

A Long-Short-Term Mixed-Integer Formulation for Highway Lane Change Planning

Rudolf Reiter, Armin Nurkanović, Daniele Bernardini, Moritz Diehl, Alberto Bemporad, *Fellow, IEEE*

Abstract—This work considers the problem of optimal lane changing in a structured multi-agent road environment. A novel motion planning algorithm that can capture long-horizon dependencies as well as short-horizon dynamics is presented. Pivotal to our approach is a geometric approximation of the long-horizon combinatorial transition problem which we formulate in the continuous time-space domain. Moreover, a discrete-time formulation of a short-horizon optimal motion planning problem is formulated and combined with the long-horizon planner. Both individual problems, as well as their combination, are formulated as mixed-integer quadratic programs (MIQPs) and solved in real-time by using state-of-the-art solvers. We show how the presented algorithm outperforms two other state-of-the-art motion planning algorithms in closed-loop performance and computation time in lane changing problems. Evaluations are performed using the traffic simulator SUMO, a custom low-level tracking model predictive controller, and high-fidelity vehicle models and scenarios, provided by the CommonRoad environment.

Index Terms—Autonomous Vehicles, Motion Planning, Control and Optimization, Vehicle Control Systems.

I. INTRODUCTION

IN recent years many approaches have been proposed for vehicle motion planning in structured multi-lane road environments. However, considering combinatorial long-term dependencies and providing optimal trajectories subject to dynamic constraints in real-time remains a challenging problem. In fact, even deterministic two-dimensional motion planning problems with rectangular obstacles are NP-hard [1], [2].

This work proposes a novel iterative planning algorithm, referred to as long-short-term motion planner (LSTMP) that reduces the combinatorial complexity by splitting the problem into a short-term motion planning formulation (STF) and a long-term motion planning formulation (LTF), both solved by one MIQP, cf. Fig. 1. The STF aims at optimizing a four-state discrete-time trajectory of a point-mass model including obstacle constraints, similar to the formulations of [3], [4]. The STF trajectory is computed for a shorter horizon to approximate a maximum of one lane change. In contrast, the LTF aims at obtaining optimal lane transitions, defined by the transition times and longitudinal transition positions, which are both continuous variables. These lane transitions are used for

R. Reiter, A. Nurkanović and M. Diehl are with the University of Freiburg, 79110 Freiburg i. B., Germany (e-mails: {rudolf.reiter, armin.nurkanovic, moritz.diehl}@imtek.uni-freiburg.com).

D. Bernardini is with ODYS S.r.l., 20159 Milano, Italy (e-mail: daniele.bernardini@odys.it).

A. Bemporad is with the IMT School for Advanced Studies Lucca, 55100 Lucca, Italy (e-mail: alberto.bemporad@imtlucca.it).

This work was supported by EU via ELO-X 953348, by DFG via Research Unit FOR 2401, project 424107692 and 525018088 and by BMWK via 03EI4057A and 03EN3054B.

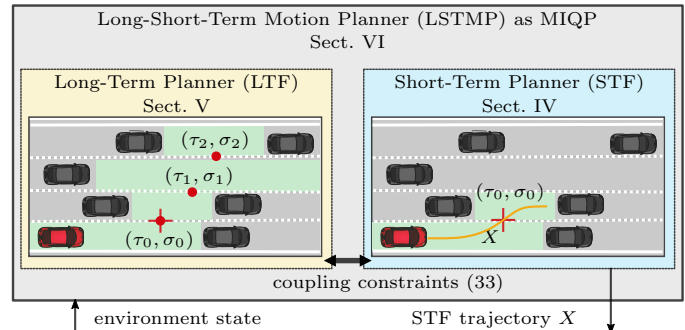


Fig. 1. Overview of the proposed MIQP formulation for motion planning, referred to as LSTMP. The MIQP consists of long-term and short-term planning formulations where the decision variables of both are coupled through consistency constraints. The short-term decision variables include a continuous point-mass model trajectory to approximate a single lane change. The long-term decision variables account for selecting gaps between SVs on each lane.

long-term planning, i.e., the choice of gaps between vehicles on several consecutive lanes. Reachability and the choice of transition gaps on consecutive lanes are modeled by disjunctive programming.

The planned trajectory of the STF and the transitions of the LTF are formulated consistently, i.e., a transition point constrains the point-mass model trajectory to the corresponding lane. Contrary to strict hierarchical decomposition, the coarser approximation of the high-level plan cannot be infeasible for the low-level planner.

A challenge of state-of-the-art motion planners is the scaling of computational complexity with the horizon length [3], which makes long-horizon planning most often intractable. Within the formulation of the LTF, the locations of transitions in time and position are continuous. The proposed modeling uses integer variables only related to the gaps between vehicles on each lane. Therefore, the number of integer variables does not scale with the horizon length within the LTF. Consequently, the constant small number of integer variables, even for long-term predictions, allows for fast computation times of the algorithm.

Evaluations of the proposed approach are performed with both deterministic and interactive closed-loop simulations that involve a CommonRoad [5] vehicle model, a low-level nonlinear model predictive controller (NMPC), and interactive traffic that is simulated with sumo.

A. Related work

An abundance of fundamentally different techniques address highway motion planning and were reviewed by [6] and, more recently, by [7], [8] and particularly for deep learning in [9], [10]. The authors in [6] identified geometric, variational, graph-search, and incremental search methods as fundamental planning categories, whereas the more recent and exhaustive survey [7] adds a major part on artificial intelligence (AI), among further refinements.

Historically, geometric and rule-based approaches were more dominant. Parametric curves are used in highly structured environments, such as highways, due to their simplicity and ease of alignment with the road geometry [11]–[13]. However, the motion plans are usually conservative and unable to cope with complex environments [7].

In several works, the state space is discretized and some sort of graph search is performed [6], [8]. Probabilistic road maps [14], [15], Dijkstra graph search and similar, rapidly exploring random trees [16]–[18] are common in highway motion planning. Nonetheless, they suffer from the curse of dimensionality, the inability to handle dynamics, a poor connectivity graph, and a poor repeatability of results [7]. By using heuristics, hybrid A* [19]–[21] aims to avoid the problem of high-dimensional discretization in graph-search. Choosing an admissible heuristic is challenging and a time-consuming graph generation is performed in each iteration. Moreover, authors try to improve graph- or sampling-based algorithms by combining them with learning-based methods [17], [22].

Optimization-based methods can successfully solve motion planning problems in high-dimensional state spaces in real-time [23]. They are appealing due to numerous advantages, e.g., consideration of dynamics and constraints, adaptability to new scenarios, finding and keeping solutions despite environmental changes, and taking into account complex scenarios. Pure derivative-based methods are often restricted to convex problem structures or sufficiently good initial guesses. Highway motion planning is highly non-convex. Nonetheless, by introducing integer variables, the problem can be formulated as an MIQP [3], [4], [24], [25] and solved by dedicated high-performance solvers, such as `gurobi` [26]. Yet, the planning horizon of MIQP with a fixed discrete-time trajectory is still limited due to the increasing number of integer variables for increased horizon lengths. Therefore, keeping the computation time limited remains a challenge, [7].

One successful structure exploitation for solving the highway motion planning problem faster is the decomposition of the state space into *spatio-temporal* driving corridors [4], [25], [27]–[30] with *simple* obstacle predictions. Still highly non-convex, the region can be decomposed into convex cells [29], [30] or used in a sampling-based planner [25].

To leverage the computational burden further, without sacrificing significantly the overall performance, the presented approach uses a short-horizon planning similar to [3] and [4], adding a long-horizon coarse geometric approximation in the spatio-temporal domain (see Fig. 1). The proposed STF differs from [4] by using only one binary variable per time step for

the first lane change and more accurately modeling occupied regions, that also consider braking due to preceding obstacles. Moreover, the consecutive gap is not fixed as in [4] but determined by the LTF. The idea of combining two horizons was presented in [31], yet not related to combinatorial motion planning and hierarchically decomposed in [21] using a graph-based planner.

AI-based methods often use exhaustive simulations to train neural networks (NNs) by reinforcement learning (RL) [10] or use expert data, such as data collected from human drivers, to perform imitation learning [32], [33]. They often struggle to consider safety critical constraints and adapt to environment changes [7], [9], [33]. Furthermore, sim-to-real challenges apply [34] since these methods are mainly trained within simulations. An advantage includes the capability of using raw sensor inputs, such as camera images and the low computational requirements of trained NNs [9].

The performance of the LSTMP is compared to a state-of-the-art hybrid A* method [20], which can be classified as both a deterministic planning AI and graph-search method [35], and to the mixed-integer motion planning and decision maker (MIP-DM) [3] which is a comparable state-of-the-art optimization-based method.

B. Contribution

This paper contributes a novel algorithm for optimal lane-changing highway maneuvers. Compared to other highway motion planning algorithms, the proposed lane change motion planner approximates long-term dependencies in the spatio-temporal (ST)-space, where the computational burden is independent of the position and the time a lane change occurs. Moreover, we solve the problem involving the long-term approximation as a consistent single problem and, therefore, avoid a problematic decoupling. The closed-loop performance is improved by 15% compared to [3] and [20] and the average computation time is lowered in randomized interactive simulations by two orders of magnitude. Compared to [3], the number of integer variables used within the underlying optimization problem reduces from $O(N_{\text{veh}}N)$ to $O(N_{\text{veh}} + N)$ for a number of N_{veh} SVs and N discrete-time prediction steps, with a comparable closed-loop performance on the evaluated scenarios.

C. Outline

In Sect. II, important background concepts are defined that are used throughout the paper. In Sect. III, the general problem definition, related assumptions, and simplifications are introduced. Next, the planning approach is described in Sect. IV to VI and evaluated in Sect. VII. The approach is discussed and conclusions are drawn in Sect. VIII.

II. PRELIMINARIES AND NOTATION

The set of non-negative real numbers is denoted by $\mathbb{R}^+ = \{x \in \mathbb{R} \mid x \geq 0\}$ and non-negative integers by $\mathbb{N}_0 = \{x \in \mathbb{Z} \mid x \geq 0\}$. Integer sets are written as $\mathbb{N}_{[m:n]} = \{z \in \mathbb{N}_0 \mid m \leq z \leq n\}$ with $m < n$. By using the notation $f(x; y)$ in the

context of optimization problems, we denote the dependency of function f on variables x and parameters y . The convex hull $C \subseteq \mathbb{R}^n$ of two polygons $A \subseteq \mathbb{R}^n$ and $B \subseteq \mathbb{R}^n$ is $C = \text{conv}(A, B)$. We use the floor function $\lfloor x \rfloor$ for rounding a number $x \in \mathbb{R}$ to the largest smaller integer and the ceil function $\lceil x \rceil$ for rounding to the smallest larger integer.

A. Propositional logic and mixed-integer notation

For a given compact set $\mathcal{X} \subset \mathbb{R}$ and continuous function $f : \mathcal{X} \rightarrow \mathbb{R}$, let $\overline{M} \geq \max_{x \in \mathcal{X}} f(x)$ and $\underline{M} \leq \min_{x \in \mathcal{X}} f(x)$ denote an upper and lower bound of $f(x)$ on \mathcal{X} , respectively. The following properties hold for a given $f : \mathcal{X} \rightarrow \mathbb{R}$ [36], [37].

