Exact Motion Planning for Tractor-Trailer Robots*

Petr Švestka Jules Vleugels

Department of Computer Science, Utrecht University
P.O. Box 80.089, 3508 TB Utrecht, the Netherlands
Email: {petr,jules}@cs.ruu.nl

Abstract

A tractor-trailer robot consists of a carlike tractor towing a passive trailer. Due to its highly nonholonomic nature, the kinematics of this type of robot are complicated and difficult to compute. We present exact closed-form solutions for the kinematic parameters of a tractor-trailer robot and use them to construct an exact and efficient motion planner in the absence of obstacles. This local planner can be employed in a probabilistic global planner, which allows us to plan motions in the presence of obstacles.

1 Introduction

The motion planning problem is well-known in the field of robotics. Its objective is to find collision-free paths for a robot moving amidst a set of obstacles. For *free-flying* robots, i.e., robots without restrictions on the motions they can perform, the motion planning problem is typically solved by computing a path in the free configuration space; such a path corresponds to a feasible free path in the workspace. This approach however is not possible for some motion planning problems, such as problems involving nonholonomic systems. In this case the robot can perform only restricted motions even in the absence of obstacles, which means that a path in the free configuration space is not necessarily feasible in the workspace. We refer to Laumond [5] for an introduction to nonholonomic motion planning; a good overview of the general motion planning problem was given by Latombe [4].

This paper investigates a particular and well-known nonholonomic system: the *tractor-trailer robot*. Informally, this robot consists of a carlike tractor towing a passive trailer. The tractor can perform motions that are similar to those of a car: it drives forwards or backwards while possibly steering left or right. The trailer follows the path that is dictated by the motion of the tractor. In Section 2 we first formally describe the tractor-trailer robot and discuss its nonholonomic constraints. We give closed-form solutions for its kinematic parameters and show how we can explicitly express its configuration at any given time instant. This makes it possible to construct an exact motion planning algorithm for the tractor-trailer robot in the absence of obstacles, which is described in Section 3. Then in Section 4 we briefly discuss how this local planner can be employed in a probabilistic framework, which allows us to plan the motion of a tractor-trailer robot in the presence of obstacles. Finally, in Section 5 we give some conclusions and indicate open problems and directions of future work.

1.1 Related work

Due to the complex nature of the tractor-trailer robot, most work has focussed on obtaining approximations for its kinematics. The field of Control Theory has produced some recent results [2, 9, 10, 11] on the n-trailer system (a tractor towing n passive trailers) of which the tractor-trailer robot is a special case, but these approaches are not exact (though a configuration can be

*This research was partially supported by ESPRIT Basic Research Action No. 6546 (project PROMotion).

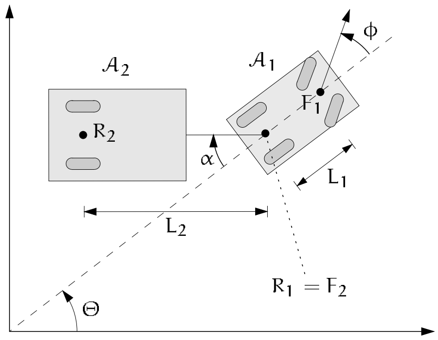

Figure 1: A model for the tractor-trailer robot.

approximated arbitrarily close) and/or not efficient. Furthermore, they tend to have problems with incorporating (possibly complex) obstacles. Barraquand and Latombe [1] propose a heuristic brute-force approach to motion planning for tractor-trailer robots. It consists of heuristically searching a graph whose nodes are small axis-parallel cells in configuration space. Two such cells are connected in the graph if there exists a feasible path between two configurations in the respective cells. The main drawback of their method is that when the heuristics fail it requires an exhaustive search in the discretized configuration space. Furthermore, the resulting path is inexact because the solution to the nonholonomic constraints is approximated numerically; this implies that the goal configuration is never reached exactly. Our closed-form solution to these constraints would transform their algorithm into an exact planner. Laumond and Siméon [6] first compute a holonomic path in the workspace in disregard of the nonholonomic constraints of the robot, and then transform it into a feasible path by means of a local method that respects the robot's nonholonomic constraints. Again, in the case of a tractor-trailer robot, an inexact method is used. Unfortunately, the local method we are about to present will not work with this approach, in the sense that the resulting global planner will lack completeness; see Section 4 for details.

