Fuzzy Control to Drive Car-Like Vehicles

Th. Fraichard and Ph. Garnier

Inria Rhône-Alpes
Zirst. 655 av. de l'Europe. 38330 Montbonnot Saint Martin, France

Robotics and Autonomous Systems, 34(1):1-22, December 2000

Abstract

The reactive component of a motion control architecture for a car-like vehicle intended to move in dynamic and partially known environments is presented in this paper. It is called the *execution monitor* (EM). Its purpose is to generate commands for the servo-systems of the vehicle so as to follow a given nominal trajectory while reacting in real-time to unexpected events. EM is designed as a fuzzy controller, *i.e.* a control system based upon fuzzy logic, whose main component is a set of fuzzy rules encoding the reactive behaviour of the vehicle. A behaviour-based approach is used to set up the fuzzy rule base: the overall behaviour of the vehicle results from the combination of several basic behaviours (trajectory following, obstacle avoidance, etc.), each of which is encoded by a specific set of rules. This approach permits an easy and incremental construction of the fuzzy rule base and also to develop and test the basic behaviours separately. It is the fuzzy control mechanism that straightforwardly handles the problems of behaviour arbitration and command fusion. The basic behaviour rules are simply obtained through direct encoding of the human expertise about car driving. In addition, weighing coefficients are attached to the rules thus permitting a fine tuning of the influence of each basic behaviour. EM has been implemented and tested on a real computer-controlled car equipped with sensors of limited precision and reliability. Experimental results obtained with the prototype vehicle are presented. They demonstrate the capability of EM to actually control a real vehicle and to perform trajectory following and obstacle avoidance in real outdoor environments by using simple fuzzy behaviours relying upon low-resolution sensor data.

Key words: motion autonomy, mobile robot, control architecture, fuzzy logic, automated learning.

Contents

1 Introduction

Motion autonomy in Robotics may be defined as the ability for a robot to perform a given movement without any external intervention. It is a central problem in Robotics. Depending on the situation considered, motion autonomy is a goal more or less easy to reach. For instance, it is easier to achieve motion autonomy in the case of a manipulator arm operating on an assembly line (a priori known, carefully engineered and highly predictable workspace) than in the case of a planetary rover (unknown, uncertain and little predictable environment).

Motion autonomy, for car-like vehicles in particular, is one of our main research goals. It has been pursued since 1986 within two different research programmes on road transport: Prometheus [1] and Praxitèle [2]. The case of a car-like vehicle moving on a road-like environment permits to address motion autonomy in a fairly general framework: the road network is a perfect example of a complex environment; it features moving obstacles (other vehicles, pedestrians, etc.) that can move fast and whose future behaviour cannot be known a priori let alone predicted reliably. These two features, *i.e. dynamicity* and *uncertainty*, are characteristic of most real environments. Moreover a car is subject to *kinematic* and *dynamic* constraints that can be found in most terrestrial vehicles.

Since 1972 and the Shakey mobile robot [19], many control architectures for robots have been proposed in the literature. They are very different but, to some extent, they all follow the classical Perception-Decision-Action paradigm. They feature *deliberative* and/or *reactive* functions. A deliberative function may be defined as one that is able to build and maintain complex models of the robot environment and to use them to perform high-level reasoning. On the other hand, reasoning is reduced to a minimum in reactive functions; there is a close coupling between perception and action; the models used are less complex and sometimes inexistent. Refs. [18,19] and [39] are representative of the deliberative architectures whereas [5,12] and [42] belong to the reactive ones. It seems now well established that motion autonomy requires both deliberative and reactive functions hence the development of *hybrid* architectures, *e.g.* [1,2,10] or [32].

The control architecture we have developed is designed to plan and control the motion of a car-like vehicle moving in a dynamic and partially known environment. It is hybrid and its decision part comprises both a deliberative

[1] Eurêka EU-153 'PROgraMme for a European Traffic with Highest Efficiency and Unprecedented Safety' *[1986-1994]*.

[2] French research programme on Public Individual Transport *[1994-1997]* `<http://www-rocq.inria.fr/praxitele/>`.

and a reactive function. It relies upon two main complementary functions:

- a *trajectory planner* that computes a nominal trajectory between the current position of the vehicle and its goal (deliberative part).
- an *execution monitor* whose purpose is to pilot the vehicle and endow it with the required reactive capabilities (reactive part).

This paper focuses on the execution monitor, the control architecture and the trajectory planner are presented in §2. The execution monitor, *i.e.* the reactive part of the system, is the module that actually pilots the vehicle. It has to fulfill the following functions:

- Generating the commands for the actuators of the vehicle so as to follow the nominal trajectory as closely as possible.
- Monitoring the execution of the trajectory.
- Reacting in real-time to unexpected events by locally adapting the trajectory actually followed by the vehicle.

The vehicle's environment is dynamic and partially known. Besides sensor information is uncertain and imprecise. Hence no accurate and complete model of the environment is available. As for the vehicle itself, it is a system whose kinematics and dynamics are complex and non-linear. In addition, the interaction between the wheels and the ground are hard to model. These reasons led us to design the execution monitor as a *fuzzy controller*, *i.e.* a control system based upon fuzzy logic [41]. Indeed, fuzzy control does not require a complete mathematical model of the controlled system. Moreover fuzzy logic's capability of approximate reasoning is particularly well suited to handle the many imprecisions and uncertainties prevailing in the situation considered. Finally, the computational efficiency of fuzzy controllers allows real-time operation which is a critical feature in the type of environment considered.

One of the first fuzzy controllers was proposed in [17]. Since then, fuzzy control has yielded a huge amount of research works and has been applied to an impressive range of control problems: cement kilns [14], helicopters [35], subway trains [40], mobile robots (*cf.* [30]), various commercial household appliances (washing machines, cameras, microwave ovens, air conditioners, etc.) and more (*cf.* [15]). As for the control of car-like vehicles, it is a problem that does not seem to have been fully addressed before. Refs. [37,36] consider a model car only and, if [27] does report experiments carried out with a real car, it should be noted that the fuzzy controller described did not actually control the car but delivered commands to the human driver instead. The fuzzy controller described in this paper has been designed and tested on a real car-like vehicle moving in an outdoor road-like environment.

Though sharing a common structure [3], the fuzzy controllers developed in the past twenty years can be very different in many respects: fuzzification/defuzzification strategies, fuzzy rules and fuzzy sets definition, decision-making logic, knowledge base organization, etc., and it is hard to compare them and decide which feature is more appropriate to a given problem (*cf.* the comparative reviews established in [4,15] or [30]). In spite of these differences, the fact remains that the key part in the design of a fuzzy controller lies in the setting up of the fuzzy rule base. Accordingly the next paragraph underlines the principles that have guided the design of the fuzzy rule base of the execution monitor.

Four classes of methods to derive the fuzzy rule base stand out [34], they are based upon: (a) expert experience, (b) the operator's control actions, (c) a fuzzy model of the process and (d) learning. The first one that consists in verbalizing the human expertise and expressing it under the form of fuzzy rules is the most widely used and it is the one that has been selected in order to set up the fuzzy rule base of the execution monitor. Besides, following [2,5] or [26], it was decided to use a behaviour-based approach. In other words, the overall reactive behaviour of the vehicle results from the combination of several basic behaviours. From a practical point of view, behaviours are implemented through a modular organization of the fuzzy rule base (like [29] or [38]). Each basic behaviour is encoded by a subset of the fuzzy rule base and it is the fuzzy control mechanism that straightforwardly handles the problems of behaviour arbitration and command fusion. However, in order to better manage the interactions between the basic behaviours and between the rules of a given basic behaviour, it was decided to attach weighing coefficients to the rules (like [29]) thus permitting a fine tuning of their respective influence.

The execution monitor was developed on a car-like vehicle simulator first. Then it was implemented and tested on a real car-like vehicle moving outdoors in a road-like environment. Promising results of experiments featuring trajectory following and unexpected obstacle avoidance have been obtained. They demonstrate the execution monitor's capability to actually control a real vehicle in a dynamic and partially known environment.

The paper is organized as follows: §2 gives a short presentation of the overall control architecture of the vehicle. Afterwards §3 presents the features that are specific to the execution monitor as a fuzzy controller. Then §4 focuses on its application to car-like vehicle's driving with details about the knowledge base that we have developed. Experimental results obtained with a real car presented in §5. Finally future developments are discussed in §6.

[3] The structure of a fuzzy controller is recalled in §3.

2 Control Architecture

As mentioned earlier, motion autonomy in a dynamic and partially known environment requires both high-level reasoning capabilities and reactivity (so as to be able to deal with unexpected events in a timely manner). The architecture we have developed follows the Perception-Decision-Action paradigm. It is depicted in Fig. 1. The decision part comprises both a deliberative and a reactive function: namely the *trajectory planner* and the *execution monitor*.

