



HAL
open science

A General Framework for Planning Landmark-Based Motions for Mobile Robots

Abed Malti, Florent Lamiroux, Michel Taïx

► **To cite this version:**

Abed Malti, Florent Lamiroux, Michel Taïx. A General Framework for Planning Landmark-Based Motions for Mobile Robots. *Advanced Robotics*, 2011, 25 (11-12), pp.1427-1450
10.1163/016918611X579457 . hal-01537687

HAL Id: hal-01537687

<https://hal.science/hal-01537687>

Submitted on 12 Jun 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A General Framework for Planning Landmark-Based Motions for Mobile Robots

Abed Malti⁽¹⁾, Florent Lamiroux⁽²⁾, and Michel Taïx⁽²⁾

(1) is with Institute of Systems and Robotics, Faculty of Science and Technology, University of Coimbra, POLO II, 3030-290 Coimbra, Portugal. (2)s are with CNRS; LAAS, University of Toulouse; UPS, INSA, INP, ISAE. 7, av. du Colonel Roche, 31077 Toulouse, France.
amalti@isr.uc.pt, florent@laas.fr, taix@laas.fr

Abstract

Our work is focused on defining a generic approach for planning landmark based motion. The proposed geometric approach deals with robots of complex kinematic moving in cluttered environment. It aims at providing safe strategy of motion for such constrained conditions. The proposed method selects automatically the most relevant landmarks along a pre-planned geometric path. It proposes a strategy to correct the trajectory and to smoothly switch among the landmarks of the environment. Experimental results highlight the relevance of the proposed formalism.

keywords: Motion Planning, Sensor-Based Motion, Landmark-Based Motion, Mobile robot

1 Introduction

The path planning problem has raised a lot of interest in the robotics community for the last two decades ([11, 4, 13]). In the case of mobile robots, it is particularly challenging when the kinematic involves nonholonomic constraints. [12] proposes an overview of the problem of path planning for mobile robots. Today, we can reasonably consider that the robotic research community has produced a sufficient set of techniques and algorithms to efficiently solve the problem of mobile robot path planning in a known environment.

However, solving the path planning problem is not enough to efficiently perform a navigation task, mainly for three reasons. First, the inaccuracy of the map of the environment used to plan the geometric path. Second, precise localization is needed in the case of complex kinematic robots moving in cluttered environment. Third, during navigation, abrupt switching among landmarks can lead to strong jumps in the localization. For small size mobile robots, these perturbations can be overcome by increasing its size during the path planning step. To take into local inaccuracies of the map, local obstacle avoidance techniques can be used [18, 9, 2, 16]. Built upon these conceptual tools, several experimental mobile robots navigate in indoor environments on a daily basis [24, 1, 7].



Figure 1: Mobile robot Hilare 2 towing a trailer.

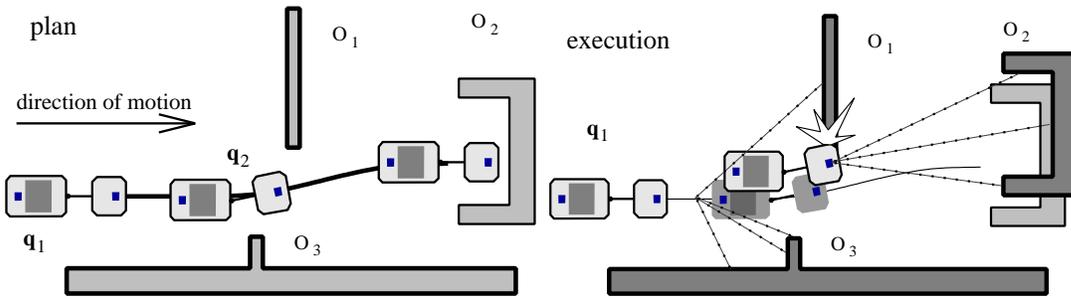


Figure 2: Mobile robot Hilare 2 towing a trailer is equipped with 2 laser range scanners: one on the robot and one on the trailer. The robot performs global localization using a polygonal 2D map of the environment. In this example, the robot is required to dock close to obstacle O_2 . The path planned corresponds to a backward motion (left). Light dotted lines represent the features used by the robot to compute global localization (right). If the position of obstacle O_2 is inaccurate in the map, using this landmark for global localization when passing close to obstacle O_1 may lead to collision.

Recent works consider either constrained environment [25] or highly constrained degree of freedom robots [19, 3]. However, for nonholonomic mobile robots navigating close to obstacles as for instance trucks towing trailers and performing docking tasks, the problem is more intricate as we are going to explain. Let us consider for instance the nonholonomic mobile robot Hilare 2 towing a trailer displayed in Figure 1. Figure 2 (top) represents a path planned for this robot in order to perform a docking task. The robot and the trailer are both equipped with a laser range scanner. Along the motion, the robot localizes itself using straight line segments stored in a map built beforehand. In this example, the environment is composed of three polygonal obstacles denoted by O_1 , O_2 and O_3 . Let us moreover assume that the position of obstacle O_2 in the map is different from the real position in the environment (Figure 2 bottom). Then if the robot localizes itself on obstacle O_2 , the estimation of the robot position will be subject to the same error as the position of obstacle O_2 . Therefore if this error is bigger than the distance between the planned path and O_1 and if the robot localizes itself too early on O_2 a collision

will happen. Of course, this reasoning is valid as well for mobile robots without kinematic constraints. However, the problem is more critical with multi-body nonholonomic mobile robots. Indeed, when the localization process switches from obstacles O_1 and O_3 to obstacle O_2 , a discontinuity appears in the global localization of the robot. Suddenly, the robot realizes that it is not anymore on the planned trajectory. The motion control law then corrects this gap but nonholonomic systems require time to correct lateral motion tracking errors. Moreover these corrections induce angular deviation of the system.

To summarize the above reasoning, navigating close to obstacles with a multi-body nonholonomic mobile robot in an inaccurate map induces localization discontinuities that make the robot leaving the reference planned trajectory. Even if the reference trajectory is checked for collision and dynamically adapted to remain collision-free as explained in [10], the localization of the robot and the tracking of the planned path are prone to fail after several trajectory deformations. Recent works use Belief Roadmap algorithm to incorporated localization problem in motion planning [6].

Through the novel definition of a landmark based motion, the contribution of this paper is threefold. First, we propose a framework to cope with the localization problem exposed above. The idea consists in defining during the motion planning step the landmarks on which the robot will localize in such a way that obstacles likely to imply collisions are taken into account at the right time and that localization discontinuities are reduced. Second, we propose a landmark-based motion planning algorithm that selects landmarks and smoothly switch among them to safely execute a geometric path. Third, we validate our approach on an experimental system in real conditions.

