



HAL
open science

Smooth motion planning for car-like vehicles

Florent Lamiroux, Jean-Paul Laumond

► **To cite this version:**

Florent Lamiroux, Jean-Paul Laumond. Smooth motion planning for car-like vehicles. IEEE Transactions on Robotics and Automation, 2001, 17 (4), pp.498-502. 10.1109/70.954762 . hal-02962892

HAL Id: hal-02962892

<https://laas.hal.science/hal-02962892>

Submitted on 9 Oct 2020

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.

Smooth motion planning for car-like vehicles

F. Lamiroux and J.-P. Laumond

Abstract—This paper presents a steering method for a car-like vehicle providing smooth paths subjected to curvature constraints. We show how to integrate this steering method in a global motion planning scheme taking obstacles into account. The main idea of the paper is to consider the car as a 4-dimensional system from a kinematic point of view and as a 3-dimensional system from a geometric point of view of collision checking. The resulting planned motions are guaranteed to be collision-free and C^2 between two cusp points.

Keywords—Smooth Motion Planning, Nonholonomic Car-like Robot.

I. INTRODUCTION

In the framework of motion planning for nonholonomic systems, the car-like vehicle has been the most investigated system. Numerous motion planners including obstacle avoidance capabilities are today available (e.g. [3], [4], [17], [15], [8], [21], [11], [24], [29]). All these approaches consider the car-like as a 3-dimensional system moving in the plane and subjected to constraints on the curvature (in addition to the nonholonomic constraint of rolling without slipping). The pioneering work by Dubins [7], and then by Reeds and Shepp [23], showed that the minimal length¹ paths for a car-like vehicle consist of a finite sequence of two elementary components: arcs of circle (with minimal turning radius) and straight line segments. From then, almost all of the proposed motion planners compute collision-free paths constituted by such sequences. As a result the paths are piecewise C^2 , i.e. they are C^2 along elementary components, but the curvature is discontinuous between two elementary components. To follow such paths, a real system has to stop at these discontinuity points in order to ensure the continuity of the linear and angular velocities.

To overcome this inconvenience several authors have proposed to smooth the sequences straight line-arc of circle by clothoids (e.g., [12], [9]). The paths are then C^2 between two cusp points. However, this approach raises another problem: clothoids do not have a closed form making the control of their shapes difficult and dangerous in the presence of obstacles. This smoothing technique usually affects the completeness of the motion planner. The only exception is the work appearing in [24].

In this paper, we propose to revisit the problem by considering a car as a 4-dimensional system from a control point of view: the steering angle is a configuration variable. Such a system has been investigated from a control point of view (e.g., [30]) but without considering the ob-

stacle avoidance problem.

After introducing the model of the car (Section II), we present the steering method which is derived from a method previously developed by the authors for a mobile robot pulling trailers (Section III). This approach guarantees the curvature of the path to be C^2 between cusp points. Then we show how to plug the steering method within two different nonholonomic motion planning schemes (Section IV). In the first scheme, the algorithm computes a collision-free holonomic path in $\mathbf{R}^2 \times S^1$ (the nonholonomic constraints are ignored, only the obstacles of the environment are taken into account); then the path is approximated by a sequence of admissible paths computed with the steering method applied to the 4-dimensional control system. In the second case, the local method is plugged into PRM (Probabilistic Roadmap Planner): A graph is constructed by picking random free configurations and by connecting them by collision-free paths returned by our local method. To ensure the completeness of these schemes, the steering method has to account for the small-time controllability of the system: to connect configurations close to each other, the steering method has to produce paths that remain close to these configurations [25].

The main contribution of this paper is not to provide a completely new method, but to combine existing techniques from an adequate model of the car and to propose a practical well-grounded algorithm for planning collision-free paths such that the curvature is continuous between cusp points.