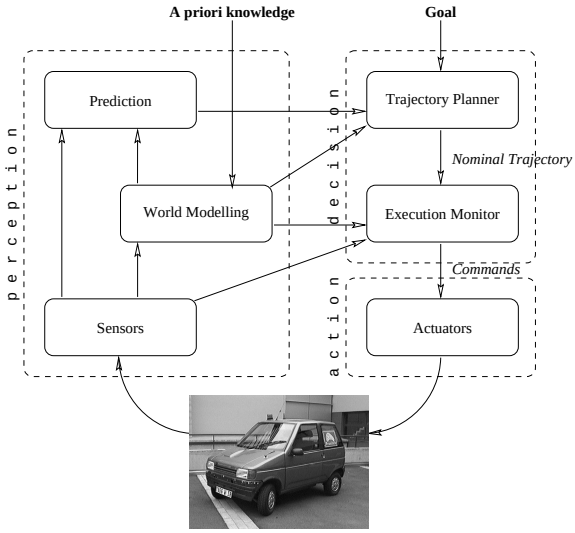

Fig. 1. the control architecture of the vehicle.

Given a goal configuration[4], the trajectory planner computes a nominal trajectory for the vehicle, *i.e.* 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:

- A priori information on the environment of the vehicle (*e.g.* position of the stationary obstacles).
- Sensor data (*e.g.* position and velocity of the moving obstacles).
- Hypotheses about the future evolution of the workspace (*e.g.* prediction of the future behaviour of the moving obstacles).

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 the execution monitor, it generates commands for the actuators of the vehicle so as 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. Additionally the execution monitor may also have to

[4] The configuration of a robotic system is an independent set of parametres that uniquely specify the position and orientation of every part of the system.

reinvoke the trajectory planner when it becomes impossible to catch up with the nominal trajectory.

The detailed presentation of the overall architecture is beyond the scope of this paper. The reader is referred to [7,8] and [9] for a complete presentation of the architecture and to [6] and [31] for more details about the trajectory planner.

3 Execution Monitor

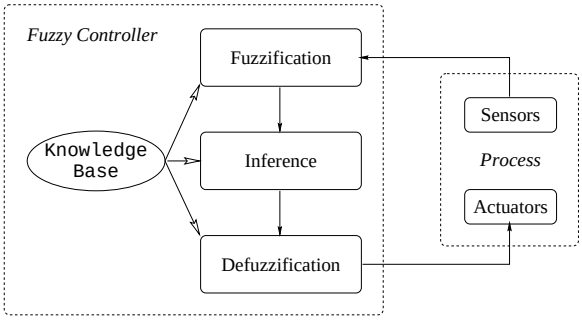

Fig. 2. the architecture of a fuzzy controller.

The execution monitor (EM) is the reactive part of our control architecture. For reasons already discussed in §1, EM was designed as a fuzzy controller (FC). The general architecture of a FC is recalled in Fig. 2 (*cf.* [4] for a presentation of the basic functionalities of a FC). It is a control system whose input is the output of the process to be controlled (sensor data, internal state). Its outputs are commands for the actuators of the process. A FC is made up of four components: the *knowledge base*, the *inference engine*, the *fuzzification* and *defuzzification* modules. The particulars of the four components of EM are given in the next sections.

3.1 Knowledge Base

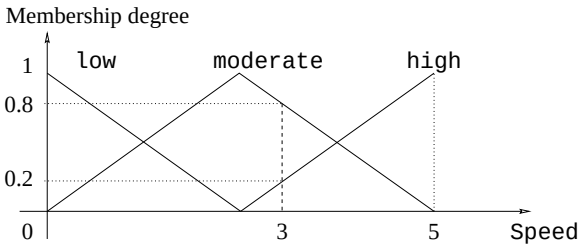

Fig. 3. The three fuzzy sets {`low`, `moderate`, `high`} associated to the fuzzy variable `Speed`. Fuzzification of a 3 m/s. value yields {(`moderate`, 0.8), (`high`, 0.2)}.

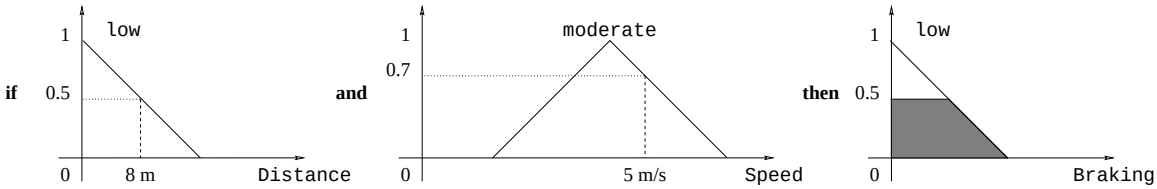

Fig. 4. Applying Mamdani's inference to the fuzzy rule '`If (Distance is low) and (Speed is moderate) then Braking is low`' yields the part of the fuzzy set associated with `Braking` that lies beneath the activation degree (grey area).

The knowledge base includes both fuzzy variables and fuzzy rules. Each fuzzy variable is associated with a term set of fuzzy sets that are characterized by their membership functions. EM uses ordinary normalized triangular-shaped fuzzy sets defining a fuzzy partition of the range of the fuzzy variable (Fig. 3). As for the fuzzy rules, EM uses ordinary rules of the type '`if condition then action`', where the action part is an elementary fuzzy proposition of the form '`fuzzy-variable is fuzzy-set`', and the condition part is a conjunction or disjunction of such propositions. As mentioned earlier in §1, the fuzzy rule base is partitioned in sets of rules representing basic behaviours and weighing coefficients are attached to the fuzzy rules.

3.2 Fuzzification

Classically, the crisp value corresponding to a fuzzy variable is fuzzified into a set of (fuzzy set, membership degree) couples by matching the crisp value against the membership function of each fuzzy set of the fuzzy variable (Fig. 3).

3.3 Inference

EM applies Mamdani's inference (*a.k.a.* minimum inference [17]): to begin with, the *activation degree* of each fuzzy rule is computed; it is a function of the membership degrees of the fuzzy variables present in the condition part of the rule; function that depends upon the definition of the conjunction and disjunction operators. Within EM, they are defined to be the *min* and *max* functions respectively. The result of Mamdani's inference for a given fuzzy rule is the part of the fuzzy set (associated with the variable of the action part of the rule) that lies beneath the activation degree (Fig. 4).

3.4 Defuzzification

EM's defuzzification combines the popular centroidal defuzzification [4] with an additive combination technique [13] so as to better take into account the influence of each and every fuzzy rules that has fired. It also takes into account the weighing coefficients attached to the fuzzy rules. Formally, EM's defuzzification is defined as follows: let $\{f_1, \ldots, f_k\}$ be the set of fuzzy sets inferred for a given fuzzy variable. The crisp value delivered by EM is the abscissa of the following point:

$$\frac{\sum_{i=1}^{k} w_i \, area(f_i) \, coa(f_i)}{\sum_{i=1}^{k} area(f_i)} \tag{1}$$

where $area(f)$ denotes the area of the fuzzy set f, and $coa(f)$ its centre of area. w_i is the weighing coefficient attached to the rule that inferred the fuzzy set f_i.

4 Driving a Car-Like Vehicle with EM

The previous section has presented the features specific to EM as a fuzzy controller. This section focuses on the application of EM to the driving of a car-like vehicle. Like every fuzzy controller, the main component of EM is its knowledge base, *i.e.* the set of fuzzy rules that encode the reactive behaviour of the controlled system. Before detailing the knowledge base that was developed to control a car-like vehicle (§4.2), let us present the model of the vehicle, *i.e.* its input and output (§4.1).

4.1 Input-Output of a Car-Like Vehicle

Is is assumed that a car-like vehicle moves on a planar surface $\mathbb{R}^2$, it has two rear wheels and two directional front wheels. Its current state is defined by the following 5-tuple:

- x, y: the coordinates of the rear axle midpoint R (Fig. 5).
- θ: the vehicle's orientation, *i.e.* the angle between the x axis and the main axis of the vehicle.
- ϕ: the orientation of the front wheels.

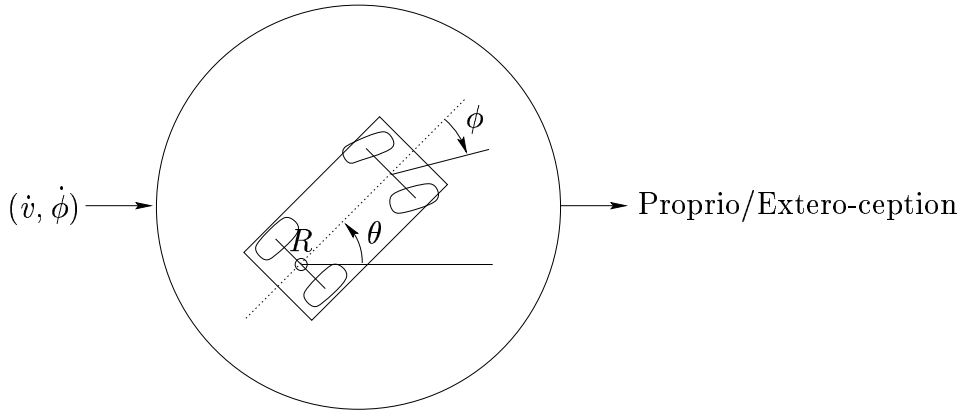

Fig. 5. A car-like vehicle.

- v: the velocity of the reference point R measured along the vehicle's main axis.

