

Path Optimization for Nonholonomic Systems: Application to Reactive Obstacle Avoidance and Path Planning

Florent Lamiroux¹, David Bonnafous¹, and Carl Van Geem²

¹ LAAS-CNRS, 7 avenue du Colonel Roche 31077 Toulouse cedex 4, France
florent@laas.fr,

WWW home page: <http://www.laas.fr/~florent>

² Kineo-CAM, 7 avenue du Colonel Roche 31077 Toulouse cedex 4, France
cvangeem@kineocam.com,

WWW home page: <http://www.kineocam.com>

Abstract. This paper presents a method of path optimization for non-holonomic systems. This method consists in iteratively modifying a given feasible path in order to make a cost related to the path decrease. The paper presents two main applications of this method: the first one is an algorithm that solves the problem of path planning for complex kinematic systems (*i. e.* trucks with two trailers) in extremely constrained environments. The second one is an application in mobile robotics and addresses the problem of reactive trajectory deformation for nonholonomic mobile robots (*i. e.* a cart towing a trailer) in order to avoid unexpected obstacles, and cope with map uncertainty and localization errors.

1 Introduction

Nonholonomic systems have raised interesting issues in robotics as well as in control theory for more than twenty years. These systems first appeared in robotics in the 70's when the first wheeled mobile robots were built. The kinematic constraint imposed by wheels raised successive problems, ranging from controllability, to motion planning and closed loop motion control that tied links between mobile robotics and control theory. Since then, kinematically more complex systems like cars or trucks towing trailers have made these links even stronger.

Joint works in control theory and in robotics have led to practical solutions in the domains of path planning in the presence of obstacles and of motion control for nonholonomic systems. The work presented in this paper is related to both issues and aims at providing new solutions for situations where common approaches generally fail.

The first such situation, in which classical path planning methods are inefficient is the problem of planning a path for a truck towing several trailers on a road when

- the trailer connections are not on the wheel axis of the truck or of the former trailer,

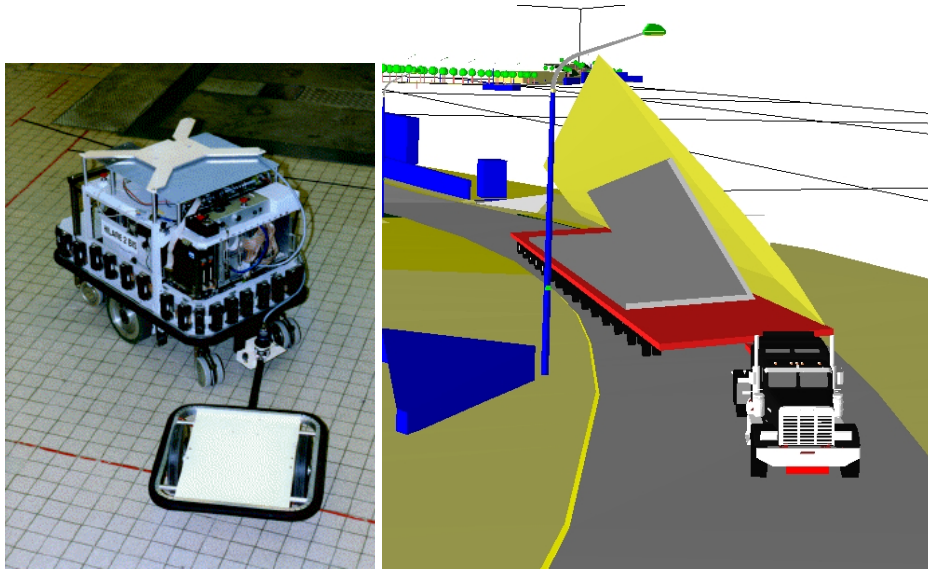


Fig. 1. Two nonholonomic systems: Hilare 2 towing a trailer (dimension 4) and a truck towing a trailer carrying a wing of the Airbus 380 (dimension 6).

- the truck is very large and the road is very narrow.

Classical path planning methods in this situation are inefficient for several reasons. First, this type of kinematic systems are not well understood in control theory : it is not differentially flat nor is it nilpotent. In other words, there is no known steering method for this type of vehicles and thus path planning methods using a steering method [2, 5, 6] are difficult to use. Other classical path planning methods explore the configuration space by expanding a tree [1, 3]. Nodes of the tree are generated by applying constant inputs to the system over small intervals of time, from an existing node (configuration). These methods suffer a major drawback in our context: they require some parameters to be correctly tuned. These parameters are mainly the number of nodes expanded from each leaf of the tree and the time interval over which constant inputs are applied. Too small time intervals and too many nodes expanded from each leaf of the tree lead to huge data structures and time of computation, whereas too large intervals of time and not enough nodes expanded may lead to failure in finding a solution for very constrained problems. Let us moreover notice that computation time and data-structure size grow with the dimension of the configuration space. In examples on which we apply our method, the dimension of the configuration space is 6 and paths are approximately 2km long and have only a few tens of centimeters clearance. In this case, tree-based motion planning methods are not applicable.

The second situation in which the method described in this paper is helpful is when a nonholonomic mobile robot tries to execute a planned motion. Usually

the map of the environment used by the path planning algorithm does not fit the real world, localization of the robot is never exact and unexpected obstacles may lie in the volume swept by the planned motion. For these reasons, executing a planned motion as such usually leads to collisions.

