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

Parametrized Maneuvers Governor for Decision Making in Automated Driving

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

TR2024-086 July 02, 2024

Abstract

For automated driving, decision making determines the next maneuver that the vehicle should execute, for which the motion planner will generate a trajectory. The feasibility of the maneuver depends on the current conditions of the vehicle, the route, and the traffic. Thus, decision making must determine which maneuvers are feasible with relatively simple calculations, so that the motion planner, which performs more time-consuming calculations, can succeed in computing the trajectories that achieve the corresponding goals. We propose an approach to solve the decision making problem based on ideas from the reference governor. Our method constructs backward reachable sets for goals and collision areas for maneuvers that are generated by dynamical models parametrized by target values of vehicle motion quantities. Online, the reference governor determines 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. The resulting online computations are simple and fast, allowing solution of the decision making process at higher rate and with minimal resources as required for standard automotive computing platforms. Furthermore, the method can provide reference maneuvers to guide the motion planning in determining the actual trajectory, can include robustness metrics, and is extended to handle uncertainty in the motion of the obstacles to be avoided. We show simulation results in scenarios involving lane change, braking at intersections, and obstacles with changing velocity.

Nonlinear and Constrained Control - Applications, Synergies, Challenges and Opportunities. 2024

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

Parametrized Maneuvers Governor for Decision Making in Automated Driving

Stefano Di Cairano[0000-0002-2363-2807],
Terrence Skibik,
Abraham P. Vinod[0000-0002-7955-9629],
Avishai Weiss[0000-0002-4261-3038],
Karl Berntorp,
Yuichi Okura

Abstract For automated driving, decision making determines the next maneuver that the vehicle should execute, for which the motion planner will generate a trajectory. The feasibility of the maneuver depends on the current conditions of the vehicle, the route, and the traffic. Thus, decision making must determine which maneuvers are feasible with relatively simple calculations, so that the motion planner, which performs more time-consuming calculations, can succeed in computing the trajectories that achieve the corresponding goals. We propose an approach to solve the decision making problem based on ideas from the reference governor. Our method constructs backward reachable sets for goals and collision areas for maneuvers that are generated by dynamical models parametrized by target values of vehicle motion quantities. Online, the reference governor determines 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. The resulting online computations are simple and fast, allowing solution of the decision making process at higher rate and with minimal resources as required for standard automotive computing platforms. Furthermore, the method can provide reference maneuvers to guide the motion planning in determining the actual trajectory, can include robustness metrics, and is extended to handle uncertainty in the motion of the obstacles to be avoided. We show simulation results in scenarios involving lane change, braking at intersections, and obstacles with changing velocity.

Stefano Di Cairano, Karl Berntorp, Abraham P. Vinod, Avishai Weiss
Mitsubishi Electric Research Laboratories (MERL), Cambridge, MA, USA
e-mail: {dicairano,karl.o.berntorp}@ieee.org,{vinod,weiss}@merl.com

Terrence Skibik
Dep. Electrical, Computer & Energy Engineering, University of Colorado, Boulder, CO, USA, interning at MERL at the time of this research.
e-mail: terrence.skibik@colorado.edu

Yuichi Okura
Advanced Technology R&D Center, Mitsubishi Electric Corporation, Amagasaki, Japan
e-mail: Okura.Yuichi@dh.MitsubishiElectric.co.jp

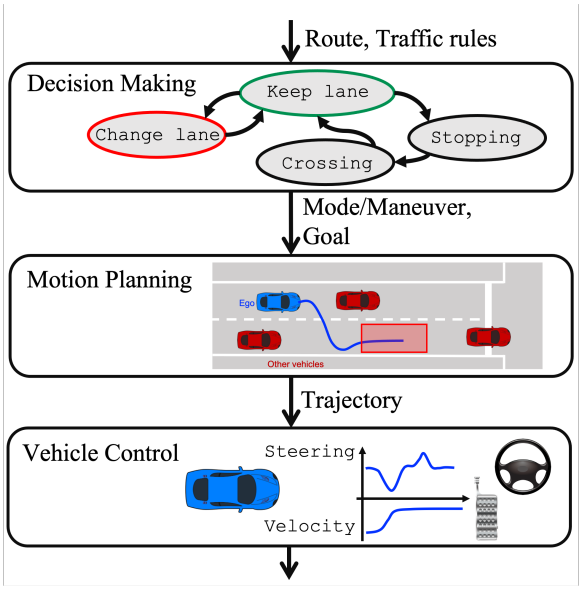


Fig. 1 Multi-layer architecture of the automated driving system: current (green) and next (red) maneuver/mode in decision making, goal (red) and trajectory (blue) in motion planning, and control signals (blue) in vehicle control.

1 Introduction

In order to achieve full autonomy [23], automated driving systems must operate a vehicle during long travels in changing scenarios, environments and traffic conditions. An effective way to achieve this is to decompose the planning and control for the entire travel into several smaller sub-problems such that, if all are solved, the vehicle achieves its final destination, while satisfying all the traffic rules and retaining safety.

Thus, the automated driving system can be implemented as a multi-layer control architecture, where the different layers have different reaction times, decision horizons and computation budgets. Figure 1 shows a prototypical example of this control architecture. At the lower level, the *vehicle control (VC)* is responsible for controlling the vehicle through the different actuation mechanisms, i.e., steering, brakes, throttle, to track given trajectories. Such trajectories are computed by the *motion planner (MP)* in the middle layer. Due to the length of the travel and the rapidly changing traffic conditions, the motion planner cannot generate a trajectory from the initial point of travel to the final destination, but rather generates only a short segment of the travel. To enable that, the route is commonly divided into segments, as seen in car navigation systems, and within each segment the *decision making (DM)* in the upper layer provides to the motion planner one or more *maneuvers*, sometimes also called *driving modes*, and their associated goals for the current segment, i.e., the possible

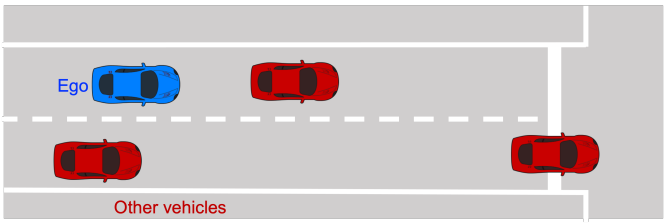


Fig. 2 The ego vehicle (blue) may execute several maneuvers: lane keeping, lane changing, stopping. However, not all may be feasible due to the other vehicles (red).

next waypoints, for which the trajectory must be computed. If the architecture in Figure 1 is properly designed, the sequence of motion plans will steer the vehicle from its initial point to the desired destination while correctly behaving in road and traffic, and the vehicle will be controlled to precisely track them.

For determining the maneuvers and their goals, decision making must consider the current conditions of the vehicle, the traffic and the road rules, because certain maneuvers may or may not be feasible, depending on 0 those. Fig. 2 shows a scenario where the ego vehicle is operating in a travel direction with two lanes, close to other vehicles, and approaching a stop line. Several maneuvers are allowed, e.g., continue following the lane, change lane, or decelerate to a stop, but some may be infeasible, e.g., changing lane may be impossible due to the positions and velocities of other vehicles, and it may be too early to begin braking to stop.

Thus, the decision making needs to determine not just a maneuver, but a *feasible maneuver*, i.e., such that the motion planner will be able to compute a trajectory from current conditions to goal conditions, while satisfying constraints imposed by ego vehicle motion, obstacle avoidance, and road rules satisfaction. If the maneuver provided by the decision maker was not achievable according to motion, traffic and rules, the motion planner would waste computations trying to achieve an impossible goal.

The brute force approach of computing one trajectory for each maneuver, discarding maneuvers where the planner fails, and choosing the maneuver by comparing the trajectories successfully computed, wastes significant amount of computational resources, which are limited in automotive platforms [11], and is actually feasible only for simple scenarios, primarily related to highway driving. Instead, driving in city-like scenarios, where multiple maneuvers are possible, and under realistic constraints imposed by automotive embedded platforms, usually requires more refined approaches.

Several methods for decision making in automated driving have been proposed based on rules, optimization, or machine learning [10, 13, 14, 17, 18, 25, 27], see also the references therein. However, these methods often do not provide both, the guarantee of a feasible motion plan for the selected maneuver, and a computationally-light implementation, especially under uncertainty.

Recently, [1, 2] proposed using set reachability in decision making for automated driving, where a maneuver was deemed achievable if (*i*) a goal region defined in the vehicle state space is reachable within a given finite horizon; and, (*ii*) the vehicle avoids collisions and violating traffic rules. In [2], achievability of the next maneuver and avoidance of collisions and traffic rules violation were obtained by backward reachable sets [9] and capture sets [26], respectively. The approach was extended in [1] to determine all feasible maneuvers with their corresponding goals, and then a motion planner determined the best maneuver, among the feasible ones, and the corresponding best trajectory.

