Proc. of the Intelligent Cars and Automated Highway Systems Workshop.
Held in association with the IEEE-RSJ Int. Conf. on Intelligent Robots and Systems.
September 11, 1997. Grenoble (FR).

Motion Autonomy Through Sensor-Guided Manœuvres

Ph. Garnier, Th. Fraichard, C. Laugier,
I. Paromtchik and A. Scheuer
Inria[a] Rhône-Alpes & Gravir[b]
Zirst. 655 avenue de l'Europe. 38330 Montbonnot Saint Martin. France

August 21, 1997

This paper deals with a novel motion control approach for a car-like vehicle evolving in a structured, dynamic and partially known environment. The overall architecture of the control system is presented. We focus on two modules: the Global Trajectory Planner (GTP) and the Manœuvre Execution (ME). The key idea of the approach is to plan and carry out sensor-guided manœuvres. A nominal trajectory is generated, based on monitoring the environment and a prediction of its evolution. In order to take into account unforeseen events, the motion control is carried out within the reactive scheme. The automatic vehicle adapts its nominal trajectory to avoid obstacles in a reactive way. Since replanning is time consuming, local trajectories associated with generic manœuvres and based on perceptive information are planned and followed by ME. The approach developed allows to obtain the smooth motion of the vehicle. Experimental results obtained with our automatic car-like vehicle are presented for two kinds of manœuvres : a lane following/changing and an autonomous parallel parking.

Keywords — vehicle, non-holonomic-system, control-architecture, motion-planning, motion-execution.

Acknowledgements — this work was partially supported by the Inria-Inrets[c] Praxitèle programme on urban public transport *[1994-1997]*, and the Inco-Copernicus ERBIC15CT960702 project "Multi-agent robot systems for industrial applications in the transport domain" *[1997-1999]*.

---

[a] Inst. Nat. de Recherche en Informatique et en Automatique.
[b] Lab. d'Informatique GRAphique, VIsion et Robotique de Grenoble.
[c] Inst. Nat. de Recherche sur les Transports et leur Sécurité.

Motion Autonomy Through Sensor-Guided Manœuvres

Ph. Garnier, Th. Fraichard, C. Laugier, I. Paromtchik and A. Scheuer
Inria* Rhône-Alpes — Gravir†
ZIRST. 655 avenue de l'Europe. 38330 Montbonnot Saint Martin. France
Christian.Laugier@inria.fr

This paper deals with a novel motion control approach for a car-like vehicle evolving in a structured, dynamic and partially known environment. The overall architecture of the control system is presented. We focus on two modules: the Global Trajectory Planner (GTP) and the Manœuvre Execution (ME). The key idea of the approach is to plan and carry out sensor-guided manœuvres. A nominal trajectory is generated, based on monitoring the environment and a prediction of its evolution. In order to take into account unforeseen events, the motion control is carried out within the reactive scheme. The automatic vehicle adapts its nominal trajectory to avoid obstacles in a reactive way. Since replanning is time consuming, local trajectories associated with generic manœuvres and based on perceptive information are planned and followed by ME. The approach developed allows to obtain the smooth motion of the vehicle. Experimental results obtained with our automatic car-like vehicle are presented for two kinds of manœuvres : a lane following/changing and an autonomous parallel parking.

1 Introduction

The autonomous manœuvring of nonholonomic vehicles in dynamic environments is being studied by many research teams. The state-of-the-art of this domain reflects approaches of various complexity. A generalized approach involves planning a global path/trajectory generally based on Dubins' curves within an available map of the environment. Because of the computational costs, global planning is usually performed offline. The subsequent following of the planned nominal trajectory involves reactive capabilities, in order to avoid collisions with unexpected obstacles. These two behaviors (trajectory following and obstacle avoidance) are in conflict, their simultaneous operation can lead to an oscillatory motion of the vehicle (if no local trajectory is defined). However, if a nominal trajectory which is obstructed by an obstacle can be modified locally to avoid the obstacle and then return to the nominal trajectory, the oscillations can be eliminated.

In this paper, we present a control architecture in which we focus on two main modules : the Global Trajectory Planner (GTP) and the Manœuvre Execution (ME). GTP allows to generate continuous-curvature trajectories (i.e. trajectories which can be followed by car-like vehicle contrary to Dubins' curves). ME executes the nominal trajectory following or, if necessary, a local trajectory obtained from a generic manœuvre (i.e. a type of manœuvre with some parameters in order to determine the local trajectory). Two manœuvres are considered : a lane following/changing and an autonomous parallel parking with experimental results obtained using our car-like automatic vehicle.This research work contributes to the French Praxitele programme that aims to develop a new urban transportation system based on a fleet of electric computer-driven vehicles [1].