To drive a car, one has to control the steering wheel, the gas pedal and the brakes. Accordingly, the command parametres (input) for the vehicle are:

- The longitudinal acceleration $\dot{v} \in [-\dot{v}_{max}, \dot{v}_{max}]$.
- The steering velocity $\dot{\phi} \in [-\dot{\phi}_{max}, \dot{\phi}_{max}]$.

$\dot{v}$ is related to the gas and brake pedals whereas $\dot{\phi}$ is related to the steering wheel. The focus of our work is on normal driving situation where the vehicle does not make back and forth motions, v is therefore never negative (hence sending a negative acceleration to the vehicle at rest is effectless). The vehicle's output is made up of information provided by different types of sensors:

- Proprioceptive sensors that give information about the current state of the vehicle: $x(t), y(t), \theta(t), \phi(t)$ and $v(t)$.
- Exteroceptive sensors that give information about the environment of the vehicle, *e.g.* location of the obstacles surrounding the vehicle.

EM takes as input this sensor information and issue $(\dot{v}, \dot{\phi})$ commands for the vehicle. The fuzzy variable corresponding to $\dot{v}$ is associated with a five term set of fuzzy sets defined over the acceleration range $[-\dot{v}_{max}, \dot{v}_{max}]$: `negative-high`, `negative-low`, `zero`, `positive-low` and `positive-high`. $\dot{\phi}$ is associated with a five term set of fuzzy sets defined over the steering velocity range $[-\dot{\phi}_{max}, \dot{\phi}_{max}]$: `left-high`, `left-low`, `zero`, `right-low` and `right-high`.

Details about the exteroceptive sensor information used by EM are given below in the description of the different set of fuzzy rules included in EM's knowledge base (§4.3, §4.4 and §4.5).

4.2 EM's Knowledge Base

As mentioned earlier in §1, the overall behaviour of the vehicle is obtained through the combination of several basic behaviours, each of which is encoded by a specific set of fuzzy rules. Recall that the main purpose of EM is to follow the nominal trajectory provided by the trajectory planner while reacting in real-time to unexpected events, *i.e.* mostly by avoiding collision with unexpected obstacles. Accordingly, two basic behaviours were developed first: *trajectory following* and *obstacle avoidance*. Other basic behaviours were added afterwards.

The principles underlying each basic behaviour, its fuzzy variables and rules are presented in the next sections. EM issues $(\dot{v}, \dot{\phi})$ commands for the vehicle. Accordingly, the action part of each fuzzy rule features either $\dot{v}$ or $\dot{\phi}$. The input variables present in the condition part of each fuzzy rules are derived from a particular subset of the sensor information or from data provided by the higher level of the control architecture. They are presented along with each basic behaviour. As for the fuzzy rules, they were obtained by a straightforward encoding of the common sense human experience and knowledge about car driving.

The way each basic behaviour works is illustrated by results obtained on a car-like vehicle simulator. This simulator was essential in the design of EM; it was used as a primary test and debugging tool. It helped tuning the weighing coefficients attached to the fuzzy rules and allowed a first validation of EM before the experiments on the real vehicle (*cf.* §5).

4.3 Trajectory Following Behaviour

The primary purpose of EM is to ensure that the vehicle follows the nominal trajectory provided by the trajectory planner. Trajectory following is therefore the first behaviour we designed. The nominal trajectory is a time-ordered sequence of states, *i.e.* a sequence of $(x_d, y_d, \theta_d, v_d)$ t-uple. At each time step t, EM is activated and selects the $(\dot{v}, \dot{\phi})$ command that minimize the error between the vehicle's current state $(x(t), y(t), \theta(t), v(t))$ and the desired one $(x_d(t), y_d(t), \theta_d(t), v_d(t))$. This minimization is achieved thanks to three sets of rules whose respective purposes are to minimize the error in position, orientation and velocity.

4.3.1 Error in position

- Input variables:

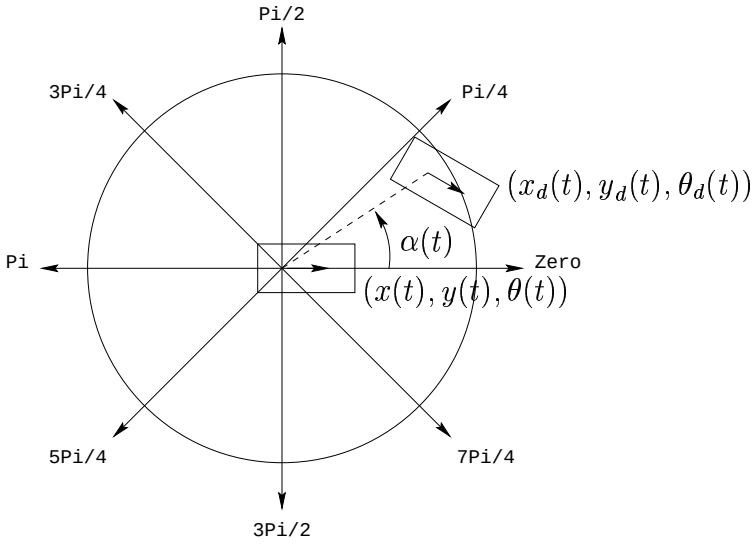

Fig. 6. Error in position between the current and desired states of the vehicle at time t.

- $\alpha(t)$: angular error between the vehicle's current position and the desired one at time t. It is associated with an eight term set of fuzzy sets defined over the $[0, 2\pi[$ range that indicate the relative heading toward the desired position: `zero`, `pi/4`, `pi/2`, `3pi/2`, etc. (Fig. 6):
- $d(t)$: distance between the vehicle's current position and the desired one. It is associated with a three term set of fuzzy sets defined over the $[0, d_{max}]$ range: `very-close`, `close` and `far`.
- *Ahead*(t): boolean variable indicating whether the desired position is ahead of the `Pi/2-3Pi/2` line or not.

• Fuzzy rules (11):

- Rules acting on $\dot{\phi}$:
  1. `if` $\alpha(t)$ `is zero or Pi then` $\dot{\phi}$ `is zero`
  2. `if` $\alpha(t)$ `is Pi/4 or 3Pi/4 then` $\dot{\phi}$ `is left-low`
  3. `if` $\alpha(t)$ `is 5Pi/4 or 7Pi/4 then` $\dot{\phi}$ `is right-low`
  4. `if` $\alpha(t)$ `is Pi/2 then` $\dot{\phi}$ `is left-high`
  5. `if` $\alpha(t)$ `is 3Pi/2 then` $\dot{\phi}$ `is right-high`
- Rules acting on $\dot{v}$:
  6. `if` $d(t)$ `is very-close and` *Ahead*(t) `then` $\dot{v}$ `is zero`
  7. `if` $d(t)$ `is close and` *Ahead*(t) `then` $\dot{v}$ `is positive-low`
  8. `if` $d(t)$ `is far and` *Ahead*(t) `then` $\dot{v}$ `is positive-high`
  9. `if` $d(t)$ `is very-close and not` *Ahead*(t) `then` $\dot{v}$ `is zero`
  10. `if` $d(t)$ `is close and not` *Ahead*(t) `then` $\dot{v}$ `is negative-low`
  11. `if` $d(t)$ `is far and not` *Ahead*(t) `then` $\dot{v}$ `is negative-high`

4.3.2 Error in orientation

• Input variable:

  - $\Delta\theta(t)$: angular difference between the vehicle's current orientation and the desired one at time t; $\Delta\theta(t) = \theta(t) - \theta_d(t)$. It is associated with an eight term set of fuzzy sets: `zero`, `pi/4`, `pi/2`, `3pi/2`, etc.
- Fuzzy rules (5):
  - Rules acting on $\dot{\phi}$:
    1. `if` $\Delta\theta(t)$ `is zero then` $\dot{\phi}$ `is zero`
    2. `if` $\Delta\theta(t)$ `is Pi/4 then` $\dot{\phi}$ `is left-low`
    3. `if` $\Delta\theta(t)$ `is Pi/2 or 3Pi/4 then` $\dot{\phi}$ `is left-high`
    4. `if` $\Delta\theta(t)$ `is 5Pi/4 or 3Pi/2 then` $\dot{\phi}$ `is right-high`
    5. `if` $\Delta\theta(t)$ `is 7Pi/4 then` $\dot{\phi}$ `is right-low`

4.3.3 Error in velocity

- Input variable:
  - $\Delta v(t)$: difference between the vehicle's current velocity and the desired one at time t; $\Delta v(t) = v(t) - v_d(t)$. It is associated with a five term set of fuzzy sets defined over the $[-v_{max}, v_{max}]$ range: `negative-high`, `negative-low`, `zero`, `positive-low` and `positive-high`.
- Fuzzy rules (5):
  - Rules acting on $\dot{v}$:
    1. `if` $\Delta v(t)$ `is zero then` $\dot{v}$ `is zero`
    2. `if` $\Delta v(t)$ `is positive-low then` $\dot{v}$ `is negative-low`
    3. `if` $\Delta v(t)$ `is positive-high then` $\dot{v}$ `is negative-high`
    4. `if` $\Delta v(t)$ `is negative-low then` $\dot{v}$ `is positive-low`
    5. `if` $\Delta v(t)$ `is negative-high then` $\dot{v}$ `is positive-high`

4.3.4 Behaviour Simulation

Given the rules of the trajectory following behaviour, a car-like vehicle simulator was used to test them. Then, in an attempt to validate the trajectory following behaviour, it was compared with a trajectory following controller based upon a control theoretic approach, namely Kanayama's controller [11]. Both controllers were implemented and compared in simulation. Fig. 7 depicts a typical simulation result: given a linear nominal trajectory to be followed at constant speed ($1m.s^{-1}$), and an initial error in position and orientation, the two charts show the evolution of the error in distance and in orientation. Other experiments were carried out with different nominal trajectories (curves) and different initial conditions (*cf.* [7, Chap. 3]). They yield similar results and show that EM compares well with Kanayama's controller whose stability and convergence has been analytically proven. However our controller has two advantages over Kanayama's: a) it can follow a trajectory including stops and b) it can easily incorporate other behaviours.

