Automatic Fault Detection and Execution Monitoring in Cooperative Maneuvering

Reza Balaghiasefi[1], Michael Düring[1], Kai Franke[1] and Armin Zimmermann[2]

***Abstract*— One strong motivation for introducing Vehicle-to-Vehicle technology is added safety. This technology will allow cooperative maneuver planing to prevent many accidents in the future. However, safety for road users can only be effectively guaranteed if the calculated motion plan for all participants is followed accurately and without deviation. A novel algorithm for monitoring of cooperative motion plans is presented. It solves a conflict by generating a state space for the possible maneuvers for the involved traffic participants, tracking the progress of these maneuvers, and reacting to deviations to alleviate accidents and ensuring a safe conflict resolution. The proposed monitoring algorithm is applicable to all kinds of road environments and road users. Preliminary results indicate the wide usability and performance of this approach.**

I. Introduction

A major motivation for current research in intelligent transportation systems (ITS) and cooperative systems with communication is the vision of accident-free driving. This can be approached by using fully or partly autonomous vehicles in safety-critical situations with fast coordinated reaction without deviation. Both factors, fast coordinated reactions and reactions without deviation, play an important role in accident avoidance.
In order to generate coordinated reaction, the focus of current research in the field of cooperative driving is on communication using Vehicle-to-Vehicle (V2V) technology. Geiger et al. [7] present wide-ranging applications of V2V technology. A comprehensive literature review Zhang et al. [25] explains the different architectures and strategies that have been developed in either centralized or decentralized methods to control and coordinate a group of vehicles. Furthermore, Franke et al. [5] present a reference architecture for cooperative driver assistance systems (CDAS) and cooperative integrated safety systems (CISS) within the field of cooperative driving. During et al. [4] present an algorithm for cooperative maneuver planning, applicable to different road users interferences in diverse traffic situations.

Fig. 1 illustrates a solution in the form of motion plans for all participants of a dangerous overtaking scenario, which can solve the conflict situation with cooperative maneuver planning. However, safety for road users can only be effectively guaranteed if the calculated motion plan for all participants is realized accurately and without deviation. In reality, during execution, unexpected changes in the plan or the environment may occur. This could be a deviation from the target condition of the reference trajectory or a sudden appearance of an unexpected obstacle on the road. Such changes, which were not taken into account in the planning phase, can have a catastrophic outcome. In order to prevent this from occurring, the execution of the cooperative driving maneuvers has to be constantly monitored and checked versus the original plan.

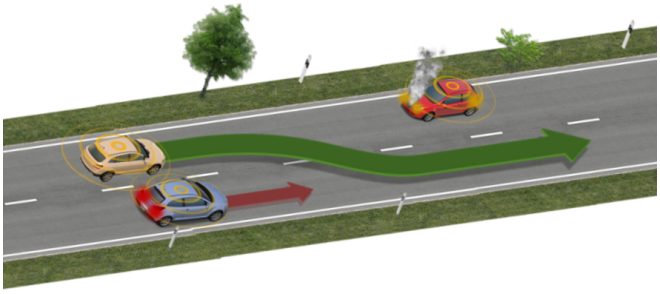


Fig. 1. Cooperative Maneuver: Solution in the form of communicated motion plan for all participants of a dangerous overtaking scenario.

In order to assess the safety of a planned maneuver, diverse approaches are known. These approaches explain interfering effects of the task demands of road users by relying on different methods. The majority of works on prediction of the behavior of traffic participants use filter techniques such as Kalman filters [16], [23] learning mechanisms such as neural networks [15], [22], [24] and Bayesian networks [11], [17], [19]. They monitor selected road sections and learn motion patterns for anomaly detection. The downside of a prognosis at fixed states is that the predictions are specific for this particular road segment and are presumably not representative for other traffic situations. Formulations based on path following control [2], [8], [13] aspire to minimize an error function concerning a given path employing the kinematic and/or dynamic model of the vehicle. In these methodologies, the problem of collision avoidance is often ignored or cast to the replanning phase of the global planner. Formulations on trajectory tracking control [3], [9], [21] can be interpreted as a system that delivers a consistent high dynamic tracking achievement to enable a smooth effectual, and continuous movement of the vehicle along a chosen trajectory. As such, approaches that do not perform collision checks for the given state of the vehicle

[1] Group Research and Development Volkswagen AG, Wolfsburg, 38440, Germany firstname.lastname@volkswagen.de
[2] Ilmenau University of Technology, Ilmenau, 98693, Germany armin.zimmermann@tu-ilmenau.de

and the environment can enter into dangerous states, especially if latencies exist in the path planning phase.

None of the above mentioned concepts analyse the deviation from the target condition of the motion plan and changes in the environment during the execution simultaneously. Moreover, none considers the integrated conflict with respect to safe end-states, safe trajectories towards these states and interaction between the vehicles. In order to address this integrated conflict, a novel approach is presented here, which concentrates on the execution monitoring of the motion plan that will enable the vehicles to safely and accurately navigate towards their destinations. It also allows detecting and evaluating hazards with reference to the mutual interaction between vehicles and can be employed in diverse traffic situations.

II. Problem Statement

In order to get a better idea of the addressed problem and to show the necessity of having a monitoring system for cooperative maneuvering, a reference architecture as a platform and the required information are described below.