A kinematic model of a car-like vehicle is shown in Fig. 1. The vehicle's coordinates are denoted as a configuration $q = (x,\, y,\, \theta)^T$ relative to some reference coordinate system where $x = x(t)$ and $y = y(t)$ are the coordinates of the midpoint of the rear wheel axle, $\theta = \theta(t)$ is the orientation of the vehicle, and t is time. The motion of the vehicle is described by the equations

$$\begin{cases} \dot{x} = v\,\cos\phi\,\cos\theta, \\ \dot{y} = v\,\cos\phi\,\sin\theta, \\ \dot{\theta} = \frac{v}{L}\,\sin\phi, \end{cases} \tag{1}$$

where $\phi = \phi(t)$ is the steering angle, $v = v(t)$ is the locomotion velocity of the midpoint of the front wheel axle, and L is the wheel base. The steering angle and locomotion velocity are two control commands $(\phi,\, v)$. Equations (1) correspond to a system with nonholonomic constraints because they involve the derivatives of the coordinates of the vehicle and are non-integrable [2]. Equations (1) are valid for a vehicle moving on flat ground with a pure rolling contact without slippage between the wheels and the ground. This purely kinematic model of the vehicle is adequate to control low-speed motions, e.g. during parallel parking or lane following/changing in areas where only low-speed motions are allowed. For the

---

*Inst. Nat. de Recherche en Informatique et en Automatique.
†Lab. d'informatique GRAphique, VIsion et Robotique de Grenoble.

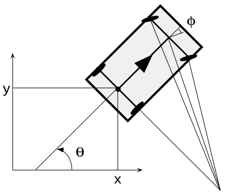

Figure 1: Kinematic model of a vehicle with front wheel steering

high-speed motions, the dynamics of the vehicle must also be considered.

The notion "automatic vehicle" means that the vehicle is equiped with: (1) - a sensor unit to measure relative distances between the vehicle and environmental objects, (2) - a servo unit for low-level control of the steering angle and locomotion velocity, (3) - a control unit that processes data from the sensor and servo units and "drives" the vehicle by issuing appropriate servo commands. The sensor unit uses range sensors to measure relative distances between the vehicle and environmental objects. The servo unit consists of a steering wheel servo-system, a locomotion servo-system for forward and backward motions, and a brake servo-system to slow down and stop the vehicle. The microcomputer-based control unit monitors the current steering angle, locomotion velocity, travelled distance, coordinates of the vehicle and range data from the environment, calculates an appropriate local trajectory and issues the required servo commands.

2 The Overall Architecture

Our control architecture is shown in Fig. 2. It fits the 'Perception-Decision-Action' paradigm. In the decision part, we focus on two main modules :

- the *Global Trajectory Planner* module computes a nominal trajectory between the current configuration of the vehicle and its goal.
- the *Manœuvre Execution* module selects the most appropriate generic manœuvre to execute in the current context (e.g. lane following or lane change in a road context), and carries out the selected manœuvre.

Global Trajectory Planner. The purpose of this module is to compute a time-ordered sequence of (configuration, velocity) couples between the current configuration of the vehicle and its goal. The determination of this trajectory relies upon 1) *a priori information* on the environment of the vehicle (e.g. position of the stationary obstacles), 2) *sensory data* (e.g. position and velocity of the moving obstacles) and 3) *prediction* about the evolution of the workspace. GTP has also to respect kinematic, dynamic and collision avoidance constraints.

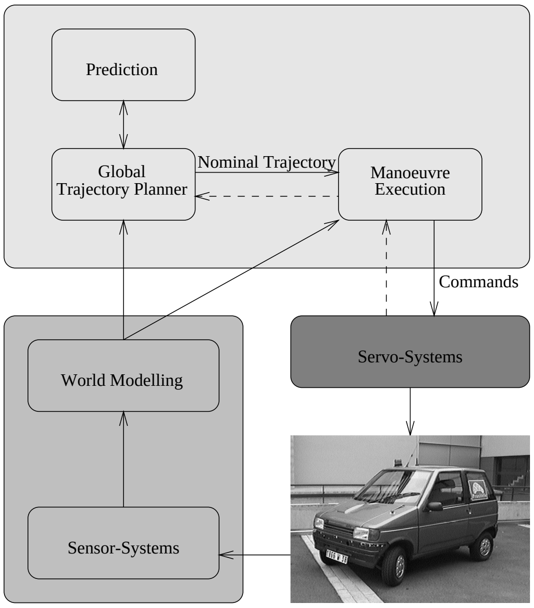

Figure 2: Overall architecture of the control system

Of course the predictions made may not be reliable, it is therefore necessary to give the vehicle the ability to deal with unpredicted events. This is the purpose of ME. It allows to follow the nominal trajectory as closely as possible while reacting in real-time to unexpected events by locally adapting the trajectory actually followed by the vehicle.

Manœuvre Execution. Depending on the context and the current traffic situation, EM selects the most appropriate generic manœuvre to carry out. The effective manœuvre is determined using parameters required for its execution (e.g. road curvature, available lateral and longitudinal displacements relative to the current position of the vehicle, velocity of the vehicle, etc.) Whenever the situation changes (intrusion of unexpected obstacles, etc.), it is detected by ME that decides on a new manœuvre.

ME deals with the coordinated steering and velocity control of the vehicle. For each type of manœuvres, the form of the steering and velocity controls can be predefined (e.g. see [3], [4] for a parking manœuvre). When a type of manœuvre is chosen, the corresponding steering and velocity controls must be computed and applied to the vehicle's servosystems.

Since the form of the controls is known for the manœuvre, the appropriate duration of the manœuvre as well as the magnitude and rate of the steering and velocity controls must be computed according to the parameter values.

