Towards Autonomous Vehicles for Future Intelligent Transportation Systems

Christian LAUGIER
INRIA Rhône-Alpes & GRAVIR
Zirst. 655 av. de l'Europe. 38330 Montbonnot Saint Martin. France
E-mail: Christian.Laugier@inrialpes.fr

Abstract

Developing new Intelligent Transportation Systems which take into consideration the socio-economical, environmental, and safety factors of the modern society, is one of the grand challenges of the next century. Recent progress in the fields of Mobile Robots, Control Architectures, and Computer Vision allows us to now envisage the integration of new autonomous and driving-assistance capabilities within future vehicles. This paper presents the concept of "Automated Urban Vehicle" which is currently developed within the scope of the French "Praxitèle" and "Automated Road" programmes. It focuses onto the novel Control and Decisional Architecture which has been developed for providing each vehicule with the required autonomous capabilities. Experimental results obtained with our automatic electric vehicles are presented for three types of manoeuvres: lane following/changing, parallel parking, and platooning.

1 Introduction

Developing new Intelligent Transportation Systems which take into consideration the socio-economical, environmental, and safety factors of the modern society, is one of the grand challenges of the next century. Recent progress in the fields of Mobile Robots, Control Architectures, and Computer Vision allows us to now envisage the integration of new autonomous and driving-assistance capabilities within future vehicles. Several national and international projects have been launched for that purpose during the last 10 years (e.g. the European Eurêka Project *Prometheus*, the French national programme *Praxitèle*, the *Path* project in USA ...).

The goal of this paper is to present the concept of *Automated Urban Vehicle* which is currently developed within the scope of the French *Praxitèle* and *Automated Road* programmes (these programmes are aimed at the development of a new urban transportation system based on a fleet of electric and computer-driven vehicles [8]). A special attention will be given to the Control and Decisional Architecture which has been developed for providing each vehicle with the required autonomous capabilities.

Motion autonomy for various types of vehicles has already been widely studied in the 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 *off-line global path/trajectory planner* (usually using the Dubins' curves [1] or the Reed & Shepp curves [11]), 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 behaviours resulting from the combination of two contradictory functions: trajectory tracking and obstacle avoidance [3]. In order to generate *smooth and safe motions* for an autonomous car-like 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, we have designed and implemented a novel control architecture. This architecture includes an off-line global trajectory planner[1], a decisional kernel that selects appropriate *sensor-based manoeuvres* in real-time, and a reactive motion controller that makes use of a set of *control experts*[2] to execute the required sensor-based manoeuvres. The same basic idea has also been used at Laas for providing the AMR robot with some au-

[1] A trajectory represents both a geometric path (*i.e.* a smooth curve) and its associated velocity profile.

[2] A *control expert* is a parameterized control program adapted to the execution of a particular type of sensor-based motion.

tonomous behaviours [6].

In the sequel, we present an overview of this control architecture along with a description of the three main types of sensor-based manoeuvres which have been implemented and experimentally validated using our autonomous vehicles: lane following/changing, parallel parking, and platooning.

2 Control and Decisional Architecture

The Control and Decisional Architecture has been designed to generate *smooth and safe motions* that satisfy both the task constraints and the kinematic and dynamic constraints of the car-like vehicle. This architecture includes an *off-line global trajectory planner*[3], a decisional Kernel which generates on-line appropriate *sensor-based manoeuvres*, and a *motion controller* which makes use of a set of *control experts* to execute the required sensor-based motions.

The key idea of our approach is to plan and carry out *sensor-based manoeuvres* where the following scheme is applied: first, a *parameterized motion plan* is produced by combining a *nominal trajectory*[4] with a set of *generic sensor-based manoeuvres* selected on-line from the library according to the current motion context; then, the involved motion controls are using appropriate *control experts* having the ability to react in real-time to unforeseen events. In the case of a failure due to an unforeseen event, the decisional kernel of the system decides either to replan the motion or to insert in real-time a more appropriate sensor-based manoeuvre (which in turn is expanded into a sequence of executable local trajectories and sensing operations)[5].

Obviously, the reactivity of the system to unforeseen events along with the quality of the executed motions mainly relies on the paradigm of *sensor-based manoeuvre* (*SBM*). 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 manoeuvre, while adapting the control parameters to the current execution context. In the sequel, we will show how this approach has been used for implementing several types of sensor-based manoeuvres.