2 Preliminaries

In this section we first formally describe the tractor-trailer robot; then, in Section 2.1, we derive solutions for its kinematic parameters. Due to space limitations, we will state various well-known properties of nonholonomic systems without further explanation.

Formally, a tractor-trailer robot $\mathcal{A}$ consists of two (arbitrarily shaped) solid bodies $\mathcal{A}_1, \mathcal{A}_2$ in the Euclidean plane that are connected by a revolute joint; see Figure 1. The tractor $\mathcal{A}_1$ (resp. trailer $\mathcal{A}_2$) has a front point F_1 (F_2) and a rear point R_1 (R_2) fixed to it; the distance between these points is given by L_1 (L_2), called the *length* of the tractor (trailer). We assume that $R_1 = F_2$ is the revolute joint connecting $\mathcal{A}_1$ with $\mathcal{A}_2$. The tractor can be modelled as a four-wheel front-wheel-drive vehicle, where F_1 (R_1) is the midpoint of the two front (rear) wheels. Similarly, the trailer is a two-wheel passive vehicle with R_2 as the midpoint of the wheels.

Any placement of a tractor-trailer robot can be uniquely described by a tuple (x, y, θ, α), where $(x, y) \in \mathbb{R}^2$ are the coordinates of both R_1 and F_2, $\theta \in [0, 2\pi)$ is the orientation of the tractor (given by the direction of the vector $F_1 - R_1$), and α is the orientation of the trailer relative to that of the tractor. Hence, its configuration space is homeomorphic to $\mathbb{R}^2 \times S^1 \times S^1$. The possible configurations of the tractor-trailer robot are restricted by a mechanical stop on the revolute joint connecting its two bodies. We limit the relative bend of the trailer α in absolute value to a *maximal bending angle* $\alpha_{\max} \in [0, \pi)$. This corresponds to the sharpest bend the trailer can make, expressed relative to the orientation of the tractor.

We assume that $\mathcal{A}$ moves in a plane and that the contact between each wheel and the ground is a pure rolling contact. These assumptions restrain the set of possible velocities that can be achieved by the robot as follows. For a given configuration c of the robot, let $l_1(c)$ be the line through F_1 and R_1, and $l_2(c)$ the line through F_2 and R_2. Now exactly those velocities of $\mathcal{A}$ are possible for which

1. the direction of R_1's velocity points (forwards or backwards) along $l_1(c)$,
2. the direction of R_2's velocity points along $l_2(c)$, and
3. the angle ϕ between $l_1(c)$ and the direction of F_1's velocity is in absolute value bounded by a constant $\phi_{\max} \in [0, \frac{\pi}{2})$, which we refer to as the tractor's *maximal steering angle*. Intuitively, this defines the sharpest turns the tractor is allowed to make.

These restrictions on the tractor-trailer's motions give rise to two nonholonomic constraints that limit the set of achievable velocities at any point in configuration space.

2.1 Nonholonomic constraints

The velocity constraints described in the previous section can be expressed by a set of equations involving the derivatives of the robot's kinematic parameters $x(t)$, $y(t)$, $\theta(t)$, and $\alpha(t)$. These equations can easily be derived from the geometry of the robot. If we denote the velocity of the tractor's rear point R_1 at any time instant t as $v(t) \in \mathbb{R}$ and its steering angle as $\phi(t) \in [-\phi_{\max}, \phi_{\max}]$, then

$$
\begin{aligned}
x'(t) &= v(t)\cos\theta(t) && (1)\\
y'(t) &= v(t)\sin\theta(t) && (2)\\
\theta'(t) &= \frac{v(t)}{L_1}\tan\phi(t) && (3)\\
\alpha'(t) &= -\frac{v(t)}{L_2}\sin\big(\alpha(t)-\theta(t)\big) && (4)
\end{aligned}
$$

