P_e^g\|.$$

The $\frac{1}{\sin \phi_+}$ factor comes with no real surprise, as, when ϕ_+ gets smaller (i.e. when the field of view of the camera shrinks), the maneuverability of the robot also gets smaller, and, for example, for a pair of close starting and ending points, a concatenation of two spirals involves a more “peaky” transition. A natural corollary is the following one,

Corollary 1. For a given configuration c in \mathcal{C}_{free} , and a neighborhood $N(c_g)$ of its projection c_g in \mathcal{C}_{free}^g , there exists another neighbourhood $N'(c_g)$ of c_g such that for all configurations c' such that $c'_g \in N'(c_g)$, the shortest path from c_g to c'_g is completely included in $N(c_g)$.

7.3. Properties of the recursive algorithm

Completeness. An important property of our system is that, in the interior of \mathcal{C}_{free} , it satisfies the LARC condition as a normal DDR. Moreover, it is a symmetric system, so that it is small-time locally controllable Choset et al. [2005], i.e. it can follow any curve in the interior of \mathcal{C}_{free} arbitrarily closely. This argument is used e.g. in Laumond et al. [1994] for the convergence of a recursive scheme based on the use of time-optimal trajectories. Here, because we use the optimal synthesis described in 5, this same argument cannot be used, as some of the points in the neighborhood of any starting configuration have to be reached by paths that cannot be enclosed in an arbitrarily small neighborhood of the starting configuration in \mathcal{C}_{free} . This is the case, for example, of the configurations reachable from the starting one such that the shortest path includes two consecutive spirals (this implies that the angle will move at least of $2\phi_+$).

Nevertheless, the same kind of coverage of the holonomic paths with small neighborhoods can be done in \mathbb{R}^2 yielding to nice convergence properties. The topological properties of the previous section help us in proving the *completeness*, and, even more, in studying the complexity of the algorithm.

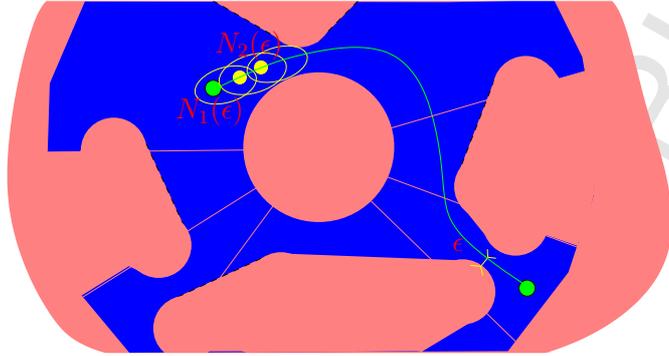


Figure 11: Convergence of the recursive algorithm is ensured by the topological property that the optimal curves have in \mathbb{R}^2 .

Indeed, given a pair of starting and ending configurations in \mathcal{C}_{free} , one will first notice that, by theorem 1, the existence of a solution for the associated holonomic robot in \mathcal{C}_{free} is an equivalent problem to the existence of a free path in \mathcal{C}_{free}^g . Because of the topological properties above, we can follow any curve in \mathcal{C}_{free}^g with the primitives from 5, so that the completeness of our algorithm is guaranteed whenever the planner which is used for the underlying holonomic system is itself complete. Here, the Voronoi-based algorithm for the underlying holonomic system being complete, we conclude that the recursive algorithm using the primitives from 5 is also complete.

Convergence and complexity. The same topological properties ensures convergence, and gives us an order of the rate of convergence of the recursive algorithm. Let ϵ be the minimal clearance along the path. For that value ϵ , we can deduce by the equivalence of the metrics induced by the optimal primitives with the Euclidean distance, that there exists some $\delta > 0$ such that for any two pairs of configurations separated by an Euclidean distance inferior to δ , the shortest path induced by the optimal primitives from 5 is totally included in a neighborhood of both configurations, $N(\epsilon)$, of diameter ϵ . The theorem 2 tells us that $\delta = O(\epsilon \sin \phi_+)$.

The path computed for the holonomic robot in \mathcal{C}_{free}^g being compact and included in the open of \mathcal{C}_{free}^g , it is possible to cover up this path by a *finite* number of such neighborhoods $N_i(\epsilon)$ totally included in the open of \mathcal{C}_{free}^g , e.g. as depicted in Figure 11. This way, to each $N_i(\epsilon)$, is attached a portion of

the path of length δ . The number of such neighborhoods is $O(\frac{1}{\delta})$, with δ proportional to $\sin(\phi_+)\epsilon$, i.e. we have $O(\frac{1}{\epsilon \sin(\phi_+)})$ of such neighborhoods, which means that the recursion, at the *worst case*, leads to $O(\frac{1}{\epsilon \sin(\phi_+)})$ maneuvers with optimal primitives. The number of rotations in place is at maximum of 3 per maneuver (one at the beginning, one, eventually, in the $S * S$ case, and one at the end) so that the total number of rotations in place is also $O(\frac{1}{\epsilon \sin(\phi_+)})$.

8. Planning paths among a set of landmarks

Based on the one-landmark algorithm explained above, we propose a solution to the problem of generating paths ensuring *at least one* visible landmark along its path, i.e. ensuring that a way of localizing itself is always available to the robot. The idea is to generate a graph G_i for each of the available landmarks, then to connect these graphs at what we call *switch points* (points where the robot changes safely from one landmark to another).

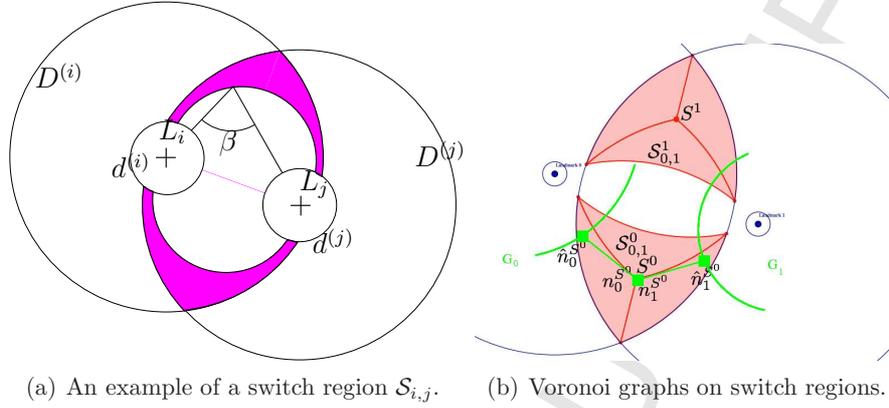
8.1. Switch regions

Our strategy is to connect the different roadmaps, i.e. graphs G_i from each landmark \mathcal{L}_i , through the use of *switch regions*. The constraints for a switch to be possible are (1) that the two landmarks are in the sensor range and (2) that at this point, it is possible to change the orientation of the sensor so that the robot can always see at least one of the two landmarks.

The first constraint defines a region of common visibility $\mathcal{V}_{i,j} = \mathcal{C}_{free,i}^g \cap \mathcal{C}_{free,j}^g$. The second constraint prohibits a region $\mathcal{A}_{i,j}$ formed by the points P such that the angle $\angle L_i P L_j$ is superior to the robot sensor field of view β . This region $\mathcal{A}_{i,j}$ is delimited by two arcs of circle, and the resulting switch area is $\mathcal{S}_{i,j} = \mathcal{V}_{i,j} \setminus \mathcal{A}_{i,j}$. In Figure 12(a), one can observe a switch region (shaded) $\mathcal{S}_{i,j}$, in the absence of obstacle, formed by just two connected components. In the presence of obstacles, this structure may be more complex.