GTP and ME are now detailed respectively in sections 4 and 3. For the last one, two kinds of

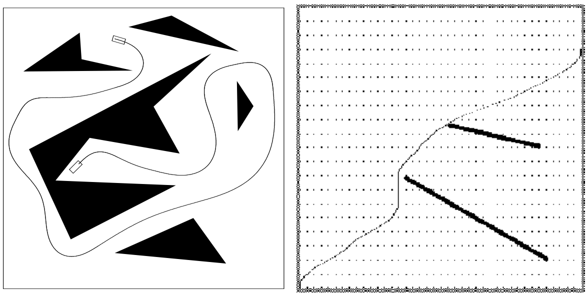

Figure 3: (a) path planning and (b) velocity planning.

manœuvres are considered : a lane following/changing manœuvre in order to avoid obstacles and a parking manœuvre for a vehicle evolving in a structured environment.

3 Global Trajectory Planner

As mentioned earlier, the purpose of the Global Trajectory Planner (GTP) is to compute a nominal trajectory between the current configuration of the vehicle and its goal. The nominal trajectory is a time-ordered sequence of states, i.e. (configuration, velocity) couples, it has to respect different types of constraints :

- *Kinematic constraints:* a car-like vehicle has a limited steering range that restricts the geometric shape of its motion.
- *Dynamic constraints:* these constraints due to engine power, ground-wheel interaction, etc., restrict the accelerations and velocities that can be applied to the vehicle.
- *Collision avoidance constraints:* collision with the stationary and moving obstacles of the environment are forbidden.

Given that a trajectory can be represented also by a geometric path and a velocity profile along this path, GTP addresses the problem at hand in two complementary steps:

1. *Path planning:* a geometric path leading the vehicle to its goal is computed. The left-hand side of Fig. 3 depicts an example of such a path: it is a continuous curve whose curvature is upper-bounded (so as to respect the kinematic constraints of a car-like vehicle). Besides it is collision-free with the stationary obstacles of the environment.
2. *Velocity planning:* the velocity profile of the vehicle along its path is computed; this profile respects the dynamic constraints of the vehicle and yields no collision between the vehicle and the moving obstacles of the environment. Velocity planning requires the knowledge of the future behaviour of the moving obstacles; this information is provided by a prediction module. The right-hand side of Fig. 3 illustrates the velocity planning: it depicts a space-time diagram (the horizontal axis being the position along the path and the vertical one the time dimension). The curve represents the motion of the vehicle through time whereas the thick black lines are the traces left by moving obstacles when they cross the path of the vehicle.

In the next section, we will focus on path planning. The reader is referred to [5] and [6] for more details about velocity planning.

3.1 Path Planning

A car-like vehicle is subject to two non-holonomic constraints: it can only move along a direction perpendicular to its rear wheels axle (continuous tangent direction), and its turning radius is lower bounded (maximum curvature) [7]. Numerous works, e.g. [7, 8, 9, 10], have been done to plan paths for such vehicles, but almost all of them generate sequence of Dubins' curves [11], i.e. paths made of circular arcs connected by tangential line segments. The main reason for this is that these paths are the shortest ones for such a vehicle [11]. The main drawback of these paths is that their curvature is not continuous. Accordingly a vehicle following such a path has to stop at each curvature discontinuity in order to reorient its front wheels.

Since we are mostly interested in planning forward paths only, i.e. paths without manoeuvres, we do not want the vehicle to stop, except possibly at the initial and final configurations. For this reason, we add a continuous-curvature constraint to the classical nonholonomic path planning problem for car-like vehicles. In addition, we introduce a constraint on the curvature derivative; it is upper bounded so as to reflect the fact that the vehicle can only reorient its front wheels with a finite velocity.

Addressing a similar problem (but without the maximum curvature constraint), Boissonnat et al. [12] proved, using the Pontryagin's Maximum Principle, that the shortest path between two vehicle's configurations is made up of line segments and clothoid[1] arcs of maximum curvature derivative. Later, Kostov and Degtiariova-Kostova proved that these shortest paths are, in the general case, made of an infinity of pieces [13, 14].

Similar results can be extended to the particular problem we consider, adding circular arcs of maximum curvature to the set of locally optimal paths. Therefore, in order to come up with a practical solution to the problem at hand, we define a set of paths, derived

---

[1] A clothoid is a curve whose curvature is a linear function of its arc length.

from Dubins' curves, that have continuous curvature and maximum curvature derivative. These paths contain at most eight pieces, each piece being either a line segment, a circular arc of maximum curvature, or a clothoid arc. They are called *SCC-paths* (for Simple Continuous Curvature paths). They are used to design a *local path planner*, i.e. a non-complete collision-free path planner, which in turn is embedded in a global path planning scheme, namely the Probabilistic Path Planner [15]. The result is the first path planner for a car-like vehicle that generates collision-free paths with continuous curvature and maximum curvature derivative.

The reader is referred to [16] for a complete presentation of this continuous curvature path planner.

