MITSUBISHI ELECTRIC RESEARCH LABORATORIES
https://www.merl.com

Decision Making for Automated Driving by Reachability of Parameterized Maneuvers

Di Cairano, Stefano; Skibik, Terrence; Vinod, Abraham P.; Weiss, Avishai; Berntorp, Karl

TR2023-084 July 11, 2023

Abstract

We consider a decision making system for automated driving that has the objective of determining what maneuvers are feasible for the current vehicle, route, and traffic conditions. For the maneuvers determined to be feasible, a motion planner can then generate the trajectories that achieve the corresponding goals, without the risk of wasting computations in searching for a trajectory of an impossible maneuver. We solve the decision making problem by constructing backward reachable sets for goals and collision areas, based on maneuvers that are generated by dynamical models with decision parameters. Online, we only need to check the existence of parameter values that provide membership of the state-parameter vector in a goal reachable set, and non-membership in all collision reachable sets, which entails simple and fast computations. We evaluate the method in scenarios involving lane change and braking maneuvers.

World Congress of the International Federation of Automatic Control (IFAC) 2023

Mitsubishi Electric Research Laboratories, Inc.
201 Broadway, Cambridge, Massachusetts 02139

Decision Making for Automated Driving by Reachability of Parameterized Maneuvers

Stefano Di Cairano * **Terrence Skibik** ** **Abraham P. Vinod** *
Avishai Weiss * **Karl Berntorp** *

* *Mitsubishi Electric Research Laboratories, Cambridge, MA 02139 USA, (e-mails: {dicairano,vinod,weiss,berntorp}@merl.com).*
** *Dep. Electrical, Computer & Energy Engineering, University of Colorado, Boulder, CO 80303 USA, (e-mail: terrence.skibik@colorado.edu)*

Abstract: We consider a decision making system for automated driving that has the objective of determining what maneuvers are feasible for the current vehicle, route, and traffic conditions. For the maneuvers determined to be feasible, a motion planner can then generate the trajectories that achieve the corresponding goals, without the risk of wasting computations in searching for a trajectory of an impossible maneuver. We solve the decision making problem by constructing backward reachable sets for goals and collision areas, based on maneuvers that are generated by dynamical models with decision parameters. Online, we only need to check the existence of parameter values that provide membership of the state-parameter vector in a goal reachable set, and non-membership in all collision reachable sets, which entails simple and fast computations. We evaluate the method in scenarios involving lane change and braking maneuvers.

Keywords: Automotive, Automated Driving Systems, Reachability, Constrained Control

1. INTRODUCTION

Automated driving requires rapidly determining the next vehicle action among those allowed by the road rules, while recognizing that some of these choices may be rendered unachievable by the current traffic. The ego vehicle may be operating in a travel direction with two lanes, close to a number of other vehicles, approaching a stop line. While several maneuvers may be allowed, e.g., follow the lane, change lane, or decelerate to a stop, some may be infeasible due to other vehicles or traffic rules.

Automated driving systems are often divided in multiple layers with different reaction times, decision horizons, and computation budgets. In the prototypical architecture shown in Fig. 1, the vehicle control at the lower level tracks trajectories generated by the motion planner in the middle layer. Due to route lengths and quickly varying traffic conditions, such trajectories span only a short segment of the travel. Thus, the decision making in the top layer provides to the motion planner one or more maneuvers, sometimes also called driving modes, and their associated goals, i.e., the possible next waypoints. Hence, the decision making must provide maneuvers with achievable goals, so that the motion planner the motion planner does not waste computation budget trying to achieve impossible goals.

Several methods for decision making have been proposed based on rules, optimization, or machine learning, see, e.g., (Buehler et al., 2009; Gu et al., 2016; Galceran et al., 2017; Esterle et al., 2018; You et al., 2018; Hubmann et al., 2018; Schwarting et al., 2018), and the references therein. In our previous works (Ahn et al., 2020b,a) we proposed decision making by set reachability, where a maneuver was deemed achievable if (i) a goal region in the vehicle state space can be reached within a given finite horizon; and (ii) the vehicle is safe from collision with other vehicles and from violating traffic rules. We determined whether the maneuver can be achieved in the current traffic conditions by leveraging capture sets and backward reachable sets.

While successfully validated, the method in (Ahn et al., 2020b) separately tests candidate trajectories for safety, by performing collision checking on the points of the trajectories, and for liveness, by evaluating membership of the state in the reachable set of the next goal or in a safety invariant. Hence, (Ahn et al., 2020b) requires checking set membership conditions for all the points of the candidate trajectories, which may be computationally expensive, and candidate trajectories for the different scenarios.

In this paper we consider trajectories obtained by parameterized maneuvers that are produced by dynamical systems with decision parameters kept constant during the maneuver. Decision making checks if there exists parameter values that ensure reaching the goal within a given finite horizon, while avoiding collisions with other vehicles. By leveraging reachable sets in the state-parameter space, maneuver feasibility is checked based only on the initial conditions and parameter values, i.e., without checking any other point along the maneuver. Furthermore, collision avoidance and goal reachability computations are integrated, which simplifies implementation.

