Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

Set-based Scheduling for Highway Entry of Autonomous Vehicles ⋆

Jan Eilbrecht * **Olaf Stursberg** *

* *Institute of Control and System Theory, Department of Electrical Engineering and Computer Science, University of Kassel, Germany, (e-mail: {jan.eilbrecht,stursberg}@uni-kassel.de).*

Abstract:
This paper proposes a framework for generation of collision-free reference trajectories in a cooperative multi-agent setting. The approach is hierarchical: a high-level controller schedules groups of cooperating agents, for which trajectories are then determined by a lower-level trajectory planner. Admissible behaviors of a cooperative group are encoded by so-called *maneuvers*, which are modeled by hybrid automata. This allows to plan trajectories by solving hybrid optimal control problems. Controllable sets characterize the solution sets of these problems and are used for quick assessment of feasibility prior to planning. This enables the high-level controller to quickly assess the feasibility of different maneuver options and cooperative groups. Special emphasis is put on safety and feasibility of the framework. The efficacy of the approach is demonstrated by simulation of a highway entry scenario for autonomous vehicles.

Keywords: Coordinating Control, Autonomous Systems, Autonomous Driving, Scheduling, Hybrid Systems, Integrated Traffic Management

1. INTRODUCTION

When controlling autonomous agents such as rovers, UAVs, or autonomous vehicles, a major challenge is to plan reference trajectories for the future states of an agent. This problem is, among others, complicated by collision avoidance constraints, which make the problem inherently non-convex. Considering a multi-agent context, another challenge is to coordinate the intents of several agents. The motivation for this paper is to propose a framework which enables computationally efficient control, provided that a certain problem structure is given. This is, for example, the case in on-road traffic of autonomous vehicles, which will be the primary considered application domain, even though an extension to different domains is conceivable.

Over the last decades, trajectory planning problems have been the subject of considerable research effort. Early approaches were often graph-based and neglected the agents' dynamics (LaValle, 2006), while more recent approaches such as sampling-based algorithms or numerical optimal control explicitly account for it (González et al., 2016). In a multi-agent context, planning typically focuses on different aspects: while consensus problems aim to establish inter-agent agreement on and convergence towards the value of certain variables, others primarily aim at ensuring collision avoidance. Often, a decentralized/distributed approach is chosen, in which agents make local plans, which are then iteratively adapted to those of others, e.g. (Kuwata and How, 2010). Due to the non-convex nature of the problem, it is challenging to prove convergence and feasibility for all involved agents. Often, approaches consider specific scenarios such as control at intersections, e.g. (de La Fortelle, 2010; Zhang and Cassandras, 2019), while the framework to be proposed is applicable to general traffic situations. Our own earlier work (outlined in Sec. 2.2) approached the problem relying on a maneuver-based approach, in which maneuvers are modeled as hybrid optimal control problems (Eilbrecht and Stursberg, 2018), for which it is possible to compute controllable sets, allowing for feasibility assessment prior to planning. The process can be sped up by relying on approximate solutions, cf. (Eilbrecht and Stursberg, 2019).

Contrasting prior work, this note not focuses on computations pertaining to *single* maneuvers, but on the use of a *set* of maneuvers ("maneuver library") in a computationally efficient framework for control of multiple agents. It is described how to determine cooperative groups that are to execute a certain maneuver, focusing on feasibility

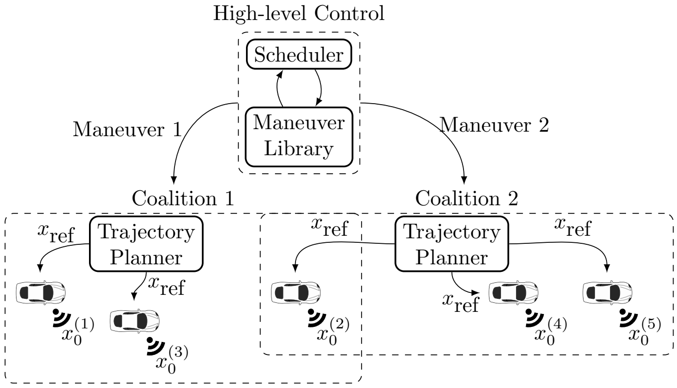

Fig. 1. Envisioned Framework.

⋆ Financial support by the German Research Foundation (DFG) within priority program (SPP) 1835 is gratefully acknowledged.

of plans. Specifically, it is investigated how to: 1) define rules for interaction between cooperative groups and non-cooperating vehicles, 2) guarantee safety of all vehicles and recursive feasibility of the controller. Note that the proposed framework is rather general, such that it can serve as platform for different cooperation mechanisms.

2. GENERAL SETTING

This section provides an overview of the considered problem as well as the proposed solution approach, setting the stage for detailed descriptions of derived subproblems and the corresponding solution approaches in later sections.

2.1 Problem setting

Consider settings where a group of autonomous vehicles is driving on a road. Let $\mathcal{V} \subset \mathbb{N}$ contain identifiers of these vehicles and assume that each vehicle $i \in \mathcal{V}$ is given knowledge of the sequence of roads that lead it from its starting point to its destination. Then, consider control of the vehicles within these road segments. Let the dynamics of a vehicle i be given by:

$$\dot{\chi}(t) = f^{(i)}\left(\chi(t), \mu(t)\right),$$

depending on its state χ at the current time t and input values μ. Assume that the inputs are determined by a given controller function $k^{(i)}$, based on the current state and a time-varying reference value $x_{\text{ref}}^{(i)}(t)$:

$$\mu(t) = k^{(i)}\left(\chi(t), x_{\text{ref}}^{(i)}(t)\right).$$

The task considered in this paper is to obtain a reference trajectory $x_{\text{ref}}^{(i)}(t)$ which must ensure that the closed-loop dynamics of the vehicle does not violate constraints on the states and inputs of the vehicle. While some of these, such as constraints on velocity or acceleration, stem from the dynamics of the vehicle, others result from the road topology or the need to avoid collisions with other vehicles. The latter implies that $x_{\text{ref}}^{(i)}$ must depend on $x_{\text{ref}}^{(j)}$, $j \in \mathcal{V}$, $j \neq i$. This problem is considered in a setting where vehicles are generally willing and able to locally communicate information and adapt their behavior to that of others.

2.2 Solution Approach

Fig. 1 shows the elements of the solution approach, which are outlined in the following (cf. Eilbrecht and Stursberg (2018) for an earlier version). At the core of the approach is the notion of a *maneuver*, cf. Sec. 3, which serves to formally encode behaviors such as merging, overtaking, or lane keeping, i.e., sets of qualitatively similar trajectories. A maneuver can specify behaviors of one or more vehicles, where it is assumed that a finite number of maneuvers is sufficient to operate vehicles on the road. A collection of maneuvers will be referred to as *maneuver library*. The proposed approach is hierarchical: based on the maneuver library, a high-level control algorithm determines a group (termed *coalition*) of vehicles which are to perform a certain maneuver. The vehicles are assigned roles, some of which require a vehicle's cooperation (e.g. if a vehicle must be ready to open a gap), while other roles only serve to model the interaction with the surrounding outside of the maneuver group, cf. Sec. 5.2. While a vehicle can only assume one cooperative role, it can be included as non-cooperating in several maneuvers simultaneously, such that coalitions may overlap. For each coalition, a trajectory planner determines reference trajectories for all cooperative vehicles.

