



# Reactive Path Deformation for Nonholonomic Mobile Robots

Florent Lamiriaux, David Bonnafeous, Olivier Lefebvre

## ► To cite this version:

Florent Lamiriaux, David Bonnafeous, Olivier Lefebvre. Reactive Path Deformation for Non-holonomic Mobile Robots. IEEE Transactions on Robotics, 2004, 20 (6), pp.967-977. 10.1109/TRO.2004.829459 . hal-02294191

**HAL Id: hal-02294191**

**<https://laas.hal.science/hal-02294191>**

Submitted on 23 Sep 2019

**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.

# Reactive Path Deformation for Nonholonomic Mobile Robots

F. Lamiriaux, D. Bonnafous, and O. Lefebvre  
LAAS-CNRS, Toulouse, France  
{florent,dbonnafo,olefebvr}@laas.fr

**Abstract**—This paper presents a novel and generic approach of path optimization for nonholonomic systems. The approach is applied to the problem of reactive navigation for nonholonomic mobile robots in highly cluttered environments. A collision-free initial path being given for a robot, obstacles detected while following this path can make it in collision. The current path is iteratively deformed in order to get away from obstacles and satisfy the nonholonomic constraints. The core idea of the approach is to perturb the input functions of the system along the current path in order to modify this path, making an optimization criterion decrease.

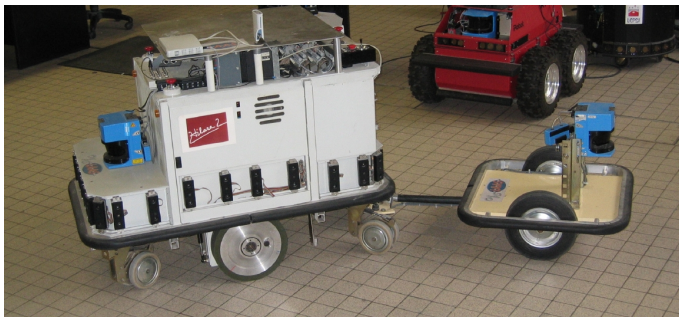


Fig. 1. Hilare 2 towing a trailer: a nonholonomic system of dimension 4 with 2 nonholonomic constraints.

## I. INTRODUCTION

Most wheeled vehicles are subject to constraints of rolling without slipping and thus belong to the large class of nonholonomic systems. Buses, trailer-truck systems, and cars are a few examples. Research efforts have been made in the past to understand and control the motion of these systems. These works were first initiated in robotics when researchers discovered that wheeled mobile robots are nonholonomic. Today, car manufacturers are very interested in motion control of wheeled vehicles. Most of them plan to equip their vehicles in a near future with computer aided motion capabilities like parallel parking or automatic stop-and-go mode in traffic jams. Thus, better understanding and controlling the motions of nonholonomic systems will open a large field of industrial applications in the domain of transportation.

Producing automatic motions for nonholonomic mobile robots soon revealed a difficult task. For this reason, the problem has been decomposed into two steps. The first step consists in computing a collision-free motion using a map of the environment. The second step consists in executing the motion. As a consequence, past research on nonholonomic systems has mainly focused on two aspects: path planning on the one

A shorter version of this paper has been published in the International Conference on Robotics and Automation

Corresponding author.

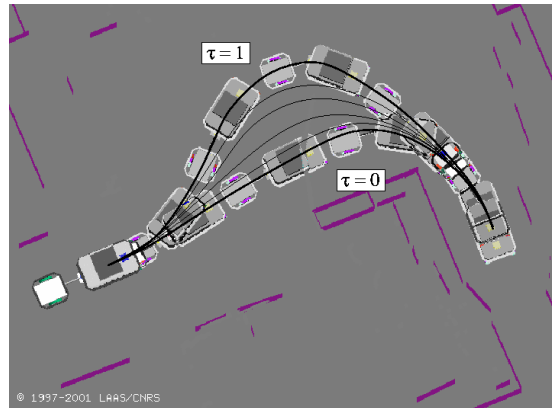


Fig. 2. Deformation of the path of mobile robot Hilare 2 towing a trailer. Each path along the deformation process is represented by a curve of abscissa  $s$  in the configuration space, while  $\tau$  represents time. In this example, the robot is on the left and obstacles (not represented on the picture) detected by a laser scanner make the initial path ( $\tau = 0$ ) in collision.

hand [16], [26], [27], [14], [17], [4], [5], [18] and motion control on the other hand [23], [19], [9], [25]. Few of these works have addressed both aspects together.

For long range motions, the above two step approach raises three issues: localization uncertainty, imprecision of the map of the environment and unexpected obstacles that are not in the map. These three issues have a common consequence: a path initially planned to be collision-free may become in collision at the execution step. To overcome these difficulties, [21] proposed a method that enables him to deform on line the path to be followed by the robot in order to get away from obstacles detected along the motion. This approach has been extended to the case of a unicycle-like mobile robot in [10] and then to the case of a holonomic mobile manipulator in [3]. In both papers, the geometry of the robot is approximated by a set of balls and no or only one very simple nonholonomic constraint is treated. None of these methods is applicable to more complex nonholonomic systems like car-like robots.

To plan and execute motions in dynamic environments, [8] developed the concept of velocity obstacles, defining the set of forbidden velocities given the velocity of the obstacles. This concept is used in [15] to perform local goal oriented obstacle avoidance. This technique is particularly efficient in environments where a lot of obstacles are moving since the velocity of the obstacles is taken into account in the avoidance strategy. However, it is based on very simple models of the robot and of the obstacles: they all are spherical. This simplification forbids applications for multi-body mobile robots moving in very clut-

tered environments where the robot needs to pass very close to the obstacles.

In this paper, we propose a novel and generic approach of path modification applicable to any nonholonomic system. We assume that a first collision-free path has been computed for the robot in the global frame. In our experiments, this path is computed by the motion planner Move3D [24] based on a map of the environment. When the robot follows this path, on-board sensors, for instance laser scanners, detect surrounding obstacles and map them in the global frame. If an obstacle not present in the map is detected, it can be in collision with the initial path. If the localization of the robot is inaccurate, or if the map is inexact, obstacles of the map might be seen in collision with the initial path by the sensors. The method we propose in this paper enables the robot to deform the initial path in order to move it away from sensed obstacles and make the current path collision-free (Figure 6 shows an example of path deformed and followed by mobile robot Hilare 2 towing a trailer). The current path thus changes along time. As a path is a mapping from an interval of real numbers into the configuration space of the robot, we naturally model a path deformation process as a mapping of two real variables  $s$  and  $\tau$  into the configuration space.  $\tau$  can be considered as time (or more generally as an increasing function of  $\tau$ ), while  $s$  is the abscissa along each path. Figure 2 illustrates this idea.

Our contribution consists of a theoretical framework in which a path deformation process is modeled as a dynamic control system, of an algorithm controlling the deformation process and of the validation of our approach by applications to two different kinematic systems.

This path deformation method can be applied to obstacle avoidance for nonholonomic systems, as well as any other optimization problem.

The paper is organized as follows. In Section II, we propose a model of path deformation as a infinite-dimensional dynamic control system the state of which is a path. In Section III, we present an iterative algorithm controlling the deformation process to make an optimization criterion decrease. At each step of the algorithm, a deformation is computed in a finite-dimensional linear-subspace (III-A). Boundary conditions can be added to constrain the initial and final configurations of the path to keep unchanged after deformation (III-B). Using configuration space potential fields, the optimization criterion can be related to obstacles and make the deformation process avoid them (III-C). Discretization of the deformation process implies however deviation of the nonholonomic constraints that are not satisfied anymore. A correction of this undesirable effect is proposed (III-D). Finally, in Section IV, the approach is applied to different kinematic systems.

## II. NONHOLONOMIC PATH DEFORMATION AS A DYNAMIC CONTROL SYSTEM

A path for a robotic system is usually represented by a mapping from an interval of  $\mathbf{R}$  into the configuration space of the system. In this section, we introduce the notion of path deformation as a mapping from an interval of  $\mathbf{R}$  into the set of paths. Equivalently a path deformation is a mapping from two intervals into the configuration space as explained later in this section.

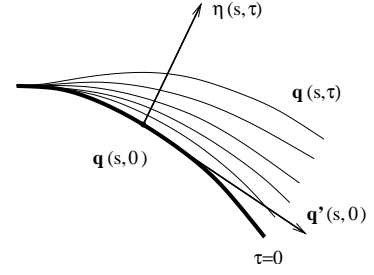


Fig. 3. Current path  $\mathbf{q}(s)$  (in bold) and direction of deformation  $\eta(s)$  along this path.

