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

Sébastien Glaser, Benoit Vanholme, Saïd Mammar, Dominique Gruyer, Lydie Nouveliere

Maneuver based trajectory planning for highly autonomous vehicles on real road with traffic and driver interaction

Sébastien Glaser, Benoit Vanholme, Saïd Mammar, Dominique Gruyer, Lydie Nouveliere

***Abstract*—This paper presents the design and first test on a simulator of a vehicle trajectory planning algorithm that takes into account traffic and obstacles on a highway. The proposed algorithm is designed to run in an embedded environment with a low computational power such as an ECU, to be implementable in commercial vehicles. The target platform has a clock frequency of less than 150 MHz, 150 KB RAM memory and 3 MB program memory. To the best of our knowledge, the methods available in literature fail to comply with these limitations.**

The trajectory planning is performed by a two steps 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 such as accelerating, decelerating and changing lanes. The second step is a more detailed evaluation of several possible trajectories within the accepted maneuvers. It optimizes according to additional performance indicators such as comfort, speed and consumption. The output of the module is a trajectory described in the vehicle frame that represents the recommended vehicle states (position, heading, speed and acceleration) for the following seconds.

***Index Terms*—Decision System, Human Machine Interface, Autonomous Vehicles, Advanced Driving Assistance Systems, Autonomous Intervention and Control, 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 the sole responsible of 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 ACC[1]. The demonstrations on Californian Highways at the end of the 90s proved that autonomous driving in a secured environment (a dedicated lane on a highway with magnets) is possible. Most of the current works focus on the interaction with the driver and the design of the HMI [27]. The recent European project SPARC [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, HAVEit, focuses on the cooperation between the driver and a co-pilot system on the maneuver decision level. It defines different automation modes ranging from fully human to highly automated. It also proposes a methodology for choosing the best mode at each moment. This is the context in which the presented system is developed.

The objective of the 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 too high, the system can decide to intervene in the driving task.

In this article, we focus on the generation of a trajectory which is optimal with respect to safety, desired speed, driver comfort, fuel consumption and the traffic rules. The trajectory planning algorithm must comply with two strong specifications. Firstly, for a strong interaction with driver, it has to be understandable in its decisions and respect the driver's wish. Secondly, the algorithm must run on an embedded system which is compliant with the vehicle industry. The target platform has a clock frequency of less than 150 MHz, 150 KB of RAM memory and 3MB of program memory. In this environment, the computation of the trajectory planning algorithm must be done in few milliseconds.

For calculating a safe maneuver or trajectory, approaches developed in robotic control (such as in [3]) integrate environment sensing and vehicle control directly in the trajectory generator. We can distinguish two main approaches:

- *Tree exploration based methods:* these methods search for solutions in a fine grid attached to the environment. In a highly dynamic road environment, the size of this grid can become very big which is not compatible with the available memory on embedded environments.
- *Potential evaluation based methods:* these methods associate a time-dependent potential field to each object in the environment. Based on these fields, a trajectory is evaluated. The disadvantage of these methods is their high computational cost.

The calculation memory and computation power of these approaches will not be available on commercial vehicles, in the next years to come. Vehicles that integrate a trajectory planning for autonomous driving already exist, like the CyberCars [4], but only work at low speeds, in slowly varying

S. Mammar and Lydie Nouveliere are with Université d'Évry val d'Essonne, France. IBISC: Informatique, Biologie Intégrative et Systèmes Complexes - FRE CNRS 3190, 40 rue du Pelvoux CE1455, 91020, Evry, Cedex, France, (e-mail: said.mammar@iup.univ-evry.fr).

B. Vanholme, S. Glaser and D. Gruyer are with INRETS/LCPC - LIVIC Laboratoire sur les Interactions Véhicule-Infrastructure-Conducteur. 14, route de la Minière, Bât 824, 78000, Versailles, France, e-mail: ((vanholme, glaser)@lcpc.fr, gruyer@inrets.fr).

environments, with a limited number of obstacles. Vehicles involved in the DARPA 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, energy limitations of an everyday vehicle. These systems have two additional drawbacks:

- *The interaction with the driver*: they focus on a full automation without any interaction with the driver. For psychological and legal reasons, we want to give the driver the possibility to remain in the driving loop. This means that interaction must be integrated at each level of the system.
- *The generation of the trajectory*: Algorithms in the literature propose to optimize the trajectory generation for avoiding collisions and respecting the vehicle dynamics. We believe that, for an optimal driving experience, others indicators should be introduced, with respect to traffic rules, fuel consumption and driver wishes and comfort.