While successfully validated in experiments [2], such a method still shows some limitations. Specifically, it verifies goal achievability and collision avoidance sequentially, by separately testing candidate trajectories for safety, by performing collision checking on the points of the trajectories, and for liveness, by ensuring that the initial state remains in the next goal reachable set or in a collision avoidance invariant. The latter amounts to checking set membership conditions for all the points of several candidate trajectories, which may be computationally expensive, and requires pre-computing candidate trajectories for the different maneuvers that may be executed.

In this paper, we propose a method to overcome such limitations. The proposed method uses dynamical systems with parameters that are target values for certain motion quantities and that remain constant throughout the maneuver. Online, a reference governor-like algorithm checks if there exist target values for which the dynamical system trajectory reaches the goal within a given finite horizon, while avoiding collisions with other vehicles. Such a check is performed on the reachable sets of the goal and the obstacles, i.e., only the initial vehicle state and constant parameter values are checked. Thus, we do not need to check for collisions and achievement of goal for all points of the trajectory of the maneuver, which saves a significant amount of computation. In addition, collision avoidance and goal achievability are integrated and use similar computations, which further simplifies the implementation of the method. Finally, the method enables simple metrics of robustness, supports modeling of uncertainty in the behavior of other vehicles, as well as possibly in the ego vehicle's motion, and provide with minimal additional computations a reference to the motion planner that may guide in generating the actual vehicle trajectory.

Among additional prior works related to this paper, set-based methods have been investigated for several uses in automated driving, such as motion planning, safety verification, and robust control, see, e.g., [3, 5, 15, 19, 24], and references therein. A related use of reachable sets appeared in [21], which determined when to modify a command signal for collision avoidance, while here we consider goal reachability for maneuver feasibility determination.

This paper is organized as 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 the implementation of the decision making as a reference governor, including some computational aspects, maneuver reference selection, and extension to robustness with respect to other

vehicle motion. Section 5 reports simulation scenarios for both known and uncertain behavior of the obstacles, and Section 6 reports 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 between them 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, Models, and Problem Definition

First we introduce models for ego vehicle, obstacle, and maneuver goals. Since the decision making (DM) method sits at the top layer in the architecture in Figure 1, it involves decisions over long horizons with relatively low update rates. As a consequence, simplified models are sufficient, since the actual trajectory and vehicle motion will be refined by the motion planner and the vehicle controller that use more detailed models, at higher rates, and over shorter horizons. For more details on the overall automated driving architecture and the integration of the different layers see [1, 2, 4, 6].

The DM determines the maneuvers for the motion planner according to a motion model of the ego vehicle with a vector of parameters, where a maneuver is completed successfully if the ego vehicle state enters a given goal set. Hence, the set $\mathcal{M}$ of maneuvers is composed of triples

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

where $M^{(i)}$ is the i^{th} maneuver, $\Sigma^{(i)}(r^{(i)})$ is the motion model with parameter vector $r^{(i)}$, $\Gamma^{(i)}$ is the goal for the i^{th} maneuver, and m is the total number of maneuvers. Some examples of maneuvers are *change lane*, *follow lane*, *stop at line*, *follow traffic*, *cross intersection*, *turn left/right*, *merge in/out*. The goal $\Gamma^{(i)}$ is defined by the set of couples

$$\mathcal{G} = \left\{\Gamma^{(i)}\right\}_{i=1}^{m} = \left\{\left(\tilde{\mathcal{P}}_g^{(i)}, \Sigma_g^{(i)}\right)\right\}_{i=1}^{m}, \tag{2}$$

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 of the goal, which allows for modeling moving goal regions. The obstacles, which may be other vehicles or actors on the road, are similarly defined by the set of couples

$$\mathcal{Q} = \left\{O^{(h)}\right\}_{h=1}^{n_o} = \left\{\left(\tilde{\mathcal{P}}_o^{(h)}, \Sigma_o^{(h)}\right)\right\}_{h=1}^{n_o}, \tag{3}$$

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, i.e., the region where a collision between ego vehicle and obstacle occurs, and $\Sigma_o^{(h)}$ is the motion model for the exclusion region of the obstacle, which enables modeling moving obstacles.

2.1 Motion Models for Maneuvers of Ego Vehicle

The ego vehicle maneuver models $\Sigma^{(i)}(r^{(i)})$, $i \in \mathbb{Z}_{[1,m]}$, describe the vehicle motion state $x^{(i)}$ when maneuver $M^{(i)}$ is executed with parameter vector value $r^{(i)}$,

$$\dot{x}^{(i)} = f^{(i)}(x^{(i)}, r^{(i)}), \tag{4}$$

which includes both the vehicle response and the action of the controllers ensuring that the vehicle tracks the maneuver.

The maneuver models (4) will be different for different maneuvers, including possibly the state dimensions. However, all are expected to favor simplicity even if some approximation may be necessary, as they need to capture only the most relevant vehicle behaviors while allowing to assess the maneuver over long horizons with simple and fast computations. Since we consider normal driving, i.e., comfortable and non-aggressive, we use linear models in (4) for assessing feasibility of the maneuvers. The actual maneuvers will be generated and executed by the MP and VC using higher precision models, more degrees of freedom, and higher update rates.

For driving maneuvers such as lane following and changing lane, the ego vehicle motion model includes the longitudinal and lateral motion. While such motions are physically coupled [7], under normal driving conditions the coupling is mild and a controller will be able to execute maneuvers obtained from decoupled longitudinal and lateral motions. The longitudinal motion can be formulated as for cruise control [22] by the linear model

$$\dot{p}_x = v_x, \tag{5a}$$

$$\dot{v}_x = -\frac{1}{\tau_v} v_x + \frac{1}{\tau_v} r_v, \tag{5b}$$

where p_x, v_x are longitudinal position and velocity, r_v is the velocity command, and $\tau_v > 0$ is the time constant for tracking velocity commands. Model (5) can be represented in standard form as

$$\dot{x}_x = \begin{bmatrix} \dot{p}_x & \dot{v}_x \end{bmatrix}' = A_x x_x + B_x r_v. \tag{6}$$

A constant value of the velocity command, i.e., reference velocity, r_v can be used as parameter in models (5), (6). The lateral motion of the ego vehicle can be built from a kinematic or dynamic lateral motion model. However, since we are considering controlled maneuvers, the motion during the maneuver can also be the output of a dynamical system describing the response of the lateral position due to a commanded

lateral position. As briefly discussed in [12], a $2^{nd}/3^{rd}$ order system is suitable to represent most driver-executed lane changes, so here we use the 3^{rd} order transfer function from lateral position command to lateral position,

$$G_y(s) = \frac{1}{\left(\frac{s^2}{\omega_y^2} + 2\frac{\zeta_y}{\omega_y}s + 1\right) \cdot (s\tau_y + 1)}, \tag{7}$$

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 such as overshoot, rise time, settling time, and actuation responsiveness, which in turn determine the aggressiveness of the lateral motion of the vehicle, and may change for different maneuvers. Sometimes, the lateral motion exhibits behavior well represented by a non-minimum phase zero, which can also be included in (7). System (7) can be realized in state space form as the linear system

$$\dot{x}_y = \left[\, \dot{p}_y \;\; \dot{v}_y \;\; \dot{a}_y \,\right]' = A_y x_y + B_y r_y. \tag{8}$$

A constant value of the lateral position command, i.e., reference lateral position, r_y, can be used as parameter in models (7), (8).

For maneuvers with significant braking, such as stopping at an intersection or in a queue, the lateral position with respect to the centerlane is fixed, $\dot{p}_y = 0$, and we represent the longitudinal motion by the double integrator model

$$\dot{p}_x = v_x, \tag{9a}$$

$$\dot{v}_x = -r_a, \tag{9b}$$

which is suitably re-written as

$$\dot{x}_b = \left[\, \dot{p}_y \;\; \dot{v}_y \,\right]' = A_b x_b + B_b r_a, \tag{10}$$

where $r_a > 0$ is the commanded deceleration due to braking. In (10), we ignored the actuation dynamics, which are much faster than the longitudinal dynamics, but they can be easily included, e.g., as a first order lag from commanded to actual deceleration. The commanded deceleration r_a can be used as parameter in model (10).

