CommonRoad: Composable Benchmarks for Motion Planning on Roads

Matthias Althoff, Markus Koschi, and Stefanie Manzinger

***Abstract*— Numerical experiments for motion planning of road vehicles require numerous components: vehicle dynamics, a road network, static obstacles, dynamic obstacles and their movement over time, goal regions, a cost function, etc. Providing a description of the numerical experiment precise enough to reproduce it might require several pages of information. Thus, only key aspects are typically described in scientific publications, making it impossible to reproduce results—yet, reproducibility is an important asset of good science. Composable benchmarks for motion planning on roads (CommonRoad) are proposed so that numerical experiments are fully defined by a unique ID; all information required to reconstruct the experiment can be found on the CommonRoad website. Each benchmark is composed by a vehicle model, a cost function, and a scenario (including goals and constraints). The scenarios are partly recorded from real traffic and partly hand-crafted to create dangerous situations. We hope that CommonRoad saves researchers time since one does not have to search for realistic parameters of vehicle dynamics or realistic traffic situations, yet provides the freedom to compose a benchmark that fits one's needs.**

I. Introduction

Reproducibility of results is a cornerstone of science [1], [2]. One obstacle towards reproducibility in motion planning of road vehicles is that details of the experimental results are often not fully provided—some reasons are page limitations of publications, an overwhelming number of required details, or simply because some details are taken for granted. Providing detailed benchmarks would help in this regard and also simplify comparing different planning methods.

First attempts to improve reproducibility and comparability of motion planning algorithms have been made in the robotics community, but mostly for (mobile) robotic manipulators and not for motion planning in the automotive sector. This work provides the first benchmark collection for motion planning on roads, which specifies in depth the motion planning problem consisting of initial state, goal region, road network, static and dynamic obstacles, and the model of the ego vehicle (vehicle for which motion planning is conducted). Before highlighting the main features of CommonRoad, we present a literature review that is categorized into *benchmark problems*, *datasets*, and *motion planning libraries*. Most previous work in robotic motion planning focused on providing libraries that facilitate benchmarking, without providing a set of benchmark problems in a standardized form. We address this problem by providing composable benchmarks that can be referenced to with a unique ID. Our proposed collection also facilitates benchmarking, but this paper does not provide performance metrics—this should be better determined by workshops to reach consensus.

a) Benchmarks: We would first like to note that we only reference benchmarks that are still publicly available. The need of benchmarks in robotics is formulated in [3], but this early work does not provide a specific benchmark. Several European projects for benchmarking in robotics have been conducted in the 2000s (e.g. [4]–[6]), but none has considered motion planning on roads. Detailed benchmarks have been developed in particular for robotic grasping [7], [8] and for robotic manipulators with a focus on indoor human environments [9]. More abstract benchmark problems for motion planning are provided by the Texas A&M University[1] and by Rice University[2].

b) Datasets: While no benchmarks for motion planning on roads exist, recordings of vehicle movements are available; however, none of them is a benchmark problem since initial state, goal regions, and a dynamic vehicle model are missing. Furthermore, there exists no data format commonly used by different research groups. One of the most popular datasets of recorded traffic participants is from the *Next Generation Simulation (NGSIM)* program [10], [11]. Other datasets exist, but they have not recorded all relevant vehicles in a common reference frame, see e.g. [12]. Another class of works provides results on recorded data, but the data has never been or is no longer publicly available, e.g. [13]–[15].

c) Motion planning libraries: One of the most successful motion planning libraries in robotics is the *Open Motion Planning Library (OMPL)* [16], which implements many of the most important sampling-based approaches. The OMPL has also been integrated into *MoveIt!* [17], but remains to be a stand-alone software. *MoveIt!* itself is integrated into the *Robot Operating System (ROS)* [18]. Currently, further infrastructure to facilitate benchmarking with OMPL is developed [19]. Earlier libraries for sampling-based motion planning are the *Online, Open-source, Programming System for Motion Planning (OOPS$_{MP}$)* [20] and the *Open Robotics and Animation Virtual Environment (OpenRAVE)* [21] with similar goals as OMPL. Both libraries contain some benchmark problems (none for automated driving), but their focus is on the implementation of planning algorithms. A library implementing graph-based search is the *Search-Based*

*This work was supported by the Deutsche Forschungsgemeinschaft (German Research Foundation) within the Priority Programme SPP 1835 Cooperative Interacting Automobiles (grant number: AL 1185/4-1) and by the BMW Group within the Car@TUM project.

All authors have equally contributed to this work and are with Faculty of Informatics, Technische Universität München, 85748 Garching, Germany {althoff, markus.koschi, stefanie.manzinger}@tum.de

[1] parasol.tamu.edu/groups/amatogroup/benchmarks

[2] plannerarena.org

Planning Library (SBPL)[3], which is useful if one e.g. uses motion primitives that span a search tree [22]. Besides graph-based techniques, there also exists the *Covariant Hamiltonian Optimization for Motion Planning (CHOMP)* library for gradient-based optimization techniques [23].

d) Automotive benchmarks beyond motion planning: One of the most successful automotive benchmarks is the KITTI benchmark targeting computer vision [24]. Another important aspect is simultaneous localization and mapping; the OpenSLAM[4] project and the Radish project[5] host a collection of benchmarks and libraries for SLAM.

e) Novelty and key features: CommonRoad is a benchmark collection for motion planning of road vehicles (available at **commonroad.in.tum.de**) with the following features:

- **Reproducibility/unambiguity:** All information required to reproduce the results of a motion planner is provided in an unambiguous way and explained by manuals on our website.
- **Composability:** Our benchmarks are composed of vehicle models, cost functions, and scenarios (including goals and constraints). All components are carefully chosen to easily combine and interchange them.
- **Representativeness:** Our benchmark problems contain recorded traffic to faithfully represent real traffic and hand-crafted problems since most recorded traffic situations are not critical/dangerous.
- **Portability:** We use XML to describe our scenarios, which is platform-independent. We also provide executable vehicle models implemented in MATLAB and Python, which are also platform-independent.
- **Scalability:** Our benchmark examples range from simple static scenarios with a few obstacles and a large driving corridor (i.e. region where collisions cannot take place) to complex scenarios with many dynamic obstacles and a small driving corridor.
- **Openness:** All benchmarks are freely available from our website with the possibility to suggest new ones.
- **Independence:** Our benchmarks are independent from planning libraries and our scenario representation could serve as an interchange format between other tools.