Our approach exists in designing a trajectory planning of two levels. At a high level, nine maneuver cases are identified by combining basic actions: three in 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 to the right). To these nine manoeuvres, a safe state manoeuvre can be added, which corresponds with stopping on the right most lane, and an emergency manoeuvre, corresponding with maximal braking till standstill. 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, several trajectories are evaluated within each of the previously accepted maneuvers. This level aims at optimizing the trajectory definition with respect to a finer 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 position and speed of the vehicle, for the following seconds.

The remainder of this paper is divided as follows. The next section presents the integration of the work within the HAVEit project. Section III gives an overview of the global architecture of the proposed system. Section IV contains the description of the high level, with the risk analysis of the maneuvers and the development of a mechanism to ensure its stability over time. Section V describes the method to generate and evaluate trajectories and to output the optimal ones. Section VI discusses how the maneuvers calculated by the automation are compared with driver's maneuvers. Section VII describes the controller used to follow the trajectory both in the longitudinal as in the lateral direction. Finally, section VIII shows some results in a simulation environment. Section IX gives the conclusion and a view on the future works.

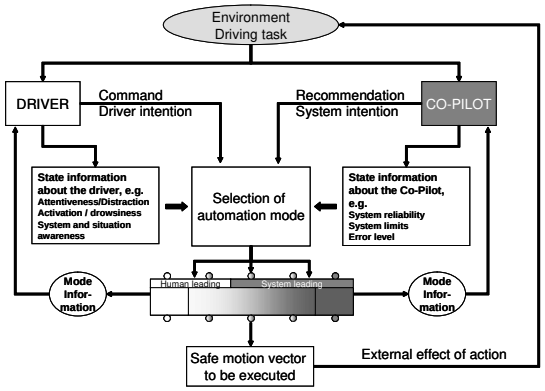

Fig. 1. Driver and Algorithm comparison (HAVEit Project)

II. Integration of the work within HAVEit project

HAVEit is a research project of the seventh framework programme of the European Commission, which co-funds 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 towards highly automated driving. HAVEit will significantly contribute to higher traffic safety and efficiency usage for passenger cars, busses 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 three measures:

- The design of the task repartition between the driver and co-driving system.
- The presentation of a failure tolerant safe vehicle architecture including an advanced redundancy management.
- The development and validation of the next generation of ADAS directed towards a higher level of automation as compared to the current state of the art.

The work presented in this paper touches with the first and the third point. It is integrated in the co-pilot of the co-system depicted in figure 1. The sub project in HAVEit that builds the co-system is leaded by DLR[2] and has as partners, INRIA[3], ICCS[4], IZVW[5], Continental and Ibeo. The co-system analyses 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. Architecture

In this section the architecture of the co-pilot is presented. The goal of the proposed system is two-fold: firstly the system allows a high driver interaction, secondly it is able to optimize a trajectory according to different performance criteria, in a everyday traffic environment. This multi-criteria optimal trajectory assures a high level of safety and can be used to control the vehicle. But, in order to meet the safety level

[2]DLR is the German Aerospace Center

[3]INRIA is the National Institute for Research in Computer Science and Control

[4]ICCS is the Institute of Communications and Computer Systems

[5]IZVW is the Center for Traffic Sciences

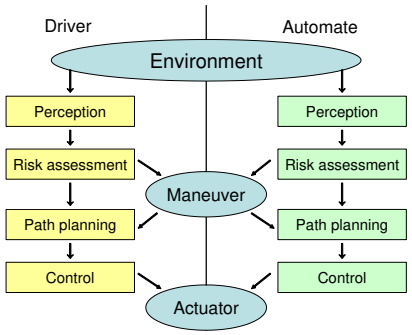

Fig. 2. Driver and automate process

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:

- The environment is highly dynamic. Even in a specific environment such as highways, there can be numerous static and dynamic.
- The computation time and memory are limited when working with a safe and embedded architecture.
- The system must be able to closely interact with the driver, both by receiving commands coming from the driver and by giving him a clear feedback.
- The system has to take into account the real road environment constraints, such as lanes, which drastically limit the action space.

A. Functional architecture

The automation and the driver follow the same process (figure 2). They can interact at different levels of this process: e.g. for the perception of the environment, the automation can help with the detection of a front vehicle in foggy weather. We can define two main ways of interaction:

- Definition of the maneuver: this is the high level verbalization of the longitudinal and lateral targets.
- Execution of the trajectory through the actuators.

