

# SOME STEPS TOWARDS AUTONOMOUS CARS

Ch. Laugier \* S. Sekhavat \* F. Large \*  
J. Hermosillo \* Z. Shiller \*\*

\* *INRIA Rhône-Alpes, Grenoble, France*

Email: *{laugier,sekhavat,large,hermosil}@inrialpes.fr*

\*\* *UCLA California, USA - Email shiller@earthlink.net*

Abstract: Developing new Intelligent Transportation Systems which take into consideration the social-economical, environmental, and safety factors of the modern human society, is one of the grand challenges of the 21th century. This paper presents the motion autonomy capabilities which have been developed for future cars in the scope of the French “Praxitele” and “Automated Road” projects. These new capabilities relies onto effective solutions that we have developed for contributing to solve three main problems: autonomous maneuvering for a car-like vehicle using the concept of *Sensor-Based Maneuver*, planning and controlling the motions of a double-steered vehicle (the CyCab) using the concept of *Differential Flatness*, and obstacle avoidance in a dynamic environment using the concept of *Non-Linear Velocity Obstacle*. Experimental results obtained with real vehicles are also presented and discussed.

Keywords: Intelligent Transportation Systems, Autonomous Vehicles.

## 1. INTRODUCTION

### 1.1 Context and motivations

Developing new Intelligent Transportation Systems (*ITS*) which take into consideration the socio-economical, environmental, and safety factors of the modern society, is one of the grand challenges of the 21th century. Several national and international projects have been launched for developing such technologies during the last 10 years: e.g. the European Project *Prometheus* whose purpose was to develop driving assistance techniques for cars, the European project *Chauffeur* aimed at developing platooning techniques for trucks, the French national programme *Praxitele* whose objective was to develop a new urban transport system based on a fleet of electric self-service cars, the *Path* project in USA aimed at developing Automated Highway System technologies, and the *AHS* (Advanced Cruise-Assist Highway System) project in Japan aimed at developing new technologies for increasing safety.

Various technologies for autonomous driving of road vehicles have been Developed within the scope of the above-mentioned projects, and some of these technologies can now be considered as almost operational. Large scale technical demonstrations have also been successfully conducted in Japan (1996 and 2000), in USA (1997), and in Europe (1994 and 1998). However, it is still unclear how and when such technologies could be introduced in everybody’s car, because some socio-economical, political, and liability problems are not well understood yet. In Japan, the ASHRA consortium (Advanced Cruise-Assist Highway System Research Association) has established in relation with the government, an agenda for deploying the *AHS* technologies on a large part of the Japanese highways network for the next 15 years, while taking into account all the abovementioned factors. In Europe, we believe that such technologies, would be best used in a first step for solving some of the increasing difficulties of urban transport (mainly traffic congestion, lack of parking areas, pollution), by offering new alternatives to

the private automobile and to mass transport; several projects have been launched in this spirit, e.g. the *Praxitele* project in France, the *Serpentine* project in Switzerland, the *Park Shuttle* project in Netherlands, the European *CyberCars* project involving several industrial companies and cities in Europe, etc.

All these projects relies onto the same basic idea: reducing the use of the private automobile in downtown areas, by offering new modern public transportation systems which are both convenient and “sustainable”. Our approach consists in developing the concept of “*Automated Public Car*”, which roughly consists in combining self-service cars technology (i.e. a fleet of cars which are available for subscribers in specific “stations” for use in restricted areas) and *PRT* technology (i.e. small automatic vehicles moving on dedicated tracks connecting a set of “stations”, e.g. the VAL system in Paris). Such a transportation system is designed for offering “automated travels” on the routes where the flows can be high at certain periods of time, and “manually driven travels” in the areas located in the vicinity of the automated network (figure 1 illustrates). The recent developments of automated driving techniques, make it now potentially feasible to move cars autonomously on paved roads between stations. If the speed is limited (to say 15 km/h), the road does not even have to be protected; this means that stations can be located on the ground level, and hence be very inexpensive compared to aboveground or underground stations needed by PRT systems. For higher speeds, it will probably be necessary in densely populated areas to have a protected network which could for example double an existing highway network.



Fig. 1. Urban Transport System involving Individual Automated Vehicles.

## 1.2 Addressed technical issues

Several technologies have still to be developed for obtaining the required motion autonomy capabilities. These capabilities includes autonomous maneuvering, controlling vehicles having various kinematic structures, and obstacle avoidance in dynamic environments.

*Motion autonomy* is certainly one of the major technical issues to be addressed for the development of

autonomous cars. This technical issue has already been widely studied in the mobile robotics literature. The state of the art on this topic shows approaches of various complexity, combining in different ways purely reactive methods with more traditional hierarchical decisional schemes. A quite classical way to solve the motion autonomy problem for a car-like vehicle moving in a partially known environment is to combine an appropriate *off-line global path/trajectory planner* with a *reactive execution controller* capable to track the nominal trajectory while avoiding collisions with unexpected obstacles; unfortunately, such an approach usually generates oscillatory movements and inconsistent behaviors resulting from the combination of the trajectory tracking and obstacle avoidance functions (see (Laugier and Fraichard, 2001) for a more complete bibliography and problem analysis). As it is described in the section 2, our approach for solving this problem in the context of autonomous cars, relies onto the combination of a novel Control and Decisional Architecture and of the concept of Sensor-Based Maneuver.

