Safe Cooperation of Automated Vehicles

Heß, Daniel; Löper, Christian; Hesse, Tobias

DLR / Institute of Transportation Systems

Lilienthalplatz 7, 38108 Braunschweig, +49 531 295 3590,
Daniel.Hess@DLR.de

Abstract

Despite the rapid progress in the development of automated vehicles, formal verification of the full automated driving process is unsolved. A promising approach investigated in the EU project UnCoVerCPS is the combination of online and offline verification as well as testing steps. The methods developed in UnCoVerCPS are applicable to many safety critical, cyber physical systems. As a specific use case, we investigate a system, which facilitates safe cooperation of automated vehicles. Leveraging a formal proof on a validated vehicle dynamics model and by negotiating exclusive access to subsets of the drivable area via Car-to-Car communication, the freedom of collisions and safe operation in other respects are ascertained online and during operation of the vehicle. One of the goals of UnCoVerCPS is to demonstrate applicability of the online verification approach in a cooperative driving scenario with two life-sized vehicles. In this paper we discuss the detailed approach and preliminary results en route to the physical realization.

1. Introduction

A significant challenge in automated vehicle design is validation and verification. Purely test-based validation approaches tend to require high numbers of test kilometers, which can be cost and time intensive. Under assumption of a stochastic model, a recent study estimates that showing with 95% confidence a fatality rate of automated cars within 20% of the fatality rate of human operated cars would require 8.8 billion miles of driving [1]. Furthermore, it is difficult to transfer results to unforeseen driving situations or new software configurations.

Results on formal correctness proves exist, but often consider very limited application scopes or very abstract problem formulations. An approach for safely entering an intersection is presented in [2]. In [3],

Appeared in: ITS automotive nord e.V. (Hrsg.) AAET - Automatisiertes und vernetztes Fahren, Beiträge zum gleichnamigen 18. Braunschweiger Symposium vom 8. und 9. Februar 2017.
ISBN: 978-3-937655-41-3.

automated cruise control is formally verified by automated theorem proving, but under the assumptions that all vehicles cooperate and communicate, and that no unexpected vehicle appears. A verified synthesis for driving assistance in traffic merging is presented in [4], but with limited consideration of the underlying dynamics required for fully automated driving. Offline verification approaches have the problem that they have to account for all values of a high number of environmental variables inherent in the application, which is especially difficult in the combination of mixed continuous and discrete dynamical systems. The number of variables can be drastically reduced, if a proof of correctness is attempted for only a specific instance of traffic situations – this though requires repeated evaluation for each decision, on-the-fly and during operation of the vehicle [5].

We present an approach that uses a combination of offline- and online computation steps to verify on-the-fly that a certain control action of a vehicle is safe to execute. The pre-computation of verified motion primitives in an offline step allows replacing time intensive online computations with simple look-ups. Safety is proven by calculating an emergency maneuver, which takes the vehicle to a safe state after execution of the control action, without intersecting worst case predictions of other vehicles. Using worst case predictions for the behavior of other traffic participants is in some situations conservative, yet it avoids the complexity of game theory or differential games and it allows using very general assumptions about other traffic participants. The presented approach is formulated especially for cooperative lane-changing situations and makes no unrealistic assumptions on communication protocols. Changing the constraint sets used for emergency maneuver planning allows to easily extend the approach to other automated driving applications or vehicle capabilities. The proposed cooperation scheme considers only direct cooperation in a 1 to n relationship. The approach is intended as a safe basis for more complex, group-based interactions, as proposed in [6] and further pursued in the DFG project CoInCiDE. The next section gives an overview of our approach and the following sections detail the software modules involved in realization.

2. Approach

We propose a supervisory module, which enforces safety of cooperative automated driving by filtering the communication exchange between vehicles, as well as filtering the driving commands sent to each vehicle's actuators.

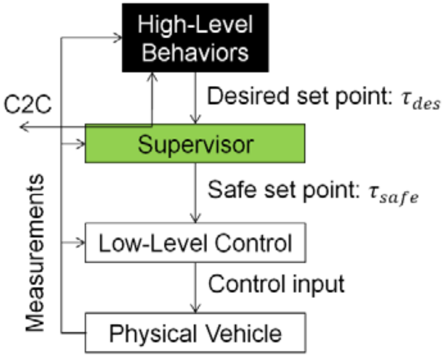

Fig. 1: Layered vehicle control architecture with Supervisor

We make use of a typical, layered architecture, fig. 1, structured analogously to the driving tasks defined by Donges [7]. At the lowest level, the physical vehicle accepts control inputs from a stabilizing control layer. Higher-level driving skills on the short term guidance / maneuvering layer as well as the navigation layer are subsumed in a "black box" termed High-level Behaviors. The contained modules are thought to handle normal, every-day driving situations by providing nominal set-points, which lead to a good average-case driving performance by such measures as duration of travel, fuel consumption, comfort, etc. We will not focus on the inner workings of the High-level Behaviors, but rather assume them to be given according to the state of the art. The black box of High-level behaviors is *expected, but not required* to provide new, desired set points τ_{des} with a fixed rate $1/T_p$, (we use $T_p = 0.1s$). In our specific application, a set point τ_{des} is a reference trajectory of duration T_p. The Low-level Controller executes at a fixed rate $1/T_c$ (e.g. here using $T_c = 0.005s$) and stabilizes the vehicle, by regulating the vehicle's deviation from the set point (reference trajectory). The Low-level Controller is assumed to be a "white box" module with known equations, as it is submitted to formal analysis in order to compute bounds on the closed-loop performance of the system, see sec. 3 and sec. 4.

In each vehicle, which is following the presented approach, a Supervisor module is inserted between High-level Behaviors and

Low-level Controller. Every time frame of length T_p, the Supervisor tries to find a proof of safety for the current desired set point. If successful, it passes the desired set point to the Low-Level Controller. This is defined as the nominal operation. Otherwise it supplies a surrogate - an emergency set point, for which a proof of safety is known. This is henceforth referred to as emergency operation.

To facilitate cooperation between automated vehicles, we assume that information is passed between the vehicles' respective High-level Behaviors via wireless car-to-car messages. A Car-to-car communication module (C2C) realizes the transmission. We require the C2C module to guarantee that messages are neither altered nor faked (e.g. by employing asymmetric encryption). Albeit, a guarantee for successful or timely transmission is *not* required, as such a performance could not be reliably provided by contemporary C2C protocols such as the ETSI GeoNetworking protocol [8]. Agreements between vehicles are allowed to be safety critical. Therefore the message content has to be standardized and known to the Supervisor and all safety critical messages have to be passed through the Supervisor. The Supervisor keeps track of incoming messages and transfers outgoing messages only, if it is able to show that the contained agreement is safe.

2.1 The Supervisor Module

According to the proposed concept, the Supervisor proves safety of a desired set point τ_{des} by constructing an emergency maneuver, which can be appended after execution of τ_{des} and which guides the vehicle to a safe terminal state, (e.g. stand still in a specific lane or emergency lane). The central part of the Supervisor, fig. 2, is a Planner, which constructs the emergency maneuvers as a concatenation of atomic motion primitives supplied by a Maneuver Database. The Maneuver Database specifies: (1) A finite set of motion primitives and their according set points (2) A guaranteed over-approximation of the X, Y, t occupancy of each motion primitive, meaning the set of positions in X, Y, t, which could be covered by any part of the vehicle body during execution of the motion primitive. (3) A graph structure that defines admissible orders of execution of the motion primitives. (4) Entry conditions for starting an emergency maneuver. The Planner tests the motion primitives' X, Y, t occupancies against constraints supplied by the Constraints module and thus creates a tree of admissible emergency maneuvers.

Eventually it finds a valid chain of motion primitives, which starts with the nominal set point and leads to a safe state. The chain of reference trajectories is designated $\phi_i := (\tau_{des}^i \tau_{em,1}^i \tau_{em,2}^i \dots \tau_{em,k}^i)$.