A priori, we do not need to establish the entire connectivity on each $\mathcal{S}_{i,j}$, we just need to use these regions to provide intermediate points for the computation of holonomic and non-holonomic paths joining parts of the space where the only visible landmarks are respectively i and j . In fact, even one single point of $\mathcal{S}_{i,j}$ in each connected component of the union is sufficient to allow for all switching strategies, as we state it below.

Theorem 3. *Let $\{Q_{ij}^k\}$ be a set of interior points associated to each connected component of the union of the two free spaces \mathcal{C}_{free}^g . If we connect the Q_{ij}^k to*



(a) An example of a switch region $\mathcal{S}_{i,j}$. (b) Voronoi graphs on switch regions.

Figure 12: Left: an example of a switch region $\mathcal{S}_{i,j}$, for the case of two close landmarks \mathcal{L}_i and \mathcal{L}_j . \mathcal{A}_{ij} is the region delimited by the two arcs of circles passing through L_i and L_j from which $\angle L_i P L_j = \beta$. Right: GVG computed on each connected component of $\mathcal{S}_{0,1}$. The connected component 0 has one interior point S^0 for which we add four nodes in the graph, $n_0^{S^0}$, $\hat{n}_0^{S^0}$ connected to G_0 and $n_1^{S^0}$, $\hat{n}_1^{S^0}$ connected to G_1 .

the graphs G_i and G_j , and denote the resulting graph as G , augmented with the two initial and final configurations P_s and P_e , the problem of determining the existence of a holonomic path between P_s and P_e guaranteeing that at least one of \mathcal{L}_i or \mathcal{L}_j is in sight is equivalent to finding a path in G .

Proof. As the points Q_{ij}^k belong to the two free spaces relative to both landmarks \mathcal{L}_i and \mathcal{L}_j , then they can be connected to both Voronoi graphs G_i and G_j . Furthermore, finding a holonomic path satisfying the visibility of at least one landmark is equivalent to find a holonomic path in the union of the free spaces. The newly formed graph G captures the connectivity of this union. Indeed, any connected component in the union is either entirely in one of the free space (in which case its connectivity is captured by a single G_i) or partly in both. In that last case, any pair of points in the connected component can be connected through G : if they are in the same visibility region, this is through one of the original graphs, otherwise, the one that sees \mathcal{L}_i can be connected to Q_{ij}^k , and the one that sees \mathcal{L}_j equally. \square

However, using only one point per connected component of the union could make the global paths quite far from the optimal ones. For instance, if one considers Figure 12(a), and if one uses just one point in one of the connected components of $\mathcal{S}_{i,j}$, e.g. in the lowest part of the upper connected

component, some of the paths connecting configurations located in the upper part of the graph would be penalized. Several strategies could be possible, for example based on sampling, but one could lose the completeness of the algorithm. Instead, here, we propose to compute explicitly all the connected components of $\mathcal{S}_{i,j}$, and, for each of these components, we capture the connectivity of the component with a Voronoi graph, as illustrated by Figure 12(b).

8.2. Planning trajectories among the set of landmarks

To plan trajectories among a whole set of landmarks, we use the switch areas to join the different roadmaps, each proper to one landmark, together. For each landmark \mathcal{L}_i , we generate G_i as explained above (case of one landmark). Then, switch regions $\mathcal{S}_{i,j}$ are computed as in 8.1 and a set of switch points is computed by taking the interior nodes on the Voronoi graphs of the connected components $\mathcal{S}_{i,j}^k$ of $\mathcal{S}_{i,j}$. For each switch point S , four new nodes are created: in G_i , a node n_i^S at location S and a node \hat{n}_i^S located at the closest point from S among the edges of G_i ; in G_j , nodes n_j^S and \hat{n}_j^S are created in the same way. Then, the edge (n_i^S, \hat{n}_i^S) (resp. (n_j^S, \hat{n}_j^S)) is added to G_i (resp. G_j), as it can be shown in Figure. 12(b). Note that the graphs G_i , G_j and the one defined in $\mathcal{S}_{i,j}$ can intersect or not, depending on the obstacles, and on the relative position of the landmarks. Furthermore, the graphs G_i may even be completely outside of the $\mathcal{S}_{i,j}$, but since this region is contained in the intersection of the individual visibility areas, any point of it can be connected to G_i or G_j .

For solving a query, given a pair of starting and goal points P_s and P_e , we first check which landmarks these points may observe. For each of these \mathcal{L}_k , we add nodes corresponding to the starting/goal point to graph G_k , following the same procedure as the switch points above. A multi-landmark graph G is built by taking the union of all G_i and by adding edges between the switch points corresponding to the same locations, i.e. edges (n_i^S, n_j^S) .

Following the strategy we described for the single landmark case, we compute a shortest path \hat{s} in G for the *holonomic* system associated to the robot; this involves evaluating several shortest paths in G , for all pairs of possible start/end nodes (i.e. we have to choose which landmarks to start and end with). If this is not possible, then we can deduce that *there does not exist a non-holonomic path satisfying all the imposed constraints*; we are also sure that if a holonomic path exists, it will be found: thanks to the G_i , any point aiming at L_i is attainable, and thanks to the \mathcal{S}_{ij} connected components, we can also “attain” any landmark switch.

Once \hat{s} is computed, we decompose it in its landmark layers, i.e. components sharing the same landmark visibility,

$$\hat{s} = \hat{s}_{i_1} \cup (n_{i_1}, n_{i_2}) \cup \hat{s}_{i_2} \cup (n_{i_2}, n_{i_3}) \cup \dots \cup \hat{s}_{i_K},$$

where i_k is the landmark to which the part \hat{s}_{i_k} is relative to. Then, the problem is divided in instances similar to Hayet et al. [2010], i.e. finding non-holonomic paths satisfying the sensory constraints for just one landmark. The solutions \tilde{s}_{i_k} of these sub-problems are then merged into a single (non-holonomic) path \tilde{s} by adding switch primitives denoted as $C_{i_k, i_{k+1}}$ (i.e. change in the sensor orientation and rotation in place) between the \tilde{s}_{i_k} . This leads to algorithm 1.

8.3. Algorithmic complexity

Algorithm 1 inherits the complexity of its sub-tasks:

- visibility operations and boolean set-operations on polygons to get $\mathcal{C}_{free,i}^g$ (the $D^{(i)}$ and $d^{(i)}$ are transformed into polygons in the implementation). The computation of visibility regions induces $O(Nn \log n)$ computation time where n is the number of vertices in the environment and N the number of landmarks with the sweep line algorithm;
- the Voronoi computation Karavelas [2004] from line segments induce an expected computation time in $O(Nn \log^2 n)$;
- the switch areas may be computed in $O(nN^2)$ in the worst case but remains in most cases in $O(N)$ (landmarks overlap with a few others, and with a few polygons);
- queries imply the formation of graph G (linear in the number of switch points) and the computation of a shortest path in G (which depends on the graph); the output of the recursive procedure, i.e. the number of maneuvers, is varying in $O(\frac{1}{\sin \phi + \epsilon})$ where ϵ is the minimal clearance, as we saw in Section 7.3.

