CHALMERS
UNIVERSITY OF TECHNOLOGY

A Safety Monitoring Concept for Fully Automated Driving

Citation for the original published paper (version of record):
Kojchev, S., Klintberg, E., Fredriksson, J. (2020). A Safety Monitoring Concept for Fully Automated Driving. IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC.
http://dx.doi.org/10.1109/ITSC45102.2020.9294307

(article starts on next page)

A safety monitoring concept for fully automated driving*

Stefan Kojchev[1], Emil Klintberg[2] and Jonas Fredriksson[3]

***Abstract*— Safe motion planning for automated vehicles requires that a collision-free trajectory can be guaranteed. For that purpose, we propose a monitoring concept that would ensure safe vehicle states. Determining these safe states, however, is usually a computationally demanding task. To alleviate the computational demand, we investigate the possibility to compute the safe sets offline. To achieve this, we leverage backward reachability theory and compute the N-step robust backward reachable set offline. Based on the current disturbances, we demonstrate the possibility to adapt this set online. The safety guarantees are then provided by computing the robust one-step forward prediction of the state vector and checking if these states are members of the adapted safe set. The numerical example demonstrates that the approach is capable of avoiding hazardous vehicle states under an unsafe motion planning algorithm.**

I. INTRODUCTION

Fully automated vehicles are believed to have the potential to drastically change the transportation industry [1]. The main expected benefit from fully automated vehicles is an increase in traffic safety, as most accidents are caused by human factors [2]. However, fully automated vehicles still face many challenges and production level maturity is not expected in the near future. One of the challenges is how to ensure safety of the vehicle in all possible scenarios with respect to the surrounding environment. This, in particular, is a difficult challenge as the vehicle would need to account for different complex aspects such as the uncertain motion of the surrounding traffic participants, collision-free trajectory, drivable trajectory, uncertain states of the ego vehicle, etc.

Automated driving functions need to determine and follow a planned trajectory with certain vehicle parameters. These common motion planning approaches check if the computed trajectory is collision-free in a finite time horizon [3], [4]. Approaches that can ensure that the vehicle can reach a desired collision-free set have been intensely investigated in recent literature. One approach is by defining inevitable collision states [5]. Inevitable collision states are defined as states for which collision with an obstacle eventually occurs regardless of any future trajectory that is followed by the system. Determining these states, however, is computationally demanding and challenging to be implemented for online computation. One can, however, argue that inevitable collision states could occur just after the predicted horizon. Furthermore, such approaches require known initial conditions that are not exposed to uncertainties and the convergence proofs of these algorithms are often difficult to obtain.

Other popular methods that gain momentum in recent years are data-driven algorithms (i.e., deep learning or machine learning) [6], [7]. Although these algorithms are quite general in terms of their applicability (i.e., one algorithm can work for multiple different scenarios), they typically require a large amount of data to be validated and trained to perform as good as their human counterpart or better [8], [9].

Another alternative approach is using reachability analysis to check if the vehicle states are collision-free [10], [11], [12]. A reachable set is the set of states that can be reached by a system for a given set of initial states, inputs, and disturbances. These approaches can also account for the future motion of the traffic participants and uncertainties from the environment. The computed reachable sets of the ego vehicle are compared to the reachable sets of the traffic participants and collisions are defined as any intersection between those sets. A drawback of the reachability analysis is that the techniques over-approximate the set of reachable states, which makes the set over conservative. Considering all feasible trajectories of the other traffic participants in combination with the over-approximation can lead to the rapid growth of unsafe regions. Furthermore, minimizing the over-approximation of the reachable sets is a computationally demanding task.

A way to reduce the conservatism is Control Barrier Functions as presented in [13], [14], [15]. The method guarantees forward invariance of a set, which in term would satisfy the safety conditions of a system. The forward invariance of the set is ensured by the barrier function satisfying Lyapunov-like conditions. However, finding such barrier functions is the main difficulty of the approach, with computing barrier functions for general classes of control systems being an open problem.

The work presented in this paper demonstrates a monitoring concept for autonomous systems that leverages backward reachability analysis as a technique to ensure that the vehicle can reach a desired safe state. In order to battle the computation burden, we propose a way to do parts of the set calculations offline. The offline part determines the set of permissible states based on the backward reachability analysis. In addition to this, we propose an approach that is capable to modify the permissible set online to account for the uncertainties of the environment, which differs from

*This work is partially funded by Sweden's innovation agency Vinnova, project number: 2018-02708.

[1]Stefan Kojchev is with Volvo Autonomous Solutions and the Mechatronics Group, Systems and Control, Chalmers University of Technology stefan.koychev@volvo.com; kojchev@chalmers.se

[2]Emil Klintberg is with Volvo Autonomous Solutions, 41873 Göteborg, Sweden emil.klintberg@volvo.com

[3]Jonas Fredriksson is with the Mechatronics Group, Systems and Control, Chalmers University of Technology, 41296 Göteborg, Sweden jonas.fredriksson@chalmers.se

concepts presented in [19], [20]. The online part also focuses on calculating a robust one-step forward prediction of the vehicle states. The safety guarantees are provided by checking if the robust one-step forward prediction set is a member of the safe set. In case this condition is not satisfied, the software initiates an evasive maneuver in order to avoid a hazardous situation. We believe that a concept of this type can be applied to any autonomous system, however, in this paper we focus on its application to fully automated driving systems. It should be noted that the contribution primarily aims at evaluating the potential of using the backward reachability analysis approach for ensuring safety of a heavy-duty vehicle in a relatively simple driving scenario.