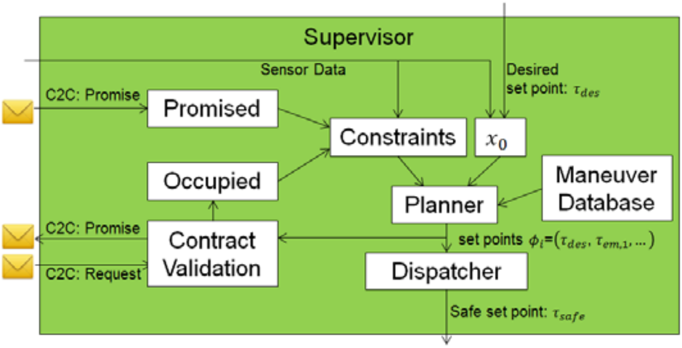

Fig. 2: Sub-components of the Supervisor

If such a solution ϕ_i is found, it is passed on to the Dispatcher, a sub-process of the Supervisor, which runs in parallel to the Planner. The Dispatcher oversees the correct timing of presenting a validated set point to the Low-level Control module. If ϕ_i is available on time, the Dispatcher will replace its previous ϕ_{i-1} with ϕ_i and send the new first entry τ_{des}^i to the Low-Level Control. This constitutes the case of normal operation: Each update, the old desired set point τ_{des}^i is directly succeeded by a new desired set point τ_{des}^{i+1}. The High-level Behavior's set points are therefore passed through the Supervisor module with a delay of T_p and execution of an emergency maneuver is regularly postponed. In contrast, the emergency case occurs, if the Dispatcher does not receive ϕ_i on time. This could be provoked for numerous reasons, such as a software or timing error in the High-level Behaviors, excessive complexity of the scene presented to the Motion Planner, a software error in the Motion Planner or in general an unsafe set point τ_{des}. In the case of such an emergency, the Dispatcher continues execution of the *previous* ϕ_{i-1}, consequently transitioning from the last desired set point τ_{des} to the first emergency set point $\tau_{em,1}$ and so forth. Accordingly the full ϕ_i will be executed until the vehicle has reached the safe terminal state, or until a new and safe desired set point is available.

The proof of correctness of the Supervisor module hinges on the set point switching of the Dispatcher: Assuming that at a certain time the Dispatcher is initialized with a first, valid emergency maneuver, one can show by induction that the vehicle will subsequently execute only such set points, which are leading to safe states, as witnessed by the according emergency maneuver.

Of course, one has to guarantee that an emergency maneuver, which is based on observations at the beginning of motion planning, will remain safe during execution of the emergency maneuver up to reaching its terminal state. Depending on the initial velocity and the number of required steering actions, environment predictions must be guaranteed to hold for an interval of several seconds. The Constraints module makes use of worst-case assumptions based on physical bounds and simple traffic rules to achieve reliable predictions, see sec. 5. These predictions are presented as X,Y,t-constraints to the Motion Planner. Furthermore, it has to account for safety critical agreements between cooperating vehicles: The observation-based X,Y,t-constraints are tightened by constraints from the Occupied set, which represents promises made by the ego-vehicle to other cooperating vehicles through an outgoing Promise message. In a symmetric fashion, Promise messages received by the ego-vehicle are stored in the Promised set: The Constraints module relaxes the X,Y,t-constraints using the information contained in the Promised set.

In our approach, vehicles are allowed to build their safety critical emergency maneuver based on information exchanged with C2C messages. Therefore Promise messages have to constitute a contract that may not be violated under any circumstances. While a High-level Behavior might decide that it is desirable (according to whichever metric) to answer a Request message with a Promise message, the Supervisor investigates inside the Contract Validation module whether a Promise is safe for the ego vehicle and whether it can be honored under all circumstances. Only then, and after saving the Promise information in the Occupied set, transmission is allowed, see sec. 5. Looking closer at fig. 2, it is apparent that the Supervisor does not handle outgoing Request messages. This is due to the fact that Requests are not safety critical: High-Level Behaviors are free to issue Requests in any manner that seems appropriate to them.

In order to create a reliable, deterministic interface to the vehicle dynamics two steps have to be taken during an offline design phase. First, a closed loop vehicle model with appropriate disturbance bounds has to be chosen. In sec. 3 the model is defined and the choice of bounds in relation to the physical vehicle is described. Based on the disturbed ego vehicle model, sec. 4 describes computation of a Maneuver Database, which provides deterministic action sets for nominal and emergency maneuvers. Online, during operation of the vehicle, a set of constraints is computed, which takes vehicle cooperation into account. The interaction and cooperation scheme is described in sec. 5. A maneuver planner, briefly covered in sec. 6, computes safe emergency maneuvers, which avoid intersection with the constraint set.

3. Vehicle Model with Error Bounds

The goal of this section is to provide a *validated* mathematical model for the vehicle movement. Given the nature of the physical system, the relation between system and mathematical model can only be established by a finite set of examples. Conformance Testing is employed to find and quantify the differences between model and system, based on a set of exemplary test drives.

We use a planar bicycle model to describe the vehicle motion. Tire forces are modeled as linear equations, yet the overall model is nonlinear in the longitudinal velocity and the kinematic equations. The vehicle state $x \epsilon \mathbb{R}^6$ is chosen as $x = \left(X, Y, \psi, v_x, v_y, \omega\right)^{\mathrm{T}}$, including the vehicle position, the orientation, the relative longitudinal and lateral velocity as well as the rotational speed. The input space is $u \epsilon \mathbb{R}^2$, $u = (a_x, \delta)^{\mathrm{T}}$ including the desired longitudinal acceleration and the steering angle. The parameter vector is $p = (\vartheta, c_F, c_R, a, b, \mu g)$ with $\vartheta = m/J$ the ratio of mass and rotational inertia, c_F, c_R the relative front and rear tire stiffness, a, b the distance from center of gravity to front and rear axle with $L = a + b$. In the following, the constants $k_F = -\mu g c_F b/L$ and $k_R = -\mu g c_R a/L$ are used. A disturbance acting on the vehicle is defined as $e_d = \left(e_{fx}, e_{fyF}, e_{fyR}\right)^{\mathrm{T}} \epsilon \mathbb{R}^3$, which contains three error forces divided by the vehicle mass, e_{fx} for combined longitudinal errors and e_{fy} and e_{fy} for front and rear lateral error terms.

Def. 3.1: The vehicle's differential equation is defined as $\dot{x} = f(x, u, e_d) = (f_1, \dots, f_6)^{\mathrm{T}}$ with:

$$f_1 = v_x \cos(\psi) - v_y \sin(\psi)\,, \qquad f_2 = v_x \sin(\psi) + v_y \cos(\psi), \tag{3.1}$$

$$f_3 = \omega, \qquad f_4 = a_x + e_{fx}, \tag{3.2}$$

$$f_5 = k_F \left(\frac{v_y + a\omega}{v_x} - \delta\right) + k_R \frac{v_y - b\omega}{v_x} - v_x\omega + e_{fyF} + e_{fyR}\,, \tag{3.3}$$

$$f_6 = a\vartheta k_F \left(\frac{v_y + a\omega}{v_x} - \delta\right) - b\vartheta k_R \frac{v_y - b\omega}{v_x} + a\vartheta e_{fyF} - b\vartheta e_{fyR}. \tag{3.4}$$

The parameter vector has been matched to our physical test vehicle, by minimizing the error between a test-drive recording (open loop) and the parametrized model. Fig. 3 shows a comparison between lateral forces estimated from the recording (blue) and the fitted relative tire stiffness (red). The parameters $\vartheta = 0.64\,m^{-2}, c_F = 10.8\,rad^{-1}, c_R = 17.8\,rad^{-1}, a = 1.16m,\ b = 1.54m, \mu g = 0.8 \cdot 9.81ms^{-2}$ are used.