The outlined procedure requires to: 1) determine which maneuvers are feasible for which vehicles, 2) make a choice among the feasible ones, and 3) determine a reference trajectory. In the following, the focus is on task 1), cf. Sec. 3.2, and task 3), see Sec. 3.1.

3. MANEUVER CONCEPT

Prior to introducing the maneuver concept, define in slight modification of (Stursberg and Krogh, 2003):

Definition 1. (Hybrid Automaton). A hybrid automaton $\mathtt{HA}$ is a tuple $(\mathcal{Q}, q_0, \mathbb{X}, \text{inv}, \mathbb{U}, \mathbb{W}, \mathcal{X}_0, \mathcal{X}_\text{T}, \Theta, g, f)$, defining:

- a finite set of *phases* $\mathcal{Q}$ with initial phase $q_0 \in \mathcal{Q}$,
- a *state space* $\mathbb{X} \subseteq \mathbb{R}^{n_x}$ and state vector $x \in \mathbb{X}$,
- a function $\text{inv} : \mathcal{Q} \to 2^{\mathbb{X}}$ called *invariant*, assigning to each phase $q \in \mathcal{Q}$ a set $\text{inv}(q) \subseteq \mathbb{X}$ in which x may evolve over time without changing the phase q,
- input vectors $u \in \mathbb{U}(q)$ taken from bounded, phase-dependent sets $\mathbb{U} : \mathcal{Q} \to 2^{\mathbb{R}^{n_u}}$,
- disturbance vectors $w \in \mathbb{W}(q)$ taken from bounded, phase-dependent sets $\mathbb{W} : \mathcal{Q} \to 2^{\mathbb{R}^{n_w}}$,
- the sets $\mathcal{X}_0 \subseteq \text{inv}(q_0)$ and $\mathcal{X}_\text{T} \subseteq \mathbb{X}$ of initial and target states, respectively,
- a set of admissible *discrete transitions* $\Theta \subseteq \mathcal{Q} \times \mathcal{Q}$,
- a *guard* function $g : \Theta \to 2^{\mathbb{X}}$ which assigns a guard set $g(\theta) \subseteq \mathbb{X}$ to each transition $\theta = (q_i, q_j) \in \Theta$ with $\text{inv}(q_i) \cap \text{inv}(q_j) \neq \emptyset$,
- and a *flow* function $f : \mathcal{Q} \to (\mathbb{X} \times \mathbb{U} \times \mathbb{W} \to \mathbb{R}^{n_x})$.

The semantics of the automaton is as follows: Starting in $\mathcal{X}_0$, the continuous state x evolves according to the flow function $f(q_0)$, depending on the state, input, and disturbance. The input is chosen without knowledge of the disturbance. Upon entry of x into a guard set $g(\theta)$, an immediate transition from q_0 to phase q_j is enforced, where the evolution of x continues with $x \in \text{inv}(q_j)$ until another guard set is entered, and so forth. Once x enters the terminal set $\mathcal{X}_\text{T}$, its evolution stops.

An exemplary hybrid automaton is shown in Fig. 2. Throughout this paper, the flow function is assumed to define affine dynamics in each phase q:

$$x(t_{k+1}) = A_q x(t_k) + B_q u(t_k) + B_{1,q} w(t_k) + a_q \qquad (1)$$

on a discrete time domain $\mathbb{T} = \{t_k = k \cdot T_\text{s} | k \in \mathbb{N}\}$ with sampling time T_s. In general, vehicle dynamics are nonlinear, such that reference trajectories based on (1) could be inadmissible for a real vehicle. Encouraged by first results investigating into this (Schürmann et al., 2017), we assume that this is not a problem and postpone a more detailed treatment to future work. Also assume that the target set $\mathcal{X}_\text{T}$ is reachable from the initial set $\mathcal{X}_0$ and that all sets in the automaton are polyhedral. Based on this definition of $\mathtt{HA}$, define:

Definition 2. (Maneuver). A maneuver is a tuple $\mathtt{M} = (\mathcal{C}, H_\text{plan}, \mathtt{HA})$, consisting of a set $\mathcal{C} \subseteq \mathcal{V}$ of involved vehicles, a time horizon H_plan, and a hybrid automaton $\mathtt{HA}$.

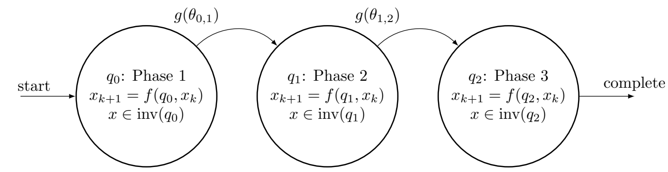

Fig. 2. Example of a hybrid automaton.

In this, the vehicles carry information about their current state and their preferences (e.g. regarding driving style or energy expenditure) in form of a cost function, while H_{plan} defines the duration of a maneuver over which a reference is needed. The hybrid automaton encodes admissible vehicle behaviors, which is exemplified in Sec. 6. The general idea is to combine the dynamics of all involved vehicles in the flow function of $\mathtt{HA}$, such that the invariants constrain their behavior, while different phases account for non-convex constraints, e.g. for collision avoidance, and impose a certain temporal structure of a maneuver.

3.1 Hybrid Optimal Control for Trajectory Planning

Denote by $u_j(\cdot) := (u(t_0)\ u(t_1)\ \dots\ u(t_{j-2})\ u(t_{j-1}))$ a sequence of j input vectors (use a corresponding notation for sequences of other quantities). Using the so-called big-M method, cf. Williams (2013), and (vectors of) binary variables δ, sequences which are admissible with respect to Def. 1 can be collected in a set:

$$\begin{aligned}
\mathcal{A}_j(\mathtt{HA}) := \Big\{ & u_j(\cdot) | \forall\ w(t_k) \in \mathbb{W}(q),\ \forall\ 0 \leq k \leq j,\ j \in \mathbb{N}, \\
& \exists x_{j+1}(\cdot), \delta_{j+1}(\cdot) : x(t_0) \in \mathcal{X}_0,\ u(t_k) \in \mathbb{U}(q), \\
& x(t_{k+1}) = A_q x(t_k) + B_q u(t_k) + B_{1,q} w(t_k) + a_q, \\
& E_1 x(t_k) + E_2 \delta(t_k) + E_3 q(t_k) + E_4 q(t_{k+1}) \leq E_0, \\
& x(t_k) \in \mathrm{inv}(q(t_k)),\ x(t_{j+1}) \in \mathcal{X}_{\mathrm{T}} \Big\},
\end{aligned} \tag{2}$$

where the semantics of the automaton have been translated into (2). A cost function assigns cost values to admissible input sequences:

$$J(x_0, u_{H_{\text{plan}}}(\cdot)) = \sum_{k=1}^{H_{\text{plan}}} \|z(t_k) - z_{\text{ref}}\|_Q + \|u(t_{k-1})\|_R .$$

It penalizes both the magnitude of the control inputs u and the deviation of an output signal:

$$z(t_k) = Cx(t_k) \tag{3}$$

from a reference value z_{ref}, weighted by matrices $Q \geq 0$ and $R \geq 0$ of appropriate dimensions, respectively. The matrices Q and R could, for example, model preferences of controlled vehicles. For the hybrid automaton $\mathtt{HA}$ of a given maneuver, solve:

Problem 3. (Trajectory Planning). Given an initial state x_0 and a planning horizon H_{plan}, determine an optimal sequence of input signals $u^*_{H_{\text{plan}}}(\cdot)$ as follows:

$$u^*_{H_{\text{plan}}}(\cdot) := \arg\min \sum_{k=1}^{H_{\text{plan}}} \|z(t_k) - z_{\text{ref}}\|_Q + \|u(t_{k-1})\|_R$$

subject to (3), $u^*_{H_{\text{plan}}}(\cdot) \in \mathcal{A}_{H_{\text{plan}}}(\mathtt{HA})$, and $x(t_0) = x_0$.