The method we propose in this paper provides an efficient solution to the problems of path planning and of motion control described above. This method takes as input a discretized feasible path, and iteratively compute modifications of this path in order to get away from obstacles and to keep the nonholonomic constraints satisfied. The output of the method is a collision-free feasible path. The method is described in Section 2. Sections 3 and 4 present two different applications of the method.

2 Nonholonomic Systems and Path Deformation

The principle of our method is the same in both applications. A first feasible path, possibly in collision being given, the method iteratively modifies this path in such a way that the path moves away from obstacles. The deformation applied at each iteration makes a potential function related to the path decrease. This potential function decreases when the distance of the path to obstacles increases. Our method is an extension to nonholonomic systems of the elastic strip algorithm proposed in [4]. This later algorithm applies to each configuration $\mathbf{q}(s)$ along the initial path a deformation vector $\eta(s)$ (Figure 2) in order to make the path move away from obstacles. If the robot is not subject to any kinematic constraint, any path is feasible and thus any deformation outputs a feasible path. In our case, the systems we work with are nonholonomic. This means that any path is not necessarily feasible and that any deformation of a feasible path may output a non-feasible path. The seminal idea of our method consists in choosing deformations that keep the nonholonomic constraints of the system satisfied. These deformations are those obtained by perturbing the input functions of the initial path as explained later.

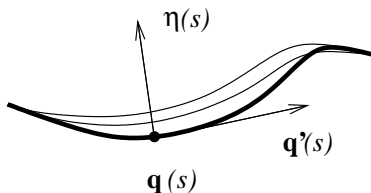


Fig. 2. Current path $\mathbf{q}(s)$ (in bold) and a deformation $\eta(s)$ along this path.

In the rest of this section, we describe one iteration of our deformation algorithm.

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 are the linear combinations of the $X_i(q)$'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 $\mathbf{q}(s)$. The reader will understand in Section 2.5 the reason for these additional vector fields. Up to this point, we invite him to forget these vector fields even though they appear in the following equations multiplied by zero functions $u_{k+1}(s), \dots, u_n(s)$.

2.2 Infinitesimal Path Deformation

To deform a given path we only need to perturb the input functions $u_1(s), \dots, u_k(s)$ of the initial path $\mathbf{q}(s)$. For that, we define n real functions $v_1(s), \dots, v_n(s)$ called *input perturbations*, 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 :

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

If we denote now by $\eta(s) = \frac{\partial \mathbf{q}}{\partial h}(s, 0)$ the *infinitesimal deformation* and by $\eta'(s) = \frac{\partial \eta}{\partial s}(s)$ the derivative w.r.t. the path parameter s , the above equation becomes for $h = 0$:

$$\eta'(s) = \sum_{i=1}^n v_i(s) X_i(\mathbf{q}(s)) + u_i(s) \frac{\partial X_i}{\partial \mathbf{q}}(\mathbf{q}(s)) \eta(s) \quad (3)$$

$$= A(s) \eta(s) + B(s) \mathbf{v}(s) \quad (4)$$

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 control 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 $\mathbf{q}(s)$. This system is in fact the tangent linearized system of (1) about the initial trajectory $\mathbf{q}(s)$. 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 $\mathbf{q}(s, h)$ move away from obstacles. This is the topic of the next section.

2.3 Obstacles and Infinitesimal Path Deformation

Given a set of obstacles in the workspace, 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 in sections 3.3 and 4.3.

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 only need to find functions $\mathbf{v}(s) = (v_1(s), \dots, v_k(s), 0, \dots, 0)$ such that the first-order variation of the path potential:

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

is negative. The set of smooth functions defined over an interval is an infinite-dimensional space. In order to be able to represent function $\mathbf{v}(s)$ by a finite-dimensional vector, we restrict the choice of $\mathbf{v}(s)$ to a finite-dimensional subspace spanned by a set of test functions $(\mathbf{e}_1(s), \dots, \mathbf{e}_p(s))$, where p is an integer, defined over $[0, S]$ into \mathbf{R}^n with the last $n - k$ components uniformly equal to 0. Different solutions are available for this finite-dimensional subspace : polynomials, truncated Fourier series for instance. We thus impose

$$\mathbf{v}(s) = \sum_{l=1}^p \lambda_l \mathbf{e}_l(s) \quad (9)$$

where the vector $\lambda = (\lambda_1, \dots, \lambda_p)$ is the vector of coefficients of the linear combination of $\mathbf{e}_i(s)$'s. Plugging this expression into (5), we get an expression of $\eta(s)$ w.r.t. the coefficients λ_i .

$$\eta(s) = \sum_{l=1}^p \lambda_l \mathbf{E}_l(s) \quad (10)$$

where

$$\mathbf{E}_l(s) = H(s) \int_0^s H^{-1}(\tau) B(\tau) \mathbf{e}_l(\tau) d\tau \quad (11)$$

are the elementary infinitesimal deformations defined over $[0, S]$, solution of System (4) for each input perturbation $\mathbf{e}_l(s)$.