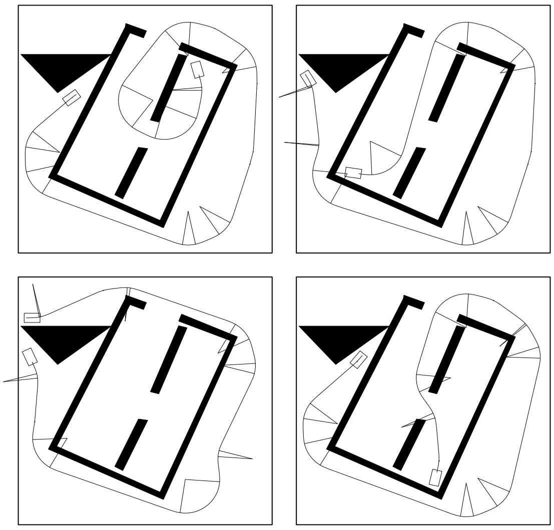

Figure 4: experiments with 4 obstacles.

The path planner was implemented and then tested in two environments taken from Laumond et al. [9] respectively containing four and five obstacles. Both correspond to a 40 m sided square workspace with a 2.5 m long and 1.5 m wide car-like vehicle. Various experimental results are depicted in Figs. 4 and 5.

4 Manœuvre Execution

4.1 Lane Following/Changing

Autonomous lane following is performed by tracking a nominal trajectory delivered by GTP. In the case of unforeseen obstacles, the nominal trajectory is modified on-line, in order to avoid collisions. The modified trajectory has to satisfy temporal motion constraints and avoid collisions. In our earlier experiments, the trajectory following and obstacle avoidance behaviors were decoupled and considered independently, followed by a fuzzy behavior merging process. However, experiments showed that this produced oscillations of the effective motion of the vehicle during obstacle avoidance [17]. To remove these oscillations, a local trajectory is generated that avoids collisions with obstacles detected on the nominal trajectory. The local trajectory also allows the vehicle to catch up with the nominal trajectory (i.e. geometrical path and velocity profile along this path) after the obstacle avoidance. The major difference with the previous behavior-based approach is that the vehicle always follows a specific trajectory.

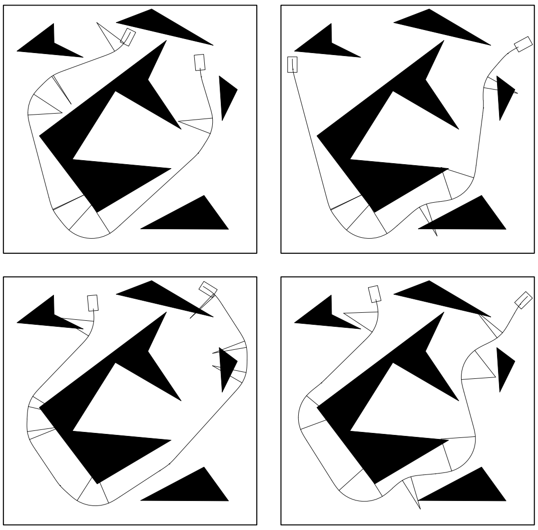

Figure 5: experiments with 5 obstacles.

Lane Following. A method of trajectory following for a nonholonomic vehicle was described in [18]. This method guarantees the stable tracking of a feasible trajectory when the vehicle's control commands are:

$$\dot{\theta} = \dot{\theta}_{ref} + v_{R,ref}(k_y y_e + k_\theta \sin\theta_e), \qquad (2)$$

$$v_R = v_{R,ref} \cos\theta_e + k_x x_e, \qquad (3)$$

where $q_e = (x_e, y_e, \theta_e)^T$ represents the error configuration 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. Lane changing is carried out by generating and following a local trajectory. Such manœuvres are performed when the preplanned nominal trajectory would collide with an unforeseen obstacle. When an obstacle is detected, the nominal trajectory is translated to one side as shown in Fig. 6, in order to avoid collisions with the obstacle. The algorithm for collision avoidance involves the following iterations:

1. Generate a local trajectory which connects the nominal one with a collision-free local trajectory

"parallel" to it (i.e. a parallel translation of the nominal trajectory).

2. Follow the local trajectory until the obstacle is overtaken.

3. Generate a local trajectory which connects the "parallel" trajectory with the nominal one.

4. Follow the local trajectory to catch up with the nominal one.

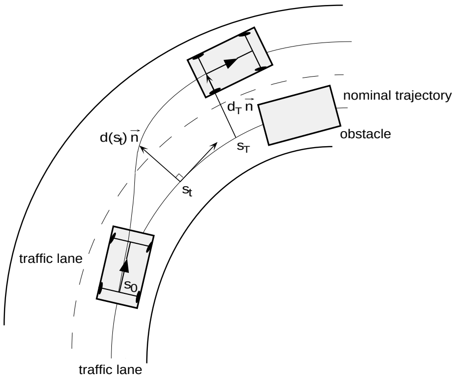

Figure 6: Translation of the nominal trajectory

A feasible trajectory for lane changing is obtained as a quintic polynomial

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

where d_T is a distance between the two traffic lanes, s_T is a length of the nominal trajectory which is necessary to complete the lane changing manœuvre, and $s = s_t$ is a length of the nominal trajectory since the start of the lane changing manœuvre [19]. The distance d_T is supposed to be known. The minimal value of s_T is estimated as

$$s_{T,min} = \frac{\pi\sqrt{k\,d_T}}{2\,\mathcal{C}_{max}}, \quad (5)$$

where $\mathcal{C}_{max}$ stands for the maximum allowed curvature:

$$\mathcal{C}_{max} = \mathbf{min}\left\{\frac{\tan(\phi_{max})}{L}, \frac{\gamma_{max}}{v^2_{R,ref}}\right\}, \quad (6)$$

γ_{max} is the maximum allowed lateral acceleration, and $k > 1$ is an empirical constant (e.g. $k = 1.17$ in our experiments).