Property II.1. For a product $y = \beta f(x)$, with $y \in \mathbb{R}$, the following equivalence holds for all $x \in \mathcal{X}$ and $\beta \in \mathbb{N}_{[0:1]}$:

$$y = \beta f(x) \Leftrightarrow \begin{cases} y \leq \overline{M}\beta, \\ y \geq \underline{M}\beta, \\ y \leq f(x) - \underline{M}(1 - \beta), \\ y \geq f(x) - \overline{M}(1 - \beta). \end{cases} \quad (1)$$

Property II.2. The implication $[\beta = 1] \implies [f(x) \geq 0]$ of a binary variable $\beta \in \mathbb{N}_{[0:1]}$ that activates constraint $f(x) \geq 0$, is formulated as

$$f(x) \geq \underline{M}(1 - \beta). \quad (2)$$

Property II.3. The implication $[f(x) > 0] \implies [\beta = 1]$ of a binary variable $\beta \in \mathbb{N}_{[0:1]}$ that gets activated if constraint $f(x) > 0$ is valid, is formulated as

$$f(x) \leq \overline{M}\beta. \quad (3)$$

Property II.4. The disjunction $\bigvee_{i=1}^N [f_i(x) \geq 0]$ is formulated by adding N binary variables $\beta_i \in \mathbb{N}_{[0:1]}$ with $i \in \mathbb{N}_{[1:N]}$, and the conditions

$$\begin{aligned} [\beta_i = 1] &\implies [f_i(x) \geq 0], \forall i \in \mathbb{N}_{[1:N]} \\ \sum_{i=1}^N \beta_i &\geq 1 \end{aligned} \quad (4)$$

B. Chebychev center

The Chebychev center (CC) of a polyhedron $P = \{x \mid Ax \leq b\}$, with $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$, is the center $x^* \in \mathbb{R}^n$ of the largest ball $B(x^*, r^*) = \{x \mid \|x^* - x\| \leq r^*\}$ contained in P [38]. The radius r^* is called the Chebychev radius. With A_i and b_i being the i -th row of A and b , respectively, the CC and Chebychev radius can be computed by solving the linear program

$$\min_{r, x} \quad -r \quad (5a)$$

$$\text{s.t.} \quad A_i x + r \|A_i\|_2 \leq b_i \quad i \in \mathbb{N}_{[1:m]}, \quad (5b)$$

$$r \geq 0 \quad (5c)$$

III. GENERAL LANE CHANGING PROBLEM

The general problem for lane changing is stated as an optimal control problem (OCP), similar to [39]. For a multi-lane environment, a total of L lanes are defined by curvilinear center curves and a lane width d_{lane} . Moreover, we assume the existence of a parametric function $\gamma : \mathbb{R}^+ \rightarrow \mathbb{R}^2$ for the right-most reference lane that maps a longitudinal path coordinate s to a Cartesian point. The reference lane is parameterized by a vector θ_γ which is included in the road geometry parameters $\theta := (\theta_\gamma, d_{\text{lane}})$. We consider a vehicle model in the Frenet coordinate frame [40]–[42] with states $x(t) \in \mathbb{R}^{n_x}$ and inputs $u(t) \in \mathbb{R}^{n_u}$, whose trajectories are governed by the nonlinear ordinary differential equation (ODE) $\dot{x} = \xi(x(t), u(t))$ with the initial condition $x(t_0) = x_0$. Using Frenet coordinates poses mild assumptions on the maximum value of the curvature, cf. [43]. Among others, the state of the Frenet model includes a longitudinal position state s , a lateral position state n , a velocity v and a heading angle mismatch α [42].

States and controls are constrained by physical limitations depending on θ , which are expressed by admissible sets $\mathcal{X}(t; \theta)$ and $\mathcal{U}(t)$.

For M vehicles on each lane, we consider $N_{\text{veh}} = LM$ SVs with states $x_i^{\text{sv}}(x(t), t)$ for $i \in \mathbb{N}_{[1:N_{\text{veh}}]}$ that depend on the planned ego trajectory $x(t)$. Note that the dependency of the states $x_i^{\text{sv}}(x(t), t)$ on the ego state $x(t)$ is due to the interaction of the ego vehicle with SVs and is a major source of complexity [44], [45]. We assume that the obstacle-free set can be approximated by $\mathcal{X}_{\text{free}}(x(t), t)$. In the latter sections IV and V, we explain how to define the set $\mathcal{X}_{\text{free}}(x(t), t)$ in an MIQP model.

In compliance with [30], multiple general objectives are proposed in the Frenet coordinate frame in order to define the desired behavior. A goal lane index $\tilde{l} \in \mathbb{N}_{[1:L]}$ and a reference velocity $\tilde{v} \in \mathbb{R}^+$, define the goal parameters

$$\Theta := (\tilde{l}, \tilde{v}).$$

Curvilinear reference paths are expressed as constant lateral references \tilde{n}_i for $i \in \mathbb{N}_{[1:L]}$. One desired behavior is to track the lateral lane reference the vehicle is currently driving on. The current reference lane index $l(n)$ w.r.t. the current lateral state n is uniquely determined by

$$l(n) = \left\lceil \frac{n}{d_{\text{lane}}} + \frac{1}{2} \right\rceil. \quad (6)$$

Note that determining the lane as in (6) within an optimization problem is not trivial and requires for instance the use of additional integer variables, as shown in Sect IV. By using the weights w_n and w_v , the cost of tracking the reference lane index $l(n(t))$ and longitudinal reference speed \tilde{v} is

$$\begin{aligned} g_{\text{ref}}(x(t), u(t); \theta, \Theta) = & \\ & w_n \left(n(t) - (l(n(t)) - 1) d_{\text{lane}} \right)^2 + \\ & w_v \left(v(t) - \tilde{v} \right)^2 + u^\top(t) R u(t), \end{aligned} \quad (7)$$

which includes a quadratic penalty on the input u , with the positive definite weighing matrix $R \in \mathbb{R}^{n_u \times n_u}$.

Next, a cost for the distance to the goal lane $\tilde{l} \in \mathbb{N}_{[1:L]}$ with a weight $w_g \in \mathbb{R}^+$ is added, which is the main objective of the presented planner and written as

$$g_{\text{lane}}(x(t); \Theta) = w_g |l(n(t)) - \tilde{l}|. \quad (8)$$

In the proposed approach, only lane changes towards the goal lane \tilde{l} are considered.

Finally, the objective functional is

$$J(x(\cdot), u(\cdot); \theta, \Theta) := \int_{t=t_0}^{\infty} \left(g_{\text{ref}}(x(t), u(t); \theta) + g_{\text{lane}}(x(t); \Theta) \right) dt, \quad (9)$$

and the considered general optimal control problem that is approximately solved by the proposed approach is

$$\min_{x(\cdot), u(\cdot)} J(x(\cdot), u(\cdot); \theta, \Theta) \quad (10a)$$

s.t.

$$x(t_0) = x_0, \quad (10b)$$

$$\dot{x} = \xi(x(t), u(t)), \quad t \in [t_0, \infty), \quad (10c)$$

$$x(t) \in \mathcal{X}_{\text{free}}(x(t), t; \theta) \cap \mathcal{X}(t; \theta), \quad t \in [t_0, \infty), \quad (10d)$$

$$u(t) \in \mathcal{U}(t), \quad t \in [t_0, \infty). \quad (10e)$$

A. Assumptions and simplifications

Several assumptions and simplifications are made for the proposed planning approach in order to approximate (10) by an MIQP. As a major simplification, the vehicle dynamics are formulated by a point-mass model with mass m in a Frenet coordinate frame [43], including the longitudinal and lateral position states s and n , as well as associated velocities v_s and v_n , with $x = [s, n, v_s, v_n]^\top$, and acceleration inputs a_s and a_n , with $u = [a_s, a_n]^\top$. The dynamics are modeled by

$$\dot{x} = [v_s, v_n, \frac{1}{m}a_s, \frac{1}{m}a_n]^\top. \quad (11)$$

We assume the absolute value of the curvature $\kappa(s)$ and its derivative $\kappa'(s)$ to be small for highway roads and, therefore, the acceleration in the Frenet coordinate frame is approximately equal to the acceleration in Cartesian coordinate frame [43]. This model was empirically shown to be valid for the cases where vehicles are not driving at their dynamical limits [4] and motivated in several other works, e.g., [3], [31], [46]–[48]. Critical evasion maneuvers are passed to a NMPC within the presented structure.

Assumption III.1. *Lane-changes of SVs can be detected.*

Ass. III.1 can be satisfied by perception techniques described in [6], [49] or by vehicle-to-vehicle communication.

Assumption III.2. *Considering two vehicles in the same lane, the rear vehicle is responsible for avoiding collisions. The leading vehicle must maintain general deceleration limits. Vehicles that change lanes must give way to vehicles on the lane they are changing to.*

Taking into account interactions among traffic participants within $\mathcal{X}_{\text{free}}(x(t), t; \theta)$ is essential for certain maneuvers in order to avoid prohibitive conservatism [50]. However, the interdependence of plans among interactive agents leads to computationally demanding game-theoretic problems [45], [51]. Similar to [4], the leader-follower interaction is simplified by ignoring collision constraints of followers on the same lane at the current state, leaving the responsibility for collision avoidance to the follower. Other SVs that are not following on the current lane are considered obstacles independent of the ego plan as long as these SVs are on adjacent lanes or in front of the ego vehicle.

The following assumptions consider constraints in the three-dimensional SLT-space [30], i.e., the space of the longitudinal and the lateral Frenet position states s and n and time t .

On each lane, a maximum of M vehicles is considered. We use indices $i \in \mathbb{N}_{[1:LM]}$ for the enumeration of the resulting maximum LM vehicles on the lanes in ascending order, starting from lane $l = 1$ from rear to front. The free space on the back of each vehicle along the lane is referred to as *gap* and enumerated according to the leading vehicle index, cf., Fig. 2. A number of L indices are added for the gaps in front of the first vehicle on each of the L lanes. Therefore the number of gap indices is $L(M + 1)$. The function $l_{\text{veh}}(i)$ returns the lane index of vehicle i . The function $l_{\text{gap}}(i)$ returns the lane index of gap i . The function $M_{\text{lane}}(i)$ returns the total number of vehicles on lane $l = l_{\text{veh}}(i)$ for a vehicle with index i .

In the following, we assume that $\mathcal{X}_{\text{free}}(x(t), t)$ can be partitioned into sets $\mathcal{X}_{\text{free}}^{\mathcal{I}}(x(t), t)$ related to SVs \hat{x}_i^{sv} within indices in the set $i \in \mathcal{I} \subseteq \mathbb{N}_{[1:LM]}$, with $\mathcal{X}_{\text{free}}(x(t), t) \subseteq \mathcal{X}_{\text{free}}^{\mathcal{I}}(x(t), t)$ and $\mathcal{X}_{\text{free}}(x(t), t) = \mathcal{X}_{\text{free}}^{\mathbb{N}_{[1:LM]}}(x(t), t)$.

By inflating the obstacle shapes and lane boundaries by a safe distance according to all allowed configurations of the ego and the SVs, the planning problem can be formulated by a point-wise set exclusion of the curvilinear ego position states $s(t)$ and $n(t)$ [2].

Assumption III.3. *For all SVs not driving at the current ego lane l , defined by the index set $\mathcal{I}(l) = \{k \in \mathbb{N}_{[1:LM]} \mid l \neq l_{\text{veh}}(k)\}$, an obstacle-free set $\mathcal{N}_l = \{n \in \mathbb{R} \mid \underline{n}_l \leq n \leq \bar{n}_l\}$ w.r.t. the ego lateral state n can be found such that*

$$n \in \mathcal{N}_l \implies x(t) \in \mathcal{X}_{\text{free}}^{\mathcal{I}(l)}(x(t), t).$$

Ass. (III.3) is used to define collision avoidance constraints to vehicles on adjacent lanes by formulating constraints on the lateral state n . Without further details, it is assumed that the bounds in Ass. III.3 are *tight* enough to contain *most* of the adjacent lanes as free space, i.e., $\mathcal{N}_l \neq \emptyset$.

Assumption III.4. *Given an SV with index i , upper position bounds \bar{s}_i^{sv} and velocity bounds \bar{v}_i^{sv} can be found that define the collision-free set*

$$\mathcal{S}_i^- = \{(s, t) \in (\mathbb{R}^+ \times \mathbb{R}^+) \mid s \geq \bar{s}_i^{\text{sv}} + t\bar{v}_i^{\text{sv}}\}. \quad (12)$$

such that it holds that

$$(t, s(t)) \in \mathcal{S}_i^- \implies x(t) \in \mathcal{X}_{\text{free}}^{\{i\}}(x(t), t).$$

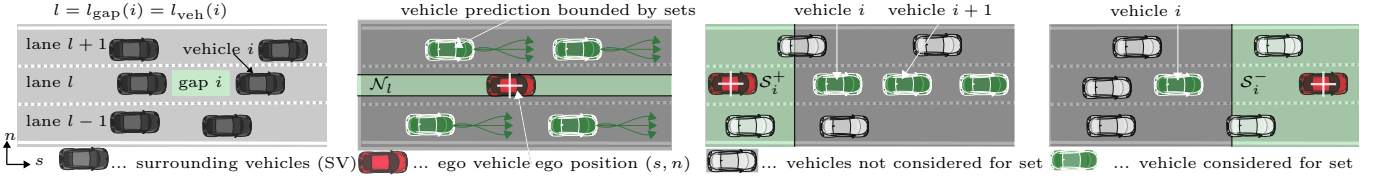


Fig. 2. The first figure shows the enumeration of lanes and gaps and the rightmost three figures show the sets related to free spaces. Surrounding vehicles (SVs) are uniquely enumerated. Gaps are the free spaces on a lane w.r.t. SVs and are enumerated according to the leading vehicles. An additional index is used for each frontmost gap. The sets \mathcal{N}_l , \mathcal{S}_i^+ and \mathcal{S}_i^- define half-spaces in the SLT-space and are plotted in green for the position dimensions. The sets are *tightened* to include all configurations of the SVs and ego vehicle to allow collision-free planning with a point-mass model. All leading vehicles on the same lane are considered to construct the set \mathcal{S}_i^+ , since any slower vehicle requires all following vehicles to brake. For the following vehicle set \mathcal{S}_i^- only the closest vehicle to gap i is considered, since preceding ones are assumed to not influence leading vehicles.