2.2 Maneuver Constraints

In certain cases, imposing an admissible range of parameter vectors $r \in \mathcal{R}^{(i)}$ may be sufficient to encode maneuvers that satisfy constraints on vehicle motion and dynamics states. However, in some cases the ego vehicle models (6), (8), (10) require imposing explicit constraints on the ego vehicle states, which we model as

$$x_y \in \mathcal{X}_y,\; x_x \in \mathcal{X}_x,\; x_b \in \mathcal{X}_b, \tag{11}$$

where $\mathcal{X}_y$, $\mathcal{X}_x$, $\mathcal{X}_b$ are suitable polyhedra representing the regions where the constraints are satisfied. □

2.3 Models for Maneuver Goals and Obstacles

Goals $\Gamma^{(i)}$, $i \in \mathbb{Z}_{[1,m]}$ and obstacles $O^{(h)}$, $h \in \mathbb{Z}_{[1,n_o]}$ are represented by sets that evolve over time according a function $\dot{f}_s$, i.e., with a little abuse of notation,

$$\dot{\mathcal{S}} = f_s(\mathcal{S}, d), \tag{12}$$

where f_s models an instantaneous change in the set $\mathcal{S}$, $d \in \mathcal{D}$ is a vector of exogenous disturbances for the obstacle and $\mathcal{D}$ is the obstacle disturbance set.

For computational tractability, we model a goal region by a constant polyhedron, the goal zone, with a goal center defined by (g_x, g_y), the longitudinal and lateral positions with respect to the (curvilinear) lane coordinates,

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

Then, a moving goal region is modeled by including a motion model for the goal center Σ_g. This allows for representing goals that change over time, such as in certain lane change maneuvers or in following behaviors. For simplicity, we use a goal center motion model with constant longitudinal velocity and constant lateral position with respect to the road centerlane,

$$\dot{g}_x = v_g \tag{13a}$$

$$\dot{v}_g = 0 \tag{13b}$$

$$\dot{g}_y = 0, \tag{13c}$$

where v_g is the goal longitudinal velocity.

The obstacles are modeled similar to the goals by a fixed polyhedron, the collision region, with an obstacle center defined by (o_x, o_y), the longitudinal and lateral positions with respect to the (curvilinear) lane coordinates, that define the exclusion region of the obstacles, i.e., where a collision occurs,

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

As obstacles may be moving, we include a motion model for the obstacle center to predict the obstacle future positions. We use the vector $d_v \in \mathcal{D}$ to represent changes in the obstacle behavior, e.g., changes in velocity or in lateral position. In this paper we consider only changes in longitudinal velocity,

$$\dot{o}_x = v_o \tag{14a}$$

$$\dot{v}_o = -\frac{1}{\tau_v} v_o + \frac{1}{\tau_v} d_v \tag{14b}$$

$$\dot{o}_y = 0, \tag{14c}$$

where the lateral position with respect to the centerline is constant and the velocity tracks a target d_v with time constant τ_v. Such a time constant is assumed equal for all vehicle as it represents the average vehicle behavior in tracking a changing velocity setpoint. On the other hand, using the same time constant in (5) and (14) allows for some simplifications, but it is not necessary.

Remark 1 The goal and obstacle regions account for the physical dimension of both the goal/obstacle and the ego vehicle. Hence, a collision is avoided when the vehicle "point-mass" position is outside of the obstacle exclusion zone, and a goal is achieved when it is in the goal zone.

Remark 2 It is fairly standard not to model changes in the obstacle lateral position, i.e., the obstacle remains in the same lane, motivating the choice for (14c). However, it is straightforward to include such changes in (14) by adding a lateral position model tracking a lateral position setpoint d_y, either similar to (7) or even as a first order model. Then, the disturbance vector that determines the possible motions of the obstacles becomes $d = [d_v,\ d_y]' \in \mathcal{D}$.

2.4 Problem Definition: Decision Making for Automated Driving

Next, we formalize the problem that is solved in this paper. To this end, we first 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* (6)*,* (8)*,* (10) *and the obstacle model* (14) *with* $d_v = v_o$. □

Assumption 1 is reasonable for models (6), (8), (10), (14), as it requires having information about quantities that are commonly measured by conventional vehicle on-board sensors, such as radars, cameras, and lidars. Assumption 1 provides a further motivation for keeping the motion models as simple as possible. According to Assumption 1, we are able to initialize (14) by $d_v = v_o$, but we may use also different values. The actual obstacle disturbance $d_v \in \mathcal{D}$ does not need to be known, although the obstacle disturbance set $\mathcal{D}$ is known. The goal model (13) is fully known since it describes the ego vehicle target.

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 $\{O^{(h)}(t)\}_{h=1}^{n_o}$ with motion models (14) for $d_v \in \mathcal{D}$, where d_v is not known and $\mathcal{D}$ is known;

- 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 (6), (8), (10) with parameter vector $r^{(i)} \in \mathcal{R}^{(i)}$, $\mathcal{R}^{(i)}$ being its admissible set;
  - the goal is $\Gamma^{(i)}(t)$ with motion model (13);

determine for which maneuvers $M^{(i)} \in \mathcal{M}(t)$ there exist $r^{(i)} \in \mathcal{R}^{(i)}$ such that the goal $\Gamma^{(i)}(t)$ is reached within N sampling periods, while satisfying the constraints on the ego vehicle (11) and 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 discussed in Section 2.1, in Problem 1 the parameter vector is a vector of constant commands/setpoints that must be chosen to satisfy safety, i.e., collision avoidance, and liveness, i.e., goal achievement, constraints. This suggests that reference governor [16] may provide a suitable method for solving Problem 1.

3 Set Construction for Automated Driving Decision Making

In order to solve Problem 1 using concepts from set-based methods [8] and reference governor [16], we recall some basic definitions and results, see [8, 9].

Definition 1 (Backward reachable set) Given set $\mathcal{S}$ and system $x_{t+1} = f(x_t)$, the (1-step) backward reachable set is the set of states that are in $\mathcal{S}$ after evolving through f for one step, i.e., $\mathrm{Pre}_f(\mathcal{S}) = \{x : \ f(x) \in \mathcal{S}\}$. The k-steps backward reachable set is recursively defined as $\mathrm{Pre}_f^k(\mathcal{S}) = \mathrm{Pre}_f(\mathrm{Pre}_f^{k-1}(\mathcal{S}))$, $\mathrm{Pre}_f^0(\mathcal{S}) = \mathcal{S}$. □

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

Result 1 (Backward reachable set for linear system)**.** *Consider a linear system $f(x_t) = Ax_t$ and a polyhedron $\mathcal{S} = \{x : H_0 x \leq K_0\}$. The 1-step backward reachable set is the polyhedron,* $\mathrm{Pre}_f(\mathcal{S}) = \{x : \ H_1 x \leq K_1\}$*, where $H_1 = H_0 A$, $K_1 = K_0$.* □

The computation of backward reachable sets for polyhedral sets and (discrete-time) autonomous linear systems involves only algebraic operations, possibly with solutions of linear programs to eliminate redundant inequalities [9, 16]. This is in contrast with non-autonomous systems, $x_{t+1} = f(x_t, u_t)$, for which the backward reachable set involves projection from the (x, u) to the x space, which is a potentially very expensive operation even in the linear case [9].

3.1 Construction of Achieving and Colliding Sets

To solve Problem 1 we need to determine whether for a maneuver it is possible to (i) achieve the goal, and (ii) avoid collisions with obstacles. We use the backward

reachable sets in Definition 1 for constructing the sets where the goal is eventually achieved and the sets where a collision with obstacles eventually occurs.

For efficiently constructing such reachable sets, first we construct relative motion models for ego vehicle and obstacles, and for ego vehicle and goals, using (6), (8), (10), (13), (14) formulated in discrete-time with sampling period T_s. At time t, for each maneuver i the relative motion of ego-vehicle with respect to the goal/obstacles,

$$\Delta x_{k+1|t}^{(i),h} = \Delta A^{(i),h} \Delta x_{k|t}^{(i),h} + \Delta B^{(i),h} w_t^{(i),h} + \Delta E^{(i),h} d_t^{(i),h} \tag{15a}$$

$$w_t^{(i),h} = \Psi^{(i),h} x_t^h + r_t^{(i)}, \tag{15b}$$

where the index $h \in \mathbb{Z}_{[0,n_0]}$ is such that if $h = 0$, (15) models the relative motion with respect to the goal, and if $h \in \mathbb{Z}_{[1,n_0]}$, $r \in \mathbb{R} \subseteq \mathbb{R}^{n_r}$, $d^{(i),h} \in \mathbb{R}^{n_d^{(i),h}} \subseteq \mathcal{D}^{(i),h}$, (15) models the relative motion with respect to obstacle $h \in \mathbb{Z}_+$. In (15), $\Delta x^{(i),h}$ is the state of the relative motion model and $w^{(i),h}$ is the relative parameter vector, which is constructed from the parameter vector $r^{(i),h}$, and from the state of the goal ($h = 0$) or obstacle ($h \geq 1$) at the beginning of the maneuver, x_t^h, by (15b), where $\Psi^{(i),h}$ is a known matrix. According to (15b), $w^{(i),h}$, $d^{(i),h}$ are constant throughout the maneuver.