When an obstacle is detected, a value $s_{T,min}$ is calculated according to (5) and compared with a distance between the vehicle and the obstacle. A decision is made, either to perform a lane changing manœuvre or to slow down and possibly stop the vehicle. For the lane changing manœuvre, the translation of the nominal trajectory is computed: at each instant t since the start of the manœuvre, 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 velocity vector along the nominal trajectory. The reference orientation θ_{ref} is converted into $\theta_{ref} + \arctan\left(\frac{\partial d}{\partial s}(s_t)\right)$, and the reference velocity $v_{R,ref}$ is obtained as

$$v_{R,ref}(t) = \frac{dist(p_{ref}(t), p_{ref}(t+\Delta t))}{\Delta t}, \quad (7)$$

where $dist$ stands for the euclidean distance.

4.2 Parallel Parking

Autonomous parallel parking involves localizing a sufficient space (parking bay), obtaining a convenient start location for the vehicle relative to the bay, and performing a parallel parking manœuvre. During localization the vehicle moves slowly along the traffic lane. Range data allows a local map of the environment alongside the vehicle to be built. Free spaces are detected, their borders are localized, and their orientation is calculated. The dimensions of the bay are compared with those of the vehicle and a decision on suitability for parking is made.

Drivers know from experience that before the parking manœuvre starts, the vehicle must be oriented near parallel to the parking bay and it must also reach a convenient start position in front of the bay. A start location for parallel parking is shown in Fig. 7 where an automatic vehicle A1 is in a traffic lane. The parking lane with parked vehicles B1, B2 and a parking bay 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 bay. 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 bay with the length L1 and width L2

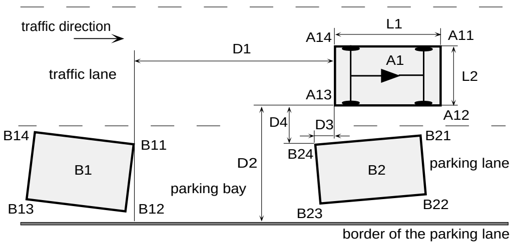

Figure 7: Start location for parallel parking

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 bay is sufficient for parallel parking.

During parallel parking, iterative low-speed backwards-and-forwards motions with coordinated control of the steering angle and locomotion velocity are performed to produce a lateral displacement of the vehicle into the parking bay. 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

$$\theta_0 - \delta_\theta \; < \; \theta_T \; < \; \theta_0 + \delta_\theta, \qquad (8)$$

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 manœuvre [3]:

$$\phi(t) = \phi_{max}\, k_\phi\, A(t), \quad 0 \leq t \leq T, \qquad (9)$$

$$v(t) = v_{max}\, k_v\, B(t), \quad 0 \leq t \leq T, \qquad (10)$$

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 bay relative to the traffic lane, $k_v = \pm 1$ corresponds to forward (+1) or backward (–1) motion,

$$A(t) = \begin{cases} 1, & 0 \leq t < t', \\ \cos \frac{\pi(t-t')}{T^*}, & t' \leq t \leq T - t', \\ -1, & T - t' < t \leq T, \end{cases} \qquad (11)$$

$$B(t) = 0.5 \left(1 - \cos \frac{4\pi t}{T}\right), \quad 0 \leq t \leq T, \qquad (12)$$

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

The commands (9) and (10) are open-loop in the (x, y, θ)-coordinates. The steering wheel servo-system and locomotion servo-system must execute the commands (9) and (10), 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 (10) 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 (9) and (10) is defined by (11) and (12) respectively. In order to evaluate (9)-(12) for the parallel parking manœuvre, the durations T^* and T, the magnitudes ϕ_{max} and v_{max} must be known.

The value of T^* is lower-bounded by the kinematic and dynamic constraints of the steering wheel servo-system. When the control command (9) is applied, the lower bound of T^* is

$$T^*_{min} = \pi \, \mathbf{max} \left\{ \frac{\phi_{max}}{\dot{\phi}_{max}}, \sqrt{\frac{\phi_{max}}{\ddot{\phi}_{max}}} \right\}, \qquad (13)$$

where $\dot{\phi}_{max}$ and $\ddot{\phi}_{max}$ are the maximal admissible steering rate and acceleration respectively for the steering wheel servo-system. The value of T^*_{min} gives duration of the full turn of the steering wheels from $-\phi_{max}$ to ϕ_{max} or vice versa, i.e. one can choose $T^* = T^*_{min}$.

The value of T is lower-bounded by the constraints on the velocity v_{max} and acceleration $\dot{v}_{max}$ and by the condition $T^* < T$. When the control command (10) is applied, the lower bound of T is

$$T_{min} = \mathbf{max} \left\{ \frac{2\pi \, v'(\mathrm{D1})}{\dot{v}_{max}}, \, T^* \right\}, \qquad (14)$$

where the empirically-obtained function $v'(\mathrm{D1}) \leq v_{max}$ serves to provide a smooth motion of the vehicle when the available distance D1 is small.