### A. Admissible paths

A nonholonomic system of dimension  $n$  is characterized by a set of  $k < n$  vector fields  $\mathbf{X}_1(\mathbf{q}), \dots, \mathbf{X}_k(\mathbf{q})$ , where  $\mathbf{q} \in \mathcal{C} = \mathbf{R}^n$  is the configuration of the system. For each configuration  $\mathbf{q}$ , the admissible velocities of the system is the set of linear combinations of the  $\mathbf{X}_i(\mathbf{q})$ 's. A path  $\mathbf{q}(s)$  is a smooth curve in the configuration space defined over an interval  $[0, S]$ . A path is said to be *admissible* if and only if there exists a  $k$ -dimensional smooth vector valued mapping  $\mathbf{u} = (u_1, \dots, u_k)$  defined over  $[0, S]$  and such that:

$$\forall s \in [0, S] \quad \mathbf{q}'(s) = \sum_{i=1}^k u_i(s) \mathbf{X}_i(\mathbf{q}(s)) \quad (1)$$

where from now on,  $'$  denotes the derivative w.r.t.  $s$ .

### B. Admissible path deformation

We call *path deformation* a mapping from a subset  $[0, S] \times [0, +\infty)$  of  $\mathbf{R}^2$  to the configuration space of the system:

$$(s, \tau) \rightarrow \mathbf{q}(s, \tau)$$

For each value of  $\tau$ ,  $s \rightarrow \mathbf{q}(s, \tau)$  is a path.  $s \rightarrow \mathbf{q}(s, 0)$  is called the *initial path*. A path deformation process can be compared to a vibrating string of abscissa  $s$  where  $\tau$  is the time. The shape of the string varies with time  $\tau$  and is given by the curve  $s \rightarrow \mathbf{q}(s, \tau) \in \mathbf{R}^2$ . In order to keep notation light and intuitive, we use the same notation  $\mathbf{q}$  to denote configurations, paths and path deformations.

We are interested in deformations  $\mathbf{q}(s, \tau)$  composed of only admissible paths. Such deformations satisfy the following constraint: there exists a  $k$ -dimensional vector valued smooth mapping  $\mathbf{u} = (u_1, \dots, u_k)$  defined over  $[0, S] \times [0, +\infty)$ , such that  $\forall (s, \tau) \in [0, S] \times [0, +\infty)$

$$\frac{\partial \mathbf{q}}{\partial s}(s, \tau) = \sum_{i=1}^k u_i(s, \tau) \mathbf{X}_i(\mathbf{q}(s, \tau)) \quad (2)$$

For each value of  $\tau$ ,  $s \rightarrow \mathbf{u}(s, \tau)$  is the input function of path  $s \rightarrow \mathbf{q}(s, \tau)$ . The above equation simply expresses constraint (1) for each path of the deformation. As well as a path is uniquely defined by its initial configuration and the input function, a path deformation is uniquely defined by the initial configuration  $\mathbf{q}(0, \tau)$  of each of its paths and by the input functions  $u_i(s, \tau)$ .

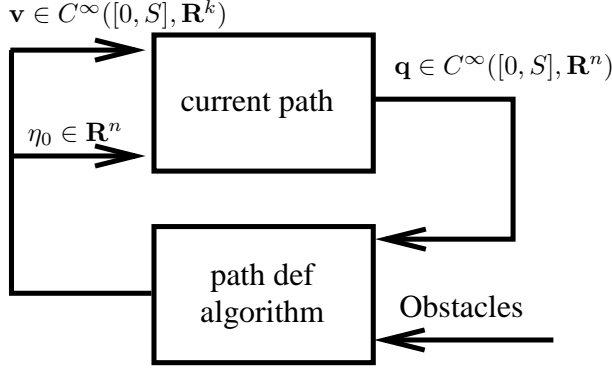


Fig. 4. A path deformation process can be modelled as a dynamic control system of time  $\tau$ . At each time, the state is a feasible path  $\mathbf{q}$ , the input is a pair  $(\eta_0, \mathbf{v})$  that uniquely defines the time derivative of the state. The path deformation algorithm we build in this paper can be considered as a closed-loop controller that computes the input of the dynamic control system with respect to the current path and a task to achieve, for instance avoiding obstacles.

By differentiating (2), we get a relation between the input variation  $\frac{\partial \mathbf{u}}{\partial \tau}$  and the infinitesimal path deformation  $\frac{\partial \mathbf{q}}{\partial \tau}$  when the deformation parameter  $\tau$  increases:

$$\frac{\partial^2 \mathbf{q}}{\partial s \partial \tau}(s, \tau) = \sum_{i=1}^k \left( \frac{\partial u_i}{\partial \tau}(s, \tau) \mathbf{X}_i(\mathbf{q}(s, \tau)) + u_i(s, \tau) \frac{\partial \mathbf{X}_i}{\partial \mathbf{q}}(\mathbf{q}(s, \tau)) \frac{\partial \mathbf{q}}{\partial \tau}(s, \tau) \right)$$

We call respectively *input perturbations* and *direction of deformation* the following vector valued functions:

$$\begin{aligned} \mathbf{v}(s, \tau) &\triangleq \frac{\partial \mathbf{u}}{\partial \tau}(s, \tau) \\ \eta(s, \tau) &\triangleq \frac{\partial \mathbf{q}}{\partial \tau}(s, \tau) \end{aligned}$$

$\eta$  is represented in Figure 3. With this notation, the above equation becomes :

$$\eta'(s, \tau) = A(s, \tau)\eta(s, \tau) + B(s, \tau)\mathbf{v}(s, \tau) \quad (3)$$

where  $\eta' = \frac{\partial \eta}{\partial s}$  and  $A(s, \tau)$  is the following  $n \times n$  matrix:

$$A(s, \tau) = \sum_{i=1}^k u_i(s, \tau) \frac{\partial \mathbf{X}_i}{\partial \mathbf{q}}(\mathbf{q}(s, \tau))$$

and  $B(s, \tau)$  is the  $n \times k$  matrix the columns of which are the control vector fields:

$$B(s, \tau) = \begin{pmatrix} \mathbf{X}_1(\mathbf{q}(s, \tau)) & \cdots & \mathbf{X}_k(\mathbf{q}(s, \tau)) \end{pmatrix}$$

According to (3), the derivative w.r.t.  $\tau$  of the path  $s \rightarrow \mathbf{q}(s, \tau)$  is related to the input perturbation  $s \rightarrow \mathbf{v}(s, \tau)$  through a linear dynamic system. This system is in fact the linearized system of (1) about  $s \rightarrow \mathbf{q}(s, \tau)$ . For a given path  $\mathbf{q}(s, \tau)$  of input  $\mathbf{u}(s, \tau)$  and for any input perturbation  $\mathbf{v}(s, \tau)$ , and any initial

condition  $\eta_0 = \eta(0, \tau)$  we can integrate Equation (3) w.r.t.  $s$  to get the corresponding direction of deformation  $\eta(s, \tau)$ . A path deformation process for nonholonomic systems can thus be considered as a dynamic control system where

- $\tau$  is the time,
- $s \rightarrow \mathbf{q}(s, \tau)$  is the state and
- the input is a pair  $(\eta_0, s \rightarrow \mathbf{v}(s, \tau))$ .

as described in Figure 4. In Section III, paths will be computed for discretized values of  $\tau$  only.

### C. Potential field and inner product

The path deformation method needs to compute at each time  $\tau$  a vector  $\eta_0$  and a function  $s \rightarrow \mathbf{v}(s, \tau)$  over  $[0, S]$  in such a way that the deformation process achieves a specified goal. This goal is expressed in terms of a potential value to minimize over the set of feasible paths. The potential value of a path is defined by integration along the path of a potential field  $U$  over the configuration space. We denote by  $V(\tau)$  the potential value of path  $s \rightarrow \mathbf{q}(s, \tau)$ :

$$V(\tau) \triangleq \int_0^S U(\mathbf{q}(s, \tau)) ds$$

If the goal to achieve is to avoid obstacles, as in [22], [11], [1], the configuration space potential field is defined in such a way that its value is high for configurations close to obstacles and low for configuration far from obstacles. Thus paths going close to obstacles have a high potential value and paths staying far from obstacles have a low potential value.

The path potential variation w.r.t.  $\tau$  is related to  $\eta(s, \tau)$  by the following expression:

$$\frac{dV}{d\tau}(\tau) = \int_0^S \frac{\partial U}{\partial \mathbf{q}}(\mathbf{q}(s, \tau))^T \eta(s, \tau) ds$$

The principle of the path deformation method consists in choosing  $(\eta_0, \mathbf{v}(s, \tau))$  in such a way that  $\frac{dV}{d\tau}(\tau)$  is negative. Let us notice that the space of vector-valued functions defined over interval  $[0, S]$  is an Euclidean space, the inner product of which is defined by:

$$(f|g)_{L^2} \triangleq \int_0^S f(s)^T g(s) ds \quad (4)$$

With this definition, the variation of the path potential value along a direction of deformation can be rewritten

$$\frac{dV}{d\tau}(\tau) = \left( \frac{\partial U}{\partial \mathbf{q}} \circ \mathbf{q} | \eta \right)_{L^2}$$

where  $\circ$  denotes the composition of mappings. Let us notice that integration is performed over variable  $s$  only. According to this expression,  $\eta = -(\frac{\partial U}{\partial \mathbf{q}} \circ \mathbf{q})$  is at equivalent  $L^2$ -norm the direction of deformation that minimizes  $\frac{dV}{d\tau}(\tau)$ . Unfortunately, this value of function  $\eta$  is not an admissible direction of deformation (*i.e.* a solution of system (3)). A solution could be obtained by orthogonally projecting  $-(\frac{\partial U}{\partial \mathbf{q}} \circ \mathbf{q})$  over the linear-subspace of admissible directions of deformation. However, the projection of a vector over an infinite-dimensional subspace does not necessarily exist.