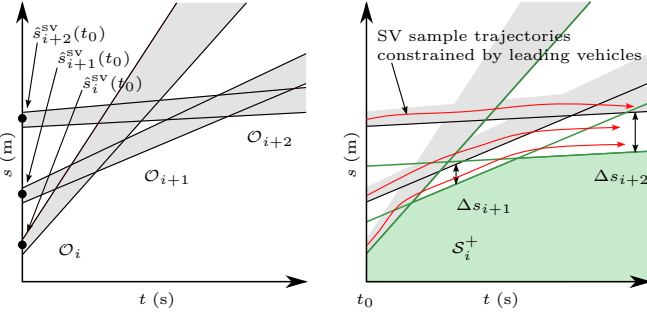


Fig. 3. Construction of longitudinal obstacle-free space \mathcal{S}_i^+ for an SV with index i and two leading vehicles. The left plot shows the nominal prediction sets \mathcal{O}_i . The right plot shows the blocking lower-bound set, enforced on the following vehicles. Red trajectories are plotted corresponding to samples of actually driven trajectories.

Assumption III.5. For all SVs on lane l , with indices $i \in \mathbb{N}_{[1:M]}$, lower position bounds $\underline{s}_i^{\text{sv}}$, velocity bounds $\underline{v}_i^{\text{sv}}$ and leading vehicle distances Δs_i can be found that define the set

$$\mathcal{S}_i^+ = \left\{ (s, t) \in (\mathbb{R}^+ \times \mathbb{R}^+) \mid \begin{aligned} s &\leq \underline{s}_i^{\text{sv}} + t\underline{v}_i^{\text{sv}} + \sum_{k=i+1}^{M_{\text{lane}}(i)} \underline{s}_k^{\text{sv}} + t\underline{v}_k^{\text{sv}} - \Delta s_{k-1} \end{aligned} \right\}. \quad (13)$$

such that for $0 \leq t \leq \bar{t}$ it holds that

$$(t, s(t)) \in \mathcal{S}_i^+ \implies x(t) \in \mathcal{X}_{\text{free}}^{\{i, \dots, M\}}(x(t), t).$$

Similar assumptions are made in related work, e.g., [4], [27], and with a more accurate lateral shape in [3]. The bounds in Ass. III.5 approximate a distribution that is generated by the *intelligent driver model* [52], where a vehicle either drives or approaches a range around a reference velocity \hat{v}_i^{sv} , with $\underline{v}_i^{\text{sv}} \leq \hat{v}_i^{\text{sv}} \leq \bar{v}_i^{\text{sv}}$, or drives within a certain distance Δs_i to a slower leading vehicle [4], [27], cf., Fig. 3. The set

$$\mathcal{O}_i = \left\{ (s, t) \in (\mathbb{R}^+ \times \mathbb{R}^+) \mid \underline{s}_i^{\text{sv}} + t\underline{v}_i^{\text{sv}} \leq s \leq \bar{s}_i^{\text{sv}} + t\bar{v}_i^{\text{sv}} \right\}$$

is referred to the nominal SV prediction set in the absence of leading vehicles. In each planning step, the bounds of \mathcal{O}_i are updated based on the current SV state \hat{x}_i^{sv} , where for the velocity bounds it additionally holds that $\underline{v}_i^{\text{sv}} \leq \hat{v}_i^{\text{sv}} \leq \bar{v}_i^{\text{sv}}$ and for the position bounds $\underline{s}_i^{\text{sv}} \leq \hat{s}_i^{\text{sv}} \leq \bar{s}_i^{\text{sv}}$ holds.

Assumption III.6. The duration of a lane-change t_{lc} is upper-bounded by $t_{\text{lc}} \leq \bar{t}_{\text{lc}}$.

For a concise notation, no offsets are assumed, i.e., the current lane and gap index are 1, the current planning time is assumed at zero seconds and the initial lateral reference and longitudinal estimated state are set to 0, therefore $\tilde{n}_0 = 0$ and $\hat{s} = 0$.

B. Obstacle-free set approximations

In the following, convex obstacle-free sets in the SLT-space are defined as intersections of the sets \mathcal{S}_i^+ , \mathcal{S}_i^- and \mathcal{N}_l , which serve as a basis for the proposed LSTM.

Two convex three-dimensional sets in the SLT-space are used to formulate a lane change from lane l and gap index g on the same lane, i.e., $l = l_{\text{gap}}(g)$, to gap index g^+ on the next lane $l+1$, i.e., $l+1 = l_{\text{gap}}(g^+)$. First, for lane-keeping, obstacle avoidance reduces to the problem of staying within the current lane boundaries (Ass. III.3), ignoring following SVs on the same lane (Ass. III.2) and consider leading SVs, with an upper-bound related to (13), stated as the convex obstacle-free set over the longitudinal and lateral position and time

$$\mathcal{F}_g^+ = \left\{ (t, s, n) \mid (t, s) \in \mathcal{S}_g^+, \quad n \in \mathcal{N}_{l_{\text{gap}}(g)} \right\}. \quad (14)$$

Next, the free set for a lane change is defined in the two-dimensional ST-space, which is a subspace of the SLT-space, as

$$\mathcal{S}_{g, g^+}^{\text{lc}} = \left\{ (t, s) \in \mathcal{S}_g^+ \cap \mathcal{S}_{g^+}^+ \cap \mathcal{S}_{g^+}^- \right\}. \quad (15)$$

Finally, as shown in Fig. 4, the free space related to a lane change from lane l and the related gap index g to lane $l+1$ and the related gap index g^+ is

$$\mathcal{F}_{g, g^+}^{\text{lc}} = \left\{ (t, s, n) \in \mathcal{S}_{g, g^+}^{\text{lc}} \times \mathbb{R} \mid n \in \text{conv}(\mathcal{N}_{l_{\text{gap}}(g)} \cup \mathcal{N}_{l_{\text{gap}}(g^+)}) \right\}. \quad (16)$$

For a lane change, both lanes are required to be free of SVs and for the next lane $l+1$, also rear vehicles need to be considered for the duration of the lane change, cf. Ass. III.2. Only the closest rear vehicle on the next lane needs to be considered, since more distant vehicles are constrained by preceding ones, cf. Fig. 4.

The convexity of (14), (15) and (16) stems from the fact that each set is an intersection of hyperplanes, which implies

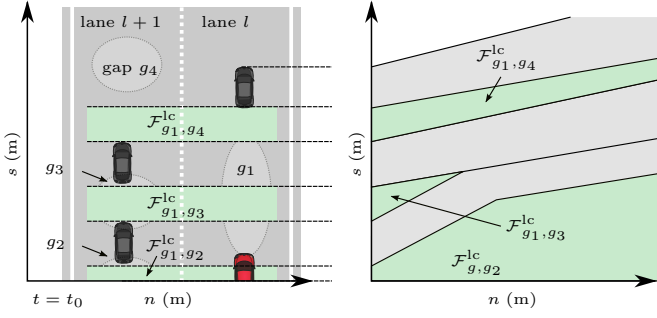


Fig. 4. Sketch of obstacle-free sets \mathcal{F}^{lc} (green) for lane changing related to three SVs on the two lanes l and $l+1$. The left plot shows the curvilinear space with coordinates s and n . The right plot shows the ST-space. Three possible gaps with indices g_2 , g_3 , and g_4 on the consecutive lane $l+1$ are available for a transition from gap index g_1 and lane l .

convexity [38]. In case of a detected lane change of an obstacle, which we assume to be detectable (Ass. III.1), both lanes are considered to be blocked for the whole prediction horizon, cf. Ass. III.1.

IV. SHORT-HORIZON APPROXIMATIONS

The short-term motion planning formulation (STF) approximates the vehicle dynamics for a prediction horizon t_f and a maximum of one lane change towards the goal lane \tilde{l} , similar to [3], [4]. The selection of the particular gap index g^+ on the next lane is part of the long-term motion planning formulation (LTF), which is vice versa constrained by the trajectory of the STF in the final LSTMP formulation.

Discretizing the model (11) with a discretization time t_d yields the linear discrete-time model

$$x_{k+1} = Ax_k + Bu_k. \quad (17)$$

A prediction horizon $t_f = Nt_d$ with N steps is used to approximate the infinite horizon in (10).

We define the acceleration bounds on the Frenet coordinate frame accelerations by the admissible control set

$$\mathcal{U} = \{u \in \mathbb{R}^{n_u} \mid \underline{u} \leq u_k \leq \bar{u}\}, \quad (18)$$

where $\underline{u} = [\underline{a}_{lon}, -\bar{a}_{lat}]^\top$ and $\bar{u} = [\bar{a}_{lon}, \bar{a}_{lat}]^\top$. Nonetheless, higher curvatures and its derivatives can be inner-approximated by convex sets according to [43]. Moreover, the constraint

$$\alpha_l v_s \leq v_n \leq \alpha_r v_s \quad (19)$$

limits the lateral velocity in order to approximate the nonholonomic motion of a kinematic vehicle model.

The reference tracking cost (7) is approximated for the STF, whereas the remaining costs of the objective (9) are approximated as part of the LTF. Binary variables $\lambda_k \in \mathbb{N}_{[0:1]}$, with $\Lambda = [\lambda_0, \dots, \lambda_N]$, are used to indicate whether the planned position is on the current lane, $\lambda_k = 0$, or on the next lane, $\lambda_k = 1$. The lane indices are always updated w.r.t. the current state such that $\lambda_k = 0$ corresponds to the current lane. A lateral reference can therefore be expressed by

$$\tilde{n}_k = d_{lane} \lambda_k, \quad k \in \mathbb{N}_{[0:N]}, \quad (20a)$$

$$\lambda_{k+1} \geq \lambda_k, \quad k \in \mathbb{N}_{[0:N-1]}. \quad (20b)$$

Constraint (20b) is used to cut off binary assignments to ease the solution of the MIQP problem. The constraint

$$\tilde{n}_k - \frac{d_{lane}}{2} \leq n_k \leq \tilde{n}_k + \frac{d_{lane}}{2}$$

is added to guarantee that from $\tilde{n}_k > 0$ the planned ego vehicle state is located on the next lane. Cost (7) can consequently be approximated with Frenet states as

$$g_{ref}^{st}(x_k, u_k, \lambda_k) = w_n (d_{lane} \lambda_k - n_k)^2 + w_v (\tilde{v} - v_{s,k})^2 + u_k^\top R u_k. \quad (21)$$

Cost (21) includes the term $(d_{lane} \lambda_k - n_k)^2 = (d_{lane} \lambda_k)^2 - 2d_{lane} \lambda_k n_k + n_k^2$ with the bilinear term $-2d_{lane} \lambda_k n_k$ which cannot directly be handled by MIQP solvers [26]. Thus, this bilinear term is reformulated by introducing continuous auxiliary variables $q_{bin,k} \in \mathbb{R}^+$, the additional constraint $q_{bin,k} = \lambda_k n_k$ and further related constraints according to Property II.1.

Finally, safety constraints approximating the set $\mathcal{X}_{free}(x(t), t)$ for the current and the next lane are formulated by considering M vehicles on the current lane, which is always set to $l = 1$ and the next lane $l^+ = 2$ and a chosen gap index $g^+ \in \mathbb{N}$ on the next lane, with $2 = l_{gap}(g^+)$.

Changing lane at time τ_1 follows three stages (cf. also [4]) where, in each stage, the constraints can be formulated as convex sets, cf., Fig. 5. First, at time $t \leq \tau_1 - \frac{1}{2} \bar{t}_{lc}$, the ego lane is tracked, second the lane is changed until the time limit $\tau_1 + \frac{1}{2} \bar{t}_{lc}$, and thirdly constraints for driving on the next lane hold for $t \geq \tau_1 + \frac{1}{2} \bar{t}_{lc}$. The lane change time on either lane is approximated by the upper-bound related to the time indices, $n_{lc} = \lceil \frac{\bar{t}_{lc}}{2t_d} \rceil$. Consequently, the lane change phases can be formulated in terms of index shifts of λ_k , with $[(1 - \lambda_{k+n_{lc}}) = 1]$ indicating the first stage, $[(\lambda_{k+n_{lc}} - \lambda_{k-n_{lc}}) = 1]$ indicating the transition phase and $[\lambda_{k-n_{lc}} = 1]$ indicating the last stage on the next lane, cf. Fig. 5. For out-of-range indices, i.e., $k < 0$ or $k > N$, the first value λ_0 or the last value λ_N are padded. For each position and time tuples (t_k, s_k, n_k) with $k \in \mathbb{N}_{[0:N]}$ and the current gap index $g = 1$, we require

$$[1 - \lambda_{k+n_{lc}} = 1] \implies (t_k, s_k, n_k) \in \mathcal{F}_1^+, \quad (22a)$$

$$[\lambda_{k+n_{lc}} - \lambda_{k-n_{lc}} = 1] \implies (t_k, s_k, n_k) \in \mathcal{F}_{1,g^+}^{lc}, \quad (22b)$$

$$[\lambda_{k-n_{lc}} = 1] \implies (t_k, s_k, n_k) \in \mathcal{F}_{g^+}^+, \quad (22c)$$

where the implications are reformulated according to Property II.2.