The paper is organized as follows. In Section 2, we draw the theoretical foundations of our framework and we give a mathematical definition of a landmark-based motion. In Section 3, we apply our framework to the mobile robot Hilare 2 towing a trailer and equipped with laser range scanners. We propose a landmark-based motion planning algorithm in an environment composed of vertical planes. In Section 4, we provide experimental results that show the relevance of the approach and we draw some conclusions in Section 5.

2 Landmark-based Motion

2.1 Definitions

In this work, a Landmark-Based Motion is a task, as defined in the task function approach [20], [5]. In this section, we perform the computations leading to the task definition.

2.1.1 Landmark

A landmark can be any geometric feature in the workspace. Let us denote by \mathbb{L} the configuration space of a landmark L . For instance, if L is a point in space, $\mathbb{L} = \mathbf{R}^3$, if L is a line segment in the plane, $\mathbb{L} = \mathbf{R}^4$. We will denote by l the configuration of L .

2.1.2 Sensor

A sensor S is a mobile device that maps one or several landmarks to a feature in the image space. Let us denote by $\mathbb{S} = SE(2)$ or $SE(3)$ the configuration space of sensor S . We will denote by s the configuration of S .

2.1.3 Sensing a landmark

The perception of landmark L by sensor S is characterized by a mapping:

$$\begin{aligned} P_{S,L}: \mathbb{S} \times \mathbb{L} &\rightarrow I_{S,L} \\ (s, l) &\rightarrow P_{S,L}(s, l) \end{aligned}$$

that maps to each configuration of the sensor and of the landmark a feature in the image space $I_{S,L}$. For a laser range finder sensing a point for instance, the image space is a half plane. Let us notice that P is usually defined over a subset of $\mathbb{S} \times \mathbb{L}$ corresponding to pairs of configurations of the sensor and landmark for which the landmark is in the field of view of the sensor.

2.2 Localization

2.2.1 Localization equation

Let us consider a mobile robot equipped with n sensors S_1, \dots, S_n . Let us denote by \mathcal{C} the configuration space of the robot. The configuration s_i of each sensor S_i , $i = 1, \dots, n$ is uniquely defined by the configuration \mathbf{q} of the robot. Let us consider m landmarks L_1, \dots, L_m of known configurations l_1, \dots, l_m each of them visible by sensor S_{i_j} where i_1, \dots, i_m is a sequence of indices with values in $\{1, \dots, n\}$, defining which sensor sees each landmark (a landmark j can be seen by the sensor defined by i_j).

Therefore, each pair (S_{i_j}, L_j) of sensor landmark gives rise to a *localization equation system*:

$$P_{S_{i_j}, L_j}(s_{i_j}(\mathbf{q}), l_j) = im_j \quad (1)$$

where $im_j \in I_{S_{i_j}, L_j}$ is the image of L_j in S_{i_j} . l_j is supposed to be known and im_j is measured. The unknown of this equation is the configuration \mathbf{q} of the robot.

2.2.2 Localization about a reference configuration

If the robot is expected to follow a reference trajectory, or if the localization is performed with a high frequency, the result of equations (1) is expected to be in the neighborhood of a reference configuration that can be the latest result of localization or a configuration along a planned trajectory. Let us denote by \mathbf{q}_0 this reference configuration and let us linearize each localization equation (1) about \mathbf{q}_0 . We get the following linear equations:

$$\frac{\partial P_{S_{i_j}, L_j}}{\partial s}(s_{i_j}(\mathbf{q}_0), l_j) \frac{\partial s_{i_j}}{\partial \mathbf{q}}(\mathbf{q}_0)(\mathbf{q} - \mathbf{q}_0) = im_j - im_{0j} \quad (2)$$

This equation expresses the approximation of order 1 of the relation between a variation of configuration about \mathbf{q}_0 and the variation of the image of each landmark in the corresponding sensor.

$\frac{\partial P_{S_{i_j}, L_j}}{\partial \mathbf{s}}(s_{i_j}(\mathbf{q}_0), l_j)$ is a Jacobian matrix representing the variation of the image with respect to the variation of the sensor configuration. This Jacobian matrix derives from the geometric properties of the sensor and landmark (the interaction matrix).

$\frac{\partial s_{i_j}}{\partial \mathbf{q}}$ represents the variation of the sensor configuration with respect to the variation of the robot configuration. This Jacobian matrix derives from the kinematics of the robot.

im_j and im_{0_j} respectively represent the image of landmark L_j in sensor S_{i_j} and the expected image, *i.e.* the image that would be seen from configuration \mathbf{q}_0 if the map was exact.

It is similar to visual servoing approach where the variation of visual cues is $\dot{s} = LJ\dot{q}$. L is the interaction matrix and J the jacobian matrix of the robot. But in this equation the relation is defined in local sensor frame while in equation (2) it is defines in a global frame. The visual control approach define and solve local task, here the aim is to solve the motion planning in global frame.

2.2.3 Weighting localization

If the number of landmarks is not sufficient, system of equations (2) may admit an infinite set of solutions, localization is said *under-determined*. If the number of landmarks is too big, system (2) has no exact solution and the localization is said *over-constrained*. In this case, the most common localization approach consists in modelling landmark and sensor configurations as Gaussian random variables and in determining the configuration of the robot that maximizes the likelihood of the measures. Each landmark is weighed with respect to the variance of its position.

In our framework, the weight associated to each landmark is part of the motion control task specification. Thus, from linear equations (2) we build a linear system of equations by weighting each equation by a positive real number λ_j . We will explain later the role of these coefficients. We thus get the following linear system:

$$W(\mathbf{q} - \mathbf{q}_0) = IM - IM_0 \quad (3)$$

where W is matrix

$$\begin{pmatrix} \lambda_1 \frac{\partial P_{S_{i_1}, L_1}}{\partial \mathbf{s}}(s_{i_1}(\mathbf{q}_0), l_1) \frac{\partial s_{i_1}}{\partial \mathbf{q}}(\mathbf{q}_0) \\ \vdots \\ \lambda_m \frac{\partial P_{S_{i_m}, L_m}}{\partial \mathbf{s}}(s_{i_m}(\mathbf{q}_0), l_m) \frac{\partial s_{i_m}}{\partial \mathbf{q}}(\mathbf{q}_0) \end{pmatrix}$$

and

$$IM = \begin{pmatrix} \lambda_1 im_1 \\ \vdots \\ \lambda_m im_m \end{pmatrix} \quad IM_0 = \begin{pmatrix} \lambda_1 im_{0_1} \\ \vdots \\ \lambda_m im_{0_m} \end{pmatrix}$$

The least square solution of this system:

$$\mathbf{q} = \mathbf{q}_0 + W^+(IM - IM_0) \quad (4)$$

where W^+ is the pseudo-inverse of W , minimizes (up to approximation of order 1) the weighed sum of the squares of the Euclidean norms of the differences between each image that would be seen from \mathbf{q} and the image really seen im_j :

$$\sum_{j=1}^m \lambda_j^2 \|P_{S_{i_j}, L_j}(\mathbf{q}) - im_j\|^2$$

The developments conducted in this section can be summarized as follows. Localizing a mobile robot on landmarks consists in solving a system of equations that relate the configuration of the robot with the images of the landmarks in the sensors of the robot. If the system is over-constrained, localization consists in finding a configuration that minimizes a weighed sum of residues. If the landmarks are exactly at the same position in the map as in reality, the choice of weights will have no effect on the result (up to approximation of order 1). If the map of landmarks is not exact however, the choice of weights will have a big influence.