The remainder of the paper is organized as follows: Section II provides the necessary preliminaries on basic definitions. In Section III we present the monitoring concept, while Section IV gives an explanation of how the permissible set is adapted online. In Section V, we present a numerical example followed by final remarks in Section VI.

II. Preliminaries

In this section, we introduce definitions on sets and reachability analysis. For further information regarding reachability analysis and set invariance theory see [16].

A. Polytopes and redundant inequalities

Let $\mathcal{P}$ be a polyhedron defined as the intersection of a finite set of closed halfspaces in $\mathbb{R}^n$:

$$\mathcal{P} = \{x \in \mathbb{R}^n | \; Qx \leq r\}, \tag{1}$$

where $Qx \leq r$ is the shorthand form for a system of inequalities. A *polytope* is a bounded polyhedron.

If an inequality can be removed from the description of a polyhedron without changing the solution set, then the inequality is *redundant*. Similarly, if an inequality is not redundant, it is *necessary*. If all inequalities describing a polyhedron are necessary, we have a *minimal-representation* of the polyhedron. An approach that describes how to find the *minimal-representation* of the polyhedron is presented in [17] and is used in this paper.

B. Polytopic linear systems

Let us consider a discrete-time linear system of the following form:

$$x(k+1) = A(k)x(k) + E(k)w(k), \tag{2}$$

where $x \in \mathbb{R}^{n_x}$ and $w \in \mathbb{R}^{n_w}$ denote the state variables and an exogenous disturbance respectively. The exogenous disturbance is assumed to be bounded,

$$w(k) \in \mathcal{W}, \tag{3}$$

for some closed and bounded set $\mathcal{W}$. It is assumed that the system matrices are contained in the convex hull of a set of matrix pairs, i.e.:

$$(A(k), E(k)) \in \Delta = \mathbf{Co}((A_1, E_1), ..., (A_k, E_k)), \;\; \forall k. \tag{4}$$

This is a well-studied class of systems that in literature is often referred to as *polytopic linear systems*.

In the following we define the one-step and the N-step backward reachable sets and the one-step forward reachable set as:

Definition 1: For a given target set $\mathcal{X}$, the one-step backward reachable set (or preimage set) $\text{Pre}(\mathcal{X}, \mathcal{W}, \Delta)$ of the system dynamics (2) is defined as:

$$\begin{aligned} &\text{Pre}\left(\mathcal{X}, \mathcal{W}, \Delta\right) = \\ &\{x \in \mathbb{R}^{n_x} | \; Ax + Ew \in \mathcal{X}, \; \forall w \in \mathcal{W}, \; \forall \left(A, E\right) \in \Delta\} \end{aligned} \tag{5}$$

In other words, the one-step backward reachable set is the set of states that gets robustly mapped to $\mathcal{X}$ by (2).

Note that if $\mathcal{X} = \left\{x \in \mathbb{R}^{n_x} | \; \bar{H}_{(j,:)}x \leq \bar{h}_j\right\}$ for some $\bar{H} \in \mathbb{R}^{m_x \times n_x}$ and $\bar{h} \in \mathbb{R}^{m_x}$, where we have introduced the notations $\bar{H}_{(j,:)}$ for the j-th row of $\bar{H}$ and $\bar{h}_j$ for the j-th element of $\bar{h}$, the one-step robust controllable set can be evaluated as:

$$\begin{aligned} &\text{Pre}(\mathcal{X}, \mathcal{W}, \Delta) = \\ &\left\{x \in \mathbb{R}^{n_x} \mid \bar{H}_{(j,:)}A_i x \leq (\tilde{h}_i)_j, \; i = 1, ..., k\right\}, \end{aligned} \tag{6}$$

where element j of $(\tilde{h}_i)_j$ is given by:

$$(\tilde{h}_i)_j = \min_{w \in \mathcal{W}} \left(\bar{h}_j - \bar{H}_{(j,:)}E_i w\right), \;\; j = 1, ..., m_x. \tag{7}$$

Thus, if $\mathcal{W}$ is polyhedral, the one-step robust controllable set can be calculated by solving $m_x \cdot k$ Linear Programs (LPs).

Definition 2: For a given target set $\mathcal{X}$, the N-step backward reachable set $\text{Pre}^N(\mathcal{X}, \mathcal{W}, \Delta)$ of the system (2) is defined recursively as:

$$\Omega_0 = \mathcal{X} \tag{8a}$$

$$\Omega_{i+1} = \text{Pre}\left(\Omega_i, \mathcal{W}, \Delta\right) \cap \mathcal{X}, \quad i = 0, \ldots, N-1 \tag{8b}$$

$$\text{Pre}^N\left(\mathcal{X}, \mathcal{W}, \Delta\right) = \Omega_N \tag{8c}$$

Hence, the N-step backward reachable set is the set of states that gets robustly mapped onto $\mathcal{X}$ by (2) in N time steps.

