# Maneuver-Based Trajectory Planning for Highly Autonomous Vehicles on Real Road With Traffic and Driver Interaction

Sébastien Glaser, Benoit Vanholme, Saïd Mammar, *Member, IEEE*, Dominique Gruyer, and Lydie Nouvelière

***Abstract*—This paper presents the design and first test on a simulator of a vehicle trajectory-planning algorithm that adapts to traffic on a lane-structured infrastructure such as highways. The proposed algorithm is designed to run on a fail-safe embedded environment with low computational power, such as an engine control unit, to be implementable in commercial vehicles of the near future. The target platform has a clock frequency of less than 150 MHz, 150 kB RAM of memory, and a 3-MB program memory. The trajectory planning is performed by a two-step algorithm. The first step defines the feasible maneuvers with respect to the environment, aiming at minimizing the risk of a collision. The output of this step is a target group of maneuvers in the longitudinal direction (accelerating or decelerating), in the lateral direction (changing lanes), and in the combination of both directions. The second step is a more detailed evaluation of several possible trajectories within these maneuvers. The trajectories are optimized to additional performance indicators such as travel time, traffic rules, consumption, and comfort. The output of this module is a trajectory in the vehicle frame that represents the recommended vehicle state (position, heading, speed, and acceleration) for the following seconds.**

***Index Terms*—Advanced driving-assistance systems (ADAS), autonomous intervention and control, autonomous vehicles, decision system, human–machine interface (HMI), trajectory planning.**

## I. Introduction

EVEN with the introduction of advanced driving-assistance systems (ADAS), the number of fatalities remains high. Most of these fatalities result from driver errors, such as slow road departures caused by driver drowsiness or inattention, fast road departures, low driver experience, or excessive speed. In existing ADAS, the driver remains solely responsible for the driving task. With the development of new technologies, it becomes possible to provide a more intrusive driving assistance, for instance, a longitudinal control with advanced cruise control (ACC). The demonstrations on Californian highways at the end of the 1990s proved that autonomous driving in a secured environment (a dedicated lane on a highway with magnets) is possible. Vehicles involved in the Defense Advanced Research Projects Agency Challenge integrate a trajectory planner in a more complex environment and at higher speeds, but the sensors and the computation power needed are not realistic with respect to the price and energy limitations of an everyday vehicle.

A lot of research is currently dedicated to the interaction between automation and driver, as the psychological and legal aspects of advanced driving assistance become more important. The design of the human–machine interface (HMI) is an essential part in these studies. The Adaptive Integrated Driver-vehicle InterfacE European project [7] and Network of Excellence Humanist [9] aim to develop a human centered generic interface for ADAS; however, they do not integrate new type of ADAS. PReVENTive and Active Safety Applications (PReVENT) [8] and SAFESPOT [30] projects have developed new driver assistances and driver interfaces. Moreover, PReVENT driver assistance has a strong intrusion in the driving task: The assistance may take a part of the driving task, i.e., braking for the driver for instance. The recent European project Secure Propulsion using Advanced Redundant Control [5] provides a safe way to interact with driver both for longitudinal and lateral control. Using these developments, it is possible to offer the driver new concepts of driving assistance. The successor of this project, Highly Automated Vehicles for Intelligent Transport (HAVEit) [6], focuses on the cooperation between the driver and a copilot system at the maneuver decision level. It defines different automation modes, ranging from fully human to highly automated. It also proposes a methodology for selecting the best mode at each moment. This is the context in which the present system is developed.

The objective of this system is to sense the environment, monitor the driver actions, and compare a set of safe trajectories computed by the automation with the trajectory followed by the driver. When the difference between the safe and the realized trajectories becomes very high, the system can decide to intervene in the driving task.

Many of the algorithms in today's literature propose to optimize the trajectory generation for avoiding collisions and respecting the vehicle dynamics. In this paper, we would like to integrate other aspects of good driving. We focus on generating longitudinal–lateral trajectories that optimize the safety, speed, driver comfort, fuel consumption, and traffic rules.

Manuscript received December 15, 2008; revised October 29, 2009 and February 15, 2010; accepted February 25, 2010. Date of publication May 10, 2010; date of current version September 3, 2010. This work was supported in part by the HAVEit project and in part by the European Commission. The Associate Editor for this paper was M. Da Lio.

S. Glaser, B. Vanholme, and D. Gruyer are with the Laboratory on Interactions between Vehicles, Infrastructure and Drivers (LIVIC), French National Institute on Transportation Research and Safety (INRETS)/Central Laboratory on Civil Engineering (LCPC), 78000 Versailles, France (e-mail: vanholme@lcpc.fr; glaser@lcpc.fr; gruyer@inrets.fr).

S. Mammar and L. Nouvelière are with the Informatique, Biologie Intégrtive et Systèmes Complexes (IBISC), Equipe Associée 4526 Centre National de la Recherche Scientifique (CNRS) 3190, Université d'Évry val d'Essonne, 91020 Evry Cedex, France (e-mail: said.mammar@iup.univ-evry.fr).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TITS.2010.2046037

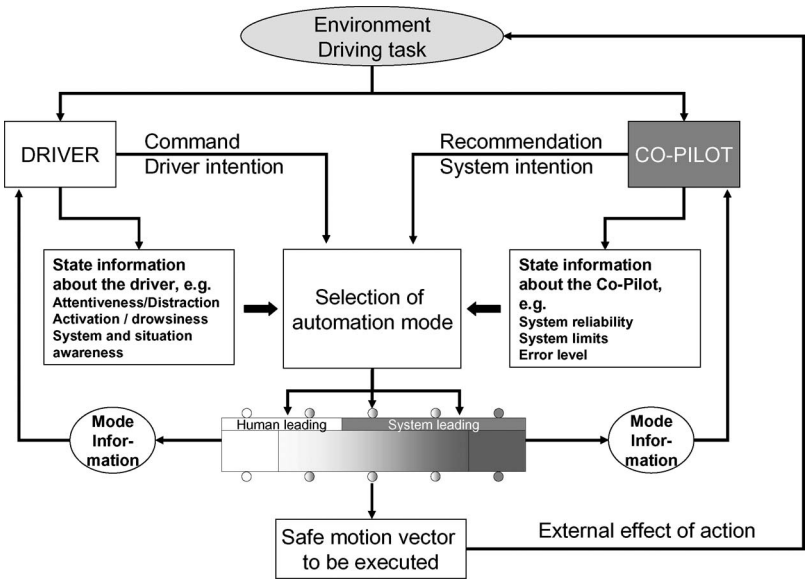


Fig. 1. Driver and algorithm comparison (HAVEit project).

The trajectory-planning algorithm must comply with two strong specifications. First, for a strong interaction with the driver, it has to be understandable in its decisions and respect the driver's wish. Second, the algorithm must run on an embedded microprocessor that is currently available in the vehicle industry and provided in the frame of the project. The target platform is fail safe and compliant with the automotive standard AUTomotive Open System Architecture to assure the scalability of its applications. It has a clock frequency of less than 150 MHz, 150 kB of RAM memory, and 3 MB of program memory. In this context, the computation of the trajectory-planning algorithm must be done within 100 ms.

The remainder of this paper is organized as follows. Section II presents the integration of the work within the HAVEit project. In Section III a short overview of the existing trajectory planning algorithms is given, with a short indication of their aptness to our target application. Section IV gives an overview of the global architecture of the proposed system. Section V contains the description of the high-level part of the algorithm, with the risk analysis of the maneuvers and the development of a mechanism to ensure its stability over the time. Section VI describes the low-level part with a method to generate and evaluate trajectories and to output the optimal ones. Section VII discusses how the computed maneuvers are compared with driver's maneuvers. Section VIII describes the controller used to follow the trajectory both in the longitudinal and in the lateral direction. Section IX shows some results on a simulation environment, and Section X gives the conclusion and a view on the future works.

## II. Integration of the Work Within the HAVEit Project

HAVEit is a research project of the seventh framework program of the European Commission, which cofunds this work. It aims at giving the long-term vision of a highly automated driving for an intelligent transport. The project will develop, validate, and demonstrate important intermediate steps toward highly automated driving. HAVEit will significantly contribute to higher traffic safety and efficiency usage for passenger vehicles, buses, and trucks, thereby strongly promoting safe and intelligent mobility of both people and goods. HAVEit will generate a significant impact on safety, efficiency, and comfort by the following three measures:

1) the design of the task sharing between the driver and codriving system;
2) the presentation of a failure-tolerant safe vehicle architecture including an advanced redundancy management;
3) the development and validation of the next generation of ADAS directed toward a higher level of automation as compared with the current state of the art.

The work presented in this paper touches these three points. It is integrated in the copilot module of the cosystem depicted in Fig. 1. The subproject in HAVEit that builds the cosystem is led by the German Aerospace Center (DLR) and has, as partners, the National Institute for Research in Computer Science and Control (INRIA), the Institute of Communications and Computer Systems (ICCS), the Wuerzburg Institute for Traffic Sciences (WIVW), Continental, and Ibeo. The cosystem analyzes the surrounding sensed environment, defines an optimal maneuver and an associated optimal trajectory, and, finally, controls the different actuators according to the state of the block labeled *Selection of Automation Mode*.

## III. Comparison With the State of the Art

A lot of research has been done on path planning, originally for robotic applications and, more recently, for vehicles. In this section, we give a short overview of the existing approaches and indicate their advantages and disadvantages for our target application. As main criteria, their optimality (the ability to find the best trajectories in the solution space) and their computational complexity (the calculation time and memory consumption) will be commented on.

### A. Potential Fields

Potential field algorithms attribute repulsive forces to obstacles in the environment and attractive forces to the target position of the vehicle and finds the trajectory along the steepest gradient of the resulting potential field. This group includes artificial potential fields [36], [37], vector field histograms [38], and elastic band algorithms [39]–[41].

In general, potential fields provide a robust real-time trajectory planning for vehicles, as [41] shows for applications in a city environment. However, optimality problems arise when the trajectory is trapped in a local minimum of the potential field. In addition, the vehicle kinematic constraints and environment dynamics are difficult to handle.

### B. Cell Decompositions

Cell decomposition algorithms divide the environment in various shaped regions known as cells. Optimal search methods, such as the $A^*$ and $D^*$ algorithms [31], are used to find a trajectory that connects an initial position to a destination while avoiding obstacles in the environment. Rapidly exploring random tree (RRT) algorithms are now widely used to improve the computational efficiency of this technique. Extensions of the basic RRT algorithm have been presented to adapt to the particular (nonholonomic and actuator) constraints of vehicle systems [32]–[34] and to deal with environment uncertainty [35].

The optimality of the solutions found is only satisfactory if the resolution of the grid cells is high at a high computational memory and time cost. In the highly dynamic environment of our application, fast random sampling methods such as the RRT can deliver real-time trajectories on powerful computers [4]. They cannot run on the simple safety-critical microprocessors that will be available on the vehicles in the near future, which will be used for this development.

### C. Optimal Control

Optimal control algorithms directly incorporate the cost function and constraints that are specific to the application, which assures the optimality of its solutions. They can also incorporate the vehicle control in the planning algorithm.

The high computational complexity and possible convergence issues of trajectory planning can now be handled by a numeral solver on a high-standard processor but not yet on the basic fail-safe architectures on today's vehicles. Some of the works use this technique to solve a part of the problem, such as [42] (for the longitudinal movement, without the presence of objects), [43] (for the vehicle control part), or [44] (for the geometry of an obstacle avoidance trajectory), without integrating the complete decisional part.

### D. Proposal for Combining a Maneuver Algorithm With a Total Trajectory Exploration Algorithm

Our approach consists of designing a trajectory planning of two levels. At a high level, nine maneuver cases are identified by combining basic actions: three in the longitudinal direction (staying in the same speed range, decelerating, or accelerating) and three in lateral direction (staying in the same lane, changing lanes to the left, or changing lanes to the right). A safe state maneuver, which corresponds to stopping on the right-most lane, and an emergency maneuver, which corresponds to maximal braking until standstill, can be added to these nine maneuvers. This level gives a first evaluation of the situation with respect to the most important criterion: the collision avoidance. The computation time of this step is very low, and it uses a limited description of the environment, mainly the relative distance to and speed of the obstacles. The output is a ranking of the possible maneuvers according to their corresponding collision risk.