9. Experiments

This section presents a few simulation results with an implementation of this planner in C++, by using in particular the library CGAL for its Voronoi diagram implementation Karavelas [2004]. In this implementation, we took

Algorithm 1 Path planning ensuring one visible landmark among L_1, \dots, L_N

Roadmap construction.

- Build a representation of $\mathcal{C}_{obst}^g = \overline{\mathcal{C}_{free}^g}$ as the union of the obstacles dilated by the robot volume.
- for** $i = 1$ to N **do**
 - Build a representation of $\mathcal{C}_{obst,i}^g = \overline{\mathcal{C}_{free,i}^g}$ as the union of \mathcal{C}_{obst}^g with the shadows induced by \mathcal{L}_i visibility.
 - Build the GVG G_i on $\mathcal{C}_{free,i}^g$; $\mathcal{C}_{free,i}^g$ being made of parts of lines or circles, the GVG is made of parts of lines, parabolas or hyperbolas.

end for
Switch areas.
for $i = 1$ to N **do**
for $j = 1$ to $i - 1$ **do**

- Build a representation $\mathcal{S}_{i,j} = \mathcal{C}_{free,i}^g \cap \mathcal{C}_{free,j}^g$ of the switch area between landmarks i and j .
- Remove from $\mathcal{S}_{i,j}$ the areas in which the robot cannot have both i and j in sight: $\mathcal{S}_{i,j} \leftarrow \mathcal{S}_{i,j} \setminus \mathcal{A}_{i,j}$.
- for all** connected components $\mathcal{S}_{i,j}^k$ in $\mathcal{S}_{i,j}$ **do**
 - Compute a Voronoi graph for $\mathcal{S}_{i,j}^k$.
 - Connect the nodes of the resulting graph to $G_i (\{n_i^g\}_g)$ and $G_j (\{n_j^g\}_g)$.

end for
end for
end for
Query resolution.

- Connect P_s and P_g to all the graphs G_k s.t. \mathcal{L}_k is visible from the query point.
- Take $G = \cup_i G_i$ and add edges (n_i^g, n_j^g) for all g corresponding to switch points.
- Compute a shortest path \hat{s} in G among all paths starting from P_s and arriving at P_g .

if No such path exists **then**

- A solution does not exist.

else

- Write $\hat{s} = \hat{s}_{i_1} \cup (n_{i_1}^{g_1}, n_{i_2}^{g_1}) \cup \hat{s}_{i_2} \cup (n_{i_2}^{g_2}, n_{i_3}^{g_2}) \dots \cup \hat{s}_{i_K}$.

for $k = 1$ to K **do**

- Connect the starting and ending points of \hat{s}_{i_k} with the optimal primitives of 5; if collision, use the middle-path in \hat{s}_{i_k} as sub-goal and apply the recursive procedure to the two resulting sub-paths; result is \tilde{s}_{i_k} .
- *Optimize* \tilde{s}_{i_k} : (i) generate n_s randomly shortcuts between the constitutive primitives of the trajectory and select the one that most shortens the path, and (ii) iterate n_i times this procedure.

end for

- $\tilde{s} = \tilde{s}_{i_1} \cup C_{i_1, i_2} \cup \tilde{s}_{i_2} \cup C_{i_2, i_3} \dots \cup \tilde{s}_{i_K}$.

end if

advantage of the built Voronoi graph to set the edges weights in such a way that the edge length, on the one hand, and the minimal clearance along this edge, on the other hand, are combined, so as to find a compromise between shortest and clearest paths. In the same vein, we assign very large weights to the connector edges, i.e. those that connect two graphs G_i , in order to penalize landmark switches.

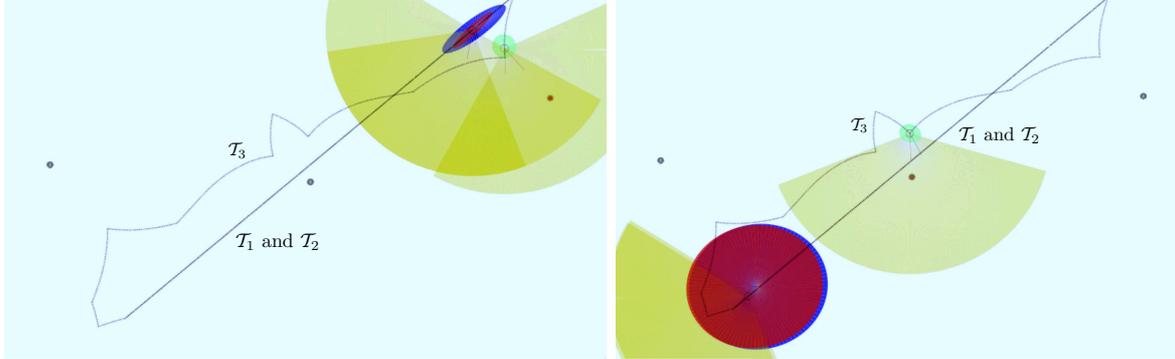


Figure 13: An illustration of the localization on the trajectories \mathcal{T}_1 , \mathcal{T}_2 and \mathcal{T}_3 . On the left, the robot can localize itself on the first landmark on \mathcal{T}_2 and \mathcal{T}_3 . On the right, the robot is localized on \mathcal{T}_3 only.

9.1. On the utility of our planner in trajectory-based navigation

We illustrate the utility of our planner through a few localization simulations. We generated 50 different environments, each of which is populated with 4 randomly located landmarks and free of obstacles, with the constraint that the resulting graph G is connected. We also generated two random start and goal configurations for this environment such that each of these configurations can be connected to G , i.e. it can see at least one of the landmarks.

For each of these environments and pairs of start/goal configurations, we plan and execute three trajectories \mathcal{T}_1 , \mathcal{T}_2 and \mathcal{T}_3 , based on two paths: (1) the shortest path in distance, i.e. the straight line connecting the configurations, that does not take into account the landmarks in the planning and (2) the path given by our planner, that generates “coastal navigation” paths Roy and Thrun [1999]. The trajectory \mathcal{T}_1 is a trajectory following the straight line, and for which the sensor is kept at $\phi = 0$; \mathcal{T}_2 is based on the same path, but it includes a rough “visual servoing” strategy that maintains the landmark in sight if possible. Finally, \mathcal{T}_3 follows the path computed by our algorithm.

In all cases, the resulting trajectories are executed while localizing the robot, if possible, with a simulation of localization that uses a common probabilistic dead-reckoning model and an observation model integrated in an Extended Kalman Filter scheme Thrun et al. [2005]. To measure the level of uncertainty that the robot has at some point, we evaluate the determinant of the covariance matrix Σ_t on the state X_t of the robot, $\det(\Sigma_t)$.

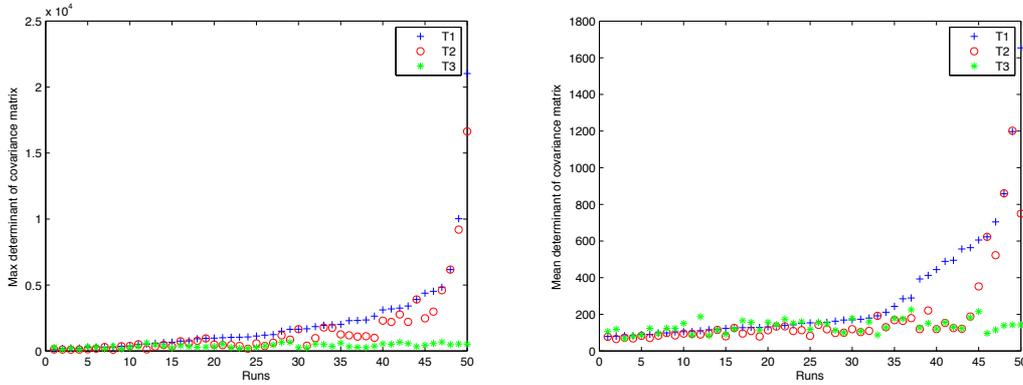


Figure 14: Maxima (left) and mean (right) values of the determinant of the uncertainty matrix for 50 random runs of the three strategies, \mathcal{T}_1 (crosses), \mathcal{T}_2 (circles) and \mathcal{T}_3 (dots). The values have been sorted according to the values obtained in \mathcal{T}_1 .