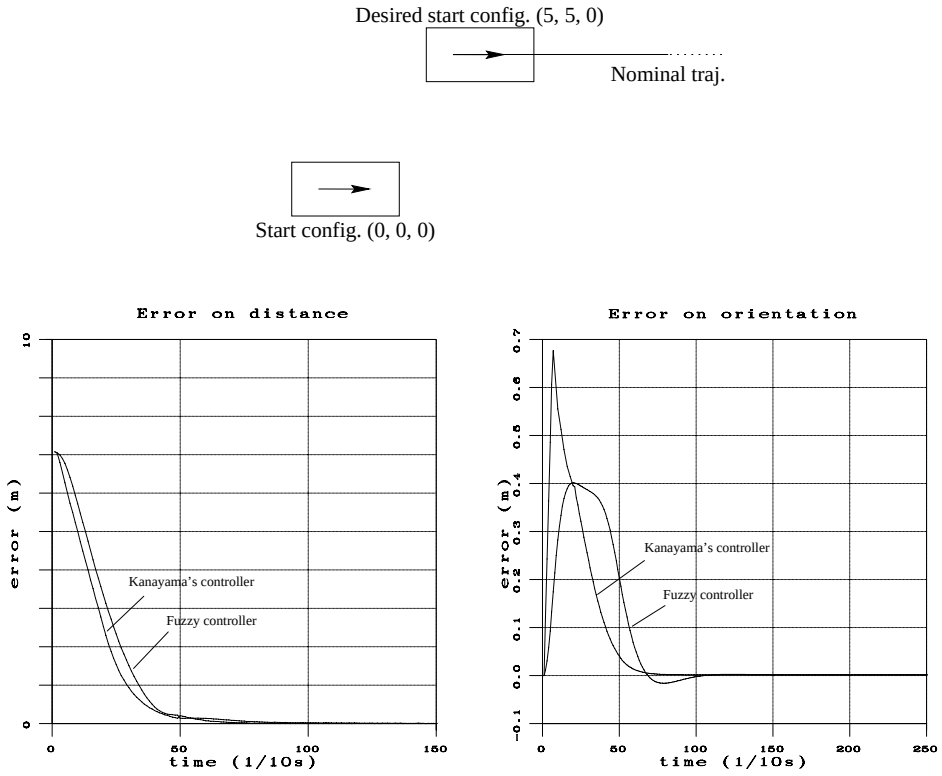

Fig. 7. Trajectory following with EM and Kanayama's controller with plots of the errors in distance and orientation.

4.4 Obstacle Avoidance Behaviour

The nominal trajectory relies upon hypotheses made about the future evolution of the environment and, in particular, the future behaviour of the moving obstacles (§2). Since moving obstacles may 'misbehave', avoiding collision with them is also a fundamental part of EM's task, hence the obstacle avoidance basic behaviour. It uses a model of the actual environment of the vehicle in order to locally adapt the trajectory followed by the vehicle so as to react to unexpected events.

4.4.1 Model of the Environment

Obstacle avoidance requires a model of the actual environment of the vehicle. Since getting a complete and accurate model of a given environment remains a costly process [5], the obstacle avoidance behaviour was designed so as to rely on a relatively poor but easy to get model of the local environment of the vehicle (thanks to range sensors such as ultrasonic sensors). This model is illustrated in Fig. 8. The local environment of the vehicle is divided in

[5] Especially when cameras are used.

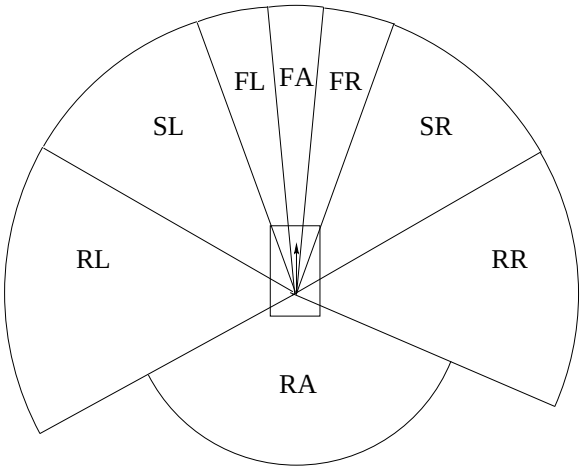

Fig. 8. The eight polar regions of the vehicle's environment (the first letter means either Front, Side or Rear; the second letter means either Ahead, Right or Left.

eight *polar regions*, and the only information used by the obstacle avoidance behaviour is the distance to the closest obstacle in the region and an indication whether the obstacle is moving and approaching the vehicle or not. Poor as it may seem, this model proved sufficient to ensure obstacle avoidance.

Information about the closest obstacle in a given polar region (distance, mobility) is obtained from a priori data (maps of the stationary obstacles), and range data obtained on-line. This information is updated at each time step.

4.4.2 Knowledge Base

To some extent, the polar regions are used to implement the concept of Virtual Deformable Zone introduced in [43]. The intrusion of an obstacle within the area defined by the polar regions yields a reaction of the vehicle and the intensity of the reaction is directly proportional to the intensity of the obstacle's penetration and the vehicle's current velocity. The obstacle avoidance behaviour applies the following principles:

- Obstacles in FL, FA and FR are on the vehicle's trajectory. They yield major velocity and steering changes. The vehicle slows down to avoid frontal collision with FA's obstacles. It steers to avoid FL and FR's obstacles.
- Approaching obstacles in SL and SR yield minor velocity and steering changes. The vehicle slows down to give way to the obstacle.
- Approaching obstacles in RL, RR and RA yield minor velocity changes only. The vehicle speeds up to avoid rear-end collision.

In every circumstances, the obstacle avoidance behaviour steers the vehicle away from the obstacle. Its knowledge base is defined as follows:

- Input variables:
  - $v(t)$: velocity of the reference point R measured along the vehicle's main axis at time t. It is associated with a three term set of fuzzy sets defined over the velocity range $[0, v_{max}]$: `low`, `moderate` and `high`.
  - $d_{pr}(t)$: distance to the closest obstacle in polar region pr. It is associated with a three term set of fuzzy sets defined over the sensor range $[0, r_{max}]$: `very-close`, `close` and `far`.
  - $Approach_{pr}(t)$: boolean variable indicating whether the obstacle in polar region pr is moving and approaching the vehicle or not.
- Fuzzy rules (17):
  - Rules acting on $\dot{\phi}$:
    1. if ($v(t)$ is high and $d_{FR}(t)$ is far) or
       ($v(t)$ is moderate and $d_{FR}(t)$ is close) or
       ($v(t)$ is low and $d_{FR}(t)$ is very-close)
       then $\dot{\phi}$ is left-high
    2. if ($v(t)$ is moderate and $d_{FR}(t)$ is far) or
       ($v(t)$ is low and $d_{FR}(t)$ is close)
       then $\dot{\phi}$ is left-low
    — Repeat the above two rules for FL with right turns.
    5. if $Approach_{SR}(t)$ and
       (($v(t)$ is moderate and $d_{SR}(t)$ is close) or
        ($v(t)$ is low and $d_{SR}(t)$ is very-close))
       then $\dot{\phi}$ is left-high
    6. if $Approach_{SR}(t)$ and
       (($v(t)$ is moderate and $d_{SR}(t)$ is far) or
        ($v(t)$ is low and $d_{SR}(t)$ is close))
       then $\dot{\phi}$ is left-low
    — Repeat the above two rules for SL with right turns.
  - Rules acting on $\dot{v}$:
    9. if ($v(t)$ is high and $d_{FA}(t)$ is far) or
       ($v(t)$ is moderate and $d_{FA}(t)$ is close) or
       ($v(t)$ is low and $d_{FA}(t)$ is very-close)
       then $\dot{v}$ is negative-high
    10. if ($v(t)$ is moderate and $d_{FA}(t)$ is far) or
       ($v(t)$ is low and $d_{FA}(t)$ is close)
       then $\dot{v}$ is negative-low
    11. if $Approach_{SR}(t)$ and
       (($v(t)$ is moderate and $d_{SR}(t)$ is close) or
        ($v(t)$ is low and $d_{SR}(t)$ is very-close))
       then $\dot{v}$ is negative-high
    12. if $Approach_{SR}(t)$ and
       (($v(t)$ is moderate and $d_{SR}(t)$ is far) or
        ($v(t)$ is low and $d_{SR}(t)$ is close))
       then $\dot{v}$ is negative-low

— Repeat the above two rules for SL.