Definition 3: For a system with inputs and disturbances, the one-step forward reachable set is defined as:

$$\text{Reach}(\mathcal{X}, \mathcal{W}) = (A \circ \mathcal{X}) \oplus (B \circ \mathcal{U}) \oplus \mathcal{W}, \tag{9}$$

where $\oplus$ denotes the Minkowski sum and $\circ$ denotes the entrywise product.

III. Monitoring concept

The nominal function of an Autonomous Driving System (ADS) is often complex and composed of algorithms that are hard to analyze. It can therefore be desirable to provide safety guarantees in a simpler component that monitors the nominal ADS function. A system architecture is illustrated in Figure 1.

The safety monitor examines if there exists a maneuver at the next decision point that would bring the system to a safe state. If such a maneuver exist, the input proposed by the nominal ADS is approved. If a maneuver does not exist, a maneuver that is verified as safe at the preceding decision

point is executed. Safety is then assured by the principle of induction.

In practice, it is desirable to have a simple test to evaluate if there exists a safe maneuver or not. To achieve this, we introduce some restrictions on the safe state and on the safe maneuvers.

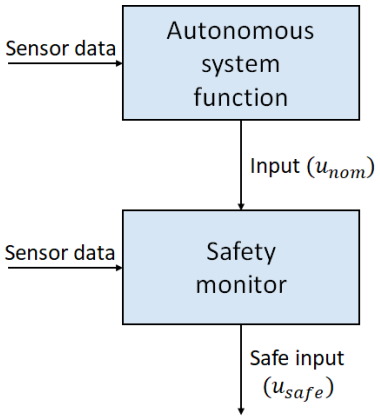

Fig. 1. Monitoring concept

A. System description, safe states and safe maneuvers

To obtain a computationally tractable monitoring problem, we place the following assumptions on the system description and on the safe maneuvers:

Assumption 1: A discrete-time linear model describing the system dynamics is available, i.e.

$$x(k+1) = A(k)x(k) + B(k)u(k) + E(k)w(k), \quad (10)$$

where x denotes the state variables, u is the control input and w are the exogenous disturbances belonging to a closed and bounded set $\mathcal{W}$.

Remark 1: Note that although it is conceptually simple to construct a model according to Assumption 1, it is a non-trivial task to validate that such a model is true, see e.g., [18]. However, uncertainties related to model errors or linearization errors can be included in the exogenous disturbance set $\mathcal{W}$.

Assumption 2: The safe maneuver is implicitly given by the following control law,

$$u(k) = L_1 z(k) - L_2 x(k), \quad (11)$$

for some reference sequence $(z(k))_{k\in\mathbb{Z}}$ and matrices L_1 and L_2.

Assumption 3: For any k, the system is at a safe state if $x(k) \in \mathcal{C}$ for some closed set $\mathcal{C}$.

Moreover, we assume that a set of safe states for the system is a subset of the state space. The set of safe states must also not be intersected by any other participating traffic agent, i.e., $\mathcal{I} \cap \mathcal{C} = \varnothing$, where $\mathcal{I}$ denotes the predicted motion of a participating agent.

Furthermore, the system states x are assumed to be measurable or can be obtained using an observer. The estimation of the system states at the current time step is denoted by $\hat{x}$. For brevity, observer design is omitted in the paper. The uncertainties related to $\hat{x}$ can be included in the exogenous disturbance set $\mathcal{W}$. An approach to model these uncertainties is to quantify them using real-world data from the same driving scenario w.r.t the predicted outcome from the model. The rare events that might not be captured in the data, could be modeled by using statistical approaches that focus on rare events properties.

In reality it is not always possible to guarantee that the whole set of safe states $\mathcal{C}$ is accessible. Therefore, we want a test of the form

$$x(k) \in \mathcal{S}, \quad (12)$$

for some closed set $\mathcal{S} \subseteq \mathcal{C}$. In the following, we refer to $\mathcal{S}$ as the *set of permissible states*.

The preimage set is calculated using the closed loop system obtained by substituting (11) in (10), the resulting closed loop matrices becomes $\bar{A}(k) = (A(k) - B(k)L_2)$ and $\bar{B}(k) = B(k)L_1$. The set of permissible states is then

$$\mathcal{S} = \bigcap_{k=1}^{N} \text{Pre}^k(\mathcal{C}, \mathcal{W}, \Delta). \quad (13)$$

The resulting monitoring concept can be formulated as the following algorithm:

Algorithm 1 Calculate u_{safe}

Input: $\bar{A}, \bar{B}, \hat{x}, \mathcal{I}, \mathcal{C}, \mathcal{S}$, nominal control input u_{nom}
Output: Safety guaranteed vehicle input u_{safe}

1: Calculate the robust forward reachable set with u_{nom} (9)
2: Check set membership:
if Reach$(\hat{x}, \mathcal{W}) \subseteq \mathcal{S}$ **and** $\mathcal{I} \cap \mathcal{C} = \varnothing$ **then**
$u_{\text{safe}} = u_{\text{nom}}$
else
Initiate evasive maneuver: $u_{\text{safe}} = u_{\text{evasive}}$
end if

Although focused on ADS, we believe that the proposed monitoring concept can work with any autonomous system (i.e., any system that behaves with high degree of autonomy). In the remainder of the paper the monitoring concept is referred to as the *safety supervisor*.

