

Reactive Trajectory Deformation for Nonholonomic Systems: Application to Mobile Robots

Florent Lamiriaux and David Bonnafoous
florent@laas.fr dbonnafo@laas.fr
LAAS-CNRS, 7 avenue du Colonel Roche
31077 Toulouse Cedex 4, France

Abstract

In this paper, we propose a novel approach to reactive obstacle avoidance for nonholonomic systems. The method is based on the deformation of an initial trajectory computed by a motion planner. The deformation we perform keeps the nonholonomic constraints of the system satisfied. The deformation algorithm is based on a potential field generated by obstacles. We applied this approach to the mobile robot Hilare 2 towing a trailer and we carried out some experiments.

1 Introduction

Executing a motion computed by a motion planner for a mobile robot is a difficult task, mainly for two reasons. The first one is that the environment is almost never static and exactly modeled. Thus the path computed by the planner might be in collision and impossible to follow. Moreover, moving obstacles, like people walking around, may interfere with the robot path. The second reason is that the imprecision of localization must be smaller than the distance between the planned path and the obstacles. Such a precise localization is not always possible. For simple mobile robots, some approaches have been proposed to overcome these difficulties. The first works in this direction [6, 4] use potential fields attracting the robot toward the goal and applying repulsive forces in the neighborhood of obstacles. The main drawback of these methods is the existence of local minima of the potential field that attract the robot. [7] then propose to deform online the path to be followed in order to avoid unexpected obstacles. This approach is commonly named the elastic band algorithm. It was extended to the case of a simple nonholonomic mobile robot [5]. The dynamic window approach [3, 1] chooses in the space of reachable velocities

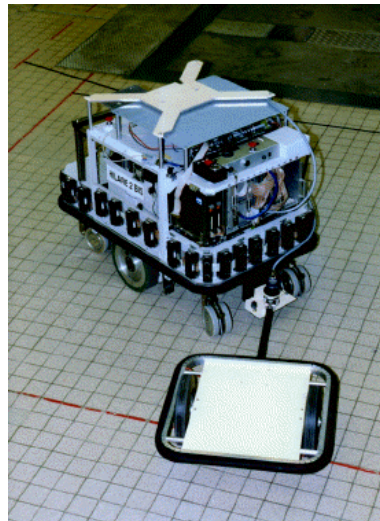


Figure 1: A nonholonomic systems of dimension 4: Hilare and its trailer.

those that enable the robot to move toward the goal without colliding obstacles. Most of these methods make the assumption (reasonable for cylinder-shaped robots) that the configuration space of the robot is of dimension 2 and that the system is subjected to at most one simple nonholonomic constraint. For more complicated systems like the robot towing a trailer shown in Figure 1, the previous approaches cannot be applied. The full dimension of the configuration space has to be considered. The goal of this paper is to propose a reactive method to deform a feasible path for a nonholonomic system in such a way that, first this path gets away from unexpected obstacles and second the deformed path remains feasible. Our method is generic and inspired by the elastic band approach. It is based on a force fields applied by obstacles that produce a deformation in such a way that the nonholonomic constraints are always satisfied.

The paper is organized as follows. In section 2, we

define the set of deformations of a path that does not affect the nonholonomic constraints of a system. Then we show how to choose among this set a deformation that makes the path get away from obstacles. Our deformation process is numerical and iterative. For this reason, after a few steps, the nonholonomic constraints of the path are not satisfied anymore. We explain how to correct this nonholonomic deviation. In section 3, we apply our method to the mobile robot Hilare towing a trailer. In section 4, we give some experimental results. paper is concluded by some comments in Section 5.

2 Nonholonomic Systems and Path Deformation

The elastic band algorithm in [7] is based on an potential function over the set of possible paths. This potential function is the sum of a term that increases when the path gets closer to obstacles and a term that increases with the length of the path. The path is then deformed incrementally. A deformation of the path is defined by a function η that maps to each configuration of the path a direction of deformation $\eta(s)$ (Figure 2). At each step the direction of deformation that makes decrease the potential value faster is chosen. This direction is given by the potential gradient. If the system is nonholonomic, however, this method does not work, since following the gradient of the potential does not guarantee that the nonholonomic constraints will still be satisfied after deformation. The main point of our method is to choose deformations that keep the nonholonomic constraints satisfied. In the rest of this section, we describe the different computations of one step of our iterative algorithm.