At a low level, we propose a total trajectory exploration algorithm that evaluates a large set of trajectories with predefined parameters within each of the previously accepted maneuvers. This second stage inherits from composition cells or tree-based methods but directly builds and evaluates a complete trajectory, instead of constructing it as a tree step by step. This level is also inspired by the cost concept of the optimal control methods. It aims at optimizing the trajectory definition with respect to a more precise definition of risk and to other important criteria for the driver: speed, comfort, energy consumption, and traffic rules. The search space can be refined by the evaluating the fusion of pairs of suboptimal trajectories. The output of the module is a trajectory that describes the optimal longitudinal–lateral positions and speed of the vehicle for the following seconds.

## IV. Architecture

In this section, the architecture of the copilot is presented. The goal of the proposed system is twofold: first, the system allows a high driver interaction, and second, it is able to optimize a trajectory according to different performance criteria in an everyday traffic environment. This multicriteria optimal trajectory assures a high level of safety and can be used to control the vehicle. However, in order to meet the safety level required on the vehicle, the system allows a high cooperation with the driver. The architecture presented is different from the ones commonly used in robotics. This is needed to tackle the following challenges.

1) The environment is highly dynamic, even in a specific environment such as highways.
2) The computation time and memory are limited when working with a fail-safe and embedded architecture.
3) The system must be able to closely interact with the driver, both by receiving commands coming from the driver and by giving the driver a clear feedback.
4) The system has to take into account the real road environment constraints, such as lanes, which drastically limit the solution space of trajectories.

### A. Functional Architecture

The automation and the driver follow the same process (see Fig. 2). They can interact at different levels of this process, e.g., for the perception of the environment, the automation can help

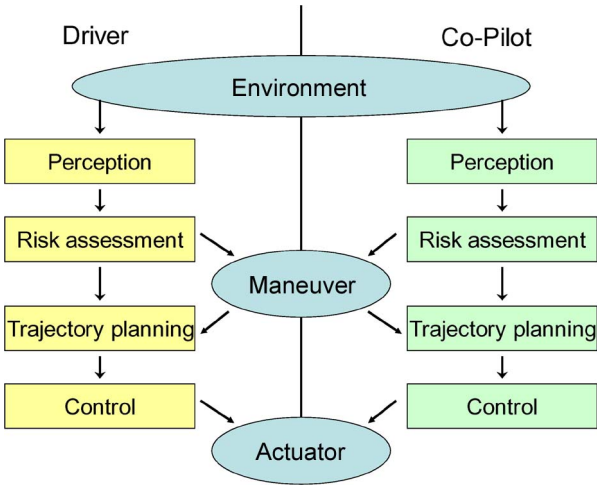


Fig. 2. Driver and automate process.

with the detection of a front vehicle in foggy weather. We can define the following two main ways of interaction:

1) definition of the maneuver, which is the high-level verbalization of the longitudinal and lateral targets;
2) execution of the trajectory through the actuators.

The automation can act on the two levels. According to Fig. 3, we define the following assistance modes:

*1) Proposition of the maneuver*: The driving assistant analyzes the environment and presents a set of possible maneuvers that enhance the safety and other performance criteria of the vehicle. The final decision and action is left to the driver (left figure).

*2) Shared control*: both driver and automation define an optimal maneuver. The execution of the corresponding trajectory is done by the control module of the automation (right figure).

The decision of the automation must be easily understandable by the driver. It is represented as a grid of possible maneuvers, as shown in Fig. 4. Maneuvers are defined relatively to the actual state of the ego vehicle. With the green maneuvers the multicriteria driving performance improves. In the yellow sections, the situation stays the same, and in the red ones, it worsens. The grid can easily be compared with the driver intentions by monitoring the state of the indicators and accelerator and braking pedals, as will be explained in Section VII.

### B. Software Architecture

For meeting the goals of the system, we have developed a two-step computation procedure for the trajectory planning (see Fig. 5). In a first step, a high-level evaluation of the future actions is given, in terms of maneuvers. Afterward, the trajectory evaluation module finds the best trajectory within the accepted (first the green and then the yellow) maneuvers. It gives a detailed description of the recommended vehicle speed and position for a time frame of 5–10 s.

The next sections give more details on the two steps of the algorithm.

*1) Maneuver Module:* This first module aims at providing a clear interface to the driver and at drastically reducing the computation time of the second module, i.e., the trajectory module. The module analyzes the state of the ego vehicle and of the environment and attributes a risk value to each maneuver. Only the collision risk is evaluated here; a finer evaluation of the risk is done by the second module. A ranked list of maneuvers is the output of this module.

*2) Trajectory Module:* The trajectory module reads the environment information and the results of the maneuver module. It generates trajectories within the best maneuvers and evaluates them according to multiple performance indicators. A finer definition of the risk is given, including the slipping risk in curves. In addition, the module evaluates the trajectory with respect to its speed, the fuel consumption, the comfort, and the traffic rules. The search process of trajectories is done in the order of the ranking of the maneuvers and can be stopped at any time to respect a prefixed computation time. All trajectories respect the longitudinal and lateral acceleration constraints to ensure their controllability. The final output is a set of spatiotemporal points with the recommended position, heading, speed, and acceleration in the vehicle frame. It can be used in order to control the vehicle or to be provided as information for the driver.

## V. Maneuver Generation

The objective of this module is to provide a high-level interaction with the driver. It also gives a fast ranking of the different zones of the solution space, drastically increasing the efficiency of the trajectory module and, thus, ensuring real-time execution. Nine maneuver cases are defined by combining three actions in the longitudinal direction (staying in the same speed range, decelerating, or accelerating) and three in the lateral direction (staying in the same lane, changing lanes to the left, or changing lanes to the right). For safety reasons, a safe state maneuver, which corresponds to stopping on the right-most lane, and an emergency maneuver, which corresponds to maximal braking until standstill, can be added to these nine maneuvers.

The ranking of the maneuvers is based on a fast risk evaluation on every lane. The risk is related to the speed and the relative distance to each vehicle in each lane. For the adjacent lanes, the ego vehicle is replaced by a virtual vehicle located at the same curvilinear position on the evaluated lane. By doing this, we do not consider the risk generated during the lane change; this is done at the trajectory level. The risk is computed for a wide ego vehicle speed range and is not defined as an absolute value but relative to the current risk state.

A common approach in risk theory is to define the risk related to an event by using the following two criteria:

1) the probability that the event occurs;
2) the gravity of the resulting situation under the assumption that the event occurs.

In our maneuver selection, the event to avoid is a collision. We determine the gravity of the possible collision, using the equivalent energetic speed (EES). A true calculation of probability of a collision is hard to achieve and may depend on several external parameters that are not available or not even measurable. Instead, we will define the possibility of a collision that is deduced from the analysis of traffic indicators. The

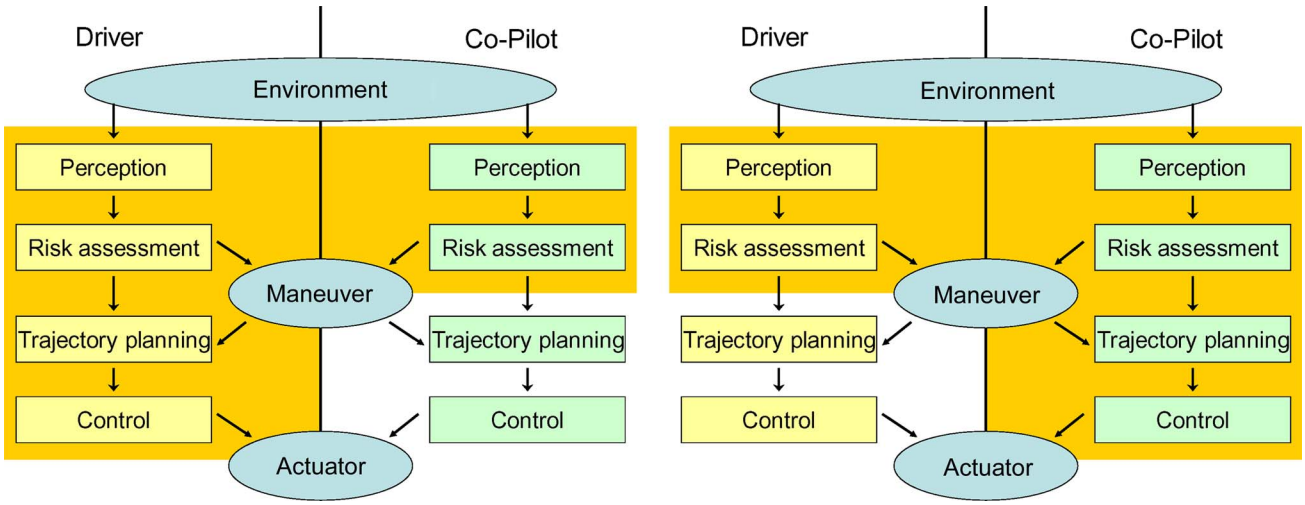


Fig. 3. Proposed driving assistant.

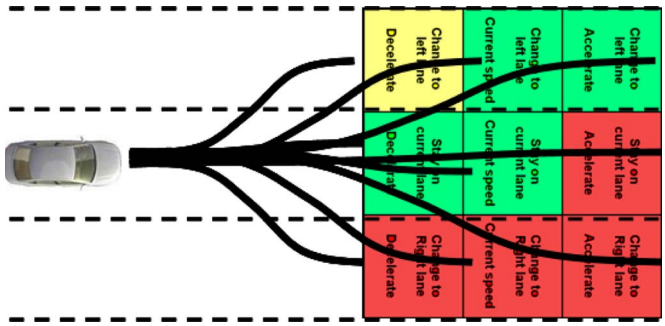

Fig. 4. Grid of maneuver.

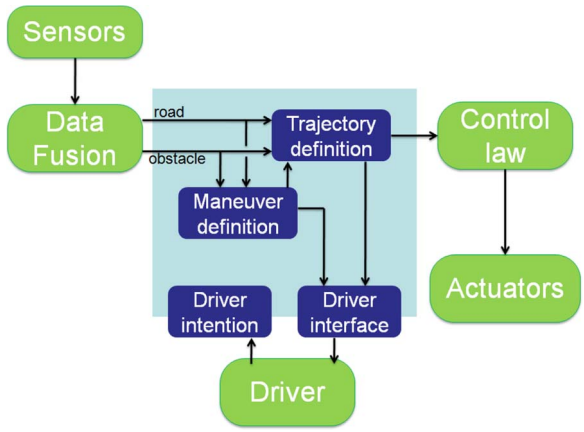


Fig. 5. General architecture.

possibility has the same limits as the probability: At 0, the collision possibility is not relevant; at 1, it is highly possible that a collision occurs. The possibility is deduced from the analysis of indicators as the intervehicular time, the time to collision (TTC), and the reaction distance.

For each maneuver, the average risk of the corresponding speed range and lane is calculated, as shown in Fig. 4. Then, it is compared with the current maneuver to determine whether it is less or more risky or about the same.

The next sections will define the two components of risk assessment.

### A. Crash Severity

The severity of a collision has been extensively studied and often uses the EES during the collision. The EES corresponds to the deformation energy of a damaged vehicle during a collision, given their respective speed and mass. This is directly linked to the damage done to the humans occupying the vehicle. The speed can be computed using the following equations:

$$\begin{cases} MV + M_i V_i = M\hat{V} + M_i \hat{V}_i \\ \frac{1}{2}MV^2 + \frac{1}{2}M_i V_i^2 = \frac{1}{2}M\hat{V}^2 + \frac{1}{2}M_i \hat{V}_i^2. \end{cases} \tag{1}$$

In these equations, the variables $X$ and $X_i$ are related to the ego vehicle and to the considered obstacle, respectively, before the collision. $\hat{X}$ and $\hat{X}_i$ represent the respective variables after the collision. The EES of the vehicle is then

$$\text{EES} = \hat{V} - V = \frac{2M_i}{M + M_i}(V_i - V). \tag{2}$$

Using data on EES and probability of injuries, we can define a scale of severity relative to the probability of a light injury, a heavy injury, or a fatality. Fig. 6 represents the likelihood of a moderate injury (maximum abbreviated injury scale (MAIS) $> 2$) with respect to the EES.