II. Benchmark Composition and Planning Problem

As previously mentioned, we compose benchmarks using *vehicle models*, *cost functions*, and *scenarios (including goals and constraints)*. This modularity makes it easy to generate many benchmarks from a smaller set of components and also simplifies comparing the effects of vehicle models or cost functions by only changing those components.

A. Benchmark Composition

Let us introduce with M, C, S, and B the respective IDs of the model, the cost function, the scenario, and the benchmark. The benchmark ID is constructed by separating partial IDs by colons in the following order:

$$\text{B} = \text{M:C:S}.$$

For instance, for M=PM2, C=JB1, S=OV001, the benchmark ID is B = PM2:JB1:OV001. If using one's own component is preferred, one can use the ID IND (for individual). For instance, if one uses an individual cost function for the previous example, the ID becomes PM2:IND:OV001. If one prefers to build upon an existing component, which is modified, the new ID should have *M-* as a prefix. After modifying the model of the first example, the new ID is M-PM2:JB1:OV001 (of course, the modification should be described in detail). In case a collaborative planning benchmark is composed, we list the models and cost functions for n controllable traffic participants by n-dimensional lists:

$$\text{B} = [\text{M}_1,\ldots,\text{M}_n]\text{:}[\text{C}_1,\ldots,\text{C}_n]\text{:C-S},$$

where the index refers to the vehicle and a collaborative scenario is indicated by using the prefix *C-*, where *M-* should be used first if both are required. The i-th model and cost function corresponds to the i-th ego vehicle specified in the scenario XML file as introduced later in Sec. V-C. If only one model and/or cost function is used, it is assumed that all controlled vehicles use the same one. For instance, the benchmark ID for M_1=PM1, C_1=JB1, M_2=PM3, C_2=JB1, M_3=ST2, C_3=SA1, and S=C-OV011, is B = [PM1,PM3,ST2]:[JB1,JB1,SA1]:C-OV011.

B. Motion Planning Problem

The proposed benchmarks codify an optimization problem whose solution is the motion plan. Let us denote by $f_M(x(t), u(t))$ the right hand side of the state space model of vehicle model M so that

$$\dot{x}(t) = f_M(x(t), u(t)), \tag{1}$$

where $x \in \mathbb{R}^n$ is the state vector and $u \in \mathbb{R}^m$ is the input vector. We further require the initial state $x_{0,S} \in \mathbb{R}^n$ ($x(t_0) = x_{0,S}$) provided by scenario S, the initial time t_0, and the final time t_f. More details on the models can be found in Sec. III. The cost function J_C of ID C consisting of terminal costs Φ_C and running costs L_C is

$$\begin{aligned}&J_C(x(t), u(t), t_0, t_f)\\&= \Phi_C(x(t_0), t_0, x(t_f), t_f) + \int_{t_0}^{t_f} L_C(x(t), u(t), t)\,\mathtt{d}t,\end{aligned}$$

which is detailed in Sec. IV. We denote the time-varying, free drivable space on the road surface as $\mathcal{W}_{\mathtt{S,free}}(t) \subset \mathbb{R}^2$ and introduce $O(x(t)) : \mathbb{R}^n \to P(\mathbb{R}^2)$ ($P()$ returns the power set) as the function that returns the occupancy of a vehicle given its state. A possible solution has to ensure that the occupancy of the vehicle is in the free space ($\forall t \in [t_0, t_f] : O(x(t)) \in \mathcal{W}_{\mathtt{S,free}}(t)$) and respects additional constraints $g_S(x(t), u(t), t) \leq 0$ provided by scenario S, such as speed limits or other traffic rules [25]. Equality constraints can be

[3]wiki.ros.org/sbpl
[4]www.openslam.org
[5]radish.sourceforge.net

constructed from inequality constraints (e.g. $x \leq 0 \wedge -x \leq 0 \equiv x = 0$). Let us further denote the goal region $\mathcal{G}_S \subset \mathbb{R}^n$ of scenario S, which can be disjoint sets (see Sec. V-C). As soon as $x(t) \in \mathcal{G}_S$ at time $t = t_f$, a feasible solution is found. After introducing an input trajectory as $u(\cdot)$ (in contrast to a value $u(t)$ at time t), we can finally formulate the motion planning problem as finding

$$u^*(\cdot) = \underset{u(\cdot)}{\arg\min}\ J_C(x(t), u(t), t_0, t_f) \qquad (2)$$

subject to

$$\dot{x}(t) = f_M(x(t), u(t)), \quad O(x(t)) \in \mathcal{W}_{\mathtt{S,free}}(t),$$
$$g_S(x(t), u(t), t) \leq 0, \quad x(t_0) = x_{0,S}, \quad x(t_f) \in \mathcal{G}_S.$$

Associated with the optimal input trajectory $u^*(\cdot)$ in (2) is an optimal state trajectory $x^*(\cdot)$ that can be obtained by a forward simulation of (1). Directly solving (2) is referred to as *trajectory planning* (see [26, Sec. 4.]). An alternative is to first find a path that the vehicle should follow for which an optimal velocity profile is computed, which we refer to as *path planning* with subsequent *velocity optimization* (see [26, Sec. 4.]). Both techniques can be used to solve our proposed benchmarks.

III. Vehicle Models

This section presents models for vehicle dynamics ranging from simple to complex. For each model it is assumed that underlying controllers exist that can realize a commanded acceleration (positive and negative within given limits). For adaptive cruise control in particular, numerous works already exist that realize a commanded acceleration, see e.g. [27], [28]. The effects of engine characteristics in terms of fuel consumption can be considered in the cost function (see Sec. IV).