To overcome this problem, we will restrict the input perturbation to a finite-dimensional subspace in the following section.

### III. NONHOLONOMIC PATH DEFORMATION ALGORITHM

Based on the theoretical framework established in the previous section, we build in this section the path deformation algorithm for nonholonomic systems. Starting from an initial admissible path  $\mathbf{q}(s, 0)$ , the algorithm iteratively computes a sequence of admissible paths  $s \rightarrow \mathbf{q}(s, \tau_j)$  for discretized values  $\tau_j$  of  $\tau$  where  $j$  is an integer. At each iteration of the algorithm, a direction of deformation  $\eta(s, \tau_j)$  is generated based on the configuration space potential field  $U$  and a new path  $\mathbf{q}(s, \tau_{j+1})$  is computed as follows:

$$\mathbf{q}(s, \tau_{j+1}) = \mathbf{q}(s, \tau_j) + \Delta\tau_j \eta(s, \tau_j) \quad (5)$$

$$\tau_{j+1} = \tau_j + \Delta\tau_j \quad (6)$$

where  $\Delta\tau_j$  is the discretization step. Let us notice that the above formula is a first-order approximation in  $\tau$ . In the rest of this section, we describe the different steps of the algorithm. In Section III-A, we compute  $\eta(s, \tau_j)$  by restricting the input perturbation to a finite-dimensional subspace of functions. This restriction enables us in Section III-B to take into account boundary conditions that force the initial and final configuration of the deformation interval to remain unchanged. In Section III-C we explain how to compute the direction of deformation that minimizes the variation of the path potential under constant  $L^2$ -norm. The first order approximation (5) induces deviations of the nonholonomic constraints. Section III-D addresses this issue and proposes a correction of this deviation.

#### A. Finite-dimensional subspace of input perturbations

As explained in section II, the control variables of a path deformation process are the input perturbation  $\mathbf{v}$  and the initial condition  $\eta_0$ .  $s \rightarrow \mathbf{v}(s, \tau_j)$  belongs to the infinite-dimensional space of smooth vector-valued functions defined over  $[0, S]$ . To simplify the control of the path deformation, we choose to restrict  $\mathbf{v}$  over a finite-dimensional subspace of functions. This restriction will make the boundary conditions introduced later in section III-B easier to deal with. Let  $p$  be a positive integer. We define  $\mathbf{e}_1, \dots, \mathbf{e}_p$ , a set of smooth linearly independent vector-valued functions of dimension  $k$ , defined over  $[0, S]$ :

$$\mathbf{e}_i : [0, S] \rightarrow \mathbf{R}^k$$

Various choices are possible for the  $\mathbf{e}_i$ 's (e.g. truncated Fourier series, polynomials,...) [20], [7], [6]. For each of these functions, we define  $\mathbf{E}_i(s, \tau_j)$  as the solution of system (3) with initial condition  $\eta_0 = 0$  and with  $\mathbf{e}_i(s)$  as input:

$$\mathbf{E}_i'(s, \tau_j) = A(s, \tau_j)\mathbf{E}_i(s, \tau_j) + B(s, \tau_j)\mathbf{e}_i(s) \quad (7)$$

$$\mathbf{E}_i(0, \tau_j) = 0 \quad (8)$$

where matrices  $A$  and  $B$  are defined in section II-B. Let us recall that these matrices depend on the current path  $\mathbf{q}(s, \tau_j)$  of input  $\mathbf{u}(s, \tau_j)$  and therefore, unlike  $\mathbf{e}_i$ ,  $\mathbf{E}_i$  depends on  $\tau_j$ .

If we restrict  $\mathbf{v}(s, \tau_j)$  in the set of functions spanned by the  $\mathbf{e}_i$ 's, that is:

$$\mathbf{v}(s, \tau_j) = \sum_{i=1}^p \lambda_i \mathbf{e}_i(s) \quad (9)$$

where  $\lambda = (\lambda_1, \dots, \lambda_p) \in \mathbf{R}^p$  is a vector, as (3) is linear, the direction of deformation  $\eta$  corresponding to  $\mathbf{v}$  is the same linear combination of solutions  $\mathbf{E}_i$

$$\eta(s, \tau_j) = \sum_{i=1}^p \lambda_i \mathbf{E}_i(s, \tau_j) \quad (10)$$

Using this restriction, the input perturbation  $\mathbf{v}$  is uniquely defined by vector  $\lambda$ .

#### B. Boundary conditions

We wish the deformation process not to modify the initial and goal configurations of the path. We thus impose the following boundary conditions:

$$\forall j > 0, \quad \mathbf{q}(0, \tau_j) = \mathbf{q}(0, 0)$$

$$\mathbf{q}(S, \tau_j) = \mathbf{q}(S, 0)$$

These constraints are equivalent to:

$$\forall j > 0, \quad \eta(0, \tau_j) = 0 \quad (11)$$

$$\eta(S, \tau_j) = 0 \quad (12)$$

Equation (8) and Expression (10) ensure us that the first constraint (11) is satisfied. The second constraint (12) together with Expression (10) becomes a linear constraint over vector  $\lambda$ :

$$L\lambda = 0 \quad (13)$$

where  $L$  is a  $n \times p$  - matrix the columns of which are the  $\mathbf{E}_i(S, \tau_j)$ 's:

$$L = ( \mathbf{E}_1(S, \tau_j) \quad \dots \quad \mathbf{E}_p(S, \tau_j) )$$

Let us notice that in general, the dimension of the subspace of solutions of the above linear system is equal to  $p - n$  and therefore  $p$  must be bigger than  $n$ . The problem is now to choose a vector  $\lambda$  satisfying the above linear constraint and generating a direction of deformation that makes the current path move away from obstacles. We address this issue in the following section.

#### C. Direction of deformation that makes the path potential decrease

As explained in II-C, a potential field  $U$  is defined over the configuration space. This potential field defines a potential function  $V$  over the space of paths by integration.

Given a vector  $\lambda \in \mathbf{R}^p$ , the variation of the path potential induced by direction of deformation  $\eta$  defined by Equation (10) is given by the following expression:

$$\frac{dV}{d\tau}(\tau_j) = \int_0^S \frac{\partial U}{\partial \mathbf{q}}(\mathbf{q}(s, \tau_j))^T \eta(s, \tau_j) ds \quad (14)$$

$$= \sum_{i=1}^p \lambda_i \int_0^S \frac{\partial U}{\partial \mathbf{q}}(\mathbf{q}(s, \tau_j))^T \mathbf{E}_i(s, \tau_j) ds \quad (15)$$

Let us define the following coefficients

$$\mu_i \triangleq \int_0^S \frac{\partial U}{\partial \mathbf{q}}(\mathbf{q}(s, \tau_j))^T \mathbf{E}_i(s, \tau_j) ds$$

These coefficients represent the variation of the path potential induced by each direction of deformation  $\mathbf{E}_i$ . With these coefficients, Expression (15) can be rewritten as follows:

$$\frac{dV}{d\tau}(\tau_j) = \sum_{i=1}^p \lambda_i \mu_i \quad (16)$$

Thus, if we set

$$\lambda_i = -\mu_i \quad (17)$$

we get a deformation  $\eta(s, \tau_j)$  that keeps the kinematic constraints satisfied and that makes the path potential decrease. Indeed,

$$\frac{dV}{d\tau}(\tau_j) = - \sum_{i=1}^p \mu_i^2 \leq 0$$

We denote by  $\lambda^0$  this value of vector  $\lambda$ . Nothing ensures us that  $\lambda^0$  satisfies the boundary conditions (13).

### C.1 Projection over the boundary condition subspace

Equation (13) states that the set of vectors  $\lambda$  satisfying the boundary conditions is a linear subspace of  $\mathbf{R}^p$ . To get such a vector that we denote  $\bar{\lambda}$ , we project  $\lambda^0$  over this subspace :

$$\bar{\lambda} = (I_p - L^+ L) \lambda^0$$

where  $L^+$  is the pseudo-inverse of  $L$ . As  $L L^+ L = L$ ,  $\bar{\lambda}$  satisfies  $L \bar{\lambda} = 0$ . We may naturally wonder whether the direction of deformation after projection  $\eta = \sum_{i=1}^p \bar{\lambda}_i \mathbf{E}_i$  still makes the path potential decrease. The following proposition answers this question.