An other issue concerns the *control of new types of vehicles* especially designed for *ITS* applications. The *Cycab* designed at INRIA and currently commercialized by the Robosoft company<sup>1</sup> (see figure 2), and the  $\pi$ *Car* prototype of *IEF*<sup>2</sup> are two examples of such vehicles having appropriate kinematic structures for increasing their maneuverability. This new kinematic structure, which strongly differ from those of a car-like vehicle in that it allows the steering of both the front and the back wheels, generates new control and motion planning problems not addressed yet in the literature. In the sequel, we define a *bi-steerable car* as a car such that the deflection of its front wheels by an angle  $\phi$  induces a deflection of its rear wheels by an angle  $f(\phi)$ ; this feature not only reduces the lower bound of the turning radius of the vehicle, but it also reduces the sweeping volume of the vehicle in motion. Therefore, it enhances its maneuverability in cluttered environments. In the section 3, we describe how we have solved the steering and motion planning problems for such a vehicle.

The last important issue concerns *obstacle avoidance in a dynamic environment*. Indeed, dynamic environments represents an important and a growing segment of modern automation, in various applications like intelligent vehicles negotiating free-way traffic, automated wheel-chairs, or air and sea traffic control. Common to these applications is the need to quickly select avoidance maneuvers that avoid potential collisions with moving obstacles. A traditional approach to solve this problem consists in applying a *zero order* method, i.e. a method based on position/orientation considerations (see (Fiorini and Shiller, 1998) for an extended survey). In the section 4, we describe a *first order* approach (i.e. velocity based), allowing the avoidance of any number of moving obstacles using the concept of

<sup>1</sup> See [www.robosoft.fr](http://www.robosoft.fr)

<sup>2</sup> Institut d'Electronique Fondamentale, Université Paris-Sud.

*Non-linear V-Obstacle*; using this approach, the vehicle can avoid in real-time obstacles moving on arbitrary known or sensed trajectories.



Fig. 2. The INRIA Cycab vehicle.

## 2. AUTONOMOUS MANEUVERING

### 2.1 Overall control architecture

In order to provide the vehicle with the required motion autonomy capabilities, we have developed a Control and Decisional Architecture having the ability to generate *smooth and safe motions* for an autonomous vehicle, while satisfying both the task constraints (i.e. the nominal trajectory and the sensed obstacles) and the kinematic and dynamic constraints of the vehicle. This architecture includes an *off-line global Trajectory Planner*<sup>3</sup>, a decisional kernel which generates on-line appropriate *Sensor-Based Maneuvers (SBM)*, and a *Motion Controller* which makes use of a set of parameterized control programs (or *Control Experts*) to execute the required sensor-based motions.

The key idea of our approach is to plan and carry out *SBMs* where the following scheme is applied (see (Laugier and Fraichard, 2001) for more details): first, a Parameterized Motion Plan is produced by combining a *Nominal Trajectory*<sup>4</sup> with a set of *Generic Sensor-Based Maneuvers* selected from a library (according to the execution context); then, the involved motion controls are executed using some selected *Control Experts* having the ability to react in real-time to some unforeseen events. In case of a failure due to an unprocessed event, the decisional kernel of the system decides either to replan the motion or to insert in real-time a more appropriate *SBM* (which in turn is expanded into a sequence of executable local trajectories and sensing

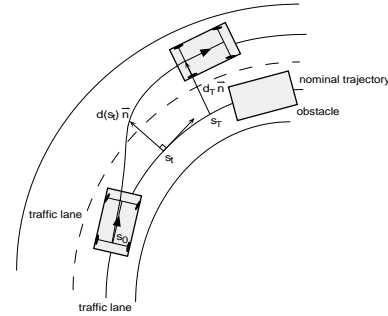


Fig. 3. Generation of smooth local trajectories to avoid an obstacle.

operations). Indeed, the reactivity of the system to unforeseen events along with the quality of the executed motions mainly relies on the *SBM* paradigm. In our approach, a *SBM* is basically a *safe and smooth motion* of the vehicle, which is executed using some predefined sensor modalities and controls; it allows the vehicle to perform in a reactive way a particular type of maneuver, while adapting the control parameters to the current execution context.

### 2.2 Lane Following/Changing

The lane following/changing *SBM* allows the vehicle to follow a nominal trajectory as closely as possible, while reacting appropriately to any unforeseen obstacle obstructing the way of the vehicle. Whenever such an obstacle is detected, the nominal trajectory is locally modified in real time, in order to avoid the collision. This local modification of the trajectory is applied in order to simultaneously satisfy a set of different constraints: collision avoidance, time constraints, kinematic and dynamic constraints. It is performed using *smooth local trajectories* allowing the vehicle to move away from the obstructed nominal trajectory, and to catch up this nominal trajectory when the obstacle has been overtaken. This type of *SBM* is executed using two complementary Control Experts: “trajectory tracking” and “lane changing”; these Control Experts have been implemented using the Kanayama trajectory tracking method (Kanayama *et al.*, 1991), and smooth local trajectories generated using appropriate quintic polynomial equations (Nelson, 1989) (cf. figure 3, see (Laugier and Fraichard, 2001) for more details).

This type of *SBM* can be used to avoid a stationary obstacle, or to overtake another vehicle. In this case, as soon as the obstacle has been detected by the vehicle (e.g. during the tracking of the nominal trajectory), the minimum space which is required for executing a lane change is calculated and compared with the current vehicle/obstacle distance. The result of this computation is used to decide which behavior to apply: avoid the obstacle, slow down or stop.

### 2.3 Parallel Parking

The purpose of the parallel parking *SBM* is to automatically park the vehicle within an unknown parking

<sup>3</sup> A trajectory represents both a geometric path (i.e. a smooth curve) and its associated velocity profile. The reader is referred to (Laugier and Fraichard, 2001) for a complete presentation of the global trajectory planner.