That is why in our approach, we suggest to use these weights as a tool for planning landmark-based motions.

2.3 Landmark-Based Motion

Based on the development of the former section, we define a landmark-based motion for a mobile robot as follows :

Given a mobile robot with n sensors S_1, \dots, S_n and an environment with m landmarks L_1, \dots, L_m , respectively associated to sensors S_{i_1}, \dots, S_{i_m} , a *landmark-based motion* is defined by:

1. a reference collision-free trajectory:

$$\begin{aligned} \gamma : [0, U] &\rightarrow \mathcal{C} \\ u &\rightarrow \gamma(u) \end{aligned}$$

where $[0, U]$ is an interval,

2. m continuous positive real valued functions $\lambda_1, \dots, \lambda_m$:

$$\begin{aligned} \lambda_j : [0, U] &\rightarrow \mathbf{R}^+ \\ u &\rightarrow \lambda_j(u) \end{aligned}$$

such that $\lambda_j(u) = 0$ for any u such that L_j is not visible by sensor S_{i_j} when the robot is in configuration $\gamma(u)$.

Following the above landmark-based motion for the mobile robot consists in estimating the current configuration of the robot within the closed-loop control task by solving system (3) about the reference configuration $\mathbf{q}_0 = \gamma(u)$ and with values of the weights $\lambda_j(u)$ for abscissa u along the trajectory

2.4 Landmark-Based Motion Planning

At a first step, a geometric non-collision path $\gamma(u)$ is planned in the configuration space of the model map from an initial configuration to a final one. In our work, we consider this assessment as being produced by any geometric path planner.

The second step consists in calculating the weights of the sensor-landmark pairs along the path in the model map. For a given type of sensor and a given type of landmark, the weight is defined as a continuous function in the space $\mathbb{S} \times \mathbb{L}$, where for any configuration of the sensor and any configuration of the landmark it associates a positive value. Afterwards, based on this generic function, a specific weight function is constructed for the most relevant landmarks viewed by the given sensor along the geometric pre-planned path.

For example, to avoid collision in the situation described in figure 2 it is necessary to take into account only O_1 in a first time and then introduce obstacle O_2 into the localization process. The transition enters O_1 and O_2 must be continuous to avoid discontinuity in localization (see figure 3).

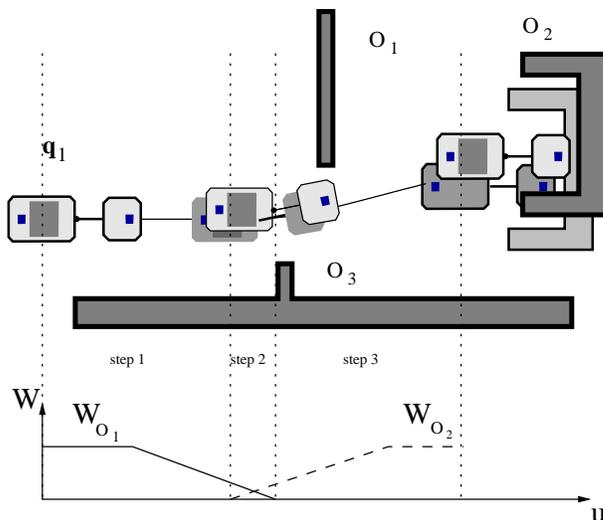


Figure 3: In figure 2 the task is to dock the robot close obstacle O_2 . To avoid collision sensor-landmark pair are weights along the free initial path. The general shape of weights along path is illustrated at the bottom of the figure. O_1 has a lot of influence at the beginning (where it is important) and O_2 is taken into account gradually to replace O_1 at the end of the path.

Henceforth, planning Landmark-Based Motion consist in building λ_j functions along the free collision path solution (a detailed implementation is given in 3.3).

2.5 Landmark-Based Motion Execution

2.5.1 Local reference configuration

The goal of the motion control is to modify the sensor-landmark perception $IM(\mathbf{q})$ by moving the robot to a reference localization configuration \mathbf{q}_{loc} which satisfy:

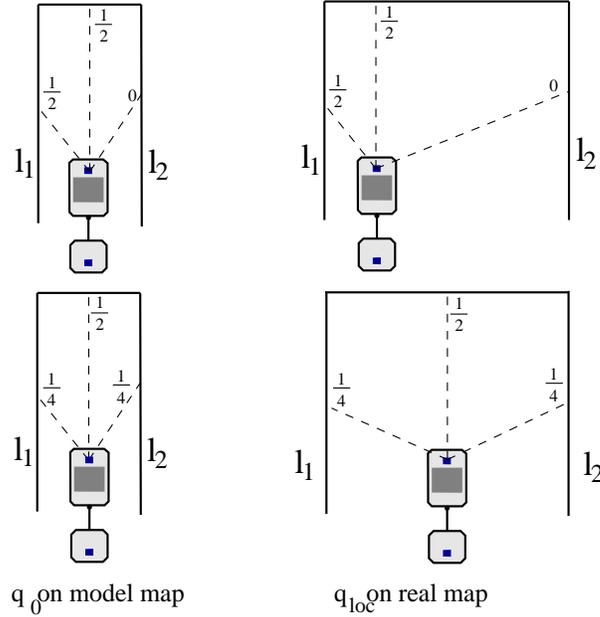


Figure 4: Local reference configuration

$$W^+(IM(\mathbf{q}_{loc}) - IM(\mathbf{q}_0)) = 0 \quad (5)$$

The figure 4 illustrates the idea of the \mathbf{q}_{loc} . You can see in left the reference configurations in map model and the corresponding \mathbf{q}_{loc} in real map (see difference between real map and model map). In bottom figures the landmark-based motion planning take into account the two wall with the same weight and the closed loop control of the robot has to converge on \mathbf{q}_{loc} to keep the robot at the middle of the walls. In top, only the left wall is take into account and the \mathbf{q}_{loc} is define to keep the same distance.

2.5.2 Closed loop motion control

It is important to note that we do not compute control law to the robot, we assume than the control law of the robot can stabilize the robot around a configuration and the landmark based motion only give a \mathbf{q}_{loc} to the feedback function which take into account the different weights of the sensor-landmark pairs. The general landmark based closed loop is illustrated in the figure 5.