A. Reference architecture

In their work on CDAS and CISS architectures [5], Franke et al. underline the necessity for a monitoring system. The solution proposed is based on a deliberative control architecture. This control architecture consists of the four phases "sensing", "modeling", "planning", and "acting". The sensing phase consists of the local environment model of one of the involved vehicles and determines the conflict. The conflict is broadcast to the other participants in the environment via V2V communication. The modeling phase characterizes an extension of the sensing phase. All relevant vehicles can share information concerning the scene and their states. The received information is aggregated in an extended, collective scene description and merged with the environmental model based on on-board sensors. During the planning phase, each vehicle can carry out its own planning and send this to all other participants in terms of an offer. The involved vehicles evaluate these offers in terms of cost and/or benefit to choose the best maneuver. The result of the selection and validation is also sent via V2V communication. These messages include a plan in the form of safe trajectories towards safe end points within a defined time horizon for all involved participants.

During et al. [4] present a cooperative maneuver planning algorithm for calculating such plans. During the acting phase, the cooperative driving maneuvers are executed using controllers located on-board the vehicles. This phase is extended beyond the deliberative control architecture to include monitoring at the end of the process. The monitoring must check that the vehicles are executing the plan correctly or detect inadmissible deviations at the earliest possible stage.

B. Required Information

The presented safety assessment requires the following information about a plan to monitor the execution:

- The number (z) of involved participants within the defined time horizon (T) for execution of the plan.
- The motion plan in the form of safe reference trajectories (RT) for each involved participant (rt_i) as a set of points (p_i) between starting point (SP) and safe end-point (sTP). Each trajectory point is characterized by a position (x, y), speed (v), yaw angle (heading) (ψ) and time (t).

$$\begin{aligned} RT &= \{rt_1, rt_2, ..., rt_z\}, \\ rt_i &= \{p_1, p_2, ..., p_w\}, \\ p_i &= (x, y, v, \psi, t). \end{aligned}$$

A plan is always generated in dependence of a certain environment. Therefore, the boundary conditions of the plan are also relevant.

- The geometric description (M) of the road segment
- The position and geometry (O) of static obstacles
- The position (x, y), speed (v) and yaw angle (heading) (ψ) for all involved participants

Note that the acquisition of this data is not the subject of this work but assumed to be available. The planned trajectories of the involved participants are also known since they are independently planned and broadcasted. The geometric description of the relevant road sections can be extracted, e.g., using the method of Smadja et al. [20]. For lane detection, one generally uses LIDAR sensors and/or cameras [1]. The same sensors are also normally used for the disclosure of static obstacles. The detection of the involved participants and the estimation of their positions, speed and yaw angle (heading) can also be estimated using on-board sensors or using V2V communication [14], [18]. Based on these information, a novel concept is presented to evaluate the execution of a planned multi-vehicle maneuver.

III. Concept and Execution Monitoring Algorithm for Cooperative Maneuvering

A. Concept

The presented concept is based upon three founding pillars: "Detection", "Evaluation", and "Reaction".

Firstly, the monitoring algorithm detects any changes in the plan and the environment. Secondly, the algorithm evaluates all detected changes with regards to the three criteria. Finally, based on the evaluation of the detected changes in the plan and/or the environment, the algorithm will decide for an appropriate reaction. These steps are described next.

Detection: The safety assessment is based on certain assumptions for the planning, which should be monitored during the execution. Any change can threaten the validity of the plan. These changes are divided into three groups.

1) Changes in the geometric description of the road segment ($M^t \neq M^{t_0}$).
2) Changes in the position and geometry of static obstacles ($O^t \neq O^{t_0}$).
3) The deviation from the target condition of the reference trajectories for any involved participants ($p^t \notin rt_i$).

An example of the type of changes found in reality is a sudden appearance of an unexpected obstacle on the road and/or disregarding any deviation from the target condition of the reference trajectories.

Evaluation: The algorithm should evaluate all detected changes relating to these criteria:

1) Reaching the safe end-state / safe end-point (sTP)
2) Following the reference trajectory (RT)
3) Avoiding collisions

As mentioned before, the motion plan is presented in the form of reference trajectories (RTs) towards safe end points (sTPs) over a defined time horizon for each involved participant. From During et al. [4] it is known that the generation of RTs and sTPs is a complex process. However, the method proposed by the authors ensures conflict resolution if and only if the road users reach the sTPs via their reference trajectories. Therefore, during the execution it must be checked that each road user follows its reference trajectory and reaches its end-state. Otherwise there is a risk of entering dangerous states in the case of any deviation from the reference trajectory and secondary accidents when there is a deviation from a sTP.

Obviously, reaching a point with an exact heading and velocity is not a feasible task in reality. Thus a region must be defined around these points. This would lead to a set of four represent values, which could be geometrically connected to a square. Within this shape, the real (or exact) values could be found. For the sake of simplicity, we consider the end state as a single point only. This does not affect the presented methodology. Additionally, the assessment evaluates the detected changes in the environment and the deviation from the target condition of the reference trajectories by applying a collision check to the trajectory combinations. The collision check ensures that the proposed plans remain safe. However, not all deviations from the target condition of the reference trajectories and not all changes in the environment should be rated equally harshly.

Reaction: Finally, based on the evaluation of the detected changes in the plan and the environment, the algorithm will decide on an appropriate reaction. Possible reactions can be divided into three different modes: plan continuation, plan adaptation and plan cancellation. These will be detailed later on.