We now need to choose the coefficients λ_i 's in such a way that the variation of the path potential V is negative. Let us express this variation w.r.t. these coefficients. We replace $\eta(s)$ by expression (10) in (8) and we get:

$$\frac{\partial V}{\partial h}(0) = \sum_{l=1}^p \lambda_l \int_0^S \frac{\partial U}{\partial \mathbf{q}}(\mathbf{q}(s))^T \mathbf{E}_l(s) ds$$

Let us notice that the first order variation of the path potential V is linear w.r.t. the λ_i 's. To make the potential decrease, we choose the λ_i 's as follows:

$$\lambda_l = - \int_0^S \mathbf{E}_l(s)^T \frac{\partial U}{\partial \mathbf{q}}(\mathbf{q}(s)) ds \quad (12)$$

Thus $\frac{\partial V}{\partial h}(0) = - \sum_{l=1}^p \lambda_l^2 < 0$.

2.4 Boundary Conditions

Once the vector $\lambda = (\lambda_1, \dots, \lambda_p)$ has been computed as explained above, and once an iteration step h has been chosen, we could plug the new input functions $u_i(s) + hv_i(s)$ as input to system (1) and get as output a new trajectory $\mathbf{q}(s, h)$. However, the end configuration of the deformation interval $\mathbf{q}(S)$ would be modified. If the deformation interval $[0, S]$ is a sub-interval of the initial path interval of definition, after deformation, the path would lose continuity. We need to impose $\mathbf{q}(S, h) = \mathbf{q}(S)$. Let us notice that

1. this boundary condition is non-linear over the vector of coefficients λ ,
2. the vector λ we have computed in the previous section does not satisfy this constraint in general.

By approximating the deformed trajectory $\mathbf{q}(s, h)$ by

$$\mathbf{q}(s, h) \approx \mathbf{q}(s) + h\eta(s) \quad (13)$$

the above boundary conditions becomes $\eta(S) = 0$ and according to (10), this constraint is linear over vector λ :

$$\eta(S) = \sum_{l=1}^p \lambda_l \mathbf{E}_l(S) = 0 \quad (14)$$

To get a vector that satisfies linear constraints from a vector that does not, we can project the latter vector over the linear subspace defined by the constraints. The constraint matrix L is the matrix the columns of which are the vectors $\mathbf{E}_l(S)$'s:

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

The linear subspace defined by constraint (14) becomes $\{\nu \in \mathbf{R}^p, L\nu = 0\}$ and the projection of vector λ defined by (12) is the following:

$$\bar{\lambda} = (I - L^+L)\lambda \quad (15)$$

where $L^+ = L^T(LL^T)^{-1}$ is the pseudo-inverse of L . Replacing λ by $\bar{\lambda}$ in (10) results in an infinitesimal deformation that satisfies the boundary condition $\eta(S) = 0$. It can easily be verified that this new infinitesimal deformation still makes the path potential decrease.

2.5 Correction of Nonholonomic Deviation

The approximation (13) we made in order to make the boundary condition a linear constraint induces a side effect: this approximation implies a slight nonholonomic deviation: in equation (1), constraints (2) are not satisfied anymore. The velocity $\mathbf{q}'(s)$ along the path has small but non zero components $u_{k+1}(s), \dots, u_n(s)$ along vectors $X_{k+1}(\mathbf{q}(s)), \dots, X_n(\mathbf{q}(s))$. If we do not correct this deviation, it tends to get amplified after a few iterations. In this section, we explain how to keep these undesirable components close to zero.

In the previous sections, we applied input perturbations to the input functions $u_1(s) \dots u_k(s)$ corresponding to the vector fields of the system. In this section, we apply input perturbation to the components $u_{k+1}(s) \dots u_n(s)$ of the velocity $\mathbf{q}'(s)$ along the additional vector fields X_{k+1}, \dots, X_n in order to keep these components close to 0. The following input perturbation

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

with $0 < \mu < 1/h$ corresponds to a proportional closed-loop regulation since after deformation and up to approximation (13), the input functions become $u_i(s) \leftarrow u_i(s) + hv_i(s) = (1 - h\mu)u_i(s)$. This regulation corrects the noise introduced by approximation (13). The deformation is now 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 over interval $[0, S]$,
2. we compute $\eta_{nhd}(s)$ from Equation (5) with

$$\mathbf{v}(s) = (0, \dots, 0, -\mu u_{k+1}(s), \dots, -\mu u_n(s)),$$

3. we compute vector λ from the obstacle potential field as described in Section 2.3 and we denote by $\eta_{obst}(s) = \sum_{l=1}^p \lambda_l \mathbf{E}_l(s)$ the corresponding infinitesimal deformation.

4. If we set $\eta(s) = \eta_{nhd}(s) + \eta_{obst}(s)$, the limit condition $\eta(S) = 0$ becomes $\eta_{obst}(S) = -\eta_{nhd}(S)$. This constraint is affine over the vector of coefficients λ :

$$L\lambda = -\eta_{nhd}(S)$$

As previously, we project vector λ obtained from (12) over the affine set of coefficients satisfying the above equation:

$$\bar{\lambda} = -L^+ \eta_{nhd}(S) + (I - L^+L)\lambda$$

to get an infinitesimal deformation that gets away from obstacles and that corrects nonholonomic deviation.

2.6 Stability Issues for Trailer Systems

Applying the path deformation method described above to truck-trailer systems raises stability issues as explained now.

Truck trailer systems are well-known for being stable when driving forward and unstable when driving backward. this stability (resp. unstability) implies exponential stability (resp. unstability) for the tangent linearized system (4). As a result, the singular values of matrix $H(s)$ decrease exponentially for forward motions and increase exponentially for backward motions. Similarly, the singular values of $H^{-1}(s)$ increase exponentially for forward motions and decrease exponentially for backward motions. This fact implies undesirable numerical effects. Formula (5) indeed combines matrices H and H^{-1} . These matrices are composed of very small and very large coefficients so that numerical errors become of the same order of magnitude as values computed, making the algorithm fail.