[3]The reader is referred to [2] and [12] for a complete presentation of the global trajectory planner.

[4]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 behaviours of the moving obstacles.

[5]The reader is referred to [5] for a more complete presentation of the Control and Decisional Architecture.

3 Lane Following/Changing

The purpose of this *SBM* is to allow 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 done, in order to satisfy a set of different constraints: collision avoidance, time constraints, kinematic and dynamic constraints. In our previous approach, we have used a fuzzy behaviour merging process for combining a trajectory tracking behaviour with a collision avoidance behaviour. However, this approach has exhibited non consistent behaviours: it generates oscillations, and it does not guaranty that all the previous constraints are always satisfied [3]. The current *SBM* approach makes use of *smooth local trajectories* for avoiding the detected obstacles. These local trajectories allow the vehicle to move away from the obstructed nominal trajectory, and to catch up this nominal trajectory when the (stationary or moving) obstacle has been overtaken. All these local trajectories verify the motion constraints and enable the vehicle to follow a smooth path. This type of manoeuvre is executed using two Control Experts: "trajectory tracking" and "lane changing".

Trajectory Tracking. Any trajectory tracking method working for non-holonomic vehicles can be used for implementing this type of control expert. We apply an approach [4] which guarantees the stable tracking of a feasible trajectory when the vehicle's control commands are of the following form:

$$\dot{\theta} = \dot{\theta}_{ref} + v_{R,ref}(k_y y_e + k_\theta \sin\theta_e), \qquad (1)$$

$$v_R = v_{R,ref}\cos\theta_e + k_x x_e, \qquad (2)$$

where $q_e = (x_e, y_e, \theta_e)^T$ represents the error between the reference configuration q_{ref} and the current configuration q of the vehicle ($q_e = q_{ref} - q$), $\dot{\theta}_{ref}$ and $v_{R,ref}$ are the reference velocities, $v_R = v\cos\phi$ is the control command for the locomotion velocity of the midpoint of the rear wheel axle, k_x, k_y, k_θ are positive constants, and $\phi = \arctan\left(\frac{\dot{\theta} L}{v_{R,ref}}\right)$.

Lane Changing. This type of control expert is applied for executing a smooth lane change manoeuvre. This manoeuvre is carried out by generating and tracking an appropriate smooth local trajectory. Let $\mathcal{T}$ be the nominal trajectory to track, d_T be the distance between $\mathcal{T}$ and the middle line of the free lane to reach, s_T be the curvilinear distance along $\mathcal{T}$ between the

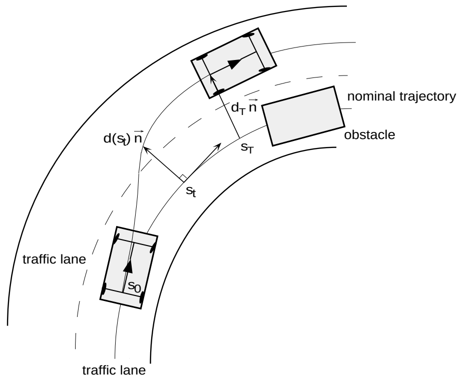

Figure 1: generation of smooth local trajectories to avoid an obstacle.

vehicle and the obstacle (or the selected end point for the lane change), and $s = s_t$ be the curvilinear abscissa along $\mathcal{T}$ since the starting point of the lane change (see Fig. 1). A feasible smooth trajectory for executing a lane change can be obtained using the following quintic polynomial [7]:

$$d(s) = d_T \left(10\left(\frac{s}{s_T}\right)^3 - 15\left(\frac{s}{s_T}\right)^4 + 6\left(\frac{s}{s_T}\right)^5\right), \quad (3)$$

In this approach, the distance d_T is supposed to be known beforehand, or computed according to the size of the sensed obstacle, and the minimal value required for s_T can be estimated using the values of the maximum allowed curvature $\mathcal{C}_{max}$ and of the maximum allowed lateral acceleration γ_{max} (see [9] for more details).