We illustrate this protocol in Figure 13, that shows, on the left, the execution of the three trajectories \mathcal{T}_1 (darker ellipse), \mathcal{T}_2 (dark ellipse) and \mathcal{T}_3 (bright ellipse). On two of the three trajectories, the robot localizes itself on the upper right landmark (and the position uncertainty is reduced). On the right, on the two trajectories that follow the straight line path, the robot cannot localize relative to any landmark; on the contrary, the robot is ensured all along \mathcal{T}_3 that it has at least one landmark to look at. A few statistics over the 50 random runs are also presented in Figure 14. These two figures show the maximum and mean determinant of the covariance matrix over (x, y, θ) along the different runs, where the runs are sorted according to the determinant values in the \mathcal{T}_1 trajectory. As expected, the determinant (its maximum or mean) remains lower or much lower in the case of the trajectory done based on the output of our algorithm. This justifies our previous reference to “coastal” navigation, i.e. ensuring landmarks visibility as we do brings implicitly some safety on how lost the robot will be. This trend is more pronounced in the case of the maximum value of the determinant along the trajectories, i.e. with the trajectories undergone with our planner path it is difficult for the robot to get lost, which is not the case for trajectories planned without taking into account the landmarks positions.

The choice of our planner paths induces paths that correspond naturally to a coastal navigation paradigm. However, this comes obviously with a cost

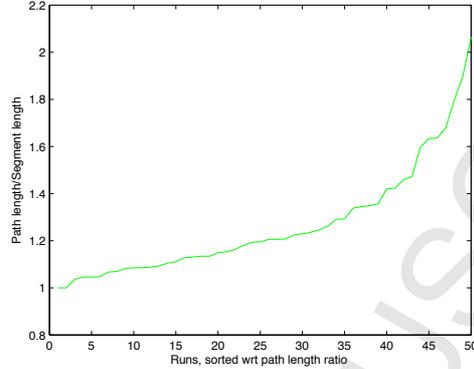


Figure 15: Length of path undergone in \mathcal{T}_3 , expressed as a ratio w.r.t. the straight line length (i.e. the path followed in \mathcal{T}_1 and \mathcal{T}_2 , for the same 50 random runs).

in terms of traveled distance. In order to evaluate roughly this cost, we depict in Figure 15, along all the 50 runs mentioned above, that depicts the ratio between our planner path length and the length of the straight line joining the start and goal configurations. This cost, as it can be seen, corresponds in most configurations to 10% to 80% of the straight line path.

9.2. Simulation of path planning among obstacles

We provide planning results in some interesting environment configurations. The first one is the one of Figure 16. The environment is populated with 5 landmarks and several obstacles generating visibility shadows. The brightest shaded region depicted in Figure 16(b) is the union of all visibility regions. The GVG is showed in Figure 16(c) and the final path \tilde{s} appears in Figure 16(d). By construction, it coincides at several points with \hat{s} . The final non-holonomic path is made of 16 non-trivial primitives (pieces of lines or spirals) and 12 rotations in place or landmark changes.

The second example is illustrated on Figure 17. It is a simpler example (just two landmarks and two obstacles), but there is a narrow passage that may have to be passed through. This narrow passage is located between the left obstacle and the minimal range circle around the right landmark, and a branch of the GVG naturally goes through it. For some of the pairs of start and goal configurations, the holonomic path \hat{s} may take this branch, as in Figure 17, and as expected, this leads to a number of maneuvers to be done.

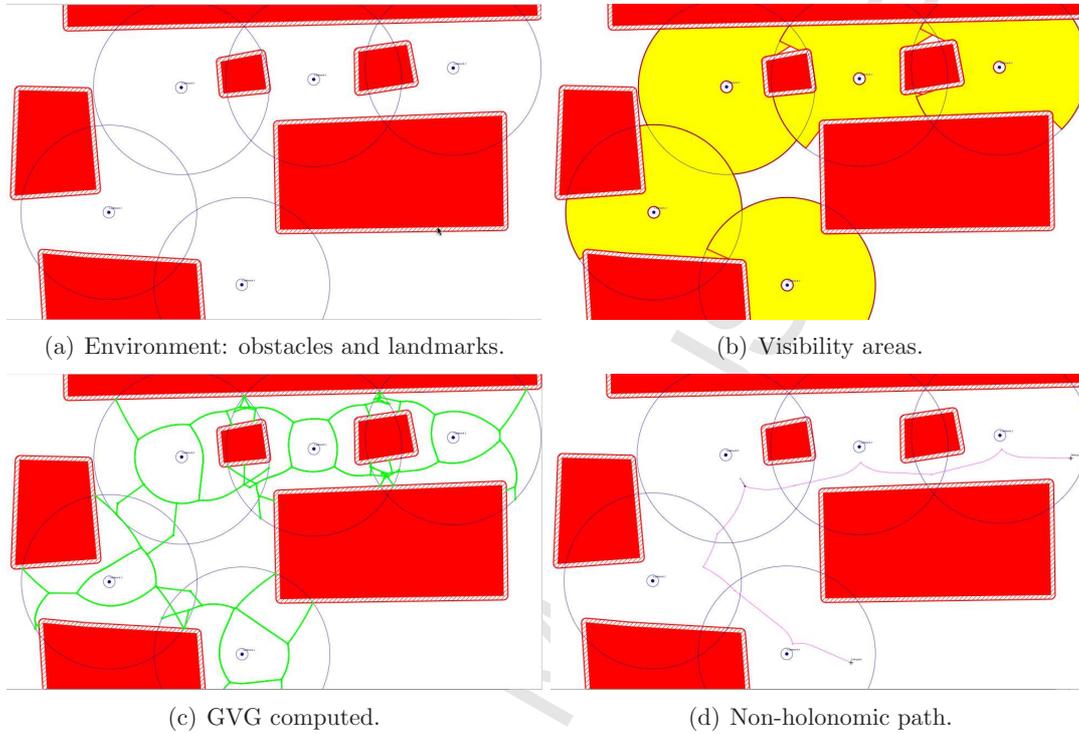


Figure 16: Example 1: computing path among obstacles. The final non-holonomic path is made of 16 non-trivial primitives (pieces of lines or spirals), and 12 rotations in place.

A zoomed view on this narrow passage is showed in 18, where a succession of spirals S_+ and S_- spirals can be observed. As mentioned above, a different choice of weights in the GVG (e.g., taking into account the clearance along the edge), could lead to paths avoiding the narrow passage.

9.3. Experiments on a DDR robot

Two different experimental setups, the first with one landmark and the second with two landmarks have been planned for and executed by a Pioneer P3-DX differential drive robot from Adept Mobilrobots. The obtained results are presented hereafter⁴.