*Proposition 1:* for any  $\mu \in \mathbf{R}^p$  and any  $n \times p$  - matrix  $L$ , if  $\lambda^0 = -\mu$  and  $\bar{\lambda} = (I_p - L^+ L) \lambda^0$ , then

$$\mu^T \bar{\lambda} < 0$$

*Proof:*

$$\mu^T \bar{\lambda} = -\mu^T \mu + \mu^T L^+ L \mu$$

As  $LL^+L = L$ ,  $(L^+L)^2 = L^+L$ .  $L^+L$  is thus the matrix of a projection operator. As such,  $L^+L$  is a positive semi-definite matrix with eigenvalues 0 and 1. Therefore, for any vector  $\mu$ ,  $\mu^T L^+ L \mu \leq \mu^T \mu$  and  $\mu^T \bar{\lambda} \leq 0$ . ■

Proposition 1 ensures us that  $\eta(s, \tau_j) \triangleq \sum_{i=1}^p \bar{\lambda}_i \mathbf{E}_i(s, \tau_j)$  is an admissible direction of deformation that satisfies the boundary conditions  $\eta(0, \tau_j) = \eta(S, \tau_j) = 0$ .

### C.2 A better direction of deformation

Let us recall that Equation (5) is a first order approximation w.r.t.  $\tau$ . For this reason,  $\Delta \tau_j \|\eta\|_\infty$  with  $\|\eta\|_\infty \triangleq \max_{s \in [0, S]} \|\eta(s, \tau_j)\|$  needs to be small.  $\Delta \tau_j$  is thus chosen in such a way that  $\Delta \tau_j \|\eta\|_\infty$  is upper bounded by a positive given value  $\eta_{max}$ . The way the  $\lambda_i$ 's are chosen in (17) is not optimal in this respect. Indeed, the goal we aim at at each iteration is to make the path potential  $V$  decrease at most for constant  $\|\eta\|_\infty$ . Therefore the optimal value of  $\lambda$  realizes the following minimum:

$$\min_{\|\eta\|_\infty=1} \frac{dV}{d\tau}(\tau_j) = \min_{\|\sum_{i=1}^p \lambda_i \mathbf{E}_i\|_\infty=1} \sum_{i=1}^p \mu_i \lambda_i$$

This value of vector  $\lambda$  is very difficult to determine since  $\|\cdot\|_\infty$  is not an Euclidean norm. Instead, we compute

$$\min_{\|\sum_{i=1}^p \lambda_i \mathbf{E}_i\|_{L^2}=1} \sum_{i=1}^p \mu_i \lambda_i$$

which is a better approximation than (17).

The idea of the computation is to express  $\eta$  in a  $L^2$ -orthonormal basis in such a way that the above sum becomes the inner product between two vectors. Let us build from  $(\mathbf{E}_1, \dots, \mathbf{E}_p)$  an orthonormal basis  $(\mathbf{F}_1, \dots, \mathbf{F}_p)$  using Gramm Schmidt orthonormalization procedure. Let  $P$  be the corresponding change of coordinates  $p \times p$  - matrix (the  $j$ -th column of  $P$  is the vector of coordinates of  $\mathbf{F}_j$  expressed in  $(\mathbf{E}_1, \dots, \mathbf{E}_p)$ ). If we express  $\eta$  in  $(\mathbf{F}_1, \dots, \mathbf{F}_p)$  instead of  $(\mathbf{E}_1, \dots, \mathbf{E}_p)$ , Equation (10) becomes:

$$\eta(s, \tau_j) = \sum_{i=1}^p \lambda_i^\perp \mathbf{F}_i(s, \tau_j)$$

and Equation (16) becomes

$$\frac{dV}{d\tau}(\tau_j) = \sum_{i=1}^p \lambda_i^\perp \mu_i^\perp = (\mu^\perp | \eta)_{L^2} \quad (18)$$

with

$$\mu_i^\perp \triangleq \int_0^S \frac{\partial U}{\partial \mathbf{q}}(\mathbf{q}(s, \tau_j))^T \mathbf{F}_i(s, \tau_j) ds$$

and  $\mu^\perp = \sum_{i=1}^p \mu_i^\perp \mathbf{F}_i$ . The second equality in (18) holds since  $(\mathbf{F}_1, \dots, \mathbf{F}_p)$  is  $L^2$ -orthonormal. At equivalent  $L^2$ -norm,  $\eta = -\mu^\perp$  (i.e.  $\lambda_i^\perp = -\mu_i^\perp$ ) is the direction of deformation that minimizes  $\frac{dV}{d\tau}(\tau_j)$ .

In fact we do not evaluate functions  $\mathbf{F}_i$ 's, but only matrix  $P$ . The expression of  $\eta$  in basis  $(\mathbf{E}_1, \dots, \mathbf{E}_p)$  is given by vector

$$\lambda = P \lambda^\perp = P P^T \lambda^0 \quad (19)$$

Using expression of  $\eta$  in the orthonormal basis  $(\mathbf{F}_1, \dots, \mathbf{F}_p)$ , the expression in  $(\mathbf{E}_1, \dots, \mathbf{E}_p)$  of the orthogonal projection of the above  $\eta$  over the sub-space of vectors satisfying the boundary conditions (12) becomes

$$\bar{\lambda} = (I_p - P(LP)^+ L) P P^T \lambda^0$$

We have noticed during the development of our method, that choosing the optimal direction of deformation makes the method behave much better. It can be explained by the fact that this choice makes the path potential decrease faster and thus is more efficient to get away from obstacles.

### D. Nonholonomic constraint deviation

Approximation (5) induces a side effect: after a few iterations, the nonholonomic constraints are not satisfied anymore and the path becomes non admissible. We call this effect the *nonholonomic constraint deviation*. The goal of this section is to correct this deviation. If a path is not admissible, the velocity along this path is not contained in the linear subspace spanned by the  $k$  control vector fields and condition (1) does not hold.

### D.1 Extended dynamic system

To take into account this issue, we add for each configuration  $\mathbf{q}$ ,  $n-k$  vector fields  $\mathbf{X}_{k+1}(\mathbf{q}), \dots, \mathbf{X}_n(\mathbf{q})$  to the  $k$  control vector fields of the system in such a way that  $\mathbf{X}_1(\mathbf{q}), \dots, \mathbf{X}_n(\mathbf{q})$  span  $\mathbf{R}^n$ . We define the *extended system* as the system controlled by all these vector fields:

$$\mathbf{q}' = \sum_{i=1}^n u_i \mathbf{X}_i(\mathbf{q}) \quad (20)$$

System (20) is not subject to any kinematic constraint. A path  $\mathbf{q}(s)$  of system (20) is admissible for system (1) if and only if for any  $j \in \{k+1, \dots, n\}$  and any  $s \in [0, S]$ ,  $u_j(s) = 0$ .

In Section II, we deformed a given path, admissible for (1) by perturbing the input functions  $u_1(s, \tau), \dots, u_k(s, \tau)$  of this path in order to avoid obstacles. In this section, we consider an initial path not necessarily admissible and we compute input perturbations that make  $u_{k+1}(s, \tau), \dots, u_n(s, \tau)$  uniformly tend toward 0 as  $\tau$  grows.

From now on, we denote by  $\bar{\mathbf{u}}(s, \tau) = (u_1(s, \tau), \dots, u_n(s, \tau))$  the input function of system (20) and by  $\bar{\mathbf{v}}(s, \tau) = (v_1(s, \tau), \dots, v_n(s, \tau))$  the perturbation of these input functions:

$$\forall i \in \{1, \dots, n\}, \quad v_i(s, \tau) = \frac{\partial u_i}{\partial \tau}(s, \tau)$$

The relation between the input perturbation  $\bar{\mathbf{v}}(s, \tau)$  and the direction of deformation  $\eta(s, \tau)$  is similar as in Section II:

$$\eta'(s, \tau) = \bar{A}(s, \tau)\eta(s, \tau) + \bar{B}(s, \tau)\bar{\mathbf{v}}(s, \tau) \quad (21)$$

but now,  $\bar{A}(s, \tau)$  and  $\bar{B}(s, \tau)$  are both  $n \times n$  matrices:

$$\bar{A} = \sum_{i=1}^n u_i \frac{\partial \mathbf{X}_i}{\partial \mathbf{q}}(\mathbf{q}) \quad \text{and} \quad \bar{B} = (B \ B^\perp) \quad (22)$$

where  $B^\perp = (\mathbf{X}_{k+1}(\mathbf{q}) \dots \mathbf{X}_n(\mathbf{q}))$  is the matrix the columns of which are the additional vector fields. With this notation, (21) can be rewritten as follows:

$$\eta'(s, \tau) = \bar{A}(s, \tau)\eta(s, \tau) + B(s, \tau)\mathbf{v}(s, \tau) + B^\perp(s, \tau)\mathbf{v}^\perp(s, \tau) \quad (23)$$

where  $\mathbf{v}^\perp(s, \tau) = (v_{k+1}(s, \tau), \dots, v_n(s, \tau))$ .