At each time t from the starting time T_0, the reference position p_{ref} is translated along the vector $d(s_t).\vec{n}$, where $\vec{n}$ represents the unit normal vector to the nominal velocity vector along $\mathcal{T}$; the reference orientation θ_{ref} is converted into $\theta_{ref}+\arctan(\partial d(s_t)/\partial s)$, and the reference velocity $v_{R,ref}$ is obtained using the following equation:

$$v_{R,ref}(t) = \frac{dist(p_{ref}(t), p_{ref}(t+\Delta t))}{\Delta t}, \quad (4)$$

where *dist* stands for the Euclidean distance.

This type of Control Expert can also 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), a value $s_{T,min}$ is calculated and compared with the distance between the vehicle and the obstacle. The result of this computation is used to decide which behaviour to apply: avoid the

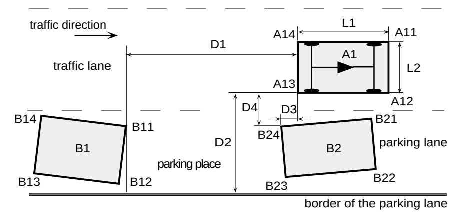

Figure 2: start location for parallel parking.

obstacle, slow down or stop. In this approach, an obstacle avoidance or overtaking manoeuvre consists of the smoothly motion towards a collision-free "parallel" trajectory, and further catching up $\mathcal{T}$ as soon as it becomes possible (see Fig. 1).

4 Parallel Parking

The purpose of the parallel parking *SBM* is to automatically park the vehicle within an unknown parking area. This *SBM* involves three main phases: (1) localizing a sufficient space (parking place)in the parking area, (2) obtaining an appropriate starting location for the vehicle relatively to the parking place, and (3) performing the parallel parking manoeuvre.

Finding a parking place. During this phase, 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 map is used for selecting an appropriate free space for parking the vehicle (see below) ; the obstacles are avoided using an other *SBM* (e.g. the lane following/changing *SBM*).

Selecting a starting location. Drivers know from experience that before the parking manoeuvre starts, the vehicle must be oriented near parallel to the parking place and it must also reach a convenient start position in front of the place. A start location for parallel parking is shown in Fig. 2 where an automatic vehicle A1 is in a traffic lane. The parking lane with parked vehicles B1, B2 and a parking place between them is on the right-hand side of the vehicle A1. L1 and L2 are respectively the length and width of A1, and D1 and D2 are the distances available for longitudinal and lateral displacements of A1 within the place. D3 and D4 are the longitudinal and lateral displacements of the corner A13 of A1 relative to the corner B24 of B2.

The distances D1, D2, D3 and D4 are computed by the control unit from data obtained by the sensor and servo units. The control unit compares the length (D1-D3) and width (D2-D4) of the parking place with the length L1 and width L2 of A1, where L1 and L2 include sufficient clearance for the vehicle to move around. If (D1-D3) > L1 and (D2-D4) > L2, the parking place is sufficient for parallel parking.

Performing the parking manoeuvres. 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 the distances D1, D2, D3, D4 and the necessary parking "depth" which depends on the width L2 of the vehicle A1. The start and end orientations of the vehicle are the same for each iterative motion $i = 1, \ldots, N$.

For the i-th iterative motion (but omitting the index "i"), let the start coordinates of the vehicle be $x_0 = x(0)$, $y_0 = y(0)$, $\theta_0 = \theta(0)$ and the end coordinates be $x_T = x(T)$, $y_T = y(T)$, $\theta_T = \theta(T)$, where T is duration of the motion. The "parallel parking" condition means that

$$\theta_0 - \delta_\theta \ < \ \theta_T \ < \ \theta_0 + \delta_\theta , \qquad (5)$$

where $\delta_\theta > 0$ is a small admissible error in orientation of the vehicle.