Recursive feasibility requires disjunctive terminal velocity constraints depending on the final lane, which is either the current lane implying $[\lambda_N = 0]$ or the next lane, implying $[\lambda_N = 1]$. Therefore, the terminal set is expressed by

$$[\lambda_N = 1] \implies v_{s,N} \leq \underline{s}_{g^+}^{sv} + t_d N \underline{v}_{g^+}^{sv}, \quad (23a)$$

$$[1 - \lambda_N = 1] \implies v_{s,N} \leq \underline{s}_1^{sv} + t_d N \underline{v}_1^{sv}, \quad (23b)$$

$$v_{n,N} = 0. \quad (23c)$$

Note that this terminal set is restrictive since it upper-bounds the final velocity with the velocity of the preceding vehicle on the respective lane. An increased terminal safe set

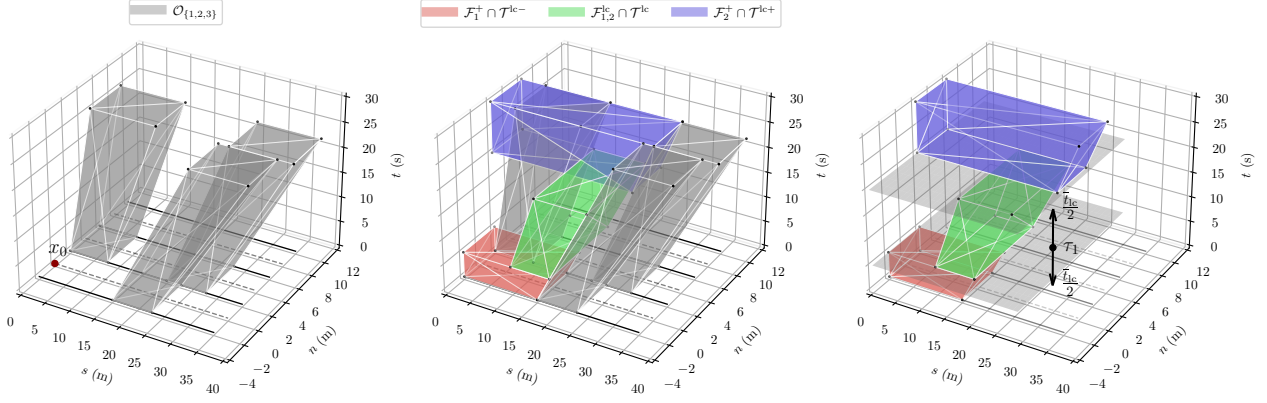


Fig. 5. Visualization of SVs sets $\mathcal{O}_{\{1,2,3\}}$ in the SLT-space and consecutive convex free regions between gap index 1 and gap index 2. The time subspaces $\mathcal{T}^{lc-} = \{(t, s, n) | t \leq \tau_1 - \tilde{t}_{lc}/2\}$, $\mathcal{T}^{lc} = \{(t, s, n) | \tau_1 - \tilde{t}_{lc}/2 \leq t \leq \tau_1 + \tilde{t}_{lc}/2\}$ and $\mathcal{T}^{lc+} = \{(t, s, n) | t \geq \tau_1 + \tilde{t}_{lc}/2\}$ define the consecutive time-related spaces on the planning horizon. The set $\mathcal{F}_1^+ \cap \mathcal{T}^{lc-}$ is the obstacle-free space on the first lane before the lane change, $\mathcal{F}_{1,2}^{lc} \cap \mathcal{T}^{lc}$ is the free space during the lane change and $\mathcal{F}_2^+ \cap \mathcal{T}^{lc+}$ is the obstacle-free space on the next lane after the lane change. Rear vehicles in the same lane are ignored, i.e., a vehicle is always allowed to brake. The binary variables λ_k determine, which set constraints are active for each x_k .

could be formulated by piece-wise linear approximations of deceleration constraints which requires further binary variables, cf. [4].

So far, constraints and costs have been introduced that are used as part of the STF to plan a collision-free discrete-time trajectory from the current lane to a certain gap at the next lane. Noteworthy, this trajectory is constrained such that it is always safe w.r.t. the obstacle constraints. The selection of the possible gap indices and also all further gaps towards the goal lane are formulated in the LTF and explained in the next section. The STF and the LTF are formulated in the final MIQP with mutual constraints, such that the rather approximate LTF cannot plan transitions that are infeasible w.r.t. the STF.

V. LONG-HORIZON APPROXIMATIONS

Within the LTF, costs and constraints are formulated that select collision-free transition gaps between two adjacent lanes. For long horizons, a fixed discretization in time is prohibitive, since the number of variables increases with the horizon length and would make the optimization problem hard to solve [3]. To circumvent the computational scaling with the prediction time, we propose a formulation in the two-dimensional continuous ST-space, where we exclusively model the transitions as points in time and longitudinal position for each lane change, with the transition times $T = [\tau_1, \dots, \tau_{L-1}]^T$ and longitudinal transition positions $\Sigma = [\sigma_1, \dots, \sigma_{L-1}]^T$.

In the following, three synergetic concepts are formulated to approximate the transitions, i.e., constraints that approximate reachability, a formulation for guaranteeing and maximizing the distance to SVs and a disjunctive formulation for choosing among gaps between vehicles for each lane. Binary variables are used to indicate whether transitions are invalid, resulting in the tuples of valid transitions for $\bar{L} \leq L$ lanes.

A. Approximate reachability

Reachability between transitions is approximated by the set $\mathcal{R}(\tau_l, \sigma_l)$ using constraints defined by operating velocity

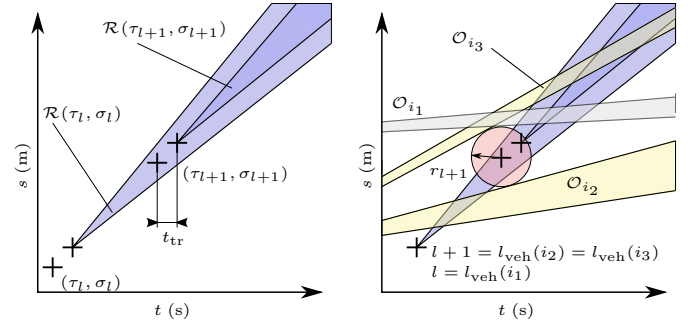


Fig. 6. The left plot shows the reachable sets after the transition to lane l and after the transition to lane $l+1$. The reachable set has an offset related to the estimated traversal time \tilde{t}_{lc} . The right plot shows the Chebychev center (CC) of the transition to lane $l+1$ from gap index i_1 to gap index i_3 with two SVs on the next lane $l+1$ (yellow) and one SV on the current lane l (grey). The time axis is scaled by the reference velocity \bar{v} , which is here assumed to be 1.

bounds \underline{v}_{op} and \bar{v}_{op} and an approximation \tilde{t}_{lc} of the time required to traverse a lane. The operating velocity bounds are artificially added to approximate the *true* reachable set around the expected velocity range of the vehicle.

The approximated set is used to define constraints for the next transition $(\tau_{l+1}, \sigma_{l+1})$, cf. Fig. 6. Each reachable set depends on the last transition (σ_l, τ_l) by the shifted cone

$$\mathcal{R}(\tau_l, \sigma_l) = \left\{ (\tau_{l+1}, \sigma_{l+1}) \left| \begin{array}{l} \sigma_{l+1} \leq \sigma_l + \bar{v}_{op}(\tau_{l+1} - \tau_l - \tilde{t}_{lc}) \\ \sigma_{l+1} \geq \sigma_l + \underline{v}_{op}(\tau_{l+1} - \tau_l + \tilde{t}_{lc}) \end{array} \right. \right\}. \quad (24)$$

The convex reachable set (24) is an approximation using the velocity bounds \bar{v}_{op} and \underline{v}_{op} . Using bounds on the acceleration would result in nonconvex quadratic constraints which could be still approximated by the problem specific parameters \tilde{t}_{lc} , \bar{v}_{op} and \underline{v}_{op} .

B. Chebychev centering for transitions

Next, criteria for determining the locations of continuous transition points are defined. Transitions require an obstacle-

free area for a minimum of the duration of the lane-change \bar{t}_{lc} , as defined for the STF. Beyond the minimum required time, an approach based on the Chebychev center (CC) is proposed that centers the transition in the ST-space related to obstacle constraints, i.e., the transition should be planned at a maximum weighted distance to constraints.

The CC formulation of (5) is used as a basis for further considerations. Centering constrains for the polytope defined by $[\tau, \sigma] \in \mathcal{S}_g^+$ are written as $h_g^+(\tau, \sigma, r) \leq 0$ according to (5), which includes the centering radius r . For the polytope defined by $[\tau, \sigma] \in \mathcal{S}_g^- \cap \mathcal{S}_g^+$, the centering constraints are denoted by $h_g(\tau, \sigma, r) \leq 0$.

Notice, that the ST-space has different units, namely longitudinal distance and time. To achieve a meaningful distance measure, the time coordinate is scaled by the reference velocity \bar{v} . Therefore the unit of the radius is Meters.

The constraint (5c) is tightened to $r \geq \underline{r}$ to guarantee the minimum distance to obstacle constraints in the ST-space, i.e., the centering is only feasible for a centering radius higher than a threshold \underline{r} . Using the sequence of gap indices $[g_1, \dots, g_{L-1}]$, with $l = l_{\text{gap}}(g_l)$, for each lane transition and the accumulated cost

$$G_{\text{safe}}(R) = -w_{\text{safe}} \sum_{l=1}^{L-1} r_l,$$

with transition radii $R = [r_1, \dots, r_{L-1}]^\top$ and a weight w_{safe} to promote a further safety distance beyond the hard constraints, all transitions can be formulated in the shared linear program

$$\min_{R, \Sigma, T} G_{\text{safe}}(R) \quad (25a)$$

$$\text{s.t.} \quad h_{g_l}^+(\tau_l, \sigma_l, r_l) \leq 0, \quad l \in \mathbb{N}_{[1:L-2]}, \quad (25b)$$

$$h_{g_l}(\tau_l, \sigma_l, r_l) \leq 0, \quad l \in \mathbb{N}_{[2:L-1]}, \quad (25c)$$

$$\underline{r} \leq r_l, \quad l \in \mathbb{N}_{[1:L-1]}. \quad (25d)$$

Fig. 6 shows the centering of a transition $(\tau_{l+1}, \sigma_{l+1}, r_{l+1})$ from lane l to lane $l+1$ in the presence of a leading SV i_1 on lane l with $l = l_{\text{veh}}(i_1)$ and two SVs i_2, i_3 on the next lane $l+1$ with $l+1 = l_{\text{veh}}(i_2) = l_{\text{veh}}(i_3)$. Besides the SVs constraints, the center of the transition $(\tau_{l+1}, \sigma_{l+1})$ is constrained by the approximated reachable set.

The linear program (25a) is not directly solved but its cost and constraints are included in the final LSTMP MIQP. Notice that, therefore, the centering is solved as a weighted trade-off to other constraints such as the duration of the lane change. The sequence of gap indices is determined by the disjunctive formulation including the constraints within a "big-M" formulation, cf. Property II.4, and explained in the next Section V-C. Notice that computing the transitions purely by maximizing the distance to SVs, without including a measure along the time axis, would ignore the safety distance related to the relative velocity of vehicles.