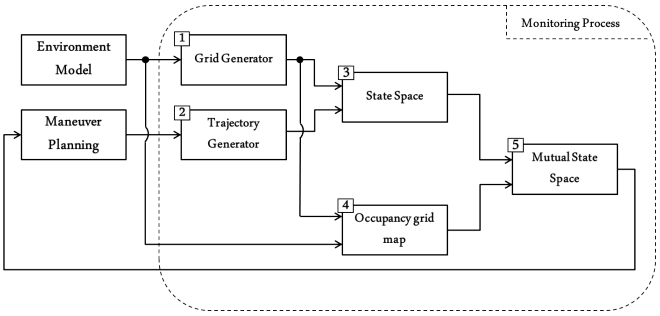


Fig. 2. Overview of the Monitoring Algorithm.

B. Monitoring Algorithm

The proposed monitoring algorithm for cooperative maneuvering solves the issues described in the problem statement with reference to the presented concept in five steps (Fig. 2). Firstly, a grid map is used as a quantized representation of the environment. Secondly, a trajectory generator calculates a large set of trajectories connecting the SP and the sTP for each cooperative road user. Thirdly, an algorithm generates state spaces based on the calculated set of trajectories and the grid map for each cooperative road user. Fourthly, in order to provide a secure guidance within a dynamic environment, an occupancy grid map representing the obstacles in the environment is continuously updated. Finally, the mutual state space combines the state spaces and the occupancy grid map with associated reference trajectories of all participants and performs a collision check. Steps four and five are executed at each time step while the first three steps are only performed once at the beginning. The monitoring algorithm evaluates the execution on the basis of the updated mutual state space in each time step over the defined time horizon of the execution. The five above mentioned modules are described in detail next.

Grid Generation

We use a simple grid map in order to describe the geometry of the relevant road segment. It is assumed, that the vehicles move in a two-dimensional environment. The grid generator creates a rectangular grid, using an arbitrary number of steps m_x , n_y in longitudinal and lateral direction. It presents the environment as an array of cells with length d_x and width d_y. The cell sizes can be freely selected.

The grid map (M) consists of $m \times n$ individual grid cells $u_{i,j}$. The grid cell $u_{i,j}$ is calculated as:

$$M = \begin{pmatrix} u_{1,1} & u_{1,2} & \cdots & u_{1,n} \\ u_{2,1} & u_{2,2} & \cdots & u_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ u_{m,1} & u_{m,2} & \cdots & u_{m,n} \end{pmatrix},$$

$$u_{i,j} = \{(x,y) \in R^2 | \forall_{x,y}((i_{-1})d_x \leq x < i.d_x) \wedge ((j_{-1})d_y \leq y < i.d_y)\}, \quad (1)$$

under the condition of $1 \leq i \leq m$ and $1 \leq j \leq n$ with $m = x_{max}$ and $n = y_{max}$.

Trajectory Generation

This module calculates trajectories, connecting the SP with the sTP. For cooperative road users, this might lead to sets of trajectories. The trajectories fulfill physical and infrastructural requirements as well as legal constraints. There are already several approaches for trajectory generation, which can be found in the literature. For example, based on tree search algorithms [12], elastic bands [6], and force fields [10]. However, each of these approaches has the objective to find one optimal solution, rather than many possible trajectories. For the purposes of this work, the trajectory generation algorithm of [4] is employed. This algorithm produces a large set of possible trajectories by using a combination of several different trajectories. The trajectory generation is composed of three steps. Firstly, a grid generation process provides a grid covering the road segment. The grid generation allows for a systematic variation of possible paths. Secondly, the path planning process calculates different kinetically feasible paths on the grid that connect the SP and the sTP via an arbitrary amount of grid points (GPs). The number of used GPs may vary, resulting in paths that consist of one or more segments. Each segment employs a motion primitive with a polynomial of fifth order representation. Thirdly, after obtaining the paths, the trajectory planning process completes the trajectories by calculating velocity profiles for each path.

A trajectory $tr_i \in ST_k$ is defined as equally spaced points p_b between SP and sTP over the defined time horizon (T).

$$\begin{aligned} ST_k &= \{tr_1, tr_2, ..., tr_q\}, \\ tr_u &= \{p_1, p_2, ..., p_z\}, \\ p_l &= (u, x, y, v, \psi, t). \end{aligned} \tag{2}$$

State Space Generation

In this step, the state space is calculated separately for each involved participant by combining the grid map and the calculated set of trajectories from the previous two steps. The state space (H) is a space in which all possible states of a system are described by a set of configuration spaces that a given problem and its environment could achieve. The grid map M is used as the basis coordinate system. The calculated set of trajectories ST_k describes all possibility actions from SP with the aim of reaching the sTP over a defined time horizon T for a vehicle k. The state space (H_k) has exactly the same size as the gird map (M) and consists of $m \times n$ configuration spaces. The configuration space $c_{i,j}$ is calculated as:

$$H_k = (M^{t_0}, ST_k^{t_0}),$$

$$H_k = \begin{pmatrix} c_{1,1} & c_{1,2} & \cdots & c_{1,n} \\ c_{2,1} & c_{2,2} & \cdots & c_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ c_{m,1} & c_{m,2} & \cdots & c_{m,n} \end{pmatrix},$$

$$c_{i,j} = \{(u, v, \psi, t) \in R^4 | \forall_{p_l} (p_l \in ST_k \wedge (x, y) \in u_{i,j})\}, \tag{3}$$

under the condition of $1 \leq i \leq m$ and $1 \leq j \leq n$ with $m = x_{max}$ and $n = y_{max}$.