This serious limitation can be overcome by combining the control vector fields of the system to get a kinematically equivalent system without the above exponential unstability. We do not provide generic solutions to this problem in this paper. For our two systems, we propose specific solutions described in sections 3 and 4.

3 Application to the Mobile Robot Hilare Towing a Trailer

In this section we describe the application of our method to the mobile robot Hilare 2 towing a trailer (Figure 1 left). 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 vector fields for this system are

$$Y_1 = \begin{pmatrix} \cos \theta \\ \sin \theta \\ 0 \\ -\frac{1}{l_t} \sin \varphi \end{pmatrix} Y_2 = \begin{pmatrix} 0 \\ 0 \\ 1 \\ -1 - \frac{l_r}{l_t} \cos \varphi \end{pmatrix}$$

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.

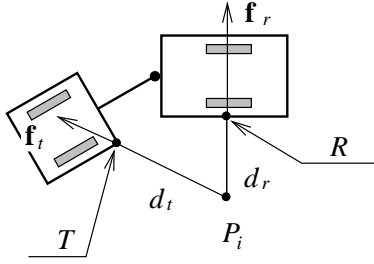


Fig. 3. Configuration space potential field generated by each obstacle point P_i .

3.1 Stability Analysis

As mentioned in Section 2.6, the exponential (un)stability of trailer systems is inherited by the tangent linearized system. As an example, let us consider a straight line motion for the robot : $\mathbf{q}(s) = (s, 0, 0, 0)$, $(u_1(s), u_2(s)) = (1, 0)$. matrices $A(s)$ in Equation 4, $H(s)$ and $H^{-1}(s)$ are in this case:

$$A(s) = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1/l_t \end{pmatrix} \quad H(s) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & s & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & e^{-s/l_t} \end{pmatrix} \quad H^{-1}(s) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & -s & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & e^{s/l_t} \end{pmatrix}$$

We notice that $H(s)$ grows linearly while $H^{-1}(s)$ grows exponentially. The exponential term in $H(s)$ means that if we start the straight line motion with a small angle φ , applying the same input functions $(u_1(s), u_2(s)) = (1, 0)$ exponentially brings φ back to its initial path value 0.

To get rid of this exponential mode, we replace Y_1 and Y_2 by the following linear combinations:

$$X_1 = Y_1 - \frac{\sin \varphi}{l_t + l_r \cos \varphi} Y_2 = \begin{pmatrix} \cos \theta \\ \sin \theta \\ -\frac{\sin \varphi}{l_t + l_r \cos \varphi} \\ 0 \end{pmatrix}$$

$$X_2 = Y_2$$

Vector field X_1 corresponds to circular motions with constant angle between the robot and the trailer. Along these motions, there is no exponential (un)stability. The reader can indeed check that along circular motions,

1. the tangent linearized system (4) is constant w.r.t. s ,
2. matrix $A(s) = A$ has 0 as unique eigenvalue and therefore,
3. $H(s)$ and $H^{-1}(s)$ grow no faster than a polynomial.

Of course, the relative stability along circular motions is not a proof of stability along any trajectory. From a practical point of view, however, we notice that the algorithm behaves much better with these new vector fields, and especially when the interval of deformation gets larger.

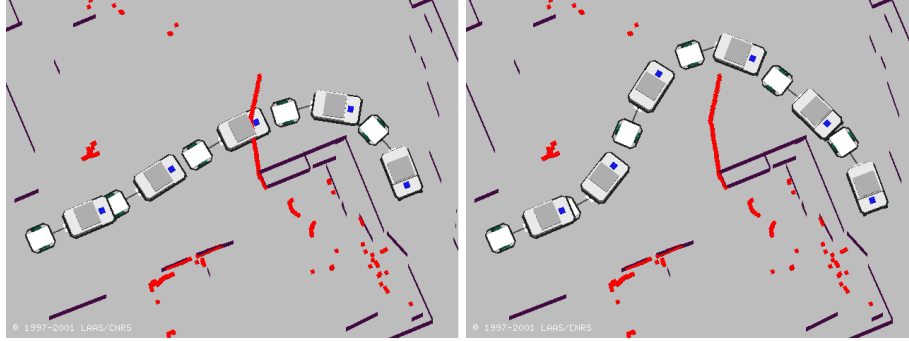


Fig. 4. On the left, a first trajectory computed by the motion planning platform Move3D. The obstacles of the map are in purple. An obstacle lies in the way of the robot. Red dots represent points detected by the laser range finder. On the right, the trajectory after deformation. On this example, the robot stays at the beginning of the path (bottom left) during deformation.