### B. Possibility of Collision

In order to define the possibility of a collision, we have to analyze the traffic conditions around the vehicle. Traffic indicators often use a time value to describe the situation, which is easily understandable by the driver. However, it is hard to achieve an accurate analysis of the risk related to the traffic with only one indicator.

The following two main types of danger exist when considering the lane with traffic: 1) low intervehicle distance and 2) high speed differences. The traffic indicators show that the first type is easily detected by the intervehicular time (TIV), while the second can be estimated using the TTC. In order to define a possibility of collision, these two criteria are used.

*1) TTC:* The first parameter is relative to the TTC. Hayward [2] defines the TTC as follows: *the time required for two*

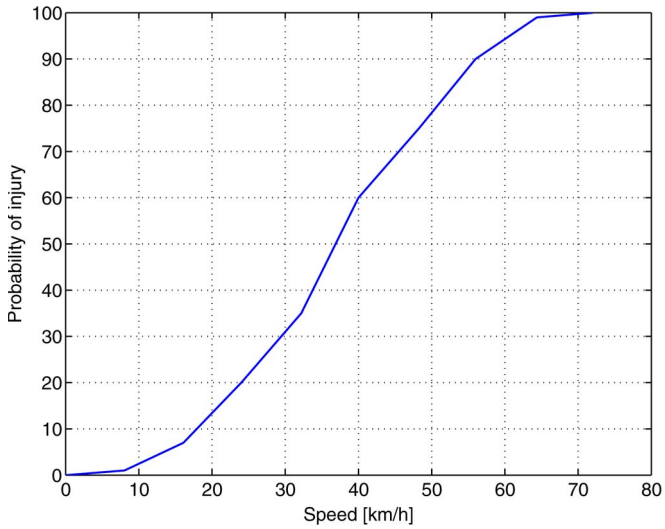


Fig. 6. EES (speed) and MAIS scale (probability) for a moderate injury.

*vehicles to collide if they continue at their present speed and on the same trajectory*. The TTC formula is

$$\mathrm{TTC} = \frac{D_i}{V - V_i} \tag{3}$$

where $D_i$ is the relative distance with vehicle $i$. Projects such as ARCOS (www.arcos2004.fr) and PREVENT (www.prevent-ip.org) have studied TTC and define several boundaries for the cooperation between the driver and automation.

1) At a TTC of 10 s, vehicle $i$ is supposed to have no interaction.
2) A TTC of 1.5 s is commonly used to trigger a first level of warning.
3) When TTC goes below 1.3 s, the system can strengthen the warning.
4) If TTC becomes lower than 1 s, the control by the automation can be activated.

Looking at the system as collision mitigation by braking [29], a collision is highly possible as soon as the TTC goes below 1 s; this is the threshold used by some collision-mitigation systems. A value of 1 s is also common as a driver reaction time.

The two extreme time values are used to determine a possibility of collision of 0 (for a TTC of 10 s and higher) and 1 (for a TTC below 1 s). Between these values, the possibility is linear with respect to TTC. Fig. 7 (top) represents the evaluation of the possibility associated with the TTC, namely, $P_{\mathrm{TTC}}$.

*2) Intervehicular Time:* However, the TTC itself is not sufficient to describe the risk related to the situation: for instance, when two vehicles are close to each other, with the same speed, the TTC is large, but the situation can be dangerous. In order to take into account this kind of situations, we enhance this definition of the risk possibility with the intervehicle time. The TIV is defined as

$$\mathrm{TIV} = \frac{D_i}{V}. \tag{4}$$

This parameter is often used in regulations. For instance, a recent French law demands a minimal TIV of 2 s. This value is commonly used for traffic safety, allowing the driver to analyze the actions of the followed vehicle. It is also frequently used in ACC systems to regulate the speed of the vehicle. Similarly to the TTC, we define a relation between the TIV and the possibility of collision, as shown in Fig. 7 (bottom). The possibility of collision associated with the TIV is now denoted as $P_{\mathrm{TIV}}$. The 1 s boundary is taken for the same reason as explained for the TTC.

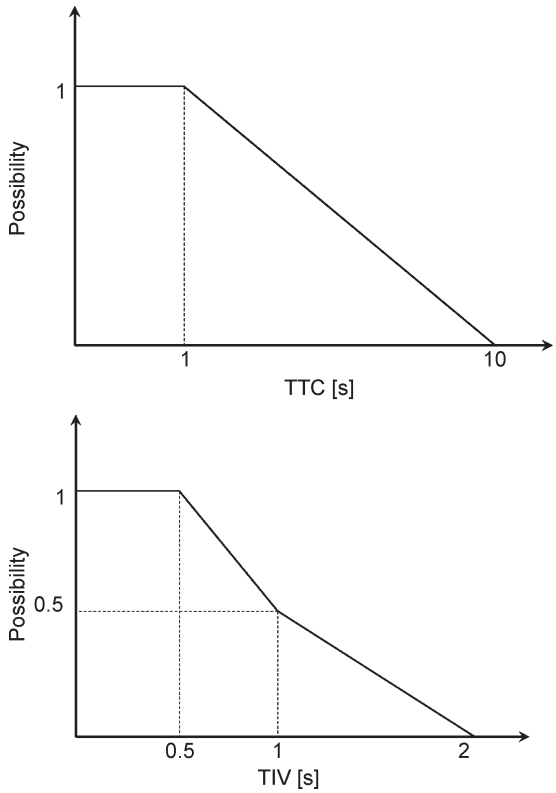


Fig. 7. (Top) TTC- and (bottom) TIV-based possibility of collision.

### C. Generating a Risk

In the two previous sections, we have described the computation of the gravity index and of two possibilities. However, these possibilities do not represent the same event: The first one mainly deals with fast approaching vehicles and the second with close vehicles. As the scenarios are not the same, the gravities in case of occurrence of the event is not the same. The final risk is expressed by

$$R = R_{\mathrm{TTC}} + R_{\mathrm{TIV}} \tag{5}$$

where $R_{\mathrm{TTC}}$ is the risk associated with the TTC, and $R_{\mathrm{TIV}}$ is the risk associated with the TIV. For each considered target speed $V$ of the ego vehicle and given speed $V_i$ of the obstacle, the risk related with the TTC is computed as follows:

$$R_{\mathrm{TTC}}(V) = P_{\mathrm{TTC}}(V)G(V, V_i) \tag{6}$$

where $G$ denotes the function that evaluates the gravity as defined previously.

The risk related to the TIV represents the problem of a hard braking vehicle in front of the ego vehicle. In order to translate

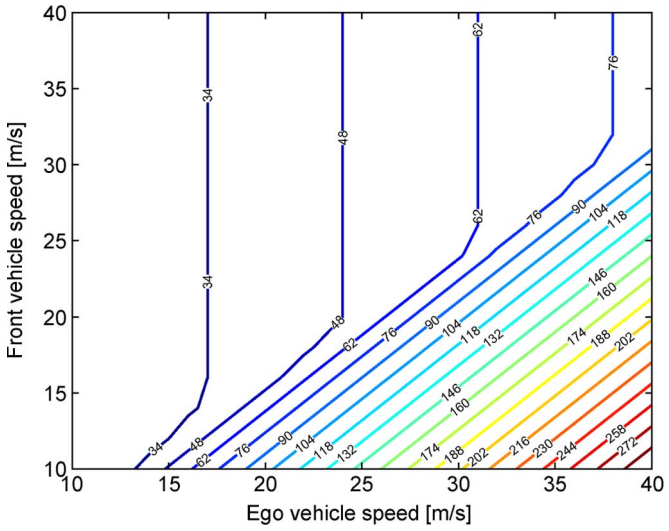


Fig. 8. Distance that ensures a zero risk.

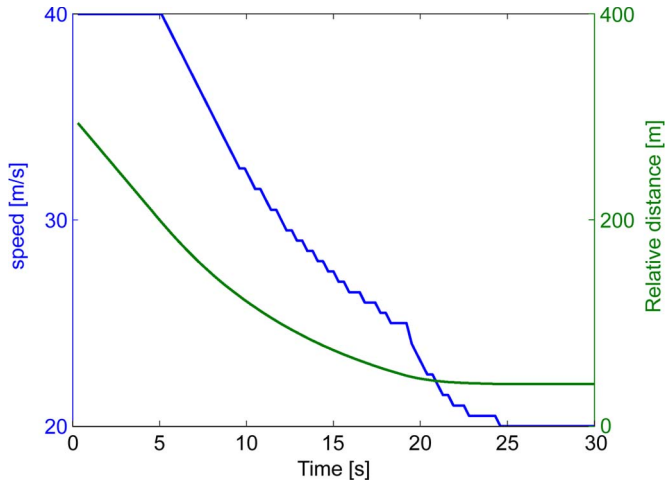


Fig. 9. Fast-vehicle-approaching scenario.

this problem in terms of risk, we use the following equation, with the variable described previously:

$$R_{\mathrm{TIV}}(V) = P_{\mathrm{TIV}}(V) \max\left(G(V, V_i), G(V, V_i - \gamma T_{\mathrm{TIV}})\right) \tag{7}$$

where $\gamma$ is the considered deceleration set to 0.8 g.

### D. Example

Previous developments explain the maneuver selection. In order to demonstrate the risk evaluation, we suppose that the vehicle exactly follows the maneuver advice given by the algorithm.

As a first remark when dealing with vehicle-following scenario, Fig. 7 and (3) and (4) clearly show that the ego vehicle speed will converge to the speed of the front vehicle with a relative distance of 2 s. Fig. 8 shows the minimal distance according to the ego vehicle speed and front vehicle speed that ensures a zero risk for the ego vehicle. For a front vehicle speed of 35 m/s, the speed is mainly regulated by the intervehicular distance factor. However, when this speed drops to 15 m/s, in a traffic jam for instance, the TTC quickly becomes the main factor.

An interesting simulation is to evaluate the approach on a slow moving vehicle. In order to evaluate this scenario, we suppose that our vehicle drives at 40 m/s (144 km/h) and the front vehicle is at 300 m with a speed of 20 m/s (72 km/h). The controlled vehicle will drive at the maximal safe speed. Fig. 9 shows the resulting speed. At the beginning, the front vehicle is too far to have an impact on our vehicle; therefore, the speed remains constant. After $T = 6$ s, the front vehicle starts to have an impact on the risk evaluation of our vehicle, corresponding to a decrease in speed. After $T = 19$ s, the intervehicular distance regulates the speed, and therefore, a stronger decrease appears. The simulation stops at $T = 25$ s when both vehicles have the same speed with a safe intervehicular distance.

During the simulation, the deceleration of the controlled vehicle remains below 0.3 g. The steps in speed profile result from the direct association of computed speed with the realized speed and not using a controller on the actuators.

To summarize the maneuver algorithm, we can highlight that the objective is to give a ranking to several possible maneuvers. Additionally, the target speed and target lane of the best maneuver can be given. The maneuver level will allow a close cooperation with the driver. Section VII will explain the maneuver grid computed by the automation and enriched by the trajectory module here can be compared with the intention of the human driver.

## VI. Trajectory Generation and Evaluation

The trajectory module finds the optimal trajectory within the accepted maneuvers (green and yellow on Fig. 4). It gives a detailed description of it for a time frame of 10 s. The fast ranking of the different zones of the solution space by the maneuver module greatly increases the efficiency of the trajectory algorithm and allows to stop it when a certain prefixed calculation time is reached. If the computation is very limited, it could only search in the green cells or even only the best green cells.

The trajectory module uses a finer evaluation of risk and additional performance indicators such as speed, legal driving, comfort, and consumption. The module also refines the solution space by investigating multiple trajectories within each maneuver cell. The core idea behind the trajectory algorithm is that for a complex problem with a multitude of good (suboptimal) solutions, it is wiser and faster to evaluate a small set of well-chosen solutions, rather than searching the best in the complete solution space.

The process can be described in three steps:

1) prediction of the evolution of the environment: giving the future positions in time for every non–ego vehicle or obstacle in the environment;
2) first generation and evaluation of trajectories in the maneuver grid: ranking the trajectories with a multicriteria performance indicator;
3) second generation and evaluation of trajectories by fusing the best ones: refine the solution space and output the (sub)optimal trajectory.