Set-based methods have been investigated for several operations in automated driving, such as motion planning, safety verification, and robust control, see, e.g., (Althoff and Dolan, 2014; Gao et al., 2014; Berntorp et al., 2019;

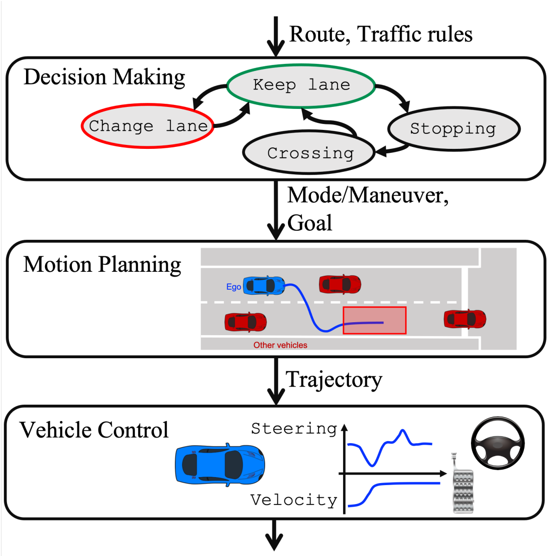


Fig. 1. Multi-layers automated driving system: current (green) and next (red) maneuver/mode in decision making, goal (red) and trajectory (blue) in motion planning, and control signals (blue) in vehicle control.

Koschi and Althoff, 2020), and references therein. A related use of reachable sets appeared in (Li et al., 2020), which enforced collision avoidance by modifying a given command signal, yet did not consider goal reachability due to focusing only on safety.

In what follows, in Section 2 we introduce the maneuver, goal, and obstacle models, in Section 3 we describe the reachable set construction and the conditions for maneuver feasibility, and in Section 4 we discuss implementation aspects and a simple robustness metric. Section 5 reports simulation scenarios and Section 6 the conclusions.

Notation: $\mathbb{Z}$ and $\mathbb{Z}_+$ are the sets of integers, and positive integers, we denote intervals as $\mathbb{Z}_{[a,b)} = \{z \in \mathbb{Z} : a \leq z < b\}$, and similarly for real numbers $\mathbb{R}$. We denote the Minkowski set sum by $\oplus$, and the logical "or" by $\vee$. For vectors x, y, $[x]_i$ denotes the i^{th} component, $(x, y) = [x' \ y']'$ the stacking, and inequalities are intended componentwise. For a discrete-time signal $x \in \mathbb{R}^n$, x_t is the value at sampling instant, $x_{k|t}$ denotes the predicted value k steps ahead of t, based on data at t, and $x_{0|t} = x_t$.

2. MANEUVERS AND THEIR MODELS

The Decision Maker (DM) determines the maneuvers that the automated driving system executes, e.g., between lane changing, stopping, lane keeping, etc. Rather than uniquely determining the maneuver, we assess what maneuvers are feasible for the current ego vehicle and traffic conditions. Then, the motion planner computes motion plans for all or a subset of those, to possibly choose the most suitable trajectory.

2.1 Ego Vehicle Maneuvers, Goals, and Obstacles

A maneuver is defined by a motion model of the ego vehicle with a vector of parameters for accomplishing it, and a goal set, which is the condition that the ego vehicle must achieve for completion. The set of maneuvers is

$$\mathcal{M} = \{M^{(i)}\}_{i=1}^m = \{(\Sigma^{(i)}(r^{(i)}), \Gamma^{(i)})\} \tag{1}$$

where m is the number of maneuvers, $M^{(i)}$ is the i^{th} maneuver, $\Sigma^{(i)}(r^{(i)})$ is the motion model, with parameter vector $r^{(i)}$, and $\Gamma^{(i)}$ is the goal for the i^{th} maneuver.

The goal $\Gamma^{(i)}$ is defined by the couple

$$\Gamma^{(i)} = (\tilde{\mathcal{P}}_g^{(i)}, \Sigma_g^{(i)}) \tag{2}$$

where $\tilde{\mathcal{P}}_g$ is the goal region, the region of space where the goal is achieved, and $\Sigma_g^{(i)}$ is the motion model. Other vehicles or other actors present on the road, which we name obstacles, are defined by the couple

$$O^{(h)} = (\tilde{\mathcal{P}}_o^{(h)}, \Sigma_o^{(h)}), \tag{3}$$

where $h \in \mathbb{Z}_{[1,n_o]}$, n_o is the number of obstacles, $\tilde{\mathcal{P}}_o^{(h)}$ is the exclusion zone of the obstacle, that is, the region where collision between ego vehicle and obstacle occurs, and $\Sigma_o^{(h)}$ is the motion model for the obstacle exclusion region.

2.2 Ego Vehicle Maneuver Motion Models

Depending on the maneuver, different motion models may be used. All such motion models are relatively simple, capturing the relevant vehicle behaviors, while allowing to determine maneuver feasibility with simple and fast computations. Because of this, and since we consider normal, i.e., comfortable and non-aggressive, driving, the DM uses linear models. Motion planner and controller normally use more accurate models to determine and track the ego vehicle motion.