C. Disjunctions among gaps

A fundamental combinatorial aspect of lane change planning is the choice of gap indices on each lane. In Sect. V-A,

it was shown how to constrain transitions to an approximate reachable set and in Sect. V-B, a formulation to center a transition in the ST-space was introduced, given a sequence of gap indices. As an essential final component of the LTF, a disjunctive formulation of choosing a single gap on each lane is proposed according to Property II.4. To activate constraints related to a certain gap, on each transition (τ_l, σ_l) binary variables $\beta_i \in \mathbb{N}_{[0:1]}$ are used and summarized in the vector $B \in (\mathbb{N}_{[0:1]})^{(L-1)(M+2)}$. The activation of gaps starts on the second lane since the current lane gap is trivially fixed. In each lane, one additional binary variable is added to account for the option of *no transition* or *lane-keeping*. Therefore, this particular binary variable with index \hat{g} implies the variables (τ_l, σ_l) to be unconstrained by defining $h_{\hat{g}}(\tau_l, \sigma_l, r_l) \leq M$, where M is a large number. For the following definitions, we define the set $\mathcal{G}_l := \{g \mid l = l_{\text{gap}}(g)\}$, i.e., the set of all gap indices on a particular lane, including the additional virtual unconstrained one and the set $\hat{\mathcal{G}} \in \mathbb{N}_{[1:L-1]}$ that contains all indices of unconstrained added gaps. The disjunctions

$$\bigvee_{g^+ \in \mathcal{G}_{l+1}} \left[h_{g^+}(\tau_l, \sigma_l, r_l) \leq 0 \right], \quad \forall l \in \mathbb{N}_{[1:L-1]}, \quad (26)$$

constrain the transition (τ_l, σ_l) onto lane $l+1$ by the leading and following vehicles of a selected gap index g^+ , where $\beta_{g^+} = 1$ and the disjunctions

$$\bigvee_{g \in \mathcal{G}_l} \left[h_g^+(\tau_l, \sigma_l, r_l) \leq 0 \right], \quad \forall l \in \mathbb{N}_{[2:L-1]}, \quad (27)$$

constrains the transition (τ_l, σ_l) from lane l to the next lane only by the leading vehicles. In the case of *no transition*, i.e., the additional binary variables $\beta_{\hat{g}}$ is activated, where \hat{g} the index of the virtual added unconstrained gap, a high cost $G_{\text{lane}}(\Sigma, T, B)$ is added that approximates (8) for not changing the lane.

Note that the binary variables B are related to the gaps on each lane, starting with the second lane $l = 2$. The transitions (τ_l, σ_l) are related to two adjacent lanes l and $l+1$, starting with the transition from the first lane $l = 1$. This distinction is crucial to the disjunctive constraints for each transition. For a transition l related to the departing lane l , only leading vehicle constraints $h_g^+(\tau_l, \sigma_l, r_l) \leq 0$ related to gap index g are considered according to Ass. III.2. For the next lane gap index g^+ both, the preceding constraints and the leading vehicle constraints in $h_{g^+}(\tau_l, \sigma_l, r_l) \leq 0$ are used. This formulation models interactive behavior by allowing to slow down SVs on the current lane to reach a certain gap on the next lane.

The following further constraint on the binary variables

$$\sum_{g \in \mathcal{G}_l} \beta_g = 1, \quad \forall l \in \mathbb{N}_{[2:L]}, \quad (28)$$

reduces the search space for the mixed-integer (MI) solver. For each pair of consecutive virtual gaps \hat{g} and \hat{g}^+ , with $l_{\text{gap}}(\hat{g}^+) = l_{\text{gap}}(\hat{g}) + 1$, the physical constraint

$$\beta_{\hat{g}^+} \geq \beta_{\hat{g}}, \quad (29)$$

sets all further lane-changes $l > l_1$ to *lane-keeping*, if a lane is blocked.

D. Cost approximations

In the following, a lane changing cost $G_{\text{lane}}(\Sigma, T, B)$ that approximates (8) and a reference velocity cost $g_{\text{ref}}^{\text{lh}}(\Sigma, T)$ that approximates (7) and penalizes transitions with deviations of the reference velocity \tilde{v} are formulated.

Cost (8) linearly penalizes the number of lanes distant from the goal lane and is integrated over time in the final objective. This integral can be approximated over a horizon t_f by the following sum

$$\int_{t=t_0}^{t_f} g_{\text{lane}}(x(t); \Theta) dt \approx G_{\text{lane}}(\Sigma, T, B) := w_g \sum_{\hat{g} \in \hat{\mathcal{G}}} \tau_l (1 - \beta_{\hat{g}}) + t_f \beta_{\hat{g}}, \quad (30)$$

which penalizes the duration τ_l on each lane l , if there was a valid transition, i.e., $\beta_{\hat{g}} = 0$. If *no transition* was computed for lane l , i.e., $\beta_{\hat{g}} = 1$, the cost for the full horizon staying on the lane is summed in (30). Note that (30) contains bilinear terms of binary variables $\beta_{\hat{g}}$ and σ_l , both decision variables. The terms are treated by introducing an additional variable $q_{\text{bi},l}$ for each bilinear term, cf., Property II.1. All auxiliary variables related to bilinear terms are summarized by $Q_{\text{bi}} = [q_{\text{bi},1}, \dots, q_{\text{bi},L}, q_{\text{bin},1}, \dots, q_{\text{bin},N-1}]$

Finally, the difference of the reference speed according to (7) is penalized for two consecutive transitions by

$$G_{\text{ref}}(T, \Sigma) = \frac{w_v t_f}{l_g} \left(\sum_{l=2}^{l_g} ((\sigma_l - \sigma_{l-1}) + (\tau_l - \tau_{l-1}) \tilde{v})^2 \right). \quad (31)$$

Cost (31) approximates the duration between lane transitions with the constant value $\frac{t_f}{l_g}$, starting from the second transition as the reference cost approximation of (7) for the first transition is included in the STF cost (21).

Notably, this cost approximation for the reference velocity neglects the planned time driving on each lane.

VI. LONG-SHORT-HORIZON MOTION PLANNER

In the following, we complete the final motion planning MIQP problem with necessary additional formulations for combining the STF of Sect. IV and the LTF of Sect. V.

First, the formulations of the STF and the LTF are combined *consistently* according to the following definition for the first transition (τ_1, σ_1) .

Definition VI.1. A transition (τ, σ) is consistent with the longitudinal states s_k and the lateral states n_k of a trajectory, with $k \in \mathbb{N}_{[0:N]}$, if and only if the following inequalities hold

$$n_k \leq \frac{d_{\text{lane}}}{2}, \forall k \in \{i \in \mathbb{N}_{[0:N]} \mid it_d \leq \tau\}, \quad (32a)$$

$$n_k > \frac{d_{\text{lane}}}{2}, \forall k \in \{i \in \mathbb{N}_{[0:N]} \mid it_d > \tau\}, \quad (32b)$$

$$s_k \leq \sigma, \forall k \in \{i \in \mathbb{N}_{[0:N]} \mid it_d \leq \tau\}, \quad (32c)$$

$$s_k > \sigma, \forall k \in \{i \in \mathbb{N}_{[0:N]} \mid it_d > \tau\}. \quad (32d)$$

Def. VI.1 states that the position states of the STF trajectory must be located on the current lane, closer than the longitudinal position σ and before the transition time τ and on the consecutive lane and position, thereafter.

In the following, the consistency formulation for the first transition (τ_1, σ_1) of a feasible solution of the LSTMP is shown. Therefore, constraints among the discrete decision variables λ_k , the transition (τ_1, σ_1) and positions s_k of the STF are defined by the pair-wise *exclusive* disjunctions according to Property II.4,

$$\begin{bmatrix} [\lambda_k = 1] \\ kt_d \geq \tau_1 \\ s_k \geq \sigma_1 \end{bmatrix} \vee \begin{bmatrix} [\lambda_k = 0] \\ kt_d < \tau_1 \\ s_k < \sigma_k \end{bmatrix}, \quad \forall k \in \mathbb{N}_{[1:N]}, \quad (33)$$

For each pair k , the disjunctions (33) use the same binary variable λ_k , yet, with the opposite indication, i.e., either $[\lambda_k = 0]$ or $[\lambda_k = 1]$, which makes it exclusively choosing the related constraints.

Moreover, a terminal set formulation for the STF is required to reach a transition (τ_1, σ_1) with $\tau_1 \geq Nt_d$, i.e., the transition time τ_1 is further distant than the final STF prediction time Nt_d . It holds that $\tau_1 \geq Nt_d \Leftrightarrow [\lambda_N = 0]$, so the reachable set can be conditioned on λ_N by

$$[\lambda_N = 0] \implies (\tau_1, \sigma_1) \in \mathcal{R}(Nt_d, s_N). \quad (34)$$

The final LSTMP, formulated as an MIQP, can be stated by decisions variables, costs, and constraints of the STF, the LTF, and with the additional coupling constraints (33) and (34).

The STF decision variables are $X_s = (X, U, \Lambda) \in V_s$, where

$$V_s = \mathbb{R}^{N \times n_x} \times \mathbb{R}^{N-1 \times n_u} \times (\mathbb{N}_{[0:1]})^N,$$

and the LTF decision variables are $X_l := (\Sigma, T, R, Q_{\text{bi}}, B) \in V_l$, where

$$V_l := \mathbb{R}^{L-1} \times \mathbb{R}^{L-1} \times \mathbb{R}^{L-1} \times \mathbb{R}^{L-1+N} \times (\mathbb{N}_{[0:1]})^{L-1 \times M+2}.$$

In total, $3(L-1)$ continuous and $(L-1)(M+2)$ binary variables are used to model the transitions for LM SVs. Another $(N-1)(n_x + n_u) + n_x$ continuous and N binary variables model the first lane change for a horizon of $t_d N$. A total of $L+N-1$ variables are used as auxiliary variables.

Remarkably, the total number of binary variables is $N_{\text{bin}} = (L-1)(M+2) + N$, which is with $O(LM+N)$ usually a much lower number in contrast to $O(LMN)$ of [3].

The cost function (9) is approximated by the cost of the STF trajectory

$$\hat{J}_s(X_s) = \sum_{k=0}^N g_{\text{ref}}^{\text{st}}(x_k, u_k, \lambda_k),$$

and the cost of the long horizon is

$$\hat{J}_l(X_l) = G_{\text{lane}}(\Sigma, T, B) + G_{\text{ref}}(T, \Sigma) + G_{\text{safe}}(R).$$

The relations of the general OCP objective in (10) approximated by the LSTMP, comprising the LTF cost $\hat{J}_l(\cdot)$ and the STF cost $\hat{J}_s(\cdot)$ are shown in Fig. 7.

Including a constraint $x_0 = \hat{x}$ that constrains the decision variable x_0 to the current state \hat{x} , the constraints of the STF are summarized by $g_s(X_s) \leq 0$ and include the discrete dynamic

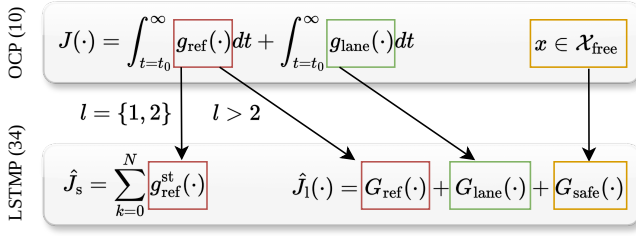


Fig. 7. Overview of the approximations of the general OCP objective function (10) by the LTF cost $\hat{J}_1(\cdot)$ and the STF cost $\hat{J}_s(\cdot)$. The reference cost $g_{\text{ref}}(\cdot)$ is approximated for the first two lanes within the STF and, thereafter, by the LTF.

model (17), the control and state constraints (18) and the constraints related to the first lane-change (20), (21), (22) and (23).

For the LTF, a constraint $g_l(X_1) \leq 0$ summarizes the reachability constraints (24), the CC constraints (25a), and the constraints used to formulate the disjunction among gaps in (26), (27), (28), and (29).

The coupling constraints (33) and (34) between states of the LTF and STF are concisely written as $g_c(X_s, X_1) \leq 0$.

Ultimately, the LSTMP approximates the solution of the OCP (10) by solving the following MIQP in each iteration

$$\min_{\substack{X_s \in V_s, \\ X_1 \in V_1}} \hat{J}_s(X_s) + \hat{J}_1(X_1) \quad (35a)$$

$$\text{s.t.} \quad g_s(X_s) \leq 0, \quad g_l(X_1) \leq 0, \quad g_c(X_s, X_1) \leq 0. \quad (35b)$$

The output $X^* = (X_s, X_1)$ of the LSTMP is always safe w.r.t. the obstacle constraints (14) and (16). This follows directly from the constraint formulations of the STF, including the terminal safe set (23). Approximation errors in the LTF formulations may lead to sub-optimal behavior. However, they do not influence safety related to the feasibility of the trajectory X_s^* .