Each of these steps will now be described. The algorithm will be explained through the example shown in the first section of

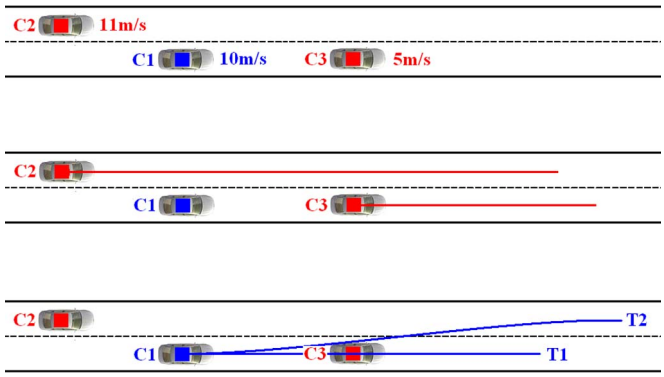


Fig. 10. Scenario for the algorithm example: overtaking of a slower vehicle by the ego vehicle.

Fig. 10: The ego vehicle $C_1$ is driving at 10 m/s on the right lane of a two-lane road. At 20 m behind it on the left lane, a vehicle $C_2$ is detected, which is driving slightly faster, at 11 m/s. At 20 m in front of the ego vehicle, another vehicle $C_3$ is driving slowly at 5 m/s. This situation can be dangerous. We will see in the trajectory generation and evaluation how the system will react.

This example speaks for the full range of the application, from zero to maximum highway speeds, as the number of calculations does not change with the ego vehicle speed. A fixed number of trajectories are calculated, and they describe a fixed period of time, as will be shown.

### A. *Prediction of the Evolution of the Environment*

The quality of the environment perception and the horizon of sight of the sensors are important information for the trajectory module. Data fusion will not only deliver information on the detected objects, but will also indicate the limits of the zone of detection. The trajectory module will create phantom vehicles outside this zone in order to always be prepared for a worst-case (legal) scenario.

For example, in a highway environment, a traffic jam is believed to exist at the detection horizon of the sensors in the front of the vehicle. This means that a large phantom object with zero speed is created there. A phantom object at high speed is created at the perception limit of the rear sensors. On a two-direction road, the constraints are stronger because at each moment, a phantom vehicle at high speed will be believed to come from the other direction. This is why a fully automated lane changes in such an environment are difficult with today's sensors.

The trajectory module deals with the phantom objects as with the detected objects to make sure to drive slow enough to be able to stop for a traffic jam and fast enough not to put fast vehicles in the back at risk.

At each moment, the first step is to predict the trajectory of every detected object using a Kalman filter. Object clustering allows us to limit the number of relevant objects to eight, surrounding the vehicle: in the front, in the rear on the current and adjacent lanes, and on the side in adjacent lanes. This guarantees a limited calculation time. A 5–10-s time description of the position and speed of each object in the environment is then available. It will be used for the risk evaluation of the trajectories of the ego vehicle $C_1$.

We need to know the uncertainty on the prediction according to the initial covariance on the data from sensors and data-fusion module. In order to predict the evolution of environment, we use the predictive phase of a Kalman filter applied on each object, i.e.,

$$\begin{cases} X_{k+1} = AX_k + BU_0 \\ P_{k+1} = AP_kA^t + Q \end{cases} \tag{8}$$

where $X_k$ is the predicted state, in which $X_0$, which is the initial state, is given by the sensors and data-fusion module, $U_0$ is the input of the system at the initial step, $Q$ is the process noise matrix, and $P_k$ is the covariance on the system, in which for the state, $P_0$ is given by the sensors and data-fusion module. We separate the longitudinal and lateral behavior of each object with respect to the road.

*1) Longitudinal Behavior:* In our application, the longitudinal behavior of a near vehicle can be deducted directly from the Kalman observation. In (8), the state vector is $X_k = [s, v]^t$, where $s$ is the curvilinear abscissa, $v$ is the vehicle speed, and the input $U_0$ is the acceleration, which is constant. The matrices $A$, $B$, and $Q$ are defined as follows:

$$A = \begin{pmatrix} 1 & \Delta t \\ 0 & 1 \end{pmatrix}, \quad B = \begin{pmatrix} 0 \\ \Delta t \end{pmatrix}, \quad Q = \begin{pmatrix} \varepsilon_s & \varepsilon_s v \\ 0 & \varepsilon_v \end{pmatrix} \tag{9}$$

where $\Delta t$ is the time step of the prediction, and $\varepsilon_i$ denotes the error to be defined.

In our example in Fig. 10, the non–ego vehicles $C_2$ and $C_3$ are believed to continue at a constant speed from their current position.

*2) Lateral Behavior:* In the next step, the lateral movement of the non–ego vehicles is predicted. This can be done in a similar way as for the driver maneuver prediction (explained in Section VII) by calculating the probability of the three lateral maneuvers, namely 1) staying in the current lane; 2) changing to the right lane; and 3) changing to the left lane. We combine information on the lateral position in the lane, of the lateral speed in the lane, and on the state of the indicators to find this probability.

A position on the right side of the lane, a speed toward the right or the activation of right blinkers all lead to a high probability—between 0.7 and 1.0—for changing lanes to the right and low probability for staying on the same lane or changing to the left lane. By combining these probabilities, we predict that the detected vehicle will change lanes to the right and we generate a trajectory for this vehicle in a similar way as we will explain for the ego vehicle $C_1$ in Section VI-B.

When information on position, speed, or indicators are conflicting, leading to comparable probabilities for two or three maneuvers, the vehicle is then replaced by two or three phantom objects with corresponding trajectories.

All information on both the vehicles $C_2$ and $C_3$ in our example gives a high probability for staying in the lane. The Kalman prediction of the movement of the these vehicles is shown in the second section of Fig. 10.

### *B. First Generation and Evaluation of Trajectories for Ego Vehicle*

For the ego vehicle, a fixed number of smooth polynomial trajectories are generated, each with a different target speed and target lane within the ranges of the maneuvers chosen by the maneuver module. These trajectories are based on the ego vehicle state and a set of predefined parameters and not on the object or lane state. The performance cost concept, which is explained below, will favor the trajectories that are best suited to the environment.

In simulations, a number of two to five trajectories per maneuver show a good compromise between calculation time and resolution of the solution space.

*1) Generation of Trajectories:* The three initial values (i.e., begin of time description) of position, speed, and acceleration and two target values (at a certain time $T$) of speed and acceleration give five constraints in the longitudinal direction. In the lateral direction, there are six constraints as both current and target values of position, speed, and acceleration are specified. This defines a fourth- and a fifth-order polynomial time description for longitudinal and lateral position, respectively. With the origin of the coordinate axis in the center of gravity of the vehicle, $X$ being the longitudinal axis in the driving direction of the vehicle, and $Y$ being the right-hand side perpendicular to $X$, we write the equations for the longitudinal and lateral positions $p$ with respect to the time $t$, which is the difference with the current time, as follows:

$$\begin{cases} p_x: \ g_{0x} + g_{1x}t + g_{2x}t^2 + g_{3x}t^3 + g_{4x}t^4 \\ p_y: \ g_{0y} + g_{1y}t + g_{2y}t^2 + g_{3y}t^3 + g_{4y}t^4 + g_{5y}t^5. \end{cases} \tag{10}$$

For the calculation of the coefficients of these polynomials using the begin and end constraints, we refer to the work of Simon and Becker [10]. The conversion of these constraints into coefficients for our application is worked out in the Appendix.

The target values on the longitudinal and lateral position, speed, and acceleration of each trajectory evaluated by the algorithm are described in a predefined parameter table, relative to the initial position, speed, and acceleration of the vehicle. Apart from the target state, the time $T$ in which this target state is reached is chosen in this parameter table. This corresponds to an "average longitudinal acceleration" to reach a longitudinal target speed or to an "average lateral speed" to reach a lateral target position.

In the generation step of the algorithm, a large set of trajectories is calculated with the values of the parameter table that correspond to the maneuvers that were accepted by the maneuver module. To this set, an emergency brake trajectory (maximal braking until standstill) and a safe state trajectory (soft braking until emergency lane) are added as worst-case solutions. The multicriteria evaluation step described in the next section will rank the trajectories to their aptness to the situation, e.g., the "comfort" concept will favor the trajectories that take a large time to reach the target, which can be in competition with the "speed" concept or with the "risk" concept, e.g., in "emergency avoidance maneuvers."

In our example, we show the generation of a trajectory for two different maneuvers proposed by Fig. 4. For the maneuver *Stay on current lane and decelerate*, trajectories with target speeds of 2, 4, 6, and 8 m/s will be calculated. We call the trajectory *Stay on current lane with target speed 8 m/s* trajectory $T_1$. For the maneuver *Change to the left lane with current speed*, trajectories with target speeds around the current speed, such as 9, 10, and 11 m/s, are calculated. The trajectory *Change to left lane at 10 m/s* is called trajectory $T_2$.

TABLE I
BEGIN AND END CONSTRAINTS FOR THE GENERATION OF TRAJECTORIES $T_1$ AND $T_2$

| trajectory $T_1$ | | trajectory $T_2$ | |
|---|---|---|---|
| begin | end | begin | end |
| $v_{x01} = 10$ | $v_{xT1} = 8$ | $v_{x02} = 10$ | $v_{xT2} = 10$ |
| $a_{x01} = 0$ | $a_{xT1} = 0$ | $a_{x02} = 0$ | $a_{xT2} = 0$ |
| $p_{y01} = 0$ | $p_{yT1} = 0$ | $p_{y02} = 0$ | $p_{yT2} = 3$ |
| $v_{y01} = 0$ | $v_{yT1} = 0$ | $v_{y02} = 0$ | $v_{yT2} = 0$ |
| $a_{y01} = 0$ | $a_{yT1} = 0$ | $a_{y02} = 0$ | $a_{yT2} = 0$ |

The begin constraints for the polynomial description are the same for both trajectories $T_1$ and $T_2$. The end constraints for the trajectories are set by their target lateral position, their target longitudinal speed, and a zero lateral speed and acceleration and a zero longitudinal acceleration. With $v$ referring to the speed, $a$ to the acceleration, 0 being the begin time, $T$ end time, and the indices 1 and 2 referring to trajectory $T_1$ and $T_2$, respectively, the constraints are defined in Table I.

With these constraints, the polynomial description of the trajectories is found directly, as explained before. The third section of Fig. 10 shows trajectories $T_1$ and $T_2$.

Setting the target and time-to-target values of the polynomial trajectory does not guarantee that it respects the physical constraints on the vehicle. A "feasibility" function checks if the longitudinal acceleration and the curvature along a trajectory are within the vehicle's kinematic and dynamic limits, based on a simple vehicle model. If this is not the case, the trajectory is discarded from the solution space. The emergency brake trajectory is never discarded and is executed as good as possible in the worst-case event that no other safe trajectory is found to be physically feasible. The emergency trajectory is the safety net that we use to deal with the inherent loss of optimality when using a discrete solution space.

*2) Evaluation of Cost:* In the next step, the performance or, alternatively, the cost of each proposed trajectory is calculated. This cost is a more complex definition than the collision risk used in the maneuver module. It integrates aspects like speed, estimated comfort, consumption, and traffic rules offenses, among other possible ones. The total cost of a trajectory is the weighted sum of each partial cost. These weights are set to encourage sportive, comfortable, or full-legal driving, depending on manufacturers' or customers' preferences.

With safety being the most important concern, risk is the first cost to be evaluated. The definition of *risk* was presented in the description of the maneuver module. Here, the risk cost takes into account possible instability from slipping if road friction information is available. We have

$$\text{Risk} = \sum_{i=1}^{N \text{ objects}} Prob(\text{TTC}) \, Grav(V, V_i) \tag{11}$$

where $Prob$ and $Grav$ denote the functions defined in the previous parts. In the trajectory generation, the risk is evaluated for every point on the trajectory; the total risk is the sum of the partial risks. This means that, in this step, the risk of a lane-change trajectory takes into account the risk during the lateral movement toward the other lane.

For trajectory $T_1$ in our example, the TTC with the vehicle $C_3$ is 4 s, leading to a small probability of collision. However, the gravity of a collision would be high as there is a big speed difference between two vehicles, even with the ego vehicle slowing down to 8 m/s. The risk of collision with the vehicle $C_3$ was found to be 10. The risk of a collision with the vehicle $C_2$ is believed to be 0 as its trajectory does not intersect with $T_1$. The addition of both risk costs of trajectory $T_1$ is 10.