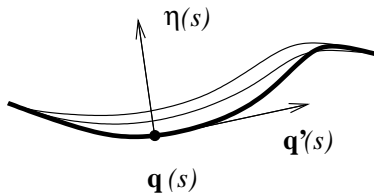


Figure 2: Current path $\mathbf{q}(s)$ (in bold) and a deformation $\eta(s)$ along this path.

2.1 Nonholonomic Systems

A nonholonomic system of dimension n is characterized by a set of $k < n$ vector fields $X_1(\mathbf{q}), \dots, X_k(\mathbf{q})$, 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 $X_i(\mathbf{q})$'s. Let us define $n - k$ additional vector fields $X_{k+1}(\mathbf{q}), \dots, X_n(\mathbf{q})$ in such a way that (X_1, \dots, X_n) is a basis of \mathbf{R}^n at each configuration. Equivalently, a path $\mathbf{q}(s)$ defined over an interval $[0, S]$ is a feasible path if and only if

$$\forall s \in [0, S] \quad \mathbf{q}'(s) = \sum_{i=1}^n u_i(s) X_i(\mathbf{q}) \quad \text{and} \quad (1)$$

$$u_i(s) = 0 \quad \text{for} \quad k+1 \leq i \leq n \quad (2)$$

where $\mathbf{q}'(s)$ is the derivative of $\mathbf{q}(s)$. We will explain in Section 2.5 why we introduced these additional vector fields. Up to this point, we invite the reader to forget these vector fields eventhough they appear in the following equations multiplied by zero functions $u_{k+1}(s), \dots, u_n(s)$.

2.2 Path Deformation

To deform a given path we only need to deform the input functions $u_1(s), \dots, u_k(s)$ of the initial path $\mathbf{q}(s)$. For that, we define n real functions $v_1(s), \dots, v_n(s)$, a real number h and we denote by $\mathbf{q}(s, h)$ the path obtained by plugging $u_i(s) + hv_i(s)$ as input to system (1). (Again, we consider that $v_i(s) = 0$ for $k+1 \leq i \leq n$). As a result,

$$\frac{\partial \mathbf{q}}{\partial s}(s, h) = \sum_{i=1}^n (u_i(s) + h v_i(s)) X_i(\mathbf{q}(s, h))$$

Let us differentiate this equation w.r.t. h :

$$\begin{aligned} \frac{\partial^2 \mathbf{q}}{\partial s \partial h}(s, h) &= \sum_{i=1}^n v_i(s) X_i(\mathbf{q}(s, h)) \\ &+ (u_i(s) + h v_i(s)) \frac{\partial X_i}{\partial \mathbf{q}}(\mathbf{q}(s, h)) \frac{\partial \mathbf{q}}{\partial h}(s, h) \end{aligned}$$

If we denote now by $\eta(s) = \frac{\partial \mathbf{q}}{\partial h}(s, 0)$ and by $\eta'(s) = \frac{\partial \eta}{\partial s}(s)$, the above equation becomes for $h = 0$:

$$\begin{aligned} \eta'(s) &= \sum_{i=1}^n v_i(s) X_i(\mathbf{q}(s)) + u_i(s) \frac{\partial X_i}{\partial \mathbf{q}}(\mathbf{q}(s)) \eta(s) \quad (3) \\ &= A(s) \eta(s) + B(s) \mathbf{v}(s) \quad (4) \end{aligned}$$

where $A(s) = \sum_{i=1}^n u_i(s) \frac{\partial X_i}{\partial \mathbf{q}}(\mathbf{q}(s))$, $B(s)$ is the $n \times n$ matrix the columns of which are the $X_i(\mathbf{q}(s))$ and $\mathbf{v}(s)$ is the n dimensional vector composed of the $v_i(s)$. Let us notice that $A(s)$ and $B(s)$ depend only on the current path $\mathbf{q}(s)$. (4) is a linear differential system, the state and input of which are respectively $\eta(s)$ and $\mathbf{v}(s)$. This system gives the relation between the first order variation of the inputs $u_i(s)$'s and the first order variation $\eta(s)$ of the path $\mathbf{q}(s)$. We can integrate System (4) to get the following expression:

$$\eta(s) = H(s) \int_0^s H^{-1}(\tau) B(\tau) \mathbf{v}(\tau) d\tau \quad (5)$$

where $H(s)$ is the $n \times n$ -matrix-valued function that satisfies:

$$H(0) = I_n \quad (6)$$

$$H'(s) = A(s)H(s) \quad (7)$$

I_n is the identity matrix of order n . Given the current path $\mathbf{q}(s)$ and obstacles, we need to choose at each step, functions $v_1(s), \dots, v_k(s)$ and a deformation step h in order to make the new path $\mathbf{q}(s, h)$ get away from obstacles. This is the topic of the next section.

2.3 Obstacles and Path Deformation

Given a set of obstacles detected while following the current path, we define a potential field $U(\mathbf{q})$ in the configuration space in such a way that the value of the potential increases when the robot gets closer to obstacles. There are different ways to design such a potential field. We will give details about this construction later in the paper.

From the potential field in the configuration space, we define the potential of a path by summing $U(\mathbf{q})$ along the path:

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

To make the path go away from obstacles, we choose the function $\mathbf{v}(s) = (v_1(s), \dots, v_k(s), 0, \dots, 0)$ that minimizes the first-order variation of the path potential:

$$\frac{\partial V}{\partial h}(0) = \int_0^S \frac{\partial U}{\partial \mathbf{q}}(\mathbf{q}(s))^T \eta(s) ds \quad (8)$$

\mathbf{v} belongs to an infinite-dimensional space of smooth vector-valued functions defined over $[0, S]$. We restrict our choice to the finite dimensional subspace of truncated Fourier series of order p :

$$\mathbf{v}(s) = \mathbf{v}_0 + \sum_{l=1}^p \mathbf{v}_l \cos\left(\frac{2l\pi}{S}s\right) + \mathbf{w}_l \sin\left(\frac{2l\pi}{S}s\right) \quad (9)$$

where the \mathbf{v}_l 's and \mathbf{w}_l 's are vectors of dimension n with $n - k$ 0's as last components. The choice of truncated Fourier series is a common and practical way of discretizing the space of continuous functions defined over an interval. It has been used for instance in [2] to solve the optimal control problem of the falling cat. Plugging this expression into (5), we get an expression of $\eta(s)$ w.r.t. the coefficients \mathbf{v}_l and \mathbf{w}_l :

$$\eta(s) = F_0(s)\mathbf{v}_0 + \sum_{l=1}^p F_l(s)\mathbf{v}_l + G_l(s)\mathbf{w}_l \quad (10)$$

where

$$F_l(s) = H(s) \int_0^s \cos\left(\frac{2l\pi}{S}\tau\right) H^{-1}(\tau) B(\tau) d\tau \quad (11)$$

$$G_l(s) = H(s) \int_0^s \sin\left(\frac{2l\pi}{S}\tau\right) H^{-1}(\tau) B(\tau) d\tau \quad (12)$$

are $n \times n$ matrix-valued functions over $[0, S]$.

To choose a deformation $\eta(s)$ of the current path, we now only need to choose the coefficients \mathbf{v}_l 's and \mathbf{w}_l 's. For that, we express the variation of the path potential V w.r.t. these coefficients. We replace $\eta(s)$ by expression (10) in (8) and we get:

$$\begin{aligned} \frac{\partial V}{\partial h} &= \sum_{l=0}^p \int_0^S \frac{\partial U}{\partial \mathbf{q}}(\mathbf{q}(s))^T F_l(s) ds \mathbf{v}_l \\ &+ \sum_{l=1}^p \int_0^S \frac{\partial U}{\partial \mathbf{q}}(\mathbf{q}(s))^T G_l(s) ds \mathbf{w}_l \end{aligned}$$