II. CONTROLLABILITY OF A CAR AND ADMISSIBLE PATHS

Control model. The modeling of vehicles according to their locomotion systems is well understood (see [6]). Let us consider the system represented in Figure 1. The distance between the reference point (x, y) and the middle point of the driving wheels is assumed to be 1. The orientation of the car is denoted by θ . The configuration space $\mathcal{C} = \mathbf{R}^2 \times (S^1)^2$ of this system is 4-dimensional. The two controls of a car are the velocity v of the driving wheels and the time derivative ω of the steering angle ζ . The steering angle is constrained by mechanical bounds: $|\zeta| \leq \zeta_{max}$. A configuration $X = (x, y, \theta, \zeta)$ is said to be *admissible* if $|\zeta| < \zeta_{max}$. A car corresponds to the following control system:

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\zeta} \end{pmatrix} = \begin{pmatrix} \cos \zeta \cos \theta \\ \cos \zeta \sin \theta \\ \sin \zeta \\ 0 \end{pmatrix} v + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \omega \quad (1)$$

Applying the technique of the Lie bracket rank condition, such a system is proved to be small-time controllable at any point (see for instance [20]). This means that, starting

A shorter version of this paper appears in the 6th International Conference on Intelligent Autonomous Systems

Florent Lamiroux and J.-P. Laumond are with LAAS-CNRS, Toulouse, France. E-mail: {florent—jpl}@laas.fr

¹More precisely, the length here is the length of the path followed by the middle of the rear wheel axis.

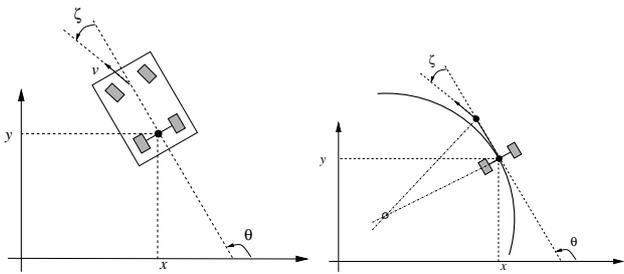


Fig. 1. Model of a car and a canonical curve.

from any configuration, for any time T , the domain reachable with bounded velocity ($|v| < 1$, $|\omega| < 1$) and in time less than T always contains a neighborhood of the starting configuration.

In a lot of path planning work, the steering angle is not a configuration variable. In this case, the model can be simplified as follows. By setting $\tilde{v} = v \cos \zeta$ and $\tilde{\omega} = v \sin \zeta$ we get the following 3-dimensional control system:

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} \cos \theta \\ \sin \theta \\ 0 \end{pmatrix} \tilde{v} + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \tilde{\omega} \quad (2)$$

This system looks like the kinematic model of the so called unicycle. The main difference lies in the admissible control domains. Here the constraints on \tilde{v} and $\tilde{\omega}$ are no longer independent. By setting $v_{max} = \sqrt{2}$ and $\zeta_{max} = \frac{\pi}{4}$ we get: $0 \leq |\tilde{\omega}| \leq |\tilde{v}| \leq 1$. The curvature of the path should be smaller than 1, whenever it is defined. The various existing motion planners for cars usually consider this 3-dimensional model.

In the next section, we build a steering method based on the combination of canonical paths. The idea is to combine two paths passing by two different configurations to get a feasible path that goes from the first to the second configuration. We define now what we call a canonical path associated to a configuration.

Canonical paths, canonical curves Let us consider a sufficiently smooth path followed by the reference point. Simple computations show that the tangent to the path gives the orientation θ of the car, while the steering angle ζ is related to the curvature² κ of the path by $\kappa = \tan \zeta$. These relations define a one-to-one mapping between the space of admissible configurations and $\mathbf{R}^2 \times S^1 \times [-1, 1]$. In other word, any configuration can be parameterized by a vector (x, y, θ, κ) where κ is the curvature defined above. Now, given a configuration $X = (x, y, \theta, \kappa)$, there exists a unique feasible path passing by X and keeping κ constant. This *canonical path*, denoted by $\Gamma(X, s)$, is obtained by integrating system (1) with $v = 1$ and $\omega = 0$ over the time interval $[0, s]$. The corresponding curve $\gamma(X, s)$ followed by the reference point (x, y) is an arc of circle if $\kappa \neq 0$ and a straight line if $\kappa = 0$. Let us notice that this curve is parameterized by arc-length s . $\gamma(X, s)$ is called the *canon-*

²This property derives from the notion of flatness recently introduced in control theory [10].

ical curve associated to X . By construction we get the following property:

Property 1: The canonical path of an admissible configuration verifies the curvature constraint.