Notice that $\mathbf{q}_{loc} - \mathbf{q}_0$ is the task function $e(\mathbf{q}, t)$ as define in [20], that we stabilize to 0.

3 Application to mobile robot Hilare 2 towing a trailer

In this section, we illustrate the above definition on the mobile robot Hilare 2 towing a trailer, equipped with two laser range scanners S_1 and S_2 , one on the mobile robot and one on the trailer (Figure 1). The sensors move in an horizontal plane: $\mathbb{S}_1 = \mathbb{S}_2 = SE(2)$. Figure 6 describes the geometry of the mobile robot. Let us denote by (O, x, y) a fixed reference frame in the space. The configurations of the sensors

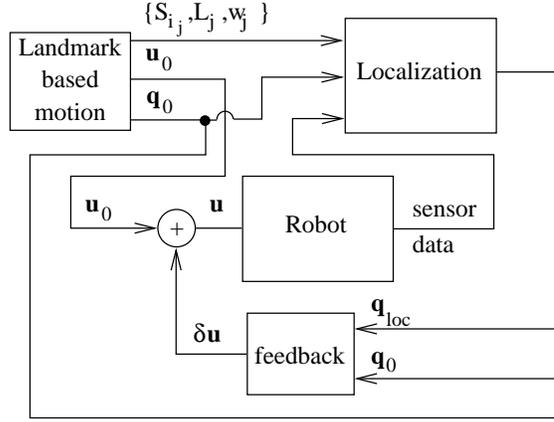


Figure 5: Landmark-based closed loop feedback motion control. The robot takes as input a control vector \mathbf{u} and returns as outputs sensor data (*i.e.* images of landmarks in sensors). These images are input to the localization function that compares them with expected images and computes the estimated configuration \mathbf{q}_{loc} of the robot as a deviation from the reference configuration \mathbf{q}_0 , using preselected weighed landmarks $\{S_{i_j}, L_j, \lambda_j\}$. This estimation of the configuration of the robot is input to the feedback function that compares it with the reference configuration and returns as output an input correction $\delta\mathbf{u}$. Note that the feedback function is not specific to our approach: any closed loop feedback controller stabilizing the system can be used.

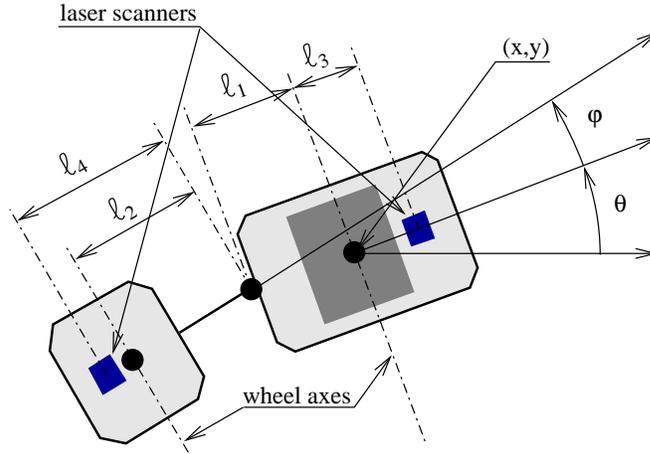


Figure 6: Mobile robot Hilare 2 towing a trailer equipped with two laser range scanners. Configuration of mobile robot is parameterized by (x, y, θ, φ) , where (x, y) is the center and θ the orientation of the robot, φ is the angle of the trailer w.r.t. the robot. l_1 is the distance between the wheel axis of the robot and the trailer connection, l_2 is the distance between the wheel axis of the trailer and the trailer connection. Configuration of sensor S_1 , mounted on the robot, is denoted by $s_1 = (x + l_3 \cos \theta, y + l_3 \sin \theta, \theta)$. Configuration of sensor S_2 mounted on the trailer is denoted by $s_2 = (x - l_1 \cos \theta - l_4 \cos(\theta + \varphi), y - l_1 \sin \theta - l_4 \sin(\theta + \varphi), \theta + \varphi + \pi)$. l_3 and l_4 are constant lengths.

w.r.t. the configuration $\mathbf{q} = (x, y, \theta, \varphi)$ of the robot are given by:

$$s_1(\mathbf{q}) = \begin{pmatrix} x + l_3 \cos \theta \\ y + l_3 \sin \theta \\ \theta \end{pmatrix} \quad (6)$$

$$s_2(\mathbf{q}) = \begin{pmatrix} x - l_1 \cos \theta - l_4 \cos(\theta + \varphi) \\ y - l_1 \sin \theta - l_4 \sin(\theta + \varphi) \\ \theta + \varphi + \pi \end{pmatrix} \quad (7)$$

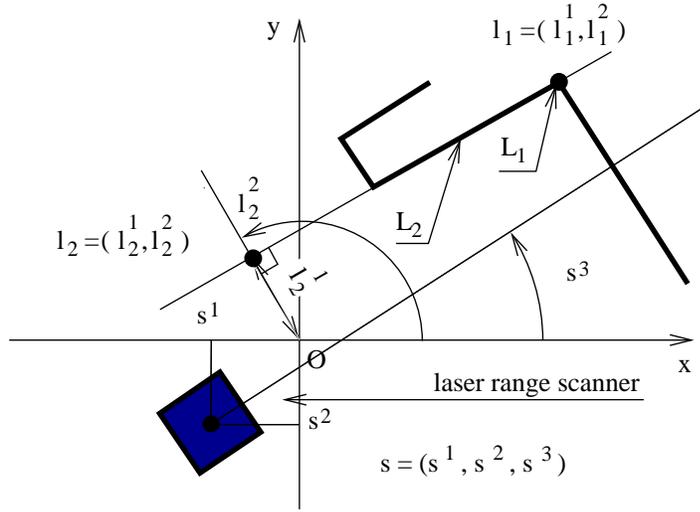


Figure 7: A laser range scanner in configuration $s = (s^1, s^2, s^3)$ sensing two landmarks, (s^1, s^2) is the position of the center, s^3 is the orientation of the sensor in the plane (O, x, y) . L_1 is a vertical straight line represented by its projection $l_1 = (l_1^1, l_1^2)$ on (O, x, y) . L_2 is a vertical plane represented by $l_2 = (l_2^1, l_2^2)$, the polar coordinates in plane (O, x, y) of the projection of the origin O on the plane.

3.1 Landmarks

Let us consider two types of landmarks: vertical planes (walls) and vertical lines (see figure 7). The vertical lines are defined as the intersection between two walls. These landmarks are projected in the image space of a laser scanner respectively as straight lines and points.