15. `if` $Approach_{RR}(t)$ `and`
    `((`$v(t)$ `is moderate and` $d_{RR}(t)$ `is close) or`
    `(`$v(t)$ `is low and` $d_{RR}(t)$ `is very-close))`
    `then` $\dot{v}$ `is positive-low`

— Repeat the above rule for RL.

17. `if` $Approach_{RA}(t)$ `and`
    `(`$v(t)$ `is low and` $d_{RA}(t)$ `is very-close)`
    `then` $\dot{v}$ `is positive-low`

4.4.3 Behaviour Simulation

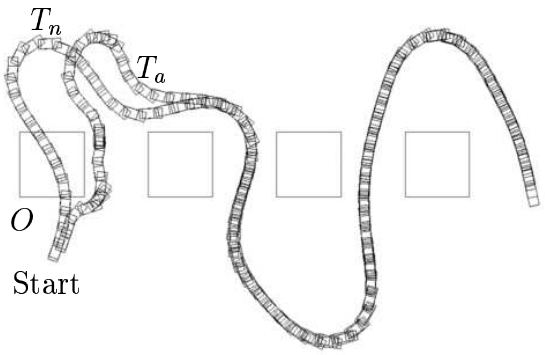

Fig. 9. a collision avoidance test: T_n (resp. T_a) denotes the nominal (resp. actual) trajectory of the vehicle.

Once the obstacle avoidance behaviour was developed, it became possible to test the capabilities of EM regarding local adaptation of the nominal trajectory so as to react to unexpected events. In order to do so, several environments were considered. To begin with, EM was tested on simple environments such as the one depicted in Fig 9 that contains square obstacles. A smooth curve to be followed at constant speed was provided to EM as a nominal trajectory. The nominal trajectory T_n had been computed in the absence of the square obstacle O. O was added afterwards and Fig 9 illustrates how the vehicle avoids the collision with O. Then, we considered more complex environments such as the road network. Several scenarios including moving obstacles were designed, *e.g.* crossing an intersection, overtaking another vehicle, etc. One of these scenario is illustrated in Fig. 10. Three vehicles are approaching an intersection. The white one is controlled by EM ('our vehicle'); the other two are regarded as moving obstacles. Based upon a prediction of the motion of the two other vehicles (constant speed), the trajectory planner computes a nominal trajectory such that our vehicle gives way to the vehicle on its left and crosses the intersection before the vehicle on its right. However, in the course of the execution of the trajectory, it turns out that the vehicle on the right accelerates unexpectedly. The intrusion of this vehicle in the SR region of

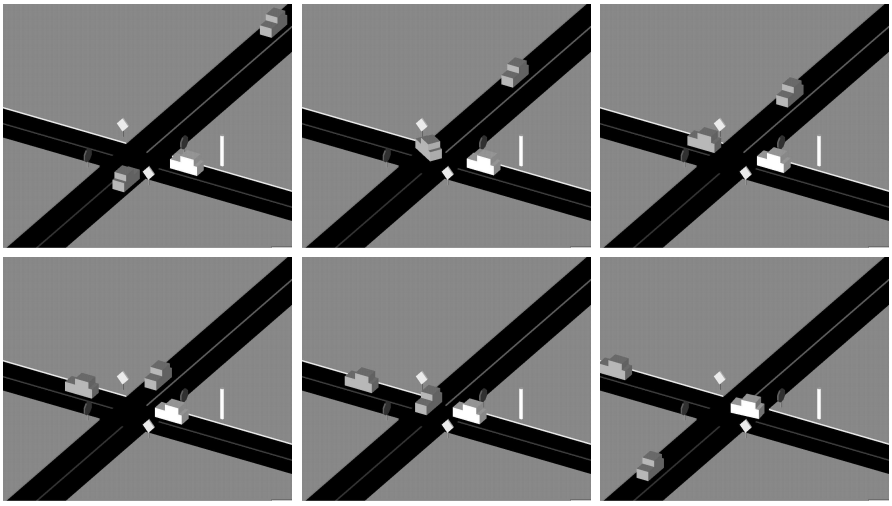

Fig. 10. an intersection crossing scenario.

our vehicle give more strength to rules #11 and #12 of the obstacle avoidance behaviour. As a result, our vehicle slows down. Later, the other vehicle reaches FA through FR thus activating rule #10. Our vehicle slows down again until it eventually stops. When the other vehicle leaves FA for FL, the obstacle avoidance behaviour becomes inactive, trajectory following resumes and our vehicle reaccelerates so as to catch up with its nominal trajectory.

4.5 Additional Behaviours

4.5.1 Trajectory Planner Activation

The purpose of the first additional behaviour that was added to EM was to monitor whether it remained possible to catch up with the nominal trajectory. If not, *i.e.* if the difference between the current state and the desired one becomes too important (in terms of distance), a boolean flag is raised so as to warn the decision level that a new nominal trajectory is needed. In the meantime, the vehicle slows down.

- Input variable:
  - *TooFar*(t): boolean variable true if $d(t) > d_{max}$, *i.e.* if the difference between the current and the desired position of the vehicle is too large (*cf.* §4.3.1).
  - *NeedTrajectory*: boolean variable indicating whether a new nominal trajectory has been asked. When the decision level provides it, this variable is reset to false.
- Fuzzy rules (2):
  1. `if not` *NeedTrajectory* `and` *TooFar*(t)
     `then` *NeedTrajectory* `is true`

2. `if` *NeedTrajectory* `then` $\dot{v}$ `is negative-high`

4.5.2 Avoid to the Left Policy

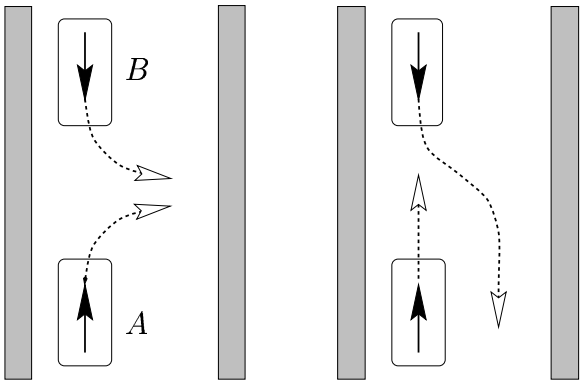

Fig. 11. a deadlock situation and its resolution thanks to an 'avoid to the left' policy.

Drawing upon Zeghal and Ferber's experience in the domain of collision avoidance between aircrafts [44], an 'avoid to the left' policy was added to EM. In other words, whenever the vehicle faces a moving obstacle, it tries to get round it on the left side. Zeghal and Ferber have shown that, in an environment where several vehicles apply this policy, a form of passive [6] cooperation between them results that reduces the risk of deadlocks that would otherwise occur in situations such as the one depicted in the left-hand side of Fig. 11. In this situation, each vehicle has only one option to avoid the other vehicle (vehicle A must turn to the right and vehicle B to the left). If they both do so then a deadlock occurs. On the other hand, if they both (or, at least, one of them) follow an 'avoid to the left' policy then the conflict will be solved as illustrated in the right-hand side of Fig. 11. The knowledge base of the 'avoid to the left' behaviour is defined as follows:

- Input variable:
  - $v(t)$: velocity of the reference point R (*cf.* §4.4).
  - $d_{pr}(t)$: distance to the closest obstacle in polar region pr (*cf.* §4.4).
  - $Moving_{pr}(t)$: boolean variable indicating whether the obstacle in polar region pr is moving or not.
- Fuzzy rules (6):
  - Rules acting on $\dot{\phi}$:
    1. `if` $Moving_{FR}(t)$ `and`
       `((`$v(t)$ `is high and` $d_{FR}(t)$ `is far) or`
       `(`$v(t)$ `is moderate and` $d_{FR}(t)$ `is close) or`
       `(`$v(t)$ `is low and` $d_{FR}(t)$ `is very-close))`
       `then` $\dot{\phi}$ `is left-high`

[6] Passive in the sense that there is no exchange of information.

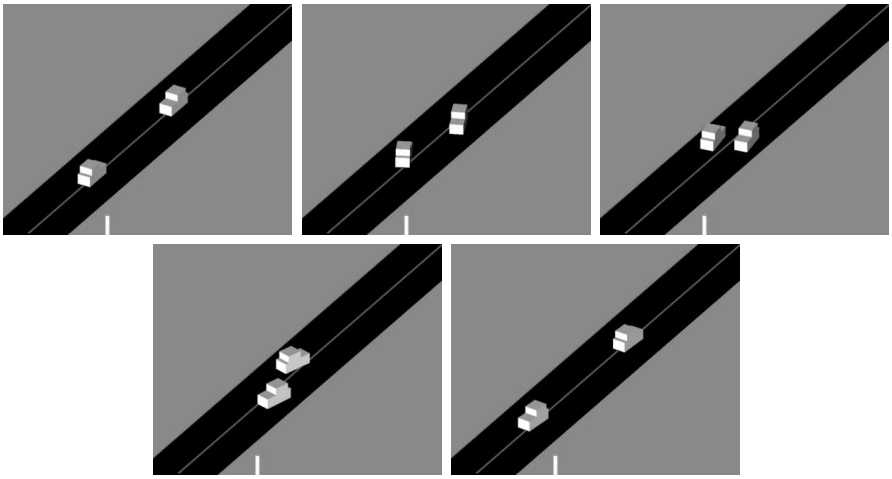

Fig. 12. The 'avoid to the left' policy

2. `if` $Moving_{FR}(t)$ `and`
   `((`$v(t)$ `is moderate and` $d_{FR}(t)$ `is far) or`
   `(`$v(t)$ `is low and` $d_{FR}(t)$ `is close))`
   `then` $\dot{\phi}$ `is left-low`

— Repeat the above two rules for FA and FL.