The lateral dynamics, however, cannot be abstracted away to the same extent using controllers, especially when constraints such as the danger of roll-over must be considered in extreme maneuvers [29], [30]. For this reason, our models consider increasingly complex lateral vehicle dynamics and tire models: point-mass model, kinematic single-track model, single-track model, and a multi-body model. Some details of the first two models are presented subsequently, whereas due to space restrictions, the full detailed description of the single-track model and the multi-body model can be found in our *vehicle model documentation* on our website. Executable MATLAB and Python implementations of all presented models are also available. We have not included Dubin or Reeds-Shepp cars since they require changing the steering angle infinitely fast (see e.g. [31]).

The model IDs are constructed by first choosing the model type (e.g. ST for single-track) followed by a number, which refers to the parameterization in the *vehicle model documentation* of our repository.

A. Point-Mass Model (M=PM)

The point-mass model is the simplest model that is commonly used for motion planning, see e.g. [32], [33]. This model abstracts the vehicle as a point mass whose absolute acceleration is bounded (Kamm's circle). Let us introduce $\square$ as the placeholder for a variable and $\square_x$ and $\square_y$ to denote the value of the corresponding variable in x and y direction (world coordinates), respectively. After further introducing position s, acceleration a, and maximum absolute acceleration $a_{\mathtt{max}}$, the dynamics is

$$\ddot{s}_x = a_x, \quad \ddot{s}_y = a_y, \quad \sqrt{a_x^2 + a_y^2} \leq a_{\mathtt{max}}.$$

The point-mass model ignores the minimum turning circle, which is considered next in the kinematic single-track model.

B. Kinematic Single-Track Model (M=KS)

The kinematic single-track model (also known as the kinematic bicycle model) considers only two wheels, where the front and rear wheel pairs are each lumped into one wheel, because the roll dynamics is neglected (see Fig. 1 and [34, Sec. 2.2]). This also explains the term *single-track model*. Tire slip is not considered, but the kinematic single-track model can be used when the vehicle does not operate close to its physical capabilities [26], [35]. For instance, when planning a parking maneuver, tire slip is not important, but the point-mass model would not be sufficient since the non-holonomic behavior and, in particular, the minimum turning radius would not be considered.

In addition to the variables already introduced, we also require the velocity of the steering angle v_δ, the steering angle δ, the heading Ψ, and the parameter l describing the wheelbase as well as the parameter v_S describing the velocity above which the engine power is limiting maximum positive acceleration rather than maximum tire forces (see Fig. 1). We further denote by $\underline{\square}$ the minimum possible value, by $\overline{\square}$ the maximum possible value, by $\square_{\mathtt{lat}}$ the value of a variable in lateral direction, and by $\square_{\mathtt{long}}$ the value in longitudinal direction (vehicle-fixed coordinates). The differential equations of the kinematic single-track model as defined in this work are

$$\dot{\delta} = v_\delta, \quad \dot{\Psi} = \frac{v}{l}\tan(\delta), \quad \dot{v} = a_{\mathtt{long}},$$
$$\dot{s}_x = v\cos(\Psi), \quad \dot{s}_y = v\sin(\Psi),$$

under consideration of the constraints

$$v_\delta \in [\underline{v}_\delta, \overline{v}_\delta], \quad \delta \in [\underline{\delta}, \overline{\delta}], \quad v \in [\underline{v}, \overline{v}], \qquad (3)$$

$$a_{\mathtt{long}} \in [-a_{\mathtt{max}}, \overline{a}], \quad \overline{a} = \begin{cases} a_{\mathtt{max}} \frac{v_S}{v} & \text{for } v > v_S, \\ a_{\mathtt{max}} & \text{otherwise}, \end{cases} \qquad (4)$$

$$\sqrt{a_{\mathtt{long}}^2 + (v\,\dot{\Psi})^2} \leq a_{\mathtt{max}} \qquad (a_{\mathtt{lat}} = v\,\dot{\Psi}). \qquad (5)$$

Constraint (3) considers that the steering velocity, the steering angle, and the vehicle velocity are bounded. Limited engine power and braking power as detailed in [36, Sec. III.B] are considered by (4). Finally, as in the point-mass model, constraint (5) models Kamm's circle.

Note that kinematic single-track models differ slightly in publications, depending on whether one considers that 1) the steering angle or the steering velocity is an input, 2) the vehicle velocity or the vehicle acceleration is an input, or 3) the front or rear wheel is the reference point (here: rear wheel, see Fig. 1). For instance, in [26, eq. (8)], the vehicle velocity and the steering velocity are inputs. Additionally, other works do not provide all the constraints of our model (which can be easily removed, but a removal should be stated since this simplifies motion planning).

C. Single-Track Model (M=ST)

The natural extension of the kinematic single-track model is the single-track model (also known as the bicycle model), which considers tire slip [34, Sec. 2.3] influencing the slip angle β, which is illustrated in Fig. 1 as the angle between the velocity vector v and the vehicle orientation Ψ. Works that perform planning of evasive maneuvers closer to physical limits require the single-track model, see e.g. [37], [38]. We additionally consider the load transfer of the vehicle due to longitudinal acceleration $a_{\texttt{long}}$ (neglecting suspension dynamics). Due to space limitations, we refer the reader to our *vehicle model documentation* for a detailed description and derivation of the single-track model.

Since the single-track model uses a linear relationship between slip angle and tire force (thus ignoring saturation effects), constraint (5) is important for limiting possible tire forces. Please note that in contrast to this work, other works often only consider constant velocity when referring to a single-track model (see e.g. [34, Sec. 2.3]). Also, the weight transfer between the front and rear axle is often neglected in single-track models (see e.g. [37]).