Problem 3 is a mixed-integer quadratic program, which can be solved using, e.g., CPLEX or GUROBI.

Assumption 4. (Initial State). Assume that the initial positions and velocities of all cooperative and non-cooperative vehicles are known at the current time t_0.

During execution of a maneuver, Problem 3 is solved anew at the current time instance to include new information regarding non-cooperating vehicles, e.g. their position and velocity, thus implementing closed-loop control (Bertsekas, 1995). Unlike in model-predictive control, Problem 3 is solved with a shrinking horizon to ensure finite termination of a maneuver (as opposed to asymptotic convergence). The general procedure to be carried out online is:

Require: Vehicle states $x^{(i)}$, $i \in \mathcal{V}$, at current time t_0, set $\mathcal{O}$ of ongoing maneuvers
$\bar{\mathcal{V}} \leftarrow$ Find vehicles in $\mathcal{V}$ not assigned a maneuver
$\{\mathtt{M}_1, \mathtt{M}_2, \dots\} \leftarrow$ Schedule maneuvers for $\bar{\mathcal{V}}$
$\mathcal{O} \leftarrow \mathcal{O} \cup \{\mathtt{M}_1, \mathtt{M}_2, \dots\}$
for each $\mathtt{M}_l \in \mathcal{O}$ **do**
    $(\mathcal{C}, H_{\text{plan}}, \mathtt{HA}) \leftarrow \mathtt{M}_l$
    if $H_{\text{plan}} = 0$ **then**
        $\mathtt{M}_l$ complete; disband $\mathcal{C}$
    else
        $x^{(i)}_{\text{ref}}(\cdot) \leftarrow \mathrm{Plan}(\mathtt{M}_l)\ \ \forall i \in \mathcal{C}$
        $H_{\text{plan}} \leftarrow H_{\text{plan}} - 1$

3.2 Feasibility Assessment using Controllable Sets

Robustly controllable sets are used in order to quickly assess feasibility of a maneuver for a given initial state:

Definition 5. (j-Step Robustly Controllable Set). Given a maneuver $\mathtt{M} = (\mathcal{C}, j, \mathtt{HA})$, define the j-step robustly controllable set $\mathcal{K}_j(\mathtt{M})$ of the maneuver as:

$$\mathcal{K}_j(\mathtt{M}) = \left\{ x | \exists u^*_j(\cdot) \text{ as solution to Problem 3} \right\}. \tag{4}$$

Note that the parameterization of the quadratic cost function of Problem 3 does not have influence on the existence of a solution. Also note that often a recursive definition of controllable sets is employed, cf. (Kerrigan, 2001) for linear systems, which embraces the same concept, but would be quite cumbersome to adapt for hybrid optimal control problems.

Given an initial state x_0 of a group of vehicles $\mathcal{C}$, explicit representations of controllable sets offer the possibility to quickly assess feasibility of a maneuver $\mathtt{M} = (\mathcal{C}, H_{\text{plan}}, \mathtt{HA})$ over a certain horizon H_{plan} prior to planning – simply by checking whether $x_0 \in \mathcal{K}_{H_{\text{plan}}}(\mathtt{M})$ holds. This allows to quickly evaluate feasibility of different options (i.e., different maneuvers with different hybrid automata and/or different maneuver durations). Here, the robustly controllable sets are described by linear inequalities, such that checking of set inclusion reduces to a matrix-vector product, making it computationally efficient.

4. CASE STUDY: HIGHWAY ENTRY

In order to fix ideas, the remainder of the paper demonstrates the functionality of the framework by means of an example, namely the scenario shown in Fig. 3, in which autonomous vehicles are driving on a highway (left lane), while other autonomous vehicles are trying to merge from a merging lane (right lane). The following rules are imposed: a minimal velocity $v_{x,\min,\text{left}}$ is enforced on the highway (as is common e.g. on German highways). The merge

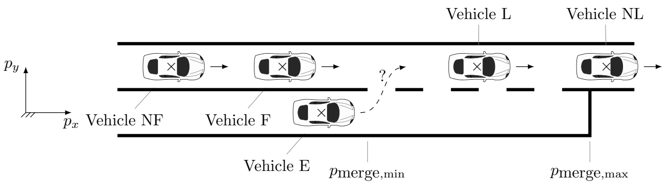

Fig. 3. Highway Entry: Where and when to merge?

lane enables vehicles to accelerate such that, at the end of the merging process, this constraint is fulfilled. Merging must take place between $p_{\text{merge,min}}$ and $p_{\text{merge,max}}$. If it should be impossible because no gap opens, a vehicle must stop on the merge lane in a position that allows to sufficiently accelerate prior to merging later on.

The scenario is to be controlled by the proposed framework. Regarding the infrastructure setup, similar assumptions are made as by (Zhang and Cassandras, 2019) for intersection control: At first, to limit the scope of the paper, assume that only autonomous vehicles are driving on the road. Then, assume that the controller is part of the infrastructure, i.e., a road side unit, and controls a certain section (the *control zone*) of the road up- and downstream (say 1.5 km in each direction). Events outside this zone will not be considered for simplicity. This setting allows to focus on the basic functionality of the approach without considering questions of decentralized computations, inter-vehicle communication, changing road topology, etc., which are deferred to future work.

Focusing on feasibility, the following simple maneuver scheduling rule is employed: all vehicles on the merge lane except for the first one are to keep the lane. For the first vehicle, it is assessed whether a feasible merging maneuver exists. If this is the case, the one with shortest duration H_{plan} is executed (clearly, more elaborate strategies are conceivable, but are beyond the scope of this paper). Otherwise, the vehicle must keep the lane until merging becomes possible. Note that both the gap into which and the time at which merging occurs result from the algorithm and are not fixed a priori.

In order to comply with (1), affine or linear vehicle models are used for a vehicle i:

$$\dot{x}^{(i)}(t) = A_{\text{c}}x^{(i)}(t) + B_{\text{c}}u^{(i)}(t) + a_{\text{c}}, \tag{5}$$

with the state and input vectors of a vehicle i:

$$x^{(i)} = \begin{bmatrix} p_x^{(i)} & p_y^{(i)} & v_x^{(i)} & v_y^{(i)} \end{bmatrix}^\mathsf{T}, \; u^{(i)} = \begin{bmatrix} u_x^{(i)} & u_y^{(i)} \end{bmatrix}^\mathsf{T}.$$

Assume for simplicity:

Assumption 6. (Homogeneous Vehicle Dynamics). All vehicles have the same dynamics.

While this assumption is clearly unrealistic, it is conceptually simple to extend our framework to heterogeneous dynamics, e.g. by assuming the existence of classes of similar vehicle dynamics and formulating maneuvers for combinations of different classes. Nonetheless, this extension is cumbersome and beyond the scope of this paper.