Fig. 12 depicts a simulation run with EM featuring the 'avoid to the left' policy. Two vehicles, both under the control of an EM, are following the same straight nominal trajectory in opposite directions. When approaching from one another, obstacle avoidance is in order and the 'avoid to the left' policy becomes active too. Both vehicles steer away from the obstacle on their left before returning to the nominal trajectory when the obstacle has disappeared.

5 Experimental Results

Up until now, EM has been developed and tested on a car-like vehicle simulator. The next step was to use a real vehicle. We started to work with an autonomous electric car prototype that we are currently developing within the framework of Praxitèle, a French research programme on road transport. The next sections will respectively present the Praxitèle programme (§5.1), the experimental vehicle used (§5.2), and the driving experiments carried out so far (§5.3).

5.1 The Praxitèle Programme

Praxitèle is a French research programme that investigates the concept of Public Individual Transport. It is a novel transportation system based upon a

fleet of small public cars that are supervised by a central computer [3,23,24]. Its aim is to develop a complement to mass transport that would be a better alternative to the use of private automobiles in urban areas (better in terms of congestion, pollution and ease of use). Such a public transportation system could be an extension of a public transport network where the demand does not justify the investment and the operation of a large capacity system. It could also be justified for local trips in specific neighborhoods, or as an alternative to the development of 'park and ride' systems. Within such a transport system, the cars are driven by their users but their operation will be automated in certain circumstances, *e.g.* to move empty cars where they are needed, or within specific environments, *e.g.* parking-maintenance stations, special tracks, etc. This programme, that is highly multidisciplinary, is a perfect opportunity for us to test our motion control architecture in real situations.

5.2 The Experimental Vehicle

Fig. 13. The Ligier prototype.

The experimental validation of our control architecture, EM included, is carried out on a prototype based upon a commercial electric car equipped to be computer-controlled: a Ligier Optima (Fig. 13). It is 2.5 m long and 1.4 m wide. It weighs around 650 kg and can accommodate two people. Like regular cars, it has two rear wheels and two directional front wheels with a maximum steering angle of 23 degrees. It is equipped with a 12 kw asynchronous electric motor powered by lead batteries that can drive the car up to 70 km/h with an average range of 80 km.

The control system of the vehicle includes a Motorola VME bus with a MVME 162 CPU board (68040 processor). This board drives three servo-motors and a three-phase controller. One servo-motor controls the steering wheel, the other two are in charge of the brakes. The three-phase controller drives the electric motor. As far as proprioception is concerned, an optical encoder measures the

steering angle while two optical encoders mounted on the rear wheels provide the longitudinal velocity of the car and a motion estimation. Classical odometric techniques that integrate the information delivered by these sensors provide EM with the required proprioceptive information, *i.e.* $x(t), y(t), \theta(t), \phi(t)$ and $v(t)$.

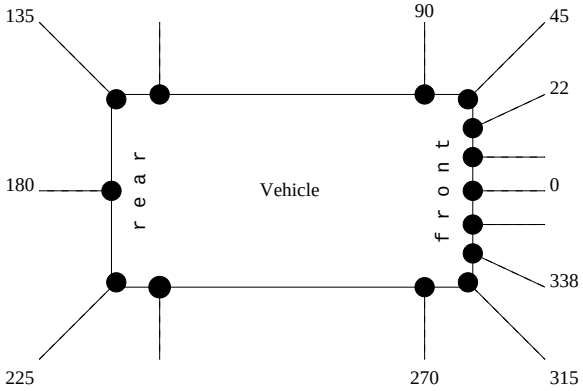

Fig. 14. The ultrasonic sensors layout.

The ligier is equipped with a range measurement system of fourteen Polaroid 9000 ultrasonic sensors whose layout is depicted in Fig. 14. These sensors can detect obstacles in a range of 10 m with a 1 cm resolution. They are driven thanks to a custom-made board including a network of four transputers dynamically linked by a C004 integrated programmable circuit. The ultrasonic sensors are operated using the synchronization techniques developed in [20]. A map of the actual environment of the vehicle is built and this map is matched against the polar regions model so as to provide EM with the necessary exteroceptive information, *i.e.* $d_{pr}(t)$, $Approach_{pr}(t)$ and $Moving_{pr}(t)$. As far as the software design is concerned, the ORCCAD[7] software [33] has been used to implement the different components of our control architecture.

5.3 Trajectory Following with Obstacle Avoidance

v_{max}	$\dot{v}_{max}$	ϕ_{max}	$\dot{\phi}_{max}$	r_{max}	d_{max}
1 m.s^{-1}	0.25 m.s^{-2}	23 deg.	6 deg. s^{-1}	10 m.	5 m.

Table 1

Because testing behaviours such as the 'avoid to the left' or 'give way to the right' policies requires at least two vehicles, our experiments have focused primarily on testing the trajectory following and obstacle avoidance behaviours. The limited range of the ultrasonic sensors further restricted us to slow speed (of the order of 1 m.s^{-1}).

[7] Open Robot Controller Computer-Aided Design.

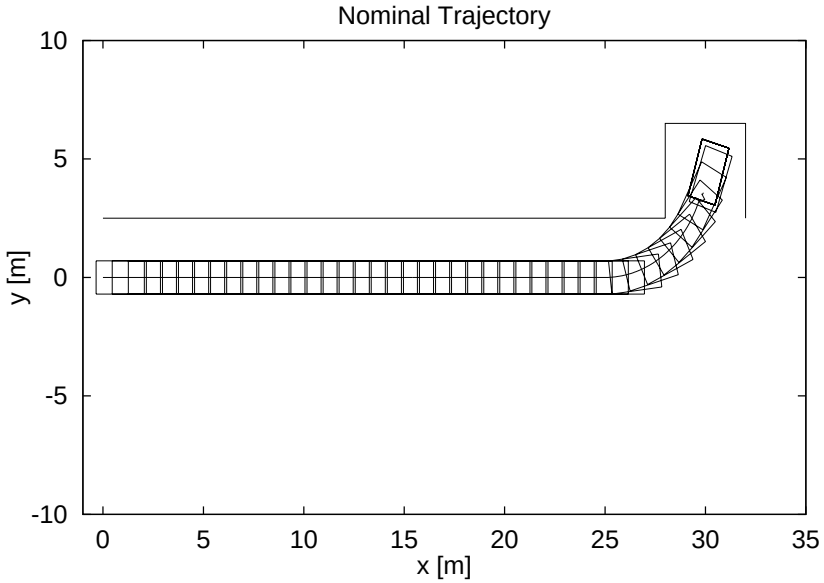

Fig. 15. A trajectory following experiment.

The different parametres used in the definition of the fuzzy variables of the execution monitor (*cf.* §4) were set according to characteristics of the experimental vehicle. They are summarized in table 1.

An actual parking lot was chosen to carry out the experiments. A 40 m long nominal parking trajectory was computed and fed to EM. At first the trajectory following behaviour was tested alone. Fig 15 shows a bird view of the trace of the experiment.

Then unexpected obstacles were introduced in order to test the obstacle avoidance behaviour in various situations. The outcome of an experiment with an unexpected obstacle right on the path of the vehicle is depicted in Fig 16. The bottom part Fig 16 shows the actual velocity followed by the vehicle. Various snapshots of this experiment are presented in Fig. 17.

6 Future Developments

The experiments carried out so far have demonstrated the capability of EM to perform trajectory following and obstacle avoidance in real outdoor environments by using simple fuzzy behaviours relying upon low-resolution sensor data. In this respect, they confirm the point advocated in [28]: outdoor navigation with sensor data of limited precision and reliability is possible thanks to fuzzy logic.

