

Taking into account velocity and acceleration bounds in nonholonomic trajectory deformation

Mathieu Hillion, Florent Lamiraux

▶ **To cite this version:**

Mathieu Hillion, Florent Lamiraux. Taking into account velocity and acceleration bounds in nonholonomic trajectory deformation. IEEE International Conference on Robotics and Automation, Apr 2007, Roma, Italy. hal-00125707

HAL Id: hal-00125707
https://hal.science/hal-00125707

Submitted on 22 Jan 2007

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.

Taking into account velocity and acceleration bounds in nonholonomic trajectory deformation

Mathieu Hillion
LAAS-CNRS
7 avenue du Colonel Roche
31077 Toulouse cedex 4, France
Email: mhillion@laas.fr

Florent Lamiraux
LAAS-CNRS
7 avenue du Colonel Roche
31077 Toulouse cedex 4, France
Email: florent@laas.fr

***Abstract*— This paper deals with the problem of autonomous navigation for nonholonomic mobile robots. To avoid obstacles while executing a planned motion, we use a nonholonomic trajectory deformation method. Initially, this method did not take into account the velocity and acceleration bounds of the robot. The contribution of this paper is a significant improvement of the method to take into account these kinematic bounds. The idea consists in applying zero input perturbation on intervals on which the kinematic bounds are not satisfied and to reparameterize the trajectory in order to remove bound overflows. Experimental results illustrate and validate the improvement.**

I. Introduction

This paper deals with autonomous navigation for nonholonomic systems. Many mobile robots are subject to kinematic constraints of rolling without slipping that make them nonholonomic. Our objective is to navigate such systems in cluttered environment. For that, we proceed in two steps. First, we plan a collision-free trajectory that satisfies the kinematic constraints of the system and then we follow the planned trajectory. The first step has given rise to a lot of research work in the 1990's and a lot of trajectory planning algorithms for nonholonomic systems are today available [1], [2], [3], [4], [5]. In the second step, we need to cope with map and localization errors and with unexpected obstacles that can make the planned trajectory in collision. To perform this task, we use a nonholonomic trajectory deformation method proposed by [6]. We have implemented and integrated this iterative method on-board a nonholonomic mobile robot towing a trailer. This method locally modifies the input function of the trajectory using a potential field over the configuration space of the robot, until collisions have disappeared. The method manages nonholonomic constraints and keeps the initial and end configurations of the trajectory unchanged. The main weakness of this method, as described in [6], is that it does not take into account the velocity and acceleration bounds of the mobile robot. This can lead to trajectories that the robot cannot follow correctly.

Our contribution in this paper is a significant improvement of the nonholonomic trajectory deformation method that takes into account velocity and acceleration bounds. The idea consists in keeping unperturbed the input functions on intervals where they exceed their bounds and in locally re-parameterizing the trajectory in time in order to keep velocities and accelerations within their bounds.

The problem of time parameterization of a path in order to take into account acceleration and velocity bounds has first been addressed in the context of minimum time path following for manipulator arms [7], [8], [9], [10]. This problem is known to be computationally complex. This complexity forbids us to integrate such computations at each iteration of the trajectory deformation algorithm. Instead, we propose to deal with velocity and acceleration overflows iteratively as soon as they appear.

II. The trajectory deformation algorithm

In this section, we recall some definitions from [6] in order to recall the main components of the trajectory deformation method. We refer the reader to this paper for details about the method. Here we give only what is necessary to the understanding of our work.

A nonholonomic systems is a locally controllable system the velocities of which are subject to linear constraints. As a consequence, the velocity of a nonholonomic system is a linear combination of control vector fields:

$$\dot{\mathbf{q}} = \sum_{i=1}^{k} u_i \mathbf{X}_i(\mathbf{q}) \tag{1}$$

where $\mathbf{q}$ is the configuration and k is the number of control variables of the system. We denote by n the dimension of the configuration space of the system.