Both states and inputs are constrained by polyhedral sets:

$$x^{(i)} \in \mathcal{X}, \; u^{(i)} \in \mathcal{U}. \tag{6}$$

In the following, let $\mathcal{U} = [u_{x,\min}, u_{x,\max}] \times [u_{y,\min}, u_{y,\max}]$ and $\mathcal{X} = [p_{x,\min}, p_{x,\max}] \times [p_{y,\min}, p_{y,\max}] \times [v_{x,\min}, v_{x,\max}] \times$ $[v_{y,\min}, v_{y,\max}]$. Values for all parameters used in this case study are given in Tab. 1. Zero-order hold discretization using sampling time T_{s} and appropriate, maneuver-dependent combination with the dynamics of other vehicles then leads to (1). The matrices A_{c}, B_{c}, and a_{c} can, e.g., be obtained by location-dependent linearization of a nonlinear vehicle model. For now, simply let:

Table 1. Parameter values used in the case study.

$u_{x,\min} = -3\,\mathrm{m\,s^{-1}}$	$p_{\text{merge,min}} = 200\,\mathrm{m}$	$v_{x,\min} = 0\,\mathrm{m\,s^{-1}}$
$u_{x,\max} = 3\,\mathrm{m\,s^{-1}}$	$p_{\text{merge,max}} = 400\,\mathrm{m}$	$v_{x,\max} = 33.3\,\mathrm{m\,s^{-1}}$
$u_{y,\min} = -3\,\mathrm{m\,s^{-1}}$	$v_{x,\text{min,left}} = 22.2\,\mathrm{m\,s^{-1}}$	$v_{y,\min} = 0\,\mathrm{m\,s^{-1}}$
$u_{y,\max} = 3\,\mathrm{m\,s^{-1}}$	$p_{x,\min} = -\infty$ m	$v_{y,\max} = 5.56\,\mathrm{m\,s^{-1}}$
$l_{x,\text{safe}} = 5\,\mathrm{m}$	$p_{x,\max} = \infty$ m	$T_{\text{s}} = 0.5\,\mathrm{s}$

$$A_{\text{c}} = \begin{bmatrix} 0 \; 0 \; 1 \; 0 \\ 0 \; 0 \; 0 \; 1 \\ 0_{2\times 4} \end{bmatrix}, \; B_{\text{c}} = \begin{bmatrix} 0_{2\times 2} \\ I_{2\times 2} \end{bmatrix}, \; a_{\text{c}} = 0. \tag{7}$$

This model assumes double integrator dynamics for both longitudinal and lateral dynamics. Despite its simplicity, it has been successfully employed in planning problems, e.g. by Schouwenaars et al. (2001); Qian et al. (2016); Eilbrecht and Stursberg (2018). This has several reasons: on the one hand, proper choice of the state and input constraints sets $\mathcal{X}$ and $\mathcal{U}$ can capture many behaviours of more complex models. For example, it is possible to introduce coupling between longitudinal and lateral dynamics (Qian et al., 2016) or to approximate the so-called friction circle (Manzinger et al., 2017). Furthermore, plans are to be obtained for comfortable on-road driving, which occurs in state space regions where more complex vehicle models are only mildly nonlinear, qualitatively speaking. Finally, obtained plans are not directly applied as control inputs to the vehicle, but serve as references for lower-layer controllers. By incorporating safety distances into the maneuver formulations, minor deviations are unproblematic.

5. FORMAL GUARANTEES

This section aims to endow the proposed framework with formal guarantees concerning 1) the capability of the scheduler to eventually enable merging of a vehicle on the ramp, 2) safety, and 3) completion of a maneuver despite acting disturbances (which were not considered in prior work (Eilbrecht and Stursberg, 2018), making this a substantial contribution of the paper at hand).

In the following, extending standard definitions as, e.g., (Kerrigan, 2001), define:

Definition 7. (Maneuver Robustly Control Invariant Set). Given a maneuver $\mathtt{M} = (\mathcal{C}, H_{\text{plan}}, \mathtt{HA})$, a robustly control invariant set of the maneuver $\mathtt{M}$ is:

$$\mathcal{L}(\mathtt{M}) := \{x | \forall w_\infty(\cdot) \in \mathbb{W}(q_\infty(\cdot)) \; \exists u_\infty(\cdot) \in \mathcal{A}_\infty(\mathtt{HA}) : \\ f(x, u_\infty(t_k), w_\infty(t_k)) \in \mathcal{L}(\mathtt{M}) \; \forall k \in \mathbb{N}\}.$$

5.1 Recursive Scheduling Feasibility

At first, the following property is considered:

Definition 8. (Recursive Scheduling Feasibility). Recursive scheduling feasibility holds if the scheduling algorithm is always eventually able to schedule a merging maneuver for every vehicle on the merge lane.

The ability to merge not only depends on the state of the vehicle on the ramp, but also on the vehicles on the highway. Even if these are willing to cooperate, worst-case scenarios (extreme traffic density, "bumper to bumper") exist in which a gap simply cannot be opened. Assume that this will not always be the case, or more optimistically:

Assumption 9. (Traffic Density). Given a vehicle i on the ramp at time t_0, there exists t_k, $k \geq 0$, such that no other vehicle is in the control zone.

Clearly, it is possible to make this assumption less restrictive by identifying distances between vehicles which are large enough to assume that these do not interact. This would allow other vehicles to exist within the control zone, but would also unnecessarily complicate the exposition. Assumption 9 allows to reduce the analysis to the case of a single merging vehicle. This can be modeled by a maneuver $\mathtt{M}_1 = (\{\mathrm{E}\}, \infty, \mathtt{HA}_1)$ with $\mathcal{Q} = \{q_0, q_1, q_2\}$, $\Theta = \{\theta_1 = (q_0, q_1),\ \theta_2 = (q_1, q_2)\}$, flow as dynamics of a vehicle (E) given by (5) and (7), invariants, and guards:

$$\mathrm{inv}(q_0) = \mathcal{X},\ \mathrm{inv}(q_1) = \left\{x | p_x^{(\mathrm{E})} \geq p_{\mathrm{merge,min}}\right\},$$
$$\mathrm{inv}(q_2) = \left\{x \in \mathrm{inv}(q_1) | p_y^{(\mathrm{E})} = p_{y,\max},\ v_y^{(\mathrm{E})} = 0\right\},$$
$$g(\theta_1) = \{x | p_y^{(\mathrm{E})} > 0\},\ g(\theta_2) = \{x | p_x^{(\mathrm{E})} \geq p_{\mathrm{merge,max}}\}.$$

Denote by $\mathcal{X}_\mathrm{T} = \left\{x | p_y^{(\mathrm{E})} = p_{y,\max},\ v_y^{(\mathrm{E})} = 0\right\}$ and by $\mathcal{X}_0 = \left\{x | p_y^{(\mathrm{E})} = 0,\ v_y^{(\mathrm{E})} = 0, p_x^{(\mathrm{E})} \leq p_{\mathrm{merge,min}}\right\}$ the target and initial set, respectively. Then, the following holds:

Proposition 10. (Capability to Merge). Given a vehicle (E) on the merge lane, a robustly control invariant set $\mathcal{X}_{\mathrm{merge}} := \mathcal{L}(\mathtt{M}_1)$, a maneuver library $\mathcal{M} = \{\mathtt{M}_3, \mathtt{M}_4, \dots\}$, and a set of vehicles $\mathcal{V}$ on the highway, then vehicle (E) will eventually be able to merge if either $x^{(\mathrm{E})}(t_0) \in \mathcal{X}_{\mathrm{merge}}$ or $\exists \mathtt{M} = (\mathcal{C}, j, \mathtt{HA}) \in \mathcal{M} :\ x(t_0) \in \mathcal{K}_j(\mathtt{M})$, where $x(t_0)$ denotes the current state of $\{\mathrm{E}, \mathcal{C}\}$ and $\mathcal{C} \subseteq \mathcal{V}$ is a suitably chosen subset of the vehicles on the highway.