VII. EVALUATION

We evaluate the proposed LSTMP approach in two different setups. First, deterministic SVs are simulated as they are modeled in the LSTMP and exact tracking of the provided plan is assumed. A second setup includes more realistic scenarios, where the traffic is simulated interactively by the traffic simulator SUMO [53], based on benchmark scenarios provided by the CommonRoad-framework [5], cf., Fig. 8. Moreover, the LSTMP is integrated into an autonomous driving (AD)-stack with a low-level NMPC tracking controller of [42] that tracks the LSTMP trajectory X^* by controlling a simulated single-track *BMW 320i* medium-sized passenger car model provided by CommonRoad. The SV states \hat{X}^{SV} and the current estimated point-mass state \hat{x} are the inputs of the planner. The point-mass state \hat{x} is obtained from the six-dimensional simulated single-track vehicle state \hat{z} .

For both setups, the LSTMP is compared against the MIP-DM of [3] and a hybrid A* formulation according to [20]. Rendered simulations can be found on the website https://rudolfreiter.github.io/lstmp_vis/

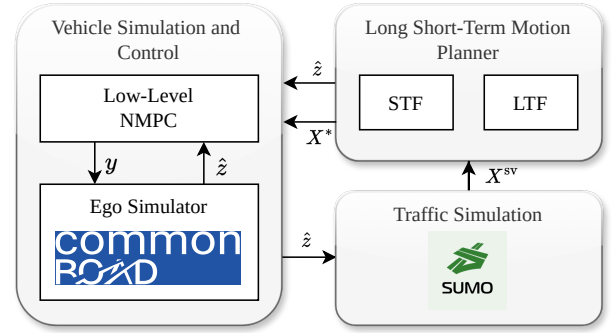


Fig. 8. Overview of the adopted simulation architecture. After obtaining the ego vehicle state \hat{z} and the SVs states X^{SV} , the LSTMP solves in each planning iteration the MIQP (35). The computed plan X^* related to the point-mass model is forwarded to the low-level NMPC for tracking. For the ego vehicle simulation, a *BMW 320i* vehicle model provided by CommonRoad is used. The position of the ego vehicle \hat{z} is passed to the SUMO traffic simulator.

TABLE I
PARAMETER FOR EVALUATIONS.

General parameters	
t_d, M	300ms, 7
$w_n, w_v, w_g, w_{\text{safe}}$	$10^{-2}, 10^{-1}, 200, 10^{-5}$
R	$\text{diag}([5 \cdot 10^{-4}, 2 \cdot 10^{-3}])$
d_{lane}	3.75m (Germany), 12 feet (US)
$[a_{\text{lon}}, a_{\text{lat}}]$	$[-8, -3] \frac{\text{m}}{\text{s}^2}$
$[\bar{a}_{\text{lon}}, \bar{a}_{\text{lat}}]$	$[5, 3] \frac{\text{m}}{\text{s}^2}$
LSTMP - deterministic scenario	
t_d, N, M	300ms, 15, 7
t_f, \bar{t}_{lc}	$10^5 \text{s}, 2.7 \text{s}$
$\bar{v}^{\text{sv}}, \underline{v}^{\text{sv}}$	$\hat{v}^{\text{sv}}, \hat{v}^{\text{sv}}$
LSTMP-V0 - interactive scenario	
$\bar{v}^{\text{sv}}, \underline{v}^{\text{sv}}, L$	$\hat{v}^{\text{sv}} + 1 \frac{\text{m}}{\text{s}}, \hat{v}^{\text{sv}} - 1 \frac{\text{m}}{\text{s}}, 5$
LSTMP-V1 - interactive scenario	
$\bar{v}^{\text{sv}}, \underline{v}^{\text{sv}}, L$	$\hat{v}^{\text{sv}} + 1 \frac{\text{m}}{\text{s}}, \hat{v}^{\text{sv}} - 1 \frac{\text{m}}{\text{s}}, 3$
LSTMP-V2 - interactive scenario	
$\bar{v}^{\text{sv}}, \underline{v}^{\text{sv}}, L$	$\hat{v}^{\text{sv}} + 3 \frac{\text{m}}{\text{s}}, \hat{v}^{\text{sv}} - 3 \frac{\text{m}}{\text{s}}, 5$

A. Implementation details

We describe the setup used for evaluation in the following. Parameters are chosen according to Tab. I.

1) *Preprocessing*: The SVs states X^{SV} are processed before either planner is executed. First, SVs that drive closer to each other than a longitudinal threshold distance of 15m are merged by setting the corresponding upper and lower velocity bounds and increasing the occupied space. Second, a maximum number of $M = 7$ SVs per lane are considered, which are the 7 closest vehicles at the current time step on each lane.

2) *Benchmark MIP-DM*: The first benchmark is based on the MIP-DM formulation of [3]. It uses a fixed discrete-time trajectory, similar to the STF, however, with four binary variables per obstacle and per time step to account for the rectangular obstacle shape. One further binary variable per time step indicates a lane change. The number of binary

variables for the MIP-DM is therefore $N_{\text{bin}} = 4NML + N$. The MIP-DM is adapted to be comparable to the LSTMP. First, only one lane change direction is allowed, which reduces the number of binary variables. Second, obstacle shapes are inflated to occupy the whole lane, equally to the LSTMP. Finally, the interactive braking behavior of succeeding SVs on the same lane is implemented by deactivating corresponding obstacles on the current lane at the current time step. For the MIP-DM a total number of $M = 3$ vehicles are considered on $L = 5$ consecutive lanes, while the horizon length N is 10, 15 or 20 steps.

3) *Benchmark hybrid A**: The second benchmark is based on the hybrid A* of [20]. This planner considers lateral motion only at the discrete lane indices, with a search space of (t, s, l) . In order to be comparable to the other planners, we modify the search space to (s, l, v_s) , which includes the velocity v_s instead of time. Since the hybrid A* of [20] does not consider lateral states between lane centers, we use a sampling time of $7t_d$ to allow full lane changes in one expansion, i.e., it is guaranteed that the final planning vertex is always located on the center of a lane. We use the same planner model (17) for vertex expansions. Note that hybrid A* could use nonlinear models without increasing the computation time, which, in contrast, would be challenging for the LSTMP and MIP-DM. As an admissible heuristic, the relaxed solution of (35) without obstacle constraints is computed for each lane. The longitudinal acceleration control is discretized into 11 intervals and the lateral acceleration is computed by using 11 lane change primitives. The lateral states correspond to the number of lanes, the longitudinal position is discretized with 100 intervals, and the velocity with 20 intervals. The number of node expansions is varied in experiments between 5 and 500.

4) *Low-level NMPC*: The low-level NMPC is formulated as shown in [42], using a nonlinear single-track vehicle model, a sampling time of 10ms and a horizon of 1.5s. The controls comprise the acceleration a and the steering rate δ .

5) *Scenarios*: For deterministic comparisons in Sect. VII-B and interactive closed-loop comparisons in Sect. VII-C, the scenarios are chosen according to Tab. II. Due to traffic congestion, the velocity can be zero. Traffic flow and density are averaged over the simulation. The velocity range V^{sv} corresponds to the measured SVs velocities during all simulations. The scenarios are simulated for 40 seconds or until the end of the road is reached. Snapshots of the CommonRoad scenarios for interactive simulations are shown in Fig. 9.

6) *Computations and numerical solvers*: The MIQPs of the LSTMP and the MIP-DM are solved with `gurobi` [26]. The nonlinear program (NLP), arising in the low-level NMPC, is solved by the open-source solver `acados` [54]. Simulations are executed on a LENOVO ThinkPad L15 Gen 1 Laptop with an Intel(R) Core(TM) i7-10510U @ 1.80GHz CPU.

B. Evaluation for deterministic traffic and exact tracking

In order to compare the performance of the planner without interference from the traffic prediction error, simulation modeling error, and controller performance, the planner is simulated

TABLE II
DIFFERENT SCENARIO SETTINGS FOR SVs USED IN EVALUATIONS.

Scenario Name	tr.-flow	tr.-density	L	V^{sv}	\bar{v}
	$\frac{\text{SV}_s}{\text{lane}\cdot\text{min}}$	$\frac{\text{SV}_s}{\text{lane}\cdot\text{km}}$		$\frac{\text{m}}{\text{s}}$	$\frac{\text{m}}{\text{s}}$
Deterministic					
custom	14.6	12.2	9	[15, 35]	25
Closed-loop interactive					
USA_US101-22_1_I-1	11.8	14.0	6	[0, 22.2]	15
DEU_Col.-63_5_I-1	22.3	26.9	3	[11.0, 16.5]	11
DEU_Col.-63_5_I-1_s	8.9	10.5	3	[11.1, 16.5]	15

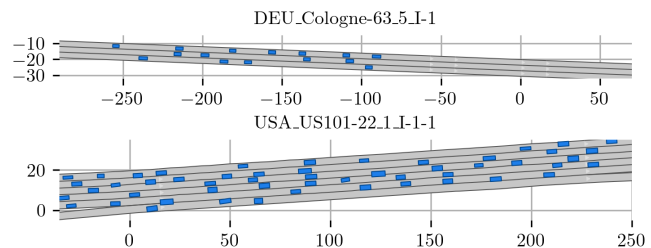


Fig. 9. Different tracks from the CommonRoad scenario database used for the closed-loop simulation.

with deterministic SVs and exact tracking. Deterministic SVs are simulated with constant speed, if they are at a minimum distance to a slower leading SVs and with the speed of the leading SV if they are below the threshold distance. The planned trajectory X^* is assumed to be tracked exactly. This setup resembles the model of the traffic used in the LSTMP, where tight bounds for the obstacle-free sets (14) and (16) can easily be found. In Fig. 10, snapshots of a randomized simulation with five lanes are shown, where the vehicle starts at the bottom lane and has to reach the top lane. Red areas indicate the SVs after pre-processing. The STF trajectory X^* of the LSTMP is shown in black, whereas the transition gaps are shown in blue. The evaluated closed-loop cost and computation time of 100 randomized *custom* scenarios according to Tab. II for the deterministic setup are shown in Fig. 11 on the Pareto front.

The comparisons include evaluations for different parameter settings of the algorithms, i.e., the number of considered consecutive lanes in the LSTMP, the maximum node expansions in hybrid A*, and the horizon length of the MIP-DM.

The MIP-DM with the longest horizon outperforms the LSTMP in the average closed-loop cost over the full simulations, however, at a high computational expense which violates the real-time requirement. In fact, the computation time of the MIP-DM is an order of magnitude higher than of the LSTMP.

The hybrid A* can be faster to execute compared to the LSTMP, but it yields a higher closed-loop cost. In our experiments, increasing the iterations of hybrid A* could not yield a better performance. This may be due to the longer duration of motion primitives to allow lane changes and the resulting

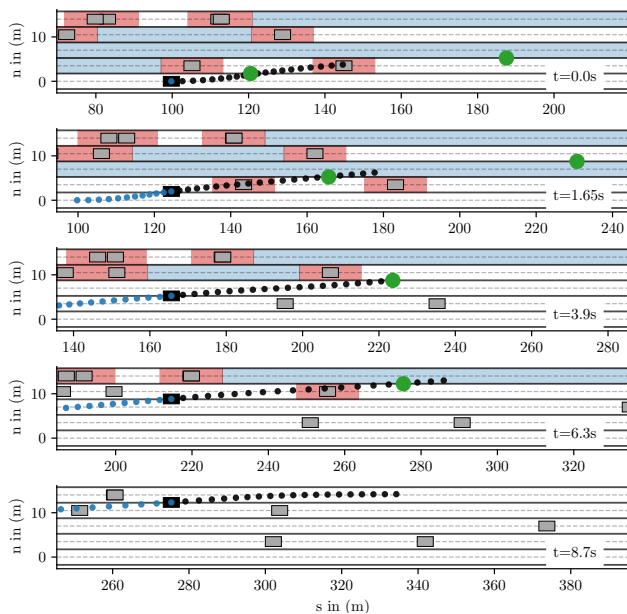


Fig. 10. Snapshots during lane-changes on a five-lane deterministic environment with randomized SV (gray) initial speeds and lanes. Blue regions indicate computed gaps of the LSTMP, with green points corresponding to the expected transition position σ_l and the black STF trajectory X^* . Red areas correspond to occupied sets $\mathcal{O}(t_{\text{sim}})$, where t_{sim} is the current simulation time of the snapshot.