Using the approach of the reference governor [16], we augment the state with the relative parameter vectors and with the obstacle motion disturbance, both with constant dynamics, resulting in the lifted model

$$\xi_{k+1|t} = \Phi^{(i),h} \xi_{k|t}^{(i),h} = \begin{bmatrix} \Delta A^{(i),h} & \Delta B^{(i),h} & \Delta E^{(i),h} \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \Delta x_{k|t}^{(i),h} \\ w_{k|t}^{(i),h} \\ d_{k|t}^{(i),h} \end{bmatrix}. \tag{16}$$

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

i) constructing the set $\bar{\mathcal{X}}^{(i),0}$ of states for the relative motion model of the ego vehicle with respect to the goal from (11) and (15), where the ego vehicle constraints are satisfied, i.e., $\Delta x^{(i),0} \in \mathcal{X}^{(i),0} \Rightarrow x_j \in \mathcal{X}_j$, for $j \in \{x, y, b\}$;

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

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

iv) constructing $\mathcal{P}_g^{(i),0} = \hat{\mathcal{P}}_g^{(i),0} \times \mathcal{W}^{(i),0} \times \{0\}$, $\mathcal{P}_o^{(i),h} = \hat{\mathcal{P}}_o^{(i),h} \times \mathcal{W}^{(i),h} \times \mathcal{D}^{(i),h}$, where $\mathcal{W}^{(i),h}$ are the admissible sets for the relative parameter vector constructed from $\mathcal{R}^{(i)}$ based on (15b), and $\mathcal{X}^{(i),0} = \bar{\mathcal{X}}^{(i),0} \times \mathbb{R}^{n_r+n_d}$, which results in sets of the appropriate dimension for ξ.

The conditions $\Delta x^{(i),0} \in \mathcal{P}_g^{(i),0}$ on relative ego vehicle-goal state are such that, when satisfied, the ego vehicle is at the goal, and similarly the conditions $x^{(i),h} \in \mathcal{P}_o^{(i),h}$, $h > 0$ on relative ego vehicle-goal states are such that, when satisfied, the ego vehicle is in collision with obstacle h.

From (15), we compute the *colliding sets* as the k-steps backward reachable sets

$$\mathcal{C}_k^{(i),h} = \mathrm{Pre}^k_{\Phi^{(i),h}}(\mathcal{P}_o^{(i),h}),\ k \in \mathbb{Z}_{[0,N]},\ h \in \mathbb{Z}_{[1,n_o]}, \tag{17}$$

where N is the maneuver duration in sampling periods, $\mathcal{C}_k^{(i),h}$ is the set of augmented states $\xi^{(i),h} = (\Delta x^{(i),h}, w^{(i),h}, d^{(i),h})$ such that if maneuver i is executed with parameter vector $w(t) = w^{(i),h}$ from initial state $\Delta x_{0|t}^{(i),h} = \Delta x^{(i),h}$ and for $d(t) = d^{(i),h}$ with relative motion model (15), $\Delta x_{k|t}^{(i),h} \in \mathcal{P}_o^{(i),h}$, i.e., the system is in collision with obstacle $h \in \mathbb{Z}_{[1,n_o]}$ after k steps. Thus, for maneuver i, the feasible set of parameter vectors for state $\Delta x^{(i),h}$ with respect to obstacle h is

$$\mathcal{F}_o^{(i),h}(\Delta x^{(i),h}, x^h) = \left\{ r^{(i)} \in \mathcal{R}^{(i)} : (\Delta x^{(i),h}, \Psi^{(i),h} x^h + r^{(i)}, d^{(i),h}) \notin \bigcup_{k=0}^{N} \mathcal{C}_k^{(i),h},\ \forall d^{(i),h} \in \mathcal{D}^{(i),h} \right\}, \tag{18}$$

which guarantees that collisions will not to occur in the future N steps. Similarly, we compute the *achieving sets* as the k-steps backward reachable sets

$$\mathcal{A}_k^{(i),0} = \mathrm{Pre}^k_{\Phi^{(i),0}}(\mathcal{P}_g^{(i),0}) \cap \mathcal{X},\ k \in \mathbb{Z}_{[0,N]}, \tag{19}$$

where N is the maneuver duration in sampling periods, $\mathcal{A}_k^{(i),0}$ is the set of augmented states $\xi^{(i),0} = (\Delta x^{(i),0}, w^{(i),0}, 0)$ such that if maneuver i is executed with parameter vector $w^{(i),0}$ from initial state $\Delta x_0^{(i),0} = \Delta x^{(i),0}$ of the relative motion model (15), $\Delta x_k^{(i),0} \in \mathcal{P}_o^{(i),h}$, i.e., the system is in the goal set after k steps, and the constraints on the ego vehicle states are satisfied. Thus, for maneuver i, the feasible set of parameter vectors for $\Delta x^{(i),h}$ with respect to the goal is

$$\mathcal{F}_g^{(i),0}(\Delta x^{(i),0}, x^0) = \left\{ r^{(i)} \in \mathcal{R}^{(i)} : (\Delta x^{(i),0}, \Psi^{(i),0} x^0 + r^{(i)}, 0) \in \bigcup_{k=0}^{N} \mathcal{A}_k^{(i),h} \right\}, \tag{20}$$

which guarantees that the goal set is reached within N steps, while satisfying constraints in ego vehicle motion (11).

Fig. 3 shows sections of the colliding/achieving sets for the case of the lateral and longitudinal motion models, (6), (8), and for the case of braking motion (10), respectively.

The next proposition summarizes our solution to Problem 1.

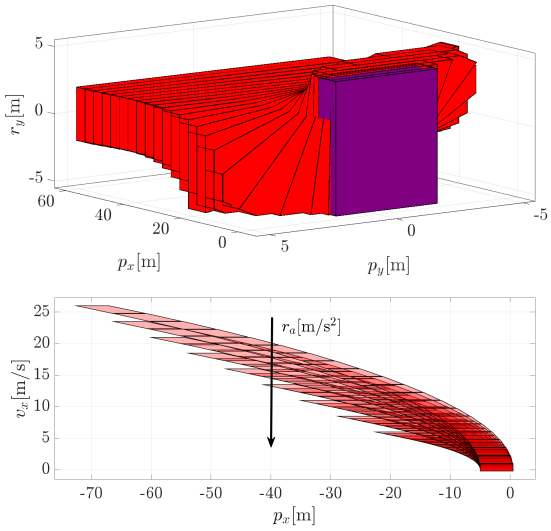


Fig. 3 Sections of reachable sets for different maneuvers. Top: longitudinal and lateral motion with initial set (purple). Bottom: braking motion for deceleration commands r_a increasing in the arrow direction.

Theorem 1 *At time $t \in \mathbb{Z}_{0+}$, for maneuver $M^{(i)} \in \mathcal{M}(t) \subseteq \mathcal{M}$, obstacle set $\{O^{(h)}(t)\}_{h=1}^{n_o}$, and obstacle disturbance set $\mathcal{D}$, let $\mathcal{F}_g^{(i),0}(\Delta x^{(i),0})$, $\mathcal{F}_o^{(i),h}(\Delta x^{(i),h})$ be the set of parameter values $r \in \mathcal{R}^{(i)}$ from* (20), (18)*. Given $\Delta x_t^{(i),h}$, constructed from the state of the ego vehicle, obstacle, and goals, maneuver $M^{(i)}$ is admissible according to Problem 1 if and only if*

$$\mathcal{F}_g^{(i),0}(\Delta x_t^{(i),0}, x_t^0) \cap \left(\bigcap_{h=1}^{n_o} \mathcal{F}_o^{(i),h}(\Delta x_t^{(i),h}, x_t^h) \right) \neq \emptyset. \tag{21}$$