3.2 Additional Vector Fields

We define two additional vector fields to get a basis at each configuration:

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

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 with a laser scanner and secondly 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 over 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

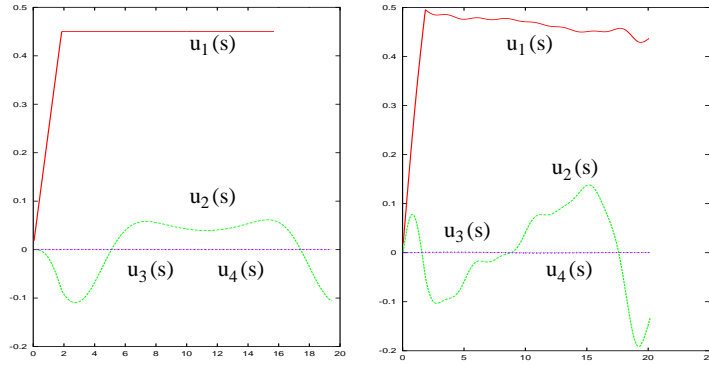


Fig. 5. Input functions $u_1(s)$, $u_2(s)$ along control vector fields and $u_3(s)$, $u_4(s)$ along additional vector fields, before (left) and after (right) deformation. Notice that, as expected, after deformation, the components of the velocity along additional vector fields stay very close to 0.

task computes $A(s)$ and $B(s)$:

$$A(s) = \begin{pmatrix} 0 & 0 & -u_1 s \theta & 0 \\ 0 & 0 & u_1 c \theta & 0 \\ 0 & 0 & 0 & -u_1 \frac{l_r + l_t c \varphi}{(l_t + l_r c \varphi)^2} \\ 0 & 0 & 0 & \frac{u_2 l_r s \varphi}{l_t} \end{pmatrix} B(s) = \begin{pmatrix} c \theta & 0 & -s \theta & -s \psi \\ s \theta & 0 & c \theta & c \psi \\ -\frac{\sin \varphi}{l_t + l_r \cos \varphi} & 1 & 0 & -l_t - l_r c \varphi \\ 0 & -1 - \frac{l_r}{l_t} c \varphi & 0 & -l_t \end{pmatrix}$$

where to make notation shorter, $c\theta = \cos \theta$, $s\theta = \sin \theta$, $c\varphi = \cos \varphi$, $s\varphi = \sin \varphi$, $c\psi = \cos(\theta + \varphi)$, $s\psi = \sin(\theta + \varphi)$. 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 $\mathbf{E}_l(s)$ using (11) where the test functions \mathbf{e}_l are the vector-valued functions with all components equal to 0 except one which is equal to $\cos(\frac{2m\pi s}{S})$ or $\sin(\frac{2m\pi s}{mS})$ where m is an integer between 0 and M : a user defined maximal order. The input perturbations are thus the truncated Fourier series of order less than $M + 1$.

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.

3.3 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 ,

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

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

$d_0 < d_1$ 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):

$$U_i(\mathbf{q}) = u_i(R) + u_i(T) \quad (16)$$

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

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 (16) w.r.t. the configuration variables (x, y, θ, φ) .

$$\frac{\partial U_i}{\partial \mathbf{q}}(\mathbf{q}) = \nabla u_i(R) \frac{\partial R}{\partial \mathbf{q}} + \nabla u_i(T) \frac{\partial T}{\partial \mathbf{q}}$$

$$= -\mathbf{f}_r \frac{\partial R}{\partial \mathbf{q}} - \mathbf{f}_t \frac{\partial T}{\partial \mathbf{q}}$$

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 .

3.4 Experimental Results:

Figure 4 (left) shows a path planned for the mobile robot Hilare 2 towing a trailer using Move3D the path planning platform developed at LAAS [7]. We put an obstacle on this path and run our method on-board. After approximately 10 seconds of deformation, the path was clear of collision as shown on Figure 4 (right). Figure 5 shows how the correction of the nonholonomic deviation keeps forbidden velocities close to 0. Without this correction, $u_3(s)$ and $u_4(s)$ grow and become quickly of the same order of magnitude as $u_1(s)$ and $u_2(s)$.

4 Application to Path Planning for Trucks and Trailers

Another field of application for our method has been opened by a project between Airbus Transportation, LAAS, the French Department of Transportation and Kineo-CAM, a start-up company created by researchers from LAAS. The goal of this project is to validate the itinerary of convoys carrying huge parts of the future Airbus 380 aircraft. Sizes of the freights do not allow the convoys to take the highway since existing bridges are too low. Therefore, the convoys need to