III. A STEERING METHOD

To any configuration X , the definition above associates a path passing by this configuration. We are now going to show how to use these canonical curves to build a feasible path between two configurations.

We define a smooth increasing function α from $[0, 1]$ into $[0, 1]$ verifying: $\alpha(0) = 0$, $\alpha(1) = 1$, $\dot{\alpha}(0) = \dot{\alpha}(1) = \ddot{\alpha}(1) = 0$. Let X_1 and X_2 be the initial and goal configurations respectively.

It can be easily verified that the curve $P(t) = (1 - \alpha(t))\gamma(X_1, t) + \alpha(t)\gamma(X_2, t - 1)$ has the same position, tangent, and curvature as $\gamma(X_1, t)$ for $t = 0$ and as $\gamma(X_2, t - 1)$ for $t = 1$. Therefore, it corresponds to a feasible path $X(t)$ in \mathcal{C} going from X_1 to X_2 when t goes from 0 to 1. The configurations along $X(t)$ are computed from the curve $P(t)$, the orientation of its tangent, and its curvature using the relation $\zeta = \arctan \kappa$. The key point here is that κ is continuous along $X(t)$.

This construction defines a steering method that allows the car to reach any configuration from any other one. However, this steering method is not suitable for integration in our collision free scheme described below since it does not account for small-time controllability as defined now (see [25] for details).

Definition: A steering method *Steer* is said to account for small-time controllability if it satisfies the following property:

$$\forall \varepsilon > 0, \exists \alpha > 0, \forall (X_1, X_2) \in \mathcal{C}^2 \\ d(X_1, X_2) < \alpha \Rightarrow \text{Steer}(X_1, X_2) \subset \mathcal{B}(X_1, \varepsilon)$$

where d is a distance in the configuration space \mathcal{C} of the system and $\mathcal{B}(X_1, \alpha)$ is the ball of radius α centered on X_1 (using distance d), and $\text{Steer}(X_1, X_2)$ is the path followed by the steering method between X_1 and X_2 .

To account for small-time controllability, it can be easily stated that a steering method has to generate cusp points. In [14], we show how to build a steering method accounting for small-time controllability using the above convex combination of canonical curves. We briefly recall here the main ideas of this construction. Let $X_1 = (x_1, y_1, \theta_1, \kappa_1)$ and $X_2 = (x_2, y_2, \theta_2, \kappa_2)$ be initial and final configurations. We define M_2 as the orthogonal projection of (x_2, y_2) on $\gamma(X_1, t)$. We define v_2 the parameter of this projection on $\gamma(X_1, t)$: $M_2 = \gamma(X_1, v_2)$. Then we slightly modify the above construction of $P(t)$ as follows:

$$P(t) = (1 - \alpha(t))\gamma(X_1, v_2 t) + \alpha(t)\gamma(X_2, v_2(t - 1)).$$

The corresponding path in \mathcal{C} , that we denote by $\text{Steer}^*(X_1, X_2)(t)$, still represents a feasible path going from X_1 to X_2 . Importantly, if X_2 is on the canonical path associated to X_1 , $\text{Steer}^*(X_1, X_2)(t)$ is exactly the canonical

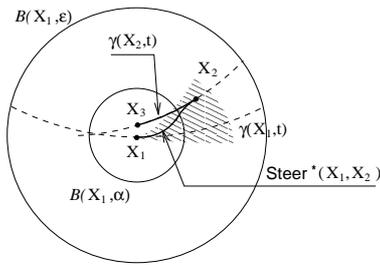


Fig. 2. Steer accounts for small-time controllability.