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

- Proposition of the maneuver: the driving assistant analyses 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).
- 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, shown in figure 4. Maneuvers are defined relatively to the actual state of the ego vehicle. With the green maneuvers the multi-criteria 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

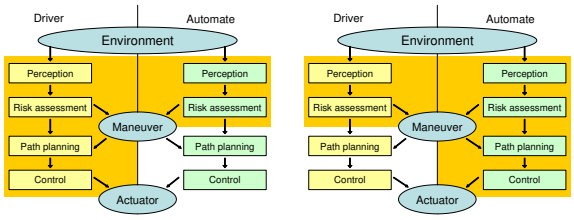

Fig. 3. Proposed driving assistant

Change to left lane Accelerate	Stay on current lane Accelerate	Change to Right lane Accelerate
Change to left lane Current speed	Stay on current lane Current speed	Change to Right lane Current speed
Change to left lane Decelerate	Stay on current lane Decelerate	Change to Right lane Decelerate

Fig. 4. Grid of maneuver

intentions, by monitoring the state of the indicators and accelerator and braking pedals, as will be explained in section VI.

B. Software architecture

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

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

1) Maneuver module: This first module aims to provide a clear interface to the driver and to drastically reduce the computation time of the second module; the trajectory module. The module analyses 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

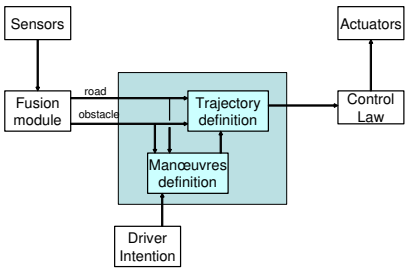

Fig. 5. General architecture

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 constraints on longitudinal and lateral acceleration, to assure their controllability. The final output is a set of spatio-temporal 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.

IV. 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 a real time running. 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 to the right). For safety reasons, these nine manoeuvres, a safe state manoeuvre can be added, which corresponds with stopping on the right most lane, and an emergency manoeuvre, corresponding with maximal braking till standstill.

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 set 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 two criteria:

- The probability that the event occurs,
- 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 which is deduced from the analysis of traffic indicators. The 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 inter vehicular time, the time to collision and the reaction distance.

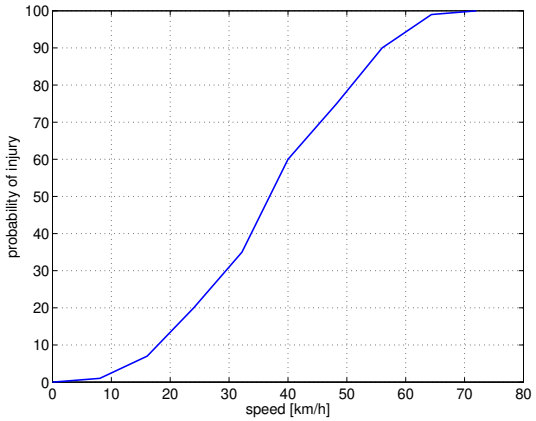

Fig. 6. EES (speed) and MAIS scale (probability) for a moderate injury

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

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 equivalent energetic speed (namely 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 with the damage done to the human in the vehicle. The speed can be computed using the following equations:

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

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

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

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 fatality. Figure 6 represents the likelihood of a moderate injury (MAIS>2, Maximum Abbreviated Injury Scale) with respect to the EES.

B. Possibility of collision

In order to define the possibility of collision, we have to analyse condition of the traffic 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.

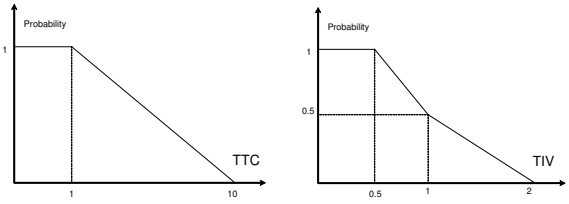

Fig. 7. TTC (left) and TIV (right) based possibility of collision

Two main types of danger exist when considering a lane with traffic: low inter vehicle distance and high speed differences. The traffic indicators show the first type is easily detected by the inter vehicular time, while the second can be estimated by using the time to collision. In order to define a possibility of collision, these two criteria are used.

1) Time to collision: The first parameter is relative to the Time To Collision. Hayward [2] defines the TTC as: *The time required for two vehicles to collide if they continue at their present speed and on the same trajectory*. The TTC formula is:

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

In this formula, D_i is the relative distance with vehicle i. Projects 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:

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

Looking at the system as collision mitigation by braking [26], a collision is highly possible as soon as the time to collision 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. Figure 7 (left) represents the evaluation of the possibility associated with the TTC, namely P_{TTC}.

2) Inter vehicular 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 inter-vehicle time. The TIV is defined as:

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

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 analyse the reactions of the followed vehicle. It is also frequently used in Advanced Cruise Control (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 figure 7 (right). The possibility of collision associated with the TIV is now denoted as P_{TIV}. The 1 s boundary is taken for the same reason as explained for the TTC.