As we have mentioned above, each trajectory $tr_u \in ST_k$ is defined as a set of points $p_l = (u, x, y, v, \psi, t)$. This means that each point in configuration space $c_{i,j}$ (Fig. 3) belongs to a single trajectory u and has a certain speed v, yaw angle (heading) ψ and time t. The set of points within each configuration space $c_{i,j}$ determine the extent of that cell. This can be visualized using a properties table, see Table I. Using this table one can calculate for each individual grid cell ($c_{i,j}$) the total number of trajectories (CT) passing though it and also the boundery conditions (b) for the three new dimensions ($V_{c_{i,j}}, \Psi_{c_{i,j}}, T_{c_{i,j}}$). This multi-dimensional boundary condition describes the limitations of the allowable actions within each configuration space $c_{i,j}$.This is calculated as:

$$\begin{aligned} CT_{c_{i,j}} &= |tr_u| = \{uniqe(u) \in N | \forall u \in c_{i,j}\}, \\ b_{c_{i,j}} &= \{V_{c_{i,j}}, \Psi_{c_{i,j}}, T_{c_{i,j}}\}, \\ V_{c_{i,j}} &= [min(v)_{c_{i,j}}, max(v)_{c_{i,j}}], \\ \Psi_{c_{i,j}} &= [min(\psi)_{c_{i,j}}, max(\psi)_{c_{i,j}}], \\ T_{c_{i,j}} &= [min(t)_{c_{i,j}}, max(t)_{c_{i,j}}]. \end{aligned} \tag{4}$$

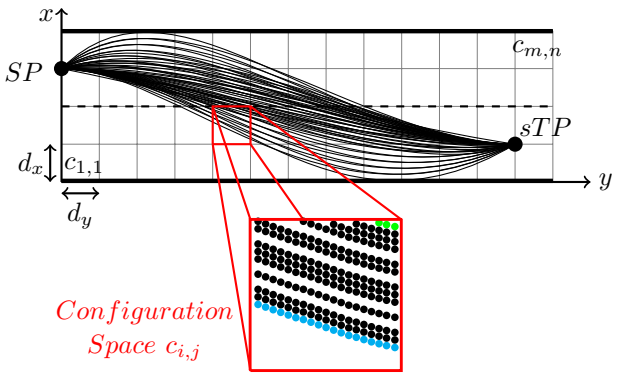


Fig. 3. State Space H_k by combining the grid map (M) and the calculated set of trajectories (ST_k).

TABLE I
CONFIGURATION SPACE $c_{i,j}$ FOR THE STATE SPACE H_k

u	v $[m/s]$	ψ $[^\circ]$	t $[s]$
122	20.00	5.02	2.21
122	20.05	5.01	2.22
122	20.10	5.00	2.23
⋮	⋮	⋮	⋮
135	21.40	10.00	2.03
⋮	⋮	⋮	⋮
135	21.95	9.17	2.22
135	22.00	9.16	2.23
$CT_{c_{i,j}} = 13$	$V_{c_{i,j}}$	$\Psi_{c_{i,j}}$	$T_{c_{i,j}}$

The state space (H_k) shows which modifications and deviations are allowed in each configuration space ($c_{i,j}$), while still being able to reach the sTP with defined configuration (u, x, y, v, ψ, t). This means that, if the vehicle (k) is positioned at $u_{i,j}$ during the execution but the current values of (v, ψ, t) are not within the boundary $b_{c_{i,j}}$, then the vehicle will not be able to reach the sTP.

The three steps (grid generation, trajectory generation and state space generation) described thus far, are calculated only once at the beginning of the process for each involved participant. As a result, a database of permitted actions is available for the evaluating of the plan execution.

Occupancy Grid Mapping

In order to provide a secure navigation within a dynamic environment and newly acquired sensor data of the environment, the occupancy grid map describes regularly the environment as an evenly spaced field of binary variables each representing the existence of an obstacle at that location in the environment. We use the same grid map to describe the position and geometry of static and dynamic obstacles (O) in relation to the geometry of the road segment (M). It is assumed, that the vehicles move in a two-dimensional environment. At any one time t the acquired state of the environment is presented as a tuple:

$$E^{(t)} = (M^{(t)}, O^{(t)}),$$

$$E^t = \begin{pmatrix} e_{1,1} & e_{1,2} & \cdots & e_{1,n} \\ e_{2,1} & e_{2,2} & \cdots & e_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ e_{m,1} & e_{m,2} & \cdots & e_{m,n} \end{pmatrix},$$

$$e_{i,j} = \{1 | \forall_{(x,y)} O^{(t)} \cap u_{i,j} \neq \emptyset\}. \tag{5}$$

including the occupancy map of the environment as an array of cells $M^t : \Re^2 \rightarrow (0, 1)$ with the set of detected obstacle $O^{(t)} = \{o_1, o_2, ..., o_n\}$. An object $o_i \in O^t$ is defined as:

$$o_i^{(t)} = \left\{l, s, \dot{rt}\right\}, \tag{6}$$

where $l \in \Re^2$ is the center, $s = (l, w)$ is the length (l) and width (w) and $\dot{rt}$ shows the associated reference trajectory of the object. Note that there is no reference trajectory for static obstacles.