3.1.1 Vertical straight line

A vertical straight line L_1 is represented by its projection $l_1 = (l_1^1, l_1^2)$ on (O, x, y) . The image of a straight line in a laser sensor in configuration $s \in \mathbb{S} = SE(2)$ is the image of l_1 by s^{-1} :

$$P_{S, L_1}(s, l_1) = \begin{pmatrix} \cos s^3 (l_1^1 - s^1) + \sin s^3 (l_1^2 - s^2) \\ -\sin s^3 (l_1^1 - s^1) + \cos s^3 (l_1^2 - s^2) \end{pmatrix}$$

3.1.2 Vertical plane

We represent a vertical plane L_2 by the polar coordinates $l_2 = (l_2^1, l_2^2)$ of the projection of the origin O on the plane. The equation of L_2 in space is thus:

$$(x, y, z) \in L_2 \Leftrightarrow x \cos(l_2^2) + y \sin(l_2^2) = l_2^1$$

We use polar coordinates to avoid the singularity arising when O belongs to the vertical plane.

The image of a vertical plane in a laser range scanner is a straight line in \mathbf{R}^2 defined by the polar coordinates $im_2 = (im_2^1, im_2^2)$ of the projection of the origin of the sensor on the straight line. $s = (s^1, s^2, s^3)$ denote respectively the position and orientation of the laser scanner. Equivalently, s represents a rigid-body motion; the composition of a rotation of angle s^3 with a translation of vector (s^1, s^2) . The image of L_2 in S is the image of the intersection of the plane with the horizontal plane through s^{-1} . After computations,

$$P_{S,L_2}(s, l_2) = \begin{pmatrix} l_2^1 - s^1 \cos l_2^2 - s^2 \sin l_2^2 \\ l_2^2 - s^3 \end{pmatrix}$$

Let us notice the above expression is not exactly the polar coordinates of the projection of the origin of the sensor onto the straight line since $l_2^1 - s^1 \cos l_2^2 - s^2 \sin l_2^2$ may be negative. This does not pose any problem.

3.2 Localization Equations

Each pair sensor-landmark gives rise to a system of linearized equations of the form (2). From the above expressions, we can compute the different matrices involved in (2). For a vertical straight line L_1 sensed by an laser scanner S ,

$$\frac{\partial P_{S,L_1}}{\partial s} = \begin{pmatrix} -\cos s^3 & -\sin s^3 & -\sin s^3(l_1^1 - s^1) + \cos s^3(l_1^2 - s^2) \\ \sin s^3 & -\cos s^3 & -\cos s^3(l_1^1 - s^1) - \sin s^3(l_1^2 - s^2) \end{pmatrix}$$

For a vertical plane,

$$\frac{\partial P_{S,L_2}}{\partial s} = \begin{pmatrix} -\cos l_2^2 & -\sin l_2^2 & 0 \\ 0 & 0 & -1 \end{pmatrix}$$

For both sensors, $\frac{\partial s}{\partial \mathbf{q}}$ derives simply from expressions (6) and (7):

$$\frac{\partial s_1}{\partial \mathbf{q}}(\mathbf{q}) = \begin{pmatrix} 1 & 0 & -\ell_3 \sin \theta & 0 \\ 0 & 1 & \ell_3 \cos \theta & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

$$\frac{\partial s_2}{\partial \mathbf{q}}(\mathbf{q}) = \begin{pmatrix} 1 & 0 & \ell_1 \sin \theta + \ell_4 \sin(\theta + \varphi) & \ell_4 \sin(\theta + \varphi) \\ 0 & 1 & -\ell_1 \cos \theta - \ell_4 \cos(\theta + \varphi) & -\ell_4 \cos(\theta + \varphi) \\ 0 & 0 & 1 & 1 \end{pmatrix}$$

3.3 Landmark-based motion

To be able to compute a landmark based motion, it is necessary to fulfill the following steps :

- Motion planning: define a free-collision path.
- Weight computation: define the weight function for all sensor-landmark pair.
- Landmark-based motion planning: compute the sensor-landmark pairs with their weights across the free-collision path.

Several algorithms and strategies can be developed to follow our generic definition of a landmark-based motion. The following development describes a strategy that is experimentally validated on our mobile robot Hilare 2 towing a trailer.

3.3.1 Motion planning

The aim is to compute a free collision path for the robot with any path planning method. We choose to use Probabilistic Roadmap algorithms with the **Move3D** platform developed in LAAS-CNRS [22, 21] because of its genericity with regard to the robot kinematic (it can be used for robots with high degree of freedom). Given a kinematic model of the robot and a geometric model of the environment, the software provides a geometric free-collision path from an initial configuration to a goal configuration.

3.3.2 Weight computation

Usually, the stochastic localization methods assuming gaussian distribution are equivalent to a weighted least square estimator, where the weighting matrix is the inverse of the variance-covariance matrix associated to the state variables of the model [8, 17, 14]. Hence, the weight function encodes the noise of the model and the data measures. In our work, the weight is an intrinsic specification of the robotic task associated to the execution of the geometric planned path. For a given configuration along this path, the weight of any sensor-landmark pair encodes the following :

- The quality of the visibility of the landmark by the sensor.
- The potential danger of collision with the landmark.
- According to the path, when the landmark starts and ends to be taken into account.

A weight of a sensor-landmark pair is defined as a positive continuous function in the configuration space by :

$$\begin{aligned} w : \mathbb{S} \times \mathbb{L} &\rightarrow \mathbf{R}^+ \\ (s, l) &\mapsto w(s, l) \end{aligned}$$

This definition explicitly takes into account the two first previous statements. The third one shows up in the planning step when going through the path the landmarks are smoothly selected and unselected. This is the purpose of the next paragraph.

3.3.3 Planning a landmark based motion

In this work, we propose a strategy of sensor-landmark selection based on the k best ones. The algorithm has as inputs the following :

1. The reference map of the environment,
2. The set of sensors \mathcal{S} and landmarks \mathcal{L}_{env} ,
3. The non-collision geometric path $\gamma(u)$, $u \in [0, U]$,
4. The number k of maximum best landmarks.