Longitudinal Motion Model For the longitudinal motion, DM uses a velocity controlled linear model

$$\dot{p}_x = v_x \tag{4a}$$

$$\dot{v}_x = -\frac{1}{\tau_v} v_x + \frac{1}{\tau_v} r_v \tag{4b}$$

where p_x, v_x are the longitudinal position and velocity, r_v is the velocity command, and $\tau_v > 0$ is a time constant that may change depending on the aggressiveness of the maneuver. A constant value of the velocity command, i.e., the reference velocity, r_v may be used as parameter in model (4).

Lateral Motion Model For lateral motion, closed-loop kinematic or dynamic lateral motion models may be used. However, we may also represent lane changes as trajectories of $2^{nd}/3^{rd}$ order systems (Di Cairano et al., 2012). Thus, we model the lateral motion with respect to the road centerlane as trajectories of a transfer function from commanded to actual lateral position

$$G_y(s) = \frac{1}{\left(\frac{s^2}{\omega_y^2} + 2\frac{\zeta_y}{\omega_y} s + 1\right) \cdot (s\tau_y + 1)}, \tag{5}$$

where ω_y, ζ_y are the second order system natural frequency and damping, and τ_y is a time constant.

The values of ω_y, ζ_y, τ_y determine the lateral trajectories and can be selected based on time specifications, and may change for different maneuvers. System (5) is realized in state space form,

$$\dot{x}_y = [\,\dot{p}_y \ \dot{v}_y \ \dot{a}_y\,]' = A_y x_y + B_y r_y. \tag{6}$$

The reference lateral position, r_y, is used as parameter in (5), (6).

Braking Motion Model For significant braking, especially for stopping, we consider the model

$$\dot{p}_x = v_x \tag{7a}$$
$$\dot{v}_x = -r_a \tag{7b}$$

where $r_a > 0$ is the commanded braking deceleration, constant in the maneuver, and the parameter in model (7). The actuation dynamics can be easily included in (7).

2.3 Goal and Obstacle Motion Models

Goal Motion Model. The goal region is

$$\tilde{\mathcal{P}}_g = \{p: \; H_g[p_x - g_x \;\; p_y - g_y]' \leq K_g\},$$

that is $\tilde{\mathcal{P}}_g = [g_x \; g_y]' \oplus \bar{\mathcal{P}}_g$, i.e., a polyhedron centered at (g_x, g_y), the goal center longitudinal and lateral positions with respect to the (curvilinear) lane coordinates. The goal motion model Σ_g allows for representing moving goals (g_x, g_y). Here, Σ_g uses longitudinal velocity and constant lateral position with respect to the centerlane,

$$\dot{g}_x = v_g \tag{8a}$$
$$\dot{v}_g = 0 \tag{8b}$$
$$\dot{g}_y = 0, \tag{8c}$$

where v_g is the goal longitudinal velocity.

Obstacle Motion Model. The obstacle exclusion region is modeled by the polyhedron

$$\tilde{\mathcal{P}}_o = \{p: \; H_o[p_x - o_x \;\; p_y - o_y]' \leq K_g\},$$

that is $\tilde{\mathcal{P}}_o = [o_x \; o_y]' \oplus \bar{\mathcal{P}}_o$, i.e., a polyhedron centered at (o_x, o_y), the obstacle center longitudinal and lateral positions with respect to the (curvilinear) lane coordinates. The obstacle exclusion region accounts for the dimensions of the obstacle and the ego vehicle, i.e., a collision is avoided when the "point-mass" ego position is outside of the obstacle exclusion zone.

The obstacle position on the road, (o_x, o_y), is predicted according to the obstacle motion model Σ_o, which here is

$$\dot{o}_x = v_o \tag{9a}$$
$$\dot{v}_o = 0 \tag{9b}$$
$$\dot{o}_y = 0, \tag{9c}$$

i.e., a constant lateral position with respect to the centerline and constant longitudinal velocity v_o. A lateral motion of the obstacle may also be included.

2.4 Problem Definition

First, we state an assumption related to the information available to the decision making system.

Assumption 1. The ego vehicle has enough information from sensors to initialize the maneuver models (4), (6), (7), and the obstacle model (9). □

Assumption 1 is reasonable for models (4), (6), (7), (9), as it requires having information about quantities that are commonly measured by standard on-board and automated driving systems sensors. Assumption 1 provides an additional motivation for selecting simple motion models.

We now formalize the problem tackled in this paper.

Problem 1. Let a sampling period T_s and a maneuver horizon $N \in \mathbb{Z}_+$ be given. At any discrete time t, given a set of obstacles $\{O^{(h)}(t)\}_{h=1}^{n_o}$, a subset of maneuvers $\mathcal{M}(t) \subseteq \mathcal{M}$, where for each $M^{(i)} \in \mathcal{M}(t)$ the motion model $\Sigma^{(i)}$ is constructed from a combination of (4), (6), (7) with parameter vector $r^{(i)} \in \mathcal{R}^{(i)}$, $\mathcal{R}^{(i)}$ being the admissible set, determine for which maneuvers $M^{(i)} \in \mathcal{M}(t)$ there exist $r^{(i)} \in \mathcal{R}^{(i)}$ such that the goal $\Gamma^{(i)}$ is reached within N steps, while the ego vehicle does not enter any obstacle exclusion zone, i.e., $p_{k|t} \in \tilde{\mathcal{P}}_g^{(i)}$ for some $k \in \mathbb{Z}_{[0,N]}$, and $p_{k|t} \notin \tilde{\mathcal{P}}_o^{(h)}$, for all $h \in \mathbb{Z}_{[1,n_o]}$, $k \in \mathbb{Z}_{[0,N]}$. □

As introduced in Section 2.2, the components of the parameter vector are constant references/commands.