B. Predicting traffic and separation of data

The surrounding traffic, on the other hand, can be considered either by making forward predictions or by defining reasonable behaviour and traffic rules in terms of the distance between vehicles that needs to be kept, similar to [8]. To define the reasonable behaviour and the traffic rules we believe it is necessary to collect real-world data. Then we would need to prove that the safety guarantees are valid when the behaviour of the traffic is on the boundary of reasonable. If that is not the case, then we should modify the traffic rules to accommodate the behaviour. This in turn eliminates the necessity to recollect traffic data.

IV. Online adaptation of the permissible set

In practice, it is possible that the disturbance set $\mathcal{W}$ is significantly different than the set used to calculate the set of permissible states. Therefore, it can be desirable to update this set online.

A. Underlying observation

For simplicity let us consider a disturbance set of the form:

$$\mathcal{W} = \{w | -\gamma \leq w \leq \gamma\} \tag{14}$$

for some vector $\gamma > 0$. When evaluating the k-step robust backward reachable set, we use the notations $\mu_{i,j} \geq 0$ and $\pi_{i,j} \geq 0$ for the Lagrange dual variables corresponding to the inequalities $w - \gamma \leq 0$ and $-w - \gamma \leq 0$ respectively. Due to complementary slackness, the Lagrange dual variables $\mu_{i,j}$ and $\pi_{i,j}$ cannot be (elementwise) non-zero simultaneously.

Let us now introduce the notation $\lambda_{i,j} = \max(\mu_{i,j}, \pi_{i,j})$, where $\max(a, b)$ is a vector containing the elementwise maximum of its arguments, and note that:

$$\frac{\partial(\tilde{h}_i)_j}{\partial \gamma} = \lambda_{i,j}^T. \tag{15}$$

It is then straightforward to express the preimage set as:

$$\text{Pre}^k(\mathcal{X}, \mathcal{W}, \Delta) = \{x \in \mathbb{R}^{n_x} \, | H_k x \leq h_k + \mathcal{J}_k \Delta\gamma\}, \tag{16}$$

where $\Delta\gamma$ denotes a possible deviation from the value of γ that is used in the evaluation of the preimage set, $H_k = \begin{bmatrix} H_{k-1}A_1 \\ \vdots \\ H_{k-1}A_\kappa \\ \bar{H} \end{bmatrix}$, $h_k = \begin{bmatrix} \tilde{h}_{k-1,1} \\ \vdots \\ \tilde{h}_{k-1,\kappa} \\ \bar{h} \end{bmatrix}$ and $\mathcal{J}_k = \begin{bmatrix} \frac{\partial \tilde{h}_{k,1}}{\partial \gamma} \\ \vdots \\ \frac{\partial \tilde{h}_{k,\kappa}}{\partial \gamma} \\ 0 \end{bmatrix}$. The element $\frac{\partial \tilde{h}_{k,j}}{\partial \gamma}$ of $\mathcal{J}_k$ is equal to:

$$\frac{\partial \tilde{h}_{k,j}}{\partial \gamma} = \sum_{i=1}^{k} \frac{\partial \tilde{h}_{i,j}}{\partial \gamma}, \; j = 1, ..., \kappa. \tag{17}$$

In a similar fashion the sensitivities can then be propagated through (8) in order to express the permissible set as:

$$\mathcal{S} = \{x \in \mathbb{R}^{n_x} \mid Qx \leq r + \frac{\partial r}{\partial \gamma} \Delta\gamma\}. \tag{18}$$

It is interesting to note that the choice of γ does not affect the orientation of the linear inequalities in (18) but only their distance from the origin.

It should be observed that redundant inequalities can become necessary if γ is updated. The minimal-representations should therefore be calculated with care, and some inequalities that are nominally redundant may have to be kept in the description of the set.

B. Adaptation strategy

For a permissible set of the form (18), it is principally simple to adjust the size of the set when the driving conditions motivate a different choice of γ. However, when the size of the set is decreased, it should be made sure that the current one-step prediction of the system dynamics is still contained in the permissible set, as depicted in Figure 2. In the following, we provide a description of how the maximum $\Delta\gamma$ can be calculated online if $\Delta\gamma$ takes the form $\Delta\gamma = c \cdot e_j$, where c is a scalar and e_j is a Cartesian unit vector.

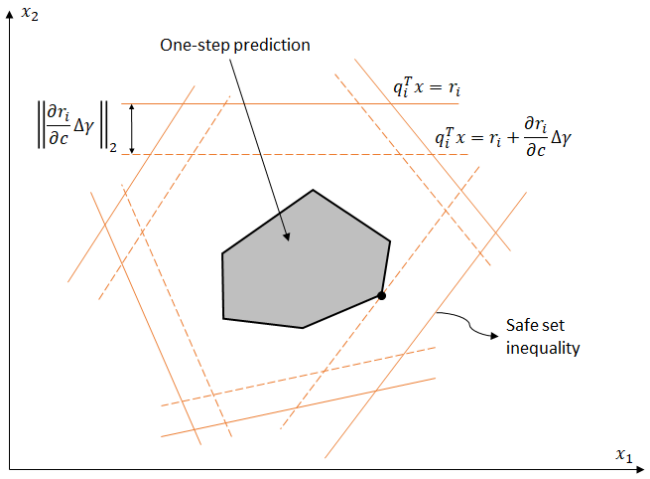

Fig. 2. Illustration of the online adaptation of the permissible set

We assume that the one-step prediction of the system dynamics is represented by the convex hull of a set of points, i.e.,