Mutual state space

The Mutual state space ($S_k^{(t)}$) combines the state spaces (H_k) and occupancy grid map ($E_k^{(t)}$) for the involved participant (k) under consideration of all the obstacles (O) and performs collision checking.

$$S_k^{(t)} = (H_k, E_k^{(t)}) \tag{7}$$

It starts by applying a collision check between the calculated set of trajectories (ST_k) in state spaces H_k and all static obstacles as well as the associated reference trajectories of the dynamic obstacles (O and $\dot{rt}$) in occupancy grid map $E_k^{(t)}$ at time step $t_0 = 0$ (Fig. 4). All collision-afflicted trajectories will be marked as colliding trajectories. The remaining trajectories describe the extant valid actions for reaching the sTP. This will result in new boundary conditions $b_{c_{i,j}}$ for each individual grid cell $c_{i,j}$. The new boundary conditions $b_{c_{i,j}}$ are based on the consideration of the environment and reference trajectories for involved participants at the beginning of the maneuver execution (Table II).

Using the boundary conditions, we define the degrees of freedom F_k^t with $f_{i,j}$ for each individual grid cell $c_{i,j}$. The degrees of freedom $f_{i,j} \in F_k^t$ describe the value of the selection for the possible allowable actions within each individual grid cell $c_{i,j}$. This is calculated as:

$$F_k^t = \begin{pmatrix} f_{1,1} & f_{1,2} & \cdots & f_{1,n} \\ f_{2,1} & f_{2,2} & \cdots & f_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ f_{m,1} & f_{m,2} & \cdots & f_{m,n} \end{pmatrix},$$

$$f_{c_{i,j}} = \alpha(\Delta V_{c_{i,j}}/v_{no}) + \beta(\Delta \Psi_{c_{i,j}}/\psi_{no}) + \gamma(\Delta T_{c_{i,j}}/t_{no}) \tag{8}$$

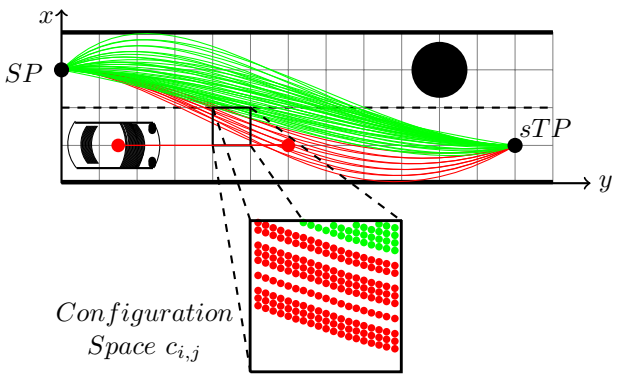


Fig. 4. Mutual state space by combining the state spaces and occupancy grid map. All collision-afflicted trajectories shown in red and all collision-free trajectories in green.

TABLE II
CONFIGURATION SPACE $c_{i,j}$ FOR THE MUTUAL STATE SPACE (ST_k)

u	v $[m/s]$	ψ $[°]$	t $[s]$
122	20.00	5.02	2.21
122	20.05	5.01	2.22
122	20.10	5.00	2.23
⋮	⋮	⋮	⋮
~~135~~	~~20.40~~	~~10.00~~	~~2.03~~
⋮	⋮	⋮	⋮
~~135~~	~~21.95~~	~~9.17~~	~~2.22~~
~~135~~	~~22.00~~	~~9.16~~	~~2.23~~
$CT_{c_{i,j}} = 4$	$V_{c_{i,j}}$	$\Psi_{c_{i,j}}$	$T_{c_{i,j}}$

Fig. 4 shows an example situation, where the ego-vehicle at SP intends to change lanes to avoid collision with an obstacle $o_i \in O^t$. The other vehicle is in the adjacent lane and is driving beside the ego-vehicle. Therefore, it limits the possible solutions (trajectories) for the ego-vehicle to reach the sTP over the defined time horizon (T). The distribution of the F_k^t for the ego-vehicle at the beginning of the maneuver execution $t = 0$ is shown in Fig. 5.

In a similar way, any change in the plan and the environment at time step (t) during the execution influences the boundary conditions and thus the degrees of freedom F_k^t for vehicle (k). Based on change in F_k^t one can detect and evaluate any changes in the position and geometry of the static obstacles and deviations from the target condition of the reference trajectories for all involved participants during the execution.

This means that, if the vehicle (k) at time step (t) is positioned at grid cell $u_{i,j}$ but the current value of the degrees of freedom $f_{i,j} \in F_k^t$ is smaller than calculated $f_{i,j} \in F_k^0$ for grid cell $u_{i,j}$ at the beginning of the maneuver execution $t = 0$, there were negative deviations from the plan up to the time point of the measurement. However, it can also happen that the current value of $f_{i,j} \in F_k^t$ is greater than calculated $f_{i,j} \in F_k^0$. This could be the case if, for example in Fig. 4, the vehicle driving in the adjacent lane beside the ego-vehicle has been deviating from its reference trajectory in the direction away from the center to the right (the y axis).