□

Proof. Assume that (21) holds, and let $r \in \mathcal{F}_g^{(i),0}(\Delta x_t^{(i),0}, x_t^0) \cap (\bigcap_{h=1}^{n_o} \mathcal{F}_o^{(i),h}(\Delta x_t^{(i),h}, x_t^h))$. Since $r \in \mathcal{F}_g^{(i),0}(\Delta x_t^{(i),0}, x_t^0)$, when r is applied from $\Delta x_{0|t}^{(i),0} = \Delta x_t^{(i),0}$, there exists $\bar{k} \in \mathbb{Z}_{[1,N]}$ such that $\Delta x_{\bar{k}|t}^{(i),0} \in \mathcal{P}_g^{(i),0}$, and $\Delta x_{k,t}^{(i),0} \in \mathcal{X}^{(i),0}$ for all $k \in \mathbb{Z}_{[0,\bar{k}]}$ due to the definition of backward reachable sets, (18) and (20). Hence $x_{j,k|t}^{(i),0} \in \mathcal{X}_j$ for $j \in \{y, x, b\}$ and $p_{k|t} \in \tilde{\mathcal{P}}_g^{(i)}$, i.e., the maneuver satisfies the ego vehicle constraints and achieves the goal in k steps, where $k \in \mathbb{Z}_{[1,N]}$. Furthermore, since $r \in \mathcal{F}_o^{(i),h}(\Delta x_t^{(i),h}, x_t^h)$, for all $h \in \mathbb{Z}_{[1,n_o]}$, when r is applied from $\Delta x_{0|t}^{(i),h} = \Delta x_t^{(i),h}$, $\Delta x_{k|t|}^{(i),h} \notin \mathcal{P}_o^{(i),h}$, for all $k \in \mathbb{Z}_{[1,N]}$, due to the definition of backward reachable sets. Hence, $p_{k|t} \notin \tilde{\mathcal{P}}_o^{(h)}$, i.e., the maneuver does not collide with obstacles for at least

N steps. Thus, the conditions for a feasible maneuver according to Problem 1 are satisfied. Conversely, when (21) does not hold, due to the definition of backward reachable sets either the trajectory of the maneuver will not reach the goal within N steps while satisfying the ego vehicle constraints, or it will be in collision with an obstacle at least at one sampling instant. Hence, the maneuver will not be admissible according to Problem 1 □

Condition (21) provides a way to solve P1 evaluating only conditions on the initial state and parameter vector, and does not require forward motion simulation. This is a major advantage over many methods in robotics [20] and also with our method in [2] that requires collision checking of candidate trajectories to ensure safety with respect to the obstacles.

Remark 3 The ego vehicle constraints do not need to be included also in the reachable set iteration (17) because the intersection in (21) will render them redundant, i.e., (18) excludes all maneuvers that collide with the obstacles, regardless whether they satisfy vehicle constraints or not, since the ones that do not, are eliminated in (21) by (20). This allows for simplifying computation since it avoids checking several unnecessary constraints. □

Parameterizing the maneuvers with a constant setpoint/command, simplifies the computations of the sets, since state-parameter sets do not grow in dimension at each iteration, which means that projection is not needed. Instead if a general control signal were used, either the set dimension grows at each step, or it requires projection, which is computationally expensive and prone to failure when N becomes large. Furthermore, our approach allows for fast checking of (21) despite this generally involving non-convex sets, as discussed later.

4 Reference Governor for Decision Making

The feasibility condition (21) in Section 3 determines the states and parameter values for which a goal of the maneuver is achieved, while avoiding collisions with other vehicles. Leveraging (21), the decision making can also provide a reference trajectory to the motion planner, which allows to focus to search for a motion plan in a smaller region of the space, thus reducing the motion planner computations. This is particularly important for automated driving applications due to the limited capabilities of automotive computational platforms [11].

Since in the approach proposed in Section 3 the maneuver parameters are constant setpoints for system variables, we can use an approach similar to the reference governor [16] to determine the parameter value for the maneuver, which results in a reference trajectory for the motion planner. For maneuver $i \in \mathcal{M}$, we determine

$$r_*^{(i)} = \arg\min_r \; J^{(i)}(r, \theta^{(i)}) \tag{22a}$$

$$\text{s.t.} \quad r \in \mathcal{F}_g^{(i),0}(\Delta x_t^{(i),0}, x_t^0) \cap \left(\bigcap_{h=1}^{n_o} \mathcal{F}_o^{(i),h}(\Delta x_t^{(i),h}, x_t^h) \right), \tag{22b}$$

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. Solving (22) for different maneuvers $i \in \mathcal{M}$ may result in multiple feasible maneuvers for the motion planner, similar to [1], which is useful for multi-mode planners such as [6]. For selecting a single maneuver, the feasible ones are compared as

$$r_t = r_*^{(i^*)}, \quad i^* = \arg\min_i \; J_a \left(\{(r_*^{(i)}, \theta^{(i)}, i), \}_{i \in \mathcal{M}_f} \right) \tag{23}$$

where $\mathcal{M}_f$ are the feasible maneuver, i.e., the maneuvers for which (22) returns a non-empty, finite value, and J_a is a cost function for comparing the maneuvers in terms of desirability, based also on the solutions of (22). The solution of (23) is straightforward as only few maneuvers are available, and hence direct comparison of the values of J_a will be sufficient.

The decision making determines the maneuver parameters to satisfy inclusion constraints, the achieving sets, and exclusion constraints, the colliding sets, similar to a reference governor that determines the virtual reference to satisfy the system constraints. However, while the reference governor aims at remaining closer to a specified user reference, and operates repeatedly in time, the decision making does not have initial reference and operates once when a maneuver needs to be determined. Next we discuss the solution of (22), which appears more challenging but can be greatly simplified by leveraging its structure.

4.1 Maneuver determination for deterministic obstacle motion

Consider first the case where the motion of the obstacle is deterministic, i.e., at any time t, $\mathcal{D} = \{d_t\}$, d_t is known, and we can assume without loss of generality that $d_t = 0$. For this case, the challenge in computing (22) is satisfying constraint (22b), since it involves a non-convex set constructed from complements and unions of convex sets. However, since we have parameterized the maneuvers, the computations are simplified by resorting to methods similar to those for reference governors [16].

According to Result 1, for every $k \in \mathbb{Z}_{[0,N]}$, the achieving sets and the colliding sets are the polyhedra, $\mathcal{A}_k^{(i),0} = \{(\Delta x^{(i),0}, w^{(i),0}) : H_k^{(i),0}[\Delta x^{(i),0\prime} \; w^{(i),0\prime}]' \leq K_k^{(i),0}\}$ and $C_k^{(i),h} = \{(\Delta x^{(i),h}, w^{(i),h}) : H_k[\Delta x^{(i),h\prime} \; w^{(i),h\prime}]' \leq K_k^{(i),h})\}$, for $h \in \mathbb{Z}_{[1,n_o]}$, respectively. Since the initial state in (22) is given, we take sections of the polyhedra at the known state value, resulting in the lower dimensional polyhedron

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

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 usually low dimensional, a simple approach of gridding or random shooting and testing the values may be sufficient to solve (22).

Further simplification is possible if the parameter vector for a maneuver is one-dimensional, i.e., the maneuver is parametrized by a scalar. The case of one-dimensional parameter vector is quite common, since braking maneuvers based on (10) only use the the reference deceleration, lane keeping and lane changing maneuvers only use the reference longitudinal velocity in (6) with the centerlane as fixed lateral setpoint in (8), and lateral sway maneuvers use only the reference lateral position in (8). When $r^{(i)}$ is a scalar, and so is $w^{(i),h}$, (22b) is a union of intervals that can be explicitly evaluated. We first determine

$$\begin{aligned} &\exists j \in \mathbb{Z}_+,\ k \in \mathbb{Z}_{[0,N]},\ [a_k^{(i),0}]_j = 0, \\ &\quad \text{if } \begin{cases} [b_k^{(i),0}]_i > 0 \Rightarrow \{H_k [\Delta x^{(i),h\prime}\ w^{(i),h\prime}]' \leq K_k^{(i),h}\} = \emptyset \\ [b_k^{(i),0}]_i \leq 0 \Rightarrow \qquad \text{Redundant constraint,} \end{cases} \end{aligned} \tag{24}$$

For the cases $[a_k^{(i),0}]_j \neq 0$ we compute

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

where j is the index for the rows of the vectors $a_k^{(i),h}$, $b_k^{(i),h}$. Then, we can discretize the 1-dimensional range of $\mathcal{R}^{(i)}$, obtaining $\{r^{(i)}(\ell)\}_{\ell=1}^{n_\ell}$, and evaluate the feasibility of the maneuver by checking that at least one value of the maneuver parameter $r^{(i)}$ is: (i), included in the section of at least one goal achieving set, $\mathcal{A}_k$, at the current state $\Delta x^{(i),0}$, by retaining the values $r^{(i)}(\ell)$ that satisfy

$$\exists k \in \mathbb{Z}_{[0,N]}\colon\ \Psi^{(i),0} x_{0|t}^0 + r^{(i)}(\ell) \in [\underline{w}_k^{(i),0}, \overline{w}_k^{(i),0}] \tag{25a}$$

where no value is retained if the set is empty due to (24); (ii), excluded from the sections of all the colliding sets, C_k, at the current state $\Delta x^{(i),0}$ for all the obstacles by retaining the remaining values $r^{(i)}(\ell)$ that also satisfy

$$\begin{aligned} \Psi^{(i),0} x_{0|t}^0 + r^{(i)}(\ell) \in \Big(&(-\infty, \underline{w}_k^{(i),h}) \vee (\overline{w}_k^{(i),h}, +\infty) \Big), \\ &\forall k \in \mathbb{Z}_{[0,N]},\ \forall h \in \mathbb{Z}_{[1,n_0]}. \end{aligned} \tag{25b}$$

Thus, the maneuver parameter value r_t can be determined by evaluating elements of $\{r^{(i)}(\ell)\}_{\ell=1}^{n_\ell}$ against (25a), (25b) for determining feasibility, and against (22) for optimality. Once r_t is determined, we obtain $\{x_{k|t}\}_{k=0}^N$ that defines the reference

trajectory for the maneuver by simulating the maneuver motion model (e.g., based on (6), (8), (10)) forward in time from x_t with $r_{k|t} = r_t$, for $k \in \mathbb{Z}_{[0,N]}$.