### D.2 Correction of nonholonomic deviation

$v_{k+1}(s, \tau), \dots, v_n(s, \tau)$  represent the derivative of the input functions  $u_{k+1}(s, \tau), \dots, u_n(s, \tau)$  w.r.t.  $\tau$ . We want these input functions to converge toward 0 as  $\tau$  grows, we thus set:

$$\forall i \in \{k+1, \dots, n\}, \quad v_i(s, \tau) = -\alpha u_i(s, \tau)$$

where  $\alpha$  is a positive constant number. This implies that, as  $\tau$  grows,

$$\forall s \in [0, S], u_i(s, \tau) = e^{-\alpha\tau} u_i(s, 0)$$

and therefore the input variables along the additional vector fields converge toward 0 exponentially. We denote by  $\eta_1$  the corresponding direction of deformation for  $\tau = \tau_j$ :

$$\begin{aligned} \eta_1'(s, \tau_j) &= \bar{A}(s, \tau_j)\eta_1(s, \tau_j) + B^\perp(s, \tau_j)\mathbf{v}^\perp(s, \tau_j) \\ \eta_1(0, \tau_j) &= 0 \end{aligned} \quad (24) \quad (25)$$

### D.3 Deformation due to obstacles

Following the procedure described in sections III-A and III-C, we restrict input functions  $(v_1, \dots, v_k)$  to the finite dimensional subspace of functions spanned by  $(\mathbf{e}_1, \dots, \mathbf{e}_p)$  and we compute  $\lambda = (\lambda_1, \dots, \lambda_p)$  according to Equation (19). We denote by  $\eta_2$  the direction of deformation corresponding to these coefficients:

$$\eta_2(s, \tau_j) = \sum_{i=1}^p \lambda_i \mathbf{E}_i(s, \tau_j)$$

where now the  $\mathbf{E}_i$ 's are solutions of system (21):

$$\mathbf{E}_i'(s, \tau_j) = \bar{A}(s, \tau_j)\mathbf{E}_i(s, \tau_j) + B(s, \tau_j)\mathbf{e}_i(s) \quad (26)$$

$$\mathbf{E}_i(0, \tau_j) = 0 \quad (27)$$

### D.4 Boundary conditions

As system (23) is linear, the deformation obtained by summing input perturbations  $\mathbf{v}$  and  $\mathbf{v}^\perp$  is the sum of  $\eta_1$  and  $\eta_2$ . Again we want this direction of deformation not to modify the initial and final configurations. Conditions (11) and (12) become:

$$\begin{aligned} \eta_1(0, \tau_j) + \eta_2(0, \tau_j) &= 0 \\ \eta_1(S, \tau_j) + \eta_2(S, \tau_j) &= 0 \end{aligned}$$

The first condition is always satisfied. The second one can be rewritten as an affine system of equations over parameters  $(\lambda_1, \dots, \lambda_p)$ :

$$\eta_2(S, \tau_j) = L\lambda = -\eta_1(S, \tau_j) \quad (28)$$

where  $L$  is the matrix defined in Section III-B. As in Section III-B, the vector  $\lambda$  obtained by Equation (19) does not satisfy (28). We orthogonally project this vector over the affine space of solutions of system (28) and we get

$$\bar{\lambda} = -P(LP)^+ \eta_1(S, \tau) + (I_p - P(LP)^+ L)\lambda$$

The direction of deformation:

$$\eta(s, \tau_j) = \sum_{i=1}^p \bar{\lambda}_i \mathbf{E}_i(s, \tau_j) + \eta_1(s, \tau_j)$$

satisfies the boundary conditions (11-12) and makes the components  $v_{k+1}, \dots, v_n$  of the velocity along additional vector fields decrease. Moreover, if these components are sufficiently close to 0, the above expression of  $\eta$  makes the potential due to obstacles decrease.

### E. Deformation algorithm: summary

Table I summarizes the path deformation algorithm for non-holonomic systems. The input of the algorithm is an initial path  $\mathbf{q}(s, 0)$ , with input function  $\mathbf{u}(s, 0)$ . At each step  $\tau = \tau_j$ , a direction of deformation  $\eta(s, \tau_j)$  is computed along the current path. This direction of deformation is applied to the path and  $\tau$  is updated.

**Algorithm : Path deformation for nonholonomic systems**

```

/* current path = initial path */
j = 0,  $\tau_j = 0$ 

while  $\mathbf{q}(s, \tau_j)$  in collision {
  compute  $\bar{A}(s, \tau_j)$  and  $\bar{B}(s, \tau_j)$  for  $s \in [0, S]$ 

  /* correction of nonholonomic deviation */
  for  $i$  in  $\{k+1, \dots, n\}$  {
    compute  $u_i(s, \tau_j)$ 
    compute  $v_i(s, \tau_j) = -\alpha u_i(s, \tau_j)$  for  $s \in [0, S]$ 
    compute  $\eta_1(s, \tau_j)$  using (24)

  /* potential gradient in configuration space */
  for  $i$  in  $\{1, \dots, p\}$  {
    compute  $\mathbf{E}_i(s, \tau_j)$  by integrating (26)
  }
  compute  $\frac{\partial U}{\partial \mathbf{q}}(\mathbf{q}(s, \tau_j))$  for  $s \in [0, S]$ 
  for  $i$  in  $\{1, \dots, p\}$  {
    compute  $\lambda_i^0 = -\int_0^S \frac{\partial U}{\partial \mathbf{q}}(\mathbf{q}(s, \tau_j))^T \mathbf{E}_i(s, \tau_j) ds$ 
  }

  /* orthonormalization of  $(\mathbf{E}_1, \dots, \mathbf{E}_p)$  */
  compute matrix  $P$  using Gramm Schmidt procedure

  /* projection of  $\lambda$  over boundary conditions */
  compute  $\bar{\lambda} = -P(LP)^+ \eta_1(S, \tau_j) + (I_p - P(LP)^+ L) P P^T \lambda^0$ 

  /* compute and apply deformation */
  compute  $\eta(s, \tau_j) = \eta_1(s, \tau_j) + \sum_{i=1}^p \bar{\lambda}_i \mathbf{E}_i(s, \tau_j)$  for  $s \in [0, S]$ 
  if  $\|\eta\|_\infty > \eta_{max}$  then
     $\Delta \tau_j = \eta_{max} / \|\eta\|_\infty$ 
  else
     $\Delta \tau_j = 1$ 
   $\tau_{j+1} \leftarrow \tau_j + \Delta \tau_j$ 
   $\mathbf{q}(s, \tau_{j+1}) \leftarrow \mathbf{q}(s, \tau_j) + \Delta \tau_j \eta(s, \tau_j)$  for  $s \in [0, S]$ 
   $j \leftarrow j + 1$ 
}

```

TABLE I

PATH DEFORMATION ALGORITHM FOR NONHOLONOMIC SYSTEMS

**IV. APPLICATION TO DIFFERENT KINEMATIC SYSTEMS**

We have applied the path deformation method for nonholonomic systems described in the former sections to two different systems. In both cases, the application is a navigation task for two different types of nonholonomic mobile robots : a mobile robot towing a trailer and a car-like mobile robot in a cluttered and partially known environment.

The method has also been applied in a path optimization problem. This latter application is described in [12], [13]. The problem raised was to validate the itinerary of trucks carrying huge components of an airplane through villages in the southwest of France.

**A. Reactive obstacle avoidance for a mobile robot towing a trailer**

In this application, the robot is the unicycle Hilare 2 towing a trailer, as shown on Figure 1. During motion, a laser scanner detects obstacles in front of the robot (behind if the robot moves backward) and maps these obstacles in the global frame of the environment. If a collision is detected between the current path and obstacles, an interval of deformation centered on the first configuration in collision of the path and not containing

the current configuration of the robot is chosen. The deformation algorithm is then applied to this interval until the collision has disappeared. If necessary, the robot stops before reaching the interval of deformation. Of course, the interval of deformation changes when the robot moves ahead and discovers new collisions. [2] describes with more precision how the path deformation and the path following tasks operate simultaneously. Figure 6 gives an example of path simultaneously deformed and followed by the robot.

In the following paragraphs, we give expressions of the linearized system, of the input perturbation functions  $\mathbf{e}_i$ 's and we give more details about the potential field generated by obstacles. We provide also a few experimental results. We invite the reader to see more experimental results on the web-page of the demo [28].