The constraints expressed by these equations are of nonholonomic nature, which means that in general it is impossible to integrate them. If however we assume the functions v and ϕ constant, say $v(t) = v_0$ and $\phi(t) = \phi_0$, it turns out to be possible to integrate the equations. Provided[1] $\phi_0 \neq 0$, integration of the first three equations gives the solutions shown by Equations (5) to (7) in Table 1 for arbitrary constants C_x, C_y, and C_θ. Equation (4) is a differential equation in $\alpha(t)$,

$$
\begin{aligned}
x(t) &= C_x + \frac{L_1}{\tan\phi_0}\left(\sin\left(\theta(0) + \frac{v_0 t}{L_1}\tan\phi_0\right) - \sin\theta(0)\right) && (5)\\
y(t) &= C_y - \frac{L_1}{\tan\phi_0}\left(\cos\left(\theta(0) + \frac{v_0 t}{L_1}\tan\phi_0\right) - \cos\theta(0)\right) && (6)\\
\theta(t) &= C_\theta + \frac{v_0 t}{L_1}\tan\phi_0 && (7)\\
\alpha(t) &= 2\arctan\left((\tan\phi_0 L_2)^{-1}\left(\tan\left(\frac{-v_0\sqrt{\xi}\,(t L_1 + C_\alpha v_0 \tan\phi_0)}{2 L_1^2 L_2}\right)\sqrt{\xi} - L_1\right)\right) && (8)
\end{aligned}
$$

where

$$\xi = \tan^2\phi_0 \, L_2^2 - L_1^2 \quad (9)$$

Table 1: Solutions for the kinematic parameters of a tractor-trailer robot.

[1] The case for $\phi_0 = 0$ is easy to solve and has been omitted. The corresponding motion (a *tractrix*) has first been utilized by Laumond and Siméon [6] to constructively prove full controllability of tractor-trailer robots.

and was previously considered impossible to integrate. With the aid of the *Maple* computer algebra system we succeeded in deriving the solution given by Equation (8) in Table 1. (A derivation of this solution is beyond the scope of this paper and has been omitted.) Again the constant C_α can be chosen arbitrarily. We want to use these equations to compute $\mathcal{A}$'s path from some given start configuration $(x_0, y_0, \theta_0, \alpha_0)$, assuming constant velocity v_0 and a constant steering angle ϕ_0. By solving the equations $x(0) = x_0$, $y(0) = y_0$, $\theta(0) = \theta_0$, and $\alpha(0) = \alpha_0$ we obtain the results given by the equations shown in Table 2.

$$C_x = x_0 \qquad (10)$$

$$C_y = y_0 \qquad (11)$$

$$C_\theta = \theta_0 \qquad (12)$$

$$C_\alpha = 2\,\frac{\arctan\Big(\xi^{-1/2}\big(-\tan\left(\tfrac{1}{2}\,\alpha_0\right)\,\tan\phi_0\,L_2 - L_1\big)\Big)\,L_1^2\,L_2}{\sqrt{\xi}\,v_0^2\,\tan\phi_0} \qquad (13)$$

Table 2: Constants for the path of a tractor-trailer robot starting from a configuration $(x_0, y_0, \theta_0, \alpha_0)$, assuming constant velocity v_0 and constant steering angle ϕ_0.

Using these constants, we can express $\mathcal{A}$'s configuration at time t under the assumptions of constant steering angle and velocity as $(x(t), y(t), \theta(t), \alpha(t))$, as given by Equations (5) to (8).[2] To our knowledge, this is a new result that makes it possible to construct an exact motion planner for the tractor-trailer robot.