$$x(k+1) \in \mathbf{Co}(x_1, \ldots, x_p). \tag{19}$$

The residual of the inequalities that define the permissible set for each point in (19), can be calculated as:

$$\varepsilon_j = Qx_j - r, \quad j = 1, \ldots, p. \tag{20}$$

For element i of ε_j, we can then calculate the scalar c that would result in $(\varepsilon_j)_i = 0$. The maximum allowed c is then obtained as the minimum of the results, i.e.,

$$c_{\max} = \min_{j=1,\ldots,p,\; i=1,\ldots,m_{\mathcal{S}}} \frac{(\varepsilon_j)_i}{\frac{\partial r_i}{\partial c}}, \tag{21}$$

where $m_{\mathcal{S}}$ denotes the number of inequalities in the description of the permissible set. Note that $(\varepsilon_j)_i$ and $\frac{\partial r_i}{\partial c}$ are scalars, and the complexity of the operation (21) is equivalent to performing $m_{\mathcal{S}} \cdot p$ divisions and finding the minimum of the results. In the case when $\Delta\gamma > c$, the safety supervisor algorithm would need to intervene and initiate the evasive maneuver. The adaptation, results in an update of Algorithm 1 by introducing it as a first step.

V. Numerical Example

To assess the validity of the proposed concept that provides safety guarantees, a highway driving scenario with no surrounding traffic is considered.

A. Vehicle model

For the numerical example, consider the kinematic vehicle model in the spatial frame as illustrated in Figure 3.

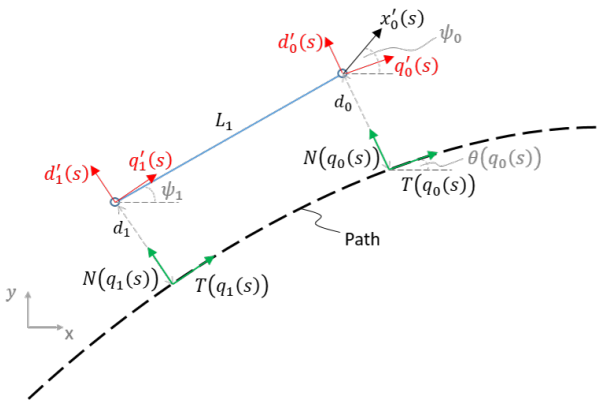

Fig. 3. Kinematic vehicle model

We define a moving coordinate system (q, d) w.r.t. a defined lane, where q is the distance traveled along the path and d is the distance from the path. Using the Frenet-Serret frame to express the relation between the tangent $(T(q_i(s)))$ and the normal $(N(q_i(s)))$ vectors, the evolution of the states can be expressed as:

$$\begin{bmatrix} q_i'(s) \\ d_i'(s) \end{bmatrix} = \begin{bmatrix} T^T(q_i(s)) \frac{1}{1-\kappa(q_i(s))d_i(s)} \\ N^T(q_i(s)) \end{bmatrix} x_i'(s), \tag{22}$$

where $x_i'(s)$ is the velocity in Cartesian space and $\kappa(q_i(s))$ is the path curvature at distance traveled $q_i(s)$.

1) Kinematic vehicle model: The equations of motion of the kinematic vehicle model depicted in Figure 3, together with (22) and using the small angles approximation can be expressed as:

$$\psi_1'(s) = \frac{|x_0'(s)|}{L_1} \sin\left(\psi_0(s) - \psi_1(s)\right) \tag{23a}$$

$$q_0'(s) = |x_0'(s)| \cos\left(\psi_0(s) - \theta(q_0(s))\right) \frac{1}{1-\kappa_0(s)d_0(s)} \tag{23b}$$

$$d_0'(s) = |x_0'(s)| \sin\left(\psi_0(s) - \theta(q_0(s))\right). \tag{23c}$$

Substituting $\psi_0^R(s) = \psi_0(s) - \theta\left(q_0(s)\right)$ and $\psi_1^R(s) = \psi_1(s) - \theta\left(q_0(s)\right)$, and assuming that $\theta'\left(q_0(s)\right) = \kappa_0(s)$ the nonlinear model equations resolve to:

$$q_0'(s) = |x_0'(s)| \cos\left(\psi_0(s) - \theta(q_0(s))\right) \frac{1}{1-\kappa_0(s)d_0(s)} \tag{24a}$$

$$d_0'(s) = |x_0'(s)| \sin\left(\psi_0^R(s)\right) \tag{24b}$$

$$\psi_1'^R(s) = \frac{|x_0'(s)|}{L_1} \sin\left(\psi_0^R(s) - \psi_1^R(s)\right) - \frac{\kappa_0(s)\,|x_0'(s)| \cos(\psi_0^R(s))}{1-\kappa_0(s)d_0(s)}. \tag{24c}$$

As the component that provides the safety guarantees requires a linear model, a first order linearization around $\bar{\Psi}_0^R = 0; \bar{\Psi}_1^R = 0; \bar{D}_0 = 0; \bar{\kappa}_0 = 0;$ is considered. Assuming that we can have a time-varying state-space matrices w.r.t. the vehicle's speed ($|x_0'(s)| = v$), the following linear state-space model is obtained:

$$\begin{bmatrix} D_0'(s) \\ \Psi_1'^R(s) \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & -\frac{v}{L_1} \end{bmatrix} \begin{bmatrix} D_0(s) \\ \Psi_1^R(s) \end{bmatrix} + \begin{bmatrix} v \\ \frac{v}{L_1} \end{bmatrix} \Psi_0^R(s) + \begin{bmatrix} 0 \\ -v \end{bmatrix} \kappa_0(s), \tag{25}$$

or in matrix form:

$$x'(s) = A(v)x(s) + B(v)u(s) + F(v)\kappa_0(s). \tag{26}$$

The attentive reader might have noticed that the linearized matrix $A(v)$ has an eigenvalue equal to 0. However, the system is controllable and therefore the evolution of the states can be controlled without instability issues.