**A.1 Linearized system**

A configuration of our system is represented by a vector  $\mathbf{q} = (x, y, \theta, \varphi)$  where  $(x, y)$  and  $\theta$  are respectively the position and orientation of Hilare 2 and  $\varphi$  is the angle between the robot and the trailer. The control vector fields of this system are

$$\mathbf{X}_1 = \begin{pmatrix} \cos \theta \\ \sin \theta \\ 0 \\ -\frac{1}{l_t} \sin \varphi \end{pmatrix} \quad \mathbf{X}_2 = \begin{pmatrix} 0 \\ 0 \\ 1 \\ -1 - \frac{l_r}{l_t} \cos \varphi \end{pmatrix}$$

where  $l_r$  (resp.  $l_t$ ) is the distance between the center of the robot (resp. the trailer) and the trailer connection. The input functions of the system are  $u_1$  and  $u_2$  the linear and angular velocities of the robot. To get a basis of  $\mathbf{R}^4$  at each configuration  $\mathbf{q}$ , we define two additional vector fields:

$$\mathbf{X}_3 = \begin{pmatrix} -\sin \theta \\ \cos \theta \\ 0 \\ 0 \end{pmatrix} \quad \mathbf{X}_4 = \begin{pmatrix} -\sin(\theta + \varphi) \\ \cos(\theta + \varphi) \\ -l_t - l_r \cos \varphi \\ -l_t \end{pmatrix}$$

Let us be given a current path, not necessarily admissible,  $\mathbf{q}(s) = (x(s), y(s), \theta(s), \varphi(s))$  defined by input functions  $u_1(s), u_2(s), u_3(s), u_4(s)$ :

$$\mathbf{q}'(s) = \sum_{i=1}^4 u_i(s) \mathbf{X}_i(\mathbf{q}(s))$$

For this path, matrices  $\bar{A}(s)$  and  $\bar{B}(s)$  defining System (21) have the following expressions:

$$\bar{A}(s) = \begin{pmatrix} 0 & 0 & -u_1 s \theta - u_3 c \theta - u_4 c \psi & -u_4 c \psi \\ 0 & 0 & u_1 c \theta - u_3 s \theta - u_4 s \psi & -u_4 s \psi \\ 0 & 0 & 0 & u_4 l_r s \varphi \\ 0 & 0 & 0 & \frac{-u_1 c \varphi + u_2 l_r s \varphi}{l_t} \end{pmatrix}$$

$$\bar{B}(s) = \begin{pmatrix} c \theta & 0 & -s \theta & -s \psi \\ s \theta & 0 & c \theta & c \psi \\ 0 & 1 & 0 & -l_t - l_r c \varphi \\ -\frac{1}{l_t} s \varphi & -1 - \frac{l_r}{l_t} c \varphi & 0 & -l_t \end{pmatrix}$$

where to make notation shorter,  $c \theta = \cos \theta$ ,  $s \theta = \sin \theta$ ,  $c \varphi = \cos \varphi$ ,  $s \varphi = \sin \varphi$ ,  $c \psi = \cos(\theta + \varphi)$ ,  $s \psi = \sin(\theta + \varphi)$ .



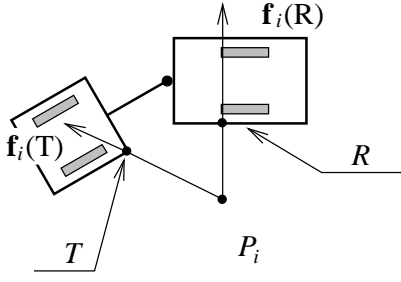


Fig. 5. Configuration space potential field generated by an obstacle point  $P_i$ .

## A.2 Subspace of input perturbations

The input space is of dimension 2. We have chosen functions  $\mathbf{e}_i$ 's in such a way that they span the sub-space of truncated Fourier series:

$$\begin{aligned} \mathbf{e}_1(s) &= (1, 0)^T & \mathbf{e}_2(s) &= (0, 1)^T \\ \mathbf{e}_3(s) &= (\cos(\frac{2\pi s}{S}), 0)^T & \mathbf{e}_4(s) &= (0, \cos(\frac{2\pi s}{S}))^T \\ \mathbf{e}_5(s) &= (\sin(\frac{2\pi s}{S}), 0)^T & \mathbf{e}_6(s) &= (0, \sin(\frac{2\pi s}{S}))^T \\ &\vdots & & \\ \mathbf{e}_{4q-1}(s) &= (\cos(\frac{2q\pi s}{S}), 0)^T & \mathbf{e}_{4q}(s) &= (0, \cos(\frac{2q\pi s}{S}))^T \\ \mathbf{e}_{4q+1}(s) &= (\sin(\frac{2q\pi s}{S}), 0)^T & \mathbf{e}_{4q+2}(s) &= (0, \sin(\frac{2q\pi s}{S}))^T \end{aligned}$$

where  $q$  is the maximal order of the truncated Fourier series and  $p = 4q + 2$ . The main advantage of this basis is that a small value of  $q$  produces smooth path deformation suitable for avoiding a big obstacle in the way of the robot, whereas a bigger value of  $q$  is more efficient in highly cluttered environment like corridors.

## A.3 Obstacle potential field

During motion, obstacles are detected by the laser scanners. Each sensor scans an horizontal plane and returns at most 360 points.

If  $P_i$  is an obstacle point, we denote by  $\nu_i(M)$  the potential field in the plane generated by  $P_i$ , where  $M$  is a point in the plane and  $d$  is the distance between  $M$  and  $P_i$ :

$$\begin{aligned} \nu_i(M) &= \frac{1}{d+d_0} + \frac{d}{(d_1+d_0)^2} & \text{if } 0 \leq d \leq d_1 \\ \nu_i(M) &= \frac{1}{d_1+d_0} + \frac{d_1}{(d_1+d_0)^2} & \text{if } d > d_1 \end{aligned}$$

$d_0 < d_1$  are constant distances. Let  $\mathbf{f}_i(M) = -\nabla \nu_i(M)$  be the force in the plane deriving from this potential. The norm of this force field w.r.t. the distance to  $P_i$  is:

$$\begin{aligned} \|\mathbf{f}_i(M)\| &= \frac{1}{(d+d_0)^2} - \frac{1}{(d_1+d_0)^2} & \text{if } 0 \leq d \leq d_1 \\ \|\mathbf{f}_i(M)\| &= 0 & \text{if } d > d_1 \end{aligned} \quad (29)$$

Let us notice that each obstacle point generates a force up to distance  $d_1$  in the plane. Let  $R(\mathbf{q})$  and  $T(\mathbf{q})$  be the closest points to  $P_i$  on the robot and on the trailer. The configuration space potential field implied by  $P_i$  is defined by evaluating the plane potential field at  $R(\mathbf{q})$  and  $T(\mathbf{q})$  (Figure 5):

$$U_i(\mathbf{q}) = \nu_i(R(\mathbf{q})) + \nu_i(T(\mathbf{q})) \quad (30)$$

If  $P_i$  is inside the robot or inside the trailer the corresponding term in  $U_i$  is set to 0.

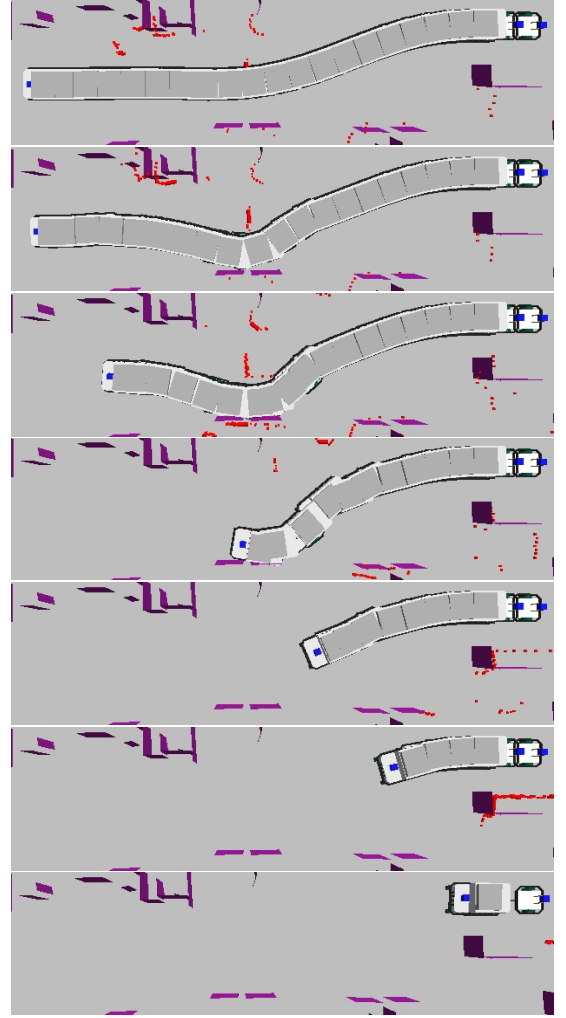


Fig. 6. A backward path computed and executed by the mobile robot Hilare 2 towing a trailer. Red dots are obstacles detected by a laser scanners (in blue) mounted on the trailer. An unexpected box lies on the path planned by the robot. The robot deforms the path while moving and reaches the goal. Let us notice that the interval of deformation changes each time a new collision is detected. For instance between snapshot 2 and 3, the first configuration in collision has changed, so has the interval of deformation.