Eventually, based on the current value of the degrees of freedom $f_{i,j} \in F_k^t$ presented as $(f_{i,j})_k^t$ at each time step t, the algorithm will decide for an appropriate reaction for the vehicle (k). Possible reactions are divided into three modes: Plan continuation, Plan adaptation and Plan cancellation. This decision is based on two thresholds: ζ_{Cont} for Plan continuation and ζ_{Can} for cancellation. The plan is adapted if the value of $(f_{i,j})_k^t$ is between the upper and lower limits:

$$\begin{aligned} &\zeta_{Cont}(f_{i,j})_k^0 < (f_{i,j})_k^t \\ &\zeta_{Can}(f_{i,j})_k^0 \leq (f_{i,j})_k^t \leq \zeta_{Cont}(f_{i,j})_k^0 \\ &(f_{i,j})_k^t < \zeta_{Can}(f_{i,j})_k^0 \end{aligned} \tag{9}$$

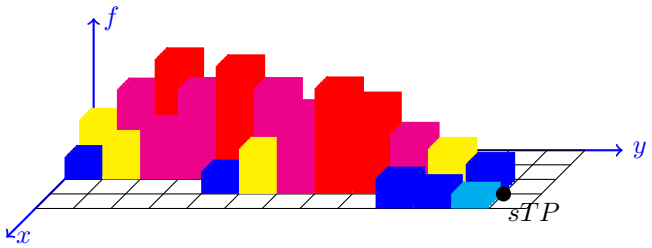


Fig. 5. The distribution of the degrees of freedom F_k^t for the ego-vehicle at the beginning of the maneuver execution $t = 0$

IV. Results

In order to prove the usability of the proposed concept, we present some simulation results. We are currently also developing a HIL framework to simulate our algorithm which further enables us to easily transfer it to our fleet of three VW Golf VI vehicles for real world experiments.

We define a dangerous overtaking test scenario on a country road comprising three vehicles. For this simulation, we assume all road users to be cooperative. Fig. 1 as well as Fig. 6a illustrate the scenario. Vehicle 1 is driving on the opposing lane overtaking the slower driving vehicle 2. Vehicle 3 has broken down in its own lane and appears on collision course. The initial location, velocities and dynamic parameters of the vehicles is specified in Table III. The centers of the two colliding vehicles are 30 m apart, resulting in a TTC =1200 ms. Each lane is 3.5 m wide and has a friction coefficient μ of 1.

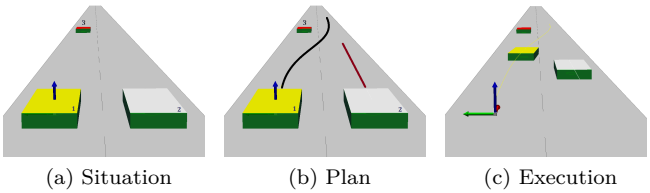
Fig. 6b shows the best trajectory combination towards safe end points sTPs for all vehicles, calculated by the cooperative maneuver planning algorithm [4]. From Table III we observe that the duration of the maneuver is 3.0 sec. The speed profiles in Figs. 6d-f illustrate the speed of the three vehicles within this plan.

The presented monitoring algorithm for cooperative maneuvering starts by applying the five steps at the beginning of the maneuver execution $t = 0$.

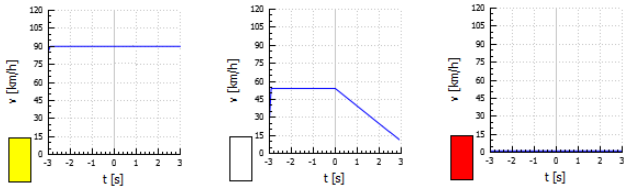
TABLE III

Start and end parameters for the scenario

Parameter	Vehicle 1	Vehicle 2	Vehicle 3
Pos. of Center	(0; 1.87)	(0; −1.87)	(30; 1.87)
Length/width	3.8/1.8	3.8/1.8	3.8 m/1.8
Speed	$25\frac{m}{s}$	$15\frac{m}{s}$	$0\frac{m}{s}$
$a_{lon-max}$	$6\frac{m}{s^2}$	$6\frac{m}{s^2}$	$0\frac{m}{s^2}$
$a_{lon-min}$	$8\frac{m}{s^2}$	$8\frac{m}{s^2}$	$0\frac{m}{s^2}$
$a_{lat-max}$	$5\frac{m}{s^2}$	$5\frac{m}{s^2}$	$0\frac{m}{s^2}$
$sTP(x, y)$	$(75, -1.87)$	$(27, -1.87)$	$(30, 1.87)$
$sTP(v, \psi, t)$	$(25, 0, 3)$	$(3, 0, 3)$	$(0, 0, 3)$

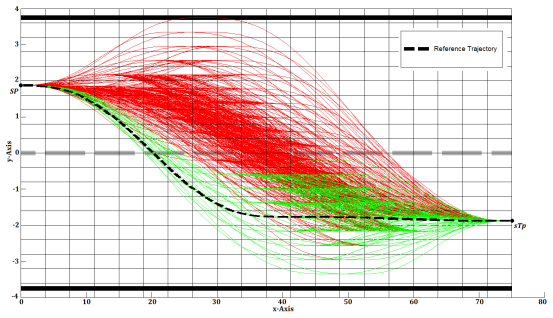


(a) Situation (b) Plan (c) Execution

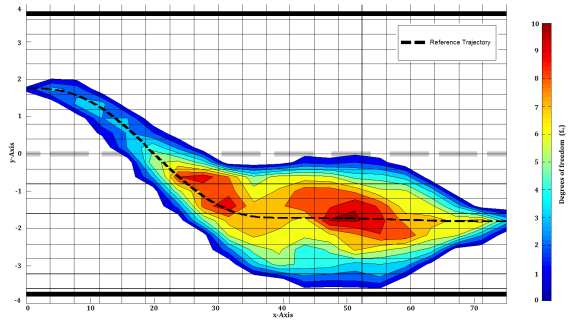


(d) Vehicle 1 (e) Vehicle 2 (f) Vehicle 3

Fig. 6. (a)(b)(c) Snapshots of the scenario and (d)(e)(f) The speed profiles of the three vehicles within this plan