2) Extended plant model: Given that we want to track a reference path heading, we modify the tractor heading state to the following error computation: $e_{\Psi_1^R}(s) = \Psi_1^R(s) - \Psi_1^{\text{ref}}(s)$. The extended plant model consists of the kinematic vehicle model and the road model. To define the road w.r.t. the path traveled distance we introduce the following states:

$$z(s) = \begin{bmatrix} \kappa(s) \\ \Psi_1^{\text{ref}}(s) \end{bmatrix}, \tag{27}$$

where $\kappa(s)$ is the road curvature and $\Psi_1^{\text{ref}}(s)$ is the reference heading.

We can modify the extended plant model control law to include the road terms as follows:

$$u(s) = -K_{\text{LQR}}x(s) + K_{\text{R}}z(s), \tag{28}$$

where K_{LQR} is the gain computed by minimizing the Riccati equation and $K_{\text{R}} = \begin{bmatrix} 0 \\ K_{\text{LQR}}(2) \end{bmatrix}$.

Assuming that the dynamics of the road model state vector $(z(s))$ are determined by external disturbances $(z'(s) = Ew(s))$, where $w(s)$ is the disturbance vector related to the road model, the closed-loop of the extended plant model can be formulated as:

$$\bar{x}'(s) = \bar{A}(v)\bar{x}(s) + \bar{B}(v)\bar{w}(s), \tag{29}$$

where: $\bar{A}(v) = \begin{bmatrix} A(v) - B(v)K_{\text{LQR}} & F(v) + B(v)K_{\text{R}} \\ 0 & 0 \end{bmatrix}$, $\bar{B} = E$ and $\bar{x}(s) = \begin{bmatrix} x(s) \\ z(s) \end{bmatrix}, \bar{w}(s) = \begin{bmatrix} w_x \\ w_z \end{bmatrix}$.

For use in the computation of the permissible states, we discretize the system (29) using a zero-order hold discretization technique.

B. Simulation scenario

The simulation scenario is represented as a highway driving scenario. The vehicle is moving with a constant speed of 70 km/h whilst perfectly tracking the path (i.e., $D_0(s) = 0$ and $\Psi_0^R(s) = 0$). The road is comprised of two straight patches and a segment in-between that has constant curvature. This is a representative definition of a Swedish road. The road has a length of 1100 meters, lane width of 3.75 meters and curvature radius of 400 meters, as depicted

in Figure 4. The evasive maneuver is defined as a constant braking event with a deceleration of 3.3 m/s^2 until the vehicle is fully stopped. The vehicle has a length of 13.6 meters, which is a representative length of a truck with a semi-trailer, and the simulation sampling step is 0.1 meters.

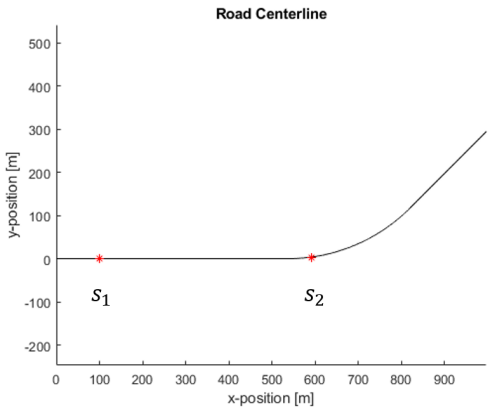

Fig. 4. Road centerline

The target set is determined from the requirement to stay within the lane bounds and to be able to account for roads with a radius as low as 20 meters. The disturbance sets for calculating the permissible set and the one-step reachable set are bounded. The linearization error for this particular scenario is calculated by model comparison and is included in the disturbance sets.

The system is controlled through the steering wheel angle. For the scenario, the nominal control input is zero in the straight parts of the road and has the necessary steering wheel angle for the curved section. To illustrate the intervention from the "safety supervisor", the nominal control input is defined in the Cartesian frame, while the vehicle model is in the spatial frame. This miss-representation would eventually cause the vehicle states to be outside of the permissible set and for the evasive maneuver to be initiated.

C. Discussion of results

Figure 5 and Figure 6 depict the simulation results. In Figure 5 each of the plots is represented w.r.t. the traveled distance. In the figure, we present the activation of the "safety supervisor", where a value of 1 denotes an activation. Furthermore, in the figure, we present plots of the vehicle speed, steering wheel angle, and road curvature. Figure 6 illustrates the safe set (gray rectangle) and the one-step forward reachable set (bright green rectangle) at different time instances. These time instances are also shown in Figure 4, where s_3 occurs one simulation step after s_2. To improve the visibility of the one step forward reachable set, a zoomed-in view of the set is also provided in the figure.