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, 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:

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

Where R_{TTC} is the risk associated with the TTC and R_{TIV} 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:

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

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

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

where γ is the considered deceleration, set to $0.8g$.

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, figure 7 and equations 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. Figure 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 inter vehicular distance factor. But when this speed drops to 15 m/s, in a traffic jam for instance, the time to collision 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. Figure 9 shows the resulting speed. At the beginning, the front

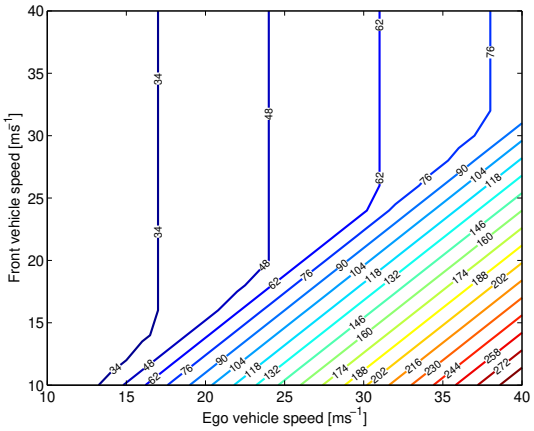

Fig. 8. Distance that ensures a zero risk

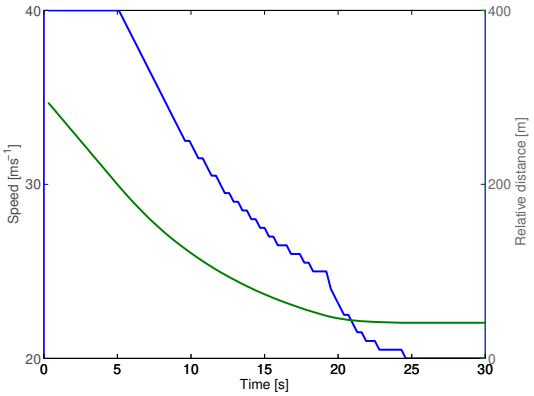

Fig. 9. Fast vehicle approaching scenario

vehicle is too far to have an impact on our vehicle, so 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 inter vehicular distance regulates the speed, so a stronger decrease appears. The simulation stops at $T = 25$ s when both vehicles have the same speed with a safe inter vehicular distance.

During the simulation, the deceleration of the controlled vehicle remains below $0.3g$. The steps in speed profile result from the direct association of computed speed with the realized speed, 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 VI 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.

V. Trajectory Generation and Evaluation

The trajectory module finds the optimal trajectory within the accepted maneuvers (green and yellow on figure 4). It gives a detailed description of it for a time frame of 5 to 10 seconds. The fast ranking of the different zones of the solution space by the maneuver module, greatly increases the efficiency of the

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

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 (sub-optimal) 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:

- Prediction of the evolution of the environment: giving the future positions in time for every non-ego vehicle or obstacle in the environment.
- First generation and evaluation of trajectories in the maneuver grid: ranking the trajectories with a multi-criteria performance indicator.
- 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 figure 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.

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 at the outside of 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 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 8 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 to 10 seconds 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:

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

where X_k is the predicted state, X_0, 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, as 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 equation 8, the state vector is $X_k = [s, v]^t$, where s is the curvilinear abscises and v the vehicle speed, the input U_0 is the acceleration, which is constant. The matrices A, B and Q are:

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

where Δt is the time step of the prediction, and ε_i denotes the error to be defined.

In our example of figure 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 a 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 VI) by calculating the probability of the three lateral maneuvers: staying on current lane, changing to right lane, and changing to 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 towards 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 the next section.

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 figure 10.

B. First generation and evaluation of trajectories for ego vehicle

For the ego vehicle a fix number of smooth polynomial trajectories is generated, each with a different target speed and target lane within the ranges of the maneuvers specified by the maneuver module. In simulations, a number of 2 to 5 trajectories per maneuver show a good compromise between calculation time and resolution of the solution space.

1) Generation of trajectories: The three current values (i.e. begin of time description) of position, speed, and acceleration and two target values (i.e. end of time description) of speed and acceleration give 5 constraints in the longitudinal direction. In the lateral direction, there are 6 constraints as both current and target values of position, speed and acceleration are specified. This defines a 4^{th} and a 5^{th} 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 the longitudinal axis in the driving direction of the vehicle and Y right hand side perpendicular on X, we write the equations for the longitudinal and lateral positions p with respect to the time t, the difference with the current time:

$$\begin{cases} p_x \; : \; g_{0x} + g_{1x}t + g_{2x}t^2 + g_{3x}t^3 + g_{4x}t^4 \\ p_y \; : \; g_{0y} + g_{1y}t + g_{2y}t^2 + g_{3y}t^3 + g_{4y}t^4 + g_{5y}t^5 \end{cases} \quad (10)$$

For the calculation of the coefficients of these polynomials using the begin and end constraints, we refer to the work of [7]. The conversion of these constraints into coefficients for our application is worked out in Annex A.