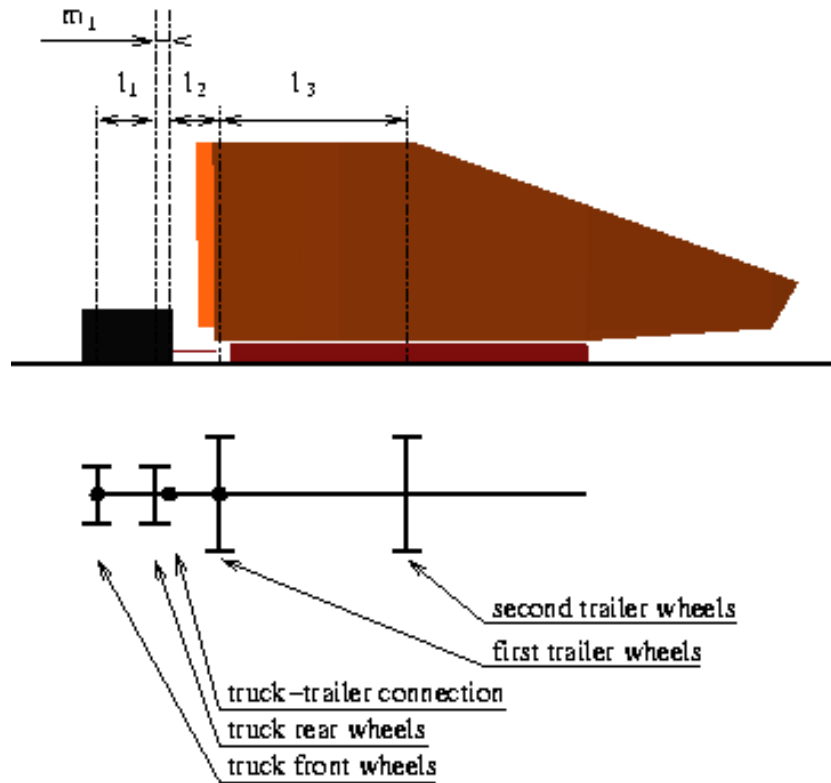


Fig. 6. Truck towing a trailer with a pole. This system is equivalent to a truck towing two trailers. The black box is the bounding box of the truck. The red volume on the trailer is a bounding volume for the wing. The vehicle is 50m long, 12.5m high and 7.7m wide.

take secondary roads through villages (Figure 1 right). The problem raised in this project is thus a path planning problem. Given the map of the road, of buildings along the road, given the modeling of the convoys, is it possible to cross the village without collision? Beside path planning, there is a problem of optimization since the convoys have to stay as far as possible from obstacles.

The path planning problem raised here is quite different from the classical formulation however. First, previous work on path planning for nonholonomic systems often have taken advantage of the property of small-time local controllability to design complete path planner. In our application, convoys are not allowed to drive backward and thus lose this controllability property. This constraint breaks down the completeness property of most path planning algorithms. Secondly, the class of homotopy of the solution path is known in advance: the convoy has to follow the road. For these reasons, but also because the distance margins are extremely tight, our path deformation method is the perfect tool for this problem. The planner we developed takes as input an initial trajectory on

the road, possibly with collisions and returns a trajectory that locally maximizes the distance to obstacles.

4.1 Modeling of the Convoys

Different truck-trailer systems have been investigated in the project. We describe here only the most interesting for us, that is the most complex and the largest. This system, described on Figure 6 is composed of a truck and of a trailer connected to the truck by a pole. The trailer is equipped with 24 orientable wheels. The orientation of each of these wheels is controlled by the angle between the pole and the trailer, in such a way that the axis of the wheel passes by the center of rotation of the trailer. From a control point of view, this system is equivalent to a truck towing two trailers, the first trailer being connected behind the wheel axis of the truck (Figure 6 bottom). This system is neither differentially flat nor nilpotent. In other words, no exact steering method is known up to now for this system.

The usual vector fields associated to this system, corresponding respectively to the linear velocity of the truck and to the steering angle time derivative are:

$$Y_1 = \begin{pmatrix} \cos(\theta) \\ \sin(\theta) \\ \frac{\sin \varphi_0}{l_1 \cos \varphi_0} \\ 0 \\ -\frac{l_2 \sin \varphi_0 + \sin \varphi_1 l_1 \cos \varphi_0 + m_1 \cos \varphi_1 \sin \varphi_0}{\frac{\sin \varphi_1 l_3 - l_2 \cos \varphi_1 \sin \varphi_2}{l_2 l_3} + \frac{\cos \varphi_0 l_1 l_2}{m_1 \sin \varphi_0 (\cos \varphi_1 l_3 - \sin \varphi_1 \sin \varphi_2 l_2)}} \\ \frac{\cos \varphi_0 l_1 l_2}{m_1 \sin \varphi_0 (\cos \varphi_1 l_3 - \sin \varphi_1 \sin \varphi_2 l_2)} \\ \frac{\sin \varphi_1 l_3 - l_2 \cos \varphi_1 \sin \varphi_2}{l_2 l_3} + \frac{\cos \varphi_0 l_1 l_2}{m_1 \sin \varphi_0 (\cos \varphi_1 l_3 - \sin \varphi_1 \sin \varphi_2 l_2)} \end{pmatrix} Y_2 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

where configurations are parameterized by vector $\mathbf{q} = (x, y, \theta, \varphi_0, \varphi_1, \varphi_2)$, (x, y) being the position of the middle point of the truck rear axis, θ the orientation of the truck, φ_0 , the steering angle, φ_1 the orientation of the first trailer w.r.t. the truck and φ_2 the orientation of the second trailer w.r.t. the first trailer (Figure 7).

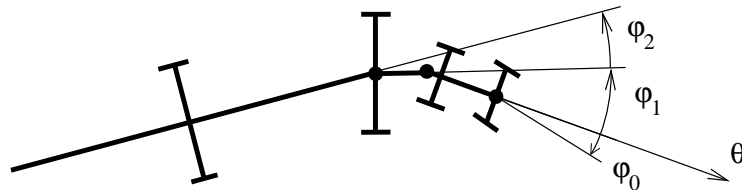


Fig. 7. Parameterization of configurations of a truck towing two trailers.

4.2 Stability Analysis

As for Hilare and its trailer the tangent linearized system of the above system produces matrices $H^{-1}(s)$ that grow exponentially and numerical troubles make the algorithm behave poorly when the interval of deformation becomes large.

Once again the solution we propose is based on intuition and we only observed that this solution makes the algorithm behave much better, without any further proof.