As far as further developments are concerned, they will follow three directions:

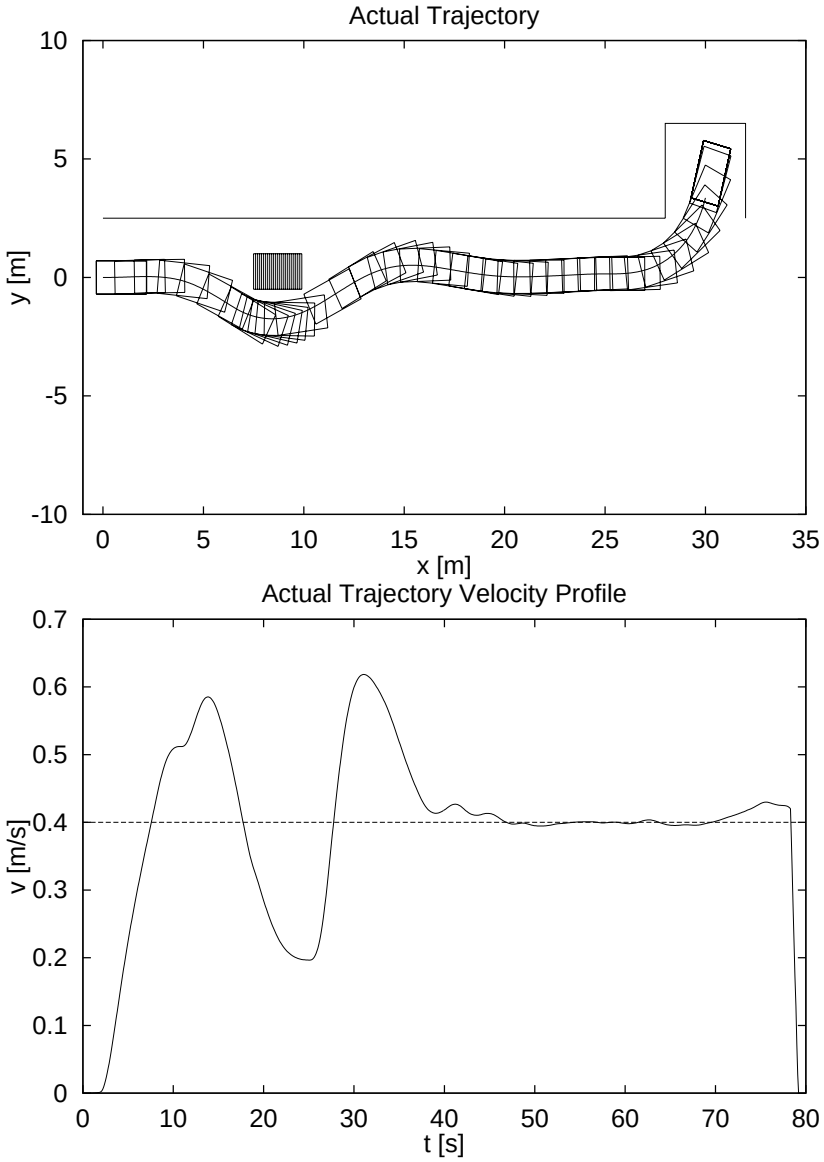

Fig. 16. An obstacle avoidance experiment : trace of the experiment (top), and actual velocity profile followed by the vehicle (bottom).

- To begin with, the limited range of the ultrasonic sensors mounted on our experimental vehicle has restricted us to slow speed (of the order of 1 $\mathrm{m.s^{-1}}$). The next step is to test EM at higher speed. In order to do so, sensors with a better range are required. We are considering the use of a linear camera coupled with an infrared flashlight (a prototype of such a system is described in [16]). Provided that the obstacles are equipped with reflectors (it is the case for other cars), it will be possible to detect them from a distance of up to 30 m. Thus, it should enable us to carry out experiments at higher speed (of the order of 3 $\mathrm{m.s^{-1}}$).
- In spite of the success of the experiments, it remains that EM is purely reflexive in nature. It lacks the looking-ahead capabilities that could prevent oscillatory behaviours to appear (this problem is commonplace in reflexive

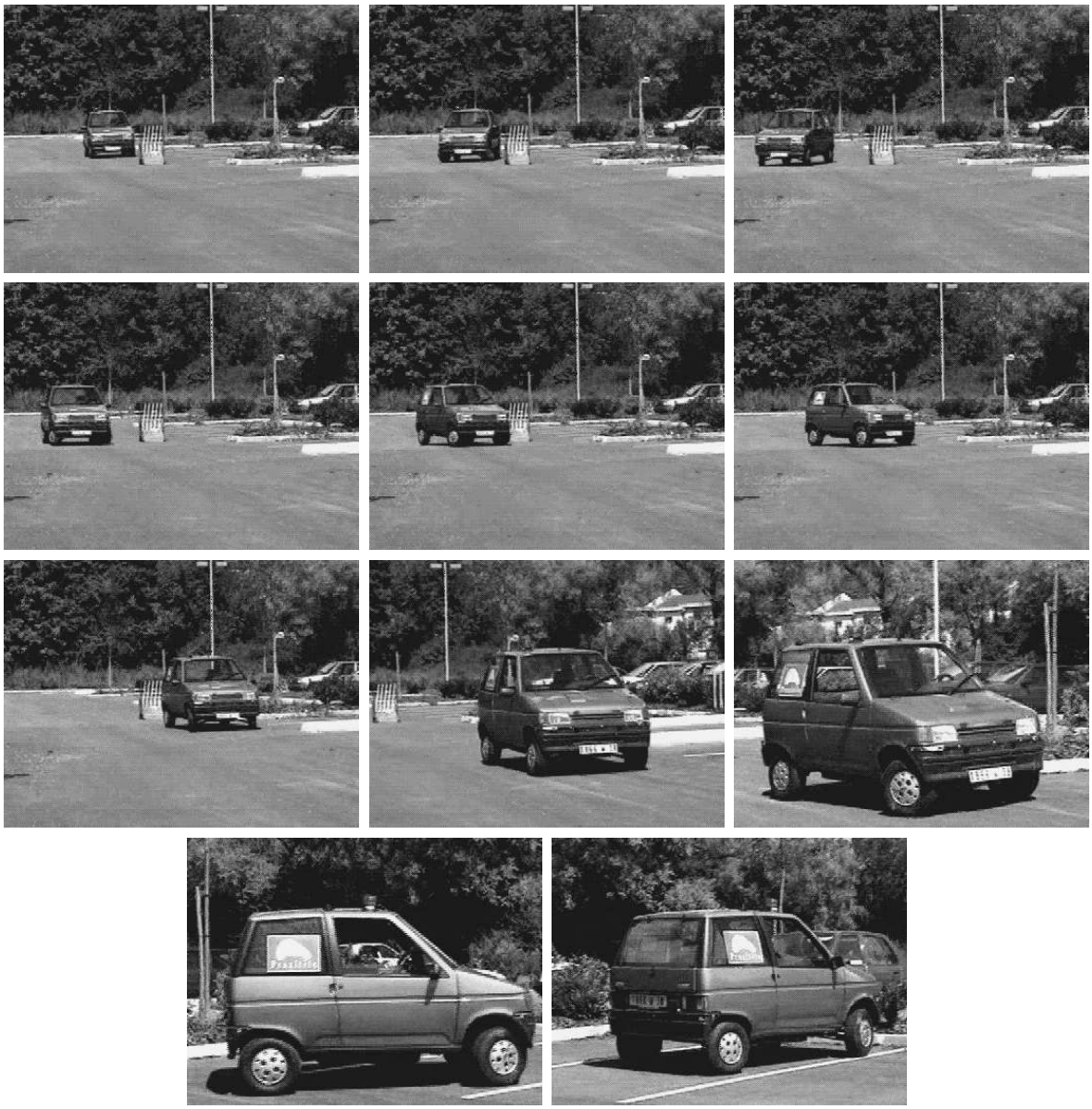

Fig. 17. Snapshots of the experiment depicted in Fig. 16.

systems). In order to deal with this problem, we have decided to introduce an intermediate level between the trajectory planner and EM: the *local navigator* [22]. The purpose of the local navigator is to compute a set of so-called 'escape-lines', *i.e.* trajectories obtained by applying simple commands (typically constant) to the vehicle over a fixed period of time. These escape lines are then checked for possible collision with the obstacles detected by the vehicle's sensors. Finally the best (typically the closest to the nominal trajectory) collision-free escape line is selected as the trajectory to be followed by EM over the next period of time. The activation frequency of the local navigator is one order of magnitude lower than that of the execution monitor so as to ensure continuity in the vehicle's behaviour and to avoid

oscillations. A detailed presentation of the local navigator and a report on preliminary experiments can be found in [21].

- Finally, we have started to experiment with the concept of *generic manœuvres*. A generic manœuvre is a parameterized trajectory, local in nature, that can be used in response to a specific situation. This concept seems very appropriate to a structured environment such as the road network. In this case, it is indeed straightforward to consider manœuvres such as lane following, lane changing, overtaking, parking, etc. These manœuvres can then be used whenever the current situation requires it. This concept along with preliminary results are presented in [25].

7 Conclusion

This paper has presented the *execution monitor* (EM), *i.e.* the reactive component of a motion control architecture for a car-like vehicle intended to move in dynamic and partially known environments. The purpose of EM is to generate commands for the servo-systems of the vehicle so as to follow a given nominal trajectory while reacting in real-time to unexpected events. EM has been designed as a fuzzy controller, *i.e.* a control system based upon fuzzy logic. Its key component is a set of fuzzy rules that encode the reactive behaviour of the vehicle.

A behaviour-based approach was used to set up the fuzzy rule base: the overall behaviour of the vehicle results from the combination of several basic behaviours (trajectory following, obstacle avoidance, etc.), each of which is encoded by a specific set of rules. This approach has permitted an incremental construction of the knowledge base and also to develop and test the basic behaviours separately. It is the fuzzy control mechanism that has straightforwardly handled the problems of behaviour arbitration and command fusion. The basic behaviour rules were simply obtained through direct encoding of the human expertise about car driving. In addition, weighing coefficients were attached to the rules thus permitting a fine tuning of the influence of each basic behaviour.