⁴Videos of one and two landmark experiments on DDR robot can be found at http://www.cimat.mx/~cesteves/MotionPlanning_DDR_VisibilityConstraints/.

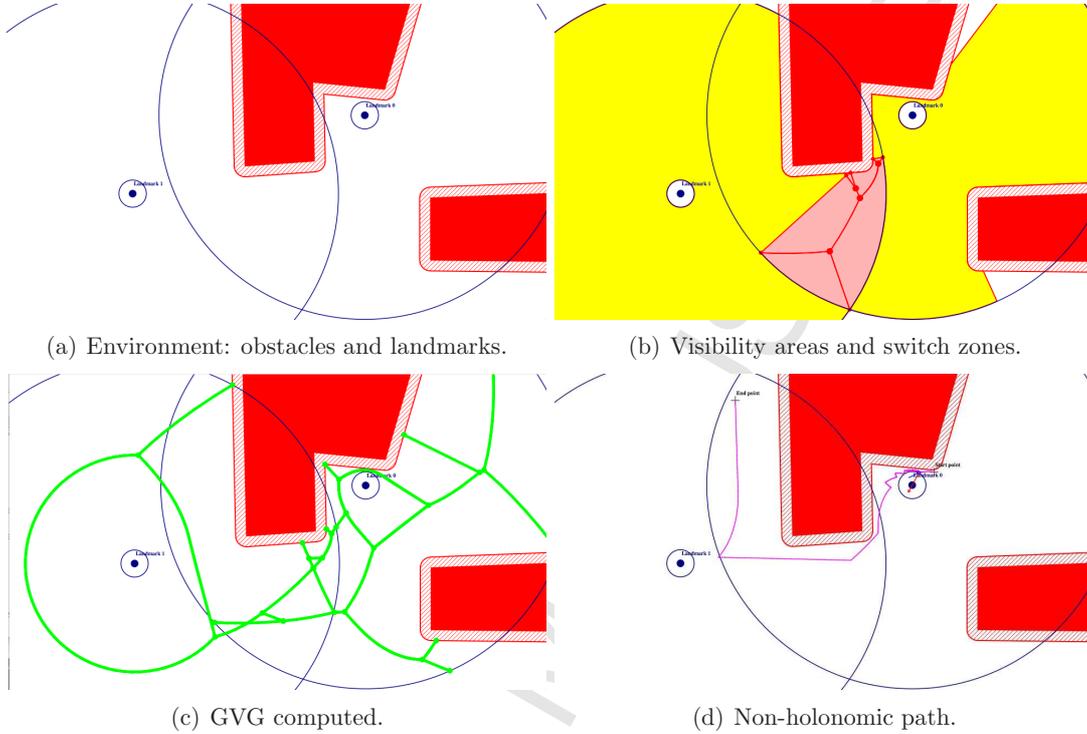


Figure 17: Example 2: a path among obstacles. The final non-holonomic path passes through a narrow passage, leading to a number of maneuvers, depicted in Figure 18.

9.3.1. An environment with one landmark

The experimental setup for this example is depicted in Figure 19. Figure 19(a) shows the map of the environment from an orthographic upper view. The red point shows the landmark that the robot has to keep within its field of view at all times. Start and end points are also shown. The obstacles are shown in dark grey. Figure 19(b) compares the planned and the executed trajectory measured with the onboard robot sensors. As it can be seen on the figure, the trajectory that keeps the landmark in sight at all times is composed of two rotations in place and two logarithmic spiral segments.

Figure 20 shows the robot view of the one-landmark experiment. In Figure 20(a) the view from the initial configuration of the robot is shown. Here, the landmark is outside the robot field of view. After an initial rotation in place, the robot gets the landmark inside its field of view. The camera

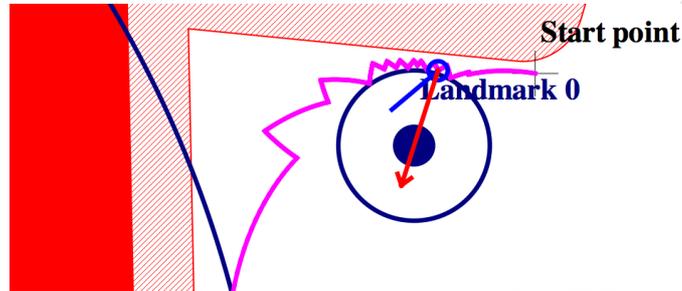


Figure 18: Example 2, zoom on the path computed in 17 in the narrow passage.

is saturated to the right, leaving the landmark (green and blue book) on the right side of the image. This can be seen in Figure 20(b). The next primitive in the planned trajectory, a segment of a logarithmic spiral, is executed approaching the landmark (Figure 20(c)) with a new rotation in place. The landmark is now centered at the left side of the image (Figure 20(d)). Finally, the last planned segment of logarithmic spiral is executed to reach the goal configuration, while keeping the landmark in sight.

Figure 21 shows the execution of the same trajectory from an external camera. Initial and goal configurations are shown in Figures 21(a) and 21(d) respectively. Figure 21(b) shows the the robot/landmark configuration after the first rotation in place. Note that the robot camera is aiming at the landmark (book in the image foreground). Figure 21(c) shows an intermediate configuration of the trajectory just before the second rotation in place. Here the robot is beginning the execution of the last two primitives of the planned trajectory: a rotation in place and a segment of logarithmic spiral.

9.3.2. An environment with two landmarks

Our second experiment takes place in a two-landmark setting in the same office environment as before, which makes the planning more constrained.

Figure 22(a) shows an upper view of the map with its initial and final configurations as well as both landmarks depicted as points, one red and one blue and their corresponding labels. Figure 22(b) shows a close-up of the computed trajectory with the starting position on the lower part of the image and the goal position on the upper-right corner. The resulting path is composed of several rotations in place and segments of logarithmic spirals. At around the point (1900,600), the robot switches from looking at one landmark to looking at the other. At some point while performing the switch,

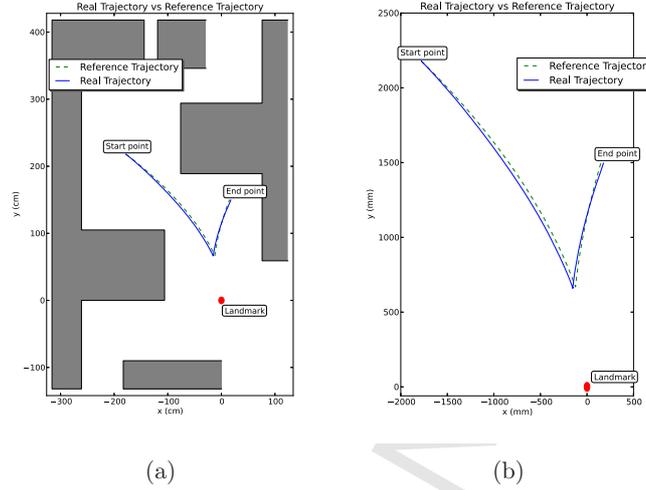


Figure 19: Experiment with one landmark to keep in sight. (a) 2D map of the environment with the landmark depicted as a (red) point. The planned trajectory with its start/end configurations is also shown (in blue). (b) Planned trajectory. The dotted line is the reference planned trajectory. The continuous line is the executed trajectory.

the robot has both landmarks inside its field of view. In this graph, the planned (dotted) and the executed trajectories are compared showing their similarity.