The following control commands of the steering angle ϕ and locomotion velocity v provide the parallel parking manoeuvre [10]:

$$\phi(t) = \phi_{max}\, k_\phi\, A(t), \quad 0 \le t \le T, \qquad (6)$$

$$v(t) = v_{max}\, k_v\, B(t), \quad 0 \le t \le T, \qquad (7)$$

where $\phi_{max} > 0$ and $v_{max} > 0$ are the admissible magnitudes of the steering angle and locomotion velocity respectively, $k_\phi = \pm 1$ corresponds to a right side (+1) or left side (–1) parking place relative to the traffic lane, $k_v = \pm 1$ corresponds to forward (+1) or backward (–1) motion,

$$A(t) = \begin{cases} 1, & 0 \le t < t', \\ \cos \frac{\pi(t-t')}{T^*}, & t' \le t \le T - t', \\ -1, & T - t' < t \le T, \end{cases} \qquad (8)$$

$$B(t) = 0.5 \left(1 - \cos \frac{4\pi t}{T}\right), \quad 0 \le t \le T, \qquad (9)$$

where $t' = \frac{T-T^*}{2}$, $T^* < T$.

The commands (6) and (7) are open-loop in the (x, y, θ)-coordinates. The steering wheel servo-system and locomotion servo-system must execute the commands (6) and (7), in order to provide the desired (x, y)-path and orientation θ of the vehicle. The resulting accuracy of the motion in the (x, y, θ)-coordinates depends on the accuracy of these servo-systems. Possible errors are compensated by subsequent iterative motions.

For each pair of successive motions $(i, i+1)$, the coefficient k_v in (7) has to satisfy the equation $k_{v,i+1} = -k_{v,i}$ that alternates between forward and backward directions. Between successive motions, when the velocity is null, the steering wheels turn to the opposite side in order to obtain a suitable steering angle ϕ_{max} or $-\phi_{max}$ to start the next iterative motion.

In this way, the form of the commands (6) and (7) is defined by (8) and (9) respectively. In order to evaluate (6)-(9) for the parallel parking manoeuvre, the durations T^* and T, the magnitudes ϕ_{max} and v_{max} must be known. The way these values are computed by the system is described in [10]. This computation is carried out using the kinematic model of the vehicle and the commands (6) and (7), and the solutions are chosen in order to minimize the number of backward/forward manoeuvres.

5 Platooning

The purpose of this *SBM* is to allow the vehicle to follow automatically an other vehicle (this other vehicle can either been moved autonomously, or been driven by a human driver). Such a *SBM* takes as input the current (velocity, position, orientation) parameters of the vehicle to control[6], and it generates in real-time the required lateral and longitudinal controls. This *SBM* operates in two phases [8]: (1) determining the relative velocity and position/orientation parameters, and (2) generating the required longitudinal and lateral controls.

Determining the state parameters. The assessment of the velocity and of the position/orientation parameters of the leading vehicle has to be performed at a rate consistent with the servo-loop frequency (50 Hz in practice). This information is computed from the sensory data provided by an appropriate sensor, i.e. a sensor having the capability to measure at high rate and with a high resolution the relative velocity, position, and orientation of the two vehicles. In our implementation, this is performed using a linear camera (equipped with appropriate optical lenses) located in the automatic vehicle, and an infrared target located at the rear side of the leading vehicle. This approach

[6] The (velocity, position, orientation) parameters of the following vehicle are computed in real-time from the sensory data ; they are expressed relatively to the leading vehicle reference frame.

allows us to obtain at the servo-loop frequency, the position/orientation parameters, i.e. the longitudinal and lateral distances DX and DY between the two vehicles, and the angle $D\psi$ between the main axes of the two vehicles; the velocity parameter is obtained by derivating the position parameters (see [8]).

Generating the required controls. Following the leading vehicle is performed by controlling, at the servo-loop frequency, the acceleration/deceleration of the automated vehicle along with the angular velocity of its steering wheel.

As for the *longitudinal control*, the basic idea is to set a linear relation between the distance and the speed of the two vehicles :

$$X_l - X_f = d_{min} + h\ V_f \qquad (10)$$

where X_l, X_f, and V_f are respectively the position of the leading vehicle, the position of the following vehicle, and the velocity of the following vehicle ; d_{min} is the minimun distance between the two vehicles, and h is a time constant ($d_{min} = 1m$ and $h = 0.35s$ in the reported experiments). This approach has led us to make use of the following controller (see [8] for more details):

$$A_f = C_v\ \Delta V + C_p(\Delta X - h\ V_f - d_{min}) \qquad (11)$$

where A_f is the acceleration of the following vehicle, $\Delta V = V_l - V_f$, and $\Delta X = X_l - X_f$; the control gains C_p and C_v have been chosen as follows : $C_v = 1/h$, and $C_p = min(1/h,\ A_{max}/V_f)$. The fact that the position gain factor is variable, allows the controller to take into account the acceleration saturation and to deal with large initial errors (since C_p decreases when the speed increases).