The output is a landmark based motion LBM composed of $\gamma(u)$ and a set of weighed sensor-landmark pairs. The pseudo-algorithm for a robot equipped with one sensor is depicted as follows :

Algorithm 1 Construction of LBM

INPUT: $Map, \mathcal{L}_{env}, \mathcal{S}, k$

OUTPUT: LBM

- 1: $\gamma(u)$, $u \in [0, U] \leftarrow$ Free-collision path(q_i, q_f)
 - 2: $u_1 \leftarrow 0$
 - 3: **while** $u_1 < U$ **do**
 - 4: $\mathcal{L}_{select} \leftarrow$ get k best landmarks at u_1
 - 5: $\delta u_{max} \leftarrow$ increase u until \mathcal{L}_{select} change
 - 6: $u_1 \leftarrow u_1 + \delta u_{max}$
 - 7: **for** $L \in \mathcal{L}_{select}$ **do**
 - 8: $LBM \leftarrow$ add weighted λ for (S, L)
 - 9: **end for**
 - 10: **end while**
-

Given a configuration along the geometric path, a comparison among the weights of all the visible pairs allows to select a subset of the k most relevants (step 4). The landmarks that either having the best properties of localization or representing a risk of collision with the path will be then chosen. Afterwards, the piece of the path where the chosen subset holds as the best is determined (step 5). Hence, the geometric path is subdivided in a set of pieces with corresponding k best landmarks. The loop-steps 7, 8 and 9 of this algorithm ensure smoothing switches among every set of k best landmarks along the geometric path. In presence of multiple sensors, this algorithm is repeated for each one of them. In [15] we define a set of weight functions for a laser scanner sensing 2D segments and we propose detailed methods for their construction.

The determination of the number k is a critical issue for such an algorithm. Considering a laser range finder, a minimum number of two landmarks are required for the localization in the plane (if we consider non-parallel lines). In practice this number is too low because of the risk of ill-conditioned localization



Figure 8: On the left picture, a collision free trajectory is planned through a corridor. On the right picture, four straight line segments used in this experiment: seg_{101} , seg_{164} , seg_{205} , and seg_{206} and a reference configuration along the trajectory are displayed.

matrix (for instance in the case of two quasi-parallel lines). Also, because an over-constrained localization is less sensitive to the data noise, at least 5 landmarks are enough in practice.

The next section shows an experimental validation of the proposed strategy. The validation goes from a simple application to a navigation in an indoor constrained environment.

4 Experimental results

We now present three real experiments. The first one illustrates how weighted landmark enables the robot to cope with map uncertainty. The second one shows how to perform a parking task between two obstacles where the position of each of them is not precisely known. The third experiment shows a navigation task with highly constrained door crossing.

4.1 Maneuver in a corridor

In this first experiment, we first plan a collision-free path supplied with landmark based features in a reference map. The planned landmark based motion is then executed in an environment slightly different from the map (Figure 8 left). Figure 8-right displays a configuration along the reference trajectory and the four straight-line segments used to execute the motion. We put a long board in front of one of the corridor-walls to simulate a different position of the wall. The real corridor is thus narrower than in the map. The purpose of this experiment is to illustrate the motion control task, the weights associated to each segment are display in Figure 9. Figure 10 shows the executed trajectory. Figure 11 shows the end of the reference trajectory (left) and the end of the executed trajectory (right). At the end of the trajectory of this experiment we can notice that only two landmarks are used. There is no problem of localization with these landmarks because the landmarks are perpendicular walls and the matrix is made well conditioned. Remark that the end configuration is at the specified distance to segment 101.

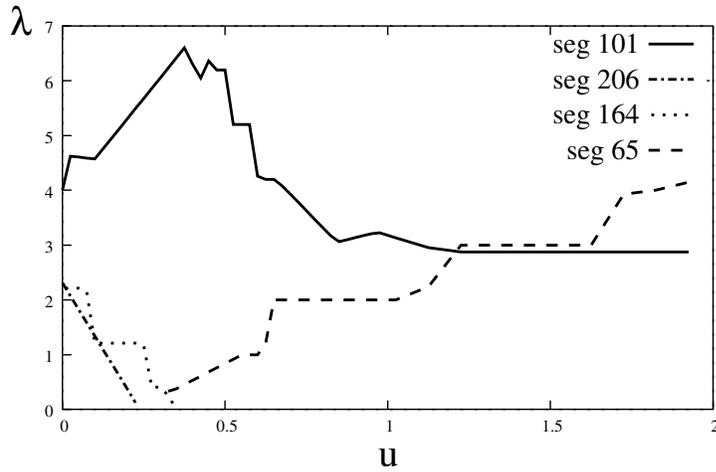


Figure 9: The weight associated to each landmark.

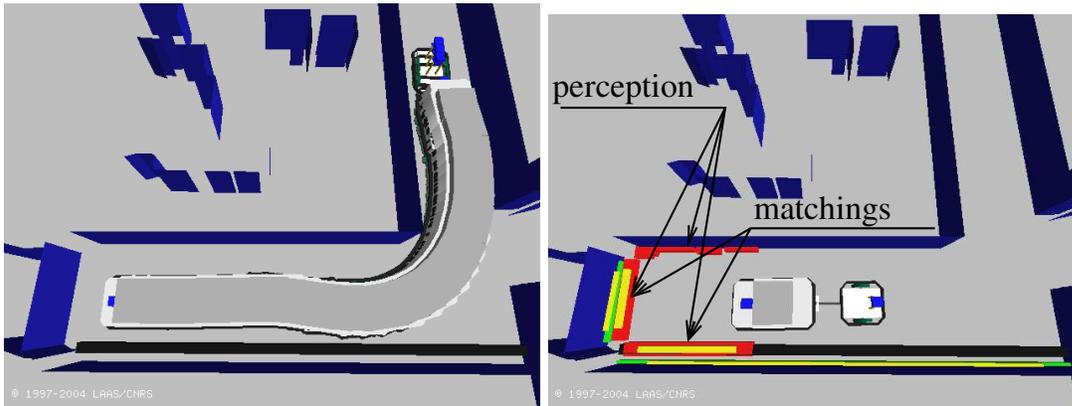


Figure 10: Execution of a landmark-based motion in an inaccurate map of the environment. On the left picture, the position of a wall is different in the map and in the real environment. Despite this error, the robot adapts the trajectory to stay at the specified distance to the wall. On the right picture, the position of the robot corresponding to the reference configuration displayed on Figure 8 (right). Two segments of the map are matched.

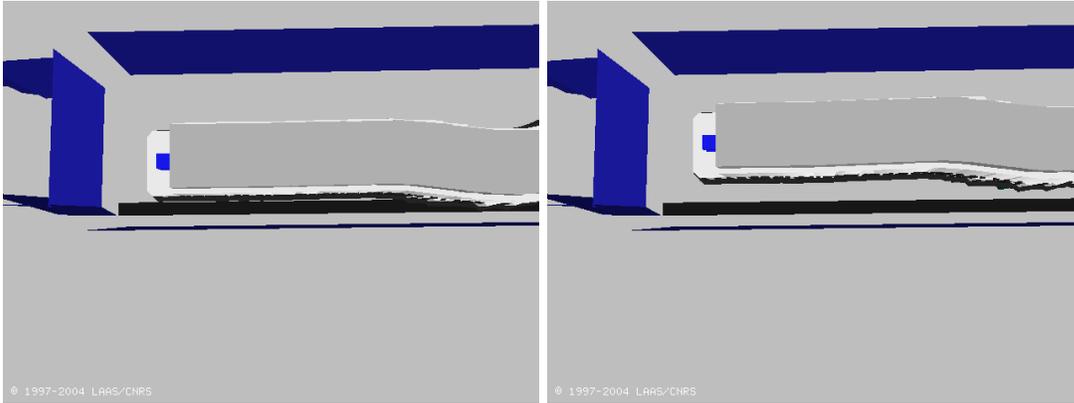


Figure 11: Zoom of the end of the trajectory. On the left, the reference trajectory; on the right, the trajectory executed. Let us notice that the final configuration is at the specified distance to segment 101.