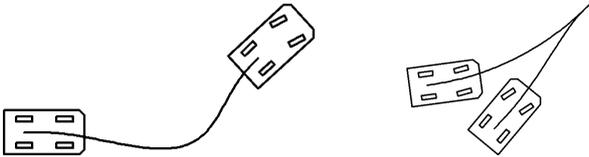


Fig. 3. Two paths generated by Steer. The first one (left) without cusp is a convex combination of the canonical curves associated to each configuration. The second one (right) is composed of a convex combination between the first configuration (bottom) and a cusp configuration, and of the canonical path associated to the second configuration (top).

path $\Gamma(X_1, t)$. The continuity of Steer^* w.r.t. X_1 and X_2 enables us to prove that an open set around the canonical path $\Gamma(X_1, t)$ (the shaded area in Figure 2) is reachable by Steer^* without escaping the ball of radius ε centered on X_1 . Using now the continuity of $\Gamma(X, t)$ w.r.t. X , we establish that if a configuration X_3 is close to X_1 , $\Gamma(X_3, t)$ intersects the open set we have just defined close to X_1 and we can choose a configuration X_2 in this intersection. This process defines another steering method Steer as follows:

1. if X_3 is in the open set reachable by Steer^* , then $\text{Steer}(X_1, X_3) = \text{Steer}^*(X_1, X_3)$,
2. otherwise $\text{Steer}(X_1, X_3)$ is composed of two sub-paths. The first one goes from X_1 to X_2 following $\text{Steer}^*(X_1, X_2)$ between X_1 and X_2 . The second one goes from X_2 to X_3 following $\Gamma(X_3, t)$.

With this construction Steer can access a neighborhood of a configuration X_1 without escaping any given ball centered on X_1 (Figure 2). It accounts for small-time controllability. As a consequence, if X_1 and X_3 are admissible configurations close enough, then ζ_1 and ζ_3 are close enough in $] -\zeta_{max}, \zeta_{max}[$ to guarantee that all the configurations (x, y, θ, ζ) generated by $\text{Steer}(X_1, X_3)$ are admissible, i.e., they verify $\zeta \in] -\zeta_{max}, \zeta_{max}[$. Gathering this result with the curvature continuity result we get the following property:

Property 2: For two sufficiently close admissible configurations X_1 and X_3 , all the configurations of the path $\text{Steer}(X_1, X_3)$ are admissible. The path followed by the reference point is C^2 between X_1 and X_2 and between X_2 and X_3 .

Remark: The collision free path planning scheme we define later builds paths composed of sequences of sub-paths generated by Steer . The continuity of the curvature between two sub-paths ensures us that a real system can follow these paths without stopping between each sub-paths.

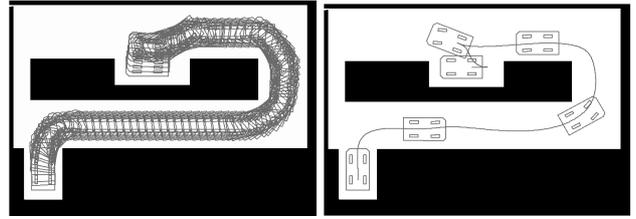


Fig. 4. A first geometric path and the feasible path generated by the approximation scheme with $\zeta_{max} = 60$ deg

Figure 3 shows two examples of paths generated by Steer .

IV. PLUG-IN Steer IN TWO MOTION PLANNING SCHEMES.

The first path planning scheme works for any small-time controllable system. Introduced in [17] it consists in approximating a collision-free (holonomic) path by a sequence of collision-free admissible ones. It only requires a steering method accounting for small-time controllability (Definition Section III).

A. Approximation of a holonomic path

Geometric planner: The first step is to find a *geometric path*, that is a collision free path that does not take into account the nonholonomic constraints. The car is viewed as a polygon moving freely in translation and rotation in \mathbf{R}^2 among obstacles. The configuration space of this system is then $\mathbf{R}^2 \times S^1$. Numerous techniques are available to address the motion planning problem in that case [16]. Among them we chose the “distributed representation approach” [2] that leads to resolution-complete algorithms (such algorithms are guaranteed to find a solution when a solution exists at a given resolution when modeling the search space by a grid). This algorithm is based on the construction of a potential field over the configuration space, the global minimum of which is the goal configuration. This potential field is built from two potential fields in the plane applied to two control points of the robot. Then the algorithm consists of an alternating sequence of gradient descent and a procedure filling the potential wells. Figure 4 shows an example of a path computed using this method.