The computation of T and ϕ_{max} aims to obtain the maximal values such that the following "longitudinal" and "lateral" conditions are still satisfied:

$$|\, (x_T - x_0) \cos\theta_0 + (y_T - y_0) \sin\theta_0 \,| \; < \; \mathrm{D1}, \qquad (15)$$

$$|\, (x_0 - x_T) \sin\theta_0 + (y_T - y_0) \cos\theta_0 \,| \; < \; \mathrm{D2}. \qquad (16)$$

Using the maximal values of T and ϕ_{max} assures that the longitudinal and, especially, lateral displacement of the vehicle is maximal within the available free parking space. The computation is carried out on the basis of the model (1) when the commands (9) and (10) are applied. In this computation, the value of v_{max} must correspond to a safety requirement for parking manœuvres (e.g. $v_{max} = 0.75\, m/s$ was found empirically).

At each iteration i the parallel parking algorithm is summarized as follows:

1. Obtain available longitudinal and lateral displacements D1 and D2 respectively by processing the sensor data.
2. Search for maximal values T and ϕ_{max} by evaluating the model (1) with controls (9), (10) so that conditions (15), (16) are still satisfied.
3. Steer the vehicle by controls (9), (10) while processing the range data for collision avoidance.
4. Obtain the vehicle's location relative to environmental objects at the parking bay. If the "parked" location is reached, stop; else, go to step 1.

When the vehicle A1 moves backwards into the parking bay from the start location shown in Fig. 7, the corner A12 (front right corner of the vehicle) must not collide with the corner B24 (front left corner of the bay). The start location must ensure that the subsequent motions will be collision-free with objects limiting the bay. To obtain a convenient start location, the vehicle has to stop at a distance D3 that will ensure a desired minimal safety distance D5 between the vehicle and the nearest corner of the bay during the subsequent backward motion. The relation between the distances D1, D2, D3, D4 and D5 is described by a function

$$\mathcal{F}(\mathrm{D1},\ \mathrm{D2},\ \mathrm{D3},\ \mathrm{D4},\ \mathrm{D5}) = 0. \qquad (17)$$

This function can not be expressed in closed form, but it can be estimated for a given type of vehicle by using the model (1) when the commands (9) and (10) are applied. The computations are carried out off-line and stored in a look-up table which is used on-line, to obtain an estimate of D3 corresponding to a desired minimal safety distance D5 for given D1, D2 and D4 [4].

When the necessary parking "depth" has been reached, some clearance between the vehicle and the parked ones is provided, i.e. the vehicle moves forwards or backwards so as to be in the middle of the parking bay between the two parked vehicles.

5 Experiments

The developed methods have been tested on an experimental automatic vehicle designed on the base of a LIGIER electric car. This is a four-wheeled vehicle with the front driven and steering wheels. The vehicle can either be driven as a car, or it can move autonomously. To allow autonomous motions, the vehicle is equiped with a control unit based on a Motorola VME162-CPU board and a transputer net. The sensor unit of the vehicle consists of ultrasonic range sensors (Polaroid 9000) and a linear CCD-camera. The steering wheel servo-system is equiped with a direct current motor and an optical encoder to measure the steering angle. The locomotion servo-system of the vehicle is equiped with 12 kW asynchronous motor and two optical encoders at the rear wheels to provide data on locomotion velocity of the vehicle. The vehicle also has an hydraulic braking servo-system. The developed steering and velocity control is implemented using ORCCAD software [20] running on a SUN workstation. The compiled code is transmitted via Ethernet to the VME162-CPU board.

An example of our experimental setup for lane following/changing on a circular road is shown in Fig. 8. The LIGIER vehicle has to follow a nominal trajectory along the circular traffic lane where another vehicle is moving at a lower velocity in front of LIGIER, as shown in Fig. 8a. When the obstacle is detected, a local trajectory for a lane change to the right is generated to avoid collisions, and LIGIER performs the lane changing manœuvre, as illustrated in Fig.8b. Then, LIGIER moves in parallel to its nominal trajectory until the obstacle is overtaken. Further, a new local trajectory for a lane change to the left is generated, and LIGIER performs the lane changing manœuvre to return to its nominal trajectory, as shown in Fig. 8c. Finally, LIGIER continues to follow its nominal trajectory, as illustrated in Fig. 8d.

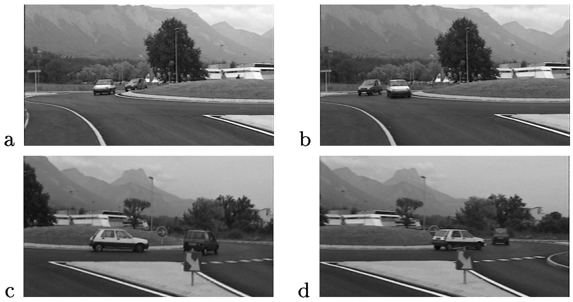

Figure 8: Sequence of motions for lane following/changing on a circular road: a - following the nominal trajectory, b - lane changing to the right and overtaking, c - lane changing to the left, d - continuing with the nominal trajectory

An example of the control commands of the steering angle and locomotion velocity during the lane following/changing manœuvres on a circular road is shown in Fig. 9. The corresponding motion of the vehicle is depicted in Fig. 10 where the nominal circular trajectory and the local one are plotted. The vehicle performs a lane change to the right, moves in parallel to the nominal trajectory and performs a lane change to the left to catch up with its nominal trajectory. The locomotion velocity of the vehicle is increased when it moves along the local trajectory: as it is illustrated in Fig. 10, the duration of the motion along the local trajectory equals the duration of the motion along the nominal trajectory without the lane changing manœuvres.