A trajectory $\mathbf{q}(s)$ of a nonholonomic system is therefore defined by an initial configuration $\mathbf{q}_0$ and a vector-valued input function defined over an interval $[0, S]$:

$$\begin{array}{rccc} \mathbf{u}: & [0,S] & \rightarrow & \mathbf{R}^k \\ & s & \rightarrow & \mathbf{u}(s) \end{array}$$

The main principle of the trajectory deformation method consists in adequately perturbing the input function of the current trajectory in order to make this trajectory move away from obstacles perceived during motion execution. In other words, at each iteration of the algorithm, an input perturbation $\mathbf{v} : [0, S] \rightarrow \mathbf{R}^k$ and a small positive real number τ are defined and input $\mathbf{u}$ is replaced by $\mathbf{u} + \tau\mathbf{v}$:

$$\forall s \in [0, S], \quad \mathbf{u}(s) \leftarrow \mathbf{u}(s) + \tau\mathbf{v}(s) \tag{2}$$

The relation between input perturbation and trajectory deformation $\eta : [0, S] \to \mathbf{R}^n$ is given by the linearized system:

$$\eta'(s) = A(s)\eta(s) + B(s)\mathbf{v}(s)$$

where η' is the derivative of η, $A(s)$ and $B(s)$ are matrices depending on the current trajectory. Thus, for small values of τ, the trajectory corresponding to the new input defined by (2) is given by:

$$\mathbf{q}(s) \leftarrow \mathbf{q}(s) + \tau\eta(s) + o(\tau)$$

where $o(\tau)$ is a negligible term w.r.t. τ.

A. Finite-dimensional subspace of input perturbations

The space of input perturbations $\mathbf{v}$ of the current trajectory is an infinite-dimensional space. To make computations easier, the choice of $\mathbf{v}$ is restricted over a subspace spanned by elementary input perturbation functions $\mathbf{e}_1, ..., \mathbf{e}_{pk}$ where p is an integer called *Fourier order*, defined as follows: for any i, $0 \leq i \leq p-1$ and any j, $1 \leq j \leq k$,

$$\mathbf{e}_{ik+j}(s) = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ \sin\left(\frac{(i+1)\pi}{S}s\right) \\ 0 \\ \vdots \\ 0 \end{pmatrix} \tag{3}$$

where the non-zero component is at the j-th line of the vector.

Let us notice that functions $\mathbf{e}_l$'s are equal to zero at 0 and at S. This implies that the velocity remains unchanged at the beginning and at the end of the deformation interval. Thus, applying the deformation algorithm over a sub-interval of a given trajectory does not affect the continuity of the velocity along the whole trajectory.

III. Velocity and acceleration bounds

A. Problem statement

In our framework, the input variables of a nonholonomic system are velocities: linear and angular velocities for a unicycle for instance. We assume in this section that system (1) is subject to velocity and acceleration bounds: for any i, $1 \leq i \leq k$,

$$u_{i\,min} \leq u_i \leq u_{i\,max} \tag{4}$$
$$a_{i\,min} \leq \dot{u}_i \leq a_{i\,max} \tag{5}$$

We assume that for any i between 1 and k, $u_{i\,min} < 0$, $a_{i\,min} < 0$, $a_{i\,max} > 0$, and $a_{i\,max} > 0$.

An admissible trajectory is thus now defined by an input function $\mathbf{u}$ that complies with the above inequalities: for any $s \in [0, S]$,

$$\mathbf{u}_{min} \leq \mathbf{u}(s) \leq \mathbf{u}_{max} \tag{6}$$
$$\mathbf{a}_{min} \leq \mathbf{u}'(s) \leq \mathbf{a}_{max} \tag{7}$$

where $\mathbf{u}'$ is the derivative of $\mathbf{u}$, $\mathbf{u}_{min}$, $\mathbf{u}_{max}$, $\mathbf{a}_{min}$, and $\mathbf{a}_{max}$ are k-dimensional vectors the components of which are respectively the $u_{i\,min}$, $u_{i\,max}$, $a_{i\,min}$, $a_{i\,max}$ and by convention, inequality between two vectors means inequality for each component.

As mentioned earlier, at each iteration, input perturbation is spanned by the $\mathbf{e}_l$'s defined by (3):

$$\mathbf{v} = \sum_{l=1}^{pk} \lambda_l \mathbf{e}_l \tag{8}$$

where $\lambda = (\lambda_1, ..., \lambda_{pk})^T$ is a vector of real coefficients chosen in such a way that the trajectory deformation moves away from obstacles. Without getting into details about the choice of λ, we can notice that step (2) and equations (6) and (7) define inequality constraints on vector λ: for any $s \in [0, S]$,

$$\begin{aligned} \mathbf{u}_{min} &\leq \mathbf{u}(s) + \textstyle\sum_{l=1}^{pk} \tau\lambda_l \mathbf{e}_l(s) \leq \mathbf{u}_{max} \\ \mathbf{a}_{min} &\leq \mathbf{u}'(s) + \textstyle\sum_{l=1}^{pk} \tau\lambda_l \mathbf{e}'_l(s) \leq \mathbf{a}_{max} \end{aligned}$$

Each component of the above vector inequalities define an infinite collection of linear inequalities over $\tau\lambda$ indexed by s. Unfortunately, characterizing the set of $\tau\lambda$ satisfying these inequalities is a complex operation that we cannot perform at each iteration of the trajectory deformation algorithm.