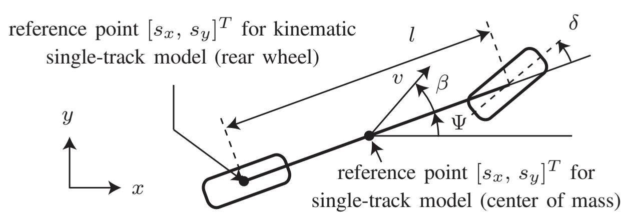


Fig. 1. Combined illustration of kinematic/standard single-track model.

D. Multi-Body Model (M=MB)

Although the previously introduced single-track model already considers many important effects of vehicle dynamics, it does not consider the vertical load of all 4 wheels due to roll, pitch, and yaw, their individual spin and slip, and nonlinear tire dynamics. An example of a multi-body model used for motion planning of a road vehicle can be found in [39]. Although many commercial multi-body models for vehicle dynamics exist[6], those models are proprietary and thus not appropriate for a benchmark that requires public accessibility. Our multi-body model is taken out of [40, Appendix A], which is one of few detailed and accessible multi-body dynamics descriptions. Due to the complexity of the multi-body model, we refer to the *vehicle model documentation* of our repository and only mention the main features.

[6]www.carsim.com, www.tesis-dynaware.com, www.mscsoftware.com

The multi-body dynamics is described by 3 masses: The unsprung mass and the sprung masses of the front and rear axles. The forces between these masses are described by the dynamics of the suspension and the tire model. We consider all suspension forces in [40, Appendix A] originating from springs, dampers, and anti-roll bars. For the tire dynamics we use the PAC2002 Magic-Formula tire model, which is widely used in industry [41]. Rewriting all equations as a state space model yields 29 state variables.

E. Numerical Experiments and Interchangeability of Models

In order to facilitate switching between different models and to compare results as done in this subsection, we describe in our *vehicle model documentation* how parameter sets and initial states can be converted in the best possible way between models. There, we further provide state-space formats of all models so that it is easier to build one's own executable models in addition to the ones in MATLAB and Python.

To illustrate better differences between models, we briefly present numerical experiments for a BMW 320i (parameter set 2 in *vehicle model documentation*). The duration of each experiment is 1 s, and the initial velocity is 15 m/s; further details of the experiments can be found in the *vehicle model documentation*. First, we compare the kinematic single-track model, the single-track model, and the multi-body model when driving a left curve. It can be easily seen in Fig. 2(a) that the kinematic single-track model realizes the tightest bend since it does not consider tire slip; the single-track model is a little wider due to considering tire slip. This effect is even stronger for the multi-body model since it already considers saturation of tire forces before constraint (5) is active. This can be seen even better when comparing the slip angles of the single-track model and the multi-body model in Fig. 2(b).

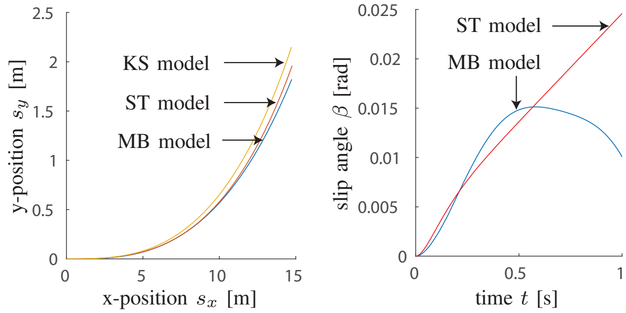
Second, we demonstrate understeering and oversteering (see [34, Sec. 3.3]) for the multi-body model during cornering by braking into the corner ($a_{\texttt{long}} = -0.7$ g, g represents the gravity constant), coasting ($a_{\texttt{long}} = 0$ g), and heavily accelerating ($a_{\texttt{long}} = 0.63$ g) the rear-wheel-driven vehicle (power oversteer) as shown by the slip angle in Fig. 3(a). It can also be easily observed, by plotting the pitch in Fig. 3(b), that the vehicle is "diving" during braking while the front lifts during acceleration.

IV. Cost Functions

This section proposes standardized cost functions for the motion planning problem in (2). Analogously to the composability of the benchmarks, we compose different types of partial cost functions to a single cost function. The partial cost functions have a unique ID p and the set $\mathcal{P}$ contains all IDs of the proposed partial cost functions. The overall cost function is obtained by the weighted sum

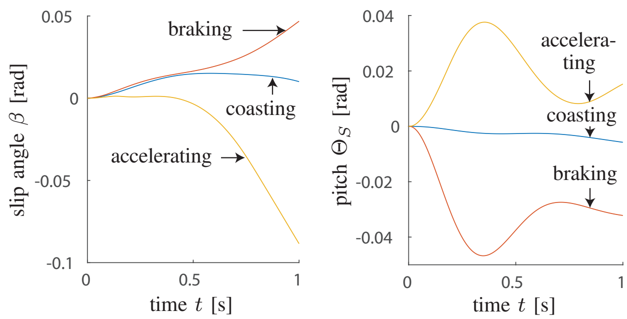
$$J_C(x(t), u(t), t_0, t_f) = \sum_{i \in \mathcal{I}} w_i\, J_i(x(t), u(t), t_0, t_f),$$

where $\mathcal{I} \subset \mathcal{P}$ contains the IDs of the applied partial cost functions and $w_i \in \mathbb{R}^+$ are weights. We first present popular



(a) Path of center of gravity. (b) Slip angle.

Fig. 2. Comparing the kinematic single-track (KS) model, the single-track (ST) model, and the multi-body (MB) model during cornering.



(a) Slip angle. (b) Pitch.