<sup>4</sup> A *Nominal Trajectory* is generated off-line by the global trajectory planner using a reconstructed model of the environment and a prediction of the most likely behaviors of the moving obstacles.

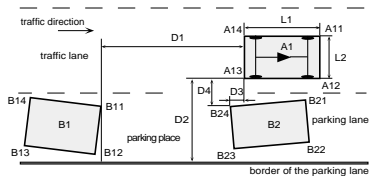


Fig. 4. Local free space and start location for parallel parking.

area. This *SBM* involves three main steps (Paromtchik and Laugier, 1996): (1) localizing a sufficient space for parking the car, (2) obtaining an appropriate starting location for the vehicle relatively to the parking place, and (3) performing the parallel parking maneuver.

During the first phase of the parallel parking *SBM* (steps 1 & 2), the vehicle moves slowly along the traffic lane, and it uses its range sensors for constructing a local map of the environment and for detecting obstacles. The local map is used to select an appropriate free space for parking the vehicle, to determine an appropriate starting position for the parking maneuver, and to compute the control parameters; the obstacles are avoided using the lane following/changing *SBM*. At the end of this phase, the system calculates the characteristics of the “local free space”, i.e. the space which is available for executing the required longitudinal and lateral displacements, figure 4 illustrates.

During parallel parking, iterative low-speed backward and forward motions with coordinated control of the steering angle and locomotion velocity are performed to produce a lateral displacement of the vehicle into the parking place. The number of such motions depends on both the geometric characteristics of the local map, and the size and kinematics characteristics of the vehicle. Parameterized sinusoidal controls have been used to perform the required backward and forward maneuvers (see (Paromtchik and Laugier, 1996) for more details).

## 2.4 Platooning

The platooning *SBM* allow the controlled vehicle to automatically follow an other vehicle. This *SBM* operates in two phases (Parent and Daviet, 1996): (1) determining the relative velocity and position/orientation parameters of the leading/following vehicles using on-board sensors, and (2) generating the required longitudinal and lateral controls on the following vehicle.

The assessment of the relative velocity and position/orientation parameters is obviously performed at a rate consistent with the servo-loop frequency (50 Hz in practice). In our implementation, these parameters are evaluated using a *linear camera* (equipped with appropriate optical lenses) located on the front side of the following vehicle (*FV*), and an *infrared target* located at the rear side of the leading vehicle (*LV*).

Then, following autonomously *LV* is performed by controlling, at the servo-loop frequency, the acceleration/deceleration and angular velocity of the steer-

ing wheel of *FV*. The *longitudinal control* is performed using the following control law (see (Parent and Daviet, 1996) for more details):

$$A_f = C_v \Delta V + C_p(\Delta X - h V_f - d_{min})$$

where  $A_f$  is the acceleration of *FV*,  $\Delta X$  and  $\Delta V$  are the relative position and velocity of the two vehicles,  $d_{min}$  is the minimum distance between *LV* and *FV*,  $h$  is a time constant (in practice,  $d_{min} = 1m$  and  $h = 0.35s$ ),  $C_v = 1/h$ , and  $C_p = \min(1/h, A_{max}/V_f)$ . The *lateral control* is performed using an approach based onto the classical “tractor model”.



Fig. 5. A platoon of three ligier vehicles.

## 2.5 Experiments

This approach has been implemented and tested on our experimental automatic vehicles (modified Ligier electric cars and Cycab). Each of these vehicles can either be manually driven, or it can be moved autonomously; it is equipped with a belt of ultrasonic range sensors (Polaroid 9000), a linear CCD-camera (the camera has 2048 pixels, it operates at a frequency of 1000 Hz, and it is equipped with a cylindrical lens and an infrared and polarized filter), and an infrared target made of three sets of LED organized along vertical lines. The Motion Controller has been implemented using the Orccad software (Simon *et al.*, 1993) running on a SUN Workstation; the related compiled code is transmitted via Ethernet to the vehicle control unit.

The figure 6 describes the execution of a lane following/changing maneuver on a circular road, along with the associated steering and velocity controls. The figure 7 shows a parallel parking maneuver executed in a street; the related steering and velocity controls are also shown. The figure 5 shows a platoon of autonomous vehicles, the leading vehicle being driven by a human driver. More details on these experiments can be found in (Laugier and Fraichard, 2001), (Paromtchik and Laugier, 1996), and (Parent and Daviet, 1996).

## 3. CONTROLLING A BI-STEERABLE CAR

From the motion planning point of view, a *bi-steerable car* (BS-car) is a very particular nonholonomic system due to its specific kinematics (see section 1.2). The nonholonomy of wheeled robots is due to the constraint of rolling without slipping of the wheels. As a