As mentioned, the steering wheel angle, which is the input to the simulation, is defined in the Cartesian frame. As the curvature of the road increases, the steering wheel angle obtains the necessary value to negotiate the corner. Due to the miss-match in coordinate frames, this input will steer

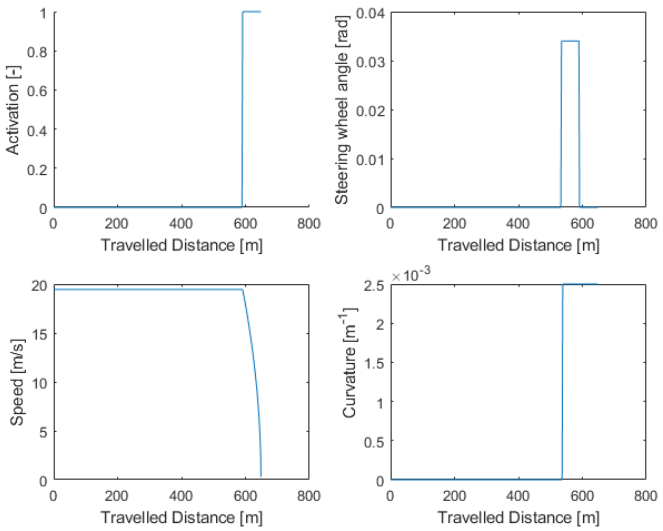

Fig. 5. Intervention of the safety supervisor

the vehicle outside the lane instead of following the desired centerline. As it can be seen, after some distance the vehicle states are deemed to not be safe and the "safety supervisor" intervenes which initiates the evasive maneuver. The decision that the vehicle states are not safe is made from the fact that the one-step reachable set is not a member of the safe set, as it can also be observed in Figure 6 at s_3. Since the evasive maneuver is defined as a constant brake event in a straight line, the steering wheel angle is brought zero in the first instance when the evasive maneuver is initiated. The evasive maneuver lasts until the vehicle is fully stopped, i.e., the vehicle speed is zero. What is an appropriate safe maneuver is part of future work, where for a heavy-duty vehicle it might be logical to always initiate a blind stop whenever the vehicle states are deemed not safe. We also believe that stopping in the road shoulder is an adequate alternative. Another alternative would be to have multiple safe sets corresponding to different safety maneuvers.

It can be concluded that the "safety supervisor" approach, that is based on calculating the safe set from the N-step backward reachability analysis, successfully detects when the nominal control input would lead to unsafe vehicle states and initiates an evasive maneuver to keep the vehicle in the desired operating bounds.

The numerical example is replicated in the IPG TruckMaker environment using a high fidelity vehicle model. For sake of brevity the results are omitted in the paper. The IPG TruckMaker environment is useful for further extensions of the approach where surrounding traffic is considered.

The computation time necessary to perform the online calculation is the time necessary to compute the one-step forward reachable set in addition to $4mnp$ flops, where m, n is the dimension of the Q matrix and p is the number of vertices of the one-step forward reachable set. The time necessary to compute the forward reachable set is jkl flops in the worst case, where j is the number of vertices of the state

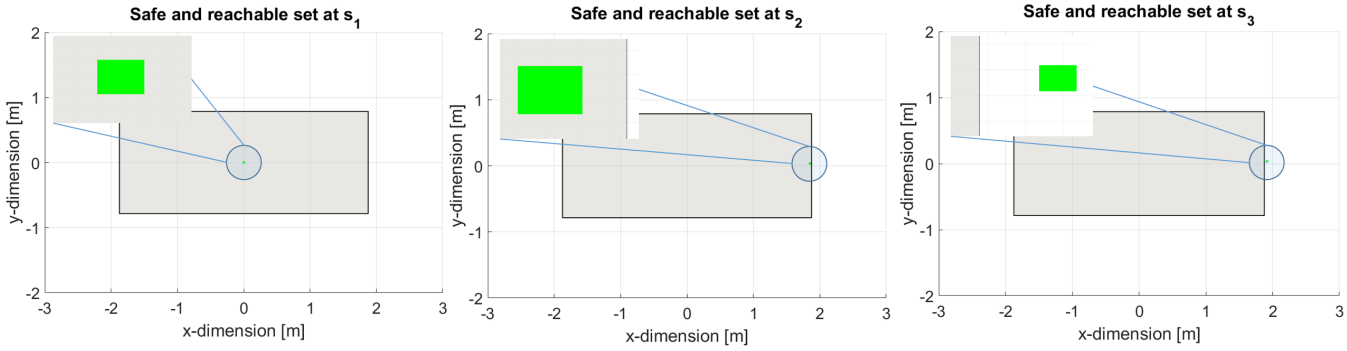

Fig. 6. Permissible (safe) and reachable sets at different points on the road

set, k the number of vertices of the input set and l the number of vertices of the disturbance set. In our setup, $k = 1$ since we use the exact input from the nominal controller, while the state and disturbance sets j and l are box bounded. This compares well with the current state of the art approaches [19], [21] and is promising for a real-time implementation which is part of future work.