Figure 23 shows a few keyframes of the experiment taken from the robot point of view. Figures 23(a) and 23(h) show the initial and final configurations respectively. Figures 23(a) through 23(e) show the execution of the first part of the trajectory, where the robot sees the first landmark (the book) at all times. Figures 23(f) through 23(h) depict the execution of the second part of the trajectory, where the focus of the robot perception is on the second landmark (the basketball). Figure 24 shows the same experiment through an external camera showing different configurations of the robot during the execution of the trajectory. Notice the keyframe in Figure 24(d), which is the configuration where the robot is switching from looking at the book to looking at the basketball by performing a rotation in place.

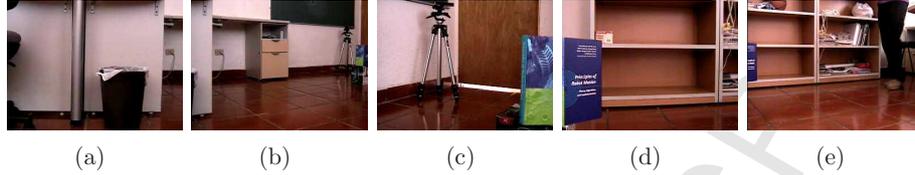


Figure 20: One-landmark example: (a) Initial configuration. (b) The robot turns towards the landmark (book at the right side of the image). (c) A logarithmic spiral is performed to approach the landmark, always keeping it at the right side of the image. (d) After a rotation in place, the landmark is seen on the left side of the image. (e) The final configuration is reached following a logarithmic spiral while keeping the landmark in sight.



Figure 21: One-landmark example, outside view of the experiment. (a) Initial configuration. (b) The robot aligns to see the landmark. (c) Configuration after the first logarithmic spiral and before rotation in place. (d) Final configuration after second logarithmic spiral.

10. Conclusion

In this paper, we have provided a motion planner to maintain landmarks visibility, in an environment populated with obstacles, for a differential drive robot equipped with a limited sensor. Our planner computes a path from an initial to a final configuration, such that, during its motion, the robot may switch from one landmark to the other, ensuring that always at least one landmark is visible to the robot. Based on the equivalence between the metric induced by the shortest path of our system and the Euclidian metric used to compute the distance between the robots and the obstacles, we have shown that if a collision free path exists for a holonomic robot in the presence of obstacles then this path can be always transformed in a feasible path — collision free and maintaining landmark visibility. We use the shortest path synthesis for this system to generate a candidate robot path; if the former is collision free, we use it to move the robot, i.e. we take the optimal path in terms of the Euclidian distance for maintaining visibility of a single landmark. If the path is in collision with an obstacle then it is recursively divided until

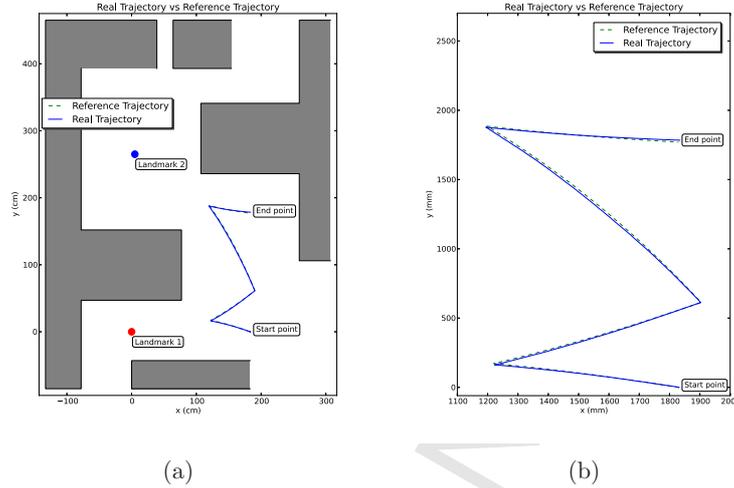


Figure 22: Two-landmark scenario. (a) Map of the environment with the initial and final configuration as well as the landmarks shown as points. (b) The computed trajectory and the real executed trajectory.

a collision free path is obtained. In this second case, we lose optimality, but we stress that the completeness of the method is guaranteed.

We have implemented all our algorithms, described path planning in challenging cases, and we have also presented simulation results showing that maintaining visibility of landmarks during the system motion helps to maintain the robot localization uncertainty bounded.

Although, our planner assumes a static map without uncertainty, we stress the fact that our planner is complete. It can determine whether or not under the assumed conditions a solution exists for the addressed task, and if there is a solution, the planner provides one. If a solution does not exist then this solution would hardly appear because of the uncertainty. In contrast, if a solution does exist in the absence of uncertainty, it can be lost because of uncertainty. Thus, our planner can be considered as a stepping stone toward the solution of the harder problem of computing feasible paths under uncertainty. Consequently, as a future work, we would like to consider the problem of planning with uncertain maps of landmarks.

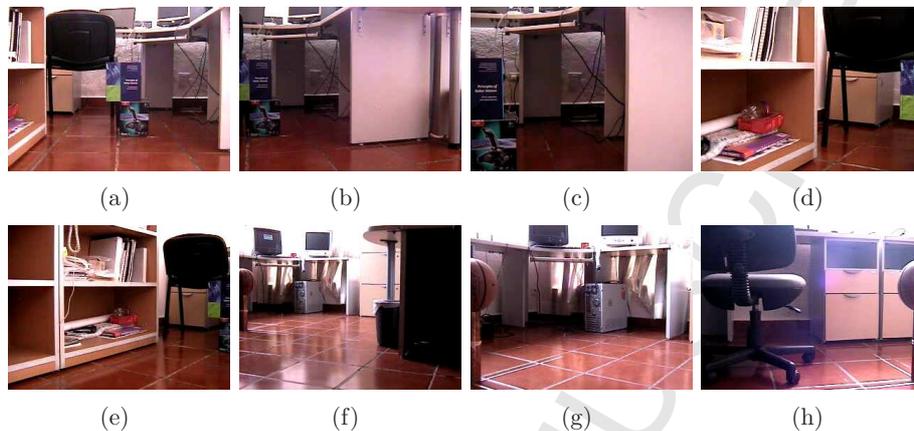


Figure 23: Two-landmark experiment. View from the robot. (a) Initial configuration. (b)-(d) Logarithmic spiral and rotation in place keeping the first landmark (the book) in sight. (e) Backward motion to reach switching point configuration. (f)-(g) Logarithmic spiral looking at the second landmark. (h) Final configuration.

References

- F. Avnaim and J. Boissonat. A practical exact motion planning algorithm for polygonal objects amidst polygonal obstacles. In *Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 1656–1661, 1988.
- D.J. Balkcom and M.T. Mason. Time optimal trajectories for bounded velocity differential drive vehicles. *Intl. J. of Robotics Research*, 21(3):199–217, 2002.
- S. Bhattacharya, R. Murrieta-Cid, and S. Hutchinson. Optimal paths for landmark-based navigation by differential-drive vehicles with field-of-view constraints. *IEEE Trans. on Robotics*, 23(1):47–59, 2007.
- A. Bicchi, G. Casalino, and C. Santilli. Planning shortest bounded-curvature paths for a class of nonholonomic vehicles among obstacles. *J. of Intelligent Robot Systems*, 16(4):387–405, 1996.
- A.J. Briggs, C. Detweiler, D. Scharstein, and A. Vandenberg-Rodes. Expected shortest paths for landmark-based robot navigation. *Intl. J. of Robotics Research*, 23(7–8):717–728, 2004.