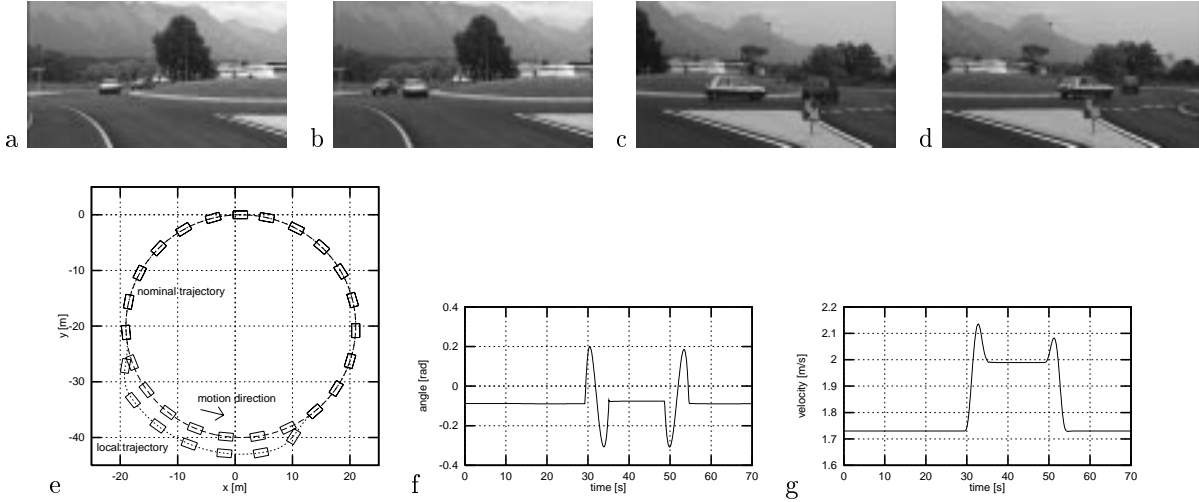


Fig. 6. Sequence of motions for lane following/changing on a circular road: (a) tracking the nominal trajectory, (b) lane changing and overtaking, (c) merging the nominal trajectory, (d) tracking the nominal trajectory, (e) executed trajectory, (f) steering angle history, (g) velocity controls.

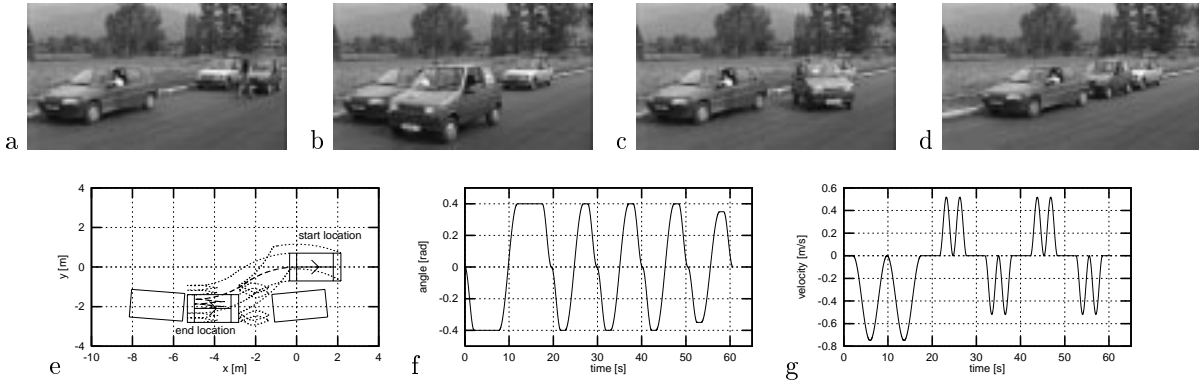


Fig. 7. Sequence of motions for parallel parking: (a) motion to localize a parking place; (b) selecting an appropriate start location; (c) backward motion into the place; (d) the parallel parking is completed, (e) executed trajectories, (f) steering angle history, (g) velocity controls.

consequence, this kind of robots have less controls than configuration variables. Therefore in general, even in the absence of obstacles, finding the sequence of admissible maneuvers steering the system from one point to another is a complex and open problem. Adding obstacles in the environment increases highly the difficulty of the problem with some additional geometric (or holonomic) constraints. That is why, treating holonomic constraints (using a *geometric planner*) separately from the nonholonomic ones (using a *steering method*) is nowadays almost a “classical” approach (e.g. see (Laumond, 1998)).

As far as the BS-car is concerned, one might consider that the geometrical planning problem for a single solid object (representing the robot body) has been solved. However, this is not the case for the steering problem. Indeed, the main research work on non-holonomic motion planning has been carried out on canonical platforms such as car-like robots and mobile robots with trailers (Reeds and Shepp, 1990), (Fliess *et al.*, 1995), (Tilbury *et al.*, 1995), (Sekhavat *et al.*, 1997), (Laumond, 1998). Results obtained on these platforms can not be directly applied to the BS-cars (including the Cycab robot).

We propose to solve the steering problem for BS-cars by showing that most of them are *differentially flat* and by computing the flat outputs. Indeed, knowledge of these linearizing outputs enables the design of an infinite number of steering methods. Furthermore, based on these flat outputs and inspired by previous work (Sekhavat *et al.*, 1997), (Lamiriaux *et al.*, 1999) we show how to obtain a steering method leading to a collision-free motion planner for a BS-car enabling maneuvers among obstacles.

### 3.1 Flatness of the Cycab

The kinematic model of the bi-steerable robot (See Fig. 8), for a reference frame located at the mid-point of the rear axle (point  $R$ ), can be expressed as follows (Sekhavat and Hermosillo, 2001):

$$\begin{cases} \dot{x}_R = v_R \cdot \cos(\theta + f(\phi)) \\ \dot{y}_R = v_R \cdot \sin(\theta + f(\phi)) \\ \dot{\theta} = v_R \cdot \frac{\sin(\phi - f(\phi))}{L \cdot \cos(\phi)} \end{cases} \quad (1)$$

Where the state parameters are defined on Fig. 8,  $v_R$  (resp.  $v_F$ ) is the linear velocity at the point  $R$  (resp.