4.2 Parking manoeuver

In this second experiment, we illustrate our approach with a parking task. Figure 12 shows the landmark used for this motion. The robot is required to park between two parallel segments (11 and 12).

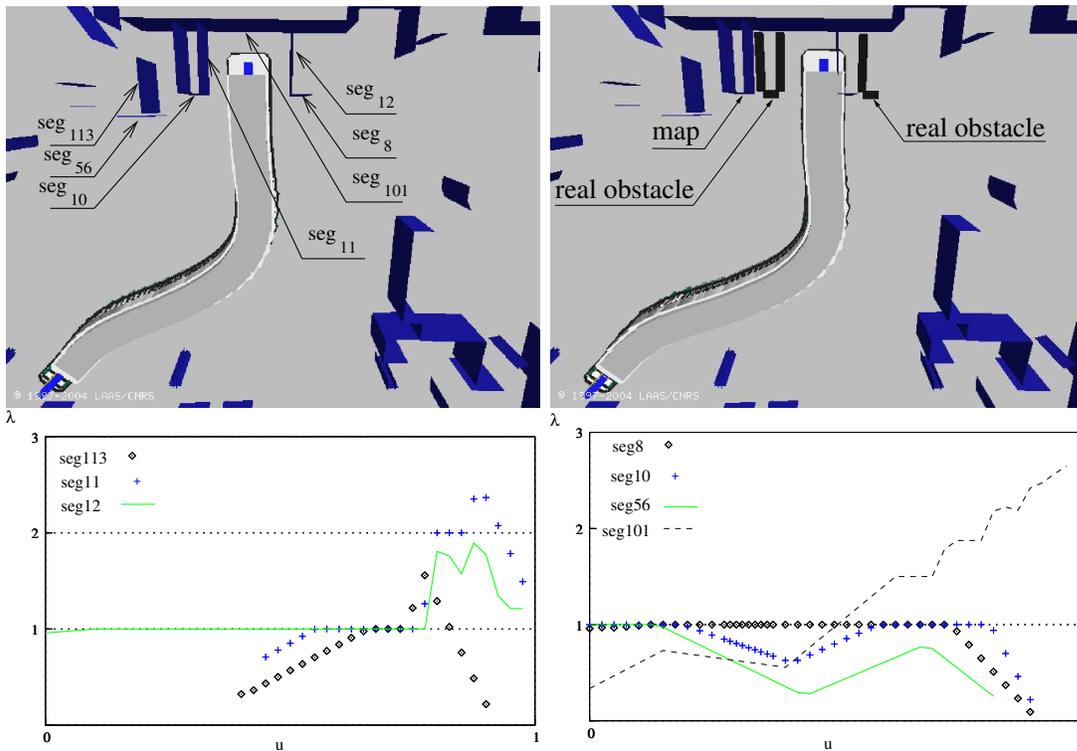


Figure 12: On top left, the reference trajectory planned in the map of the environment and the segments used to execute the motion. On top right, segments 8, 10, 11, and 12 are moved to the right. However, the robot adapts the motion to park between these landmarks. Middle and bottom, the weights associated to each landmark.

The reference trajectory planned in the map of the environment is displayed on the left. On the right, the segments defining the parking configuration are moved to the right. the figure on the right shows the executed trajectory.

At a first stage, a geometric non-collision path is planned in this map from an initial to a final configuration so that the robot will be able to enter the car park, as show in figure (12). The second stage involves planning the landmark based motion with the generic platform we developed. Along the geometric planned path, the five best sensor-landmark pairs are selected according to their weights. Before executing the landmark based motion, the car park is shifted right to modify the real environment in relation to the reference map. During the execution of the movement, the robot corrects its trajectory regarding to the new pose of the parking and then the task is led with success.

4.3 Navigation in a long corridor



Figure 13: The reference geometric planned path. The selected landmarks and their weights in the first tricky passage (enclosed by the circle in bottom left) are depicted in the figures below.

In the corridors of our laboratory we plan a geometric free path with Move3D (see figure 13). Thereafter, the produced path and the reference map of the environment are used by the landmark based motion planner to select the most relevant landmarks.

The navigation task we describe involves some difficulties that have to be raised :

- The reference map is constructed based on the odometry and the laser scanner information and is not exact. Indeed, we pick up some errors in terms of distances between walls in the reference map and in the real environment (the difference is about an average value of some ten centimeters).

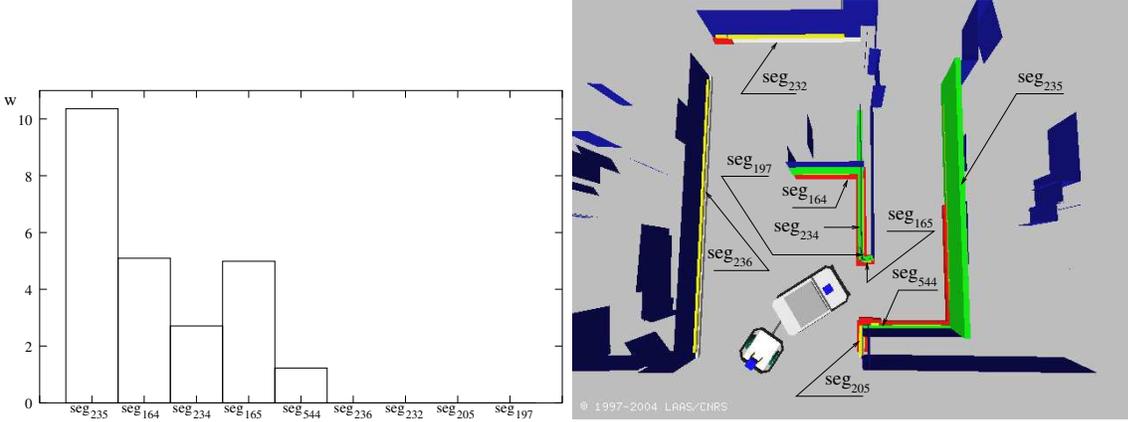


Figure 14: The left figure shows the instantaneous weights associated to selected landmarks viewed by the front sensor for three configurations of the right figure. The green segments are selected by the *LBM* algorithm.