The idea of this solution is basically the same as for Hilare with trailer and consists in finding a combination of the control vector fields

$$\begin{aligned} X_1 &= Y_1 + \alpha(\mathbf{q})Y_2 \\ X_2 &= Y_2 \end{aligned}$$

such that matrix $A(s)$ of the tangent linearized system (4) about integral curves of X_1 has 0 as unique eigenvalue. To design such a vector field, we set $\alpha(\mathbf{q}) = a_0\varphi_0 + a_1\varphi_1 + a_2\varphi_2$. $\alpha(\mathbf{q})$ does not depend on x, y, θ in order to keep symmetries of the system. Thus

$$\frac{\partial X_1}{\partial \mathbf{q}}(\mathbf{q}) = \frac{\partial Y_1}{\partial \mathbf{q}} + \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & a_0 & a_1 & a_2 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Then we impose the characteristic polynomial of $\frac{\partial X_1}{\partial \mathbf{q}}(\mathbf{q})$ to be equal to s^6 . Let us denote by $P(s) = s^6 + \alpha_5s^5 + \alpha_4s^4 + \alpha_3s^3 + \alpha_2s^2 + \alpha_1s + \alpha_0$, this characteristic polynomial, where the α_i 's are expressions containing $a_0, a_1, a_2, \varphi_0, \varphi_1$ and φ_2 . Among equations $\alpha_i = 0, i \in \{0, \dots, 5\}$, 3 are trivial ($0 = 0$) due to the structure of matrix $\frac{\partial X_1}{\partial \mathbf{q}}(\mathbf{q})$. By solving the 3 remaining equations, we get expressions of a_0, a_1, a_2 w.r.t. φ_0, φ_1 and φ_2 . These expressions are too long to be reported in this paper. They can be easily obtained using any good symbolic computation software.

4.3 Potential Field in \mathcal{C}

The interaction between the environment and the different bodies of the vehicle is slightly different from the Hilare-trailer case. The configuration space potential field is now a sum of three terms: one related to the truck and two related to the trailer. In order to make interactions between the vehicle and the environment smoother, the trailer is divided into two symmetric parts, defining two separate bodies, as explained on Figure 8.

$$u(\mathbf{q}) = u_{truck}(\mathbf{q}) + u_{trailer-right}(\mathbf{q}) + u_{trailer-left}(\mathbf{q})$$

The potential of each body in a given configuration depends on the distance d_{body} between the body and the obstacles:

$$\begin{aligned} u_{body}(\mathbf{q}) &= \frac{1}{(d_{body}+d_0)^2} - \frac{1}{(d_1+d_0)^2} \text{ if } 0 \leq d \leq d_1 \\ u_{body}(\mathbf{q}) &= 0 \text{ if } d > d_1 \end{aligned}$$

similarly as in Section 3.3.

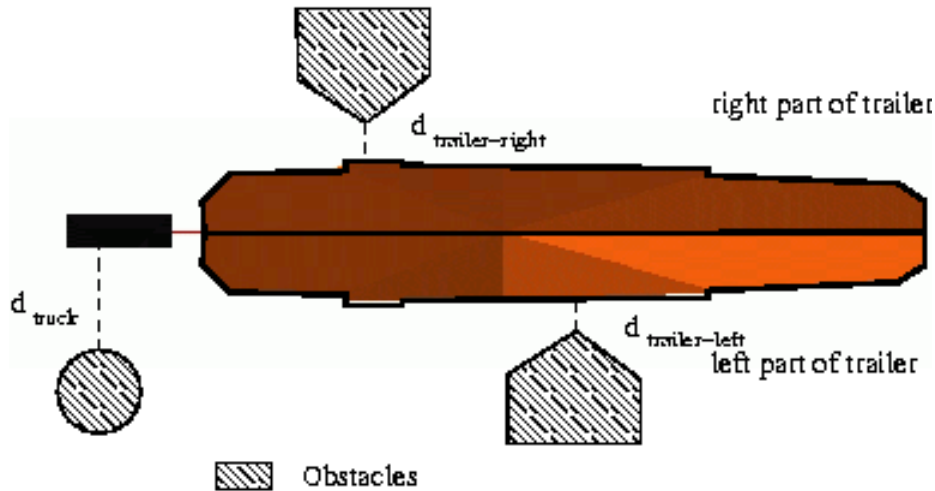


Fig. 8. The potential of a configuration of the truck is the sum of three terms, each one related to a body of the vehicle: truck, left and right side of the trailer. The trailer is far wider than the truck and thus usually much closer to obstacles. If the trailer was considered as a unique body, for configurations of the trailer approximately equidistant from two obstacles, only the closer would interact, making the deformation process oscillate.

4.4 Experimental Results

The algorithm described in this section has been integrated in a software designed by Kineo-CAM and used to optimize trajectories for different vehicles in two villages in the Southwest of France: Lé vignac and Gimont. Figure 9 shows the truck towing a trailer carrying a wing (left) and a path computed in Lé vignac (right). Figure 10 shows a path in a curve. As for Hilare towing a trailer (Figure 5), the components of the velocity along forbidden vector fields is kept very close to 0.

5 Conclusion and Future Work

We have described in this paper a nonholonomic path optimization method. We have applied this method to two different systems in two different contexts. The method has been initially designed for Hilare and its trailer, a system of dimension 4 and later applied to path planning for more complex systems. This extension has been possible because of the genericity of the approach: we basically only need to know the control vector fields of the system to be able to apply this method.