(a) $S_k^{(t)}$



(b) F_k^t

Fig. 7. (a) Mutual state space $S_k^{(t)}$ and (b) Distribution of the F_k^t for the ego-vehicle $k = 1$ at the beginning of the maneuver execution $t = 0$

The grid map (M) consists of 20×20 grid cells $u_{i,j}$ with length $d_x = 3.75$ m and width $d_y = 0.4$ m. A total number of 4050 trajectories ST_k^t are computed between SP and sTP, which satisfy the kinematic, dynamic, and time horizon $T = 3$ s requirements. The state space H_k is generated by combining (M) and ST_k^t. The occupancy grid map $E^{(t)}$ is updated with the set of detected obstacles $O^{(t)} = \{vehicle(2), vehicle(3)\}$ from the point of view of the ego-vehicle. The mutual state space $S_k^{(t)}$ is generated by applying a collision check between ST_k^t in state spaces H_k and all static obstacles (in this case: vehicle 3). Furthermore, it considers the associated reference trajectories of the dynamic obstacles (in this case: vehicle 2) in occupancy grid map $E_k^{(t)}$. All collision-afflicted trajectories (3091 trajectories shown in red) are marked as colliding trajectories. The remaining trajectories (959 shown in green) describe the extent of valid actions for reaching sTP (Fig 7a). Using the mutual state space $S_k^{(t)}$, we calculate the degrees of freedom F_k^t with $f_{i,j}$ for each individual grid cell $c_{i,j}$. The distribution of the F_k^t for the ego-vehicle at the beginning of the maneuver execution $t = 0$ is shown in Fig 7b.

The reference trajectory from the maneauver planning algorithm [4] is highlighted in Fig. 7. As we traverse this reference trajectory, within each cell of F_k^t, we can observe the available degrees of freedom (Fig. 7b).

By comparing the degrees of freedom $(f_{i,j})_k^t$ at time $t = i$ during execution with time $t = 0$, the algorithm decides on an appropriate reaction (see Sec. III-B). This

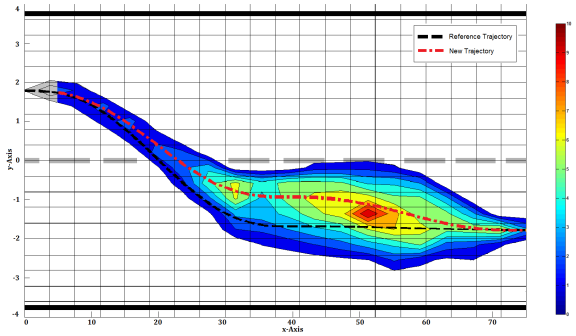


Fig. 8. Distribution of the F_k^t after t =0.3 s, with the newly calculated motion plan

decision is based on the two thresholds $\zeta_{Cont} = 0.66$ and $\zeta_{Can} = 0.33$ (Eq. (9)). In our test scenario, we are going to assume that Vehicle 2 is not cooperative. Instead of following the provided reference motion plan, the vehicle does not decelerate but continues with the initial speed $v_0 = 15\ \frac{m}{s}$. The monitoring algorithm notes after 0.3 s that the differences between the current value of the degrees of freedom $(f_{i,j})_k^{0.3} = 1.87$ compared to $(f_{i,j})_k^0 = 2.88$ is now smaller than the defined threshold ζ_{Cont} $(\frac{1.87}{2.88} < 0.66)$. Given that the ratio is between ζ_{Cont} and ζ_{Can}, the algorithm decides for an "Adaptation plan". Fig. 8 shows the degrees of freedom at t =0.3 s, with the newly calculated motion plan overlayed in red while the old motion plan is shown in black. We notice that the available degrees of freedom have been considerably limited and the new trajectory is pushed upwards avoiding a collision with vehicle 2. In summary, the possibility of colliding with other vehicles is reduced, because the new trajectory passes through cells with higher degrees of freedom than the reference trajectory.

V. Conclusion and Outlook

An algorithm for monitoring the execution of a cooperative maneuver planning is presented. The algorithm solves a conflict by generating a state space for the possible maneuvers for all traffic participants, tracking the progress of their maneuvers, and responds to any deviation ensuring a safe conflict resolution.

The algorithm evaluates the execution of the plan by updating the mutual state space at each time step over the maneuver horizon. It allows detecting and estimating hazards with reference to the mutual interaction between vehicles and can be employed in diverse traffic situations.

Further work will focus on optimizing computation time to meet real-time on-board processing requirements as well as an enhanced state space generator and faster combination algorithms for mutual state space.

In this work, all participants were considered to have V2V communication available. Future work will consider mixed traffic situations using a prediction model determining the most likely target point and trajectory for the non-cooperative road user(s).