Fig. 3. Investigating oversteering and understeering as well as pitch for the multi-body model.

partial cost functions using the variables already introduced in Sec. III:

- **Time**: $J_T = t_f$ (see [42, eq. 2]).
- **Acceleration**: $J_A = \int_{t_0}^{t_f} a(t)^2 \,\mathrm{d}t$ (see [43, Sec. III.B]).
- **Jerk**: $J_J = \int_{t_0}^{t_f} \dot{a}(t)^2 \,\mathrm{d}t$ (see [44, Sec. III]).
- **Steering angle**: $J_{SA} = \int_{t_0}^{t_f} \delta(t)^2 \,\mathrm{d}t$ (see [45]).
- **Steering rate**: $J_{SR} = \int_{t_0}^{t_f} v_\delta(t)^2 \,\mathrm{d}t$ (see [45]).
- **Energy**: $J_E = \int_{t_0}^{t_f} P(x(t), u(t)) \,\mathrm{d}t$, where $P(x(t), u(t))$ is the required power of the engine for the state x and the input u, which can be obtained from engine mappings (see [28, Sec. III.B]).
- **Yaw rate**: $J_Y = \int_{t_0}^{t_f} \dot{\Psi}(t)^2 \,\mathrm{d}t$ (see [43, Sec. III.B]).
- **Lane center offset**: $J_{LC} = \int_{t_0}^{t_f} d(t)^2 \,\mathrm{d}t$, where d is the distance to the lane center or a driving corridor (see [43, Sec. III.B]).
- **Velocity offset**: $J_V = \int_{t_0}^{t_f} (v_{\mathtt{des}}(x(t)) - v(t))^2 \,\mathrm{d}t$, where $v_{\mathtt{des}}(x(t))$ is the desired velocity for the vehicle state x (see [43, Sec. III.B]).
- **Orientation offset**: $J_O = \int_{t_0}^{t_f} (\theta_{\mathtt{des}}(x(t)) - \theta(t))^2 \,\mathrm{d}t$, where $\theta_{\mathtt{des}}(x(t))$ is the desired orientation for the vehicle state x (see [45]).
- **Distance to obstacles**: $J_D = \int_{t_0}^{t_f} \max(\xi_1(t), \ldots, \xi_o(t)) \,\mathrm{d}t$, where o is the number of obstacles, $\xi_i(t) = e^{-w_{\mathtt{dist}} d_i(t)}$, $d_i(t)$ is the distance of the ego vehicle to an obstacle, and $w_{\mathtt{dist}}$ is an additional required weight (see [46, eq. 7-8]).
- **Path length**: $J_L = \int_{t_0}^{t_f} v(t) \,\mathrm{d}t$ (see [46, Tab. 1]).
- **Terminal offset**: $J_{TO} = d(t_f)^2$ (see [44, eq. 2]).
- **Terminal distance to goal**: $J_{TG} = d_{\mathtt{goal}}(t_f)^2$, where $d_{\mathtt{goal}}$ is the distance to the goal (see [47, Sec. IV.D]).

Let us now introduce a notation for writing the used weights compactly. We write $w_T = 0.1$, $w_{SA} = 0.4$, and $w_Y = 0.7$ in short as $[(T|0.1), (SA|0.4), (Y|0.7)]$. After agreeing that we use SI units for all variables, this notation uniquely defines a cost function. Most works, however, do not provide such weights, so we cannot include their values in the current version of the benchmark. We therefore hope that once the structure is fixed, other researchers will contribute their used weights. Works that published their used weights are listed below, where the cost function ID is chosen as the initials of the first authors plus a running number:

- J_{JB1} from [42, eq. 2]: $[(T|1)]$.
- J_{SA1} inspired by [48, eq. 2]: $[(SA|0.1), (SR|0.1), (D|10^5)]$ (we use fewer parameters).
- J_{WX1} inspired by [46, Tab. IV]: $[(T|10), (V|1), (A|0.1), (J|0.1), (D|0.1), (LC|10)]$ (we use fewer parameters and velocity difference instead of absolute velocity).

V. Scenarios

As a last component, we introduce scenarios specified by an XML file, which is composed of 1) a formal representation of the road network, 2) static and dynamic obstacles, and 3) the planning problem of the ego vehicle(s) as shown in Fig. 4, where details of child elements are omitted for clarity. In the following subsections we briefly describe each data format in more detail. A detailed description can be found in the *XML documentation* on our website. We also provide a *scenario documentation* listing all available scenarios.

A. Road Network

For our benchmarks we use *lanelets* [49] as atomic, interconnected, and drivable road segments to represent the road network. A lanelet is defined by its *left* and *right bound*, where each bound is represented by an array of points (a polyline), as shown in Fig. 5. We have chosen lanelets since they are as expressive as other formats, such as e.g. OpenDRIVE[7], yet have a lightweight and extensible representation. Using lanelets allows the road network to be modeled as a *directed graph*, where each node has four types of outgoing edges: successor, predecessor, adjacentLeft, and adjacentRight (see Fig. 4; predecessor is not required but added for implementation reasons). Lanelets additionally contain *traffic regulations*, e.g. the speed limit. All road networks are stored using XML. The XML data structure of OpenStreetMap[8] can represent lanelets in the WGS84 coordinate frame using references between lanelets and primitive elements as described in [49]. Since we require Cartesian coordinates and a compact element structure to also represent obstacles and the planning problem, we propose our *CommonRoad* XML data format as specified on our website.

[7] opendrive.org
[8] openstreetmap.org

```
/
├─ lanelet
│   ├─ leftBound
│   │   ├─ point
│   │   └─ lineMarking
│   ├─ rightBound
│   │   ├─ point
│   │   └─ lineMarking
│   ├─ predecessor (ref to lanelet)
│   ├─ successor (ref to lanelet)
│   ├─ adjacentLeft (ref to lanelet)
│   ├─ adjacentRight (ref to lanelet)
│   └─ trafficRegulations
├─ obstacle
│   ├─ role: static
│   ├─ type: parkedVehicle/.../unknown
│   └─ shape
├─ obstacle
│   ├─ role: dynamic
│   ├─ type: car/truck/.../unknown
│   ├─ shape
│   └─ trajectory
│       └─ state
├─ obstacle
│   ├─ role: dynamic
│   ├─ type: car/truck/.../unknown
│   └─ occupancySet
│       └─ occupancy
├─ obstacle
│   ├─ role: dynamic
│   ├─ type: car/truck/.../unknown
│   ├─ shape
│   └─ probabilityDistribution
└─ planningProblem
    ├─ initialState
    └─ goalRegion
        └─ state
```

Fig. 4. Structure of the XML files encoding each scenario. For clarity we do not show all elements of the XML structure.