Approximation step Let us denote by Γ_{hol} the geometric path computed in $\mathbf{R}^2 \times S^1$ by the previous step. From now on, we consider Γ_{hol} as a path in \mathcal{C} by setting $\zeta = 0$.

The approximation step recursively decomposes Γ_{hol} as follows. A configuration X is chosen in the middle of Γ_{hol} . This configuration is connected to X^{start} and X^{goal} using Steer , generating two feasible sub-paths. Collision with obstacles and the curvature constraint $|\zeta| < \zeta_{max}$ are checked along these sub-paths. If one of these constraints is violated, the corresponding sub-path is discarded and a new subgoal is chosen on Γ_{hol} between the beginning and the end of the discarded sub-path.

Figure 4(right) shows the result of the approximation scheme performed on the holonomic path of Figure 4(left). All the computation are performed in a few seconds.

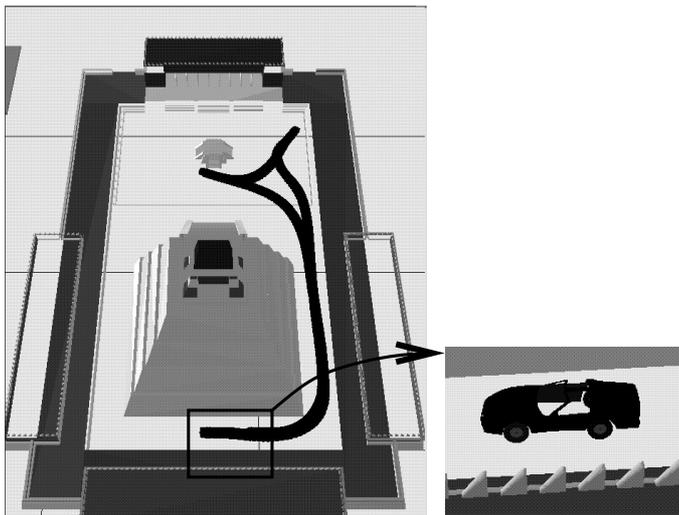


Fig. 5. A path computed by Move3D. The environment is a Mayan city with a pyramid. The car has to make a U-turn in a constrained corridor. Initial configuration is shown on the right. Final configuration is the same with opposite direction. This problem necessarily requires a long detour.

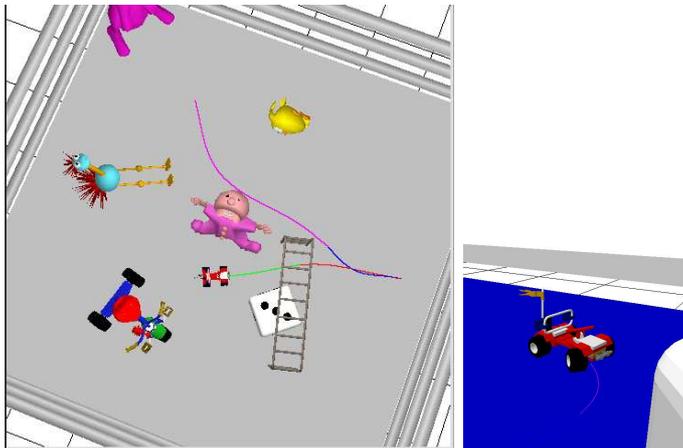


Fig. 6. A path computed for a car-like toy using Move3D.

B. Probabilistic roadmap approach

We have plugged our local method in Move3D, a generic platform for path planning [27]. Move3D can solve path planning problems for any system as long as a geometric description of the system and a local steering method is provided. We have implemented the steering method defined in Section III within Move3D. Move3D plans path using the probabilistic roadmap approach [13]. Free configurations are randomly picked. A roadmap is built by connecting to each other configurations between which the steering method returns a local path without collision. A path planning problem is solved once the initial and goal configurations lie in the same connected component of the roadmap.

Figures 5 and 6 show paths computed for a car by Move3D. The maximal steering angle ζ_{max} is 30 degrees in both cases.

C. Convergence and completeness

The convergence of the approximation step is guaranteed to finish in finite time as soon as the holonomic path belongs to an open domain of the collision-free configuration space and the steering method accounts for small-time controllability as defined above. The completeness of the algorithm thus inherits from the completeness of the geometric planner: it is resolution complete.