Proof. Clearly, $x(t_0) \in \mathcal{K}_j(\mathtt{M})$ enables merging by definition of the controllable set. On the other hand, $x^{(\mathrm{E})}(t_0) \in \mathcal{X}_{\mathrm{merge}}$ allows that $x^{(\mathrm{E})}(t_k) \in \mathcal{X}_{\mathrm{merge}}\ \forall k > 0$ by definition. Based on Assumption 9, this eventually allows merging. □

In an implementation, the computation of a set $\mathcal{X}_{\mathrm{merge}}$ can be carried out using, e.g., the MPT toolbox (Kvasnica et al., 2004), with adaptions to the hybrid systems setting.

5.2 Robust Maneuver Feasibility and Safety

Prior to analyzing their impact, assume the following:

Assumption 11. Disturbances as in Def. 1 solely model actions of non-cooperative vehicles, while other disturbances (modeling errors, measurement noise, etc.) are counteracted by subordinate tracking controllers.

Assumption 12. Actions of non-cooperative vehicles only affect the longitudinal dynamics of other vehicles.

The latter can be ensured by proper formulation of maneuvers.

Safety The aim is to guarantee safety of the involved vehicles in the following sense:

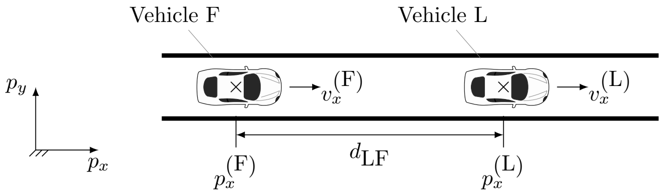

Fig. 4. Longitudinal safety distance between two vehicles.

Definition 13. (Safety). Safety holds if every vehicle is able to keep a longitudinal safety distance of at least $l_{x,\mathrm{safe}}$ to preceding vehicles at any time.

The basis for the following analysis is the constellation shown in Fig. 4, where a vehicle (L) is driving ahead of a vehicle (F). Interaction of these vehicles is modeled by a maneuver $\mathtt{M}_2 := (\{\mathrm{L}, \mathrm{F}\}, \infty, \mathtt{HA}_2)$, in which the acceleration of the leading vehicle is perceived as disturbance, while the one of the following vehicle is controlled. It is assumed that the disturbance does not violate velocity constraints, since these are imposed by traffic rules. $\mathtt{HA}_2$ consists of three phases $\mathcal{Q} = \{q_0, q_1, q_2\}$ with identical invariants and input sets, but different disturbance sets:

$$\mathrm{inv} = \{x | x \in \mathcal{X} \times \mathcal{X}, p_y^{(\mathrm{L})} = p_y^{(\mathrm{F})} = 0,\ v_y^{(\mathrm{L})} = v_y^{(\mathrm{F})} = 0,$$
$$p_x^{(\mathrm{L})} - p_x^{(\mathrm{F})} \geq l_{x,\mathrm{safe}}\}, \mathbb{W}(q_0) = \left\{u \in \mathcal{U} : u_y^{(\mathrm{L})} = 0\right\},$$
$$\mathbb{W}(q_1) = \left\{u \in \mathbb{W}(q_0) : u_x^{(\mathrm{L})} \geq 0\right\}, \mathbb{U} = \left\{u \in \mathcal{U} : u_y^{(\mathrm{F})} = 0\right\}$$
$$\mathbb{W}(q_2) = \left\{u \in \mathbb{W}(q_0) : u_x^{(\mathrm{L})} \leq 0\right\}.$$

Based on (5), the flows in all locations are discrete-time versions of:

$$\dot{x} = \begin{bmatrix} A_\mathrm{c} & 0 \\ 0 & A_\mathrm{c} \end{bmatrix} x(t) + \begin{bmatrix} B_\mathrm{c} \\ 0 \end{bmatrix} u_x^{(\mathrm{F})}(t) + \begin{bmatrix} 0 \\ B_\mathrm{c} \end{bmatrix} u_x^{(\mathrm{L})}(t), \quad (8)$$

Transitions are $\Theta = \{(q_0, q_1), (q_1, q_0), (q_0, q_2), (q_2, q_0)\}$. Starting in q_0, a transition to q_1 is enforced if $v_x^{(\mathrm{L})} = v_{x,\min}$, from where a transition back to q_0 occurs if $v_x^{(\mathrm{L})} > v_{x,\min}$. A transition from q_0 to q_2 occurs if $v_x^{(\mathrm{L})} = v_{x,\max}$, and in the reverse direction if $v_x^{(\mathrm{L})} < v_{x,\max}$.

Proposition 14. (Safety). Given a robustly control invariant set $\mathcal{L}(\mathtt{M}_2)$, a polytope

$$\mathcal{X}_{\mathrm{safe}} = \{x | A_{\mathrm{safe}} x \leq b_{\mathrm{safe}}\} \subset \mathcal{L}(\mathtt{M}_2), \quad (9)$$

and a current state $x \in \mathcal{X}_{\mathrm{safe}}$, longitudinal collision between L and F can be avoided at all future times.

Proof. Clearly, $x(t_k) \in \mathcal{X}_{\mathrm{safe}} \Rightarrow x(t_k) \in \mathcal{L}(\mathtt{M}_2)$. By definition of $\mathcal{L}(\mathtt{M}_2)$, there exists $u^{(\mathrm{F})}(t_k)$ such that $x(t_{k+1}) \in \mathcal{L}(\mathtt{M}_2)\ \forall\ u_x^{(\mathrm{L})}$. □

Property 14 obviously carries over to groups of vehicles when guaranteeing the set memberships for according pairs of vehicles. The sets $\mathcal{L}(\mathtt{M}_2)$ and $\mathcal{X}_{\mathrm{safe}}$ can be determined as follows: Safety holds for all $u_x^{(\mathrm{L})}$ if it holds for $u_x^{(\mathrm{L})} = u_{x,\min}$ until $v_x^{(\mathrm{L})} = v_{x,\min}$, i.e., max. braking. This is intuitively clear and a formal treatment is omitted due to lack of space. Then, in phase q_1, $u_x^{(\mathrm{L})} = 0$, such that (8) is undisturbed. For this dynamics, the maximal control invariant set is computed, again by using a tool like the MPT-toolbox. Next, sets of states which are robustly controllable to this control invariant set are computed. The union of these sets is a non-convex set $\mathcal{L}(\mathtt{M}_2)$, for which a polytopic inner approximation is derived by visual inspection of the projection of $\mathcal{L}(\mathtt{M}_2)$ on its three-dimensional affine hull. In these computations, set $v_{x,\min} = 0\,\mathrm{m\,s^{-1}}$ in order to allow braking to complete standstill in case of emergency.