As for the *lateral control*, we have applied a simple approach based onto the classical "tractor model". This approach leads the controller to always set the orientation of the steering wheel in a direction parallel to the orientation of the leading vehicle. This approach generate stable behaviours, but it leads the following vehicle to weakly cut the turns (this might be a problem for controlling a platoon of several vehicles in a constrained area).

6 Experiments

The approach described in the paper has been implemented and tested on our experimental automatic vehicles (modified Ligier electric cars). Each of these vehicles is equipped with the following capabilities:

Figure 3: a platoon of three ligier vehicles.

a b

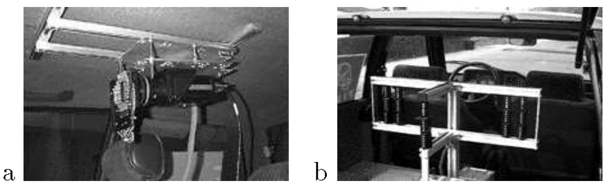

Figure 4: experimental setup for platooning: (a) the linear camera, (b) the first experimental infrared target.

(1) - a *sensor unit* to measure relative distances between the vehicle and environmental objects or vehicles, (2) - a *servo unit* to control the steering angle and the locomotion velocity, and (3) - a *control unit* that processes data from the sensor and servo units in order to "drive" the vehicle by issuing appropriate servo commands. Each vehicle can either be manually driven, or it can move autonomously using the *control unit* based on a Motorola VME162-CPU board and a transputer net. The *sensor unit* makes use of a belt of ultrasonic range sensors (Polaroid 9000) and of 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) ; this unit also makes use of an infrared target made of three sets of LED organized along vertical lines, figure 4 illustrates . The steering wheel servo-system is equipped with a direct current motor and an optical encoder to measure the steering angle. The locomotion servo-system of the vehicle is equipped with a 12 kW asynchronous motor and two optical encoders located onto the rear wheel axles (for odometry data). The vehicle also has an hydraulic braking servo-system. The Motion Controller has been implemented using the Orccad software [13] running on a SUN Workstation ; the related compiled code is transmitted via Ethernet to the VME162-CPU board.

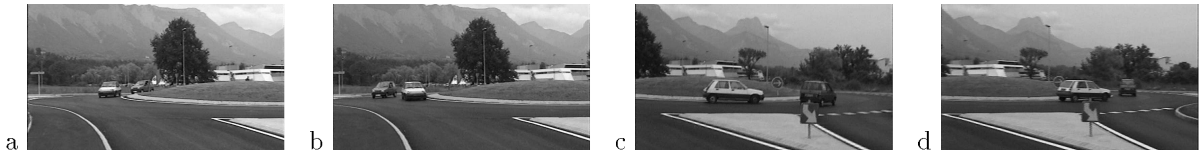

Figure 5: sequence of motions for lane following/changing on a circular road: (a) following the nominal trajectory,(b) lane changing (to the right) and overtaking, (c) lane changing to the left, (d) continuing with the nominal trajectory.

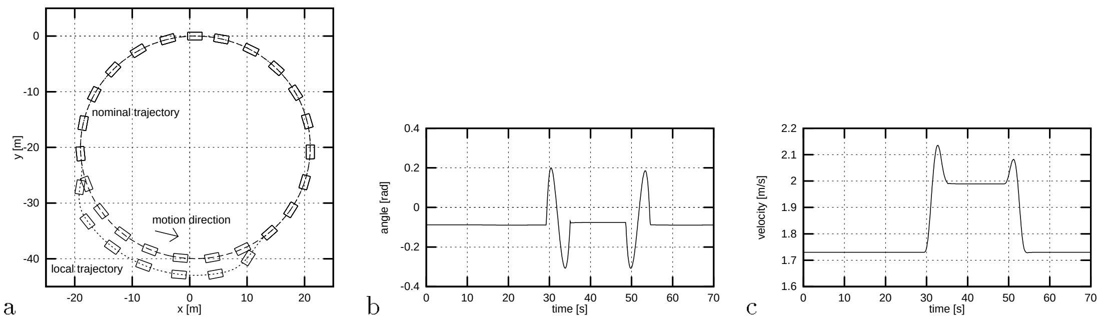

Figure 6: lane following/changing on a circular road: (a) related motion, (b) steering angle and (c) locomotion velocity controls applied.