To manage velocity and acceleration overflows, we proceed in two steps. Firstly, we detect for each component of the input function the intervals over which a kinematic bound is saturated and we require the corresponding component of the input perturbation to be equal to 0 over these intervals. To achieve this requirement, we redefine the elementary input perturbations $\mathbf{e}_l$'s as explained in the next section. Secondly, we uniformly re-parameterize the deformation interval in order to lower velocities and accelerations down.

B. Overflow Limitation

Given a current trajectory of input function $\mathbf{u}$, we define for each input variable u_i, $1 \leq i \leq k$, the subset I_i of interval $[0, S]$ over which input function component u_i satisfies inequalities (6-7). We assume that this subset is the finite union of n_i disjoint intervals that we denote by:

$$I_i = \bigcup_{j=1}^{n_i} [\sigma_{ij}, \rho_{ij}]$$

where σ_{ij} and ρ_{ij} are the upper and lower bounds of each sub-interval. We denote by N the total number of intervals:

$$N = \sum_{i=1}^{k} n_i$$

We are now going to redefine elementary input perturbations $\mathbf{e}_l$'s by filling each sub-interval with $\sin$ functions of increasing frequencies, following Algorithm 1. The output of this algorithm is a collection of pk input perturbation functions $\mathbf{e}_l$, $1 \leq l \leq pk$, satisfying the following properties:

1) for any $s \in [0, S]$ and any integer i, $1 \leq i \leq k$, if $u_i(s)$ or $u'_i(s)$ is saturated (*i.e.* does not satisfy (6-7)),

Algorithm 1 *Definition of* $\mathbf{e}_l$

```
/* initialization */
ω_min ← ∞; l ← 1
/* for each sub-interval */
for i in [1..k] do
  for j in [1..n_i] do
    ω_ij ← π/(ρ_ij − σ_ij)
    α_ij ← 1 /* multiple of basis pulsation */
  end for
end for
while l ≤ k * p do
  /* find smallest pulsation */
  for i in [1..k] do
    for j in [1..n_i] do
      if ω_ij < ω_min then
        ω_min ← ω_ij /* store smallest pulsation */
        i_min ← i; j_min ← j /* and corresponding interval*/
      end if
    end for
  end for
  e_l ← 0
  i ← i_min; j ← j_min;
  e_li(s) ← sin(ω_ij(s − σ_ij)) for s ∈ [σ_ij, ρ_ij]
  α_ij ← α_ij + 1
  ω_ij ← α_ij π/(ρ_ij − σ_ij)
  l ← l + 1
end while
```

then for any l, $1 \leq l \leq pk$, $e_{li}(s) = 0$ (e_{li} is the i-th components of $\mathbf{e}_l$);

2) for any l, $1 \leq l \leq pk$, $\mathbf{e}_l$ is continuous over $[0, S]$, $\mathbf{e}_l(0) = \mathbf{e}_l(S) = 0$.

These two properties imply that any linear combination (8) does not change the components of the input function and their derivatives over intervals where they are saturated. This first step ensures us that only small velocities and acceleration overflows can occur. However, after several iterations, it may happen that all inputs are saturated over the whole interval $[0, S]$. To overcome this limitation, we perform a each iteration of the deformation algorithm a re-parameterization of the trajectory as described in the following section.

a) Example: to illustrate the above algorithm, we consider a nonholonomic system with two inputs (k=2), one of which u_2 does not satisfy the kinematic bounds. Figure 1 shows the elementary input perturbations $\mathbf{e}_l$'s for $p = 4$ corresponding to this example. It can be easily checked that any linear combination of the $\mathbf{e}_l$'s will not affect u_2 on both intervals where the kinematic bounds are not satisfied.