Robust Maneuver Feasibility As outlined in Sec. 3.2:

Proposition 15. (Robust Maneuver Feasibility) Given a maneuver $\mathtt{M} = (\mathcal{C}, H_{\text{plan}}, \mathtt{HA})$ and an initial state x_0 of the vehicles in $\mathcal{C}$, a maneuver can be executed despite all admissible disturbances if $x_0 \in \mathcal{K}_{H_{\text{plan}}}(\mathtt{M})$.

Proof. Follows directly from Definition 4. □

Computation of these sets relies on the modeling of the disturbances in the maneuver. Based on Assumption 12, the only possible disturbance is from leading non-cooperative vehicles on their immediate cooperative followers. In this setting, the most adverse disturbance again is maximal braking of all non-cooperative, leading vehicles – if a maneuer is feasible for this setting, it is for all other disturbances. Set computations can be further simplified by assuming that $v_x = v_{x,\min}$ for all non-cooperative, leading vehicles from the beginning on. This clearly increases conservatism, but not too drastically, because for regular maneuvers (as opposed to emergency braking), setting $v_{x,\min} \gg 0\,\mathrm{m\,s^{-1}}$ in accordance to traffic rules on highways limits conservatism. By fixing the disturbance, the disturbed dynamics is transformed into affine, deterministic dynamics, for which controllable sets can be computed as discussed below.

In general, the controllable sets of Problem 3 are a non-convex union of polytopes:

$$\mathcal{K}_{H_{\text{plan}}}(\mathtt{M}) = \bigcup_{n=1}^{N} \mathcal{K}_{H_{\text{plan}}}\left(\mathtt{M}, \delta^{(n)}_{H_{\text{plan}}}\right), \tag{10}$$

in which a sequence of binary variables $\delta^{(n)}_{H_{\text{plan}}}(\cdot) := \big(\delta(t_0)\ \delta(t_1)\ \ldots\ \delta(t_{H_{\text{plan}}-1})\ \delta(t_{H_{\text{plan}}})\big)$ can be assigned to each single polytope $\mathcal{K}_{H_{\text{plan}}}\left(\mathtt{M}, \delta^{(n)}_{H_{\text{plan}}}\right)$ and N is the number of different admissible binary sequences.

Even though it is –in principle– possible to compute each $\mathcal{K}_{H_{\text{plan}}}\left(\mathtt{M}, \delta^{(n)}_{H_{\text{plan}}}\right)$ numerically exactly, using for example methods as provided by the MPT-toolbox, this is impractical for two reasons: Firstly, for a fixed $\delta^{(n)}_{H_{\text{plan}}}$, these computations are based on algorithms from computational geometry (especially Minkowski addition of polytopes) which are known to scale badly with increasing state space dimension. Secondly, depending on H_{plan} and the number of binary variables resulting from a specific maneuver formulation, N may be large, and likewise the number of polytopes to be computed.

The following remedy is proposed: In order to keep the number of sets to be computed sufficiently low, these are only computed for a subset $\mathcal{I} \subset \{1, 2, \ldots, N\}$ of all admissible δ-sequences. To facilitate computation for a given sequence $\delta^{(n)}(\cdot)$, $n \in \mathcal{I}$, approximate computations are employed as in (Eilbrecht and Stursberg, 2019), now extended by the well-established procedure from (Lotov et al., 2013). The basic idea is to extend a given initial approximation $\mathcal{K}^{(0)}$ by incrementally adding new states x_0, in which a new set $\mathcal{K}^{(i+1)}$ is defined as convex hull of the old set $\mathcal{K}^{(i)}$ and an added state: $\mathcal{K}^{(i+1)} = \mathtt{CH}\left(x_0, \mathcal{K}^{(i)}\right)$. If a solution to Problem 3 exists for an added state x_0 and the given $\delta^{(n)}(\cdot)$, then $\mathcal{K}^{(i+1)}$ is also a controllable set (Kerrigan, 2001). States to be added are chosen based on an estimate of the Hausdorff-distance between approximation and the (unknown) exact set $\mathcal{K}$; the largest values tend to increase the volume of the approximation the most. For more details (e.g. regarding convergence), cf. (Lotov et al., 2013). Clearly, the resulting approximated controllable set is an inner approximation of the exact set:

$$\tilde{\mathcal{K}}_{H_{\text{plan}}}(\mathtt{M}) := \bigcup_{n \in \mathcal{I}} \tilde{\mathcal{K}}_{H_{\text{plan}}}(\mathtt{M}, \delta^{(n)}_{H_{\text{plan}}}) \subseteq \mathcal{K}_{H_{\text{plan}}}(\mathtt{M}). \tag{11}$$

6. MANEUVER FORMULATIONS

In this section, exemplary formulations of the maneuvers in our maneuver library are detailed. Let $\mathtt{proj}_i(x)$ denote the projection of vector x on the state space of vehicle i.

Lane Keeping: Single Vehicle The simplest maneuver defines the behavior of a single vehicle which is driving on a lane without any other vehicles in the closer surrounding. The corresponding hybrid automaton contains only one phase q_0 and therefore no guards or transitions. Inputs, states, dynamics, and input and state constraints correspond to those defined in Sec. 4, while the invariant (equivalent to the initial set $\mathcal{X}_0$) depends on the lane the vehicle is driving on: on the left lane, $\text{inv}(q_0) = \left\{x \in \mathcal{X} | v_y^{(\text{E})} = 0,\ p_y^{(\text{E})} = p_{y,\max},\ v_x^{(\text{E})} \geq v_{x,\min,\text{left}}\right\}$, while on the right (merge) lane, the vehicle's state must also lie in the control invariant set $\mathcal{X}_{\text{merge}}$ from Sec. 5.1: $\text{inv}(q_0) = \left\{x \in \mathcal{X} \cap \mathcal{X}_{\text{merge}},\ v_y^{(\text{E})} = 0,\ p_y^{(\text{E})} = p_{y,\min}\right\}$.

Lane Keeping: Two Vehicles A slightly more complex maneuver defines the behavior of a vehicle E driving behind a non-cooperating vehicle (NL), while both keep the lane (similar to Fig. 4). The hybrid automaton of this maneuver also consists of only one phase q_0 and no transitions or guard sets. The dynamics now also incorporate those of (NL), leading to the state vector $x = \left[x^{(\text{E})\mathsf{T}}\ x^{(\text{NL})\mathsf{T}}\right]^{\mathsf{T}}$, while the input vector only consists of $u_x^{(\text{E})}$ because the inputs of NL cannot be controlled within the maneuver. Instead, conservative predictions according to Sec. 5.2 are used, leading to affine dynamics, thus $\mathbb{U}(q_0) = \mathcal{U}$. Similar to the single vehicle's lane keeping maneuver, the invariant depends on the lane the vehicles are driving on. Generally, $\mathcal{X}_0 = \text{inv}(q_0)$, with

$$\text{inv}(q_0) = \Big\{x \in \mathcal{X} \times \mathcal{X},\ v_y^{(\text{E})} = 0,\ \mathtt{proj}_{(\text{NL},\text{E})}(x) \in \mathcal{X}_{\text{safe}},$$
$$p_y^{(i)} = \begin{cases} p_{y,\max}\ (\text{left}) \\ p_{y,\min}\ (\text{right}) \end{cases}, v_x^{(\text{E})} \geq \begin{cases} v_{x,\min,\text{left}}\ (\text{left}) \\ v_{x,\min}\ (\text{right}) \end{cases}\Big\},$$

where "left" and "right" refer to the respective lanes.