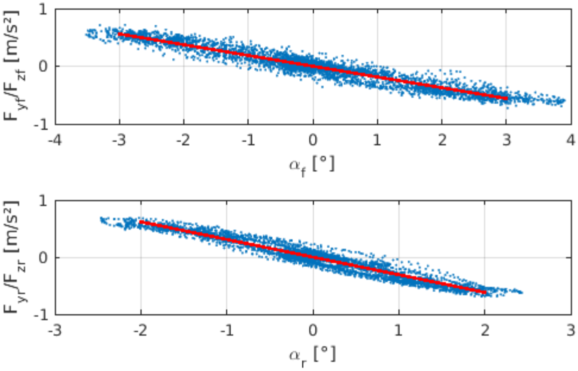

Fig. 3: Cornering stiffness (red) matched to test drive data (blue)

3.1 Low-Level Control

Def. 3.2: A reference trajectory $\tau: \mathbb{R} \to \mathbb{R}^{10}$ is defined as a solution to the initial value problem (IVP) for an error-free vehicle model ($e_d \equiv 0$) with the reference state vector $x^{ref} = \left(X, Y, \psi, v_x, v_y, \omega, a_x, \delta, w_1, w_2\right)^{\mathrm{T}}$ containing the reference input a_x, δ and the reference input change w_1, w_2 and with the differential equation $\dot{x}^{ref} = f^{ref}(x^{ref})$:

$$f_{ref} := (f^{\mathrm{T}}, w_1, w_2, 0{,}0)^{\mathrm{T}} \qquad (3.5)$$

Def. 3.3: The feedback control function makes use of a nonlinear transformation of a control point in front of the vehicle similar to [16]. The control point is located a distance $\lambda \in \mathbb{R}^2$ relative to the vehicle and a *reference* control point is located λ relative to the *reference* vehicle. The control error is defined as the difference between both control points, expressed in the local coordinates of the reference vehicle, using the rotation matrix $R(\psi) = \begin{pmatrix} \cos(\psi) & -\sin(\psi) \\ \sin(\psi) & \cos(\psi) \end{pmatrix}$:

$$e_{tn} := R\left(-x_3^{ref}\right) \cdot \left(x_{1:2} - x_{1:2}^{ref} + R(x_3) \cdot \lambda\right) - \lambda. \qquad (3.6)$$

Def. 3.4: Including a feed-forward and a linear PD feed-back term, the control function $u = c(x, x^{ref})$ is defined:

$$c \coloneqq x_{7:8}^{ref} - K\,(e_{tn}^{\mathrm{T}}, \dot{e}_{tn}^{\mathrm{T}})^{\mathrm{T}}. \tag{3.7}$$

Def. 3.5: The closed-loop system used for conformance testing and reachability analysis is defined as the differential equation f_c with the exogenous disturbance e_d and the measurement error $e_m \epsilon \mathbb{R}^6$:

$$f_c(t, x, e_m, e_d) = f\left(x, c\left(x + e_m, x^{ref}(t)\right), e_d\right) \tag{3.8}$$

3.2 Conformance Testing

The goal of conformance testing is to validate the vehicle dynamics model and to bound the size of measurement errors e_m and disturbance e_d that have to be expected during closed loop operation of the vehicle. Given a parametrized vehicle model f_c, the error sets E_m and E_d with $\forall t\colon e_m(t) \in E_m \wedge e_d(t) \in E_d$ and some recordings of test drives of the actual vehicle, the conformance testing step tries to invalidate the hypothesis that the combination $\{f_c, E_m, E_d\}$ suffices to explain the relevant physical processes. If the hypothesis is falsified, $\{f_c, E_m, E_d\}$ has to be adapted in the quest for a reliable model. If the hypothesis cannot be falsified, a certain degree of confidence in the model is provided, according to the test cases' density of coverage of the operational regime of the vehicle. Roehm et. al. [9] describe different types of conformance relations between model and physical process. Here, we make use of trace conformance: Given a measurement trace $Y \in \mathbb{R}^{n\times k}$ containing measurements $y \in \mathbb{R}^n$ at times $T \in \mathbb{R}^k$, the model is conformant, if traces $\varepsilon_m \in E_m^k$ and $\varepsilon_d \in E_d^k$ exist, for which holds $\forall i < k$:

$$\left(Y_{i+1} - \varepsilon_{m,i+1}\right) = \left(Y_i - \varepsilon_{m,i}\right) + \int_{T_i}^{T_{i+1}} f_c\left(\tau, x(\tau), \varepsilon_{m,i}, \varepsilon_{d,i}\right) d\tau. \tag{3.9}$$

We formulate and solve a constrained optimal control formulation with local linearization of f_c, in order to find a valid pair $\varepsilon_m, \varepsilon_d$ for each Y. First results haven achieved on real test-drive recordings, although these are matched against an open loop vehicle model, instead of using the ultimately desired closed loop formulation. In this preliminary formulation, u has been replaced by the measured actuator values. As an exemplary test suite, a double lane change maneuver has been executed at 10m/s for five times. Fig. 4 displays the estimated state and actuator measurement error traces as well as disturbance traces. The error traces are conformant with the test drive recordings according to the error bounds displayed in red. The bounds derived for the measurement errors are $\hat{e}_{X,Y} = 0.05m, \hat{e}_\psi = 0.5°, \hat{e}_{vx} = 0.1ms^{-1}, \hat{e}_{vy} = 0.1ms^{-1}, \hat{e}_\omega = 0.8°s^{-1}, \hat{e}_{ax} = 0.1ms^{-2}, \hat{e}_\delta =$

1° and $\hat{e}_{fx} = 0.049ms^{-2}$, $\hat{e}_{fyf} = 0.028ms^{-2}$,$\hat{e}_{fyr} = 0.021ms^{-2}$ for the disturbance errors.

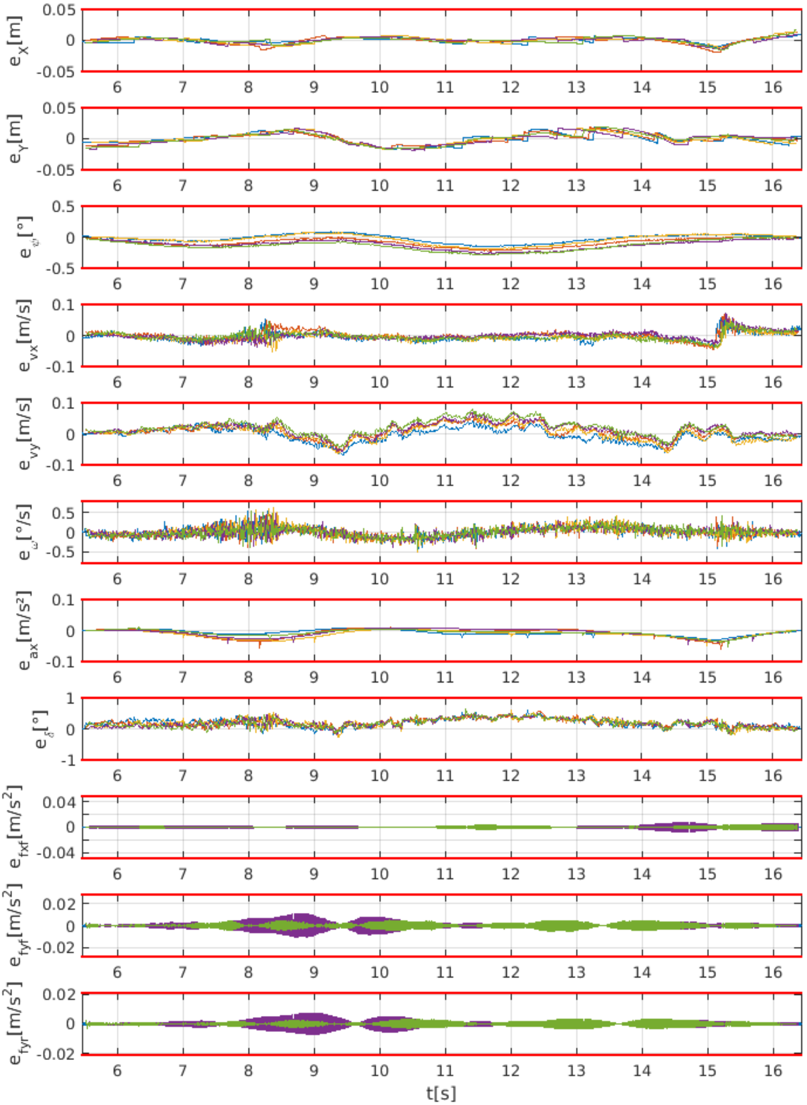

Fig. 4: Conformant error traces for five executions of a double lane change maneuver

4. Maneuver Database

In our application, the Maneuver Database is used by the Planner as a deterministic, offline computed control interface to the originally non-deterministic motion of the physical vehicle. This is achieved by selecting a set of short, exemplary motions (reference trajectories) and using Reachability Analysis [10] to compute an upper bound on the disturbed vehicle's maximum possible deviation from a reference. Adding the extent of the vehicle body to this upper bound yields the area, which has to be reserved for collision-free execution of a so called motion primitive. A motion planner may then construct a sufficiently long emergency maneuver as a concatenation of short, collision-free motion primitives.