EM was designed and tested on a car-like vehicle simulator first. Then it was implemented and tested on a real computer-controlled car equipped with sensors of limited precision and reliability. Experimental results obtained with the prototype vehicle have been presented. They have demonstrated the capability of EM to actually control a real vehicle and to perform trajectory following and obstacle avoidance in real outdoor environments by using simple fuzzy behaviours relying upon low-resolution sensor data. Further experiments and developments are underway.

Acknowledgements

This work was partially supported by the Inria [8] -Inrets [9] Praxitèle programme on Public Individual Transport *[1994-1997]*. The authors would like to thank Dr. M. Cherif and the anonymous reviewers for their valuable comments on the earlier versions of the paper.

References

[1] R. Alami, R. Chatila, S. Fleury, M. Ghallab, and F. Ingrand. An architecture for autonomy. *Int. Journal of Robotics Research*, 17(4):315–337, April 1998. Special issue on integrated architectures for robot control and programming.

[2] R. C. Arkin. Motor schema based navigation for a mobile robot. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pages 264–271, Raleigh, NC (US), April 1987.

[3] D. Augello, E. Bénéjam, J.-P. Nerrière, and M. Parent. Complementarity between public transport and a car sharing service. In *Proc. of the World Congress on Appplication of Transport Telematics and Intelligent Vehicle Highway System*, Paris (FR), November–December 1994.

[4] H. R. Berenji. Fuzzy logic controllers. In R. R. Yager and L. A. Zadeh, editors, *An introduction to fuzzy logic applications in intelligent systems*, pages 69–96. Kluwer Academic Press, 1992.

[5] R. A. Brooks. A robust layered control system for a mobile robot. In G. Shafer and J. Perl, editors, *Readings in Uncertain Reasoning*, pages 204–213. Morgan Kaufmann, 1990.

[6] Th. Fraichard. Trajectory planning in a dynamic workspace: a 'state-time' approach. *Advanced Robotics*, 13(1):75–94, 1999.

[7] Ph. Garnier. *Contrôle d'exécution réactif de mouvement de véhicules en environnement dynamique structuré.* Thèse de doctorat, Inst. Nat. Polytechnique de Grenoble, (FR), December 1995.

[8] Ph. Garnier, C. Novales, and C. Laugier. An hybrid motion controller for a real car-like robot evolving in a multi-vehicle environment. In *Proc. of the IEEE Int. Symp. on Intelligent Vehicles*, pages 326–331, Detroit, MI (US), September 1995.

[9] Ph. Garnier, C. Novales, D. Pallard, and G. Baille. Autonomy for electric cars. In *Proc. of the Electric Vehicle Technology Conf.*, volume 2, pages 24–33, Paris (FR), November 1995.

[8] Institut National de Recherche en Informatique et Automatique.
[9] Institut National de Recherche sur les Transports et leur Sécurité.

[10] E. Gat. On three-layer architectures. In D. Kortenkamp, R. P. Bonnasso, and R. Murphy, editors, *Artificial Intelligence and Mobile Robots*. MIT/AAAI Press, 1997.

[11] 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*, pages 1236–1241, Osaka (JP), November 1991.

[12] M. Khatib and R. Chatila. An extended potential field approach for mobile robot sensor-based motions. In *Proc. of Intelligent Autonomous Systems*, pages 490–496, March 1995.

[13] B. Kosko. *Neural networks and fuzzy systems. A dynamical systems approach to machine intelligence*. Prentice-Hall Int., 1992.

[14] P. M. Larsen. Industrial applications of fuzzy logic control. *Int. Jour. of Man-Machine Studies*, 12(1):3–10, 1980.

[15] C. C. Lee. Fuzzy logic in control systems: fuzzy logic controller—part I and II. *IEEE Trans. Systems, Man and Cybernetics*, 20(2):404–418, 419–435, March/April 1990.

[16] P. Lesoille. Localisation par balises optiques. Technical Report, Ecole Nationale Supérieure des Techniques Avancées, Paris (FR), 1993.

[17] E. H. Mamdani. Applications of fuzzy algorithms for simple dynamic plant. *Proc. of the IEE*, 121(12):1585–1588, 1974.

[18] H. P. Moravec. The stanford cart and the CMU rover. *Proc. of the IEEE*, 71(7):872–884, 1983.

[19] N. J. Nilsson. Shakey the robot. Technical note 323, AI Center, SRI International, Menlo Park, CA (US), April 1984.

[20] C. Novales. *Pilotage par actions réflexes et navigation locale de robots mobiles rapides*. Thèse de doctorat, Université de Montpellier (FR), October 1994.

[21] C. Novales, D. Pallard, and C. Laugier. Controlling the motions of an autonomous vehicle using a local navigator. In *Proc. of the Int. Symp. on Robotics and Manufacturing*, Montpellier (FR), May 1996.

[22] C. Novales and R. Zapata. A local architecture for controlling the movements of fast mobile robots. In *Proc. of the Int. Symp. on Robotics and Manufacturing*, Hawaii, HI (US), 1994.

[23] M. Parent and P. Daviet. Automated urban vehicles: towards a dual mode PRT (Personnal Rapid Transit). In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, volume 4, pages 3129–3134, Minneapolis, MN (US), April 1996.

[24] M. Parent and P.-Y. Texier. A public transport system based on light electric cars. In *Proc. of the Int. Conf. on Automated People Movers*, Irving, TX (US), March 1993.

[25] 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.

[26] D. W. Payton. An architecture for refexive autonomous vehicle control. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pages 1838–1845, San Franciso, CA (US), 1986.

[27] F. G. Pin and H. Watanabe. Steps toward sensor-based navigation in outdoor environments using a fuzzy behaviorist approach. *Journal of Intelligent and Fuzzy Systems*, 1:95–107, 1993.

[28] F. G. Pin and H. Watanabe. Using fuzzy behaviors for the outdoor navigation of a car with low-resolution sensors. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pages 548–553, Atlanta, GA (US), May 1993.

[29] F. G. Pin, H. Watanabe, J. Symon, and R. S. Pattay. Autonomous navigation of a mobile robot using custom-designed qualitative reasoning VLSI chips and boards. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, volume 1, pages 123–128, Nice (FR), May 1992.

[30] A. Saffiotti. The uses of fuzzy logic in autonomous robot navigation: a catalogue raisonné. *Soft Computing*, 1(4):180–197, 1997.

[31] 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*, volume 2, pages 997–1003, Grenoble (FR), September 1997.

[32] R. G. Simmons. Structured control for autonomous robots. *IEEE Trans. Robotics and Automation*, 10(1):34–43, 1994.

[33] D. Simon, B. Espiau, K. Castillo, and K. Kapellos. Computer-aided design of a generic robot controller handling reactivity and real-time control issues. *IEEE Trans. Control Systems Technology*, 1(4):213–229, December 1993.

[34] M. Sugeno. An introductory survey of fuzzy control. *Information Science*, 36:59–83, 1985.

[35] M Sugeno, M. F. Griffin, and A. Bastian. Fuzzy hierarchical control of an unmanned helicopter. In *Proc. of Int. Fuzzy System Association Conference (IFSA)*, pages 179–182, Seoul (KR), 1993.

[36] M. Sugeno and K. Murakami. An experimental study on fuzzy parking control using a model car. In M. Sugeno, editor, *Industrial applications of fuzzy control*, pages 125–138. Elsevier Science Publishers, 1985.

[37] M. Sugeno and M. Nishida. Fuzzy control of model car. *Fuzzy Sets and Systems*, 16:103–113, 1985.

[38] E. Tunstel, H. Danny, T. Lippincott, and M. Jamshidi. Adaptive fuzzy behavior hierarchy for autonomous navigation. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, volume 1, pages 829–834, Albuquerque, NM (US), April 1997.

[39] A. M. Waxman, J. Le Moigne, and B. Srinivasan. Visual navigation of roadways. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pages 862–867, Saint Louis, MI (US), 1985.

[40] S. Yasunobu, S. Miyamoto, and H. Ihara. Fuzzy control for automatic train operation system. In *Proc. of the Int. Conf. on Control in Transportation Systems*, pages 33–39, Baden-Baden (DE), April 1983.

[41] L. A. Zadeh. Fuzzy sets. *Information and Control*, 12:338–353, 1965.

[42] R. Zapata, B. Jouvencel, and P. Lepinay. Sensor-based motion control for fast mobile robots. In *Proc. of the IEEE Int. Workshop on Intelligent Motion Control*, Istambul (TR), August 1990.

[43] R. Zapata, M. Perrier, P. Lepinay, P. Thompson, and B. Jouvencel. Fast mobile robots in ill-structured environments. In *Proc. of the IEEE-RSJ Int. Conf. on Intelligent Robots and Systems*, Osaka (JP), November 1991.

[44] K. Zeghal and J. Ferber. CRAASH : A Coordinated Collision Avoidance System. In *European Simulation Multiconference*, Lyon (FR), June 1993.