Let us notice that the first order variation of the path potential V is linear w.r.t. the \mathbf{v}_l 's and \mathbf{w}_l 's. The coefficients of the \mathbf{v}_l 's and \mathbf{w}_l 's in this latter expression can be seen as the components of the path potential gradient in the space of Fourier coefficients. To make the potential decrease, we choose the \mathbf{v}_l 's and \mathbf{w}_l 's as the opposite of this gradient:

$$\mathbf{v}_l = - \int_0^S F_l(s)^T \frac{\partial U}{\partial \mathbf{q}}(\mathbf{q}(s)) ds \quad (13)$$

$$\mathbf{w}_l = - \int_0^S G_l(s)^T \frac{\partial U}{\partial \mathbf{q}}(\mathbf{q}(s)) ds \quad (14)$$

Thus $\frac{\partial V}{\partial h}(0) = -\|\mathbf{v}_0\|^2 - \sum_{l=1}^p \|\mathbf{v}_l\|^2 + \|\mathbf{w}_l\|^2 < 0$.

2.4 Boundary conditions

In the deformation process, we constrain the initial and final configurations of the path to remain unchanged. This implies the following equalities: $\mathbf{q}(0, h) = \mathbf{q}(0, 0)$ and $\mathbf{q}(S, h) = \mathbf{q}(S, 0)$. If we compute $\mathbf{q}(s, h)$ by integrating system (1), replacing u_i by $u_i + hv_i$, the above constraints are non-linear and difficult to express w.r.t. the \mathbf{v}_l 's and \mathbf{w}_l 's. To overcome this problem, we approximate $\mathbf{q}(s, h)$ by $\mathbf{q}(s) + h\eta(s)$. With this approximation, the boundary conditions can be expressed by:

$$\eta(0) = F_0(0)\mathbf{v}_0 + \sum_{l=1}^p F_l(0)\mathbf{v}_l + G_l(0)\mathbf{w}_l = 0$$

$$\eta(S) = F_0(S)\mathbf{v}_0 + \sum_{l=1}^p F_l(S)\mathbf{v}_l + G_l(S)\mathbf{w}_l = 0$$

As $F_l(0) = G_l(0) = 0$ for any $0 \leq l \leq p$, the first constraint is always satisfied. The second one is linear in the coefficients \mathbf{v}_l 's and \mathbf{w}_l 's:

$$C\mathbf{z} = 0$$

where $\mathbf{z} = (\mathbf{v}_0 \mathbf{v}_1 \mathbf{w}_1 \dots \mathbf{v}_p \mathbf{w}_p)$ is the vector of all the Fourier coefficients and

$$C = \begin{pmatrix} F_0(S) & F_1(S) & G_1(S) & \dots & F_p(S) & G_p(S) \end{pmatrix}$$

is a matrix. To get Fourier coefficients that satisfy the constraint $C\mathbf{z} = 0$ from the one computed above (13-14), we need to project \mathbf{z} over the subspace defined

by this constraint. The following formula gives the orthogonal projection $\bar{\mathbf{z}}$:

$$\bar{\mathbf{z}} = (I - C^+ C) \mathbf{z} \quad (15)$$

where $C^+ = C^T (C C^T)^{-1}$ is the pseudo-inverse of C . The deformation $\eta(s)$ we apply to the path is obtained by (10), where the \mathbf{v}_l 's and \mathbf{w}_l 's are the coefficients extracted from $\bar{\mathbf{z}}$.

2.5 Correction of nonholonomic deviation

The approximation of the deformed path

$$\mathbf{q}(s, h) \approx \mathbf{q}(s) + h \eta(s) \quad (16)$$

implies a nonholonomic deviation, *i.e.* in equation (1), constraints (2) are not satisfied anymore after a few iterations. The velocity $\mathbf{q}'(s)$ along the path has non zero components $u_{k+1}(s), \dots, u_n(s)$ along vectors $X_{k+1}(\mathbf{q}(s)), \dots, X_n(\mathbf{q}(s))$. We need to keep these components close to 0 in order the path to stay feasible.

For that, we use functions $v_i(s)$'s, $k+1 \leq i \leq n$ that were 0 in Section 2.2 to design a proportional closed-loop regulation:

$$v_i(s) = -\lambda u_i(s) \quad k+1 \leq i \leq n$$

where $\lambda < 1$ is a positive constant. Let us recall that between two steps, the input functions $u_i(s)$ become $u_i(s) + h v_i(s) = (1 - \lambda) u_i(s)$. This regulation corrects the noise introduced by approximation (16). The deformation is then computed according to the following steps.