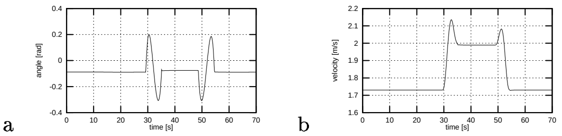

Figure 9: Control commands during lane following/changing: a - steering angle, b - locomotion velocity

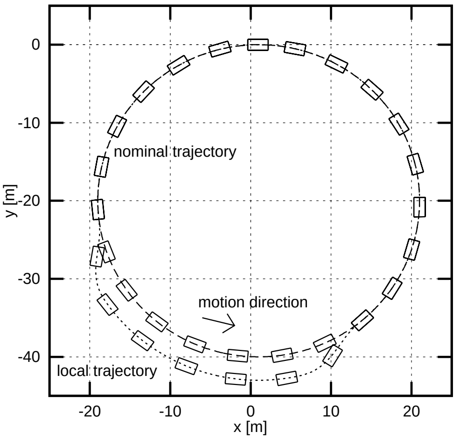

Figure 10: Lane following/changing on a circular road

An example of our experimental setup for autonomous parallel parking in a street is shown in Fig. 11. Autonomous parking can be carried out in an environment where there are moving obstacles, e.g. a pedestrian and another vehicle. As shown in Fig. 11a, the parking bay is in front of LIGIER at its right side between the two vehicles. Initially, LIGIER was driven to a position near the bay, the driver started the autonomous parking and left the vehicle. Then, LIGIER moves forwards autonomously in order to localize the parking bay, obtain a convenient start location and perform a parallel parking manœuvre. When during this motion a pedestrian crosses the street in a dangerous proximity to the vehicle, as shown in Fig. 11a, 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 bay. A decision to carry out the parking manœuvre is made and a convenient start position for the initial backward movement is obtained, as shown in Fig. 11b. Then, LIGIER moves backwards into the bay, as shown in Fig. 11c. During this backward motion, the front human-driven vehicle starts to move backwards, reducing the length of the bay. The change in the environment is detected and taken into account. The range data shows that the necessary "depth" in the bay has not been reached, so further iterative motions are carried out until it has been reached. Then, LIGIER moves to the middle between the rear and front vehicles, as shown in Fig. 11d. The parallel parking manœuvre is completed.

An example of the control commands (9) and (10) for parallel parking into a bay situated at the right side of the vehicle is shown in Fig. 12. The corresponding motion of the vehicle is depicted in Fig. 13 where the motion of the corners of the vehicle and the midpoint of the rear wheel axle is plotted. The available distances are D1=4.9 m, D2=2.7 m relative to the start location of the vehicle. The lateral distance D4=0.6 m was measured by the sensor unit. The longitudinal distance D3=0.8 m was estimated so as to ensure the minimal safety distance D5=0.2 m. In this case, five iterative motions are performed to park the vehicle. As seen in Fig. 12 and Fig. 13, 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 bay (e.g. the values of T, ϕ_{max} and v_{max} differ for the first and last iterative motion).

The developed methods of motion generation and control for the lane following/changing and parallel parking manœuvres were tested. Because the vehicle is equiped with very simple ultrasonic sensors, only low-speed motions were allowed during the experiments. Also, small vertical objects such as posts can not be detected reliably. The execution of the manœuvres was found to be quite sensitive to the calibration of the steering wheel servo-system. To avoid accumulation of errors when computing the position and orientation of the vehicle during the lane following/changing manœuvres, landmarks were to be used. In the future, the experimental vehicle will be equiped with a more advanced sensor system.

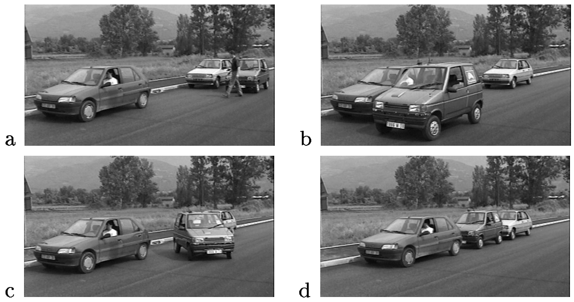

Figure 11: Sequence of motions for parallel parking: a - autonomous motion to localize a parking bay, b - obtaining a convenient start location, c - backward motion into the bay, d - parallel parking is completed

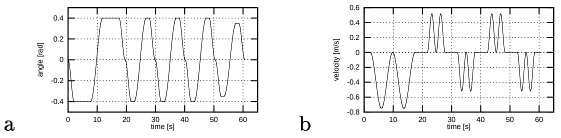

Figure 12: Control commands for parallel parking when backward and forward motions are performed: a - steering angle, b - locomotion velocity

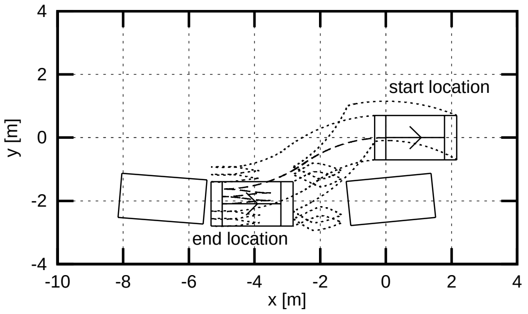

Figure 13: Parallel parking when backward and forward motions are performed

6 Conclusion

In this paper, we have presented a control architecture for an automatic car-like vehicle evolving in a structured dynamic environment. A Global Trajectory Planner (GTP) and a Manœuvre Execution (ME) modules are detailed. GTP generates nominal continuous-curvature trajectories, taking into account kinematic, dynamic and collision-avoidance constraints. It also uses a prediction of the environment in which the vehicle evolves.