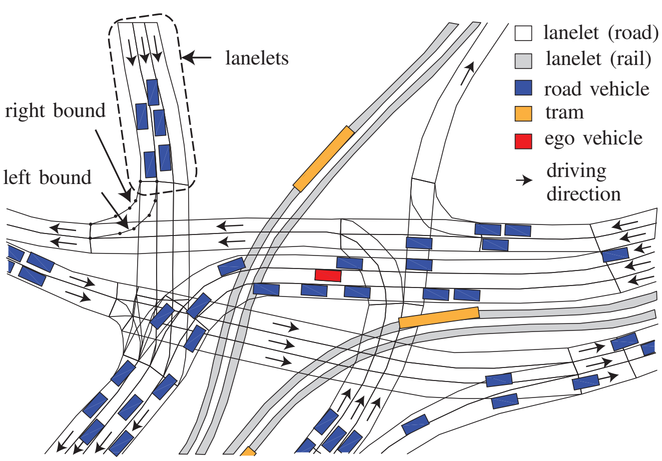


Fig. 5. Lanelets of a complex intersection in the city center of Munich (scenario ID S=GER_Muc_1a). Besides roads, tram rails are also modeled by lanelets.

B. Obstacles

Obstacles are characterized by their role (static/dynamic), type (car/truck/bus/bicycle/pedestrian/constructionZone/parkedVehicle/priorityVehicle/unknown), shape (rectangle/circle/polygon), and movement over time (if the obstacle is dynamic). We have restricted ourself to the shapes rectangle, circle, and polygon since rectangles are a good description for cars and trucks, circles are a good description of pedestrians, and any other two-dimensional shape can be modeled by a polygon if the number of points approaches infinity. If motion planners depend on other representations, one has to enclose the provided shape, see e.g. [50].

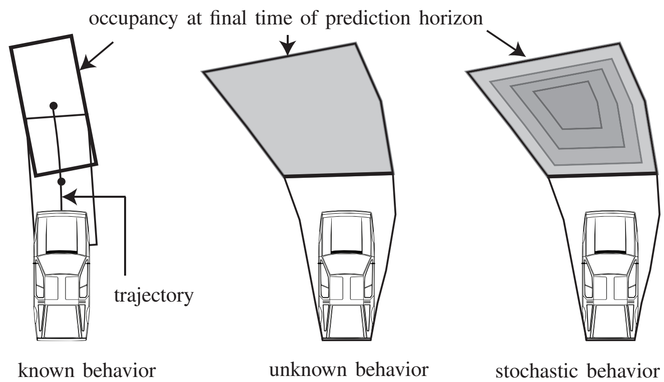


Fig. 6. Supported occupancy representation of predicted obstacle movements.

When the obstacle is dynamic, we provide three possibilities to describe the movement over time as illustrated in Fig. 6: known behavior, unknown behavior bounded by sets, and unknown behavior described by probability distributions.

a) Known behavior: We describe known behavior with a trajectory, which is modeled as state sequence containing position and orientation. After defining the reference points of shapes of obstacles, the occupancy of an obstacle along a trajectory is uniquely defined: the reference point of a rectangle and a circle is their geometric center and the reference point of a polygon is its first point (polygons are stored as an ordered list of points).

b) Unknown behavior: Occupancy sets that evolve over time are used to represent unknown behavior [36]. For occupancy sets we only allow polygons as a representation that can be obtained from our tool SPOT [51]. Please note that one can also represent known behavior by evolving occupancy sets, which do not change their size over time.

c) Unknown stochastic behavior: One can describe unknown stochastic behavior with probability distributions of states. Since many different probability distributions are used (e.g. Gaussian [52], piecewise constant [53], etc.), we provide a placeholder for probability distributions in our XML structure. Please note that for stochastic behavior, the distribution of the state and the dimension of the vehicle have to be stored separately to correctly compute crash probabilities [53, Sec. VI]. For this reason, we also store the shape of obstacles as we do for known behavior.

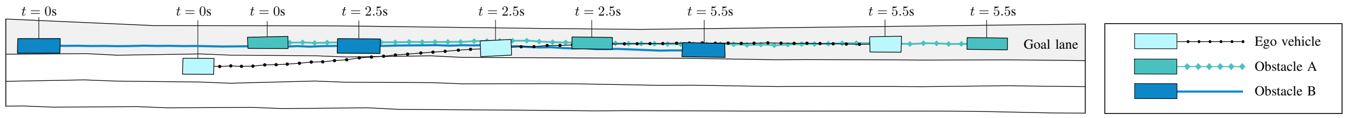


Fig. 7. Solution of our applied trajectory planner for scenario "NGSIM_US101_0".

C. Planning Problem

Each ego vehicle has an initial state as well as one or several goal regions. If several goal regions are provided, we implicitly assume that only one of them has to be reached, modeling options like overtaking or staying behind a vehicle. The position of the goal region is defined by a point, shape (rectangle/circle/polygon), or lanelet. For orientation, velocity, and time, intervals or exact values can be provided. Since different vehicle models can be used (see Sec. III), the shape of the ego vehicle is part of the parameterization of the model. Despite the fact that the different models have different state variables, we can initialize all models by the initial state of a single-track model as described in the *vehicle model documentation*.

VI. Example

We demonstrate our proposed benchmark collection with a deliberately simple scenario, which is based on recorded traffic data from the NGSIM U.S. 101 dataset (07:50 a.m. to 08:05 a.m.). Fig. 7 shows the trajectories of two vehicles and the initial position of the ego vehicle. We consider all lanes of the U.S. 101 highway provided by the NGSIM dataset; however, we only depict three out of six lanes in Fig. 7 for the sake of clarity. The goal of this scenario is to plan a lane-change maneuver for the ego vehicle to the left-most lane within a time horizon of $t_f \in [5.5, 6.0]$ s (see Fig. 7).