In our example we show the generation of a trajectory for two different maneuvers proposed by figure 4. For the maneuver *Stay on current lane and decelerate*, trajectories with target speeds of 2 m/s, 4 m/s, 6 m/s 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 m/s, 10 m/s and 11 m/s are calculated. The trajectory *Change to left lane at* 10 *m/s* is called trajectory T_2.

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 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.

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

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

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

2) Evaluation of cost: In a 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 offences, amongst 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.

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.

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

where $Prob$ and $Grav$ denote the functions defined in the previous parts.

For trajectory T_1 in our example, the time to collision 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 time to collision 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, also the gravity of the collision is 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 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 as 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 offence cost* integrates penalties for speeding and for crossing full road marks. Driving on the left lane on a high way does not necessarily lead to penalties by law but in the algorithms it gets small offence costs inviting the pilot to choose the right lane when possible.

This is the case for trajectory T_2 which gets an offence cost of 5 for driving on the left lane. Trajectory T_1 is offence 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 a neutral driving, not too sporty, 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.

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 a 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 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 figure 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.

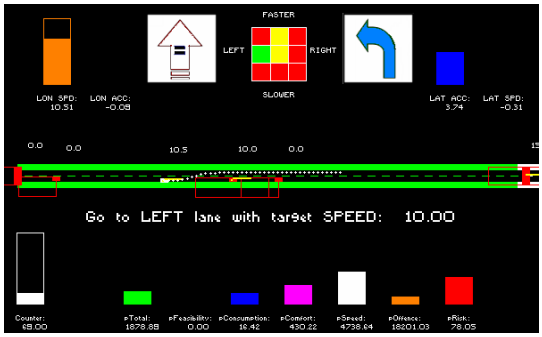

Fig. 11. Trajectory research framework

The white dots in Human Machine Interface (HMI) presented in figure 11 show an (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 9-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 pilot is greatly influenced by the great number of parameters used in the algorithm. Optimization can be done with a genetic algorithm in a simulation environment. This promotes pilot behavior with low average costs on test tracks, imitating the training process of human pilots.

VI. Co-pilot and driver maneuver generation

The maneuver grid view presented in 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 co-pilot and the grid by the driver can easily be cross-evaluated. The two-ways communication between the two principle actors in the vehicle is essential for a reliable and understandable cooperation.

A. Co-pilot 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 multi-criteria 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:

- Hold current speed: a target speed within -2 m/s and $+2$ m/s of the current vehicle speed.
- Decelerate: a target speed of minimum 2 m/s below the current vehicle speed.
- 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 considering one specific maneuver. The emergency braking assistance which is common on new vehicles, for instance, analyses the brake pedal position and pedal speed in order to detect a emergency maneuver initiated by the driver. In [9], an advanced estimate on the driver state is used to enhance the acceptability of the collision warning system. [10] uses 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 wished 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 is 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 wished 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 [14]. 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 is available on recent vehicles. A longitudinal accelerometer, can give extra redundant information.
2) *To determine the driver wish*: the variations in the position of the gas and brake pedals are also used in order to estimate the driver 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.

Figure 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.

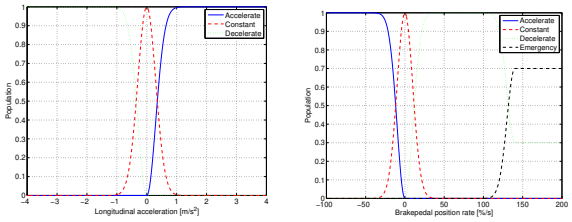

Fig. 12. Membership functions for longitudinal behavior classification (left: longitudinal vehicle acceleration, right: brake pedal speed)

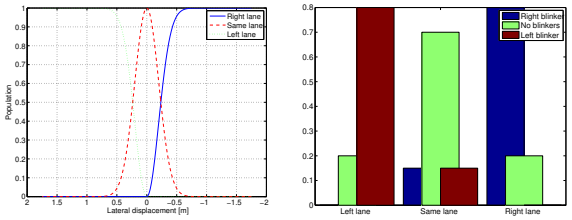

Fig. 13. Membership functions for lateral behavior classification (left: the lateral displacement, right: the blinkers)

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 wish*: A third indicator uses the Time to Line Crossing (TLC, see [17]), 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 [16], can be used for tuning. Finally, the state of the blinkers can be integrated in a forth indicator.