Cooperative Merging The very core of our maneuver library is a cooperative merging maneuver. It defines roles for five vehicles, cf. Fig. 3: the non-cooperating leading (NL) and following vehicles (NF), the ego vehicle which is to merge (E), and the vehicles cooperating with it, (L) and

(F). Note that (E) is only allowed to merge between (F) and (L), while (NF) and (NL) are incorporated into the maneuver formulation in order to model the interaction between the cooperating vehicles and their surrounding according to Sec. 5.2. The associated hybrid automaton consists of three phases $\mathcal{Q} = \{q_0, q_1, q_2\}$ and has the topology as shown in Fig. 2, i.e., the set of admissible transitions is $\Theta = \{\theta_{0,1} = (q_0, q_1), \theta_{1,2} = (q_1, q_2)\}$. The phases correspond to: 1) (E) driving on the right lane with zero lateral velocity, 2) (E) changing lanes, not having reached the end of the acceleration lane $p_{\text{merge,max}}$ yet, and 3) (E) having passed $p_{\text{merge,max}}$.

Based on (5), the continuous state space of the automaton combines the states of the involved vehicles in:

$$x = \left[x^{(\text{NF})^\intercal}\ x^{(\text{F})^\intercal}\ x^{(\text{L})^\intercal}\ x^{(\text{NL})^\intercal}\ x^{(\text{E})^\intercal}\right]^\intercal \in \mathbb{X}. \tag{12}$$

The corresponding input vector is:

$$u = \left[u^{(\text{F})^\intercal}\ u^{(\text{L})^\intercal}\ u^{(\text{E})^\intercal}\right]^\intercal. \tag{13}$$

Note that the inputs of the non-cooperating vehicles (NF) and (NL) are not contained because they are beyond our control. Instead, conservative predictions are used according to Sec. 5.2. The matrices A, B, and a in (1) result from combining each vehicle's dynamics (5) appropriately according to the state vector (12), the input vector (13), and the conservative predictions. All phases have the same polytopic input constraints, which result from:

$$\mathbb{U}(q) = \mathcal{U} \times \mathcal{U} \times \mathcal{U}.$$

The state constraints in each mode combine: 1) general constraint sets $\mathcal{X}$ on the dynamics of single vehicles with 2) safety constraint sets $\mathcal{X}_{\text{safe}}$ in the state spaces of pairs of vehicles, and 3) location-dependent constraints. Let:

$$\mathcal{X}_{\text{lat}} = \left\{x | v_y^{(i)} = 0,\ p_y^{(i)} = p_{y,\max}\right\} \subset \mathbb{X},\ i \in \{\text{NF}, \text{F}, \text{L}, \text{NL}\},$$

and define $\tilde{\mathcal{X}} := (\mathcal{X} \times \mathcal{X} \times \mathcal{X} \times \mathcal{X} \times \mathcal{X}) \cap \mathcal{X}_{\text{lat}}$, then:

$$\begin{aligned}
\text{inv}(q_0) = \big\{x | x \in \tilde{\mathcal{X}},\ &\texttt{proj}_{(\text{NL},\text{L})}(x) \in \mathcal{X}_{\text{safe}},\\
&\texttt{proj}_{(\text{L},\text{F})}(x) \in \mathcal{X}_{\text{safe}},\ \texttt{proj}_{(\text{F},\text{NF})}(x) \in \mathcal{X}_{\text{safe}},\\
&v_x^{(\text{NF})} = v_{x,\max},\ v_x^{(\text{NL})} = v_{x,\min}\big\},\\
\text{inv}(q_1) = \big\{x | x \in \tilde{\mathcal{X}},\ &\texttt{proj}_{(\text{NL},\text{L})}(x) \in \mathcal{X}_{\text{safe}},\\
&\texttt{proj}_{(\text{L},\text{E})}(x) \in \mathcal{X}_{\text{safe}},\ \texttt{proj}_{(\text{E},\text{F})}(x) \in \mathcal{X}_{\text{safe}},\\
&\texttt{proj}_{(\text{F},\text{NF})}(x) \in \mathcal{X}_{\text{safe}}, p_x^{(\text{E})} \geq p_{\text{merge,min}},\\
&v_x^{(\text{NF})} = v_{x,\max},\ v_x^{(\text{NL})} = v_{x,\min}\big\},\\
\text{inv}(q_2) = \big\{x | x \in \text{inv}(q_1),\ &p_y^{(\text{E})} = p_{y,\max},\ v_y^{(\text{E})} = 0\big\}.
\end{aligned}$$

The initial set is $\mathcal{X}_0 = \{x \in \text{inv}(q_0) : p_y^{(\text{E})} = p_{y,\min}, v_y^{(\text{E})} = 0\}$, and $\mathcal{X}_\text{T} = \left\{x | p_y^{(\text{E})} = p_{y,\max},\ v_x^{(\text{E})} \geq v_{x,\text{min,left}}, v_y^{(\text{E})} = 0\right\}$, which enforces that (E) is driving on the left lane with zero lateral velocity and longitudinal velocity above $v_{x,\text{min,left}}$.

The guard sets corresponding to the transitions are:

$$g(\theta_{0,1}) = \{x | p_y^{(\text{E})} > p_{y,\min}\},\ g(\theta_{1,2}) = \{x | p_x^{(\text{E})} > p_{\text{merge,max}}\}.$$

Note that these guard sets are unbounded, low-dimensional, and overlapping, which, however, is not a problem according to the semantics of the hybrid automaton.

From this cooperative merging maneuver, other merging maneuvers can be derived by omitting vehicles and their states from the hybrid automaton and adapting its formulation accordingly. This allows to cover situations in which fewer vehicles are present than depicted in Fig. 3, thus making the maneuver library more flexible. For example, a non-cooperative merging maneuver which only considers (E) and a single non-cooperating vehicle (either (NF) or (NL) is defined, depending on whether (E) should merge before another vehicle or behind). This maneuver is non-cooperative in the sense that (E) must plan without other vehicles adapting their behavior, only relying on the conservative treatment as described in Sec. 5.2.

7. SIMULATION RESULTS AND DISCUSSION

A Matlab-based simulation environment has been implemented in order to analyze the effectiveness of the proposed framework. The test setup was chosen as follows: vehicles are generated every t seconds, where t is randomly chosen from the intervals $[3, 5]$ s (left lane) and $[3, 4]$ s (right lane). The initial longitudinal position is set to 0 and the lateral position to the respective lane center with zero lateral velocity. Longitudinal velocities are chosen in compliance with velocity-dependent safety constraints $\mathcal{X}_{\text{safe}}$ to a preceding vehicle, but are completely random apart from that (right lane) or as close to $100\,\text{km}\,\text{h}^{-1}$ as possible (left lane).

The simulation was run for 40 s, during which five vehicles were generated on the left lane and three on the right lane. Fig. 5 shows the final constellation of the vehicles, where the numbering reflects the order of their generation (2,4, and 7 started on the merge lane). The emerging behavior is as follows: after 19 s, vehicle 2 executes a cooperative merging maneuver with $H_{\text{plan}} = 13$, in a coalition with vehicles 3, 5, 6, and 8 (in which 3 and 8 are non-cooperative). The resulting longitudinal velocities of the cooperating vehicles 2, 5, and 6 are shown in Fig. 6. The cooperative nature of the maneuver is illustrated by the fact that the leading vehicle 5 accelerates slightly in order to allow vehicle 2 to merge behind it. The following vehicle 6, on the other hand, is so far behind that it does not need to adapt its velocity to open a gap. Vehicles 4 and 7 are unable to find a gap into which they can merge cooperatively, such that they wait until all vehicles have passed and merge behind after 25.5 s and 34.5 s, with $H_{\text{plan}} = 18$ and $H_{\text{plan}} = 14$, respectively. These maneuvers take so long because they already include the process of slowing down and waiting on the merge lane. The lateral velocities of the merging vehicles 2, 4, and 7 are given in Fig. 8, while Fig. 7 compares the actual longitudinal distances (dotted) for a selected pair of vehicles to the exact (thin black) and approximated (thick gray) safety distances. The plot shows that the constraints are never violated and demonstrates that the approximation is not overly conservative.