The probabilistic roadmap approach is probabilistically complete (i.e. the probability of finding a path if one exists tends toward 1 when the searching time increases) if the steering method accounts for small-time controllability.

Smoothing step: Both planning algorithm provide a sequence of elementary admissible paths computed by Steer. This sequence usually include useless maneuvers and detours. A smoothing step tries to connect pairs of configurations randomly chosen on the path using the steering method to shorten the first solution path.

Remark on optimal paths The minimal length paths for the system 2 have been characterized by Reeds and Shepp [23]. This result is proven in the absence of obstacles. Adding obstacles give rise to a challenging problem: solutions exist via dynamic programming approaches [3], approximated approaches [26] or for special classes of obstacles [1], [5]. All these work do not consider any constraint on the continuity of the path curvature. Computing minimal length paths for the system 1 remains today an open problem even in the absence of obstacle [28]. Therefore the paths computed by the algorithm presented in this paper are not optimal. We just argue that they are satisfactory from a practical point of view.

V. CONCLUSION

A path for car-like robot is a finite sequence of curves linking cusp configurations. Between two cusps the curves should be sufficiently smooth to allow non zero velocity at any point. In other words, the curvature should be continuous on these curves. The main purpose of this paper is to propose an efficient steering method for a car-like vehicle that computes such piecewise smooth paths. Moreover the proposed steering method has been integrated within two motion planning schemes. In the approximation scheme, the global solution path of Figure 4 has been computed within a few seconds. In the probabilistic approaches (Figures 5 and 6) the solution paths have been computed within a few seconds after a pre-processing time of a few minutes.

The main idea underlying the method proposed in this paper is to consider the car as a four-dimensional system. In such a way the constraint on the steering angle is treated as an obstacle. The approach avoids numerical issues such as the one arising in previous methods based on clothoids.

Acknowledgment We are grateful to Elodie Sims for providing the picture of Figure 6. Elements of environments of Figures 5 (Mayan city) and 6 come from <http://www.3dcafe.com>.