Acknowledgement

The authors would like to thank their colleagues at Volkswagen Group Research, in particular Dr. Navid Nourani-Vatani for the constructive discussions.

References

[1] C. Adam et al. "Probabilistic Road Estimation and Lane Association Using Radar Detections". In: *14th International Conference on Information Fusion Chicago, USA*. 2011.

[2] A. Pedro Aguiar and Antonio M. Pascoal. "Coordinated Path-Following and Control for Nonlinear and Systems with and Logic-Based Communication". In: *IEEE Conference on Decision and Control New Orleans, USA*. IEEE, 2007.

[3] A. Broggi et al. "Autonomous vehicles control in the VisLab Intercontinental Autonomous Challenge". In: *Annual Reviews in Control 36.1*. 2012.

[4] M. During et al. "Adaptive Cooperative Maneuver Planning Algorithm for Conflict Resolution in Diverse Traffic Situations". In: *International Conference on Connected Vehicles & Expo, IEEE*. 2014.

[5] K. Franke et al. "A Reference Architecture for CISS/CDAS within the Field of Cooperative Driving". In: *International Conference on Connected Vehicles & Expo, IEEE*. 2014.

[6] C. Frese and J. Beyerer. "A comparison of motion planning algorithms for cooperative collision avoidance of multiple cognitive automobiles". In: *Intelligent Vehicles Symposium (IV) IEEE*. 2011.

[7] A. Geiger et al. *Team AnnieWAY's entry to the Grand Cooperative Driving Challenge 2011*. Tech. rep. IEEE Transactions on Intelligent Transportation Systems, 2011.

[8] R. Ghabcheloo et al. "Coordinated Path and Following Control and of Multiple and Wheeled Robots and with and Directed Communication and Links". In: *Decision and European Control Conference 44th*. IEEE, 2005.

[9] Y. Hattori, Eiichi Ono, and Shigeyuki Hosoe. "An Optimum and Vehicle Trajectory and Control for and Obstacle Avoidance and with the Shortest and Longitudinal". In: *International Conference on Mechatronics and Automation*. IEEE, 2008.

[10] S. LaValle. "Planning Algorithms". In: *Cambridge University Press*. 2006.

[11] S. Lefevre, Christian Laugier, and Javier Ibanez-Guzman. "Evaluating Risk at Road Intersections by Detecting Conflicting Intentions". In: *RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2012.

[12] L. Li and F. Wang. "Cooperative driving at adjacent blind intersections". In: *IEEE International Conference on Systems, Man and Cybernetics*. 2005.

[13] K. Macek, Roland Philippsen, and Roland Siegwart. "Path Following and for Autonomous and Vehicle Navigation and Based on and Kinodynamic Control". In: *Journal of Computing and Information Technology*. 2009.

[14] G. Mitropoulos and Irene S. Karanasiou. *Wireless Local and Danger Warning and Cooperative and Foresighted Driving and Using*. Tech. rep. IEEE Transactions on Intelligent Transportation Systems, 2010.

[15] S. Oh and Younguk Yim. "Modeling of Vehicle Dynamics from Real Vehicle Measurements Using a Neural Network with Two-Stage Hybrid Learning". In: *Transactions on Vehicular Technology*. IEEE, 1999.

[16] A. Pentland and Andrew Liu. "Modeling and Prediction and of Human and Behavior". In: *LETTER Communicated by Dean Pomerleau, Massachusetts Institute of Technology*. 1999.

[17] D. Polling et al. "Inferring the driver's lane change intention using context-based dynamic Bayesian networks". In: *Systems, Man and Cybernetics*. IEEE, 2005. DOI: ng..

[18] M. Rockl, Gacnik, and Schomerus. "Integration of Car-2-Car communication as a virtual sensor in Automotive sensor fusion for ADAS". In: 2008.

[19] J. Schneider, A. Wilde, and K. Naab. *Probabilistic Approach and for Modeling and Identifying Driving and Situations*. Tech. rep. IEEE Intelligent Vehicles Symposium Eindhoven University of Technology, 2008.

[20] L. Smadja and J. Ninotand T. Gavrilovic. "Road extraction and environment integration from lidar sensor". In: *Paparoditis N., Pierrot-Deseilligny M., Mallet C., France*. 2010.

[21] R. Solea and U. Nunes. *Trajectory Planning and Sliding-Mode Control Based Trajectory-Tracking for Cybercars*. Tech. rep. Integrated Computer-aided Engineering, Int. Journal, 2007.

[22] R. Tomar and S. Verma. "Safety of Lane Change Maneuver Through A Priori Prediction of Trajectory Using Neural Networks". In: *Network Protocols and Algorithms*. 2012.

[23] Y. Wang et al. "Vehicle Collision Warning System and Collision Detection Algorithm Based on Vehicle Infrastructure Integration". In: *The Seventh Advanced Forum on Transportation of China*. 2011.

[24] W. Xiao, D. Xie, and T. Tan. "Traffic Accident Prediction Using Vehicle Tracking and Trajectory Analysis". In: *Intelligent Transportation Systems, IEEE*. 2003.

[25] Y. Zhang and H. Mehrjerdi. *A Survey on Multiple Unmanned Vehicles Formation Control and Coordination: Normal and Fault Situations*. Tech. rep. Unmanned Aircraft Systems (ICUAS), 2013.