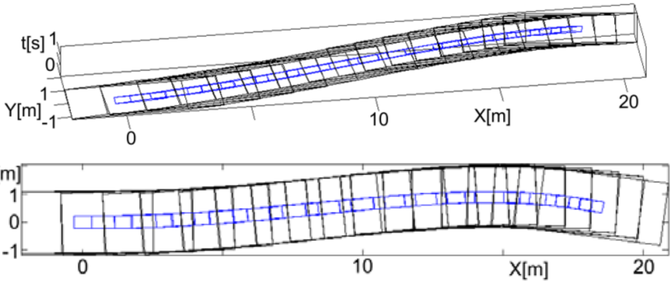

Fig. 5: Parallelotope hull of reachable sets (blue) and covered X, Y, t area, P_i (black), for an exemplary motion primitive

Def. 4.1: A motion primitive m_i is a tuple consisting of a reference trajectory $\tau_i\colon \mathbb{R} \rightarrow \mathbb{R}^{10}$, the number of sampled time steps K_i, an ordered list of reachable state sets $R_i[\cdot]$, $R_i[j] \subset \mathbb{R}^n$, with $R_i[1] = R_i^S$ the start set and $R_i[K_i] = R_i^F$ the final reachable state set of the motion primitive, with one reachable set per time step, as well as the motion primitive's X, Y, t-hull $P_i \subset \mathbb{R}^3$:

$$m_i \coloneqq \{\tau_i, K_i, R_i[\cdot], P_i\}. \qquad (4.1)$$

An exemplary motion primitive is displayed in fig. 5.

Def. 4.2: A Maneuver Database is defined similar to [11] as a tuple:

$$MDB \coloneqq \{M, \Delta, M^0, G\}, \qquad (4.2)$$

where $M = \{m_1, m_2, \ldots\}$ is a finite set of motion primitives, $\Delta \subseteq M \times M$ is the set of directed transitions between motion primitives, $M^0 \subseteq M$ is the set of possible initial motion primitives and $G \subseteq M$ is the set of final motion primitives, which lead to a standstill of the vehicle.

Def. 4.3: The Maneuver Database is sound, if all states contained in a motion primitive's final reachable set R_i^f continue to be pursued after a transition to the next motion primitive, i.e. if the first motion's final set is completely contained in the second motion's start set:

$$(m_i, m_j) \in \Delta \Longrightarrow R_i^F \subseteq R_j^S \qquad (4.3)$$

In order to create a sound and useful maneuver database, three questions have to be answered: How to facilitate the nominal set point selection from a continuous manifold? How to select the exemplary motion primitives, given that a finite number is required with which a maximally diverse set of emergency maneuvers should be constructible? And how to interconnect the motion primitives, in order to fulfill eq. (4.3)? These questions will be discussed in the following.

4.1 Sampling the Nominal Set Point Space

We want the Supervisor module to impose as little constraints as possible on the High-Level Decision modules. To facilitate operation in a usual manner, High-Level Decision modules should be allowed to select nominal set points from a continuous set. It has been shown previously [11] that a continuous range of reference trajectories may be considered in the reachability analysis, by incorporating both the actual vehicle's state space as well as the reference state space into the reachability analysis. Instead of following this approach for the complete MDB, we here allow continuous reference trajectory sets only for the entry points of the MDB, e.g. the nominal set points.

Def. 4.4: A set point bundle is defined by a set of initial reference states, $R_{x_{ref}}(0) \subset \mathbb{R}^{10}$, where one reference trajectory τ is the solution to an IVP with $\tau(0) \in R_{x_{ref}}(0)$ and $\dot{\tau} = f_{ref}(\tau)$.

In order to keep a grip on the number of bundles, which are required for a gap-free coverage of the ten-dimensional reference trajectory space, two observations are useful: First, the dynamics are invariant to the dimensions X, Y, ψ. Therefore a single sample $X = Y = \psi = 0$ is sufficient. Second, it is desirable to operate the system near a steady state surface, which allows constraining v_y, ω, δ according to the choice of v_x and a steady-state acceleration a_y^{SS}.

Def. 5.5: Using interval sets $I_d \; := d_{min} \; + \{[0, \Delta d], [\Delta d, 2\Delta d], \dots, [(k_d - 1)\Delta d, k_d \Delta d]\}$, a vector set $G = \{g_1, \dots, g_k\}, g \in \mathbb{R}^{10}$ and $\oplus$ denoting Minkovski addition, the total set of trajectory bundles for coverage of the nominal set point space is defined:

$$\Re_{NM} = \left\{R^1_{x_{ref}}(0), \dots, R^N_{x_{ref}}(0)\right\} := I_{v_x} \times I_{a_x} \times I_{a_y^{SS}} \times I_{w_1} \times I_{w_2} \oplus G \qquad (4.4)$$

The sampling of $I_{v_x} \times I_{a_y^{SS}}$ defines a two-dimensional surface of steady-states. The set G adds a certain width in the dimensions

v_y, ω, δ to each tile on the surface, in order to allow the nominal set point to digress slightly from the steady-state. The resulting structure is visualized in fig. 6.

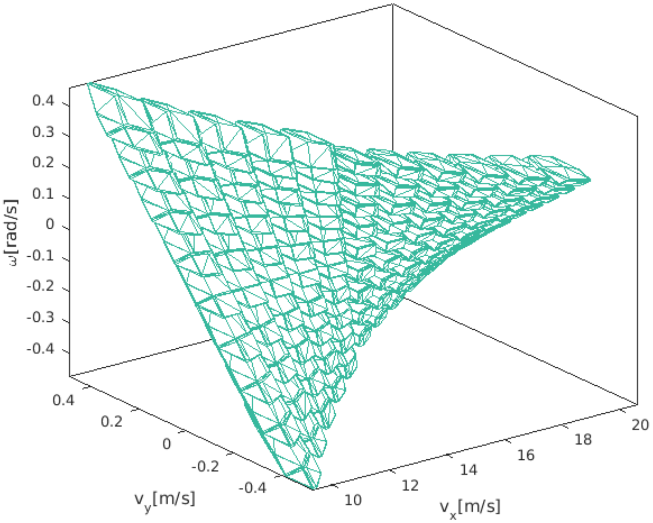

Fig. 6: Projection of $R^i_{x_{ref}}(0)$ sets (nominal set point sets)

Using the values $v_{x,min} = 10m/s$, $\Delta v_x = 1m/s$, $v_{x,max} = 20m/s$, $a_{x,min} = -4m/s^2$, $\Delta a_x = 1m/s^2$, $a_{x,max} = 2m/s^2$, $a^{SS}_{y,min} = -4m/s^2$, $\Delta a_y = 1m/s^2$, $a^{SS}_{y,max} = 4m/s^2$, as well as three intervals for I_{w_1}, I_{w_2} each, a cardinality of $\#\mathfrak{R}_{NM} = 6237$ nominal set point bundles is achieved. Thereby the operational region for nominal driving is constrained to $v_x \in [10,20]m/s$ for this study. In the current set up, it is difficult to extend the region to lower velocities, as the dynamic bicycle model is hard to analyze with reachability analysis for very low velocities.