3. SET-BASED DECISION MAKING WITH PARAMETERIZED MANEUVERS

Next, we propose an algorithm that solves Problem 1, i.e., determines the feasibility of the maneuvers.

3.1 Definitions and Preliminary Results

We recall some basic definitions and results.

Definition 1. For a set $\mathcal{S}$ and system $x_{t+1} = f(x_t)$, the (1-step) backward reachable set is $\text{Pre}_f(\mathcal{S}) = \{x: \; f(x) \in \mathcal{S}\}$. The k-steps backward reachable set is recursively defined as $\text{Pre}_f^k(\mathcal{S}) = \text{Pre}_f(\text{Pre}_f^{k-1}(\mathcal{S}))$. □

When $\mathcal{S}$ in Definition 1 is a polyhedron and f is linear, the backward reachable sets enjoy some useful properties.

Result 1. Let $f(x_t) = Ax_t$,and $\mathcal{S} = \{x: H_0 x \leq K_0\}$, then $\text{Pre}_f(\mathcal{S}) = \{x: \; H_1 x \leq K_1\}$, i.e., also a polyhedron, where $H_1 = H_0 A$, $K_1 = K_0$. □

Thus, the computation of backward reachable sets for polyhedral sets and (discrete-time) linear systems involves only algebraic operations, optionally with linear programs to eliminate redundant hyperplanes.

3.2 Design: Construction of Achieving and Colliding sets

We use the backward reachable sets in Definition 1 for determining the values of the parameter vectors for which the maneuver: (i) achieves the goal, and (ii) does not collide with the obstacles. While the two requirements are of opposite nature (reaching-in versus staying-out), we use similar constructions, as described next.

First, we formulate the ego vehicle motion models of (4), (6), (7) and the goal and obstacle models (8), (9) in discrete-time with sampling period T_s. Then, at time t, for each maneuver i we construct the relative motion of ego-vehicle with respect to the goal/obstacles,

$$\Delta x_{k+1|t}^{(i),h} = \Delta A^{(i),h} \Delta x_{k|t}^{(i),h} + \Delta B^{(i),h} w_{0|t}^{(i),h} \tag{10a}$$
$$w_{0|t}^{(i),h} = \Psi^{(i),h} x_t^h + r_t^{(i)} \tag{10b}$$

where $h \in \mathbb{Z}_{[0,n_0]}$, so that (10) models the relative motion with respect to the goal for $h = 0$, and with respect to obstacle h, for $h \in \mathbb{Z}_{[1,n_0]}$. In (10), $\Delta x^{(i),h}$ is the state of the relative motion model and $w^{(i),h}$ is the relative parameter vector, which is constructed from the parameter vector $r^{(i)}$ and the state $x^{(i),h}$ of the ego vehicle and goal ($h = 0$) or obstacle ($h \in \mathbb{Z}_{[1,n_0]}$) at the beginning of the maneuver by (10b), where $\Psi^{(i),h}$ is a known matrix for the change of coordinates. From (10b), $w^{(i),h}$ is constant throughout a maneuver, i.e., $w^{(i),h}_{0|t}$ is applied at all steps $k \in \mathbb{Z}_{[0,N]}$.

Next, similar to a reference governor (Garone et al., 2017), we augment the state with the relative parameter vector subject to constant dynamics,

$$\xi_{k+1|t} = \Phi^{(i),h}\xi^{(i),h}_{k|t} = \begin{bmatrix} \Delta A^{(i),h} & \Delta B^{(i),h} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \Delta x^{(i),h}_{k|t} \\ w^{(i),h}_{k|t} \end{bmatrix}, \tag{11}$$

and construct the goal and obstacle sets for the augmented state in relative coordinates by:

i) lifting the polyhedra centered at the goal and obstacle coordinates, $\bar{\mathcal{P}}^{(i)}_g$, $\bar{\mathcal{P}}^h_o$ to the dimension of $\Delta x^{(i),h}$;

ii) intersecting the lifted sets with $\underline{\Delta x}^{(i),h}_{\min} \leq \Delta x^{(i),h} \leq \overline{\Delta x}^{(i),h}_{\max}$, i.e., a bounding box of the relative states for maneuver i with respect to the goal/obstacle h, to obtain $\hat{\mathcal{P}}^{(i),0}_g$, $\hat{\mathcal{P}}^{(i),h}_o$ that are compact;

iii) constructing $\mathcal{P}^{(i),0}_g = \hat{\mathcal{P}}^{(i),0}_g \times \mathcal{W}^{(i),0}$, $\mathcal{P}^{(i),h}_o = \hat{\mathcal{P}}^{(i),h}_o \times \mathcal{W}^{(i),h}$, where $\mathcal{W}^{(i),h}$ are the admissible sets for the relative parameter vector, constructed from $\mathcal{R}^{(i)}$ based on (10b), which results in sets for ξ in (10).