From a practical point of view however, in order to make the method efficient, adequate combinations of vector fields have to be found to avoid exponential

unstability. We have proposed specific solutions for each of our applications. More work needs to be done to find generic solutions to this issue.

Two other issues need to be addressed to make our method really efficient for an autonomous mobile robot.

1. As most algorithms, a lot of parameters have to be tuned to make the method work in different situations. Autonomy requires that the robot finds itself the right parameters for each situation.
2. When collision cannot be removed by a simple path deformation, re-planning is necessary. The method has to detect these situations and return failure to inform the upper level that replanning is necessary.

We are currently working on both issues.

Acknowledgement.

This work has been conducted in the framework of a common project involving Airbus Transportation and the Direction Régionale de l'Équipement of the French Department of Transportation.

References

1. J. Barraquand and J.-C. Latombe. Nonholonomic multibody mobile robots: Controllability and motion planning in the presence of obstacles. *Algorithmica*, 10:121–155, 1993.
2. L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. Overmars. Probabilistic roadmaps for fast path planning in high dimensional configuration spaces. *IEEE Tr. on Rob. and Autom.*, 12:566–580, 1996.
3. J. Kuffner and S. LaValle. Rrt-connect: An efficient approach to single-query path planning. In *International Conference on Robotics and Automation*, San Francisco, CA, April 2000. IEEE.
4. S. Quinlan. *Real-Time Modification of Collision Free Paths*. PhD thesis, Stanford University, 1995.
5. S. Sekhavat, P. Svestka, J.-P. Laumond, and M. Overmars. Multi-level path planning for nonholonomic robots using semi-holonomic subsystems. *International Journal on Robotics Research*, 17:840–857, 1998.
6. T. Simeon, J.-P. Laumond, and C. Nissoux. Visibility-based probabilistic roadmaps for motion planning. *Journal of Advanced Robotics*, 14(6):477–494, 2000.
7. T. Siméon, JP. 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.

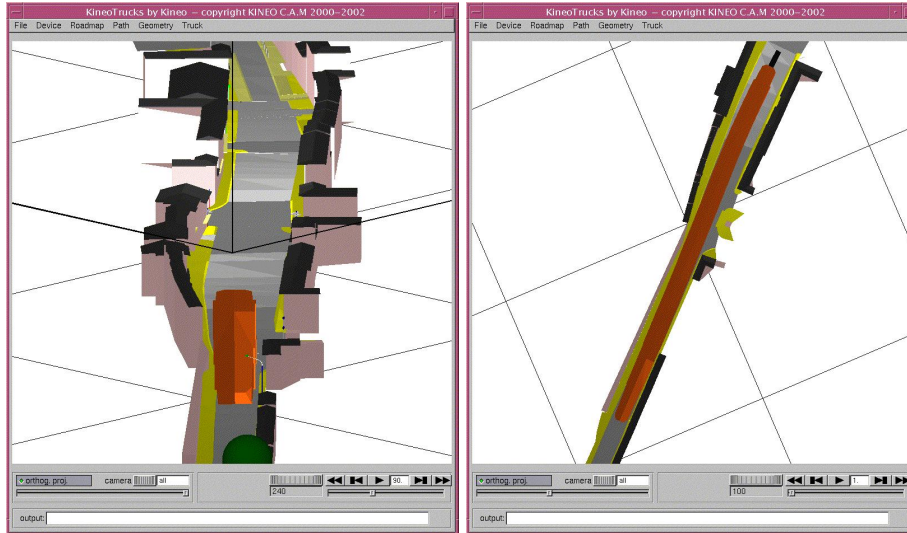


Fig. 9. Truck with trailer in Lévignac. Although the problem seems easy since the road is almost straight, the path clearance here is of only 40cm. On the right, the volume swept by the trailer is in red

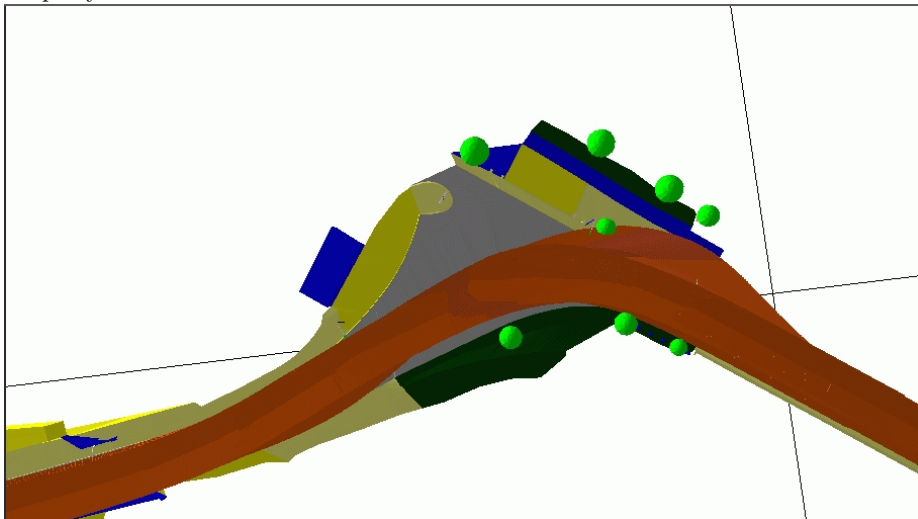


Fig. 10. Volume swept by the trailer in a large curve through Gimont. Trees are not taken into account as obstacles.