To bound the behavior of the disturbed, closed-loop vehicle model, when tracking any of the set points $\tau(0) \in R^i_{x_{ref}}(0)$ of the set point bundle, the initial deviation of the vehicle state from the set-point has to be defined. We chose an initial tracking-error set $E_{T0} \subset \mathbb{R}^6$. Using the measurement inaccuracy E_m (sec. 3.2), the combined initial state set $R^i_{x,x_{ref}}(0) \subset \mathbb{R}^{16}$ is defined as:

$$R^i_{x,x_{ref}}(0) \coloneqq \begin{pmatrix} R^i_{x_{ref},1:6} \oplus E_m \oplus E_{T_0} \\ R^i_{x_{ref}} \end{pmatrix} \qquad (4.5)$$

Reachability analysis is then executed for the combined system dynamics $\left(f_c^{\mathrm{T}}, f_{ref}^{\mathrm{T}}\right)^{\mathrm{T}}$and the initial reachable set $R_{x,x_{ref}}^i(0)$, so that for each nominal set point tile $R_{x,x_{ref}}^i(0) \in \Re_{NM}$ a motion primitive m_0^i is constructed and placed in the MDB's set of initial motion primitives M_0. During operation of the vehicle, a nominal set point $\tau(0) \in R_{x_{ref}}^i(0)$ may be selected for execution, if the measured vehicle state x_m is inside the assumed tracking error bounds, $x_m - \tau(0) \in E_{T_0}$. The validity of transitioning from the nominal set point to a subsequent emergency maneuver is automatically guaranteed by the following construction of the MDB and does not have to be tested during operation.

It is important to note that the reference system f_{ref} is not stabilized. Therefore, the reference trajectories in a set point bundle tend to spiral away from each other after a short time, leading to increased reachable sets and difficulties in the reachability analysis. Fortunately, the necessity to keep the nominal maneuvers short is mirrored by the High-Level Behaviors' requirement to fast switching between desired set points. We are thus using short nominal set points for a duration of only $0.1\ s$ each.

4.2 Sampling the Emergency Maneuver Set Point Space

In contrast to the nominal maneuvers, emergency maneuvers are here defined to use discretely valued (singular) set points and are thus comparatively simple. We create a discrete graph with vertices $g_i \in V$ and directed edges $e \in E \subset V^2$ as a template for the creation of the maneuver automaton. Each vertex $g_i = (v_{x,i}, \varphi_i)$ represents an operating point at maximum absolute acceleration, allocated to longitudinal and lateral directions as $a_{x,i} \coloneqq -a_{max}\cos(\varphi_i)$ and $a_{y,i} \coloneqq a_{max}\sin(\varphi_i)$. An edge $(g_i, g_j) \in E$ defines a trajectory, which steers from the operating point $\left(v_{x,i}, \varphi_i\right)$ to $\left(v_{x,j}, \varphi_j\right)$. The idea, which was already pursued in [12], is to concentrate on maximum acceleration trajectories, if the number of selectable trajectories has to be limited. The duration of a trajectory is chosen in order to respect input change limitations and to comply to the end point velocity. An edge is created for each source node g_i and each target acceleration direction φ_j. The velocity of the target node g_j is chosen in such a way that the duration of the trajectory is inside a desired range. For each node below a certain velocity threshold, an edge to the standstill node $(0,0)$ is inserted. The resulting graph is visualized in fig. 7.

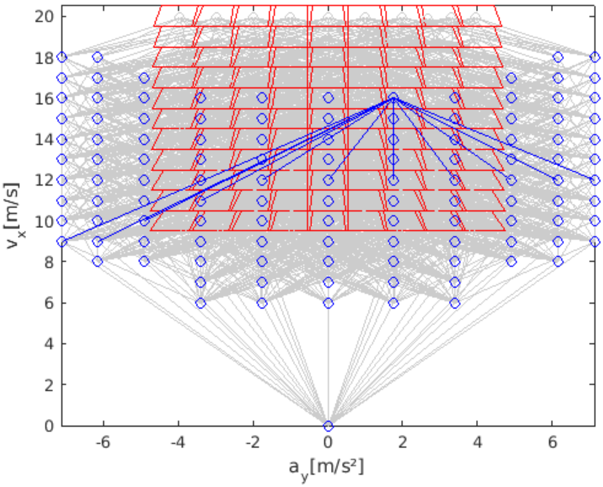

Fig. 7: Nominal set point tiles (red) and emergency maneuver grid (blue vertices, blue edges for one example vertex).

To apply the graph structure to the MDB, a function is defined, which translates an initial reference state $\tau(0)$ and a target vertex into a reference trajectory $\tau = REF\left(\tau(0), g_j\right)$ by solving the IVP for f_{ref} with appropriately chosen, constant w_1 w_2. As a concrete parametrization, an angular range of $\varphi_i \in 0.8[-\pi, \pi]$ with 11 samples and a velocity range with $v_x \in [6,20]m/s$ with a $\Delta v_x = 1m/s$ subdivision is used.

4.3 Composition of the Maneuver Database

The main purpose of the maneuver database is to provide fast access to formal bounds on the space required for set point execution and the admissible order of set point selection. In the following offline algorithm this information is assembled. The algorithm's core is a procedure $m_i \leftarrow \mathrm{REACH}(\mathrm{R}_\mathrm{i}^S, \tau_i)$, which computes the sets of reachable states for the disturbed closed loop system f_c, as defined in sec. 3, following a reference trajectory τ_i. For each motion primitive, reachability analysis begins with an initial set of vehicle states $\mathrm{R}_\mathrm{i}^S \subset \mathbb{R}^6$ and then computes the subsequent reachable sets $R_i[k], 1 < k < K_i$, for all sampled time steps. The resulting motion primitive m_i is supplemented with an over-approximation P_i of all potentially covered X, Y positions, based on the reachable sets.

As defined by eq. (4.3), the choice of the maneuvers' initial sets R_i^S is essential for the connectivity of the resulting MDB. Previously, an algorithm was proposed, which increased the initial set size iteratively to re-create the connectivity of a cyclic template graph [11]. Here, we consider nominal maneuvers and emergency maneuvers separately and make the assumption that an a-cyclic graph sufficiently represents possible maneuvers. For nominal maneuvers, a connectivity check is postponed to online analysis, as it is not safety critical. For emergency operation we assume that each motion primitive is a partial braking maneuver, thus always reducing the vehicle's velocity and thereby guaranteeing an a-cyclic structure. The proposed algorithm, alg. 1, receives as input the set of initial motion primitives M_0, which represent nominal driving as defined in sec. 4.1, as well as the template graph V, E as defined in sec. 4.2. Due to the a-cyclic nature of the template graph, alg. 1 computes the total set of motion primitives M, the set of final motion primitives G and the transition function Δ in one pass through V in the order of decreasing velocity, (line 2). For a template node g_0, all incoming motion primitives computed so far are collected, M_{in} (line 3) and grouped, (line 4). We define a distance metric based on the center and the interval hull of the end set R_i^F of a maneuver m_i. The procedure $C \leftarrow \mathrm{CLUSTER}(M_{in})$, $C \in \mathbb{N}^{\#M_{in}}$ uses k-means clustering to group the incoming motion primitives according to this metric, resulting in groups of motion primitives M_{in}^c with similar end sets, (line 6). The vector of indices C links a maneuver m_j to its cluster and the associated centroid maneuver $m_{C(j)}$. For a group M_{in}^c, the hull R_0^c of the end sets of all associated maneuvers is computed so that $m_j \in M_{in}^c \Rightarrow R_j^F \subseteq R_0^c$ is guaranteed, (line 7). The algorithm then attempts to create one motion primitive per edge in the template graph, which starts at g_0: Based on the target velocity and acceleration direction of the node g_1, REF creates a reference trajectory as a continuation of the reference trajectory of the group's centroid motion primitive, and a new motion primitive m_k is thus created, (line 10). If the reachable sets of m_k guarantee compliance with all dynamics constraints, as computed by $\mathrm{VALIDATE}$, m_k is admitted, (line 11-14). In line 15, connections are created from all incoming motion primitives of this group, to all valid outgoing motion primitives, as eq. (4.3) is guaranteed by construction, (line 7).