4.2 Robustness metrics and uncertain obstacle motion

Besides the performance metrics in the ego vehicle maneuver, e.g., high/low velocity/accelerations, robustness is another important metric for choosing the reference trajectory. We can include a robustness metrics such that even for a nominal prediction of the obstacle motion, some deviations from the nominal behavior of the obstacle will not cause the maneuver to become infeasible.

Without including yet another uncertainty in the obstacle motion, for computational simplicity, we formulate a robustness metrics quantifying how large perturbations on the parameter vector results in a successful completion of the maneuver. To this end, we define the robustness radius of $r^{(i)} \in \mathcal{R}_f^{(i)}$ as

$$\delta_{r^{(i)}} = \max\{\delta \in \mathbb{R}_+ : \; r^{(i)} + \varsigma \in \mathcal{R}_f^{(i)}, \; \forall \varsigma, |\varsigma| \leq \delta\}. \tag{26}$$

For discretized $\mathcal{R}^{(i)}$, the conditions (26) 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. Also, a discretized $\mathcal{R}^{(i)}$ may cause a resolution error in the robustness radius where infeasible regions of size less than the discretization grid are not detected. The robustness radius can be included as part of the objective in the cost function $J^{(i)}$ in (22), which may be useful to address the difference in models used by motion planner and decision making, as well as imperfect predictions of the obstacle motions.

For the case where we explicitly model uncertainty in the motion of the obstacle, i.e., $\mathcal{D}^{(i),h}$ is not a singleton, the maneuver can be determined in two ways. If $\mathcal{D}^{(i),h}$ is discrete, the process in Section 4.1 may be repeated to determine the maneuver parameters values, where (18) is implemented as

$$\mathcal{F}_o^{(i),h}(\Delta x^{(i),h}, x^h) = \\ \bigcap_{d^{(i),h} \in \mathcal{D}^{(i),h}} \left\{ r^{(i)} \in \mathcal{R}^{(i)} : \; (\Delta x^{(i),h}, \Psi^{(i),h} x^h + r^{(i)}, d^{(i),h}) \notin \bigcup_{k=0}^{N} \mathcal{C}_k^{(i),h} \right\}, \tag{27}$$

i.e., for each obstacle, checking avoidance of the collision sets for all the discrete obstacle behaviors encoded in the exogenous disturbances.

Alternatively, consider the case when sets $\mathcal{D}^{(i),h}$ are uncountable. One can first select $r^{(i)} \in \{r^{(i)}(\ell)\}_{\ell=1}^{n_\ell}$ and substitute it into $\mathcal{C}^{(i),h}$ obtaining

$$C_{d,k}^{(i),h}(\Delta x^{(i),h}, x^h, r^{(i)}) = \left\{ d^{(i),h} \in \mathcal{D}^{(i),h} : \ (\Delta x^{(i),h}, \Psi^{(i),h} x^h + r^{(i)}, d^{(i),h}) \in C_k^{(i),h} \right\}, \quad (28)$$

Corollary 1 *Let $d^{(i),h} \in \mathcal{D}^{(i),h}$, $h \in \mathbb{Z}_{[0,n_o]}$, the parameter value $r^{(i)} \in \{r^{(i)}(\ell)\}_{\ell=1}^{n_\ell}$ makes maneuver $M^{(i)}$ feasible if and only if there exists $k \in \mathbb{Z}_{[1,N]}$ such that $(\Delta x^{(i),h}, \Psi^{(i),h} x^0 + r^{(i)}, 0) \in \mathcal{A}_k^{(i),0}$, and $C_{d,k}^{(i),h} = \emptyset$ for all $h \in \mathbb{Z}_{[1,n_o]}$, $k \in \mathbb{Z}_{[0,N]}$.*

Proof. The condition $(\Delta x^{(i),h}, \Psi^{(i),h} x^0 + r^{(i)}, 0) \in \mathcal{A}_k^{(i),0}$ for some $k \in \mathbb{Z}_{[1,N]}$ guarantees that the maneuver goal is achieved within at most N steps as for Theorem 1, while if such condition does not hold, the goal is not achieved and hence the maneuver does not succeed.

As for obstacle avoidance, if there exists $k \in \mathbb{Z}_{[1,N]}$, $h \in \mathbb{Z}_{[1,n_o]}$ such that $C_{d,k}^{(i),h} \neq \emptyset$, then there exists $d_k \in \mathcal{D}^{(i),0}$ such that $(\Delta x^{(i),h}, \Psi^{(i),h} x^h + r^{(i)}, d^{(i),h}) \in C_k^{(i),h}$, which means that a collision happens and the maneuver is not safe. Instead if $C_{d,k}^{(i),h} = \emptyset$ for all $k \in \mathbb{Z}_{[1,N]}$, $h \in \mathbb{Z}_{[1,n_o]}$, there exists no disturbance value causing a collision, and the maneuver is safe. □

According to Corollary 1 a maneuver parameter is feasible if the state and parameter values belongs to an achieving goal set $\mathcal{A}_k^{(i),0}$, and all the collision disturbance sets $C_{d,k}^{(i),h}$ are empty, i.e., no collision occurs for any value of the disturbance. When the motion models of ego and obstacles are linear and the obstacle and goal sets are polyhedral, as those in Section 2, sets $C_{d,k}^{(i),h}$ are polyhedral, and specifically, they are polyhedral sections of $C_k^{(i),h}$ at parameter value $r^{(i)}$. The set-emptyness checks are straightforward if $d^{(i),h}$ is a scalar, and require at most linear feasibility programs.

5 Simulations

We evaluate the decision making approach though several scenarios on a straight road with 2 lanes in the ego vehicle travel direction.

5.1 Simulations with Known Obstacle Behavior

First, we consider a lane change scenario, in which we allow 3 types of lane change maneuvers, where the lateral position is the output of 3^{rd} order systems as in (7) 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

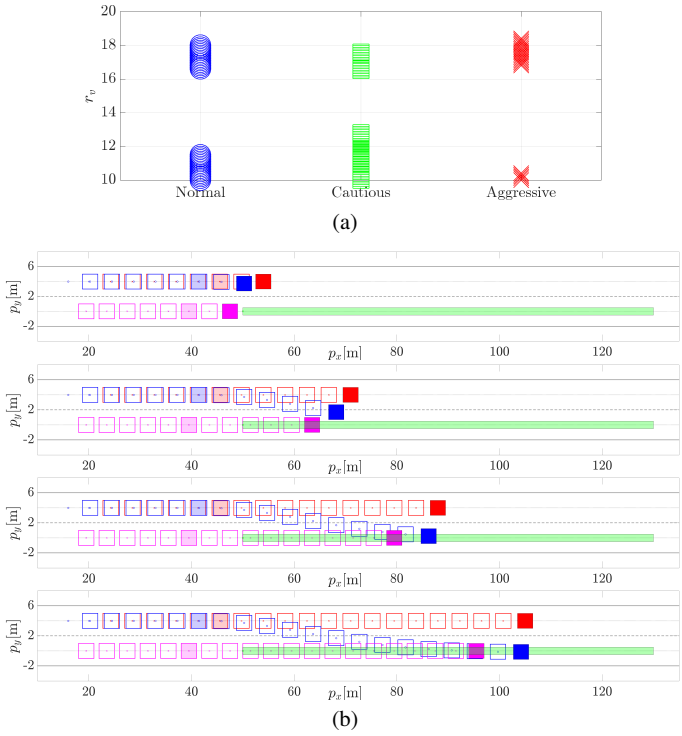


Fig. 4 Lane change scenario with 3 feasible maneuvers: (a) allowed values of commanded velocity r_v for each maneuver; (b) execution of the *normal* maneuver at maximum feasible r_v. Ego (blue) and other (red, purple) vehicles: current (solid), past (frame), and at maneuver start (shaded) positions. Region of lane change completion (green). Snapshots after $t = 0.5, 1.5, 2.5, 3.5$s from maneuver start.

time and increasing overshoot[1]. The goal is to complete a lane change from 10m to 120m ahead of the ego vehicle, within 5s with sampling period $T_s = 0.25$s, i.e., a horizon of $N = 20$ steps, commanding velocities $r_v \in [10, 20]$m/s, where the range is discretized in 100 points with a resolution 0.1m/s.