Figure 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, the left lane and right lane classes are directly deduced from the same lane class. This last one is tuned using an admissible lateral displacement. For 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 co-pilot grid, a first level of driving assistance can be defined. In most situations, both grids will be similar or, at least, the driver most wished maneuver will be the same as or adjacent to the co-pilot optimal maneuver. When a considerable difference is detected, the co-system can warn the driver and indicate possible risks. In a more active way it give 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 co-pilot grids and warn the driver on this specific risky situation.

We could also define an obstacle warning application, which detects wether the driver adapts his speed when approaching an obstacle or a slow moving vehicle. In this case the optimal maneuver in the co-pilot grid will be a 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.

VII. Longitudinal and Lateral control

As presented in previous sections, the co-pilot 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 limits of the road adherence, 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 [12], [13], [11] present a coupled longitudinal and lateral control. A control for highly dynamic trajectories, such as obstacle avoidance trajectories, are developed in [13].

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 [18]. With this formulation, the trajectory curvature ρ_{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 for measuring these two variables [20]. It is assumed that the vision algorithm allows to estimate the curvature of the trajectory, for a feed-forward controller that improves the performance during the transient phases. The steering angle has the following form:

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

where $K = [K_1, K_2]$ is computed using an H_∞ optimization [19]:

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

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

Figure 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.

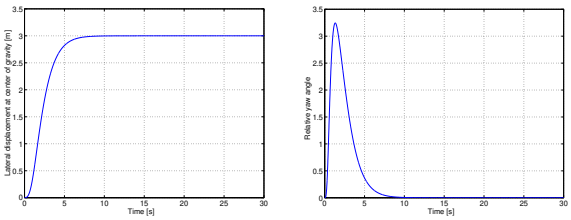

Fig. 14. Lateral control results during a lane change maneuver

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})
θ	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})
ω_e	Engine speed (rpm)
m	Vehicle mass (kg)

TABLE II
VEHICLE MODEL PARAMETERS

1) 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.

a) Vehicle modeling [25]: The vehicle model used for control synthesis is a simplified non-linear model. The longitudinal equation of the drive train is described by:

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

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

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

with

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

b) Formulation procedure: A second order sliding modes algorithm is chosen to perform the presented control, for the following reasons:

- Sliding modes techniques are easy to implement,
- It is robust with regards to the model errors and variations,
- Second order sliding modes techniques avoid the so-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:

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

This sliding surface takes into account, the speed error between the vehicle speed v and the speed imposed by the trajectory

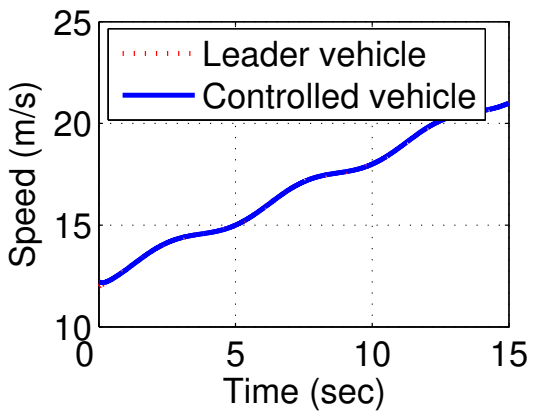

Fig. 15. Desired speed profile given by the trajectory module versus actual vehicle speed

v^* in a first term, and the acceleration error in a second. The coefficient β sets the weight of both terms. The term $(v - v^*)$ performs the speed tracking, 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 modes technique. This equivalent control u_{eq} is obtained as:

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

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

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

The second order sliding modes algorithm is the twisting algorithm presented in [22], [23], [21]:

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

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

$$k_m > K_M > 0$$
$$k_m > 4\frac{C_M}{S_0}$$
$$k_m > \frac{C_0}{c_m}$$
$$K_M > \frac{C_M k_m}{c_m} + 2\frac{C_0}{c_m}$$

The parameters C_M, c_m, S_0 and C_0 are determined with the four conditions, through the Utkin Theorem [24].

c) Simulation results: The control algorithm is designed and simulated in MATLAB/SIMULINK with the vehicle model presented in paragraph VII-A1a. The gains k_m and K_M are tuned to 10 and 30. The weight factor β 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 in 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

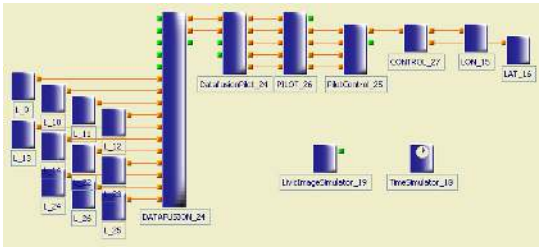

Fig. 16. RT-Maps interface for the co-system

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 figure 15. The constant τ is chosen to be 0.05. A good speed tracking is obtained, with a very short convergence time. Such a control algorithm can be easily implemented in a prototype vehicle.