VI. Conclusions and Future Work

We have introduced a concept that provides safety guarantees on an automated driving policy, by utilizing the backward reachability theory. Furthermore, we have proposed a method to adapt the permissible set online, based on the current disturbances and uncertainties. The concept is evaluated through a simple numerical example of highway driving with no participating traffic. Our ongoing research is dedicated to modeling the linearization errors coming from the nonlinear model and investigating their influence. Furthermore, future extensions would be to create and evaluate more representative scenarios, along with defining how to measure and model the disturbances and uncertainties that affect the system, and investigating what is the appropriate evasive maneuver. Another potential topic for future work is creating a vehicle model that has higher fidelity.

References

[1] Bagloee, Saeed Asadi, Madjid Tavana, Mohsen Asadi, and Tracey Oliver. "Autonomous vehicles: challenges, opportunities, and future implications for transportation policies." Journal of modern transportation 24, no. 4 (2016): 284-303.

[2] Singh, Santokh. Critical reasons for crashes investigated in the national motor vehicle crash causation survey. No. DOT HS 812 115. 2015.

[3] Nilsson, Julia, Mattias Brännström, Erik Coelingh, and Jonas Fredriksson. "Lane change maneuvers for automated vehicles." IEEE Transactions on Intelligent Transportation Systems 18, no. 5 (2016): 1087-1096.

[4] Ji, Jie, Amir Khajepour, Wael William Melek, and Yanjun Huang. "Path planning and tracking for vehicle collision avoidance based on model predictive control with multiconstraints." IEEE Transactions on Vehicular Technology 66, no. 2 (2016): 952-964.

[5] Hoel, Carl-Johan, Krister Wolff, and Leo Laine. "Automated speed and lane change decision making using deep reinforcement learning." In 2018 21st International Conference on Intelligent Transportation Systems (ITSC), pp. 2148-2155. IEEE, 2018.

[6] Sallab, Ahmad EL, Mohammed Abdou, Etienne Perot, and Senthil Yogamani. "Deep reinforcement learning framework for autonomous driving." Electronic Imaging 2017, no. 19 (2017): 70-76.

[7] Kalra, Nidhi, and Susan M. Paddock. "Driving to safety: How many miles of driving would it take to demonstrate autonomous vehicle reliability?." Transportation Research Part A: Policy and Practice 94 (2016): 182-193.

[8] Shalev-Shwartz, Shai, Shaked Shammah, and Amnon Shashua. "On a formal model of safe and scalable self-driving cars." arXiv preprint arXiv:1708.06374 (2017).

[9] Fraichard, Thierry, and Hajime Asama. "Inevitable collision states—A step towards safer robots?." Advanced Robotics 18, no. 10 (2004): 1001-1024.

[10] Nilsson, Jonas, Jonas Fredriksson, and Anders CE Ödblom. "Verification of collision avoidance systems using reachability analysis." IFAC Proceedings Volumes 47, no. 3 (2014): 10676-10681.

[11] Althoff, Matthias. "Reachability analysis of nonlinear systems using conservative polynomialization and non-convex sets." In Proceedings of the 16th international conference on Hybrid systems: computation and control, pp. 173-182. 2013.

[12] Althoff, Matthias, and John M. Dolan. "Online verification of automated road vehicles using reachability analysis." IEEE Transactions on Robotics 30, no. 4 (2014): 903-918.

[13] Ames, Aaron D., Samuel Coogan, Magnus Egerstedt, Gennaro Notomista, Koushil Sreenath, and Paulo Tabuada. "Control barrier functions: Theory and applications." In 2019 18th European Control Conference (ECC), pp. 3420-3431. IEEE, 2019.

[14] Ames, Aaron D., Xiangru Xu, Jessy W. Grizzle, and Paulo Tabuada. "Control barrier function based quadratic programs for safety critical systems." IEEE Transactions on Automatic Control 62, no. 8 (2016): 3861-3876.

[15] Xu, Xiangru, Paulo Tabuada, Jessy W. Grizzle, and Aaron D. Ames. "Robustness of control barrier functions for safety critical control." IFAC-PapersOnLine 48, no. 27 (2015): 54-61.

[16] Borrelli, Francesco, Alberto Bemporad, and Manfred Morari. Predictive control for linear and hybrid systems. Cambridge University Press, 2017.

[17] Klintberg, Emil, Magnus Nilsson, Lars Johannesson Mårdh, and Ankit Gupta. "A primal active-set minimal-representation algorithm for polytopes with application to invariant-set calculations." In 2018 IEEE Conference on Decision and Control (CDC), pp. 6862-6867. IEEE, 2018.

[18] Ljung, Lennart. Model validation and model error modeling. Linköping University Electronic Press, 1999.

[19] Berntorp, Karl, Richard Bai, Karl Fredrik Erliksson, Claus Danielson, Avishai Weiss, and Stefano Di Cairano. "Positive invariant sets for safe integrated vehicle motion planning and control." IEEE Transactions on Intelligent Vehicles (2019).

[20] Pek, Christian, Markus Koschi, and Matthias Althoff. "An online verification framework for motion planning of self-driving vehicles with safety guarantees." In AAET-Automatisiertes und vernetztes Fahren. 2019.

[21] Pek, Christian, and Matthias Althoff. "Ensuring Motion Safety of Autonomous Vehicles through Online Fail-safe Verification." In Robotics: Science and Systems–Pioneers Workshop. 2019.