Fig. 4 shows the case with two other vehicles, one in the same lane and moving at the same speed as the ego vehicle, $v_o^{(1)} = v_x = 17$m/s, ahead by 4m, and the other in the lane to the right, moving at $v_o^{(2)} = 16$m/s and behind by 2m. The other vehicles are maintaining their velocities, i.e., $d_v^{(h)} = v_o^{(h)}$, $h \in \{1, 2\}$. From Fig. 4(a), the lane change is feasible with all 3 maneuvers for different commanded velocities. For each maneuver there is a higher range of admissible velocities that allows lane change ahead of the vehicle in the next lane, a lower range that allows lane change

[1] Despite the name used here for differentiation, all these maneuvers are mild enough that linear models in Section 2 are appropriate.

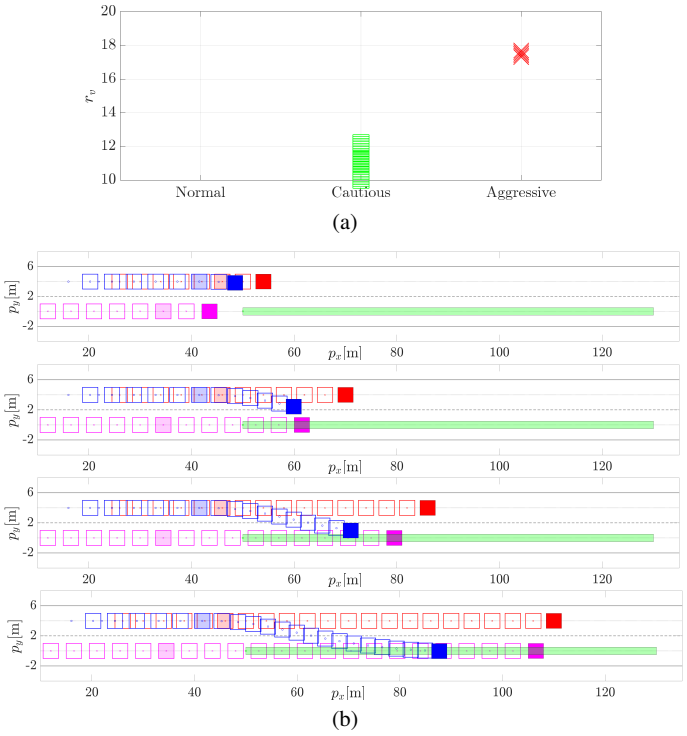


Fig. 5 Lane change scenario with 2 feasible maneuvers: (a) allowed values of commanded velocity r_v for each maneuver; (b) execution of the *cautious* maneuver with most robust r_v. Ego (blue) and other (red, purple) vehicles: current (solid), past (frame), and at maneuver start (shaded) positions. Region of lane change completion (green). Snapshots after $t = 0.5, 1.5, 2.5, 4.0$s from maneuver start.

behind, and a range of inadmissible velocities between the two, where a collision would occur. Fig. 4(b) shows the execution of the *normal* maneuver for its maximum allowed value $r_v = 18$, which results in changing lane ahead of the other vehicle in the next lane, while avoiding collisions and completing the lane change in the designated area.

Fig. 5 shows the case when the other vehicle initial position and velocities have been changed to 4.5m ahead, 7m behind, and $v_o^{(1)} = 16$m/s, $v_o^{(2)} = 18$m/s for the one in the same and next lane, respectively. The other vehicles are still maintaining their velocities, i.e., $d_v^{(h)} = v_o^{(h)}$, $h \in \{1, 2\}$. As shown in Fig. 5(a), the *normal* lane change is infeasible, while the *cautious* and *aggressive* lane changes are feasible by changing lane behind and ahead of the vehicle in the next lane, respectively. Fig. 5(b) shows the waypoints generated by choosing the value $r_v = 11.1$m/s, which has the largest robustness radius according to (26) for the *cautious* lane change.

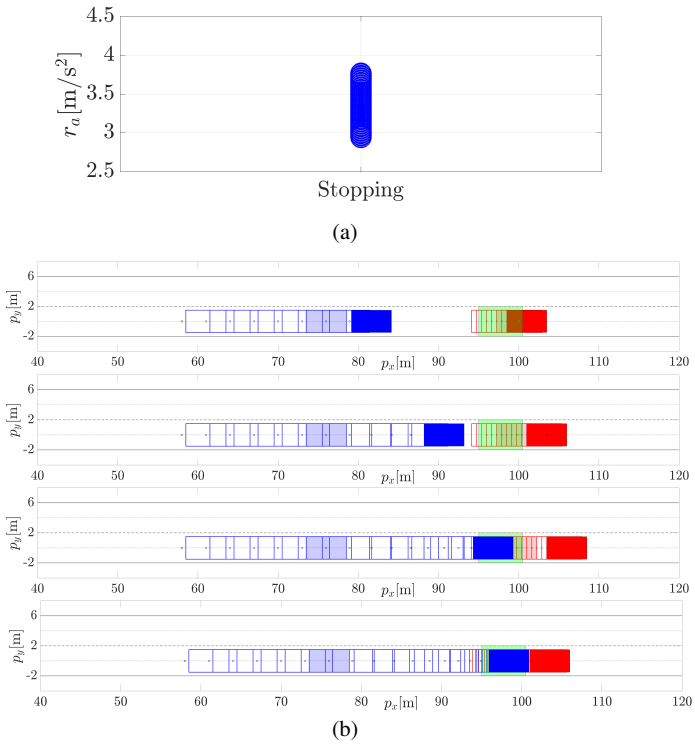


Fig. 6 Stopping scenario with 1 maneuver: (a) feasible values of commanded acceleration r_a; (b) execution of the maneuver with minimum r_a. Ego (blue) and other (red) vehicles: current (solid), past (frame), and at maneuver start (shaded) positions. Stop region (green). Snapshots after $t = 0.5, 1.5, 2.5, 3.5$s from maneuver start.

In Fig. 6 we show a scenario where the ego vehicle is driving at $v_x = 12$m/s, and has to reach a stop in a target area that starts 80m ahead, while another vehicle is slowly, $v_0 = d_v = 2.5$m/s, departing from it. This simulates stopping at an intersection, while a preceding vehicle starts crossing it. The decision making uses a single maneuver with the braking motion model (10), where the reference deceleration command $r_a \in [1, 5]$m/s^2 is the parameter, discretized with resolution of 0.01m/s^2. Fig. 6(a) shows that the maneuver is feasible, and Fig. 6(b) shows the waypoints obtained for the least deceleration $r_a = 2.94$m/s^2, for which the vehicle stops with a trajectory following just behind the departing vehicle.

Finally, Fig. 7 shows the total computing time, i.e., for checking all maneuvers, in the scenario shown in Figure 5, for different number of discretization values in each reference range, i.e., $n_\ell^{(i)}$, $i \in \mathbb{Z}_{[1,3]}$. The total time for the discretization at 0.1m/s, i.e., 100 points per maneuver, used in Fig. 4–5 is less than 7.5ms, i.e., 2.5ms per maneuver for checking all the points, in a non-optimized purely Matlab 2021b implementation on a 2020 MacBook Pro, with Intel i5 processor and 16GB of RAM.

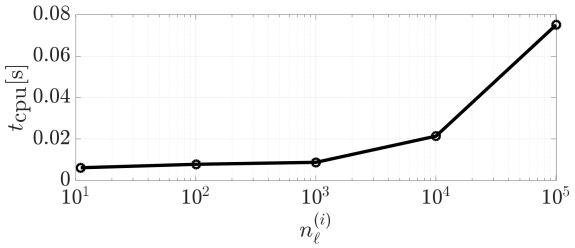


Fig. 7 Total computation time t_{cpu} to evaluate the 3 maneuvers in the scenario of Fig. 5 for different numbers $n_\ell^{(i)}$ of discretized parameter values.