Goal motion primitives are created based on their reference velocity at the end point. Further post-processing steps may imbue the MDB with additional information, as for example a precomputed heuristic value for the online graph search, or they could enforce that only motion primitives are contained, which eventually lead to a standstill. Applying the algorithm to the concrete numbers given above yields a database with a total number of $\#M = 12{,}434$ motion primitives and $\#(M \backslash M_0 \backslash \mathrm{G}) = 5{,}469$ intermediate motion primitives.

```
$\textbf{COMPUTE\_MDB}(V, E, M^0)$
$M \leftarrow M^0$
For each $g_0 \in V$, ordered by $v_x(g_0)$, decreasing
    $M_{in} \leftarrow \{m_j \in M \mid endsAt(m_j, g_0)\}$
    $C \leftarrow \mathrm{CLUSTER}(M_{in})$
    For each unique $c \in C$
        $M_{in}^c \leftarrow \{m_j \in M_{in} | C(j) = c\}$
        $R_0^c \leftarrow \mathrm{HULL}(\{R_j^F | m_j \in M_{in}^c\})$
        $M_{out}^c \leftarrow \{\ \}$
        For each $g_1 \in V$, if $(g_0, g_1) \in E$, then
            $m_k \leftarrow \mathrm{REACH}\left(R_0^c, \mathrm{REF}(\tau_c(t_{K_c}), g_1)\right)$
            If $\mathrm{VALIDATE}(m_k)$, then
                $M_{out}^c \leftarrow M_{out}^c \cup \{m_k\}$
                $M \leftarrow M \cup \{m_k\}$
                If $v_x(g_1) \leq v_{min}$, then $G \leftarrow G \cup \{m_k\}$
        $\Delta \leftarrow \Delta \cup M_{in}^c \times M_{out}^c$
```

Alg. 1: Computes a maneuver database based on an a-cyclic template graph structure and a set of initial motion primitives.

5. Constraints: Interaction and Cooperation

The purpose of the Constraints module is to supply information to the planner, which allows discriminating between admissible and inadmissible motions of the vehicle, according to the current traffic situation. A set $F \subset \mathbb{R}^3$ has to be computed, which describes the forbidden X, Y, t region. Besides static constraints resulting from lane boundaries, it is especially interesting to consider the forbidden region resulting from possible actions of other traffic participants as well as the exchange of guarantees between cooperating vehicles, which either decrease or increase the size of F. The following sec.

5.1 defines F in a non-cooperative setting, where vehicles do not exchange information, but may nonetheless interact without colliding (merge into gaps) based on static assumptions. Sec. 5.2 defines a cooperation scheme based on contracts between vehicles, as well as the changes to F, which are required to represent the contracts. Sec. 5.3 proposes a C2C-message based realization of the cooperation scheme for automated vehicles and 5.4 proposes a realization based on implicit communication, in order facilitate a certain degree of cooperation between vehicles, where no direct communication channel is available, (e.g. between automated vehicles and human drivers).

5.1 Conservative Predictions

It is certainly possible to find conservative bounds on the behavior of other traffic participants by considering physically imposed acceleration limits only. But, as can be readily imagined, the exclusive use of acceleration bounds results in huge reachable sets, which cover the entire drivable area after short time. A remedy is to introduce additional constraints, as for example in [13], which may include speed limits or non-intersection constraints between third-party vehicles or third-party vehicles and static environment features, (e.g. lane boundaries), as well as legal constraints, to structure the possible behavior of other traffic participants. For the presented approach acceleration limits are further restricted by considering lane assignments and road traffic regulations.

According to §7 (5) StVO, a lane change may only be executed if no other traffic participants are endangered and if the lane change has been timely and clearly indicated. Interpreting that a vehicle executing a lane change has to guarantee safety of the lane change, admits the assumption that other vehicles either stick to their lane or are already certain about the safety of their lane change. In the following, the standard prediction is therefore that vehicles stick to their lane, if they do not indicate and if they have not yet begun leaving their lane.

Def. 5.1: A lane with index k out of all lanes K is defined to have a shape parametrized by the distance s along the lane center $c_k\colon \mathbb{R} \to \mathbb{R}^2$ and a lateral offset $n_k(s) \in \mathbb{R}^2$ with $n_k(s) \perp \frac{\partial c_k(s)}{\partial s}$, which extends to the lane boundary. A cross section through the lane at s is:

$$L_k(s) \coloneqq c_k(s) \oplus [-1,1] \cdot n_k(s) \qquad (5.1)$$

Def. 5.2: A vehicle of possible initial positions $S_0 \subset \mathbb{R}$, initial velocities $dS_0 \subset \mathbb{R}$, with length $2l$, predicted after t_0 along a lane k, reserves a space $\hat{X}_{L_k}(t)$ at time t, if a set of additional constraints $\Gamma = \{C_1, C_2 \ldots\}$ is applied:

$$\hat{X}_{L_k}(t;\ t_0, S_0, dS_0, \Gamma) \coloneqq \left\{ L_k(s(t) \oplus [-l, l]) \middle|\ t > t_0 \bigwedge_i \mathrm{C_i} \in \Gamma \right\} \tag{5.2}$$

We use the following constraints, in order to fix the initial state and to bound the acceleration:

$$C_0 \colon s(t_0) \in S_0 \wedge \dot{s}(t_0) \in dS_0 \tag{5.3}$$

$$C_a \colon a_{min} \leq \ddot{s}(t) \leq a_{max} \tag{5.4}$$

Predictions of type $\hat{X}_{L_k}(t;\ t_0, S_0, dS_0, \{\mathrm{C_0}, C_a\})$ extend along the lane ad infinitum and thereby prevent any lane changes of the ego vehicle to a lane occupied by following, non-communicating or human-steered cars: Their prediction would invalidate any emergency maneuvers reaching $v = 0$ in the target lane. In order to resolve this issue, we demand that each traffic participant maintains an emergency maneuver, which could bring it to a standstill under a velocity dependent time bound and furthermore that the traffic participant must be able to detect a lane change onto its lane early enough to react by applying a moderate deceleration a_b. Therefore the following, additional constraint on the velocity is defined, using a reaction delay T_r, a local speed limit $v_{max}(s)$ and the moderate braking capability $a_b < 0$ with $a_{min} < a_b$:

$$C_v \colon \dot{s}(t) \leq \max\big(0, v_{max}\big(s(t)\big) + a_b \cdot max(0, t - t_0 - T_r)\big) \tag{5.5}$$

An interesting question is which values are acceptable for interaction with human drivers: Using a high T_r and small $|a_b|$ leads to a conservative vehicle automation behavior, whereas low T_r and higher $|a_b|$ might overestimate human driving capabilities.

Def. 5.3: A vehicle i is matched to a lane, if any part of the vehicle body intersects with the lane area. Matched lane indices are collected in the set $K_i \subset K$.

Def. 5.4: If the state measurement s_i^m, v_i^m of a vehicle i is attained at time t_i^m with a bounded uncertainty $E_s \subset \mathbb{R}$, $E_v \subset \mathbb{R}$, we define the initial sets $S_{i,0} \coloneqq s_i^m \oplus E_s$ and $dS_{i,0} \coloneqq v_i^m \oplus E_v$.

Def. 5.5: If vehicle i is matched to multiple lanes K_i, the total prediction set assumed by vehicle j is the union of the lane based predictions, excepting the lane behind vehicle j:

$$\hat{X}_i(t; t_i^m) \coloneqq \bigcup_{k \in K_i} \begin{cases} \emptyset & \text{if } k \in K_j \wedge s_i^m < s_j \\ \hat{X}_{L_k}\big(t;\ t_i^m, S_{i,0}, dS_{i,0}, \{\mathrm{C_0}, C_v, C_a\}\big) & \text{otherwise} \end{cases} \tag{5.6}$$

To satisfy the StVO, we require the ego vehicle to use its indicator for a duration of T_i before it is allowed to traverse to another lane.

After T_i, $k_{ego}+1$ or $k_{ego}-1$ are added to K_{ego}. If the indicator is switched off before entering the adjacent lane, or after leaving the previous lane, indices are obviously also removed.

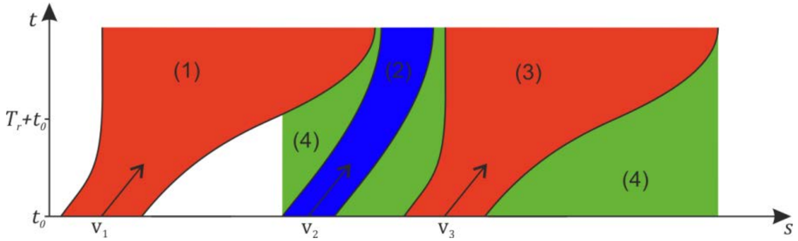

Fig. 8: A lane change without cooperation requirement: Vehicle v_2 can safely merge into the gap between v_1 and v_3, as an emergency maneuver (2) exists, which is non-intersection with predicted sets $\hat{X}_{v_1}$ (1) and $\hat{X}_{v_3}$ (3).