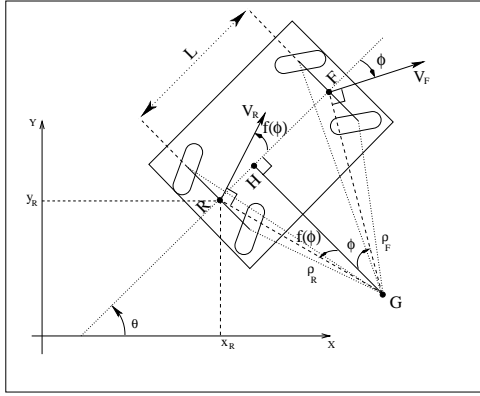


Fig. 8. Model of the bi-steerable car.

$F$ ) and  $f$  is the *characteristic function* of the robot between the rear and the front steering angle.

A system is said to be differentially flat if there exist *linearizing outputs*  $\mathbf{y} = (y_1, \dots, y_m)$  differentially independent such that:

- the linearizing outputs can be expressed as a function of the system variables and their successive derivatives,
- any system variable (state, controls, ...) can be expressed only from the linearizing outputs and their successive derivatives.

This implicates a bijective correspondence (locally) between arbitrary curves of  $\mathbf{y} = (y_1, \dots, y_m)$  and the smooth trajectories of the system.

The interesting point here is that, unlike the configuration coordinates of the nonholonomic system,  $(y_1, \dots, y_m)$  are differentially independent. Therefore any smooth curve in  $(y_1, \dots, y_m)$  space corresponds to an admissible path for the system; we cannot say the same for the configuration space. Hence, path planning becomes easier in the linearizing space since we do not have to take into account any kinematic constraint along the path. The only constraints that have to be considered are those on the successive time derivatives of the curve at its both ends  $(\mathbf{y}_i, \mathbf{y}_f)$ . These constraints are imposed by the starting and goal configurations and their successive derivatives  $(\mathbf{q}_i, \mathbf{q}_f, \dot{\mathbf{q}}_i, \dot{\mathbf{q}}_f, \dots)$ . Therefore, many families of curve such as parameterized polynomials can be employed to solve the planning problem in the flat space.

The compelling question is: can we apply these results to the BS-car? In order to give an answer, the first question to answer turns to be: is this system “flat”? The answer will obviously depends on the characteristic function  $f$ . Actually, it turns out that in general, the BS-car is flat. Indeed, in (Sekhavat and Hermosillo, 2001) we present a necessary and sufficient condition on  $f$  for the BS-car to be flat :

**Proposition 1** Given a Bi-steerable car with a characteristic function  $f$  between the rear and the front steering angle. The system is flat iff  $f$  is such that for

any front steering angle  $\phi$ :

$$\Lambda_1(\phi) = 0 \quad \text{or} \quad \Lambda_2(\phi) = 0$$

With

$$\begin{aligned} \Lambda_1(\phi) &= f'(\phi)^2 \cos^2 \phi + \cos^2 f(\phi) \\ &\quad - 2f'(\phi) \cos \phi \cos k\phi \cos(\phi - f(\phi)) \\ \Lambda_2(\phi) &= f''(\phi) + 2f'(\phi)^2 \tan(f(\phi)) - 2f'(\phi) \tan(\phi) \end{aligned}$$

If we consider for example the case of a linear characteristic function ( $f(\phi) = k\phi$ ), a corollary of the proposition above is that the BS-car is flat for any  $k$  except  $k = 1$ , which is the case of a robot with the front and rear wheels always parallel. Note that this robot is even not controllable (since the orientation can never be changed).

### 3.2 The Flat Outputs of the Cycab

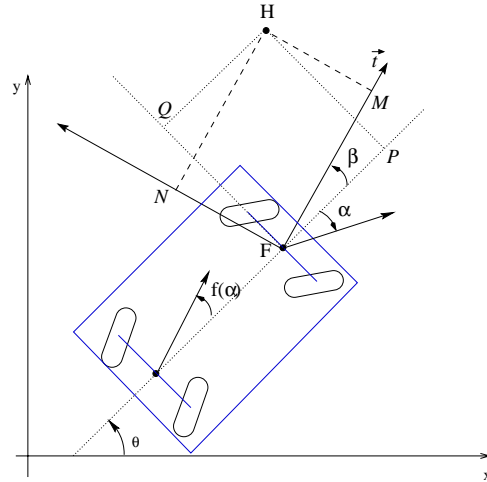


Fig. 9. Frames and coordinates for the flat outputs' computation.

For most of the differentially flat mobile robots (such as the car-like or the tractor-trailer systems), the flat outputs (more precisely *one* selection of the flat outputs) are the coordinates of a reference point, fixed with respect to the robot. Using tools from the exterior differential algebra, we show in (Sekhavat *et al.*, 2001) that this is not the case for the BS-car. Indeed for the BS-car, the position of the reference point with respect to the robot is a function of the steering angle. Let us define a turning frame  $(F, \vec{t}, \vec{t}^\perp)$  with respect to the robot (see Fig. 9) such that, the relative angle with respect to the robot is a function of the steering angle :

$$\beta(\alpha) = (\vec{u}_\theta, \vec{t}) = \tan^{-1} \frac{B(\alpha)}{A(\alpha)}$$

with

$$\begin{aligned} A(\alpha) &= \cos^2(\alpha) f'(\alpha) - \cos^2(f(\alpha)) \\ B(\alpha) &= \cos(\alpha) \sin(\alpha) f'(\alpha) - \cos(f(\alpha)) \sin(f(\alpha)) \end{aligned}$$