VIII. 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[15], which is internally developed software. The communication between SiVIC and different modules of co-system is done by RTMaps[6]. 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 figure 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 co-system through the simulated sensors denoted with a L. The output of SiVIC is the information we can expect from real sensors and digital map information in the test vehicles. All data is fused by a data fusion module. This module is not the topic of our research; it is a black box which 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 vehicles, the description of road marks, and possibly the speed limits. As most (accurate) data is 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 GPS sensors for example, could easily be converted in the ego vehicle axis with a Cartesian transformation.

The maneuver and trajectory algorithms 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

[6]RT-Maps is developed by Intempora, www.intempora.com

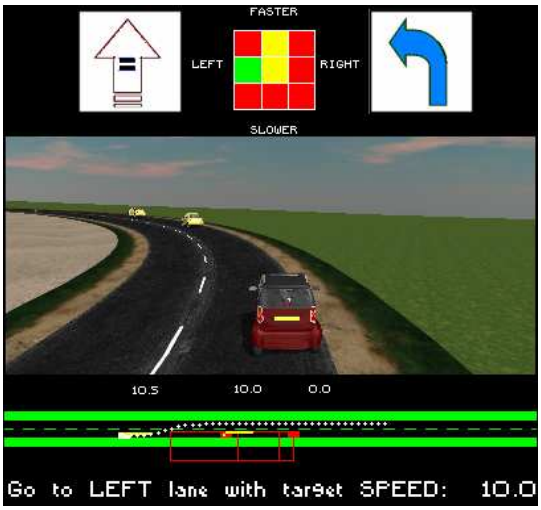

Fig. 17. Visualization of the maneuver grid and optimal trajectory in the HMI

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 of 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 OEM's is to combine human driver actions with co-pilot 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 data fusion-pilot and pilot-control act as a translation between the data structures used by the co-pilot and the data fusion and control modules.

A separate Human Machine Interface (HMI) module, shown in figure 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 color code. The different cost components could be added as additional information.

B. Results of simulation

The cost indicators used by the co-pilot algorithm are a good base to evaluate the performance of the co-pilot. They are also used to compare the co-pilot performance with the performance of the human driver, or even with the performance

of other co-pilot 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 co-pilot, with the same intelligence. Through the HMI, the human driver is able to switch between normal, sportive, comfortable and low consumption co-pilot.

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 co-pilot.

On a two lanes road and five near objects, the total calculation time is 10 ms on a standard office PC. As explained in the description of the Trajectory Module, the number of relevant objects could be reduced to a maximum of 8. Therefore considering code optimizations, the maximum calculation time is believed to be below 100 ms on an embedded system, corresponding to an acceptable 3 to 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-tech 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.

IX. Conclusion

In this paper a method was 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 ECU's 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 co-pilot, 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 co-pilot 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 co-pilot and can be tuned to the character chosen by the car manufacturer or driver.

The method allows an easy comparison of the co-pilot's and driver's decision, at the high maneuver level. This can be used in two ways: to correct or to execute the driver's decision. With the grid visualization, driving assistance systems as blind spot warning 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 co-pilot 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.

Next steps are further developing the interaction with the driver, implementing the algorithms in a physical vehicle and enhancing the control laws.

Acknowledgement

The authors would like to thank the Project HAVEit and the European Commission for co-funding this research.

References

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

[2] Hayward, J.Ch. (1972). *Near miss determination through use of a scale of danger.* Report no. TTSC 7115, The Pennsylvania State University, Pennsylvania.

[3] A. Lambert, N. Le Fort-Piat. *Safe task planning integrating uncertainties and local maps federations* Int. Journal of Robotics Research, 19(6), pp. 597-611, 2001.

[4] Michel Parent *Advanced Urban Transport: Automation Is on the Way* in IEEE Intelligent Systems, April 2007

[5] SPARC European project *www.sparc-eu.net/*

[6] HAVEit European project *see http://www.haveit-eu.org*

[7] Andreas Simon and Jan C. Becker *Vehicle Guidance for an Autonomous Vehicle* in IEEE International Conference on Intelligent Transportation Systems, 1999.