Def. 5.5: From the perspective of the ego vehicle with index j, the drivable lane area D and its complement, the forbidden region F_D are defined as:

$$D(t) \coloneqq \bigcup_{k\in \mathrm{K_j}} \{L_k(s) | s \geq s_j(t)\}, \qquad F_D(t) \coloneqq \mathbb{R}^3 \backslash D(t) \tag{5.7}$$

Def. 5.6: The total, non-cooperative constraint set used by vehicle j for planning an emergency maneuver, which starts at t_0 is therefore defined as:

$$F_{NC}^j(t;t_0) \coloneqq \mathrm{F_D}(t_0) \bigcup_{i\in V\backslash\{j\}} \hat{X}_i(t;t_i^m)\,; \ \text{ with } \forall i \in V{:}\, t_0 > t_i^m \tag{5.8}$$

This particular definition facilitates non-cooperative lane changes. It is based on a global requirement induced by the constraint C_v, which limits the size of prediction sets. Depending on the driving performance that can be expected from other traffic participants, (reaction time and deceleration), the tightness of possible lane change maneuvers varies. Fig. 8 shows the prediction sets for vehicles 1 and 3 on a lane k, which allow vehicle 2 to make a lane change to lane k.

5.2 Cooperation

In order to facilitate cooperation between vehicles, the terms reservation, promise, occupied set and promised set are introduced.

The non-cooperative forbidden set F_{NC} is then augmented with constraints, which guarantee safety of vehicle cooperation.

Def. 5.7: A reservation $\rho_{r,i}$ is a tuple consisting of the reservation id r, the reserving vehicle id i, the target lane id k, a reservation start time $t_{r,0}$, acceleration bounds $a_{r,min}$ and $a_{r,max}$ used to refine C_a, an initial position set S_0 and an initial velocity set dS_0:

$$\rho_{r,i} \coloneqq \{r, i, k, t_{r,0}, a_{r,min}, a_{r,max}, S_{r,0}, dS_{r,0}\} \tag{5.9}$$

A reservation $\rho_{r,i}$ is dual to vehicle i claiming an option on the space on lane k, which is accessible to it at time t according to its latest state at time t_i^m.

Def. 5.8: At first, the preliminary reservation $\hat{X}_i^{r0}$ is defined, using a lane-based prediction with the modified constraint set $\Gamma_\mathrm{r} = \{C_0, C_v, C_a(a_{r,min}, a_{r,max})\}$:

$$\hat{X}_i^{r0}(t) \coloneqq \hat{X}_{L_k}(t; t_{r,0}, S_{r,0}, dS_{r,0}, \Gamma_\mathrm{r}) \tag{5.10}$$

Subsequently, a disjunction is applied, in order to shrink the reservation area with newer observations of the vehicle state: If, according to the usual lane based prediction, a vehicle i is no longer able to attain certain states covered by $\hat{X}_i^{r0}$, these states do not have to be held free by others.

Def.: Considering a vehicle state measurement at time t_i^m and a prediction based on constraints $\Gamma = \{C_0, C_v, C_a\}$, the area dedicated to a reservation with index r is:

$$\hat{X}_i^r(t; t_i^m) \coloneqq \hat{X}_i^{r0}(t) \cap \hat{X}_{L_k}(t; t_i^m, s_i(t_i^m), v_i(t_i^m), \Gamma) \tag{5.11}$$

Def. 5.9: The occupied set O_j of a vehicle j is a set of reservations, which the vehicle j is going to honor:

$$O_j = \{\rho_{1,1}, \rho_{2,1}, \rho_{1,2}, \dots\} \tag{5.12}$$

A vehicle j promises not to plan any emergency maneuvers starting at a time t_0, which would conflict with reservation sets defined by reservations in O_j. In order to maintain invariant safety, a vehicle, which wants to add a reservation to its occupied set, has to make sure first that it can still construct an emergency maneuver inside the augmented forbidden set.

Def. 5.10: A promise is a tuple consisting of a reservation and the id j of a vehicle, which promises to honor the reservation:

$$p_{j,r,i} \coloneqq \{j, \rho_{r,i}\} \tag{5.13}$$

Def. 5.11: The set P_i of a vehicle i keeps track of received promises, which describe that it is known to vehicle i that other vehicles $j, h, \dots$ will honor a certain reservation of vehicle i:

$$P_i = \{p_{j,r,i}, p_{h,r,i}, \dots\} \tag{5.14}$$

It must be certain, that vehicles $j, h, \dots$ honor the reservation:

$$\{j, \rho_{r,i}\} \in P_i \Longrightarrow \rho_{r,i} \in O_j \tag{5.15}$$

Given a state of the set P_j of vehicle j, the constraint set used to predict vehicle i can be refined:

$$\hat{X}_i(t; t_i^m, P_j) := \hat{X}_i(t; t_i^m) \setminus \bigcup_{p_{i,r,j} \in P_j} \hat{X}_j^r(t, t_j^m) \tag{5.16}$$

The cooperation-based forbidden set used by vehicle j for planning an emergency maneuver starting at t_0 is therefore defined as:

$$F_C^j(t; t_0, O_j, P_j) := \mathrm{F_D}(\mathrm{t}) \bigcup_{i \in V\{j\}} \hat{X}_i(t; t_i^m, P_j) \bigcup_{\rho_{r,i} \in O_j} \hat{X}_i^r(t, t_i^m) \tag{5.16}$$

An example of a refined prediction is given in fig. 9: The tighter gap between vehicle 1 and 3 becomes accessible to vehicle 2, if vehicle 1 positively answers the reservation request of vehicle 2.

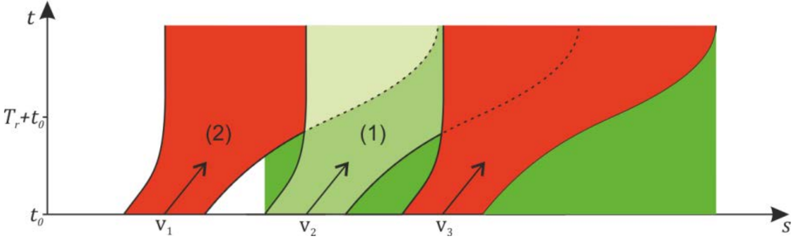

Fig. 9: A lane change with cooperation requirement: In order to allow vehicle v_2 to safely merge into the smaller gap between v_1 and v_3, a reservation $\rho_{r,2}$ (1) has to be defined. If $\rho_{r,2} \in O_1$, then $\{1, \rho_{r,2}\} \in P_2$ is admissible, so that v_2 may plan an emergency maneuver based on (2), the augmented prediction $\hat{X}_1(t; t_1^m, P_2) = \hat{X}_1(t; t_1^m) \backslash \hat{X}_2^r(t, t_0)$.

5.3 Cooperation based on C2C Communication

It is possible to explicitly transmit intentions between vehicles, which are automated and equipped with a C2C module. Therefore a realization of the cooperation strategy described in sec. 5.2 is straightforward. For the message protocol, we define two types of messages, Request and Promise. A Request message contains the information of a reservation as defined in eq. (5.9). A vehicle sends a Request message via broadcast to all vehicles in its vicinity. Each vehicle is assumed to be known by a unique id and is able to keep track of all request identifiers, it has been using so far. Therefore the combination of request id and vehicle ID (r, i), is unique. Recipients of a Request message have the option to ignore the Request, or to answer it with a Promise message. A request message is not answered, if the recipient cannot or does not want to guarantee integrity of the reservation. E.g. vehicle v_3 in fig. 9 could prefer to be uncooperative, in order to maintain a steep emergency maneuver.

A Promise message is realized as a data structure containing the three IDs (j, r, i). If a recipient of a Request decides to reply with a Promise message, it has to test first, whether the continued existence of at least one emergency maneuver is guaranteed under the tightened constraints. The question can be decided by re-planning the emergency maneuver under the tightened constraints, yet this could incur unwanted computational demands if many Requests are received. In our approach we opt to test for intersection between the current emergency maneuver and the reservation to determine whether they are compatible. The test is carried out by the Contract Validation module, see fig. 2. This simpler test is guaranteeing correctness, but is more conservative and less cooperative than re-planning. Before a Promise message is issued, the Supervisor module assures that an entry is made in the Occupied set. In this way, eq. (5.15) and therefore safety of the cooperation is guaranteed, irrespective of the performance of the communication layer. Due to the possibility of message loss, eq. (5.15) is not an equivalence relationship. The proposed protocol is guaranteeing safety and is also resilient to misuse, as reservations are limited in space and are only relevant with respect to the reserving vehicle's actual state.