3 An exact motion planner

In this section we use the results of the previous section to devise an exact motion planning algorithm for a tractor-trailer robot in the absence of obstacles. To this end we introduce two special constructs: rotational motions and translational motions, both of which leave the relative orientation of the tractor with the trailer constant. Next we introduce motions that are used to move the robot from a translational motion to a rotational motion and vice versa, called stretches and bends.

We define a *path* for a tractor-trailer robot to be a continuous (except for discontinuities of θ and α in 0 and 2π) function of type $[0,1] \to C$, mapping time t to configurations $(x(t), y(t), \theta(t), \alpha(t))$. A path is said to be *feasible* for a tractor-trailer robot $\mathcal{A}$ if and only if it respects $\mathcal{A}$'s constraints.

3.1 Simple path constructs

Recall that α represents the orientation of the trailer relative to that of the tractor. We call a path $p : t \mapsto (x(t), y(t), \theta(t), \alpha(t))$ *α-stable* if and only if α is a constant function. This is the case if the tractor's instantaneous center of rotation Ω_1 (which is uniquely defined by its steering angle) coincides with the trailer's instantaneous center of rotation Ω_2 (uniquely defined by its bending angle). When performing a motion along an α-stable path, the tractor-trailer robot behaves as if it were a single solid body.

Lemma 3.1 *A feasible path $p : t \mapsto (*, *, \theta(t), \alpha(t))$ is α-stable with $\alpha(t) = \alpha_0$ if and only if for all $t \in [0,1]$ such that $v(t) \neq 0$:*

$$\phi(t) = -\arctan\left(\frac{L_1\,\sin\alpha_0}{L_2}\right).$$

[2] Note that ξ evaluates to negative values in $\mathbb{R}$ for $|\phi_0| < \arctan(L_1/L_2)$. The equations for $\alpha(t)$ and C_α compute the square root of this expression and therefore have to be evaluated using complex arithmetic.

Proof. Due to space limitations, we only prove the reverse implication. To this end, assume that $p : t \mapsto (x(t), y(t), \theta(t), \alpha(t))$ is a feasible and α-stable path with $\alpha(t) = \alpha_0$, and that $v(t) \neq 0$. Then $\alpha'(t) = 0$, and therefore

$$\frac{v(t)}{L_2} \sin \alpha_0 = -\frac{v(t)}{L_1} \tan \phi(t).$$

Solving $\phi(t)$ from this equation gives

$$\phi(t) = -\arctan\left(\frac{L_1 \sin \alpha_0}{L_2}\right) \tag{14}$$

as desired. □

As a result of this theorem, Equation (14) shows that for every bending angle $\tilde{\alpha}$ there exists a constant *induced steering angle* $\phi_{\text{stable}}(\tilde{\alpha})$, such that any feasible path from a configuration $(*, *, *, \tilde{\alpha})$ with $\phi(t) = \phi_{\text{stable}}(\tilde{\alpha})$ is α-stable.

For a given bending angle $\tilde{\alpha}$ it is possible that $|\phi_{\text{stable}}(\tilde{\alpha})| > \phi_{\max}$, i.e., the resulting path is not feasible because it violates the constraints on the robot. It can be shown that in this case no feasible α-stable path with bending angle $\tilde{\alpha}$ exists.

3.1.1 Rotational paths

The motion described by a feasible α-stable path $p : t \mapsto (x(t), y(t), \theta(t), \alpha_0)$ with $\alpha_0 \neq 0$ is a rotation around the shared instantaneous center of rotation $\Omega(p)$ of the tractor and the trailer. We refer to such α-stable paths with non-zero bending angle as *rotational paths.* The *radius* of P, denoted by $\text{rad}(p)$, is defined as the (constant) distance between R_1 and $\Omega(p)$. We will use rotational paths as 'primitive building blocks' of more complex path constructs. Formally, for configurations $a, b \in \mathcal{C}$ we define a function $r_{a,b} : [0, 1] \mapsto \mathcal{C}$ as $r_{a,b}(t) = (x(t), y(t), \theta(t), \alpha(t))$, such that

1. $r_{a,b}(0) = a$, $r_{a,b}(1) = b$, and for all $t \in (0, 1) : r_{a,b}(t) \neq b$,
2. $\alpha(t) = \alpha_0$ and $\phi(t) = \phi_{\text{stable}}(\alpha_0)$, and
3. $v(t) = v_0$.