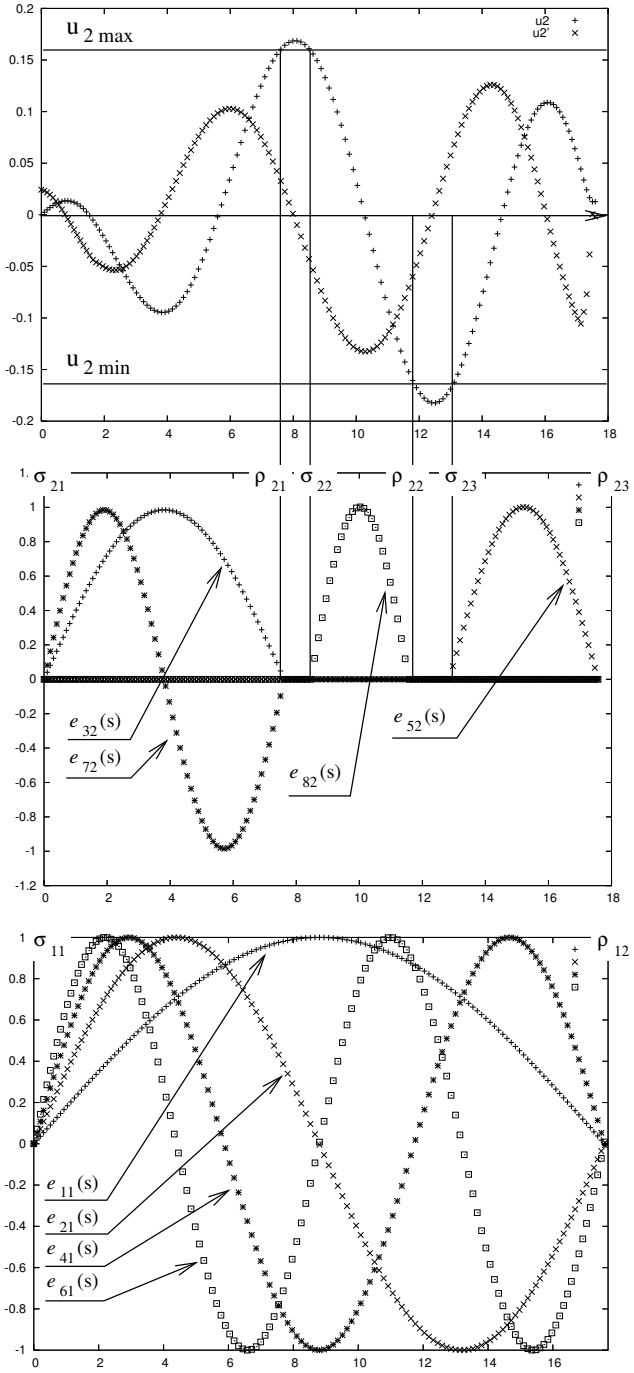


Fig. 1. In this figure, we illustrate algorithm 1 with a two input ($k = 2$) nonholonomic system. Input u_1 and its derivative (not displayed here) satisfy the kinematic bounds along the whole interval. Input u_2 (top) goes above $u_{2\,max}$ and below $u_{2\,min}$ on two intervals; u_2' satisfy kinematic bounds. Therefore, $n_1 = 1$, $n_2 = 3$. p is set to 4, so that the 8 input perturbations $\mathbf{e}_l$ have the following properties: only one component of each $\mathbf{e}_l$, $1 \leq l \leq 8$ is non uniformly equal to 0 and the pulsation of the non zero coordinate of the $\mathbf{e}_l$ increases with l. The middle and bottom picture display the non uniformly zero coordinates of the $\mathbf{e}_l$'s.

C. Re-parameterization

As mentioned in the introduction, computing the minimum-time parameterization of the current trajectory to take into account velocity and acceleration bounds is a

computationally complex task. In this section, we propose a simpler solution.