5.4 Implicit Cooperation

Usually, human drivers initiate cooperation using gestures and the vehicle's indicators. Easily observable and interpretable to the automated vehicle is probably only the indicator. Yet even the indicator is an implicit form of communication, as relevant details considering start time, start velocity or intended accelerations of a lane change cannot be unambiguously derived. In order to nonetheless show a degree of cooperation towards unequipped vehicles, we create reservation requests, when an active indicator is observed: If an unequipped vehicle indicates a lane change ahead of the automated vehicle, the automated vehicle creates a virtual Request message addressed to itself. The reservation is set to the earliest lane-change, which is compatible with the current emergency maneuver.

6. Planning

We use an anytime weighted A* algorithm based on [14], which builds a search tree from the motion primitives in the Maneuver Database. The concept has been evaluated in [12] and is only sketched here. The root node of the search tree is created by selecting the nominal set point bundle $R^{i}_{x_{ref}}$, which contains the High-level behavior's desired set point τ_{des}. The entry-point into the MDB is thereby defined. To create edges in the search tree, motion primitives are chosen from the transition function Δ according to their predecessor. A motion primitive's X, Y, t-hull is translated and rotated according to the progress made by its predecessors. The transformed hull is then tested for intersection with the forbidden set F_c, using hierarchically applied separating axis tests [15]. In case of any intersection, the edge is marked invalid and is discarded. As soon as a valid edge with a motion primitive from the goal set G is explored, a safe emergency maneuver and therefore a proof of safety for the set point τ_{des} has been found.

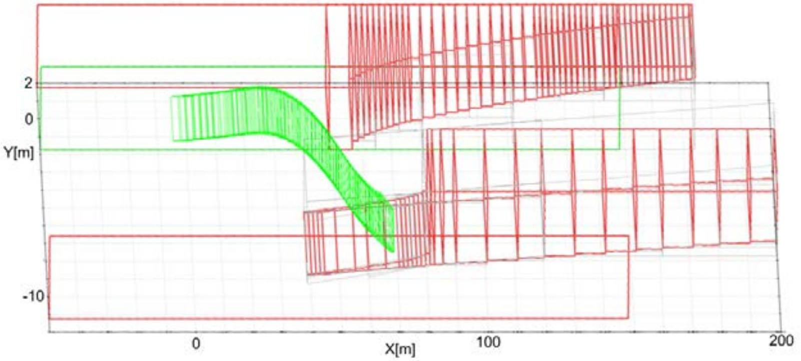

Fig. 10: Example of a valid emergency maneuver (green) under consideration of a forbidden set F (red).

The anytime extension is used, in order to give the search a depth-first bias, as the primary goal is to find any kind of safety proof. While the current T_p time slice is not at its end, the planner refines the solution, in order to provide better results in case the emergency maneuver should be selected for execution after the next time slice. If the planner cannot find a valid emergency maneuver in the current time slice, it will break off searching and not provide any update to

the Dispatcher. The Dispatcher will then automatically switch to execution of the previously computed emergency maneuver.
An example for an emergency maneuver is shown in fig. 10. In the given traffic situation, one car is stopped in the lane in front of the ego vehicle, and one car is moving in the adjacent (right) lane. The maneuver planner is able to compute a valid emergency maneuver, which brings the ego-vehicle to a standstill in the adjacent lane.

7. Conclusion and Outlook

Validation approaches for automated vehicles, which are based solely on testing or offline verification are difficult to realize. We present a detailed description of an approach to safety analysis for cooperative, automated driving, which is based on a combination of testing, offline- as well as online-verification. A Supervisor module makes use of offline pre-computed results, relates them to the current traffic situation and shows whether set points or cooperation agreements are safe. While an agreement between two cooperating, automated vehicles is rather straight-forward, the cooperation with human drivers is difficult: There are no legal guidelines, which driving performance can be expected from a human driver, (e.g. reaction times). Furthermore, without a direct communication channel, no certainty about the intent of human drivers can exist. It is worthwhile to further investigate human capabilities as well as implicit cooperation based on conservative predictions. The presented approach, as well as similar ones, which are making use of over-approximations are necessarily restrictive on the vehicle behavior. It is interesting to further quantify the limitations, which are resulting from over-approximative reachability analysis, the finiteness of sampled motion primitive sets and the conservative environment prediction.

8. Acknowledgements

The authors gratefully acknowledge financial support by the European Commission project UnCoVerCPS under grant number 643921 and by Deutsche Forschungsgemeinschaft (DFG) for the project CoInCiDE under grant number 623093.

References

[1] Kalra, N., Paddock, S., Driving to Safety: How Many Miles of Driving Would It Take to Demonstrate Autonomous Vehicle Reliability? RAND Corporation, Santa Monica, 2016.

[2] Colombo, A., Del Vecchio, D., Least Restrictive Supervisors for Intersection Collision Avoidance: A Scheduling Approach, IEEE Transactions on Automatic Control, vol. 60, no. 6, pp. 1515-1527, 2015.

[3] Loos, M. S., Platzer, A., Nistor, L., Adaptive Cruise Control: Hybrid, Distributed, and now Formally Verified, Proc. of the 17th International Symposium on Formal Methods, pp. 42-56, 2011.

[4] Damm, W., Peter, H. J., Rakow, J., Westphal, B., Can we build it: Formal synthesis of control strategies for cooperative driver assistance systems, Mathematical Structures in Computer Science, vol. 23, pp. 676-725, 2013.

[5] Althoff, M., Dolan, J. M., Online Verification of Automated Road Vehicles Using Reachability Analysis, IEEE Transactions on Robotics, vol. 30, no. 4, pp. 903-918, 2014.

[6] Jain, V., Heß, D., Löper, C., Frankiewicz, T., Hesse, T.: Hierarchical Approach for Safety of Multiple Cooperating Vehicles. Symposium Automatisierungssysteme, Assistenzsysteme und eingebettete Systeme für Transportmittel, (AAET), Feb. 8./9., Braunschweig, 2017.

[7] Donges, E., Aspekte der Aktiven Sicherheit bei der Führung von Personenkraftwagen. Automobil-Industrie 27, pp. 183-190, 1982.

[8] Final draft EN 302-636-4-1 v1.2.1, Intelligent Transport Systems (ITS); Vehicular Communications; GeoNetworking; Part4: Geographical addressing and point-to-multipoint communications; Sub-part 1: Media-Independent Functionality. ETSI 2014.

[9] Roehm, H., Oehlerking, J., Woehrle, M., Althoff, M., Reachset Conformance Testing of Hybrid Automata, Proc. of Hybrid Systems: Computation and Control, pp. 277-286, 2016.

[10] Althoff, M., An Introduction to CORA 2015, Proc. of the Workshop on Applied Verification for Continuous and Hybrid Systems, 2015.

[11] Heß, D., Althoff, M., Sattel, T., Formal Verification of Maneuver Automata for Parameterized Motion Primitives, Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1474-1481, 2014.

[12] Salvado, J., and Heß, D., Contingency planning for automated vehicles. Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2016.

[13] Althoff, M., Heß, D., Gambert, F., Road Occupancy Prediction of Traffic Participants, Proc. of the 16th International IEEE Conference on Intelligent Transportation Systems, pp. 99-105, 2013.

[14] Hansen, E. A., Zhou, R., Anytime Heuristic Search, Journal of Artificial Intelligence Research, vol. 28, pp. 267-297, 2007.

[15] Gottschalk, S., Ming, C. L., Manocha, D., OBBTree: A hierarchical structure for rapid interference detection. Proc. of the 23rd annual conference on Computer graphics and interactive techniques. ACM, 1996.

[16] Werling, M., Ein neues Konzept für die Trajektoriengenerierung und -stabilisierung in zeitkritischen Verkehrsszenarien. Schriftenreihe des Instituts für Angewandte Informatik / Automatisierungstechnik, KiT, Band 34, 2010.