1. We project the velocity of the current path $\mathbf{q}'(s)$ over vector fields X_i 's ($1 \leq i \leq n$) to get input functions $u_i(s)$'s.
2. we compute $\eta_1(s)$ from Equation (5) with $\mathbf{v}(s) = (0, \dots, 0, -\lambda u_{k+1}(s), \dots, -\lambda u_n(s))$ and
3. we compute $\eta_2(s)$ using the obstacle potential field as described in Section 2.3 and we set $\eta(s) = \eta_1(s) + \eta_2(s)$.
4. The boundary condition $\eta(S) = 0$ becomes $\eta_2(S) = -\eta_1(S)$. This constraint is affine over Fourier coefficients \mathbf{v}_l 's and \mathbf{w}_l 's:

$$Cz = -\eta_1(S)$$

As previously, we project the set of Fourier coefficient obtained from (13-14) over the affine set of coefficients satisfying the above equation:

$$\bar{\mathbf{z}} = -C^+ \eta_1(L) + (I - C^+ C) \mathbf{z}$$

3 Application to the Mobile Robot Hilare Towing a Trailer

In this section, we give some details about the implementation of our method applied to the case of the LAAS mobile robot Hilare 2 towing a trailer (Figure 1). For this system, a configuration is represented by the vector $\mathbf{q} = (x, y, \theta, \varphi)$ where (x, y) and θ are the position and orientation of the robot and φ is the angle of the trailer w.r.t. the robot. The control vectors for this system are

$$X_1 = \begin{pmatrix} \cos \theta \\ \sin \theta \\ 0 \\ -\frac{1}{l_t} \sin \varphi \end{pmatrix} X_2 = \begin{pmatrix} 0 \\ 0 \\ 1 \\ -1 - \frac{l_r}{l_t} \cos \varphi \end{pmatrix}$$

where l_r (resp. l_t) is the distance between the center of the robot (resp. the trailer) and the trailer connection. The inputs of the system are u_1 and u_2 the linear and angular velocities of the robot. We define two additional vector fields to get bases at each configurations:

$$X_3 = \begin{pmatrix} -\sin \theta \\ \cos \theta \\ 0 \\ 0 \end{pmatrix} X_4 = \begin{pmatrix} -\sin(\theta + \varphi) \\ \cos(\theta + \varphi) \\ -l_t - l_r \cos \varphi \\ -l_t \end{pmatrix}$$

We assume that a first path has been computed by a path planner using a map of the environment. This path is sampled at a sample step δs . Then two tasks are executed at the same time: first, the robot starts following the path, detecting obstacles using a laser scanner and second the current path is deformed on parts where collisions are detected. The velocity of the robot along the path decreases when the first collision along the current path is close. If s_0 is the abscissa of the first collision, an interval $[s_0 - h_0, s_0 + h_1]$ is defined on which the current path is deformed. At each step, this interval corresponds to the interval $[0, S]$ of the previous section. Along each sample position of this interval, the deformation task computes $A(s)$ and $B(s)$:

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

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

where to make matrices fit in the column, $c \theta = \cos \theta$, $s \theta = \sin \theta$, $c \varphi = \cos \varphi$, $s \varphi = \sin \varphi$, $c \psi = \cos(\theta + \varphi)$, $s \psi = \sin(\theta + \varphi)$. Then $H(s)$ is computed using expressions (6-7). For each sample position along the path,

$$H(s + \delta s) = H(s) + A(s)H(s)\delta s$$

Once the matrices $H(s)$'s and their inverses have been computed, we compute the $F_l(s)$ and $G_l(s)$ using (11-12). The integrals are approximated by summing the value of the integrand for each sample.

The last step of the deformation computation requires the expression of the configuration space potential field. We describe this potential field in the next section.