An example of our experimental setup for *lane following/changing* on a circular road is shown in Fig. 5. In this experiment, the Ligier vehicle follows a nominal trajectory along the curved traffic lane, and it finds on its way another vehicle moving at a lower velocity (see Fig. 5a). When the moving obstacle is detected, a local trajectory for a right lane change is generated by the system, and Ligier performs the lane changing manoeuvre, as illustrated in Fig.5b. Afterwards, Ligier moves along a trajectory parallel to its nominal trajectory, and a left lane change is performed as soon as the obstacle has been overtaken (see Fig. 5c). Finally, Ligier continues to follow its nominal trajectory, as illustrated in Fig. 5d. The related motion of the vehicle is depicted in Fig. 6a. The steering and velocity controls applied during this manoeuvre are shown in Fig. 6b and Fig. 6c.

An example of our experimental setup for *parallel parking* in a street is shown in Fig. 7. This manoeuvre can be carried out in an environment including moving obstacles, *e.g.* a pedestrian or some other vehicles. In this experiment, Ligier was manually driven to a position near the parking place, the driver started the automatic parking and left the vehicle. Then Ligier moves forward autonomously in order to localize the parking place, obtains a convenient start location, and performs a parallel parking manoeuvre. When during this motion a pedestrian crosses the street in a dangerous proximity to the vehicle, as shown in Fig. 7a, this moving obstacle is detected, Ligier slows down and stops to avoid the collision. When the way is free, Ligier continues its forward motion. Range data is used to detect the parking place. A decision to carry out the parking manoeuvre is made and a convenient start position for the initial backward movement is obtained, as shown in Fig. 7b. Then, Ligier moves backwards into the parking place, as shown in Fig. 7c. During this backward motion, the front human-driven vehicle starts to move backwards, reducing the length of the parking place. The change in the environment is detected and taken into account. The range data shows that the necessary "depth" in the parking place has not been reached, so further iterative motions are carried out until it has been reached. The motion of the vehicle for a parallel parking manoeuvre is depicted in Fig. 8a. The control commands (6) and (7) generated are shown in Fig. 8b and Fig. 8c respectively. As shown in Fig. 8, the durations T of the iterative motions, magnitudes of the steering angle ϕ_{max} and locomotion velocity v_{max} correspond to the available displacements D1 and D2 within the parking place (*e.g.* the values of T, ϕ_{max} and v_{max} differ for the first and last iterative motion).

An example of our experimental setup for *platooning* in a street is shown in Fig. 3. The linear camera and the infrared target is shown in Fig. 4. During the

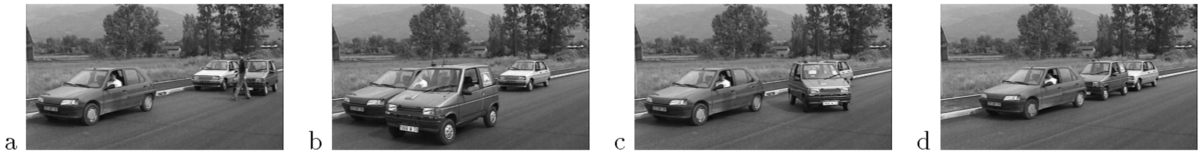

Figure 7: sequence of motions for parallel parking: (a) motion to localize a parking place; (b) selecting an appropriate start location; (c) backward motion into the place; (d) the parallel parking is completed.