The configuration space potential field is defined as the sum of the potential fields relative to each obstacle point:

$$U(\mathbf{q}) = \sum_i U_i(\mathbf{q})$$

The gradient of the potential field is obtained by differentiating (30) w.r.t. the configuration variables  $(x, y, \theta, \varphi)$ .

$$\begin{aligned} \frac{\partial U_i}{\partial \mathbf{q}}(\mathbf{q}) &= \nabla \nu_i(R(\mathbf{q})) \frac{\partial R}{\partial \mathbf{q}}(\mathbf{q}) + \nabla \nu_i(T(\mathbf{q})) \frac{\partial T}{\partial \mathbf{q}}(\mathbf{q}) \\ &= -\mathbf{f}_i(R) \frac{\partial R}{\partial \mathbf{q}} - \mathbf{f}_i(T) \frac{\partial T}{\partial \mathbf{q}} \end{aligned}$$

## A.4 Experimental Results

Figure 6 gives a typical example of application of the path deformation method to Hilare 2 towing a trailer. In this example, a path is computed by a motion planner given a map of the environment. An obstacle not represented in this map lies on the

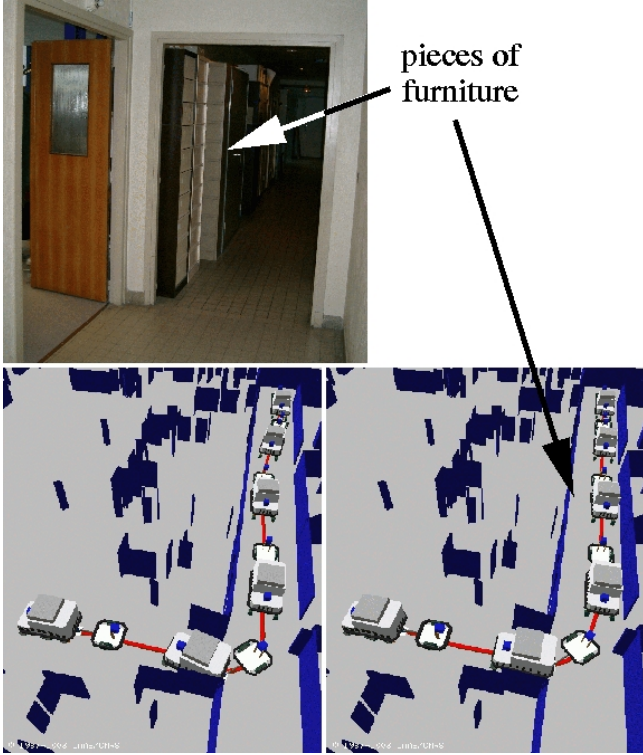


Fig. 7. Due to renovation, pieces of furniture were placed in the corridors of LAAS-CNRS (top). We planned a path without taking into account these new obstacles (bottom left) and we ran our path deformation method on the mobile robot Hilare 2 towing a trailer. The method was successful and produced a collision free path (bottom right).

path computed by the planner. This obstacle is detected and the robot deforms the path while following the collision-free part of it. Let us notice that due to localization errors, the obstacles detected by the sensor are slightly different from the obstacles in the map. The path deformation method enables the robot to achieve the navigation task even with poor localization.

During summer 2003, the robotics building of LAAS-CNRS was renovated. During this period, pieces of furniture were placed in the corridors. We took advantage of these real conditions to test our path deformation method. We planned a path without taking into account the new obstacles and we asked our robot to adapt and follow this path. Figure 7 displays the result of this experiment.

### B. Reactive obstacle avoidance for a car-like mobile robot

We have applied the path deformation method to the mobile robot Dala, an ATRV displayed on Figure 8 in an outdoor environment. This mobile robot is a differential-driven robot: rotation is performed by applying different velocities to the right and left wheels. To avoid too much slipping, we consider Dala as a car-like robot with a bounded virtual steering angle. A configuration of the robot is thus represented by a vector  $\mathbf{q} = (x, y, \theta, \varphi)$  where  $(x, y)$  and  $\theta$  are respectively the position and orientation of Dala and  $\varphi$  the virtual steering angle. The input perturbation functions  $e_i$ 's are similar to the Hilare-trailer case in Section IV-A.2. The control vector fields of Dala are the

following:

$$\mathbf{X}_1 = \begin{pmatrix} \cos \theta \\ \sin \theta \\ \frac{\tan \varphi}{l} \\ 0 \end{pmatrix} \quad \mathbf{X}_2 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

and the additional vector fields are:

$$\mathbf{X}_3 = \begin{pmatrix} -\sin \theta \\ \cos \theta \\ 0 \\ 0 \end{pmatrix} \quad \mathbf{X}_4 = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}$$

We do not give expressions of matrices  $\bar{A}$  and  $\bar{B}$  they can be easily deduced from the above vector fields.

The obstacle potential field computation is also similar, except that we now consider the steering angle bounds as a possible obstacle from which the robot must move away. A force is thus computed as a function of the steering angle, in a similar way as Equation (29). The value of the force increases when the angle gets closer to the bounds and prevents the steering angle to go beyond these bounds.

### B.1 Experimental results

Figure 9 presents an example of initial path in collision and iteratively modified by our algorithm until collisions have disappeared. Figure 10 displays the input functions corresponding to each iteration of the deformation process displayed in Figure 9.

## V. CONCLUSION AND FUTURE WORK

In this paper, we have described a novel and generic approach to path deformation for nonholonomic systems. This approach has been applied to the problem of mobile robot navigation but is suitable for other path optimization problems related to non-holonomic systems.

In a future work, we are going to work on the extension of this method to systems with drift. This natural extension of the method will enable us to take into account bounds on the velocity of the system by including the velocities in the state-space of the acceleration-controlled corresponding system.

**Acknowledgment:** This work has been partially supported by the European Project MOVIE (IST-2001-39250) and by the CNRS interdisciplinary program ROBEA.

## REFERENCES

- [1] J. Barraquand and J.-C. Latombe. Robot motion planning: A distributed representation approach. *International Journal on Robotics Research*, 10(6):628–649, June 1991.
- [2] D. Bonnafoos and F. Lamiroux. Sensor based trajectory following for non-holonomic systems in highly cluttered environment. In *International Conference on Intelligent Robots and Systems*, pages 892–897, Las Vegas, NE, October 2003. IEEE/RSJ.
- [3] O. Brock and O. Khatib. Real-time replanning in high dimensional configuration spaces using sets of homotopic paths. In *International Conference on Robotics and Automation*, pages 550–555, San Francisco, CA, April 2000. IEEE.
- [4] F. Bullo and K. Lynch. Kinematic controllability for decoupled trajectory planning in underactuated mechanical systems. *IEEE Transactions on Robotics and Automation*, 17(4):402–412, August 2001.
- [5] S. Chitta and J. Ostrowski. Motion planning for heterogeneous modular mobile systems. In *International Conference on Robotics and Automation*, pages 4077–4082, Washington D.C., May 2002. IEEE.