Now if we call  $H$  the reference point such that its Cartesian coordinates  $(x_H, y_H)$  are some flat outputs of the BS-car, we show in (Sekhavat *et al.*, 2001) that we can compute  $H$ 's coordinates  $(M, N)$  in the turning frame attached to the vector  $\vec{t}$  :

$$M(\alpha) = \frac{L \cos^2(f(\alpha))}{(A^2(\alpha) + B^2(\alpha))^{\frac{1}{2}}}$$

$$N(\alpha) = - \int_0^\alpha \frac{L \cos^2(f(u))(B'(u)A(u) - A'(u)B(u))}{(A^2(u) + B^2(u))^{\frac{3}{2}}} du$$

A change of coordinate will then give us the flat outputs  $(x_H, y_H)$  in function of the system coordinates  $(x, y, \theta, \alpha)$ . Some more computation lead to the inverse expressions giving the system coordinates in function of the flat outputs and their successive derivatives (see (Sekhavat *et al.*, 2001)).

### 3.3 A Motion Planner for the BS-car

The computation of the flat outputs is the major step towards the design of steering methods for BS-cars. Indeed, finding the flat outputs offers an infinite number of possible steering methods (see Section 3.1) for the BS-car. Therefore the problem of the open loop control for this system is solved. Moreover, a good choice of the steering method combined with a complete geometric planner allows to compute open loop controls that not only lead to the goal but also avoid obstacles (Sekhavat *et al.*, 2001). Fig. 10 shows the output of such a planner. Experiments have successfully been conducted using our Cycab vehicle moving in the parking area of INRIA Rhône-Alpes.

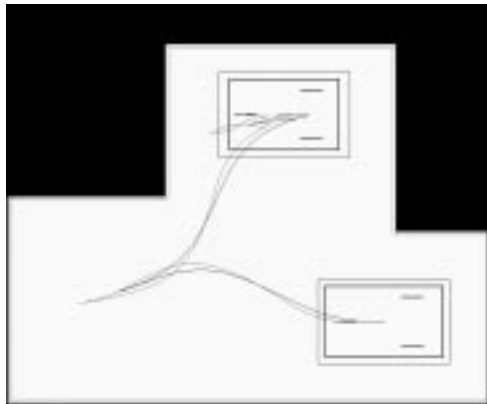


Fig. 10. Simulations results of the Cycab executing a parking.

## 4. OBSTACLE AVOIDANCE IN A DYNAMIC ENVIRONMENT

### 4.1 Velocity Obstacles

Most of the existing planners dealing with moving obstacles cannot be used on-line in our context of application. This is due in practice either to the number of moving obstacles, or to the high relative velocities, or to the kinematic and dynamic constraints of our robot ,or... However, the underlying reason in our opinion is the fact that until recently, the problem was largely addressed in what we consider to be a *zero order* approach because it relies explicitly on the positions of both robot and obstacles to determine potential collisions. In the control design process, the kinematic equations of the system and the obstacles are integrated

and a collision checking is done in a configuration space augmented by a time dimension. A *first order*, velocity based, approach was presented in (Fiorini and Shiller, 1998), introducing the concept of Velocity Obstacles (V-obstacles) assuming that both robot and obstacles are moving in straight lines and at constant speed. The control design is not done anymore by checking the collision at the configuration level, but rather at the velocity level (hence the *first order* definition). The velocity obstacles correspond to the set of velocities that will induce a collision sometimes in the future. In the case of linear movements, they can be represented in a 2D velocity space by a union of simple geometric shapes (such as triangles and trapezoids). Choosing a velocity vector outside the velocity obstacles induces that the robot will never collide any obstacles. Thus, we can choose in a single shot a safe straight trajectory avoiding all the obstacles. In this approach, the initial problem has been over-constrained, since a velocity in the V-obstacle may actually be admissible if we change it early enough before the collision. However, this rough presentation of the V-obstacle is the very kernel of the real-time compliance of this approach. Indeed, the V-obstacles as described above are easy to compute and at each iteration, any successful choice of the velocity is not only valid for that iteration, but above that, gives directly a solution that is safe for ever (or until the assumptions on the obstacles motion change).

However, in most of the applications, the obstacles are not moving in straight lines. Approaching the obstacle real motion by a straight line could be valid only for a short laps of time. An approximation of the obstacles motion by piecewise linear motions will induce an iterative computation of the V-obstacles followed by a (heuristic) decision making. Designing such an algorithm that would be complete and safe is not an easy task. Moreover, if the time period of the algorithm becomes too small, the real-time constraint may not be satisfied anymore. A way of reducing the problem is to directly get the representation of the V-obstacles corresponding to the real trajectories of the obstacles instead of iteratively approximating it by linear V-obstacles. In this way, we need typically only one V-obstacle building for any moving obstacle with known trajectory. In the case of obstacles with unknown motion, the V-obstacle would still be built for the best estimation of their motion, which do not have to be linear anymore. Building the Non-linear Velocity Obstacles (NLVO) is what we propose to do (Shiller *et al.*, 2001).

### 4.2 Non-Linear Velocity Obstacles

For simplicity, we consider circular robots and obstacles. Growing the obstacles by the radius of the robot transforms the problem to a point vehicle avoiding circular obstacles. It is also assumed that the “instantaneous” states (position, velocity, and acceleration) of obstacles moving along arbitrary trajectories are either known or measurable.

A few words about notation: henceforth,  $A$  denotes a point robot, and  $\mathcal{B}$  denotes the set of points defining the geometry of an obstacle  $B$ . Since the obstacle is solid,  $\mathcal{B}$  does not depend on  $t$ .  $B(t)$  denotes the set of points occupied by the obstacle  $B$  at time  $t$ .