- The size of the robot in relation to the free space of the environment is a critical issue for the achievement of the navigation task. Indeed, unlike the parking manoeuvre (c.f. section 4.2) where the robot has a large free space, here the crossings are narrower and the manoeuvres are geometrically very constrained.

Although this difficulty constrains hardly the achievement of the navigation task, Hilare 2 drives with success the circuit using the landmark based motion to correct its path. The figure (14) shows the details on the first constraint narrow crossing of the figure 13. The details of this navigation task are in [23].

Even if the odometric error is not important at this step of the navigation, the errors of the map can generate failure in the experiment. The segments seg_{164} , seg_{165} , seg_{234} , seg_{235} and seg_{544} , are used for localization. The passage being narrower in reality than in the map, the segments seg_{165} and seg_{544} have big values of weights to ensure a safe crossing.

The scenario of this experiment shows the relevance of the formalism in a local area when the reference map is inaccurate.

5 Conclusion

The main contribution of this work is the definition of the landmark based motion. It is a geometric-free path provided with weighed sensor-landmark features. The weight functions define the local relevance of the obstacles and the switching strategy among the landmarks of the environment. The local relevance is used in a weighed localization process while the switching strategy enables smooth correction during the execution of the trajectory. Our formalism is integrated as a generic framework architecture and applied with succes on a non-holonomic mobile robot towing a trailer. Our robot uses two laser scanners as

external sensors and planar segments as landmarks of the environment. The real experiments highlight how the robot corrects its configuration in the real world with respect to the planned geometric path based on the selected landmark. Moreover, this strategy handles critical crossings as emphasized by the last experiment.

REFERENCES

- [1] R. Alami, R. Chatila, S. Fleury, M. Herrb, F. Ingrand, M. Khatib, B. Morrisset, P. Moutarlier, and T. Simon. Around the lab in 40 days... In *IEEE International Conference on Robotics and Automation*, pages 88–94, San Francisco, CA, Apr 2000.
- [2] O. Brock and O. Khatib. High-speed navigation using the global dynamic window approach. In *IEEE International Conference on Robotics and Automation*, pages 20–100, San Francisco, CA, Apr 2000.
- [3] J. Choi and E. Amir. Combining planning and motion planning. In *IEEE International Conference on Robotics and Automation.*, pages 238–244, May 2009.
- [4] Howie Choset, Kevin M. Lynch, Seth Hutchinson, George A Kantor, Wolfram Burgard, Lydia E. Kavraki, and Sebastian Thrun. *Principles of Robot Motion: Theory, Algorithms, and Implementations*. MIT Press, 2005.
- [5] B. Espiau, F. Chaumette, and P. Rives. A new approach to visual servoing in robotics. *IEEE Transactions on Robotics and Automation*, 8(3):313–326, 1992.
- [6] Ruijie He, Sam Prentice, and Nicholas Roy. Planning in information space for a quadrotor helicopter in a gps-denied environments. In *IEEE International Conference on Robotics and Automation*, pages 1814–1820, Los Angeles, CA, 2008.
- [7] B. Jensen, G. Froidevaux, X. Greppin, A. Lorotte, L. Mayor, M. Meisser, G. Ramel, and R. Siegwart. The interactive autonomous mobile system roblox. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1221–1227, Lausanne, Switzerland, Oct 2002.
- [8] L. Jetto, S. Longhi, and G. Venturini. Development and experimental kalman filter for the localization of mobile robots. *IEEE Transactions on Robotics and Automation*, 15(2):219–229, April 1999.
- [9] M. Khatib, H. Jaouni, R. Chatila, and J.-P. Laumond. Dynamic path modification for car-like nonholonomic mobile robots. In *IEEE International Conference on Robotics and Automation*, pages 2920–2925, Albuquerque, NM, Apr 1997.
- [10] F. Lamiroux, D. Bonnafous, and O. Lefebvre. Reactive path deformation for nonholonomic mobile robots. *IEEE Transactions on Robotics*, 20(6):967–977, Dec 2004.

- [11] J.-C. Latombe. *Robot Motion Planning*. Kluwer, Boston, MA, 1991.
- [12] J.-P. Laumond, editor. *Robot Motion Planning and Control*. Springer, 1998.
- [13] S. M. LaValle. *Planning Algorithms*. Cambridge University Press, 2006.
- [14] R. Lerner, E. Rivlin, and I. Shimshoni. Landmark selection for task-oriented navigation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2785–2791, Oct. 2006.
- [15] A.C. Malti, F. Lamiroux, and M. Taix. Sensor landmark succession for motion planning along a planned trajectory. In *IEEE Conference on Mechatronics and Robotics*, Aachen, Germany, 2004.
- [16] J. Minguez and L. Montano. Nearness diagram (nd) navigation: collision avoidance in troublesome scenarios. *IEEE Transactions on Robotics and Automation*, 20(1):45–59, Feb 2004.
- [17] Clark F. Olson. Probabilistic self-localization for mobile robots. *IEEE Transactions on Robotics and Automation*, 16(1):55–66, February 2000.
- [18] S. Quinlan and O. Khatib. Elastic bands: Connecting path planning and control. In *IEEE International Conference on Robotics and Automation*, pages 802–807, Atlanta, GA, May 1993.
- [19] Nathan Ratliff, Matt Zucker, J. Andrew Bagnell, and Siddhartha Srinivasa. Chomp: Gradient optimization techniques for efficient motion planning. In *IEEE International Conference on Robotics and Automation.*, pages 489–494, May 2009.
- [20] C. Samson, B. Espiau, and M. Le Borgne. *Robot Control The Task Function Approach*. Oxford science publications, 1991.
- [21] T. Simeon, J.-P. Laumond, and C. Nissoux. Visibility-based probabilistic roadmaps for motion planning. *Journal of Advanced Robotics*, 14(6):477–494, 2000.
- [22] T. Simon, J.-P. Laumond, and F. Lamiroux. Move3d : a generic platform for path planning. In *IEEE Int. Symp. on Assembly and Task Planning*, pages 25–30, 2001.
- [23] M. Taix, A. Malti, and F. Lamiroux. Planning robust landmarks for sensor based motion. In *European Robotics Symposium 2008*, Springer Tracts in Advanced Robotics. Kluwer, 2008.
- [24] S. Thrun, M. Bennewitz, W. Burgard, A. Cremers, F. Dellaert, D. Fox, D. Hahnel, C. Rosenberg, N. Roy, J. Schulte, and D. Schulz. Minerva: a second-generation museum tour-guide robot. pages 1999–2005, Detroit, MC, May 1999.
- [25] Liangjun Zhang, S.M. LaValle, and D. Manocha. Global vector field computation for feedback motion planning. In *IEEE International Conference on Robotics and Automation.*, pages 477–482, May 2009.