3.1 Potential field in \mathcal{C}

Each obstacle point P_i detected by the on-board sensors of the robot produces a potential field in the plane defined as follows. If M is a point in the plane at distance d from P_i ,

$$u_i(M) = \frac{1}{(d+d_0)^2} - \frac{1}{(d_1+d_0)^2} \quad \text{if } 0 \leq d \leq d_1$$

$$u_i(M) = 0 \quad \text{if } d > d_1$$

$d_0 < d_1$ are constant distances. Let $\mathbf{f}_i(M) = -\nabla u_i(M)$ be the force in the plane deriving from this potential. Let R and T 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 and T (Figure 3):

$$U_i(\mathbf{q}) = u_i(R) + u_i(T) \quad (17)$$

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

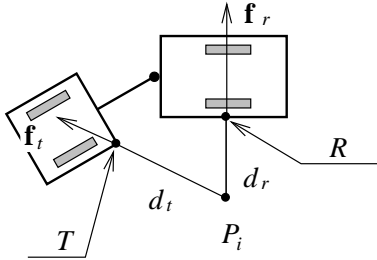


Figure 3: Configuration space potential field generated by each obstacle point P_i .

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

$$U(\mathbf{q}) = \sum_i U(P_i, \mathbf{q})$$

The gradient of the potential field is obtained by differentiating (17) w.r.t. the configuration variables (x, y, θ, φ) .

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

where $\mathbf{f}_r = \mathbf{f}_i(R)$ and $\mathbf{f}_t = \mathbf{f}_i(T)$ are the values of the plane force field induced by u_i at R and at T . We do

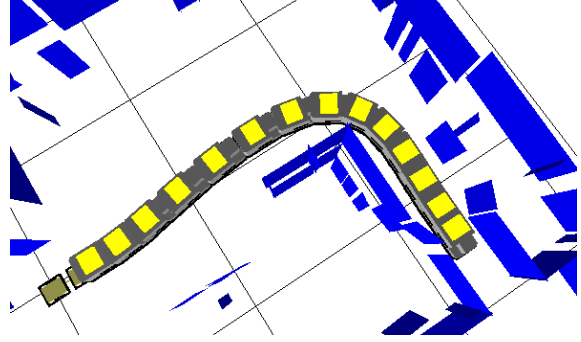


Figure 4: A first trajectory computed by the motion planning platform Move3D. Blue vertical planes are obstacles of the map.

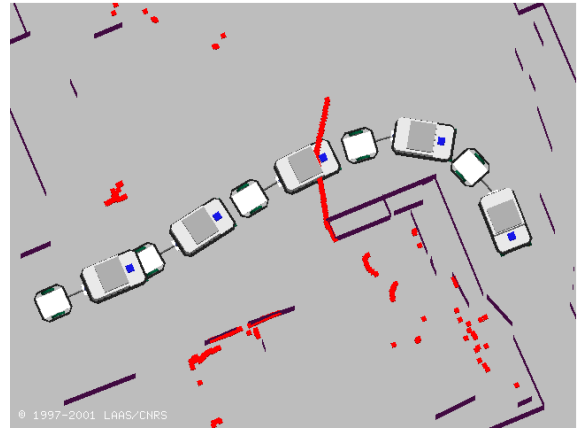


Figure 5: An obstacle lies in the way of the robot. Red dots represent obstacle points detected by the laser telemeter.

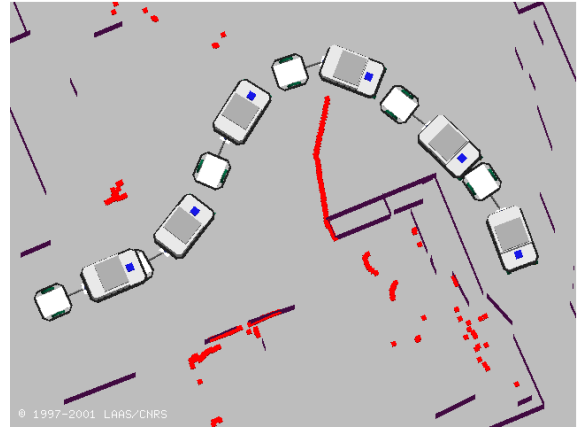


Figure 6: The trajectory after deformation.

not expand further here the expression of the configuration space gradient in order to save space but they can be easily established from the above definitions.