The applied trajectory planner is based on numerical optimization; for a detailed explanation of the algorithm, the interested reader is referred to [45, Sec. III.1]. In this paper, we use a kinematic single-track model as described in Sec. III based on the parameters KS1 described in the *vehicle model documentation*. However, in order to demonstrate how parameters can be modified, the parameter v_S is changed to $v_S \to \infty$. The cost function is chosen as

$$\begin{aligned} J_{SM1}(x(t), u(t), t_0, t_f) = \; & w_A J_A + w_{SA} J_{SA} \\ & + w_{SR} J_{SR} + w_{LC} J_{LC} + w_V J_V + w_O J_O, \end{aligned}$$

which minimizes the acceleration (J_A), steering effort (J_{SA} and J_{SR}), the distance and orientation offset to a reference path (J_{LC} and J_O), and the velocity offset (J_V). The chosen weights are

$$[(A|50), (SA|50), (SR|50), (LC|1), (V|20), (O|50)].$$

Since the ego vehicle should perform a lane-change to the left lane, the reference path is set to the center of the goal lane for computing the costs J_{LC} and J_O. Furthermore, the optimization horizon is 5.5 s and the desired velocity is $v_{\mathrm{des}} = 25$ m/s.

The unique ID of the benchmark is B = M-KS1:SM1:NGSIM_US101_0, with $v_S \to \infty$. Our obtained trajectory has a total cost of $J_{SM1}(x(t), u(t), t_0, t_f) = 5.69 \cdot 10^4$. In contrast to other work, all details on the vehicle model, the cost function, and the scenario are precisely given by our unique ID. Please note that without the ID we also would not have had the space to present all the details of the scenario in this work, although it is quite simple.

VII. Conclusions

To the best of our knowledge, we provide the first set of composable benchmark problems for motion planning on roads accessible from **commonroad.in.tum.de**. While this paper only provides a rough overview, all details can be found in the provided documentation on our website. Each composed benchmark has a unique ID that can be used in publications or for one's own organization of benchmarks. This is demonstrated by an example for which we also provide a solution. Our benchmark collection contains a mix of recorded and constructed scenarios as well as scenarios on highways, on rural roads, and in urban settings. Our platform-independent repository can be extended by other researchers and will also be extended by ourselves.

References

[1] F. Amigoni, M. Reggiani, and V. Schiaffonati, "An insightful comparison between experiments in mobile robotics and in science," *Autonomous Robots*, vol. 27, pp. 313–325, 2009.

[2] F. Bonsignorio and A. P. del Pobil, "Toward replicable and measurable robotics research," *IEEE Robotics & Automation Magazine*, vol. 22, no. 3, pp. 32–35, 2015.

[3] J. Baltes, "A benchmark suite for mobile robots," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2000, pp. 1101–1106.

[4] R. Dillmann. (2004) Benchmarks for robotics research. Deliverable KA 1.10 of the EU project EURON.

[5] K. Pfeiffer, A. Bubeck, N. Blümlein, and M. Hägele. (2007) Action plan and recommendation on benchmarks for mobile manipulation and service robots. Deliverable D4.3 of the EU project Robot Standards and Reference Architectures (RoSta).

[6] W. Nowak, A. Zakharov, S. Blumenthal, and E. Prassler. (2010) Benchmarks for mobile manipulation and robust obstacle avoidance and navigation. Deliverable D3.1 of EU Project Best Practice in Robotics (BRICS).

[7] C. Goldfeder, M. Ciocarlie, H. Dang, and P. K. Allen, "The Columbia Grasp Database," in *Proc. of IEEE International Conference on Robotics and Automation*, 2009, pp. 1710–1716.

[8] S. Ulbrich, D. Kappler, T. Asfour, N. Vahrenkamp, A. Bierbaum, M. Przybylski, and R. Dillmann, "The OpenGRASP benchmarking suite: An environment for the comparative analysis of grasping and dexterous manipulation," in *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2011, pp. 1761–1767.

[9] B. Cohen, I. A. Şucan, and S. Chitta, "A generic infrastructure for benchmarking motion planners," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012, pp. 589–595.

[10] V. G. Kovvali, V. Alexiadis, and L. Zhang, "Video-based vehicle trajectory data collection," in *Proc. of the Transportation Research Board 86th Annual Meeting*, 2007.

[11] V. Punzo, M. T. Borzacchiello, and B. Ciuffo, "On the assessment of vehicle trajectory data accuracy and application to the Next Generation SIMulation (NGSIM) program data," *Transportation Research Part C*, vol. 19, pp. 1243–1262, 2011.

[12] E. Romera, L. M. Bergasa, and R. Arroyo, "Need data for driver behaviour analysis? Presenting the public UAH-DriveSet," in *Proc. of the 19th IEEE International Conference on Intelligent Transportation Systems*, 2016, pp. 387–392.

[13] W. Hu, X. Xiao, Z. Fu, D. Xie, T. Tan, and S. Maybank, "A system for learning statistical motion patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, pp. 1450–1464, 2006.

[14] B. T. Morris and M. M. Trivedi, "Learning, modeling, and classification of vehicle track patterns from live video," *IEEE Transactions on Intelligent Transportation Systems*, vol. 9, pp. 425–437, 2008.

[15] Y. Zheng, J. Wang, X. Li, C. Yu, K. Kodaka, and K. Li, "Driving risk assessment using cluster analysis based on naturalistic driving data," in *Proc. of the 17th IEEE International Conference on Intelligent Transportation Systems*, 2014, pp. 2584–2589.

[16] I. A. Şucan, M. Moll, and L. E. Kavraki, "The open motion planning library," *IEEE Robotics & Automation Magazine*, vol. 19, no. 4, pp. 72–82, 2012.

[17] S. Chitta, I. Sucan, and S. Cousins, "MoveIt!" *IEEE Robotics & Automation Magazine*, vol. 19, no. 1, pp. 18–19, 2012.