When the nominal trajectory cannot be followed accurately, local trajectories are computed by ME, in order to adapt and to catch up with the nominal trajectory.

Motion generation and control methods to perform autonomous lane following/changing and parallel parking manœuvres were developed. The vehicle's constraints were taken into account to obtain feasible trajectories and control commands for the vehicle. The methods developed were implemented on an automatic electric vehicle and experimentally verified. The results obtained show the effectiveness of the developed methods of motion generation and control for autonomous manœuvres.

Acknowledgement

This work was partially supported by the Inria-Inrets[2] Praxitèle programme on urban public transport *[1994-1997]*, and the Inco-Copernicus ERBIC15CT960702 project "Multi-agent robot systems for industrial applications in the transport domain" *[1997-1999]*. The authors would like to thank the members of the Praxitele and Sharp teams for their support during this work.

References

[1] Parent M, Daviet P 1996 Automated urban vehicles: towards a dual mode PRT (Personal Rapid Transit). *Proc. of the IEEE Int. Conf. on Robotics and Automation*, Minneapolis, USA, April 22-28, 1996, pp 3129-3134

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

[3] Paromtchik I E, Laugier C 1996 Motion generation and control for parking an autonomous vehicle. *Proc. of the IEEE Int. Conf. on Robotics and Automation*, Minneapolis, USA, April 22-28, 1996, pp 3117-3122

[4] Paromtchik I E, Laugier C 1996 Autonomous parallel parking of a nonholonomic vehicle. *Proc. of the IEEE Intelligent Vehicles Symp.*, Tokyo, Japan, September 19-20, 1996, pp 13-18

[5] Th. Fraichard and C. Laugier. Dynamic trajectory planning, path-velocity decomposition and adjacent paths. In *Proc. of the Int. Joint Conf. on Artificial Intelligence*, volume 2, pages 1592–1597, Chambéry (FR), September 1993.

[6] Th. Fraichard and A. Scheuer. Car-like robots and moving obstacles. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, volume 1, pages 64–69, San Diego (CA), May 1994.

[7] J. Barraquand and J.-C. Latombe. On nonholonomic mobile robots and optimal maneuvering. *Revue d'Intelligence Artificielle*, 3(2):77–103, 1989.

[8] B. Mirtich. Using skeletons for nonholonomic motion planning. Research Report ESRC 92-16/RAMP 92-6, Departement of Electrical Engineering and Computer Science, University of California, Berkeley, June 1992.

---

[2] Inst. Nat. de Recherche sur les Transports et leur Sécurité.

[9] J.-P. Laumond, P. E. Jacobs, M. Taïx, and R. M. Murray. A motion planner for non-holonomic mobile robots. *IEEE Trans. Robotics and Automation*, 10(5):577–593, October 1994.

[10] P. Švestka and M. H. Overmars. Coordinated motion planning for multiple car-like robots using probabilistic roadmaps. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, volume 2, pages 1631–1636, Nagoya (JP), May 1995.

[11] 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.

[12] J.-D. Boissonnat, A. Cerezo, and J. Leblond. A note on shortest paths in the plane subject to a constraint on the derivative of the curvature. Research Report 2160, Inst. Nat. de Recherche en Informatique et en Automatique, January 1994.

[13] V. Kostov and E. Degtiariova-Kostova. Some properties of clothoids. Research Report 2752, Inst. Nat. de Recherche en Informatique et en Automatique, December 1995.

[14] V. Kostov and E. Degtiariova-Kostova. Irregularity of optimal trajectories in a control problem for a car-like robot. Research Report, Inst. Nat. de Recherche en Informatique et en Automatique, To appear. to appear.

[15] P. Švestka and M. H. Overmars. Probabilistic path planning. Technical Report UU-CS-1995-22, Utrecht University, P.O.Box 80.089, 3508 TB Utrecht, the Netherlands, May 1995.

[16] A. Scheuer and Th. Fraichard. Continuous-curvature path planning for car-like vehicles. In *Proc. of the IEEE-RSJ Int. Conf. on Intelligent Robots and Systems*, Grenoble (FR), September 1997.

[17] Garnier Ph, Fraichard Th 1996 A fuzzy motion controller for a car-like vehicle. *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Osaka, Japan, November 4-8, 1996, pp 1171-1178

[18] Kanayama Y, Kimura Y, Miyazaki F, Noguchi T 1991 A stable tracking control method for a nonholonomic mobile robot. *IEEE/RSJ Int. Workshop on Intelligent Robots and Systems*, Osaka, Japan, November, 1991, pp 1236-1241

[19] 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), May 1989.

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