Let us define an increasing re-parameterization function:

$$\begin{array}{rccc} \varphi: & [0,S] & \rightarrow & [0,\bar{S}] \\ & s & \rightarrow & \varphi(s) \end{array}$$

where $\bar{S}$ is a positive real number and let us denote by

$$\bar{\mathbf{q}}(s) = \mathbf{q}(\varphi^{-1}(s))$$

the current trajectory re-parameterized by φ. The relation between the derivatives of both trajectories is given by:

$$\forall s \in [0,S], \quad \bar{\mathbf{q}}'(\varphi(s))\varphi'(s) \quad = \quad \mathbf{q}'(s)$$

From (1), we can easily deduce that the re-parameterized trajectory satisfies the nonholonomic constraints and corresponds to input function $\bar{\mathbf{u}}$ defined over $[0,\bar{S}]$ by:

$$\bar{\mathbf{u}}(\varphi(s)) \quad = \quad \frac{1}{\varphi'(s)}\mathbf{u}(s) \tag{9}$$

The relation between the derivatives of the input functions before and after re-parameterization is thus given by:

$$\bar{\mathbf{u}}'(\varphi(s)) \quad = \quad \frac{1}{\varphi'(s)^2}\mathbf{u}'(s) - \frac{\varphi''(s)}{\varphi'(s)^3}\mathbf{u}(s) \tag{10}$$

1) Boundary conditions: let us recall that we want the input function to remain unchanged at both end of the deformation interval. This requires that

$$\varphi'(0) = \varphi'(S) = 1 \tag{11}$$

A current trajectory being given, the input function $\mathbf{u}$ of which does not satisfy (6) and (7) , the problem is thus now to find a function φ satisfying (11) and such that the new velocities and accelerations defined by (9) and (10) satisfy constraints (6) and (7):

$$\mathbf{u}_{min} \leq \quad \frac{1}{\varphi'(s)}\mathbf{u}(s) \quad \leq \mathbf{u}_{max} \tag{12}$$

$$\mathbf{a}_{min} \leq \quad \frac{1}{\varphi'(s)^2}\mathbf{u}'(s) - \frac{\varphi''(s)}{\varphi'(s)^3}\mathbf{u}(s) \quad \leq \mathbf{a}_{max} \tag{13}$$

Although (12) defines a set of $2k$ constraints over φ' for each value of parameter s, constraints (13) are much more difficult to deal with since φ' and φ'' are obviously not independent.

2) Our solution: let us define φ as follows:

$$\begin{aligned} \varphi(0) &= 0 \\ \varphi'(s) &= \frac{1}{\sqrt{1 - as(S-s)}} \text{ for } s \in [0,S] \end{aligned}$$

where $a < \frac{4}{S^2}$ is a real parameter. From this expression, we can easily derive that:

$$\frac{1}{\varphi'(s)^2} \quad = \quad 1 - as(S-s) \tag{14}$$

$$\frac{\varphi''(s)}{\varphi'(s)^3} \quad = \quad a(\frac{S}{2} - s) \tag{15}$$

From expression of φ' above, we can state that:

$$\begin{array}{l} (1 - as(S-s))\mathbf{u}(s)^2 \leq \mathbf{u}_{max}^2 \\ (1 - as(S-s))\mathbf{u}(s)^2 \leq \mathbf{u}_{min}^2 \end{array} \Rightarrow (12)$$

where $\mathbf{u}^2$ denotes the vector the components of which are the components of $\mathbf{u}$ squared. The counterpart is true if and only if $\mathbf{u}_{min} = -\mathbf{u}_{max}$.

Thus, if for any $s \in [0,S]$ and any integer i between 1 and k,

$$a \geq \frac{u_i(s)^2 - \min(u_{i\,max}^2, u_{i\,min}^2)}{u_i(s)^2 s(S-s)} \tag{16}$$

Inequalities (12) are satisfied over $[0,S]$.

Let us now rewrite inequalities (13) using expressions (14) and (15):

$$\mathbf{a}_{min} \leq \quad (1 - as(S-s))\mathbf{u}'(s) - a(\tfrac{S}{2} - s)\mathbf{u}(s) \quad \leq \mathbf{a}_{max}$$

which is equivalent to:

$$\mathbf{u}'(s) - \mathbf{a}_{max} \leq a\left(s(S-s)\mathbf{u}'(s) + (\frac{S}{2} - s)\mathbf{u}(s)\right) \leq \mathbf{u}'(s) - \mathbf{a}_{min}$$

Each of these inequalities define an interval for parameter a. We need now to choose a in the intersection of all these intervals and those defined by (16).

For that, we collect lower and upper bounds for a by evaluating the above inequalities for each sample value of s along the deformation interval of the current trajectory. Let us denote by α_{min} and α_{max} the corresponding lower and upper bounds of a.