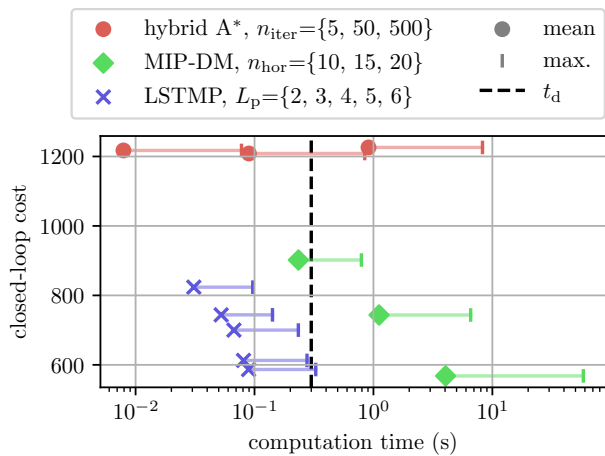


Fig. 11. Pareto comparison of planners in randomized deterministic traffic scenarios. The hybrid A* planner can be parameterized to have the fastest computation time and the MIP-DM achieves the lowest costs. However, the novel LSTMP formulation performs best when both, a low computation time as well as low costs are required.

coarser time discretization. Further relevant properties related to the lane change multi-objective (9) are shown in Tab. III. This includes the mean deviation from the reference speed $\Delta\bar{v}$, mean and maximum values for the lateral and longitudinal accelerations, and the maximum reached lane l_{max} at the end of the simulation. It shows that the LSTMP with a longer horizon better keeps the reference speed and also changes lanes more often. By utilizing large acceleration values, the MIP-DM achieves the highest number of lane changes and the overall lowest closed-loop cost, c.f., Fig. 11.

TABLE III
COMPARISON OF PLANNERS IN RANDOMIZED DETERMINISTIC TRAFFIC SCENARIOS FOR DIFFERENT MEAN (MAXIMUM) QUANTITIES.

Planner		$\Delta\bar{v}$	a_{lat}	a_{lon}	l_{max}	
Par.	Val.	$\frac{\text{m}}{\text{s}}$	$10^{-1} \frac{\text{m}}{\text{s}^2}$	$10^{-1} \frac{\text{m}}{\text{s}^2}$		
LSTMP						
L_p	2	22	1.75	0.71 (4.58)	1.40 (5.30)	3.92
	3	29	0.42	0.63 (3.82)	1.46 (6.35)	4.20
	4	36	0.03	0.66 (3.70)	1.48 (6.45)	4.34
	5	43	0.07	0.64 (3.62)	1.37 (4.96)	4.62
	6	50	0.03	0.67 (3.62)	1.55 (6.15)	4.66
MIP-DM						
N	10	610	2.50	0.48 (2.68)	0.69 (5.14)	3.62
	15	915	2.10	0.52 (2.67)	1.06 (8.16)	4.17
	20	1220	1.98	0.65 (2.67)	1.83 (8.70)	4.75
hybrid A*						
iter.	5	N/A	1.78	0.47 (2.40)	0.14 (1.00)	2.61
	50	N/A	1.73	0.47 (2.40)	0.17 (1.14)	2.63
	500	N/A	1.70	0.43 (2.40)	0.22 (1.43)	2.58

Notably, the number of binary variables required in the MIP-DM is much larger than in the LSTMP, which leads to a significantly longer computation time. For a prediction horizon of $N = 10$ and settings of Tab. I, the MIP-DM requires 610 binary variables and for a prediction horizon of $N = 20$ a total of 1220 binary variables. The LSTMP that considers in total $L_p = 2$ lanes, requires only 22 binary variables, whereas considering $L_p = 5$ lanes requires only 50 binary variables.

C. Evaluation for interactive traffic and closed-loop control

For different randomized scenarios according to Tab. II and Fig 9, the lane changing problem is simulated with interactive SVs, using a software architecture corresponding to Fig. 8. The ego vehicle starts at random free positions and has to reach the leftmost lane, according to cost (9), with parameters of Tab. I. The LSTMP, hybrid A*, and MIP-DM are compared with a low-level tracking controller in closed-loop simulations. States are assumed to be estimated exactly, however, the velocity range of SVs is unknown. Different settings of the planners are used according to Tab. I to create the statistical evaluation of performance measures as shown in Fig. 12.

The performance evaluations show rare collisions of all planners due to prediction errors. For the conservative LSTMP-V1 configuration, no collisions were recorded. Computation times are lowest for the LSTMP planner and well below the planning time threshold t_{plan} . The computation times for the hybrid A* are nearly constant since the planning nodes are expanded with a fixed number of iterations. Notably, the computations for hybrid A* were not performed on a runtime-optimized code. The velocity varies the most for the LSTMP, which promotes acceleration and deceleration to reach certain gaps. This can also be verified by the high number of lane transitions of the LSTMP. Particularly, in

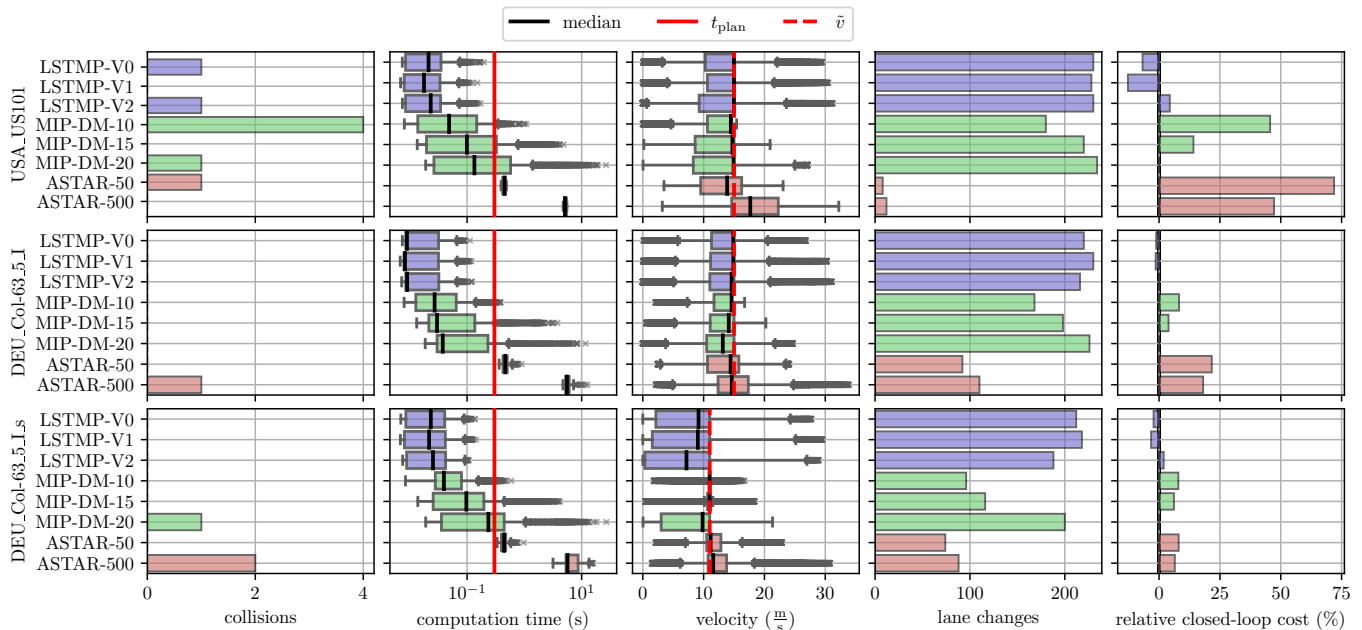


Fig. 12. Closed-loop evaluation of variants of the proposed LSTMP planner (blue) compared to hybrid A* (red) and MIP-DM (green). The total number of collisions, the computation time, velocities, the total number of lane changes, and the closed-loop cost compared to the most exact MIP-DM formulation are compared for different randomized scenarios. The proposed LSTMP has the lowest computation time below the planning time t_{plan} and a high number of lane changes for all scenarios. Moreover, it can reduce the closed-loop cost significantly. Occasional collisions are observed for all variants, irrespective of their parameters.

the DEU_Col.-63_5_I-1_s scenario long-term decisions significantly raised the number of lane transitions in the LSTMP. The closed-loop cost for LSTMP configurations are below the benchmark comparisons, particularly, below the MIP-DM-20 with the longest horizon of 20 steps, which we define as *expert*. Costs are relatively expressed to the cost of MIP-DM-20 and outperformed by LSTMP-V0 and LSTMP-V1.

VIII. CONCLUSION AND DISCUSSION

Under the variety of different planning methods for autonomous driving (AD), the proposed long-short-term motion planner (LSTMP) for lane change planning achieves a good trade-off between performance and computational costs, thanks to the use of state-of-the-art mixed-integer quadratic program (MIQP) solvers. The considered problem has relevant combinatorial and continuous parts, which makes MIQP formulations particularly suited to solve the proposed motion planning problem. Building on previous work to minimize the number of combinatorial variables, we introduced a novel long-horizon approximation. Together with a discrete-time trajectory, a single MIQP, which is computationally very efficient, was formulated consistently.

We compared our approach to the mixed-integer motion planning and decision maker (MIP-DM) [3], which uses more integer variables to model rectangular obstacle shapes that are not required to be aligned with the lane boundaries and to model lane transitions in both directions. This makes the MIP-DM a more versatile approach, i.e., lane changing is only a subset of problems that can be addressed with it.

The fundamental modeling approach of the LSTMP is the decomposition into convex cells together with a simplification due to the road alignment. The authors assume that is possible to add integer variables to achieve lane transitioning in both directions for a fixed maximum number of transitions and additional convex decompositions to resemble nonconvexities in the spatio-temporal (ST)-space, as for instance traffic lights.

In the future we will evaluate whether more flexible mixed-integer nonlinear programming can achieve better performance under real-time requirements.

REFERENCES

- [1] J. H. Reif, "Complexity of the mover's problem and generalizations," in *20th Annual Symposium on Foundations of Computer Science (sfcs 1979)*, 1979, pp. 421–427.
- [2] S. M. LaValle, *Planning Algorithms*. USA: Cambridge University Press, 2006.
- [3] R. Quirynen, S. Safaoui, and S. Di Cairano, "Real-time mixed-integer quadratic programming for vehicle decision making and motion planning," *ArXiv*, vol. abs/2308.10069, 2023.
- [4] C. Miller, C. Pek, and M. Althoff, "Efficient mixed-integer programming for longitudinal and lateral motion planning of autonomous vehicles," in *IEEE Intelligent Vehicles Symposium (IV)*, 2018, pp. 1954–1961.
- [5] M. Althoff, M. Koschi, and S. Manzing, "Commonroad: Composable benchmarks for motion planning on roads," in *IEEE Intelligent Vehicles Symposium (IV)*, 2017, pp. 719–726.
- [6] B. Paden, M. Cap, S. Z. Yong, D. Yershov, and E. Frazzoli, "A survey of motion planning and control techniques for self-driving urban vehicles," *IEEE Transactions on Intelligent Vehicles*, vol. 1, no. 1, pp. 33–55, 2016.
- [7] L. Claussmann, M. Revilloud, D. Gruyer, and S. Glaser, "A review of motion planning for highway autonomous driving," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 5, pp. 1826–1848, 2020.
- [8] M. Reda, A. Onsy, A. Y. Haikal, and A. Ghanbari, "Path planning algorithms in the autonomous driving system: A comprehensive review," *Robotics and Autonomous Systems*, vol. 174, p. 104630, 2024.