Thus, $\Delta x^{(i),0} \in \mathcal{P}^{(i),0}_g$ means that the ego is at the goal, and $\Delta x^{(i),h} \in \mathcal{P}^{(i),h}_o$, $h \in \mathbb{Z}_+$ means it is in the exclusion maregions of obstacle h, i.e., in a collision.

From (10), we compute the colliding sets as the k-steps backward reachable sets

$$\mathcal{C}^{(i),h}_k = \mathrm{Pre}^k_{\Phi^{(i),h}}(\mathcal{P}^{(i),h}_o),\ k \in \mathbb{Z}_{[0,N]},\ h \in \mathbb{Z}_{[1,n_o]}. \tag{12}$$

$\mathcal{C}^{(i),h}_k$ is the set of augmented states $\xi^{(i),h} = (\Delta x^{(i),h}, w^{(i),h})$ such that if maneuver i is executed with parameter vector $w^{(i),h}$ from initial state $\Delta x^{(i),h}_0 = \Delta x^{(i),h}$ of the relative motion model (10), $\Delta x^{(i),h}_{k|t} \in \mathcal{P}^{(i),h}_o$, i.e., the system is in collision after k steps. Thus, for maneuver i, the parameter vectors for $\Delta x^{(i),h}$ guaranteeing that collisions with obstacle h will not to occur for N steps, are

$$\mathcal{F}^{(i),h}_o(\Delta x^{(i),h}, x^h) = \left\{ r^{(i)} \in \mathcal{R}^{(i)} : (\Delta x^{(i),h}, \Psi^{(i),h}x^h + r^{(i)}) \notin \bigcup_{k=0}^{N} \mathcal{C}^{(i),h}_k \right\}. \tag{13}$$

Similarly, we compute the achieving sets as the k-steps backward reachable sets

$$\mathcal{A}^{(i),0}_k = \mathrm{Pre}^k_{\Phi^{(i),0}}(\mathcal{P}^{(i),0}_g),\ k \in \mathbb{Z}_{[0,N]}. \tag{14}$$

$\mathcal{A}^{(i),0}_k$ is the set of augmented states $\xi^{(i),0} = (\Delta x^{(i),0}, w^{(i),0})$ such that if maneuver i is executed with parameter vector $w^{(i),0}$ from initial state $\Delta x^{(i),0}_0 = \Delta x^{(i),0}$ of the relative motion model (10), $\Delta x^{(i),0}_k \in \mathcal{P}^{(i),h}_o$, i.e., the system is in the goal set after k steps. Thus, for maneuver i, the parameter vectors for $\Delta x^{(i),h}$ guaranteeing that the goal set is reached within N steps are

$$\mathcal{F}^{(i),0}_g(\Delta x^{(i),0}, x^0) = \left\{ r^{(i)} \in \mathcal{R}^{(i)} : (\Delta x^{(i),0}, \Psi^{(i),0}x^0 + r^{(i)}) \in \bigcup_{k=0}^{N} \mathcal{A}^{(i),h}_k \right\}. \tag{15}$$

Next we summarizes our solution to Problem 1.

Proposition 1. At time t, let $\mathcal{F}^{(i),0}_g(\Delta x^{(i),0})$, $\mathcal{F}^{(i),h}_o(\Delta x^{(i),h})$ be the set of feasible parameter ranges for maneuver $M^{(i)} \in \mathcal{M}(t) \subseteq \mathcal{M}$, obstacle set $\{O^{(h)}(t)\}^{n_o}_{h=1}$ and $r \in \mathcal{R}^{(i)}$, from (15), (13). Given $\Delta x^{(i),h}_t$, constructed from the state of the ego vehicle, obstacle, and goals, if

$$\mathcal{F}^{(i),0}_g(\Delta x^{(i),0}_t) \cap \left(\bigcap_{h=1}^{n_o} \mathcal{F}^{(i),h}_o(\Delta x^{(i),h}_t) \right) \neq \emptyset, \tag{16}$$

maneuver $M^{(i)}$ is admissible according to Problem 1. □

Condition (16) is expressed only on the initial state and parameter vector, as opposed to (Ahn et al., 2020b).

Parameterizing the maneuvers with a fixed parameter vector, e.g., a constant setpoint/command, simplifies the computations of the sets, and allows fast checking of (16) despite this being a non-convex set, as discussed next.

4. IMPLEMENTATION CONSIDERATIONS

Next we look at some of the implementation details that results in fast computation for condition (16).