- [6] A. Divelbiss and J. Wen. A path space approach to nonholonomic motion planning in the presence of obstacles. *IEEE Transactions on Robotics and Automation*, 13(3):443–451, June 1997.
- [7] C. Fernandes, L. Gurvits, and Z. Li. *Nonholonomic Motion Planning*, chapter Optimal Nonholonomic Motion Planning for a Falling Cat, pages 379–421. Kluwer Academic Press, 1993.
- [8] P. Fiorini and Z. Shiller. Motion planning in dynamic environments using velocity obstacles. *International Journal on Robotics Research*, 17(7):760–772, July 1998.
- [9] M. Fliess, J. Lévine, P. Martin, and P. Rouchon. Design of trajectory stabilizing feedback for driftless flat systems. In *European Control Conference*, Roma, Italy, 1995.
- [10] M. Khatib, H. Jaouni, R. Chatila, and J.-P. Laumond. Dynamic path modification for car-like nonholonomic mobile robots. In *International Conference on Robotics and Automation*, pages 2920–2925, Albuquerque, NM, April 1997. IEEE.
- [11] O. Khatib. Real-time obstacle avoidance for manipulators and mobile robots. *International Journal on Robotics Research*, 5(1):90–98, Spring 1986.
- [12] F. Lamiriaux, D. Bonnafoous, and C. Van Geem. *Control Problems in Robotics*, chapter Path Optimization for Nonholonomic Systems: Application to Reactive Obstacle Avoidance and Path Planning, pages 1–18. Springer, 2002.
- [13] F. Lamiriaux, J.-P. Laumond, C. VanGeem, D. Boutonnet, and G. Raust. Trailer-truck trajectory optimization for airbus a380 component transportation. *IEEE Robotics and Automation Magazine*, (Robotic Technologies applied to Intelligent Transportation Systems), 2004. to appear.
- [14] F. Lamiriaux, S. Sekhavat, and J.-P. Laumond. Motion planning and control for hilare pulling a trailer. *IEEE Transactions on Robotics and Automation*, 15(4):640–652, August 1999.
- [15] F. Large, S. Sekhavat, Z. Shiller, and C. Laugier. Toward real-time global motion planning in a dynamic environment using the nlvo concept. In *International Conference on Intelligent Robots and Systems*, pages 607–612, Lausanne, Switzerland, October 2002. IEEE/RSJ.
- [16] J.-P. Laumond. Controllability of a multibody mobile robot. *IEEE Transactions on Robotics and Automation*, 9(6):755–763, June 1993.
- [17] J.-P. Laumond, S. Sekhavat, and F. Lamiriaux. *Robot Motion Planning and Control*, chapter Guidelines in Nonholonomic Motion Planning for Mobile Robots, pages 2–53. Lecture Notes in Control and Information Science. Springer, NY, 1998.
- [18] S. M. LaValle and J. J. Kuffner. Randomized kinodynamic planning. *International Journal of Robotics Research*, 20(5):378–400, May 2001.
- [19] A. De Luca, G. Oriolo, and C. Samson. *Robot Motion Planning and Control*, chapter Feedback Control of a Nonholonomic Car-like Robot, pages 171–253. Lecture Notes in Control and Information Sciences, Springer, NY, 1998.
- [20] R. M. Murray and S. Sastry. *Nonholonomic Motion Planning*, chapter Steering nonholonomic control systems using sinusoids, pages 23–52. Kluwer Academic Press, 1993.
- [21] S. Quinlan. *Real-Time Modification of Collision Free Paths*. PhD thesis, Stanford University, 1995.
- [22] E. Rimon and D. Koditschek. Exact robot navigation using artificial potential functions. *IEEE Tr. on Rob. and Autom.*, 8:501–518, 1992.
- [23] C. Samson. Control of chained systems. application to path following and time-varying point-stabilization. *IEEE Transactions on Robotics and Automation*, 40(1):64–77, 1995.
- [24] T. Siméon, J.P. Laumond, C. Van Geem, and J. Cortés. Computer aided motion: Move3d within molog. In *International Conference on Robotics and Automation*, Seoul KR, May 2001. IEEE.
- [25] O. Sjørdalen and O. Egeland. Exponential stabilization of nonholonomic chained systems. *IEEE Transactions on Automatic Control*, 40(1):35–49, Jan 1995.
- [26] P. Svestka and M. Overmars. Coordinated motion planning for multiple car-like robots using probabilistic roadmaps. In *International Conference on Robotics and Automation*, pages 1631–1636, Nagoya, 1995. IEEE.
- [27] P. Svestka and M. Overmars. Probabilistic path planning. In J.-P. Laumond, editor, *Robot Motion Planning and Control*, pages 255–304. Lecture Notes in Control and Information Sciences, Springer, NY, 1998.
- [28] <http://www.laas.fr/RIA/RIA-demo-hilare-trailer.html>

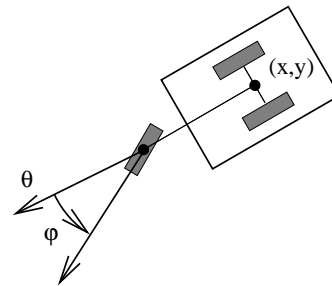


Fig. 8. Mobile robot ATRV Dala as a car-like robot: a nonholonomic system of dimension 4 with 2 nonholonomic constraints.

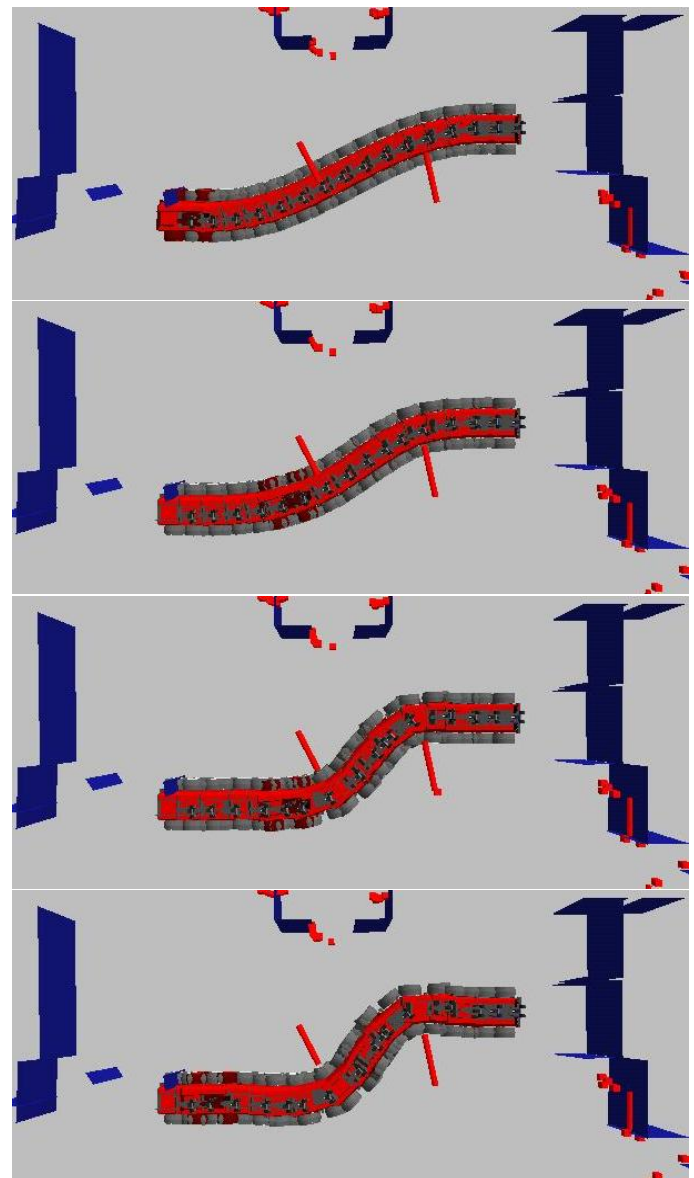


Fig. 9. Two obstacles lie on the path planned by the robot Dala (top). The path is iteratively deformed (intermediate) until the path is collision-free (bottom).

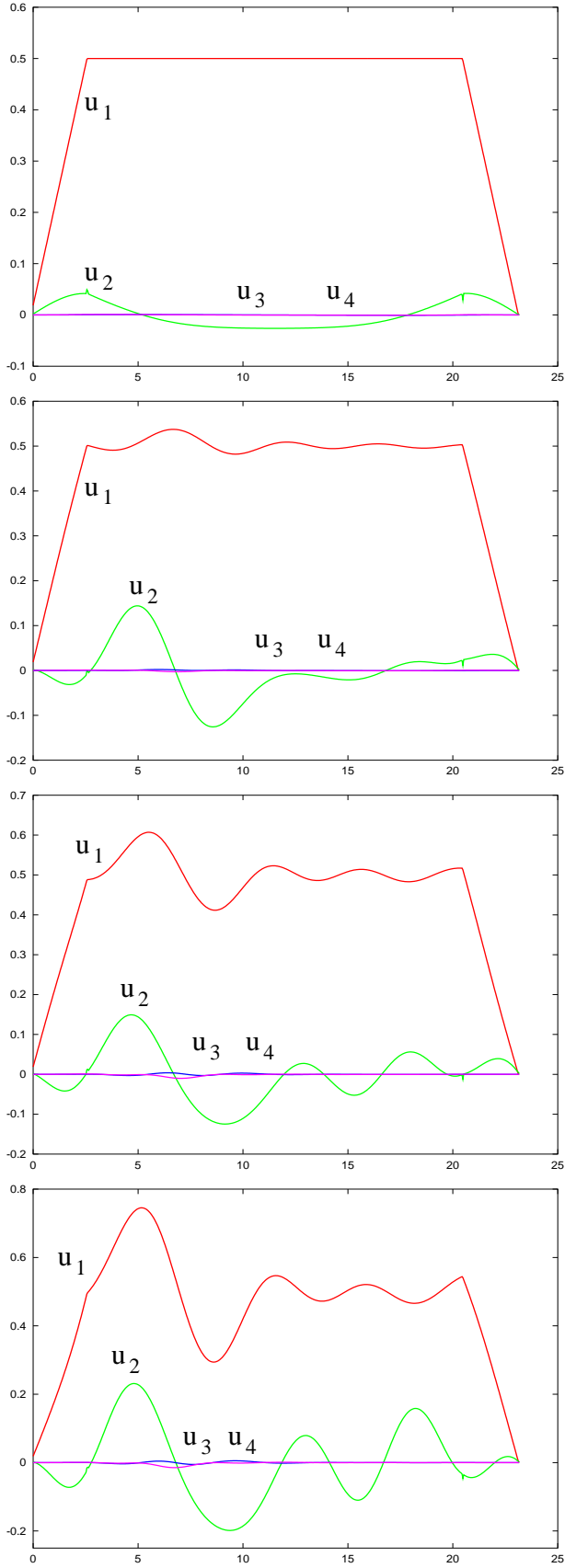


Fig. 10. Input functions  $u_i(s)$  corresponding respectively to the initial path and to several steps of the deformation process.  $u_1$  and  $u_2$  are perturbed in order to avoid the obstacles. We notice that input functions  $u_3$  and  $u_4$  along the additional vector fields remain very close to 0.