For trajectory $T_2$ the TTC with the vehicle $C_2$ is 20 s, which gives a very low probability of collision. As its speed difference with the ego vehicle is small, the gravity of the collision is also small. The risk of collision with $C_2$ is calculated as 1. There is no collision with $C_3$ as the ego vehicle changes lanes. The risk cost of trajectory $T_2$ totals to 1.

The risk of collision with the phantom objects turns out to be zero.

The *speed cost* is calculated as the difference of the distance which could be reached at legal speed limits and the distance actually reached within the time period of the suggested trajectory.

Trajectory $T_1$ takes a big speed cost as it proposes to drive at 8 m/s which is much slower than the legal speed limit of 15 m/s. The speed cost was calculated as 20. Trajectory $T_2$, with its target speed of 10 m/s, is slightly faster, leading to a speed cost of 15.

The *comfort cost* is calculated as the quadratic integration of the variations in acceleration during the execution of the trajectory.

Both trajectory $T_1$ and trajectory $T_2$ take a substantial comfort cost. Trajectory $T_1$ requests braking in the longitudinal direction. This is not the case for trajectory $T_2$, but this one has an important acceleration in the lateral direction. Both take a comfort cost of 10.

The *consumption cost* is deducted by a weighted quadratic integration of the longitudinal acceleration and speed values needed for following the trajectory.

Trajectory $T_1$ corresponds to braking, which is consumption free. Trajectory $T_2$ takes a small consumption cost of 1 for maintaining the same cruise speed.

The *traffic rules offense cost* integrates penalties for speeding and for crossing full road marks. Driving on the left lane on a highway does not necessarily lead to penalties by law, but in the algorithms, it gets small offense costs, inviting the pilot to choose the right lane when possible.

This is the case for trajectory $T_2$, which gets an offense cost of 5 for driving on the left lane. Trajectory $T_1$ is offense cost free.

*3) Ranking of Trajectories:* After all trajectories are generated and evaluated, they are ranked by their total cost.

With the partial costs we found, we can calculate the total costs for trajectory $T_1$ and trajectory $T_2$. As the customer wants neutral driving that is not too sporty and not too conservative, we set all weights to 1. This results in a total cost of 40 for trajectory $T_1$ and 32 for trajectory $T_2$.

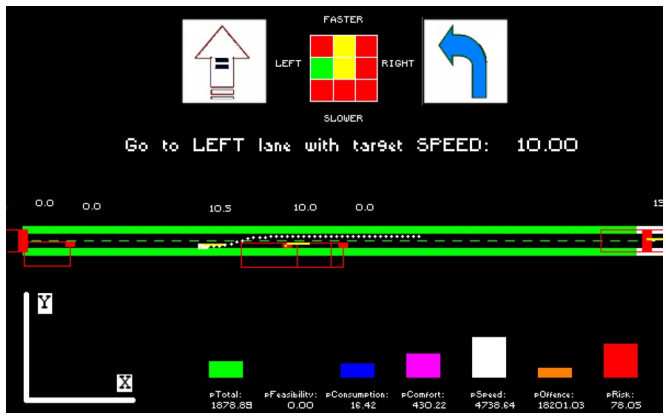


Fig. 11. Trajectory research framework.

Both trajectory $T_1$ and trajectory $T_2$ have advantages and disadvantages, but the latter takes the smallest total cost. For sporty driving, the weight of the speed cost in the total cost could be increased, leading to an even bigger difference between both trajectories. With a more conservative setting, trajectory $T_1$ could be found the best option.

In the next step, the best trajectories are mixed for a second generation of trajectories.

### C. Second Generation and Evaluation of Trajectories for Ego Vehicle

The second generation refines the discrete solution space of trajectories, leading to better performing trajectories and to smoother transitions from one trajectory to another.

A trajectory of the second generation is created by combining two best trajectories of the first generation, weighted by their total costs. This means that the new trajectory lies in between its two parent trajectories but is closer to the parent with the lowest total cost.

With the same method as in the first generation, the total cost of each of these new trajectories is calculated. They are inserted in the ranking of all trajectories.

The best trajectory of both generations, for example, trajectory $T_2$ in Fig. 10 is chosen to be executed in the case of automated driving. Alternatively, in assisted driving, several best trajectories to choose from can be presented to the driver.

The white dots in HMI presented in Fig. 11 show a (sub)optimal trajectory of the ego vehicle in white on a two-lane road in green with other objects shown as red squares. The best trajectories can easily be communicated on the nine-maneuver grid with a green case and with a message stating the target speed and target lane. Histograms show the total cost of the best trajectory, with its different cost components.

With the same algorithm or intelligence, the character of the copilot is greatly influenced by the weights of the different costs (e.g., risk, speed, comfort, and consumption) in the total performance cost and by the weights of the subcosts of the costs (e.g., slipping risk in risk cost and collision risk in risk cost). These parameters have to be tuned to clearly express the desired character of the copilot (i.e., sportive, comfortable,

and economic) and, at the same time, assure the safety of the system. This could be done with a genetic algorithm in a simulation environment, imitating the training process of human pilots.

## VII. Copilot and Driver-Maneuver Generation

The maneuver grid view presented in the previous sections turns out to be a very intuitive way to give instructions to the driver. In the opposite direction, a simple analysis of the driver's actions can give a grid representation of the driver's wishes. The grid by copilot and the grid by the driver can easily be cross evaluated. The two-way communication between the two principle actors in the vehicle is essential for a reliable and understandable cooperation.

### A. Copilot Maneuver Grid

As seen in previous sections, the maneuver module gives a performance value to each maneuver in the grid, based on a fast estimation of the collision risk. The trajectory module generates and evaluates several solutions within the maneuvers, with a multicriteria performance measurement. For communication with the driver, the results from the trajectory module are used to enrich the maneuver grid. The total cost or performance value of each calculated trajectory is mapped on the corresponding maneuver. The average of all mapped values is used as the final performance value of the grid maneuver.

The mapping between the trajectory and maneuver grid is done with respect to the target lane and the target speed of the trajectory. The following limits on the speed of each maneuver were found understandable for the driver:

1) *hold current speed*: a target speed within $-2$ m/s and $+2$ m/s of the current vehicle speed;
2) *decelerate*: a target speed of minimum 2 m/s below the current vehicle speed;
3) *accelerate*: a target speed of minimum 2 m/s above the current vehicle speed.

### B. Driver Maneuver Grid

The prediction of driver intention is usually done by considering one specific maneuver. The emergency braking assistance, which is common on new vehicles, for instance, analyzes the brake pedal position and pedal speed in order to detect an emergency maneuver initiated by the driver. In [12], an advanced estimate on the driver state is used to enhance the acceptability of the collision-warning system. Bouslimi *et al.* [13] use an approach based on a Bayesian network for identifying a driver-behavior model. Most of these methods require either an intrusive monitoring of the driver or strong computational power. We propose a fast analysis combining the driver commands and the vehicle position in the environment in order to assess the maneuver desired by the driver. This analysis is separated in a longitudinal and a lateral analysis. Both analyses are merged to give the maneuver grid of the driver.

*1) Longitudinal Analysis:* As done in existing braking-assistance systems, we use the commands of the driver (pedal position, pedal speed) and the state of the vehicle (longitudinal acceleration, speed) in order to estimate the longitudinal component of the maneuver desired by the driver. We could use a detailed map of the road to make the distinction between the power required by the slope and the real wishes of the driver or, alternatively, directly work with a slope observer, as described in [17]. This is done as follows.

*1) To determine the current state:* Using the road slope observer and knowing the current speed of the vehicle, the torque on the motor axle can be determined, together with the required position of the gas pedal and the brake pedal. By comparing these positions with the real position of those pedals, we find if the driver wants to accelerate, hold speeds, or decelerate. All data used are available on recent vehicles. A longitudinal accelerometer can give extra redundant information.

*2) To determine the driver's wish:* The variations in the position of the gas and brake pedals are also used in order to estimate the driver's wish. As for the emergency braking maneuver, a separate process can be defined that checks if the position and speed of the brake pedal exceed a given limit.

Fig. 12 shows the three membership functions for the longitudinal acceleration of the vehicle and for the speed of the brake pedal. The special function of the emergency maneuver is presented on the figure of the brake pedal speed. The functions used are normalized Gaussian functions. Only the *constant* class is tuned, the other classes are directly deducted. For the emergency braking, the maximum value is not set to 1 so that this mode is only selected if confirmed by the other indicators.

Finally, theses values are merged in order to find the longitudinal wishes of the driver.

*2) Lateral Analysis:* The analysis of the lateral behavior of the driver has the same structure.

*1) To determine the current state:* Using the knowledge of the road curvature, the vehicle speed, and vehicle parameters, the nominal steering angle to follow the lane is defined. Comparing this steering angle with the real steering angle gives a first indicator for the lateral behavior. A second, redundant, indicator uses the lateral position of the vehicle with respect to the center of the lane. This indicator is tuned with a driver-behavior analysis during normal driving on one lane. The average lateral displacement from the center of the lane is around 0.20 m for most drivers.

*2) To determine the driver's wish:* A third indicator uses the time to line crossing (see [20]) with a first-order approximation. This value integrates the lateral positioning and heading with respect to the lane. Here, too, the values found in previous experiments [19] can be used for tuning. Finally, the state of the blinkers can be integrated in a fourth indicator.

Fig. 13 shows the membership functions of two criteria, the lateral displacement, and the blinkers status. For the first one, a normalized Gaussian function is used, and the left and right lane classes are directly deduced from the same lane class. This last one is tuned using an admissible lateral displacement. For

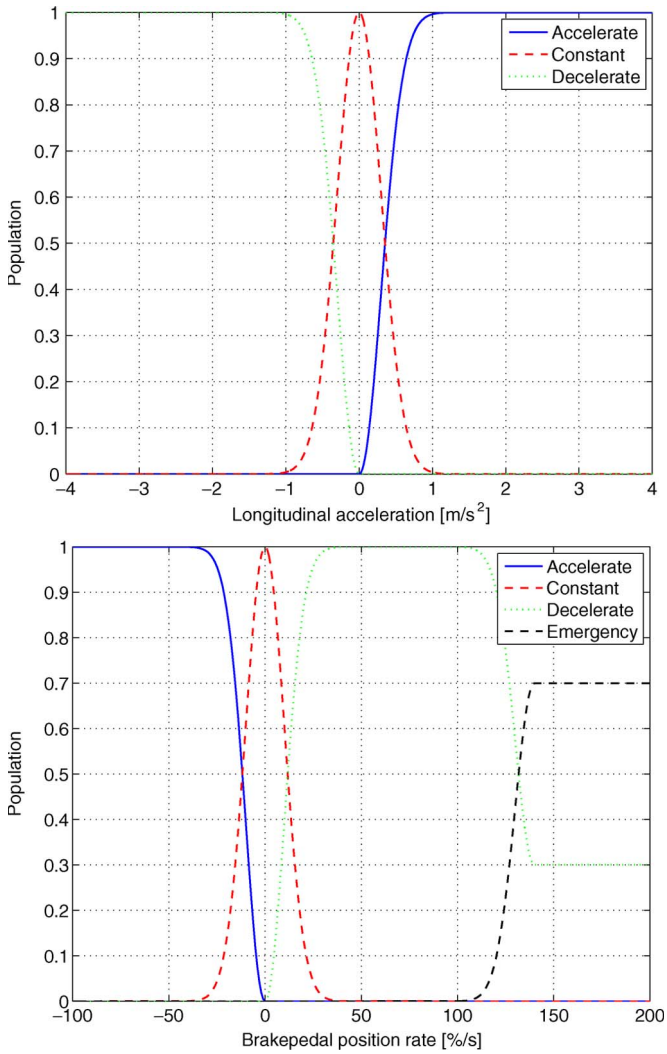


Fig. 12. Membership functions for longitudinal behavior classification. (Top) Longitudinal vehicle acceleration. (Bottom) Brake pedal speed.

the blinkers status, no class membership is set to 0 in order to be able to detect lane changes without the activation of the blinkers.

### C. First Level of Assistance

By comparing the driver-maneuver grid with the copilot grid, a first level of driving assistance can be defined. In most situations, both grids will be similar or, at least, the driver's most wished maneuver will be the same as or adjacent to the copilot optimal maneuver. When a considerable difference is detected, the cosystem can warn the driver and indicate possible risks. In a more active way, it gives a haptic feedback on the steering wheel and pedals or even temporarily take over the control of the vehicle.