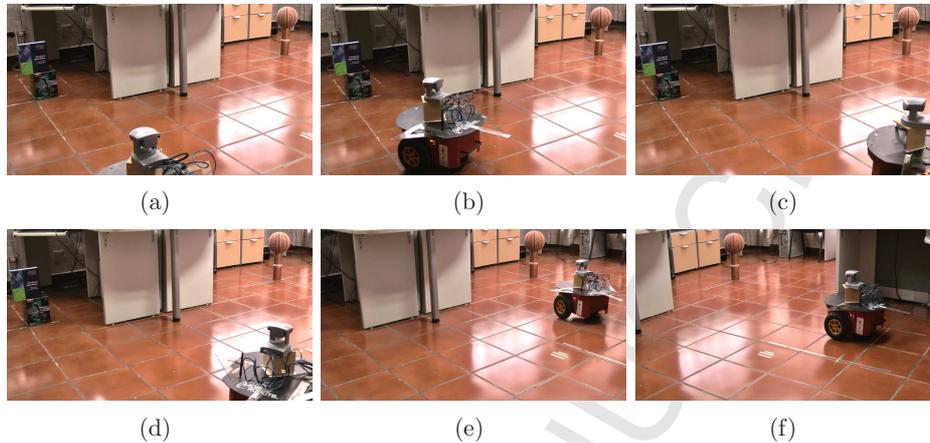


Figure 24: Two-landmark experiment, external camera view. (a) Initial configuration (b)-(c) The robot keeps the book in sight performing rotations in place, and a logarithmic spiral until it arrives to a point when it sees both landmarks and switches to approach the second one. (d)-(e) The robot follows the computed primitives to reach the final configuration while keeping the basketball in sight. (f) Final configuration.

H. Chitsaz, S.M. LaValle, D.J. Balkcom, and M.T. Mason. Minimum wheel-rotation paths for differential-drive mobile robots. *Intl. J. of Robotics Research*, 28:66–80, 2009.

H. Choset, K.M. Lynch, S. Hutchinson, G. Cantor, W. Burgard, L.E. Kavraki, and S. Thrun. *Principles of Robot Motion: Theory, Algorithms, and Implementations*. MIT Press, Boston, 2005.

A. Divelbiss and J. Wen. A path space approach to nonholonomic motion planning in the presence of obstacles. *IEEE Trans. on Robotics and Automation*, 13(3):443–451, 1997.

J.-B. Hayet and R. Murrieta-Cid. Shortest paths for differential drive robots under visibility and sensor constraints. Technical Report I-09-02/24-02-2009, CIMAT, 2009. <http://www.cimat.mx/~jbhayet/PUBLIS/Hayet-Robotica.pdf>.

J.-B. Hayet, F. Lerasle, and M. Devy. A visual landmark framework for mobile robot navigation. *Image and Vision Computing*, 8(25):1341–1351, 2007.

- J.-B. Hayet, C. Esteves, and R. Murrieta-Cid. A motion planner for maintaining landmark visibility with a differential drive robot. In G.S. Chirikjian et al., editor, *Proc. of the Intl. Workshop on the Algorithmic Foundations of Robotics (WAFR)*, volume 57, pages 333–347. Springer Tracts in Advanced Robotics, 2010.
- V. Isler, D. Sun, and S. Sastry. Roadmap based pursuit-evasion and collision avoidance. In *Proc. of Robotics Science and Systems (RSS)*, pages 257–264, 2005.
- M. I. Karavelas. A robust and efficient implementation for the segment voronoi diagram. In *Proc. Int. Symp. on Voronoi diagrams in Science and Engineering (VD2004)*, pages 51–62, 2004.
- J.-C. Latombe. *Robot motion planning*. Kluwer, 1991.
- J.-P. Laumond. *Robot Motion Planning and Control*. Springer, 1998.
- J.-P. Laumond, P.E. Jacobs, M. Taïx, and R.M. Murray. A motion planner for nonholonomic mobile robots. *IEEE Trans. on Robotics and Automation*, 10(5):577–593, 1994.
- S.M. LaValle. *Planning Algorithms*. Cambridge University Press, 2006.
- A. Lazanas and J.-C. Latombe. Landmark-based robot navigation. *Algorithmica*, 13:472–501, 1995.
- G. López-Nicolás, N.R. Gans, S. Bhattacharya, C. Sagües, J.J. Guerrero, and S. Hutchinson. Homography-based control scheme for mobile robots with nonholonomic and field-of-view constraints. *IEEE Transactions on Systems, Man, and Cybernetics: Part B*, 10(4):1115–1127, 2010.
- R. Madhavan and H.F. Durrant-Whyte. Natural landmark-based autonomous vehicle navigation. *Robotics and Autonomous Systems*, 46(2): 79–95, 2004.
- N. Mansard and F. Chaumette. Task sequencing for high level sensor-based control. *IEEE Trans. on Robotics*, 23(1):60–72, 2007.
- M. Michalek and K. Kozłowski. Feedback control framework for car-like robots using the unicycle controllers. *Robotica*, 30:517–535, 2011.

- P. Michel, C. Scheurer, J. Kuffner, N. Vahrenkamp, and R. Dillman. Planning for robust execution of humanoid motions using future perceptive capability. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 3223–3228, 2007.
- R.M. Murray and S.S. Sastry. Nonholonomic motion planning: Steering using sinusoids. *IEEE Trans. on Automatic Control*, 5(38):700–716, 1993.
- R. Murrieta-Cid, C. Parra, and M. Devy. Visual navigation in natural environments: From range and color data to a landmark-based model. *Autonomous Robots*, 13(2):143–168, 2002.
- P. Ranganathan, J.B. Hayet, M. Devy, S. Hutchinson, and F. Lerasle. Topological navigation and qualitative localization for indoor environment using multisensory perception. *Robotics and Autonomous Systems*, 41(3):137–144, 2002. ISSN 0921-8890.
- J.A. Reeds and L.A. Shepp. Optimal paths for a car that goes both forwards and backwards. *Pacific J. of Mathematics*, 145(2):367–393, 1990.
- N. Roy and S. Thrun. Coastal navigation with mobile robots. In *Advances in Neural Processing Systems*, volume 12, pages 1043–1049, 1999.
- P. Salaris, F. Belo, D. Fontanelli, L. Greco, and A. Bicchi. Optimal paths in a constrained image plane for purely image-based parking. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 1673–1680, 2008.
- P. Salaris, D. Fontanelli, L. Pallottino, and A. Bicchi. Shortest paths for a robot with nonholonomic and field-of-view constraints. *IEEE Trans. on Robotics*, 26:269–281, 2010.
- S. Se, D. Lowe, and J. Little. Vision-based global localization and mapping for mobile robots. *IEEE Trans. on Robotics*, 21(3):364–375, 2005.
- R. Sim and G. Dudek. Mobile robot localization from learned landmarks. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 1060–1065, 1998.
- R. Sim and G. Dudek. Comparing image-based localization methods. In *Proc. of the Intl. Joint Conf. on Artificial Intelligence (IJCAI)*, pages 1560–1562, 2003.

P. Souères and J.-P. Laumond. Shortest paths synthesis for a car-like robot. *IEEE Trans. on Automatic Control*, 41(5):672–688, 1996.

S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. MIT Press, 2005.