Two situations may occur. Either $\alpha_{min} \leq \alpha_{max}$ and the set of parameters a that satisfy the kinematic bounds is not empty. In this case, we choose

$$a = \alpha_{min}$$

in order to minimize the length (time of execution) of the interval of definition of the reparameterized trajectory. It can also happen that $\alpha_{min} > \alpha_{max}$ and no value of parameter a can satisfy all the kinematic constraints. In this case, we choose $a = 0$. This corresponds to skipping the reparameterization step. Thus, if the initial trajectory satisfies the kinematic bounds, bound overflow will occur when updating the trajectory at step (2). In this case, on intervals where the kinematic bounds are not satisfied, the bound overflows will not increase. By choosing τ sufficiently small in (2), we can make sure that bound overflow will be upper bounded by a positive number. By defining a security margin on the kinematic bounds, we can make sure that the bounds will always be satisfied everywhere.

IV. Experimental results

In this section, we propose two experimental results. The first one illustrates influence of the two steps we have added to the nonholonomic trajectory deformation method. The second one shows the result of the new method for a mobile robot towing a trailer in a real environment.

A. Point-wise unicycle

We consider in this example a point-wise unicycle mobile robot going through an obstacle (Figure 2).

The configuration space of this system is of dimension 3. A configuration is represented by a vector (x, y, θ) where (x, y) is the position of the center of the robot and θ the

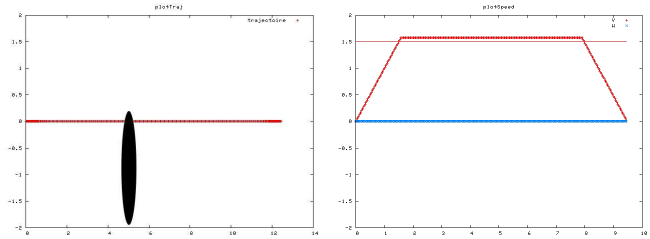


Fig. 2. On the left, the initial trajectory: the unicycle mobile robot goes straight into an obstacle. On the right, the angular velocity u_2 is uniformly equal to 0, the linear velocity linearly increases, stays constant and linearly decreases.

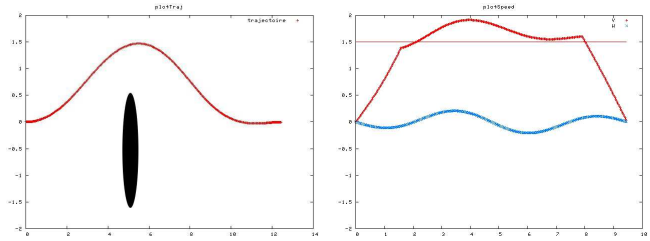


Fig. 3. On the left, the final trajectory after a few iterations of the trajectory deformation algorithm without taking into account the kinematic bounds: the unicycle mobile robot avoids the obstacle. On the right, the angular velocity u_2 is not anymore uniformly equal to 0 and the linear velocity has increased on most of the interval.

orientation. The control vector fields of this mobile robot are respectively the linear and angular velocities:

$$X_1(x, y, \theta) = \begin{pmatrix} \cos\theta \\ \sin\theta \\ 0 \end{pmatrix} \qquad X_2(x, y, \theta) = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

We deform the initial trajectory three times in order to get away from the obstacle, keeping the initial and end configuration of the trajectory unchanged.

The first time, we do not manage velocity and acceleration bounds. The second time, we apply zero input perturbations over intervals where constraints are not satisfied anymore (Section III-B). The third time, we do the same as in the second time, but moreover, we re-parameterize the trajectory at each iteration of the trajectory deformation process.

In these experiments, the kinematic bounds are the following:

$$\begin{aligned} -1.5 \leq \quad & u_1 \quad \leq 1.5 \\ -1.5 \leq \quad & u_2 \quad \leq 1.5 \\ -1 \leq \quad & u_1' \quad \leq 1 \\ -1 \leq \quad & u_2' \quad \leq 1 \end{aligned}$$

B. Without taking into account kinematic bounds

In this first experiment, we deform the trajectory without taking into account the kinematic bounds, as described in [6]. The algorithm is iterative and applies small input perturbation at each step. Figure 3 shows the trajectory obtained after 40 iterations. The obstacle is avoided but the linear velocity is beyond the maximal value over most of the interval of definition.