The maneuver feasibility condition (16) involves a non-convex set. However, since we have parameterized the maneuvers, the computations are simplified by methods similar to those for reference governors (Garone et al., 2017). According to Result 1, for every $k \in \mathbb{Z}_{[0,N]}$, the achieving sets are the polyhedra $\mathcal{A}^{(i),0}_k = \{(\Delta x^{(i),0}, w^{(i),0}) : H^{(i),0}_k[\Delta x^{(i),0\prime}\ w^{(i),0\prime}]' \leq K^{(i),0}_k\}$ and the colliding sets are $\mathcal{C}^{(i),h}_k = \{(\Delta x^{(i),h}, w^{(i),h}) : H_k[\Delta x^{(i),h\prime}\ w^{(i),h\prime}]' \leq K^{(i),h}_k)\}$, for $h \in \mathbb{Z}_{[1,n_o]}$. Since the initial state in (16) is given, we take sections of the polyhedra at the known state value, resulting in

$$a^{(i),h}_k w^{(i),h} \leq b^{(i),h}_k,\ h \in \mathbb{Z}_{[0,n_0]},$$

where $h = 0$ for the achieving sets of the goal, and $h > 0$ for the colliding sets of obstacle h. Since the parameter vector $w^{(i),h}$ is low dimensional, gridding the values and testing them one by one is sufficient.

Often, a maneuver depends only on one component of the parameter vector, e.g., the reference deceleration in braking for (7), the reference velocity in lane keeping and lane changing for (4), (6) when the target lateral position is fixed at the centerlane of the target lane. When $r^{(i)}$ is a scalar, and so is $w^{(i),h}$, we compute

$$\overline{w}^{(i),h}_k = \max_{j:[a^{(i),h}_k]_j > 0} \frac{[b^{(i),h}_k]_j}{[a^{(i),h}_k]_j} \quad \underline{w}^{(i),h}_k = \min_{j:[a^{(i),h}_k]_j < 0} \frac{[b^{(i),h}_k]_j}{[a^{(i),h}_k]_j}$$

where j is the index for the rows of the vectors $a^{(i),h}_k$, $b^{(i),h}_k$. Then, we can discretize the 1-dimensional range of $\mathcal{R}^{(i)}$, obtaining $\{r^{(i)}(\ell)\}^{n_\ell}_{\ell=1}$, and evaluate the maneuver feasibility by checking that at least one value of $r^{(i)}$ is:

(i) included in the section of at least one goal achieving set, $\mathcal{A}_k$, at the current state $\Delta x^{(i),0}$,

$$\exists k \in \mathbb{Z}_{[0,N]}: \ \Psi^{(i),0} x^0_{0|t} + r^{(i)}(\ell) \in [\underline{w}^{(i),0}_k, \overline{w}^{(i),0}_k] \tag{17a}$$

(ii) excluded from the sections of all the colliding sets, $\mathcal{C}_k$, at the current state $\Delta x^{(i),0}$ for all the obstacles

$$\Psi^{(i),0} x^0_{0|t} + r^{(i)}(\ell) \in \Big((-\infty, \underline{w}^{(i),h}_k) \vee (\overline{w}^{(i),h}_k, +\infty)\Big), \quad \forall k \in \mathbb{Z}_{[0,N]}, \ \forall h \in \mathbb{Z}_{[1,n_O]}, \tag{17b}$$

Remark 1. In (17) we did not include the case $[a^{(i),h}_k]_j = 0$. If for some j, $[a^{(i),h}_k]_j = 0$, then if $[b^{(i),h}_k]_i < 0$, the corresponding set is empty, while if $[b^{(i),h}_k]_i \geq 0$ the inequality is trivially satisfied. □

In practice, if the objective is only to verify the feasibility of condition (16) according to Problem 1, one can stop as soon as a value $r^{(i)}(\ell)$ that satisfies (17) is found. Characterizing all feasible values of $r^{(i)}(\ell)$ may be useful for generating a trajectory reference for the motion planner that allows for a initializing the search, hence reducing the motion planner computations.

After the set of feasible parameter vectors $\mathcal{R}^{(i)}_f \subseteq \mathcal{R}^{(i)}$ that satisfies (16) has been determined, the most desirable parameter vector can be selected as

$$r^{(i)*} = \arg \min_{r \in \mathcal{R}^{(i)}_f} J^{(i)}(r, \theta^{(i)}) \tag{18}$$

where $\theta^{(i)}$ is a vector containing information on the desired behavior, such as desired velocity, deceleration, etc, and $J^{(i)}$ is a maneuver-dependent cost function encoding the desirability of the maneuver. Once again, (18) may be solved on the discrete set $\mathcal{R}_f$ by a direct search.

An important metric to choose a reference trajectory may be its robustness,and one way to evaluate it, is to assess how large perturbations on the parameter vector still results in a successful maneuver completion. Thus, we define the robustness radius of $r^{(i)} \in \mathcal{R}^{(i)}_f$ as

$$\delta_{r^{(i)}} = \max\{\delta \in \mathbb{R}_+ : \ r^{(i)} + \varsigma \in \mathcal{R}^{(i)}_f, \ \forall \varsigma, |\varsigma| \leq \delta\}. \tag{19}$$

For discretized $\mathcal{R}_f$, the conditions (19) are checked only for points included in $\{r^{(i)}(\ell)\}_{\ell=1}^{n_\ell}$, i.e., assuming an equispaced discretization, for ς such that $r^{(i)} + \varsigma \in \mathcal{R}^{(i)}$. The non equispaced discretization case entails the same operations but a slightly more involved definition. The robustness radius can be included as part of the objective in the cost function $J^{(i)}$ in (18).