This is more than 3 times faster than the approach in [2]. Furthermore, an equivalent standard C implementation, e.g., from non-optimized code generation, may be up to 10-40 faster. For a higher number of discretization points, which are unnecessary in this application other than to assess the computational burden, the curve is fairly flat until 1000 points, since the evaluation of feasibility by (25) is inexpensive after the upper and lower bounds have been determined, which needs to be done only once for a given initial state.

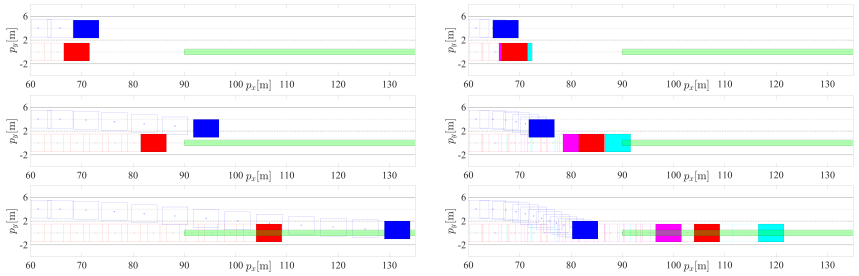
5.2 Simulations with Uncertain Obstacle Behavior

Next we show simulations comparing the case of known and uncertain behavior of the obstacles, in scenarios that are similar to the ones described in Section 5.1, though with some different initial conditions to better highlight the impact of the uncertainty in the other vehicle motion.

First we consider a lane change scenario where the goal is to change lane within 5s, with sampling period $T_s = 0.25$s, objective of maximum feasible velocity, $r_v \in [5, 25]$m/s and there is an obstacle vehicle slightly ahead in the target lane, and no vehicle in the lane where the ego vehicle is. The other vehicle has initial velocity $v_o = 15$m/s and in the nominal case it is maintaining such velocity, i.e., $d_v = v_0 = 15$m/s. In the uncertain case the obstacle may be maintaining the current velocity 15m/s, may be accelerating to 20m/s, or decelerating to 12m/s, hence $d_v \in \{12, 15, 20\}$m/s.

Fig. 8 shows that in the nominal case, the ego vehicle is able to change lane ahead of the other vehicle by selecting $v_r = 25$m/s. On the other hand, with the uncertain behavior of the other vehicle, the ego vehicle cannot pass in front and instead has to slow down and lane change behind the other vehicle by selecting $v_r = 5.5$m/s.

Then, we consider the stopping scenario where the goal is to stop in the stopping area while remaining in the same lane within 5s, with sampling period $T_s = 0.25$s, objective of smallest feasible deceleration, and there is an other vehicle departing from the stopping area. The other vehicle is initially stopped $v_o = 0$m/s and in the nominal case has a target velocity $d_v = 4$m/s. In the uncertain case the obstacle may

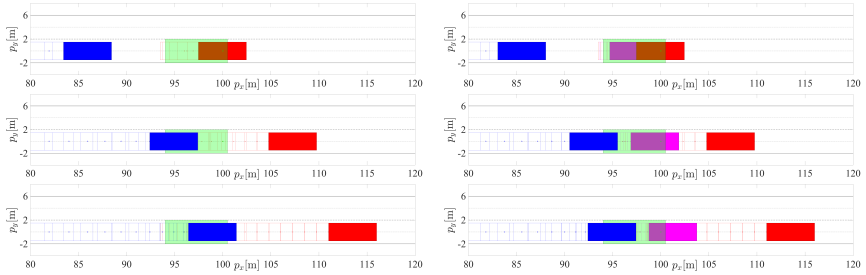


(a) (b)

Fig. 8 Lane change scenario with known and uncertain obstacle behavior. Maximum velocity objective, initial obstacle velocity $v = 15$m/s: (a) known obstacle velocity target $d_v = v_o = 15$m/s; (b) uncertain obstacle velocity target, $d_v \in \{12, 15, 20\}$m/s. Ego (blue) and other vehicle with constant speed (red), acceleration (cyan), deceleration (magenta): current (solid), past (frame), goal region (green). Snapshots after $t = 0.5, 1.5, 3.0$s from maneuver start.

be targeting the same velocity of the nominal casex 4m/s, or creeping slowly with a target of 1.5m/s, hence $d_v \in \{1.5, 4\}$m/s.

Fig. 9 shows that in the nominal case the ego vehicle stops in the forward part of the stop area and decelerates mildly, with $r_a = 2.01\text{m/s}^2$. On the other hand, when the behavior of the other vehicle is uncertain, the ego needs to stop further behind and hence applies a more aggressive deceleration with $r_a = 2.51$.



(a) (b)

Fig. 9 Stopping scenario with known and uncertain obstacle behavior. Minimum deceleration objective, initial obstacle velocity $v = 0$m/s: (a) known obstacle velocity target $d_v = v_o = 4.0$m/s; (b) uncertain obstacle velocity target, $d_v \in \{1.5, 4.0\}$m/s. Ego (blue) and other vehicle nominal speed (red), slow speed (magenta): current (solid), past (frame), goal region (green). Snapshots after $t = 0.5, 1.5, 3.0$s from maneuver start.

6 Conclusions

We developed a method for decision making for automated driving that determines feasibility of maneuvers using reachable sets parametrized by setpoints of vehicle motion quantities. The feasible maneuvers, the corresponding goals and the reference trajectories can be provided to the motion planner with certainty that it will be able to compute a trajectory to accomplish the maneuver while avoiding collisions. The proposed method is simple to implement, fast to compute, which is a requirement to limit the cost of the automotive computation platform, and we showed how it can include robustness metrics and account for uncertainty in the predicted motion of the other vehicles.

References

[1] Ahn H, Berntorp K, Di Cairano S (2020) Cooperating modular goal selection and motion planning for autonomous driving. In: 59th IEEE Conf. Dec. and Control, pp 3481–3486
[2] Ahn H, Berntorp K, Inani P, Ram AJ, Di Cairano S (2020) Reachability-based decision-making for autonomous driving: Theory and experiments. IEEE Trans Con Sys Tech 29(5):1907–1921
[3] Althoff M, Dolan JM (2014) Online verification of automated road vehicles using reachability analysis. IEEE Trans Robotics 30(4):903–918
[4] Berntorp K, Hoang T, Quirynen R, Di Cairano S (2018) Control architecture design for autonomous vehicles. In: IEEE Conf. Control Technology and Applications, pp 404–411
[5] Berntorp K, Bai R, Erliksson KF, Danielson C, Weiss A, Di Cairano S (2019) Positive invariant sets for safe integrated vehicle motion planning and control. IEEE Trans Intelligent Vehicles 5(1):112–126
[6] Berntorp K, Hoang T, Di Cairano S (2019) Motion planning of autonomous road vehicles by particle filtering. IEEE trans intelligent vehicles 4(2):197–210
[7] Berntorp K, Quirynen R, Uno T, Di Cairano S (2020) Trajectory tracking for autonomous vehicles on varying road surfaces by friction-adaptive nonlinear model predictive control. Vehicle System Dynamics 58(5):705–725
[8] Blanchini F, Miani S (2008) Set-theoretic methods in control, vol 78. Springer
[9] Borrelli F, Bemporad A, Morari M (2017) Predictive control for linear and hybrid systems. Cambridge University Press
[10] Buehler M, Iagnemma K, Singh S (2009) The DARPA urban challenge: autonomous vehicles in city traffic, vol 56. Springer
[11] Di Cairano S, Kolmanovsky IV (2018) Real-time optimization and model predictive control for aerospace and automotive applications. In: American Control Conf., pp 2392–2409
[12] Di Cairano S, Tseng HE, Bernardini D, 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

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

[14] Galceran E, Cunningham AG, Eustice RM, Olson E (2017) Multipolicy decision-making for autonomous driving via changepoint-based behavior prediction: Theory and experiment. Autonomous Robots 41(6):1367–1382

[15] Gao Y, Gray A, Tseng HE, Borrelli F (2014) A tube-based robust nonlinear predictive control approach to semiautonomous ground vehicles. Vehicle System Dynamics 52(6):802–823

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

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

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

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

[20] LaValle SM (2006) Planning algorithms. Cambridge university press

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

[22] Li S, Li K, Rajamani R, Wang J (2010) Model predictive multi-objective vehicular adaptive cruise control. IEEE Trans control systems technology 19(3):556–566

[23] SAE J3016: 2018 06 (2018) Taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles. Standard, Society of Automotive Engineers (SAE)

[24] Schurmann B, Klischat M, Kochdumper N, Althoff M (2021) Formal safety net control using backward reachability analysis. IEEE Trans Aut Control

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

[26] Verma R, Del Vecchio D (2011) Safety control of hidden mode hybrid systems. IEEE Trans Automatic Control 57(1):62–77

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