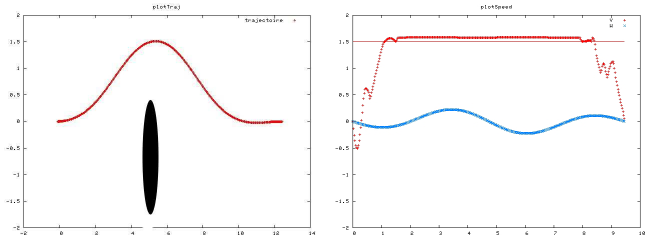


Fig. 4. Input functions are perturbed only on intervals where they are not saturated. As a consequence, kinematic bound overflows remain small. However, at the end the linear velocity or acceleration are over their bounds on almost the whole interval.

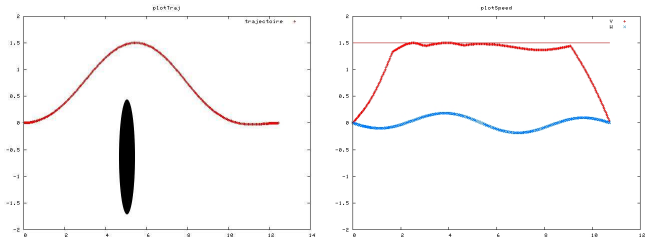


Fig. 5. Input functions are perturbed only on intervals where they are not saturated and the trajectory is re-parameterized at each iteration of the trajectory deformation algorithm. As a consequence, at the end of the process all input functions are within their bounds and the interval of deformation is a little longer: 10.8 instead of 9.4.

C. Keeping input functions unchanged when they are saturated

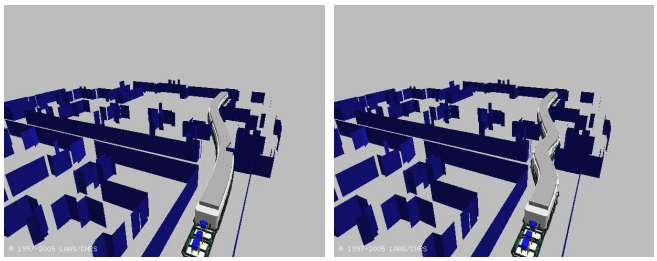
In this experiment, we apply the trajectory deformation algorithm as explained in Section III-B. Elementary input perturbation are sinus functions of multiple frequencies, distributed over intervals where the input function components are within their bounds. Figure 4 shows the trajectory and the input functions after 40 iterations. The kinematic bound overflow remains small. However, at the end of the deformation process, the linear velocity (or acceleration) is saturated on the whole interval.

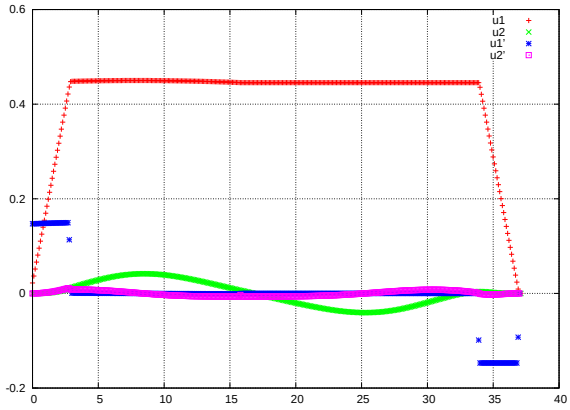
D. Keeping input functions unchanged when they are saturated and with re-parameterization

In this last experiment, we apply the same technique as in the former section, but moreover, we re-parameterize the trajectory at each iteration as explained in Section III-C. Figure 5 shows the resulting trajectory and input function. The kinematic bounds are satisfied everywhere both in velocity and acceleration. As a consequence, the interval of definition of the trajectory is longer at the end of the deformation process than at the beginning.

E. Mobile robot towing a trailer

In this experiment, we have applied the improved trajectory deformation algorithm to a unicycle mobile robot towing a trailer. The input function of this system is two-dimensional ($k = 2$) and corresponds to the linear (u_1) and the angular (u_2) velocities of the unicycle. The kinematic bounds of the