8. CONCLUSION

The proposed approach is able to solve practically relevant problems such as vehicle coordination, being both computational efficient and theoretically sound with regard to feasibility of the control algorithm. Practical applicability holds despite the conservative, cautious treatment of non-cooperating traffic participants, which is somewhat

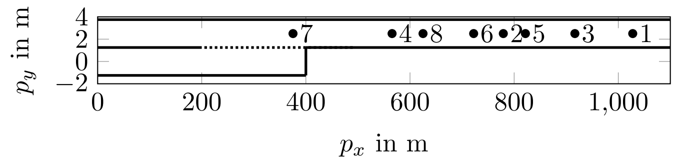

Fig. 5. Constellation after 40 s of simulation (bullets mark vehicles; numbers indicate order of their generation).

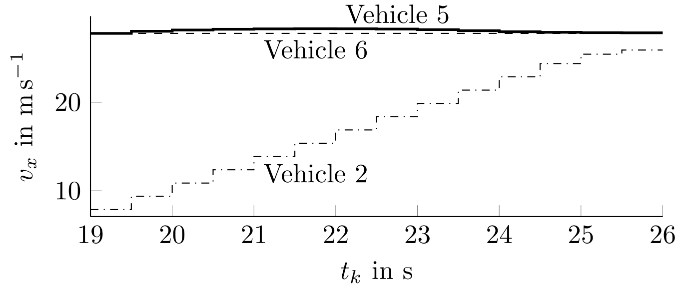

Fig. 6. Longitudinal velocities of cooperating vehicles during merging of vehicle 2.

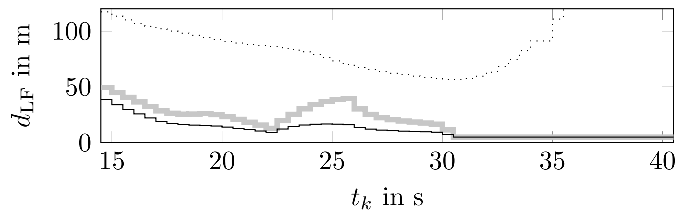

Fig. 7. Longitudinal distance (dotted) and required safety distance (gray: conservative; black: exact) for vehicle pairing: L = 4, F = 7.

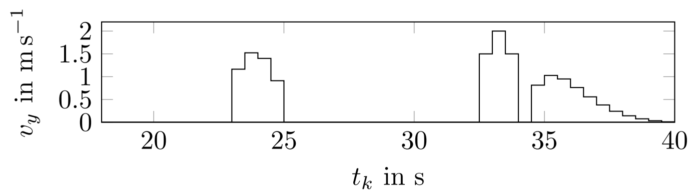

Fig. 8. Lateral speed of vehicles 2, 4, and 7 (left to right).

surprising because conservative problem formulations often tend to prevent practical application in autonomous driving. Computation of controllable sets inherently suffers from the curse of dimensionality and remains a challenge for larger sets of vehicles. This necessitates careful maneuver formulation in order to keep the state space dimensionality as low as possible.

While this paper has mainly focused on establishing feasibility, future work will aim at the development of more elaborate scheduling algorithms. In order to compare the performance of different approaches, realization within a large-scale traffic simulation environment (e.g., SUMO (Krajzewicz et al., 2002)), which is more performant than our Matlab-based implementation, is desirable.

REFERENCES

Bertsekas, D. (1995). *Dynamic programming and optimal control*, volume 1. Athena scientific.

de La Fortelle, A. (2010). Analysis of reservation algorithms for cooperative planning at intersections. In *Proc. Intelligent Transportation Systems Conf.*, 445–449. IEEE.

Eilbrecht, J. and Stursberg, O. (2018). Optimization-based maneuver automata for cooperative trajectory planning of autonomous vehicles. In *Proc. European Control Conf.*, 82–88.

Eilbrecht, J. and Stursberg, O. (2019). Reducing computation times for planning of reference trajectories in cooperative autonomous driving. In *Proc. Intelligent Vehicle Symp.*, 114–120. IEEE.

González, D., Pérez, J., Milanés, V., and Nashashibi, F. (2016). A review of motion planning techniques for automated vehicles. *IEEE Trans. Intelligent Transportation Systems*, 17(4), 1135–1145.

Kerrigan, E.C. (2001). *Robust constraint satisfaction: Invariant sets and predictive control.* Ph.D. thesis, Citeseer.

Krajzewicz, D., Hertkorn, G., Rössel, C., and Wagner, P. (2002). Sumo (simulation of urban mobility)-an open-source traffic simulation. In *Proc. Middle East Symp. on Simulation and Modelling*, 183–187.

Kuwata, Y. and How, J.P. (2010). Cooperative distributed robust trajectory optimization using receding horizon milp. *IEEE Trans. Control Systems Technology*, 19(2), 423–431.

Kvasnica, M., Grieder, P., Baotic, M., and Morari, M. (2004). Multi-parametric toolbox (mpt). In *Int. Workshop on Hybrid Systems: Computation and Control*, 448–462. Springer.

LaValle, S.M. (2006). *Planning Algorithms.* Cambridge University Press.

Lotov, A.V., Bushenkov, V.A., and Kamenev, G.K. (2013). *Interactive decision maps: Approximation and visualization of Pareto frontier.* Springer.

Manzinger, S., Leibold, M., and Althoff, M. (2017). Driving strategy selection for cooperative vehicles using maneuver templates. In *Proc. Intelligent Vehicles Symp.*, 647–654. IEEE.

Qian, X., Altché, F., Bender, P., Stiller, C., and de La Fortelle, A. (2016). Optimal trajectory planning for autonomous driving integrating logical constraints: An miqp perspective. In *Proc. Intelligent Transportation Systems Conf.*, 205–210. IEEE.

Schouwenaars, T., De Moor, B., Feron, E., and How, J. (2001). Mixed integer programming for multi-vehicle path planning. In *Proc. European Control Conf.*, 2603–2608. IEEE.

Schürmann, B., Heß, D., Eilbrecht, J., Stursberg, O., Köster, F., and Althoff, M. (2017). Ensuring drivability of planned motions using formal methods. In *Proc. Intelligent Transportation Systems Conf.*, 1–8. IEEE.

Stursberg, O. and Krogh, B.H. (2003). Efficient representation and computation of reachable sets for hybrid systems. In *Int. Workshop on Hybrid Systems: Computation and Control*, 482–497. Springer.

Williams, H.P. (2013). *Model Building in Mathematical Programming.* John Wiley & Sons.

Zhang, Y. and Cassandras, C.G. (2019). Decentralized optimal control of connected automated vehicles at signal-free intersections including comfort-constrained turns and safety guarantees. *Automatica*, 109.