For instance, a blind spot assistance could easily be derived from the comparison of the two maneuver grids. If the driver engages a lane-change maneuver to the left, without seeing a fast approaching vehicle on the target lane, the system will find an incompatibility between the driver and copilot grids and warn the driver on this specific risky situation.

We could also define an obstacle-warning application, which detects whether the driver adapts his speed when approaching an obstacle or a slow-moving vehicle. In this case, the optimal maneuver in the copilot grid will be an emergency brake or a lane-change maneuver.

Most of the current ADAS could be derived from the simple grid comparison. This framework offers a generic way for building driving-assistance systems.

## VIII. Longitudinal and Lateral Control

As presented in previous sections, the copilot can cooperate with the driver on different levels of the driving task. It can just inform the driver or take over the control of the vehicle. This section explains how the latter is done.

The trajectories described are designed to respect the physical limits of the vehicle, both in the longitudinal as lateral direction. We also avoid braking while taking a curve. This means that a decoupled longitudinal and lateral control can be used to guide the vehicle along the trajectory.

Works as [14]–[16] present a coupled longitudinal and lateral control. A control for highly dynamic trajectories, such as obstacle-avoidance trajectories, is developed in [16].

### A. Lateral Control

The vehicle positioning relative to the selected trajectory is done with the following variables: the relative yaw angle $\Delta\psi$ and the lateral displacement $y_L$ at a certain distance $l_s$ ahead, relative to the center of gravity of the vehicle [21]. With this formulation, the trajectory curvature $\rho_{\text{ref}}$ is seen as a disturbance input, and the control problem becomes a disturbance rejection problem with zero as target values for $\Delta\psi$ and $y_L$. Only a front-view video camera is needed to measure these two variables [23]. It is assumed that the vision algorithm allows us to estimate the curvature of the trajectory for a feedforward controller that improves the performance during the transient phases. The steering angle has the following form:

$$\delta_f = K_1\Delta\psi + K_2 y_L + K_3\rho_{\text{ref}} \tag{12}$$

where $K = [K_1, K_2]$ is computed using an $H_\infty$ optimization [22], i.e.,

$$K = \arg\min_K \left\| T_{\rho_{\text{ref}}\to[\Delta\psi, y_L]^T} \right\|_\infty \tag{13}$$

while $K_3$ is computed from the closed-loop system steady-state gain.

Fig. 14 shows a lane change maneuver using this algorithm. Transitions, both in position and yaw angle, are efficient and smooth, and there is no overshoot on the target lane.

### B. Speed Control

The following longitudinal control computes the required acceleration to follow the speed profile proposed by the trajectory module. It is decoupled from the lateral control.

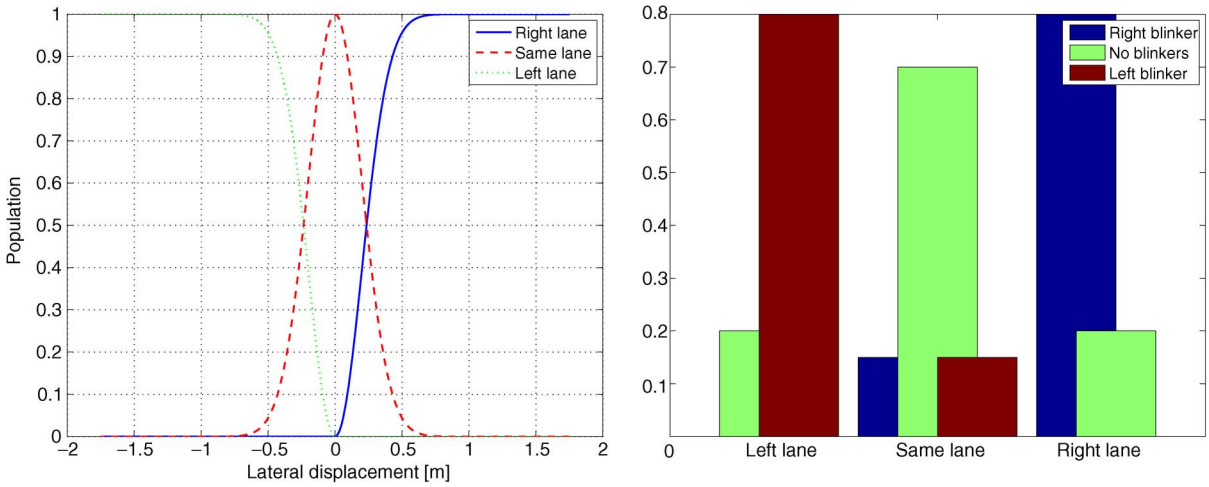


Fig. 13. Membership functions for lateral behavior classification. (Left) Lateral displacement. (Right) Blinkers.

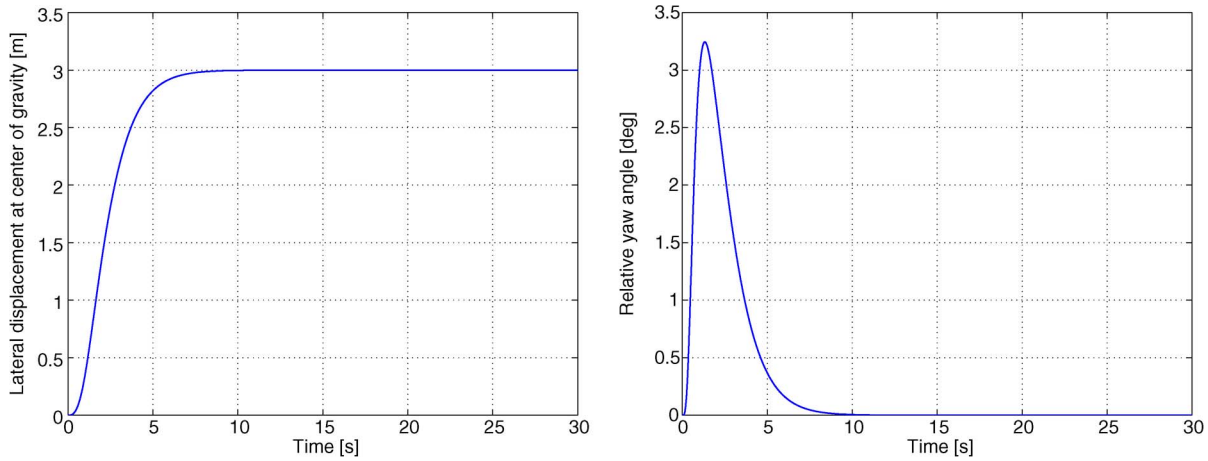


Fig. 14. Lateral control results during a lane-change maneuver.

*1) Vehicle Modeling [28]:* The vehicle model used for control synthesis is a simplified nonlinear model. The longitudinal equation of the drive train is described by (see Table II)

$$\left(m+\frac{(J_{wr}+J_{wf})}{h^2}\right)a=\frac{T_s-T_b-M_{rr}}{h}-F_a-mg\sin(\theta). \tag{14}$$

With a nonslip assumption $(v=R_g h w_e)$ and $(T_e=R_g T_s)$, the term of for the traction effort in (14) can be eliminated. This gives

$$T_e-R_g\left(T_b+M_{rr}+hF_a+mgh\sin(\theta)\right)=I_t a \tag{15}$$

with

$$I_t=\frac{\left(J_e+R_g^2(J_{wr}+J_{wf}+mh^2)\right)}{R_g h}. \tag{16}$$

*2) Formulation Procedure:* A second-order sliding-mode algorithm is chosen to perform the presented control for the following reasons.

1) Sliding-mode techniques are easy to implement.
2) It is robust with regard to the model errors and variations.

TABLE II
VEHICLE MODEL PARAMETERS

| | |
|---|---|
| $T_b$ | Brake torque (N.m) |
| $M_{rr}$ | Rolling resistance torque (N.m) |
| $h$ | height of the center of the wheel (m) |
| $F_a$ | aerodynamic force (N) |
| $g$ | gravity (9.81 m.s$^{-2}$) |
| $\theta$ | Road slope angle (deg) |
| $a$ | acceleration (m.s$^{-2}$) |
| $J_{wr}$ $J_{wf}$ | Rear/front wheel inertias (1.2825 kg.m$^2$) |
| $J_e$ | Engine/ transmission inertias(0.2630 kg.m$^2$) |
| $R_g$ | gear ratio ( final gear included) |
| $T_s$ | Shaft torque (N.m) |
| $v$ | Vehicle speed (m.s$^{-1}$) |
| $\omega_e$ | Engine speed (rpm) |
| $m$ | Vehicle mass (kg) |

3) Second-order sliding-mode techniques avoid the known chattering problem linked to the classical sliding modes.

In order to synthesize the control algorithm, a sliding surface $S$ has to be defined, with the objective to be reached, i.e.,

$$S=(v-v^*)+\beta(\dot{v}-\dot{v}^*).$$

This sliding surface takes into account the speed error between the vehicle speed $v$ and the speed imposed by the trajectory $v^*$ in a first term and the acceleration error in a second. The coefficient $\beta$ sets the weight of both terms. The term $(v - v^*)$ performs the speed tracking, and the term $(\dot{v} - \dot{v}^*)$ assures a given comfort level in the longitudinal direction.

The equivalent control method is applied in order to refine the sliding-mode technique. This equivalent control $u_{\text{eq}}$ is obtained as

$$u_{\text{eq}} = \left\{ u / \dot{S}(X, u) = 0 \right\}$$

since the degree of the considered system is equal to 1. $X$ is the state vector, and $u$ is the control input. This gives

$$u_{\text{eq}} = \dot{v}^* - \beta(\ddot{v} - \ddot{v}^*).$$

The second-order sliding modes algorithm is the twisting algorithm presented in [24]–[26]. We have

$$\dot{u} = \begin{cases} u, & \text{if } |u| < |u_{\text{eq}}| \\ -K_M sign(S), & \text{if } S\dot{S} > 0 \text{ and } |u| \leq |u_{\text{eq}}| \\ -k_m sign(S), & \text{if } S\dot{S} \leq 0 \text{ and } |u| \leq |u_{\text{eq}}|. \end{cases} \quad (17)$$

Two gains $k_m$ and $K_M$ are to be tuned to obtain a convergence of the method in a finite time $t_f$, i.e.,

$$k_m > K_M > 0$$

$$k_m > 4\frac{C_M}{S_0}$$

$$k_m > \frac{C_0}{c_m}$$

$$K_M > \frac{C_M k_m}{c_m} + 2\frac{C_0}{c_m}.$$

The parameters $C_M$, $c_m$, $S_0$, and $C_0$ are determined with the four conditions through the Utkin theorem [27].

*3) Simulation Results:* The control algorithm is designed and simulated in MATLAB/SIMULINK with the vehicle model presented in Section VIII-B1. The gains $k_m$ and $K_M$ are tuned to 10 and 30. The weight factor $\beta$ is chosen for having smooth variations in the speed: $\beta = 0.25$. The simulation results are quite good in terms of speed following, but it is hard to get the longitudinal acceleration smooth enough. In a prototype vehicle, it would generate high jerks in the longitudinal direction.

A solution to this problem consists of replacing the function $sign$ in the twisting algorithm by a *saturation* function to soften the variation in the accelerations. The *saturation* function is chosen close to the $sign$ function in order to respect the theoretical background of the algorithm. The *saturation* function can be written as

$$\begin{cases} sat(S) = sign(S), & \text{if } S \in ]-\inf, -\tau] \text{ and } S \in [\tau, \inf[ \\ sat(S) = \frac{S}{\tau}, & \text{if } -\tau < S < \tau. \end{cases}$$

For this method, the simulation of the speed tracking is given in Fig. 15. The constant $\tau$ is chosen to be 0.05. A good speed tracking is obtained with a very short convergence time. Such a control algorithm can easily be implemented in a prototype vehicle.