In other words, $r_{a,b}(t)$ describes the shortest rotational path from a to b with constant velocity.

A tractor-trailer robot $\mathcal{A}$ has a *minimal turning radius* $r_{\min} \in \mathbb{R}$ such that there exist feasible rotational paths with radius $r_{\min}$ but not with smaller radii. Rotational paths with radii less than $r_{\min}$ force $\mathcal{A}$ to violate the constraints on either its steering angle or its bending angle, and are hence not feasible. We refer to rotational paths with minimal turning radius as *maximally curved* rotational paths; the corresponding ϕ and α are denoted as $\phi_{\min}$ and $\alpha_{\min}$.

3.1.2 Translational paths

The motion described by a feasible α-stable path with a constant bending angle of 0 is a *translation* along the line through R_1 and R_2. We therefore refer to α-stable paths with bending angle equal to 0 as *translational paths.*

Like rotational paths, translational paths will be used as primitive building blocks for more complex paths. We define a function $t_{a,b} : [0, 1] \mapsto \mathcal{C}$ as $t_{a,b}(t) = (x(t), y(t), \theta(t), \alpha(t))$, such that:

1. $x(t) = x_a + t \cdot (x_b - x_a)$, $y(t) = y_a + t \cdot (y_b - y_a)$,
2. $\theta(t) = \theta_0$, and
3. $\alpha(t) = 0$;

thus $t_{a,b}(t)$ gives the constant-velocity translational path from a to b.

3.1.3 Stretches and bends

In this section we introduce two additional primitive path constructs: stretches and bends. They differ from rotational and translational motions in that the bending angle of the trailer does not remain constant during these motions. To compute these paths in an exact way, the solution for $\alpha(t)$ given by Equation (8) in Section 2.1 is indispensable.

Definition 3.2 *A* stretch *is a feasible path with $\alpha(0) \neq 0$, $\alpha(1) = 0$, $v(t) = v_0$, and $\phi(t) = \phi_0$ where $|\phi_0| = \phi_{\max}$.*

In other words, a stretch moves the robot from a configuration with nonzero bending angle to one with zero bending angle (the way one stretches its arm). A stretch can thus be used to move the robot from a rotational to a translational configuration. We refer to stretches with $v_0 > 0$ as *forward stretches*, and such with $v_0 < 0$ as *backward stretches*. A *partial stretch* is a connected sub-path of a stretch, in other words: if $s : [0, 1] \to \mathcal{C}$ is a stretch, then any function s' identical to s but defined only on a closed subinterval of $[0, 1]$ is a partial stretch.

A *bend* is defined as the inverse of a stretch; it moves the robot from a translational to a rotational configuration. Similarly, a partial bend is the inverse of a partial stretch.

We now state some properties of stretches (and of bends, implicitly) without proof.

Lemma 3.3 *For any configuration c the forward stretch p with $p(0) = c$ is uniquely defined, and $\alpha(0) > 0 \Leftrightarrow \phi(t) = \phi_{\max}$.*

Lemma 3.4 *For any configuration c with $\phi_{\text{stable}}(\alpha(0))$ in absolute value less or equal to $\phi_{\max}$, the backward stretch p with $p(0) = c$ is uniquely defined, and $\alpha(0) > 0 \Leftrightarrow \phi(t) = -\phi_{\max}$.*