Appendix A. Superior bounds on the paths lengths

Case of a line segment and a spiral (L-S). Without loss of generality, we consider here the case of a line followed by a S_+ spiral, the $L-S_-$ case being symmetric. In that case (Figure 9, left), the distance between P_s and P_e is given by the sum of the line part and the spiral part, i.e.

$$d_B(P_s, P_e) = \|P_s^g M\| + l_{\phi_+}(M, P_e^g).$$

One can refer to Hayet and Murrieta-Cid [2009] to make this distance more explicit, in terms of the angle α_M depicted in Figure A.25, which corresponds to the relative position of the transition point M between the line and the spiral, and which is located on an arc of circle $L_i P_s^g$ also depicted in Figure A.25. As M is the transition between line and spiral (i.e. the point at which saturation of ϕ occurs), one can get

$$\|P_s^g M\| = r_{P_s} \frac{\sin \alpha_M}{\sin \phi_+},$$

and its distance to the landmark, r_M , satisfies

$$r_M = r_{P_s} \frac{\sin(\phi_+ - \alpha_M)}{\sin \phi_+},$$

so that, by using Equation 14

$$d_B(P_s, P_e) = r_{P_s} \frac{\sin \alpha_M}{\sin \phi_+} + \frac{1}{\cos \phi_+} r_{P_s} \frac{\sin(\phi_+ - \alpha_M)}{\sin \phi_+} (1 - e^{\frac{\alpha_M - \alpha_{P_e} + \alpha_{P_s}}{\tan \phi_+}}).$$

This expression can be simplified when considering that P_e^g is close enough to P_s^g . In that case, one will notice that α_M is small too (first order), so that the exponential can be developed at the first order by

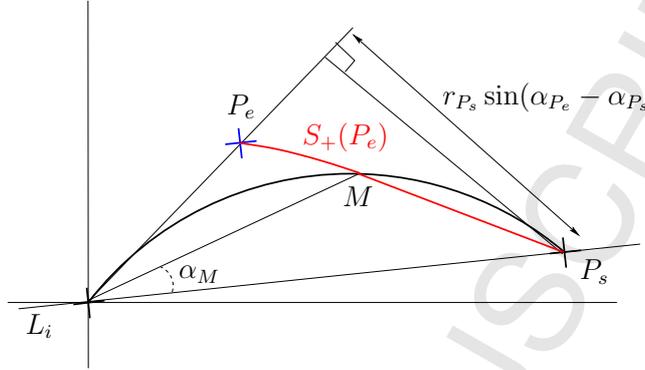


Figure A.25: Line-spiral path from P_s to P_e . M is the point in which the line ends, and is located on an arc of circle passing through L_i and P_s .

$$1 - e^{\frac{\alpha_M - \alpha_{P_e} + \alpha_{P_s}}{\tan \phi_+}} \approx -\frac{\alpha_M - \alpha_{P_e} + \alpha_{P_s}}{\tan \phi_+},$$

which allows to write, for P_e^g in a close neighborhood of P_s^g

$$d_B(P_s, P_e) \approx r_{P_s} \frac{\sin \alpha_M}{\sin \phi_+} - r_{P_s} \frac{\sin(\phi_+ - \alpha_M)}{\sin^2 \phi_+} (\alpha_M - \alpha_{P_e} + \alpha_{P_s}),$$

and by keeping only the first order terms, and by using $\sin \alpha_M \approx \alpha_M$, and $\sin(\phi_+ - \alpha_M) \approx \sin \phi_+$, since the reasoning is on a small neighborhood of P_s^g , hence $\alpha_M \ll 1$,

$$d_B(P_s, P_e) \approx r_{P_s} \frac{\alpha_M}{\sin \phi_+} - r_{P_s} \frac{\alpha_M}{\sin \phi_+} + \frac{r_{P_s}}{\sin \phi_+} (\alpha_{P_e} - \alpha_{P_s}),$$

and finally $d_B(P_s, P_e) \approx \frac{r_{P_s}}{\sin \phi_+} |\alpha_{P_e} - \alpha_{P_s}|$. Since $(\alpha_{P_e} - \alpha_{P_s}) \approx \sin(\alpha_{P_e} - \alpha_{P_s})$ and by examining Figure A.25, one sees that for P_e^g close to P_s^g ,

$$r_{P_s} |\alpha_{P_e} - \alpha_{P_s}| \leq \|P_s^g P_e^g\|.$$

It follows that for $K > 1$, one can set a neighborhood of P_e^g where,

$$d_B(P_s, P_e) \leq \frac{K}{\sin \phi_+} \|P_s^g P_e^g\|.$$

*Case of two spirals (S^*S).* Here we consider the case of a S_+ spiral followed by a S_- spiral (Figure 9, right). By using the Equation 14 twice on the two spirals, and with $t_+ = \tan \phi_+$, $t_- = \tan \phi_- = -t_+$,

$$\begin{aligned} d_B(P_s, P_e) &= l_{\phi_+}(P_s, Q) + l_{\phi_-}(Q, P_e) \\ &= \frac{r_{P_s}}{\cos \phi_+} (1 - e^{\frac{\alpha_{P_s} - \alpha_Q}{t_+}}) + \frac{r_{P_e}}{\cos \phi_-} (1 - e^{\frac{\alpha_{P_e} - \alpha_Q}{t_-}}) \\ &= \frac{r_{P_s}}{\cos \phi_+} (1 - e^{\frac{\alpha_{P_s} - \alpha_Q}{t_+}}) + \frac{r_{P_e}}{\cos \phi_+} (1 - e^{\frac{\alpha_Q - \alpha_{P_e}}{t_+}}). \end{aligned}$$

The intersection point Q between spirals can be shown Hayet and Murrieta-Cid [2009] to be

$$\alpha_Q = -\frac{t_+}{2} \log \frac{r_{P_s}}{r_{P_e}} + \frac{1}{2}(\alpha_{P_s} + \alpha_{P_e}),$$

which is plugged into the previous equation to give,

$$d_B(P_s, P_e) = \frac{1}{\cos \phi_+} [r_{P_s} + r_{P_e} - 2e^{-\frac{\alpha_{P_e} - \alpha_{P_s}}{2t_+}} \sqrt{r_{P_s} r_{P_e}}].$$

If P_e^g is sufficiently close to P_s^g , we can use a Taylor expansion around P_s^g , at the first order in $\delta r = r_{P_e} - r_{P_s}$ and $\delta \alpha = \alpha_{P_e} - \alpha_{P_s}$

$$d_B(P_s, P_e) \approx \frac{1}{\cos \phi_+} [2r_{P_s} + \delta r - 2(1 - \frac{\delta \alpha}{t_+}) r_{P_e} \sqrt{1 + \frac{\delta r}{r_{P_e}}}],$$

and, as $\sqrt{1 + \frac{\delta r}{r_{P_e}}} \approx 1 + \frac{1}{2} \frac{\delta r}{r_{P_e}}$,

$$d_B(P_s, P_e) \approx \frac{1}{\cos \phi_+} [2r_{P_s} + \delta r - 2(1 - \frac{\delta \alpha}{t_+}) r_{P_e} (1 + \frac{1}{2} \frac{\delta r}{r_{P_e}})],$$

in which the δr terms cancel out to give $d_B(P_s, P_e) \approx \frac{1}{\sin \phi_+} r_{P_s} \delta \alpha$. By using 15, we finally get the same inequality as in the $L-S$ case, i.e., for $K > 1$, one can define a neighborhood of P_e^g such that

$$d_B(P_s, P_e) \leq \frac{K}{\sin \phi_+} \|P_s^g P_e^g\|.$$