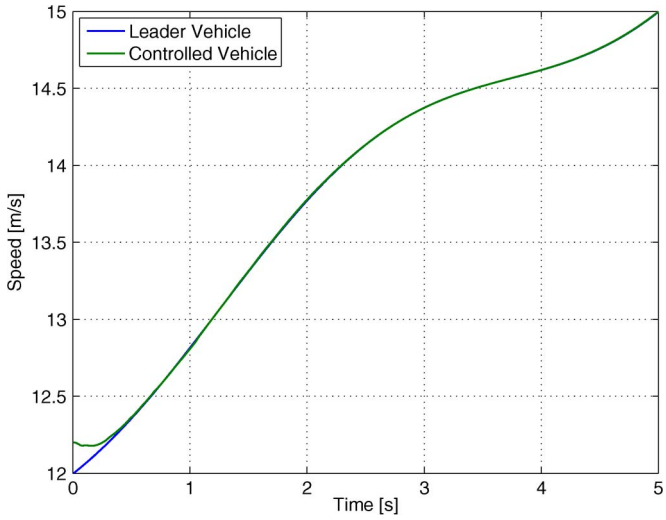


Fig. 15. Desired speed profile given by the trajectory module versus actual vehicle speed.

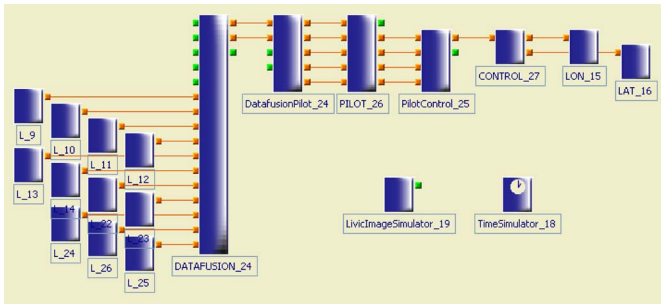


Fig. 16. RTMaps interface for the cosystem.

## IX. Simulation

### A. Description of Simulation Environment

With several parameters needed to be tuned, a simulation tool is crucial in the development of these algorithms. The simulation of different scenarios is done with SiVIC,[1] which was internally developed software [18]. The communication between SiVIC and different modules of cosystem is done by RTMaps.[2] This allows us to directly plug and play the software in the test vehicles when the algorithm is optimized. The software structure is shown in Fig. 16. The environment data, the behavior of the ego vehicle, and ten surrounding vehicles are calculated by SiVIC and sent to the modules of the cosystem through the simulated sensors denoted with a $L$. The output of SiVIC is the information that we can expect from real sensors and digital map information in the test vehicles. All data are fused by a data-fusion module. This module is not the topic of our research; it is a black box that results in a local map with all relevant data for our trajectory-calculation algorithms, such as position, speed, and acceleration of the ego vehicle and other

[1] SiVIC is now a professional software of CIVITEC, http://www.civitec.net/.
[2] RT-Maps is developed by Intempora, www.intempora.com.

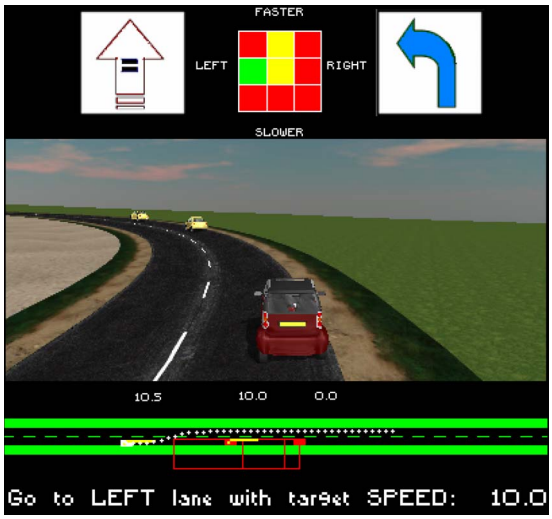


Fig. 17. Visualization of the maneuver grid and optimal trajectory in the HMI.

vehicles, the description of road marks, and possibly the speed limits. As most (accurate) data are provided by sensors on the ego vehicle, the output of data fusion is given in a coordinate axis attached on the ego vehicle. Additional information in an absolute axis, from global positioning system sensors, for example, could easily be converted in the ego vehicle axis with a Cartesian transformation.

The maneuver and trajectory algorithms that we developed are brought together in the pilot module. It reads the local map provided by the data-fusion module and delivers a spatial description of the best trajectory to be followed and a temporal description of the recommended speed. The cost components of the pilot are passed for information for the driver. In the HAVEit project, these performance indicators are used by a mode-selection unit. This high-level module constantly compares driver and automation behavior and decides on the best automation mode. In dangerous situations for example, this module (temporarily) hands over control from driver to automation. In situations with partial or full automation, the control module is charged of following the optimal trajectory calculated by the pilot module. In the case of pure driving assistance without automation control, the driver remains fully responsible for the driving task. The output of control module is then a copy of the driver's steering, acceleration, and braking actions. Today's challenge for car equipment supplier is to combine human driver actions with copilot actions and is one of the main focuses of the European HAVEit project. During simulation, the data of the control module for longitudinal and lateral actions are passed to the ego vehicle via the simulation software. In the test vehicles, these outputs are passed to actuators on the steering wheel and pedals. Two interface modules, namely data fusion–pilot and pilot–control, act as a translation between the data structures used by the copilot and the data fusion and control modules.

A separate HMI module, shown in Fig. 17, reads and combines all relevant information from the data fusion, pilot, and the control module. The challenge is to provide complete information in a clear way. In a first version of our HMI, a vehicle position for the next 10 s according to the best trajectory is drawn on a simple local environment map, together with the position and speed of other objects. The corresponding longitudinal and lateral actions are summarized by simple arrows and words. The maneuver grid shows the corresponding maneuver by a green case and gives a value on the other possible maneuvers in a color code. The different cost components could be added as additional information.

### B. Results of Simulation

The cost indicators used by the copilot algorithm are a good base to evaluate the performance of the copilot. They are also used to compare the copilot performance with the performance of the human driver, or even with the performance of other copilot algorithms. The most important tuning parameters are the weights of each of the partial costs with respect to the total cost. During the simulations, they greatly influenced the character of the algorithm, making the difference between a sporty and comfortable copilot, with the same intelligence. Through the HMI, the human driver is able to switch between normal, sportive, comfortable, and low-consumption copilot.

Our algorithm is designed to be integrated on simple embedded microsystems. This means that the calculation cycle time and total memory used are other important performance indicators for our copilot.

On a two-lane road and five near objects, the total calculation time is 10 ms on a standard office personal computer. As explained in the description of the trajectory module, the number of relevant objects could be reduced to a maximum of eight. Therefore, considering code optimizations, the maximum calculation time is believed to be below 100 ms on an embedded system, corresponding to an acceptable 3–4 m longitudinal displacement before reaction at highway speeds.

The algorithm is now ready to be tested on a physical test vehicle, which was in-house equipped with high-technology proprioceptive and exteroceptive sensors and control actuators. First tests will be done in an assistance mode to gradually move to higher forms of automation.

The result of the HAVEit project will be a demonstrator that combines human and automation actions, switching between different modes of automation, which go from giving simple warnings to perform a high form of automation.

## X. Conclusion

In this paper, a method has been presented to determine an optimal vehicle trajectory with the consideration of the road environment and other vehicles (moving or not). Various methods, mainly from robotics, exist in this domain, but their computation time and the memory needed are not available on today's engine control units, and they do not allow an interaction with the driver. The simulation results show that the proposed method is very fast, outputting a new optimal trajectory every 10 ms.

In the first step of the copilot, the possible maneuvers are ranked based on fast collision-avoidance criteria. A grid with nine maneuvers and their associated risk is outputted, together with the target lane and target speed of the best maneuver.

In the second step, the copilot evaluates a small set of trajectories within the best maneuvers and can also evaluate the fusion of the best trajectories. On each trajectory, several performance indicators are evaluated, such as risk, speed, consumption, comfort, and legal driving. A weighted sum of the different indicators gives the total performance indicator of a trajectory. The weights in this sum set the character of the copilot and can be tuned to the character chosen by the car manufacturer or driver.

The method allows an easy comparison of the copilot's and driver's decision at the high maneuver level. This can be used in the following two ways: 1) to correct or 2) to execute the driver's decision. With the grid visualization, driving-assistance systems and blind-spot warnings can be easily defined.

For the automatic control of the vehicle along the trajectory, a decoupled longitudinal and lateral control is presented. The complete system has been integrated in the SiVIC simulator to test and evaluate the copilot in several scenarios. Results show a very good behavior both in assistance mode as in an automated mode during lane following, lane changes, and distance regulation to a vehicle in the front.

The next steps further develop the interaction with the driver, implementing the algorithms in a physical vehicle and enhancing the control laws.

## APPENDIX
## POLYNOMIAL COEFFICIENTS EVALUATION

This Appendix presents the calculation of the coefficients of the fourth-order polynomial describing future longitudinal positions and the fifth-order polynomial describing future lateral positions. The coefficients are directly linked with the kinematical constraints we put on the begin and the end state of the future motion.

In the following equations, we refer with $p$, $v$, and $a$ to the position, speed, and acceleration, respectively. The begin state is indicated by 0 and the end state by $T$.

For the longitudinal direction, we use a fourth-order polynomial. Polynomial coefficients are determined using the constraints of continuity, which can be written as follows:

$$\{\, p_x(t) :\ g_{0x} + g_{1x}t + g_{2x}t^2 + g_{3x}t^3 + g_{4x}t^4 \tag{18}$$

$$\begin{cases} p_x(0) = p_{x0} \\ \dot{p_x}(0) = v_{x0} \\ \ddot{p_x}(0) = a_{x0} \\ \dot{p_x}(T) = v_{xT} \\ \ddot{p_x}(0) = a_{xT} = 0. \end{cases} \tag{19}$$

From (18) and (19), we develop the coefficients as a function of the constraints, i.e.,

$$\begin{cases} g_{0x} = p_{x0} \\ g_{1x} = v_{x0} \\ g_{2x} = a_{x0}/2 \\ g_{3x} = (-v_{x0}T - 2/a_{x0}T^2 + v_{xT}T)/T^3 \\ g_{4x} = (-1/2v_{x0}T + 1/4a_{x0}T^2 - 1/2v_{xT}T)/T^4. \end{cases} \tag{20}$$

In the lateral direction, the constraints of continuity in begin longitudinal position, speed, and acceleration and in end longitudinal position, speed, and acceleration can be met by a fifth-order polynomial. This polynomial is expressed relatively to the road frame. We have

$$\{\, p_y :\ g_{0y} + g_{1y}t + g_{2y}t^2 + g_{3y}t^3 + g_{4y}t^4 + g_{5y}t^5 \tag{21}$$

$$\begin{cases} p_y(0) = p_{y0} \\ \dot{p_y}(0) = v_{y0} \\ \ddot{p_y}(0) = a_{y0} \\ p_y(T) = p_{yT} \\ \dot{p_y}(T) = v_{yT} \\ \ddot{p_y}(T) = a_{yT} = 0. \end{cases} \tag{22}$$

From (21) and (22), we develop the coefficients as a function of the constraints

$$\begin{cases} g_{0y} = p_{y0} \\ g_{1y} = v_{y0} \\ g_{2y} = a_{y0}/2 \\ g_{3y} = (-10p_{y0} - 6v_{y0}T - 3/2a_{y0}T^2 + 10p_{yT})/T^3 \\ g_{4y} = (-15p_{y0} + 8v_{y0}T + 3/2a_{y0}T^2 - 15p_{yT})/T^4 \\ g_{5y} = (-6p_{y0} - 3v_{y0}T - 1/2a_{y0}T^2 + 6p_{yT})/T^5. \end{cases} \tag{23}$$

## REFERENCES


[1] J. A. Michon, "A critical view of driver behavior models: What do we know, what should we do?" in *Human Behavior and Traffic Safety*, L. Evans and R. C. Schwing, Eds. New York: Plenum, 1985, pp. 485–520.

[2] J. C. Hayward, "Near miss determination through use of a scale of danger," Penn. State Univ., University Park, PA, Rep. TTSC 7115, 1972.

[3] A. Lambert and N. Le Fort-Piat, "Safe task planning integrating uncertainties and local maps federations," *Int. J. Robot. Res.*, vol. 19, no. 6, pp. 597–611, Jun. 2000.

[4] M. Parent, "Advanced urban transport: Automation is on the way," *IEEE Intell. Syst.*, vol. 22, no. 2, pp. 9–11, Mar./Apr. 2007.