In order to minimize the length of a stretch, we have the robot move with maximal steering angle while stretching. Note that this cannot violate the constraints on the trailer's bending angle since it decreases monotonically (in absolute value) during a stretch. To compute the forward stretch p starting at some configuration c, i.e., $p(0) = c$, we take $\phi_0 = \text{sign}(\alpha_0)\phi_{\max}$ and then solve v_0 from Equation (8) in Section 2.1 by substituting $\alpha(1) = 0$. This gives the following solution:

$$v_0 = 2L_1 L_2 \left(\arctan \eta - \arctan\left(\xi^{-1/2} L_1\right)\right) \tag{15}$$

where

$$\eta = \xi^{-1/2} \tan \frac{\alpha(0)}{2} \tan \phi_0 \, L_2 + L_1 \tag{16}$$

Substitution of the above expressions for ϕ_0 and v_0 in the Equations (5) to (8) of Section 2.1 gives us a definition of the forward stretch from c. A backward stretch (if it exists) can be computed in the same way, but taking $\phi_0 = -\text{sign}(\alpha_0)\phi_{\max}$.

3.2 A local planner

Having defined the various simple constructs, we now discuss how they can be concatenated to form feasible paths between a given pair of configurations and in the absence of obstacles.

For now, consider only the tractor of the tractor-trailer robot, which by itself is a carlike robot. A feasible path connecting two given configurations of the robot can be constructed [3, 4, 8] by concatenating three sub-paths: circular arcs around both configurations, and a straight line segment touching both arcs. The circular arcs are centered at the instantaneous centers of rotation of the robot at the respective configurations. The radii of these arcs determine the effective turning radius of the robot and can be chosen arbitrarily as long as the robot's constraints are respected; in order to reduce the length of the resulting path we take them to be equal to the tractor's minimal turning radius $r_{\phi_{\max}}$ (induced by its maximal steering angle). The resulting paths will thus be maximally curved.

Since the bending angle of the trailer remains constant during the rotational and translational motions described in the previous section, the tractor-trailer can effectively be treated as a carlike

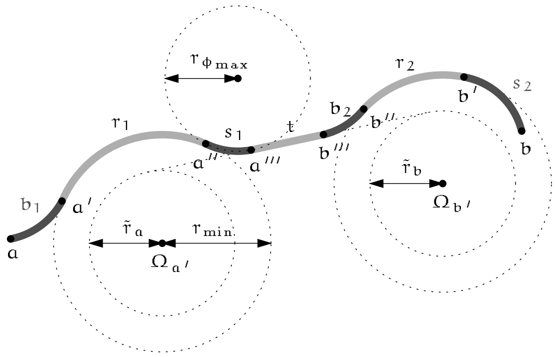

Figure 2: Constructing a feasible path for a tractor-trailer robot.

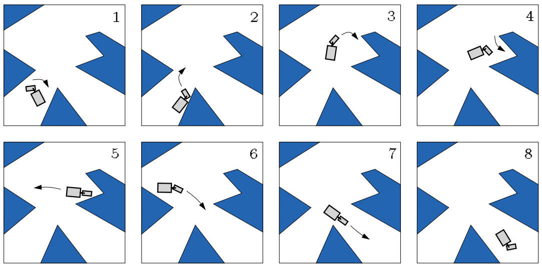

Figure 3: A path for a tractor-trailer robot in the presence of obstacles.

robot during these motions (with a turning radius that depends on both the tractor and the trailer, as described in Section 3.1.1). This means that the we could connect a pair of configurations by concatenating a translational path and two rotational paths. Unfortunately, this would cause the bending angle of the trailer to jump from nonzero (during the rotational motion) to zero (during the translational motion) and vice versa. Hence the resulting path transitions would not be feasible. This problem can be solved by concatenating the two types of paths through the stretches and bends defined in the previous section (one such construction is shown in Figure 2). Furthermore, additional (partial) stretches and bends will have to be performed to move the robot from the initial configuration and to the final configuration. Any path will thus consists of a (partial) bend b_1, followed by a rotational motion r_1, a stretch s_1, a translational motion t, a bend b_2, another rotational motion r_2, and a final (partial) stretch s_2—some of which can have zero length. A more detailed description of the path construction can be found in the full paper.