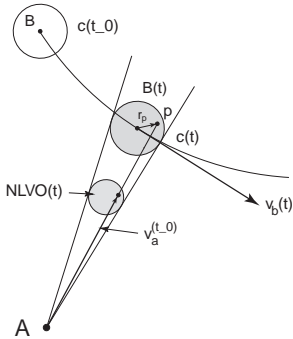


Fig. 11. Construction of the Non-Linear V-obstacle.

The Non-Linear V-obstacle ( $NLVO$ ) applies to the scenario shown in Figure 11, where, at time  $t_0$ , a point vehicle,  $A$ , attempts to avoid a circular obstacle,  $B$ , located at  $c(t_0)$ , and is following a general known trajectory,  $c(t)$ .

The  $NLVO$  is constructed by first determining the absolute velocities of  $A$ ,  $v_a$ , that would result in collision at a specific time  $t$ . Referring to Figure 11,  $v_a^{t_0}(t, p)$  that would result in collision with point  $p \in B(t)$  at time  $t > t_0$ , expressed in a frame centered at  $A(t_0)$ , is simply

$$v_a^{t_0}(t, p) = \frac{c(t) + r_p}{t - t_0},$$

where  $r_p$  is the vector to point  $p$  in the obstacle's fixed frame.

Similarly, the set,  $NLVO(t)$  of all absolute velocities of  $A$  that would result in collision with any point in  $B(t)$  at time  $t > t_0$  is:

$$NLVO(t) = \frac{c(t) + \mathcal{B}}{t - t_0}$$

Geometrically,  $NLVO(t)$  is a scaled  $\mathcal{B}$ , bounded by the cone formed between  $A$  and  $B(t)$ . This leads to the construction of the Non-Linear V-obstacle, as stated in the following theorem:

**Theorem 1:** Let  $A$  be a point vehicle, located at time  $t = t_0$  at the origin, and  $B$  be an obstacle that is moving along a general trajectory  $c(t)$ ,  $t = [t_0, \infty)$ . The Non-Linear V-obstacle,  $NLVO$ , representing the set of all linear velocities of  $A$  that would collide with  $B(t)$  at time  $t = (t_0, \infty)$  is defined by

$$NLVO = \bigcup_{t > t_0} \frac{c(t) + \mathcal{B}}{t - t_0}, \quad (2)$$

Equation 2 formally define the  $NLVO$ . However, even though it gives a constructive way of obtaining the  $NLVO$ , it does not give a compact (analytical) representation of the  $NLVO$ . Such an expression is of

great importance when we consider the computation time constraint. Moreover, this analytical expression is capital in designing a strategy for choosing a sequence of safe velocities in time. Some more computations using an intermediate virtual obstacle followed by an approximation lead us to the analytical expression of the  $NLVO$  borders (see (Shiller *et al.*, 2001) for details). The approximation is not necessary, but it greatly simplifies the expressions and it is conservative in the sense that the approximated  $NLVO$  contains the exact one. Notice also that the analytical expressions are valid for the case of circular obstacles. However, since the method works for any number of obstacles, we can consider any vehicle and approximate it as closely as desired by a collection of circles.

As an example, below are the formula of the  $NLVO$  centerline  $c_v(t)$  and borders  $vo^+(t)$ ,  $vo^-(t)$ , for a circular obstacle of radius  $r$  that moves along a circular trajectory of radius  $R$ , centered at point  $O = De^{i\phi}$  in a coordinate frame centered at  $A$ , at angular speed  $\omega$ , starting at  $\theta(t_0) = \theta_0$ :

$$\begin{aligned} c_v(t) &= \frac{1}{t} [De^{i\phi} + Re^{i\theta(t)}] \\ vo^\pm(t) &= c_v(t) \pm i \frac{r}{t} c_l(t) / \|c_l(t)\| \\ \text{with } c(t) &= De^{i\phi} + Re^{i\theta(t)} \\ c_l(t) &= \frac{1}{t} [De^{i\phi} + (1 - i\omega t)Re^{i\theta(t)}] \end{aligned}$$

Figure 12 shows the use of the  $NLVO$  to solve a difficult traffic merging problem. Vehicle  $A$ , coming from the left, wishes to merge tangentially with the traffic in the right lane of a curved road after crossing the left lane with opposing traffic, using a constant velocity. The vehicles on the curved lanes move at constant speeds. In Figure 12, (a) represents the initial configuration with the trajectories and the velocities of the robots; (b) represents the same situation as (a), but with the  $NLVO$  drawn. The complexity of this situation is apparent from the many discontinuous sets of avoiding velocities. The choice of a velocity in the free space in (b) permits to perform the entire maneuver safely at a constant speed, as shown in the remaining snapshots.

## 5. CONCLUSION

In this paper, we have addressed some of the major problems to be solved for automatizing cars, and we have presented the solutions that we have designed, implemented, and tested onto our experimental vehicles. Three important technical issues have been addressed: autonomous maneuvering, controlling a bi-steerable car and more particularly our *Cycab* vehicle, and obstacle avoidance in a dynamic environment. Original solutions have been proposed for contributing to solve these problems; these solutions relies onto the concepts of Sensor-Based Maneuver, of Differential Flatness, and of Non-Linear V-Obstacle. Several experimentations involving our experimental vehicles (several electric *Ligier* vehicles and *Cycabs*), have successfully been