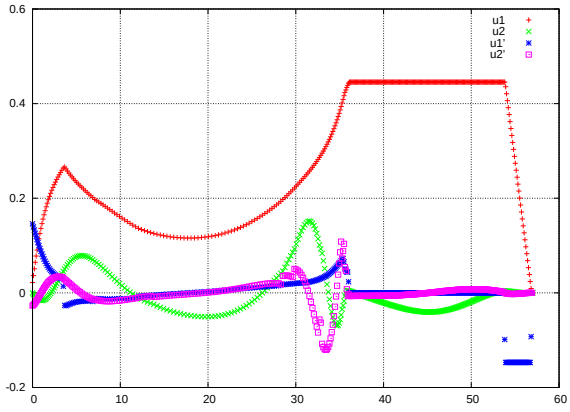


Fig. 6. The trajectory deformation algorithm taking into account kinematic bounds. The robot is a differentially driven mobile robot towing a trailer. The two inputs of the system are the linear and angular velocities of the unicycle. The top left picture shows the initial trajectory, in collision with obstacles of the environment. The top right picture shows the deformed trajectory after 8 iterations. The middle picture shows the input functions components u_1, u_2 and their derivatives u_1', u_2' along the initial trajectory. The bottom picture shows the same functions corresponding to the end trajectory. Let us notice that only the beginning of the trajectory is deformed and that the kinematic bounds remain satisfied.

system are given by the following inequalities:

$$\begin{aligned} -0.45 \leq \quad & u_1 \quad \leq 0.45 \\ -0.15 \leq \quad & u_2 \quad \leq 0.15 \\ -0.2 \leq \quad & u_1' \quad \leq 0.2 \\ -0.15 \leq \quad & u_2' \quad \leq 0.15 \end{aligned}$$

Figure 6 shows the result of the experiment. The initial trajectory is in collision. After 7 iterations of the improved trajectory deformation algorithm, the trajectory is collision-free and satisfies almost everywhere the kinematic constraints. Notice that the interval of definition of the trajectory is longer at the end than at the beginning. This is due to the constraints on the angular velocity and acceleration: to avoid obstacles, the robot needs to turn more than in the initial trajectory. The kinematic constraints relative to the rotation requires robot to slow down. That is why, the trajectory takes more time.

V. Conclusion

In this paper, we have proposed a significant improvement of the trajectory deformation algorithm for nonholonomic systems defined in [6]. The improvement is based on the redefinition of the elementary input perturbation function and on the re-parameterization of the trajectory in order to lower down the velocities and accelerations. Experimental results show that the improvement yields good results in practice.

Acknowledgment

References

[1] J. Barraquand and J.-C. Latombe, "Nonholonomic multibody mobile robots: Controllability and motion planning in the presence of obstacles," *Algorithmica*, vol. 10, pp. 121–155, 1993.

[2] J. Laumond, P. Jacobs, M. Taix, and R. Murray, "A motion planner for nonholonomic mobile robots," *IEEE Transactions on Robotics and Automation*, vol. 10, pp. 577–593, 1994.

[3] P. Svestka and M. Overmars, "Coordinated motion planning for multiple car-like robots using probabilistic roadmaps," in *Proc. IEEE Int. Conf. Robotics and Automation*, Japan, 1995, pp. 1631–1636.

[4] S. M. LaValle and J. J. Kuffner, "Randomized kinodynamic planning," *International Journal of Robotics Research*, vol. 20, no. 5, pp. 378–400, May 2001.

[5] J.-P. Laumond, Ed., *Robot Motion Planning and Control*, ser. Lectures Notes in Control and Information Sciences 229. Springer, 1998, no. ISBN 3-540-76219-1.

[6] F. Lamiraux, D. Bonnafous, and O. Lefebvre, "Reactive path deformation for nonholonomic mobile robots," *IEEE Transactions on Robotics*, vol. 20, no. 6, pp. 967–977, Dec 2004.

[7] K. Shin and N. McKay, "Minimum-time control of robotic manipulators with geometric path constraints," *IEEE Transactions on Automatic Control*, vol. 30, pp. 531–541, 1986.

[8] J. Bobrow, S. Dubowsky, and J. Gibson, "Time-optimal control of robotic manipulators along specified paths," *International Journal on Robotics Research*, vol. 4, pp. 3–17, 1985.

[9] J. Slotine and H. Yang, "Improving the efficiency of time-optimal path-following algorithms," *IEEE Transactions on Robotics and Automation*, vol. 5, no. 1, pp. 118–124, 1989.

[10] Z. Shiller and S. Dubowsky, "On computing time-optimal motions of robotic manipulators in the presence of obstacles," *IEEE Transactions on Robotics and Automation*, vol. 7, pp. 785–797, 1991.