4 Experimental Results

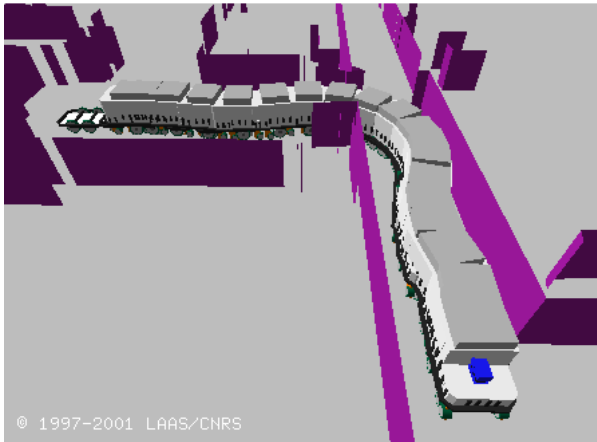


Figure 7: Trajectory in a corridor

We applied our trajectory deformation algorithm to the robot Hilare 2 towing a trailer (Figure 1). This system is of dimension 4 and is subjected to 2 non-holonomic constraints. The first step in the experiment consists in planning a collision-free motion using Move3D, the generic motion planning platform developed at LAAS [8]. Figure 4 shows a trajectory computed by Move3D. The deformation algorithm is then applied to this trajectory. Figure 5 shows the point obstacles detected by a laser range scanner: we put an obstacle in the way of the robot so that the initial trajectory is in collision. Figure 6 displays the trajectory after deformation. In this example, the deformation took approximately 10 seconds on an on-board Motorola PowerPC-750 (350Mhz).

4.1 Static and moving obstacles

The example shown previously shows that our method can avoid in reasonable time static obstacles that lie in the robot way. If the obstacle is detected soon enough, the robot can deform the path without slowing down. Of Course, the computation cost is still too high to make a robot with trailer reactive to moving obstacles like people walking around.

4.2 trajectory deformation and localization uncertainties

In fact, the situation in which our algorithm is the most helpful and the least spectacular is when the planned trajectory turns into a corridor as shown in Figure 7. In this case, when if the corridor is narrow w.r.t. the robot, as it is the case for our robot, a small error of localization or a small inaccuracy in the map

of the environment can cause the robot to collide a wall. The deformation procedure makes this type of maneuver secure and most of the time, the deformation required is very small. The motion of Figure 7 was formerly impossible to execute without collision.

5 Concluding Comments

Our method has the same shortcomings as other variational methods based on a potential fields. It can be trapped in local minima and fail to deform the trajectory until all collisions have disappeared. These situations have to be identified automatically and the deformation process has to be stopped in order to re-plan a new trajectory. For instance, if a door initially opened happens to be closed, replanning is necessary. We will address these problems in future work.

The main advantage of our approach is that it can be applied to any driftless nonholonomic system, even those very complicated for which path planning is still an open problem. As soon as we know the control vector fields of the system, the method can be applied. Moreover, let us notice that the method is completely symmetric. Backward motions can be deformed as well as forward motion. We plan to put a laser telemeter on the trailer in order to illustrate this point.

References

- [1] O. Brock and O. Khatib. High-speed navigation using the global dynamic window approach. In *International Conference on Robotics and Automation*, San Francisco, CA, Apr. 2000. IEEE.
- [2] C. Fernandes, L. Gurvits, and Z. Li. *Nonholonomic Motion Planning*, chapter Optimal Nonholonomic Motion Planning for a Falling Cat, pages 379–421. Kluwer, 1993.
- [3] D. Fox, W. Burgard, and S. Thrun. The dynamic window approach to collision avoidance. *IEEE Robotics and Automation Magazine*, 4(1):23–33, March 1997.
- [4] M. Khatib and R. Chatila. An extended potential field approach for mobile robot sensor-based motions. pages 490–496.
- [5] 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*, Albuquerque, NM, Apr. 1997. IEEE.
- [6] O. Khatib. Real-time obstacle avoidance for manipulators and mobile robots. *International Journal on Robotics Research*, 5:90–98, 1986.
- [7] S. Quinlan. *Real-Time Modification of Collision Free Paths*. PhD thesis, Stanford University, 1995.
- [8] T. Simon, J. Laumond, C. V. Geem, and J. Cortes. Computer aided motion: Move3d within molog. In *International Conference on Robotics and Automation*, Seoul KR, May 2001. IEEE.