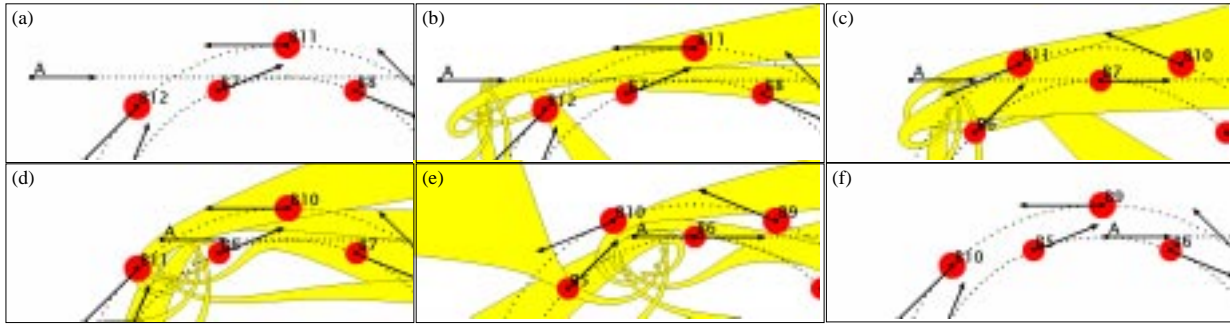


Fig. 12. Merging with traffic along a curved road at a constant speed.

conducted in slightly protected areas. However, all these techniques have to be robustified in order to be really applicable in normal traffic conditions. Our current work deals with this problem, in particular for obtaining robust sensory information, for processing hazards and exit/merging operations in a platoon, and for performing fast obstacle avoidance maneuvers among moving obstacles sensed using on-board sensors; these issues are respectively studied in the scope of the *Carsense* and *Cybercars* European project, and of the *Arcos2003* French project.

**Acknowledgements** This work was partially supported by the Inria-Inrets *Praxitéle* programme on urban public transport [1993-1997], by the Inco-Copernicus ERB-IC15-CT96-0702 project “Multi-agent robot systems for industrial applications in the transport domain” [1997-1999], and by the “Automated Road” (*LaRA*) Inria programme [1999-2003]. We would like to thank Igor Paromtchik, Philippe Garnier, and Pascal Daviet for their technical contributions to the work presented in this paper.

## 6. REFERENCES

- Fiorini, P. and Z. Shiller (1998). Motion planning in dynamic environments using velocity obstacles. *Int. Journal of Robotics Research* **17**(7), 760–772.
- Fliess, M., J. Lévine, P. Martin and P. Rouchon (1995). Flatness and defect of nonlinear systems : Introductory theory and examples. *Int. Journal of Control* **61**(6), 1327–1361.
- Kanayama, Y., Y. Kimura, F. Miyazaki and T. Noguchi (1991). A stable tracking control method for a non-holonomic mobile robot. In: *IEEE Int. Conf. on Intelligent Robots and Syst.*. Osaka (JP).
- Lamiroux, F., S. Sekhavat and J.-P. Laumond (1999). Motion planning and control for Hilare pulling a trailer. *IEEE Trans. Robotics and Automation* **15**(4), 640–652.
- Laugier, C. and Th. Fraichard (2001). Decisional architectures for motion autonomy. In: *Intelligent vehicle technologies* (L. Vlacic, F. Harashima and M. Parent, Eds.). Butterworth-Heinemann.
- Laumond, J.P., Ed.) (1998). *Robot Motion Planning and Control*. Springer-Verlag. Lecture notes in control and Information Sciences.
- Nelson, W. L. (1989). Continuous curvature paths for autonomous vehicles. In: *Proc. of the IEEE Int. Conf. on Robotics and Automation*. Vol. 3. Scottsdale, AZ (US). pp. 1260–1264.
- Parent, M. and P. Daviet (1996). Automated urban vehicles : towards a dual mode prt (presonnal rapid transit). In: *IEEE Int. Conf. on Robotics and Automation*. Minneapolis.
- Paromtchik, I. E. and C. Laugier (1996). Motion generation and control for parking an autonomous vehicle. In: *Proc. of the IEEE Int. Conf. on Robotics and Automation*. Minneapolis, MN (US). pp. 3117–3122.
- Reeds, J. A. and L. A. Shepp (1990). Optimal paths for a car that goes both forwards and backwards. *Pacific Journal of Mathematics* **145**(2), 367–393.
- Sekhvat, S. and J. Hermosillo (2001). Cycab bi-steerable cars – a new family of differentially flat systems. Technical report. Inst. Nat. de Recherche en Informatique et en Automatique.
- Sekhvat, S., F. Lamiroux, J.-P. Laumond, G. Bauzil and A. Ferrand (1997). Motion planning and control for hilare pulling a trailer: experimental issues. In: *Proc. of the IEEE Int. Conf. on Robotics and Automation*. Albuquerque, NM (US). pp. 3306–3311.
- Sekhvat, S., J. Hermosillo and P. Rouchon (2001). Motion planning for a bi-steerable car. In: *Proc. of the IEEE Int. Conf. on Robotics and Automation*. Seoul (KR).
- Shiller, Z., F. Large and S. Sekhavat (2001). Motion planning in dynamic environments: Obstacles moving along arbitrary trajectories. In: *Proc. of the IEEE Int. Conf. on Robotics and Automation*. Seoul (KR).
- Simon, D., B. Espiau, E. Castillo and K. Kapellos (1993). Computer-aided design of a generic robot controller handling reactivity and real-time control issues. In: *IEEE Transactions on Control Systems Technology*. pp. 213–229.
- Tilbury, D., R. Murray and S. Sastry (1995). Trajectory generation for the  $n$ -trailer problem using goursat normal form. *trans-ac* **40**(5), 802–819.