4 Dealing with obstacles

The motion planning algorithm described in the previous section is only a local planner that constructs paths in the absence of obstacles. A possible way to deal with obstacles in the workspace is by employing this local planner in the learning paradigm proposed by Overmars and Švestka [7]. Briefly, their planner incrementally builds up knowledge about the environment by trying to connect randomly generated configurations through a local planner. The key issue in this is that the local planner is not required to always find a path—it is allowed to fail for some configurations. This means that a local planner that simply disregards the obstacles can be used and is a viable choice [7]. Therefore we try to connect given configurations by constructing a path without taking the obstacles into account. If the robot does not intersect any obstacles along this path, the planner has successfully connected the configurations; it returns failure otherwise. Although this approach seems valid, our local planner does not satisfy the so-called *ϵ-reachability property* [8] that would guarantee probabilistic completeness of the global planner. (A lengthier discussion can be found in the full paper.) We are currently investigating a different path construction to meet this property.

The algorithm described in this paper has been implemented in C++ on a Silicon Graphics Indigo II workstation. Although we currently have only a preliminary version, first experimental results indicate that the method works and is fast. For example, Figure 3 shows a path that was computed (and subsequently smoothed) by the program in approximately 1.5 seconds; the path is indicated by a number of snapshots of the robot. We expect that the program can be optimized to run substantially faster.

5 Conclusions and future work

In this paper we presented a closed-form solution for the kinematic parameters of a tractor-trailer robot. We described an exact motion planner for a tractor-trailer robot in the absence of obstacles based on this solution. To our knowledge, this is the first exact and efficient algorithm that solves this problem. This local planner has been integrated into a probabilistic global planner, resulting in an efficient motion planner for tractor-trailer robots in the presence of obstacles. Unfortunately, the way in which we currently construct paths does not satisfy a property that would guarantee probabilistic completeness. Efficiently constructing paths that have this property remains an interesting open problem.

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] A. W. Divelbiss and J. Wen. A global approach to nonholonomic motion planning. In *Proc. IEEE Control and Decision Conf.*, pages 1597–1602, Tucson, Arizona, 1992.

[3] S. Fortune and G. Wilfong. Planning constrained motion. In *Proc. 20th Annu. ACM Sympos. Theory Comput.*, pages 445–459, 1988.

[4] J.-C. Latombe. *Robot Motion Planning*. Kluwer Academic Publishers, Boston, 1991.

[5] J.-P. Laumond. Singularities and topological aspects in nonholonomic motion planning. In Zexiang Li and J.F. Canny, editors, *Nonholonomic Motion Planning*, chapter 5, pages 149–199. Kluwer Academic Publishers, 1993.

[6] J.-P. Laumond and T. Siméon. Motion planning for a two degrees of freedom mobile robot with towing. Technical Report 89148, LAAS/CNRS, Toulouse, France, April 1989.

[7] M. H. Overmars and P. Švestka. A probabilistic learning approach to motion planning. In *Proc. The First Workshop on the Algorithmic Foundations of Robotics*. A. K. Peters, Boston, MA, 1994.

[8] P. Švestka and M. Overmars. Motion planning for car-like robots using a probabilistic learning approach. Technical Report UU-CS-1994-33, Dept. Comput. Sci., Utrecht Univ., Utrecht, the Netherlands, August 1994.

[9] D. Tilbury, J.-P. Laumond, R. Murray, S. Sastry, and G. Walsh. Steering car-like systems with trailers using sinusoids. In *Proc. 1992 IEEE Int. Conf. Robotics and Automation*, pages 1993–1998, May 1992.

[10] D. Tilbury, R. Murray, and S. Sastry. Trajectory generation for the n-trailer problem using gourmat normal form. Technical Report UCB/ERL M93/12, Univ. California, Berkeley, CA, February 1993.

[11] D. Tilbury, O. Sørdalen, L. Bushnell, and S. Sastry. A multi-steering trailer system: Conversion into chained form using dynamic feedback. Technical Report UCB/ERL M93/55, Univ. California, Berkeley, CA, July 1993.