5. SIMULATIONS

We evaluate the decision making approach in scenarios on a straight road with 2 lanes in the ego vehicle travel direction. First, we consider a lane change scenario. We allow 3 lane change maneuvers, where the lateral position is the output of 3^{rd} order systems as in (5) with r_y set to the centerlane of the next lane, each with different settling time and overshoot, and dubbed *cautious*, *normal*, *aggressive* in order of decreasing settling time, i.e., slower lateral movement. The goal is to complete a lane change from about 10m to about 120m ahead of the vehicle, within a finite horizon of $N = 20$ steps, with $T_s = 0.25$s, i.e., 5s, commanding velocities $r_v \in [13, 23]$m/s, where the range is discretized in 100 points with a resolution 0.1m/s.

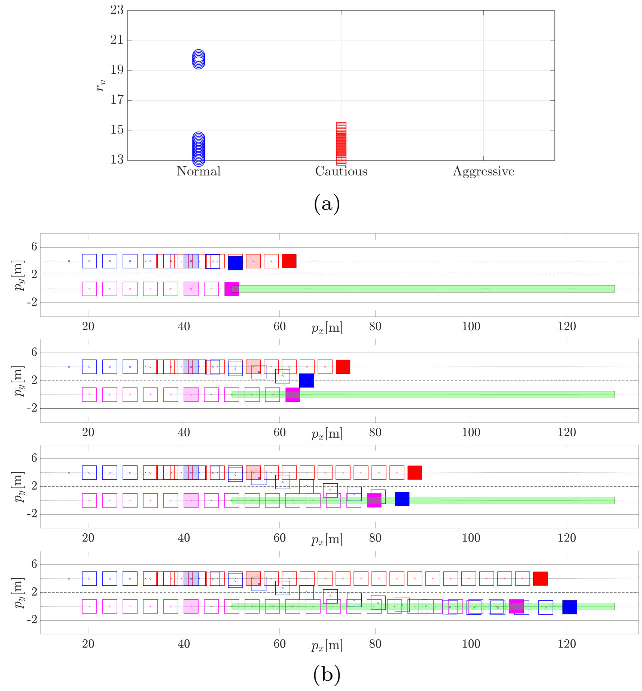
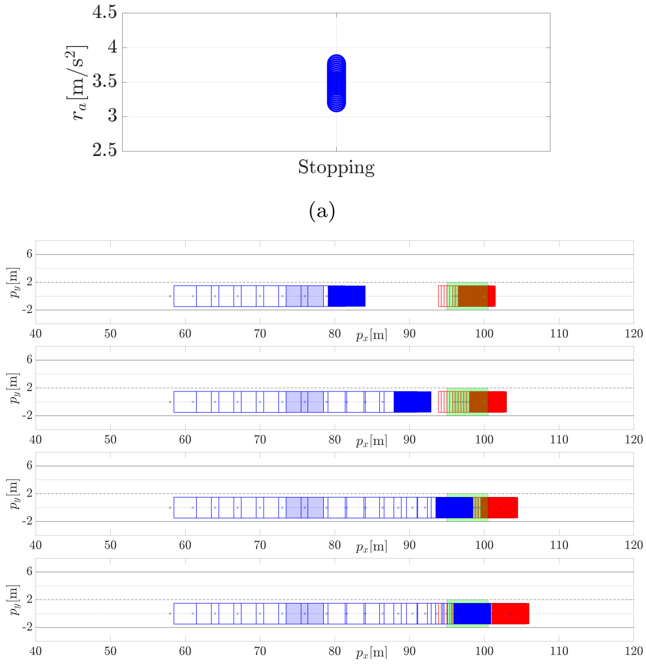


Fig. 2. Lane change secenario with 3 feasible maneuvers. (a) allowed values of commanded velocity r_v for each maneuver. (b) execution of the *normal* maneuver at maximum feasible r_v. Ego (blue) and other (red, purple) vehicles: current (solid), past (border only), and at maneuver start (shaded) positions. Region of lane change completion (green). Snapshots after $t = 0.5, 1.25, 2.25, 4$s from maneuver start.

Fig. 2 reports the results for a case with two other vehicles, one in the lane to the right of the ego vehicle, moving at the same speed of 17m/s and with same longitudinal position, and the other in the same lane of the ego vehicle, 9m ahead, and moving at 15m/s. Fig. 2(a) shows that a lane change is feasible with 2 of the 3 maneuvers for different commanded velocities. The *aggressive* maneuver is not feasible, as it moves too rapidly towards the opposite lane, before clearing ahead of the vehicle in such lane. The *cautious* maneuver can only change lane behind the vehicle in the right lane, i.e., by slowing down, while the *normal* maneuver can change lane both ahead, by speeding up, and behind, by slowing down, the vehicle in the right lane, showing also the non-convexity of the admissible set of target velocities. Fig. 2(b) shows the execution of the *normal* maneuver for its maximum allowed target velocity $r_v = 20$m/s, which completes the lane change ahead of the vehicle in the right lane, while avoiding collisions and completing the lane change in the designated area.

In Fig. 3 we show another scenario where the ego vehicle is driving at 12m/s, and has to reach a stop in a target area that starts 80m ahead, while another vehicle is slowly (1.5m/s) departing from it, i.e., stopping at an intersection while a preceding vehicle starts crossing. The DM uses a single maneuver with the braking motion model (7) where the reference deceleration command $r_a \in [1, 5]$m/s^2 is the parameter, discretized with resolution of 0.01m/s^2.