[5] SPARC Eur. Project. [Online]. Available: www.sparc-eu.net/

[6] HAVEit Eur. Project. [Online]. Available: http://www.haveit-eu.org

[7] AIDE Eur. Project. [Online]. Available: http://www.aide-eu.org

[8] PReVENT Eur. Project. [Online]. Available: http://www.aide-eu.org

[9] HUMANIST Network of Excellence. [Online]. Available: http://www.noehumanist.org

[10] A. Simon and J. C. Becker, "Vehicle guidance for an autonomous vehicle," in *Proc. IEEE Int. Conf. Intell. Transp. Syst.*, 1999, pp. 429–434.

[11] S. Glaser, A. Rakotonirainy, D. Gruyer, and L. Nouveliere, "An integrated driver-vehicle-environment (I-DVE) model to assess crash risks," in *Proc. Australasian Road Safety Res., Policing Edu. Conf.*, Melbourne, Australia, 2007.

[12] K. Hayashi, Y. Kojima, K. Abe, and K. Oguri, "Prediction of stopping maneuver considering driver's state," in *Proc. IEEE Intell. Transp. Syst. Conf.*, 2006, pp. 1191–1196.

[13] W. Bouslimi, M. Kassaagi, D. Lourdeaux, and P. Fuchs, "Augmented naıuml;ve Bayesian network for driver behavior modeling," in *Proc. IEEE Intell. Vehicle Symp.*, 2005, pp. 236–242.

[14] X.-Y. Lu and J. K. Hedrick, "Impact of combined longitudinal, lateral and vertical control on autonomous road vehicle design," *Int. J. Vehicle Auton. Syst.*, vol. 2, no. 1/2, pp. 40–70, 2004.

[15] R. Rajamani, H.-S. Tan, B. K. Law, and W. B. Zhang, "Demonstration of integrated longitudinal and lateral control for the operation of automated vehicles in platoons," *IEEE Trans. Control Syst. Technol.*, vol. 8, no. 4, pp. 695–708, Jul. 2000.

[16] T. Acarman, Y. Pan, and U. Ozguner, "A control authority transition system for collision and accident avoidance," *Vehicle Syst. Dyn.*, vol. 39, no. 2, pp. 149–187, Feb. 2003.

[17] Y. Sebsadji, S. Glaser, S. Mammar, and J. Dakhlallah, "Road slope and vehicle dynamics estimation," in *Proc. Amer. Control Conf.*, Seattle, WA, Jun. 11–13, 2008, pp. 4603–4608.

[18] D. Gruyer, C. Royere, N. du Lac, G. Michel, and J. M. Blosseville, "SiVIC and RTMaps, interconnected platforms for the conception and the evaluation of driving assistance systems," in *Proc. ITS World Congr.*, London, U.K., Oct. 2006.

[19] S. Glaser, R. Labayrade, S. Mammar, J. Douret, and B. Lusetti, "Validation of a vision based time to line crossing computation," in *Proc. IEEE Intell. Vehicle Conf.*, Tokyo, Japan, 2006, pp. 200–205.
[20] S. Mammar, S. Glaser, and M. Netto, "Time to line crossing for lane departure avoidance: A theoretical study and an experimental setting," *IEEE Trans. Intell. Transp. Syst.*, vol. 7, no. 2, pp. 226–241, Jun. 2006.
[21] N. Minoiu Enache, M. Netto, and S. Mammar, "Driver steering assistance for lane departure avoidance based on hybrid automata and composite Lyapunov function," *IEEE Trans. Intell. Transp. Syst.*, vol. 11, no. 1, pp. 28–39, Mar. 2010.
[22] T. Raharijaona, G. Duc, and S. Mammar, "Linear parameter-varying control and H221e-synthesis dedicated to lateral driving assistance," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Parma, Italy, 2004, pp. 407–412.
[23] S. Mammar, "Two-degree-of-freedom H1 optimization and scheduling, for robust vehicle lateral control," *Vehicle Syst. Dyn.*, vol. 34, pp. 401–422, 2000.
[24] S. V. Emelyanov, S. K. Korovin, and L. V. Levantovsky, "Second order sliding modes in controlling uncertain systems," *Sov. J. Comput. Syst. Sci.*, vol. 24, no. 4, pp. 63–68, 1986.
[25] L. V. Levantovsky, "Second order sliding algorithms: Their realization," in *Dynamics of Heterogeneous Systems*. Moscow, Russia: Inst. Syst. Stud., 1985, pp. 32–43.
[26] A. Levant, "Sliding order and sliding accuracy in sliding mode control," *Int. J. Control*, vol. 58, no. 6, pp. 1247–1263, Dec. 1993.
[27] V. I. Utkin, *Sliding Modes in Control Optimization*. New York: Springer-Verlag, 1992.
[28] J. K. Hedrick, J. C. Gerdes, D. B. Maciuca, and D. Swaroop, "Brake system modeling, control and integrated brake/throttle switching: Phase I," Calif. PATH Res. Rep., Univ. Calif., Berkeley, CA, UCB-ITS-PRR-97-21, 1997.
[29] S. Glaser, D. Gruyer, L. Nouvelière, and J. M. Blosseville, "Collision mitigation system improvements with avoidance trajectory computation," in *Proc. AVEC*, Kobe, Japan, 2008.
[30] SAFESPOT Eur. Project. [Online]. Available: http://www.safespot-eu.org
[31] A. Stentz, "Optimal and efficient path planning for partially-known environments," in *Proc. IEEE Int. Conf. Robot. Autom.*, 1994, pp. 3310–3317.
[32] P. Cheng, Z. Shen, and S. M. LaValle, "RRT-based trajectory design for autonomous automobiles and spacecraft," *Arch. Control Sci.*, vol. 11, no. 3/4, pp. 167–194, 2001.
[33] C. Urmson and R. Simmons, "Approaches for heuristically biasing RRT growth," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2003, vol. 2, pp. 1178–1183.
[34] Y. Kuwata, J. Teo, S. Karaman, G. Fiore, E. Frazzoli, and J. How, "Motion planning in complex environments using closed-loop prediction," in *Proc. AIAA Guid., Navigat., Control Conf. Exhibit*, Honolulu, HI, Aug. 2008.
[35] N. A. Melchior and R. Simmons, "Particle RRT for path planning with uncertainty," in *Proc. IEEE Int. Conf. Robot. Autom.*, Apr. 10–14, 2007, pp. 1617–1624.
[36] J. Barraquand, B. Langlois, and J. C. Latombe, "Numerical potential field techniques for robot path planning," *IEEE Trans. Syst., Man, Cybern.*, vol. 22, no. 2, pp. 224–241, Mar./Apr. 1992.
[37] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *Int. J. Robot. Res.*, vol. 5, no. 1, pp. 90–98, 1995.
[38] J. Borenstein and Y. Koren, "The vector field histogram—Fast obstacle avoidance for mobile robots," *IEEE Trans. Robot. Autom.*, vol. 7, no. 3, pp. 278–288, Jun. 1991.
[39] S. Quinlan and O. Khatib, "Elastic bands: Connecting path planning and control," in *Proc. IEEE Conf. Robot. Autom.*, 1993, pp. 802–807.
[40] T. Brandt, T. Sattel, and J. Wallaschek, "On automatic collision avoidance systems," in *Proc. SAE Trans. J. Passenger Cars: Electron. Elect. Syst.*, 2005, pp. 431–440.
[41] T. B. To, M.-M. Meinecke, F. Schroven, S. Nedevschi, and J. C. Knaup, "CityACC—On the way towards an intelligent autonomous driving," in *Proc. 17th World Congr. Int. Fed. Autom. Control*, Seoul, Korea, Jul. 6–11, 2008, pp. 9534–9539.
[42] F. Biral, M. Da Lio, and E. Bertolazzi, "Combining safety margins and user preferences into a driving criterion for optimal control-based computation of reference maneuvers for an ADAS of the next generation," in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 6–8, 2005, pp. 36–41.
[43] P. Falcone, F. Borrelli, J. Asgari, H. E. Tseng, and D. Hrovat, "Predictive active steering control for autonomous vehicle systems," *IEEE Trans. Control Syst. Technol.*, vol. 15, no. 3, pp. 566–580, May 2007.
[44] Z. Shiller and S. Sundar, "Emergency lane-change manoeuvres of autonomous vehicles," *ASME Trans. Dyn. Syst., Meas. Control*, vol. 120, no. 1, pp. 37–44, Mar. 1998.

**Sébastien Glaser** received the Dipl.-Ing. degree from the École Nationale des Travaux Publics de l'État, Vaulx en Velin, France, in 2000, the M.S. degree in image analysis and synthesis from the Université Jean Monnet, Saint Etienne, France, and the Ph.D. degree in automatic control, with emphasis on the vehicle dynamic analysis, from the Université d'Evry, Evry, France, in 2004.

Since 2004, he has been a Researcher with the Laboratoire sur les Interactions Véhicule Infrastructure Conducteur, which is a research unit of the Central Laboratory on Civil Engineering (LCPC), Versailles, France. He works on driving-assistance design and on driver studies. He is currently involved in several European Union initiatives (HAVEit, SAFESPOT) and leads a French initiative on low-speed automation (ANR-ABV).

**Benoit Vanholme** received the master's degree in electromechanical engineering from the University of Ghent, Ghent, Belgium, in 2005, with a thesis on the universal design of an autopilot for ships.

He began his career as a Project Engineer with Altran Belgium and Altran France, with two projects on lean manufacturing. He is currently a Research Engineer with the Central Laboratory on Civil Engineering (LCPC), Versailles, France, working on a copilot algorithm for the European FP7-project HAVEit.

**Saïd Mammar** (M'02) received the Dipl.-Ing. degree from the École Supérieure d'Électricité, Gif-sur-Yvette, France, in 1989, the Ph.D. degree in automatic control from the Université Paris XI-Supelec, Orsay, France, in 1992, and the Habilitation to Direct Research (HDR) degree from Université d'Evry, Evry, France, in 2001.

From 1992 to 1994, he held a research position with the French National Institute on Transportation Research and Safety (INRETS), France, where he worked on traffic network control. From 1994 to 2002, he was an Assistant Professor with the Université d'Évry, Évry, France. He is currently a Professor with and the Head of the Department of Electrical Engineering, Université d'Évry. He is currently a scientific and university attaché with the French Embassy, The Hague, The Netherlands. His actual research interests include robust control, vehicle longitudinal and lateral control for driving assistance, and intelligent transportation systems.

**Dominique Gruyer** was born in France, in 1969. He received the M.S. and Ph.D. degree from the University of Technology of Compiègne, Compiègne, France, in 1995 and 1999, respectively.

Since 2001, he has been a Researcher with the French National Institute on Transportation Research and Safety (INRETS), with the perception team of the Laboratory on Interactions between Vehicles, Infrastructure, and Drivers (LIVIC). He works on the study and the development of multisensor/source association, combination, and fusion. His works include the conception of on-board driving-assistance systems and, more precisely, carrying out multiobstacle detection and tracking, extended perception, accurate localization, anticollision systems, collision mitigation, numerical tests, and evaluation of advanced driving-assistance systems. He is involved in multisensor fusion tasks in several European and French projects dealing with intelligent vehicles (HAVEit, Isi-PADAS, CVIS, CARSENSE, eMOTIVE, LOVE, ARCOS, MICADO, etc.). He is a multisensor fusion expert for several companies, and teaches at Orsay (Paris XI), the Ecole Nationale des Ponts et Chaussées, the Ecole des Mines de Paris, and the University of Evry. He is responsible of the Simulation for Vehicle, Infrastructure, and Sensors (SiVIC) team.

**Lydie Nouvelière** was born in France in 1975. She received the Ph.D. degree from the University of Versailles, Versailles, France, in 2002.

She is currently an Assistant Professor with the Informatique, Biologie Intégrative et Systèmes Complexes (IBISC) Laboratory, Department of Electrical Engineering, University of Evry, Evry, France. Since 2007, she has been a Scientific Advisor with the Laboratory on Interactions between Vehicles, Infrastructure, and Drivers, Central Laboratory on Civil Engineering (LCPC-LIVIC), Versailles. Her research interests include robust control, vehicle longitudinal control, and low fuel consumption for driving assistance. She is currently involved in the European integrated project HAVEit and the French PREDIT-ANR project ANGO.