[8] Glaser, Sebastien and Rakotonirainy, Andry and Gruyer, Dominique and Nouveliere, Lydie *An Integrated Driver-Vehicle-Environment (I-DVE) model to assess crash risks*. In Proceedings 2007 Australasian Road Safety Research, Policing and Education Conference, Melbourne, Australia.

[9] Hayashi, K.; Kojima, Y.; Abe, K.; Oguri, K. *Prediction of stopping maneuver considering driver's state* IEEE Intelligent Transportation Systems Conference, 2006.

[10] Bouslimi, W.; Kassaagi M.; Lourdeaux D. and Fuchs P. *Augmented Naïve Bayesian Network for Driver Behavior Modeling* IEEE Intelligent Vehicle Symposium, 2005

[11] Lu, X.-Y. and Hedrick, J. K. *Impact of combined longitudinal, lateral and vertical control on autonomous road vehicle design* Int. J. Vehicle Autonomous Systems, Vol. 2, Nos. 1/2, 2004

[12] Rajamani, R.; Tan, H.-S.; Law, B. K. and Zhang, W. B. *Demonstration of Integrated Longitudinal and Lateral Control for the Operation of Automated Vehicles in Platoons*, IEEE Transaction on Control Systems Technology, Vol. 8, No. 4, July 2000

[13] Acarman, T.; Pan, Y. and Ozguner U. *A Control Authority Transisition System for Collision and Accident Avoidance* Vehicle System Dynamics, 2003, Vol. 39, No. 2, pp. 149-187

[14] Sebsadji, Y.; Glaser, S.; Mammar, S. and Dakhlallah, J. *Road Slope and Vehicle Dynamics Estimation*, American Control Conference 2008, Seattle, USA, June 11-13th, 2008.

[15] Gruyer, D.; Royere, C.; du Lac, N.; Michel, G.; Blosseville, J.M. *SiVIC and RTMaps, interconnected platforms for the conception and the evaluation of driving assistance systems* ITS World Congress, October 2006, London

[16] S. Glaser S., R. Labayrade, S. Mammar, J. Douret, and B. Lusetti. *Validation of a vision based time to line crossing computation.* In IEEE Intelligent Vehicle Conference, Tokyo, Japon, 2006.

[17] Mammar, S.; Glaser, S. and Netto, M. *Time to Line Crossing for Lane Departure Avoidance: a Theoretical Study and an Experimental Setting*, IEEE Transactions on Intelligent Transportation Systems, vol 7, No 2, pp 226-241, June 2006.

[18] N. Minoiu, M. Netto, S. Mammar, *Driver steering assistance for lane departure avoidance based on hybrid automata and composite Lyapunov function*, to be published in IEEE Transactions on Intelligent Transportation Systems, 2008.

[19] T. Raharijaona, G. Duc, S. Mammar, *Linear Parameter-varying control and H221e-synthesis dedicated to lateral driving assistance*, IEEE Intelligent Vehicles Symposium IV '04, Parme, 2004.

[20] S. Mammar, *Two-Degree-of-Freedom H1 Optimization and Scheduling, for Robust Vehicle Lateral Control*, Vehicle Systems Dynamics journal, vol 34, pp. 401-422, 2000.
[21] Emelyanov S.V., Korovin S.K., Levantovsky L.V., *Second order sliding modes in controlling uncertain systems*, Soviet Jour. of Computer and System Science, 24(4):63-68, 1986.
[22] Levantovsky L.V., *Second order sliding algorithms: their realization in dynamics of heterogeneous systems*, Institute for System Studies, Moscow, pages 32-43, 1985.
[23] Levant A., *Sliding order and sliding accuracy in sliding mode control*, Int. Jour. of Cont., 58(6):1247-1263, 1993.
[24] Utkin V.I., *Sliding modes in control optimization*, Springer-Verlag, 1992.
[25] Hedrick J.K., Gerdes J.C., Maciuca D.B., Swaroop D., *Brake system modeling, control and integrated brake/throttle switching : Phase I*, UCB-ITS-PRR-97-21, California PATH Research Report, University of California, Berkeley, 1997.
[26] Glaser, S.; Gruyer, D.,Nouvelière, L.; Blosseville, J.M. *Collision Mitigation System Improvements with Avoidance Trajectory Computation*, AVEC 2008, Kobe Japan
[27] SAFESPOT European project *http://www.safespot-eu.org*

APPENDIX

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:

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

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

>From 18 and 19, we develop the coefficients in function of the constraints:

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

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.

$$\left\{ \; p_y \; : \; g_{0y} + g_{1y}t + g_{2y}t^2 + g_{3y}t^3 + g_{4y}t^4 + g_{5y}t^5 \right. \tag{21}$$

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

>From 21 and 22 we develop the coefficients in function of the constraints:

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