- [9] S. Aradi, "Survey of deep reinforcement learning for motion planning of autonomous vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 2, pp. 740–759, 2022.
- [10] B. R. Kiran, I. Sobh, V. Talpaert, P. Mannion, A. A. Sallab, S. Yogamani, and P. Pérez, "Deep reinforcement learning for autonomous driving: A survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 6, pp. 4909–4926, 2022.
- [11] H. Mouhagir, R. Talj, V. Cherfaoui, F. Aioun, and F. Guillemard, "Integrating safety distances with trajectory planning by modifying the occupancy grid for autonomous vehicle navigation," in *IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, 2016, pp. 1114–1119.
- [12] M. Shekells, T. M. Caldwell, and M. Kobilarov, "Fast approximate path coordinate motion primitives for autonomous driving," in *IEEE 56th Annual Conference on Decision and Control (CDC)*, 2017, pp. 837–842.
- [13] G. Klančar, M. Seder, S. Blažič, I. Škrjanc, and I. Petrović, "Drivable path planning using hybrid search algorithm based on e* and bernstein-bézier motion primitives," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, no. 8, pp. 4868–4882, 2021.
- [14] L. Kavragi, P. Svestka, J.-C. Latombe, and M. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 4, pp. 566–580, 1996.
- [15] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *Int. J. Rob. Res.*, vol. 30, no. 7, p. 846–894, jun 2011.
- [16] O. Arslan, K. Berntorp, and P. Tsotras, "Sampling-based algorithms for optimal motion planning using closed-loop prediction," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 4991–4996.
- [17] J. Wang, W. Chi, C. Li, C. Wang, and M. Q.-H. Meng, "Neural RRT*: Learning-based optimal path planning," *IEEE Transactions on Automation Science and Engineering*, vol. 17, no. 4, pp. 1748–1758, 2020.
- [18] J. Wang, B. Li, and M. Q.-H. Meng, "Kinematic constrained bi-directional RRT with efficient branch pruning for robot path planning," *Expert Systems with Applications*, vol. 170, p. 114541, 2021.
- [19] M. Montemerlo, J. Becker, S. Bhat, H. Dahlkamp, D. Dolgov, S. Ertinger, D. Haehnel, T. Hilden, G. Hoffmann, B. Huhnke, D. Johnston, S. Klumpp, D. Langer, A. Levandowski, J. Levinson, J. Marcil, D. Orenstein, J. Paefgen, I. Penny, A. Petrovskaya, M. Pflueger, G. Stanek, D. Stavens, A. Vogt, and S. Thrun, "Junior: The Stanford entry in the urban challenge," *Journal of Field Robotics*, vol. 25, no. 9, pp. 569–597, 2008.
- [20] Z. Ajanovic, B. Lacevic, B. Shyrokau, M. Stolz, and M. Horn, "Search-based optimal motion planning for automated driving," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 4523–4530.
- [21] D. Kim, G. Kim, H. Kim, and K. Huh, "A hierarchical motion planning framework for autonomous driving in structured highway environments," *IEEE Access*, vol. 10, pp. 20 102–20 117, 2022.
- [22] J. Rong, S. Arrigoni, N. Luan, and F. Braghin, "Attention-based sampling distribution for motion planning in autonomous driving," in *39th Chinese Control Conference (CCC)*, 2020, pp. 5671–5676.
- [23] M. Diehl, H. G. Bock, and J. P. Schlöder, "A real-time iteration scheme for nonlinear optimization in optimal feedback control," *SIAM Journal on Control and Optimization*, vol. 43, no. 5, pp. 1714–1736, 2005.
- [24] X. Qian, F. Althé, P. Bender, C. Stiller, and A. de La Fortelle, "Optimal trajectory planning for autonomous driving integrating logical constraints: An miqp perspective," in *IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, 2016, pp. 205–210.
- [25] J. Li, X. Xie, Q. Lin, J. He, and J. M. Dolan, "Motion planning by search in derivative space and convex optimization with enlarged solution space," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022, pp. 13 500–13 507.
- [26] Gurobi Optimization, LLC, "Gurobi Optimizer Reference Manual," 2023. [Online]. Available: <https://www.gurobi.com>
- [27] J. Johnson and K. Hauser, "Optimal longitudinal control planning with moving obstacles," in *IEEE Intelligent Vehicles Symposium (IV)*, 2013, pp. 605–611.
- [28] P. Bender, O. S. Tas, J. Ziegler, and C. Stiller, "The combinatorial aspect of motion planning: Maneuver variants in structured environments," in *IEEE Intelligent Vehicles Symposium (IV)*, 2015, pp. 1386–1392.
- [29] T. Marcucci, M. Petersen, D. von Wrangel, and R. Tedrake, "Motion planning around obstacles with convex optimization," *Science Robotics*, vol. 8, no. 84, 2023.
- [30] S. Deolasee, Q. Lin, J. Li, and J. M. Dolan, "Spatio-temporal motion planning for autonomous vehicles with trapezoidal prism corridors and Bézier curves," in *American Control Conference, San Diego, CA, USA*. IEEE, 2023, pp. 3207–3214.
- [31] V. A. Laurence and J. C. Gerdes, "Long-horizon vehicle motion planning and control through serially cascaded model complexity," *IEEE Transactions on Control Systems Technology*, vol. 30, no. 1, pp. 166–179, 2022.
- [32] P. Wang, D. Liu, J. Chen, H. Li, and C.-Y. Chan, "Decision making for autonomous driving via augmented adversarial inverse reinforcement learning," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 1036–1042.
- [33] E. Bronstein, M. Palatucci, D. Notz, B. White, A. Kuefler, Y. Lu, S. Paul, P. Nikdel, P. Mouglin, H. Chen, J. Fu, A. Abrams, P. Shah, E. Racah, B. Frenkel, S. Whiteson, and D. Anguelov, "Hierarchical model-based imitation learning for planning in autonomous driving," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022, pp. 8652–8659.
- [34] Z. Szalay, T. Tettamanti, D. Esztergár-Kiss, I. Varga, and C. Bartolini, "Development of a test track for driverless cars: Vehicle design, track configuration, and liability considerations," *Periodica Polytechnica Transportation Engineering*, vol. 46, no. 1, p. 29–35, 2018.
- [35] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach (4th Edition)*. Pearson, 2020.
- [36] F. Torrisi and A. Bemporad, "Hysdel-a tool for generating computational hybrid models for analysis and synthesis problems," *IEEE Transactions on Control Systems Technology*, vol. 12, no. 2, pp. 235–249, 2004.
- [37] H. P. Williams, *Model Building in Mathematical Programming*. Hoboken, N.J.: Wiley, 2013.
- [38] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.
- [39] J. Zhou, "Interaction and uncertainty-aware motion planning for autonomous vehicles using model predictive control," Ph.D. dissertation, Linköping University Electronic Press, 2023.
- [40] J. Ziegler, P. Bender, T. Dang, and C. Stiller, "Trajectory planning for Bertha - a local, continuous method," in *IEEE Intelligent Vehicles Symposium, Proceedings*, 06 2014, pp. 450–457.
- [41] M. Werling, J. Ziegler, S. Kammel, and S. Thrun, "Optimal trajectory generation for dynamic street scenarios in a Frenet frame," in *IEEE International Conference on Robotics and Automation*, 06 2010, pp. 987 – 993.
- [42] R. Reiter, A. Nurkanović, J. Frey, and M. Diehl, "Frenet-Cartesian model representations for automotive obstacle avoidance within nonlinear MPC," *European Journal of Control*, p. 100847, 2023.
- [43] J. Eilbrecht and O. Stursberg, "Challenges of trajectory planning with integrator models on curved roads," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 15 588–15 595, 2020, 21st IFAC World Congress.
- [44] M. Wang, N. Mehr, A. Gaidon, and M. Schwager, "Game-theoretic planning for risk-aware interactive agents," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 6998–7005.
- [45] S. Le Cleac, M. Schwager, and Z. Manchester, "Algames: A fast augmented Lagrangian solver for constrained dynamic games," *Auton. Robots*, vol. 46, no. 1, p. 201–215, jan 2022.
- [46] C. Burger and M. Lauer, "Cooperative multiple vehicle trajectory planning using miqp," in *21st International Conference on Intelligent Transportation Systems (ITSC)*, 2018, pp. 602–607.
- [47] D. Heß, R. Lattarulo, J. Pérez, J. Schindler, T. Hesse, and F. Köster, "Fast maneuver planning for cooperative automated vehicles," in *21st International Conference on Intelligent Transportation Systems (ITSC)*, 2018, pp. 1625–1632.
- [48] B. Schürmann, D. Heß, J. Eilbrecht, O. Stursberg, F. Köster, and M. Althoff, "Ensuring drivability of planned motions using formal methods," in *IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, 2017, pp. 1–8.
- [49] J. Guanetti, Y. Kim, and F. Borrelli, "Control of connected and automated vehicles: State of the art and future challenges," *Annual Reviews in Control*, vol. 45, pp. 18–40, 2018.
- [50] P. Trautman and A. Krause, "Unfreezing the robot: Navigation in dense, interacting crowds," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010, pp. 797–803.
- [51] N. Buckman, A. Pierson, W. Schwarting, S. Karaman, and D. Rus, "Sharing is caring: Socially-compliant autonomous intersection negotiation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 6136–6143.
- [52] M. Treiber, A. Hennecke, and D. Helbing, "Congested traffic states in empirical observations and microscopic simulations," *Physical review. E*,

Statistical physics, plasmas, fluids, and related interdisciplinary topics, vol. 62 Pt A, pp. 1805–24, 2000.

- [53] P. A. Lopez, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y.-P. Flötteröd, R. Hilbrich, L. Lücken, J. Rummel, P. Wagner, and E. Wießner, "Microscopic traffic simulation using SUMO," in *21st IEEE International Conference on Intelligent Transportation Systems*. IEEE, 2018.
- [54] R. Verschueren, G. Frison, D. Kouzoupis, J. Frey, N. van Duijkeren, A. Zanelli, B. Novoselnik, T. Albin, R. Quirynen, and M. Diehl, "acados – a modular open-source framework for fast embedded optimal control," *Mathematical Programming Computation*, Oct 2021.



Rudolf Reiter received his Master's degree in 2016 in Electrical Engineering from the Graz University of Technology, Austria. From 2016 to 2018 he worked as a Control Systems Specialist at the Anton Paar GmbH, Graz, Austria. From 2018 to 2021 he worked as a researcher for the Virtual Vehicle Research Center, in Graz, Austria, where he started his Ph.D. in 2020 under the supervision of Prof. Dr. Moritz Diehl. Since 2021, he continued his Ph.D. at a Marie-Sklodowska Curie Innovative Training Network position at the University of Freiburg, Germany.

His research focus is within the field of learning- and optimization-based motion planning and control for autonomous vehicles and he is an active member of the Autonomous Racing Graz team.



Armin Nurkanović received the B.Sc. degree from the Faculty of Electrical Engineering, Tuzla, Bosnia and Herzegovina, in 2015, and the M.Sc. degree from the Department of Electrical and Computer Engineering, Technical University of Munich, Munich, Germany, in 2018. He defended his Ph.D. in 2023 at the Systems Control and Optimization Laboratory, Department of Microsystems Engineering, University of Freiburg, Germany, under the supervision of Prof. Moritz Diehl. He received the IEEE Control Systems Letters Outstanding Paper Award in 2022.

His research interests are in the area of numerical methods for model predictive control, nonlinear optimization, and optimal control of nonsmooth and hybrid dynamical systems.



Daniele Bernardini received the Master's degree in computer engineering in 2007 and the Ph.D. degree in information engineering in 2011 from the University of Siena, Italy, with a focus on model predictive control. From 2011 to 2015 he was a Post-Doctoral Fellow at IMT Lucca, Italy. In 2011 he co-founded ODYS S.r.l., where he is CTO, specializing in the development of efficient MPC solutions for embedded systems. He received the 2021 SAE Environmental Excellence in Transportation Award. His research interests include MPC, stochastic control,

networked control systems, hybrid systems, and their application to real-time problems in the automotive, aerospace, robotics, and energy domains.



Moritz Diehl studied mathematics and physics at Heidelberg University, Germany and Cambridge University, U.K., and received the Ph.D. degree in optimization and nonlinear model predictive control from the Interdisciplinary Center for Scientific Computing, Heidelberg University, in 2001. From 2006 to 2013, he was a professor with the Department of Electrical Engineering, KU Leuven University Belgium. Since 2013, he is a professor at the University of Freiburg, Germany, where he heads the Systems Control and Optimization Laboratory, Department of Microsystems Engineering (IMTEK), and is also with the Department of Mathematics. His research interests include optimization and control, spanning from numerical method development to applications in different branches of engineering, with a focus on embedded and on renewable energy systems.



Alberto Bemporad (Fellow, IEEE) received his Master's degree cum laude in Electrical Engineering in 1993 and his Ph.D. in Control Engineering in 1997 from the University of Florence, Italy. In 1996/97 he was with the Center for Robotics and Automation, Department of Systems Science & Mathematics, Washington University, St. Louis. In 1997-1999 he held a postdoctoral position at the Automatic Control Laboratory, ETH Zurich, Switzerland, where he collaborated as a Senior Researcher until 2002. In 1999-2009 he was with the

Department of Information Engineering of the University of Siena, Italy, becoming an Associate Professor in 2005. In 2010-2011 he was with the Department of Mechanical and Structural Engineering of the University of Trento, Italy. Since 2011 he has been a Full Professor at the IMT School for Advanced Studies Lucca, Italy, where he served as the Director of the institute in 2012-2015. He spent visiting periods at Stanford University, University of Michigan, and Zhejiang University. In 2011 he co-founded ODYS S.r.l., a company specialized in developing model predictive control systems for industrial production. He has published more than 400 papers in the areas of model predictive control, hybrid systems, optimization, automotive control, and is the co-inventor of 21 patents. He is the author or coauthor of various software packages for model predictive control design and implementation, including the Model Predictive Control Toolbox (The Mathworks, Inc.) and the Hybrid Toolbox for MATLAB. He was an Associate Editor of the IEEE Transactions on Automatic Control during 2001-2004 and Chair of the Technical Committee on Hybrid Systems of the IEEE Control Systems Society in 2002-2010. He received the IFAC High-Impact Paper Award for the 2011-14 triennial, the IEEE CSS Transition to Practice Award in 2019, and the 2021 SAE Environmental Excellence in Transportation Award. He has been an IEEE Fellow since 2010.