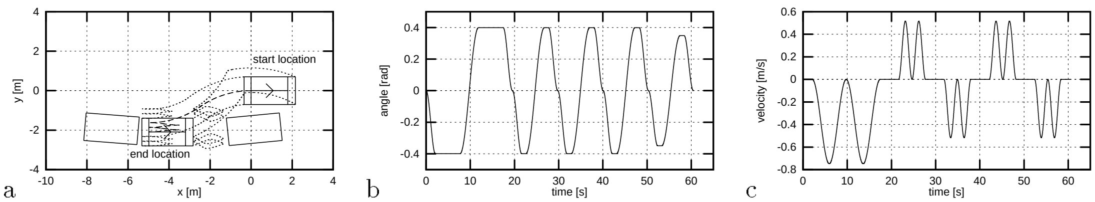

Figure 8: parallel parking manoeuvre: (a) related motion, (b) steering angle, (c) locomotion velocity controls applied.

execution of a platooning manoeuvres, the linear camera operates at a frequency of 1000 Hz for providing the relative position/orientation parameters of the two vehicles ; the accuracy of the measurement has been estimated at a value of 1mm for a distance of 10m. It has been experimentally shown that the system is robust according to various lighting and light reflecting conditions (thanks to the camera characteristics, to the pulsing infrared target, and to the used filters). Experiments have been conducted at speeds up to 60 km/h, with decelerations up to $2m/s^2$. The distance between the vehicles is proportional to the speed (see previous section), with a gap of 0.3s.

Acknowledgements. This work was partially supported by the Inria-Inrets Praxitèle programme on urban public transport *[1993-1997]*, and the Inco-Copernicus ERB-IC15-CT96-0702 project “Multi-agent robot systems for industrial applications in the transport domain” *[1997-1999]*.

References

[1] 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*, 79:497–516, 1957.

[2] Th. Fraichard. Dynamic trajectory planning with dynamic constraints: a ‘state-time space’ approach. In *Proc. of the IEEE-RSJ Int. Conf. on Intelligent Robots and Systems*, pages 1394–1400, Yokohama (JP), July 1993.

[3] Ph. Garnier and Th. Fraichard. A fuzzy motion controller for a car-like vehicle. In *Proc. of the IEEE-RSJ Int. Conf. on Intelligent Robots and Systems*, volume 3, pages 1171–1178, Osaka (JP), November 1996.

[4] Y. Kanayama, Y. Kimura, F. Miyazaki, and T. Noguchi. A stable tracking control method for a non-holonomic mobile robot. In *Proc. of the IEEE-RSJ Int. Conf. on Intelligent Robots and Systems*, Osaka (JP), November 1991.

[5] C. Laugier, Th. Fraichard, I. E. Paromtchik, and Ph. Garnier. Sensor-based control architecture for a car-like vehicle. In *Proc. of the IEEE-RSJ Int. Conf. on Intelligent Robots and Systems*, Victoria BC (CA), October 1998.

[6] J.-P. Laumond, A. de Saint Vincent, R. Alami, R. Chatila, and V. Pérébaskine. Supervision and control of the AMR intervention robot. In *Proc. of the Int. Conf. on Advanced Robotics*, volume 2, pages 1057–1062, Pisa (IT), June 1991.

[7] W. L. Nelson. Continuous curvature paths for autonomous vehicles. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, volume 3, pages 1260–1264, Scottsdale AZ (US), May 1989.

[8] M. Parent and P. Daviet. Automated urban vehicles: towards a dual mode PRT (personal rapid transit). In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pages 3129–3134, Minneapolis MN (US), April 1996.

[9] I. Paromtchik, Ph. Garnier, and Ch. Laugier. Autonomous maneuvers of a nonholonomic vehicle. In *Proc. of the Int. Symp. on Experimental Robotics*, Barcelona (ES), June 1997.

[10] I. E. Paromtchik and C. Laugier. Motion generation and control for parking an autonomous vehicle. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pages 3117–3122, Minneapolis MN (US), April 1996.

[11] J. A. Reeds and L. A. Shepp. Optimal paths for a car that goes both forwards and backwards. *Pacific Journal of Mathematics*, 145(2):367–393, 1990.

[12] A. Scheuer and Th. Fraichard. Collision-free and continuous-curvature path planning for car-like robots. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, volume 1, pages 867–873, Albuquerque NM (US), April 1997.

[13] D. Simon, B. Espiau, E. Castillo, and K. Kapellos. Computer-aided design of a generic robot controller handling reactivity and real-time control issues. In *IEEE Transactions on Control Systems Technology*, pages 213–229, December 1993.