REFERENCES

- [1] P.K. Agarwal, P. Raghavan and H. Tamaki, "Motion planning for a steering-constrained robot through moderate obstacles," *ACM Symp. on Computational Geometry*, 1995.
- [2] J. Barraquand and J.C. Latombe, "Robot motion planning: a distributed representation approach," *Int. Journal of Robotics Research*, 1991.
- [3] J. Barraquand and J.-C. Latombe, "On non-holonomic mobile robots and optimal maneuvering," *Revue d'Intelligence Artificielle*, Vol 3 (2), pp. 77-103, 1989.
- [4] J. Barraquand and J.C. Latombe, "Nonholonomic multibody mobile robots: controllability and motion planning in the presence of obstacles," *Algorithmica*, Springer Verlag, Vol 10, pp. 121-155, 1993.
- [5] J.D. Boissonnat and S. Lazard, "A polynomial-time algorithm for computing a shortest path of bounded curvature amidst moderate obstacle," *ACM Symp. on Computational Geometry*, 1996.
- [6] G. Campion, G. Bastin and B. d'Andréa-Novel, "Structural properties and classification of kinematic and dynamic models of wheeled mobile robots," *IEEE Trans. on Robotics and Automation*, Vol 12 (1), 1996.
- [7] L. E. Dubins, "On curves of minimal length with a constraint on average curvature and with prescribed initial and terminal positions and tangents," *American Journal of Mathematics*, Vol 79, pp. 497-516, 1957.
- [8] P. Ferbach, "A method of progressive constraints for nonholonomic motion planning," *IEEE Int. Conf. on Robotics and Automation*, pp. 2929-2955, Minneapolis, 1996.
- [9] S. Fleury, P. Souères, J.P. Laumond and R. Chatila, "Primitives for smoothing mobile robot trajectories," *IEEE Transactions on Robotics and Automation*, Vol 11 (3), pp. 441-448, 1995.
- [10] M. Fliess, J. Lévine, P. Martin and P. Rouchon, "Flatness and defect of non-linear systems: introductory theory and examples," *Int. Journal of Control*, Vol 61 (6), pp. 1327-1361, 1995.
- [11] T. Fraichard, "Dynamic trajectory planning with dynamic constraints: a 'state-time space' approach," *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 1393-1400, Yokohama, 1993.
- [12] Y. Kanayama and N. Miyake, "Trajectory generation for mobile robots," *Robotics Research*, Vol 3, MIT Press, pp. 333-340, 1986.
- [13] L. Kavraki, P. Švestka, J.-C. Latombe and M.H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces", *IEEE Transactions on Robotics and Automation*, vol 12, n4, 1996.
- [14] F. Lamiroux, S. Sekhavat and J.-P. Laumond, "Motion Planning and Control for Hilare Pulling a Trailer", *IEEE Transactions on Robotics and Automation*, Vol 15 (4), 1999.
- [15] J.C. Latombe. A fast path planner for a car-like indoor mobile robot. *9th Nat. Conf. on Artificial Intelligence, AAAI, Anaheim, CA*, pages 659-665, July 1991.
- [16] J.C. Latombe. *Robot Motion Planning*. Kluwer Academic Publishers, 1991.
- [17] J.P. Laumond, P. Jacobs, M. Taïx, and R. Murray. A motion planner for non holonomic mobile robots. *IEEE Transactions on Robotics and Automation*, 10(5):577-593, 1994.
- [18] J.P. Laumond. Feasible trajectories for mobile robots with kinematic and environment constraints. *International Conference on Intelligent Autonomous Systems, Amsterdam*, 1986.
- [19] J.P. Laumond, "Controllability of a Multibody Mobile Robot," *IEEE Transactions Robotics and Automation*, pp. 755-763, Vol 9 (6), 1993.
- [20] J.P. Laumond, S. Sekhavat and F. Lamiroux. Guidelines in non-holonomic motion planning. *Robot Motion Planning and Control* (J.P. Laumond Ed.), LNCIS 228, Springer Verlag, 1998 (available on <http://www.laas.fr/~jpl>).
- [21] B. Mirtich and J. Canny. Using skeletons for nonholonomic motion planning among obstacles. *IEEE International Conference on Robotics and Automation, Nice (France)*, pages 2533-2540, 1992.
- [22] R.M. Murray and S. Sastry, "Steering nonholonomic systems using sinusoids," *IEEE Int. Conf. on Decision and Control*, pp. 2097-2101, 1990.
- [23] J. A. Reeds and R. A. Shepp, "Optimal paths for a car that goes both forward and backwards," *Pacific Journal of Mathematics*, 145 (2), pp. 367-393, 1990.
- [24] A. Scheuer and T. Fraichard, "Collision-free and continuous-curvature path planning for car-like robots," *IEEE International Conference on Robotics and Automation*, Albuquerque, 1997.
- [25] S. Sekhavat and J.-P. Laumond, "Topological property for collision-free nonholonomic motion planning: the case of sinusoidal inputs for chained form systems," *IEEE Transaction on Robotics and Automation*, Vol. 14 (5), 1998.
- [26] J. Sellen, "Approximation and decision algorithms for curvature-constrained path planning: a state-space approach," *Robotics: the Algorithmic Perspective*, P.K. Agarwal et al (Eds), A.K. Peters Pub., 1998.
- [27] T. Siméon, J.-P. Laumond, and F. Lamiroux, "Move3D: a generic platform for path planning", submitted to *International Symposium on Assembly and Task Planning*, Fukuoka, Japan, May 2001.
- [28] P. Souères and J.-D. Boissonnat, "Optimal trajectories for non-holonomic mobile robots," *Robot Motion Planning and Control* (J.P. Laumond Ed.), LNCIS 228, Springer Verlag, 1998 (available on <http://www.laas.fr/~jpl>).
- [29] P. Švestka and M. Overmars. Coordinated motion planning for multiple car-like robots using probabilistic roadmaps. *IEEE International Conference on Robotics and Automation, Nagoya, Aichi (Japan)*, 1995.
- [30] D. Tilbury, R. Murray and S. Sastry, "Trajectory generation for the n-trailer problem using Goursat normal form," *IEEE Transaction on Automatic Control*, Vol. 40, 1995.