(b)

Fig. 3. Stopping scenario with 1 maneuver. (a) feasible values of commanded acceleration r_a. (b) execution of the maneuver with minimum r_a. Ego (blue) and other (red) vehicles: current (solid), past (border only), and at maneuver start (shaded) positions. Stop region (green). Snapshots after $t = 0.5, 1.5, 2.5, 3.5$s from maneuver start.

Fig. 3(a) shows that the maneuver is feasible, and Fig. 3(b) shows the reference motion for the least deceleration $r_a = 3.2\text{m/s}^2$, for which the vehicle stops following closely the departing vehicle. The total time for the discretization at 0.1m/s, i.e., 100 points per maneuver, used in Fig. 2 is less than 7.5ms, i.e., 2.5ms per maneuver, for checking all the points in a non-optimized Matlab 2021*b* implementation, on a 2020 MacBook Pro, with Intel i5 processor and 16GB of RAM, which is in general 10-40 times slower than an equivalent C implementation, e.g., from non-optimized code generation. The computational burden remains almost constant up to 1000 discretization points, since the evaluation of (17) is inexpensive after the upper and lower bounds have been determined, which is done only once for a given initial state.

6. CONCLUSIONS

We have presented a decision making method that determines feasibility of maneuvers to provide the next action and corresponding goal to a motion planner for automated driving by using reachable set over parameterized maneuvers. The obtained method is simple to implement, fast to compute, and provides guarantees on goal achievability while avoiding collisions, i.e., liveness and safety.

REFERENCES

Ahn, H., Berntorp, K., and Di Cairano, S. (2020a). Cooperating modular goal selection and motion planning for autonomous driving. In *59th IEEE Conf. Dec. and Control*, 3481–3486.

Ahn, H., Berntorp, K., Inani, P., Ram, A.J., and Di Cairano, S. (2020b). Reachability-based decision-making for autonomous driving: Theory and experiments. *IEEE Trans. Con. Sys. Tech.*, 29(5), 1907–1921.

Althoff, M. and Dolan, J.M. (2014). Online verification of automated road vehicles using reachability analysis. *IEEE Trans. Robotics*, 30(4), 903–918.

Berntorp, K., Bai, R., Erliksson, K.F., Danielson, C., Weiss, A., and Di Cairano, S. (2019). Positive invariant sets for safe integrated vehicle motion planning and control. *IEEE Trans. Intelligent Vehicles*, 5(1), 112–126.

Buehler, M., Iagnemma, K., and Singh, S. (2009). *The DARPA urban challenge: autonomous vehicles in city traffic*, volume 56. Springer.

Di Cairano, S., Tseng, H.E., Bernardini, D., and Bemporad, A. (2012). Vehicle yaw stability control by coordinated active front steering and differential braking in the tire sideslip angles domain. *IEEE Trans. Control Systems Technology*, 21(4), 1236–1248.

Esterle, K., Hart, P., Bernhard, J., and Knoll, A. (2018). Spatiotemporal motion planning with combinatorial reasoning for autonomous driving. In *Proc. Int. Conf. Intelligent Transportation Systems*, 1053–1060.

Galceran, E., Cunningham, A.G., Eustice, R.M., and Olson, E. (2017). Multipolicy decision-making for autonomous driving via changepoint-based behavior prediction: Theory and experiment. *Autonomous Robots*, 41(6), 1367–1382.

Gao, Y., Gray, A., Tseng, H.E., and Borrelli, F. (2014). A tube-based robust nonlinear predictive control approach to semiautonomous ground vehicles. *Vehicle System Dynamics*, 52(6), 802–823.

Garone, E., Di Cairano, S., and Kolmanovsky, I. (2017). Reference and command governors for systems with constraints: A survey on theory and applications. *Automatica*, 75, 306–328.

Gu, T., Dolan, J.M., and Lee, J. (2016). Automated tactical maneuver discovery, reasoning and trajectory planning for autonomous driving. In *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, 5474–5480.

Hubmann, C., Schulz, J., Becker, M., Althoff, D., and Stiller, C. (2018). Automated driving in uncertain environments: planning with interaction and uncertain maneuver prediction. *IEEE Tr. Intell. Veh.*, 3(1), 5–17.

Koschi, M. and Althoff, M. (2020). Set-based prediction of traffic participants considering occlusions and traffic rules. *IEEE Trans. Intelligent Vehicles*, 6(2), 249–265.

Li, N., Han, K., Girard, A., Tseng, H.E., Filev, D., and Kolmanovsky, I. (2020). Action governor for discrete-time linear systems with non-convex constraints. *IEEE Control Sys. Lett.*, 5(1), 121–126.

Schwarting, W., Alonso-Mora, J., and Rus, D. (2018). Planning and decision-making for autonomous vehicles. *Ann. Rev. Control, Robotics, Aut. Sys.*, 1(1), 187–210.

You, C., Lu, J., Filev, D., and Tsiotras, P. (2018). Highway traffic modeling and decision making for autonomous vehicle using reinforcement learning. In *IEEE Intell. Vehicles Symp.*, 1227–1232.