[18] M. Quigley, K. Conley, B. P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "ROS: an open-source Robot Operating System," in *ICRA Workshop on Open Source Software*, 2009.

[19] M. Moll, I. A. Şucan, and L. E. Kavraki, "Benchmarking motion planning algorithms: An extensible infrastructure for analysis and visualization," *IEEE Robotics & Automation Magazine*, vol. 22, no. 3, pp. 96–102, 2015.

[20] E. Plaku, K. E. Bekris, and L. E. Kavraki, "OOPS for motion planning: An online, open-source, programming system," in *IEEE International Conference on Robotics and Automation*, 2007, pp. 3711–3716.

[21] R. Diankov and J. Kuffner, "OpenRAVE: A planning architecture for autonomous robotics," Carnegie Mellon University, Tech. Rep. CMU-RI-TR-08-34, 2008.

[22] E. Frazzoli, M. A. Dahleh, and E. Feron, "Maneuver-based motion planning for nonlinear systems with symmetries," *IEEE Transactions on Robotics*, vol. 21, no. 6, pp. 1077–1091, 2005.

[23] N. Ratliff, M. Zucker, J. A. Bagnell, and S. Srinivasa, "CHOMP: Gradient optimization techniques for efficient motion planning," in *Proc. of IEEE International Conference on Robotics and Automation*, 2009, pp. 489–494.

[24] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.

[25] A. Rizaldi and M. Althoff, "Formalising traffic rules for accountability of autonomous vehicles," in *Proc. of the 18th IEEE International Conference on Intelligent Transportation Systems*, 2015, pp. 1658–1665.

[26] B. Paden, M. Čáp, 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.

[27] S. E. Shladover, C. A. Desoer, J. K. Hedrick, M. Tomizuka, J. Walrand, W.-B. Zhang, D. H. McMahon, H. Peng, S. Sheikholeslam, and N. McKeown, "Automated vehicle control developments in the PATH program," *IEEE Transactions on Vehicular Technology*, vol. 40, no. 1, pp. 114–130, 1991.

[28] D. Kim, H. Peng, S. Bai, and J. M. Maguire, "Control of integrated powertrain with electronic throttle and automatic transmission," *IEEE Transactions on Control Systems Technology*, vol. 15, no. 3, pp. 474–482, 2007.

[29] D. Odenthal, T. Bünte, and J. Ackermann, "Nonlinear steering and braking control for vehicle rollover avoidance," in *Proc. of the European Control Conference*, 1999, pp. 598–603.

[30] P. Gaspar, I. Szaszi, and J. Bokor, "Reconfigurable control structure to prevent the rollover of heavy vehicles," *Control Engineering Practice*, vol. 13, pp. 699–711, 2005.

[31] J. A. Reeds and L. A. Shepp, "Optimal paths for a car that goes both forwards and backwards," *Pacific Journal of Mathematics*, vol. 145, no. 2, pp. 367–393, 1990.

[32] J.-B. Tomas-Gabarron, E. Egea-Lopez, and J. Garcia-Haro, "Vehicular trajectory optimization for cooperative collision avoidance at high speeds," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 4, pp. 1930–1941, 2013.

[33] D. N. Godbole, V. Hagenmeyer, R. Sengupta, and D. Swaroop, "Design of emergency maneuvers for automated highway system: Obstacle avoidance problem," in *Proc. of the 36th Conference on Decision & Control*, 1997, pp. 4774–4779.

[34] R. Rajamani, *Vehicle Dynamics and Control*. Springer, 2012.

[35] S. Petti and T. Fraichard, "Safe motion planning in dynamic environments," in *Proc. of the Conference on Intelligent Robots and Systems*, 2005.

[36] M. Althoff and S. Magdici, "Set-based prediction of traffic participants on arbitrary road networks," *IEEE Transactions on Intelligent Vehicles*, vol. 1, no. 2, pp. 187–202, 2016.

[37] J. H. Jeon, S. Karaman, and E. Frazzoli, "Anytime computation of time-optimal off-road vehicle maneuvers using the RRT*," in *Proc. of the 50th IEEE Conference on Decision and Control and European Control Conference*, 2011, pp. 3276–3282.

[38] Z. Shiller and Y.-R. Gwo, "Dynamic motion planning of autonomous vehicles," *IEEE Transactions on Robotics and Automation*, vol. 7, no. 2, pp. 241–249, 1991.

[39] E. Bertolazzi, F. Biral, and M. Da Lio, "Real-time motion planning for multibody systems," *Multibody System Dynamics*, vol. 17, no. 2, pp. 119–139, 2007.

[40] R. W. Allen, H. T. Szostak, D. H. Klyde, T. J. Rosenthal, and K. J. Owens, "Vehicle dynamic stability and rollover," U.S. Department of Transportation, Final Report DOT HS 807 956, 1992.

[41] *Adams/Tire help*, MSC Software, 2 MacArthur Place, Santa Ana, CA 92707, April 2011, documentation ID: DOC9805. [Online]. Available: http://simcompanion.mscsoftware.com/infocenter

[42] J. E. Bobrow, "Optimal robot path planning using the minimum-time criterion," *IEEE Journal of Robotics and Automation*, vol. 4, no. 4, pp. 443–450, 1988.

[43] J. Ziegler, P. Bender, T. Dang, and C. Stiller, "Trajectory planning for BERTHA – a local, continuous method," in *Proc. of the IEEE Intelligent Vehicles Symposium*, 2014, pp. 450–457.

[44] M. Werling, J. Ziegler, S. Kammel, and S. Thrun, "Optimal trajectory generation for dynamic street scenarios in a frenét frame," in *Proc. of the IEEE International Conference on Robotics and Automation*, 2010, pp. 987–993.

[45] S. Magdici and M. Althoff, "Fail-safe motion planning of autonomous vehicles," in *Proc. of the 19th International IEEE Conference on Intelligent Transportation Systems*, 2016, pp. 452–458.

[46] W. Xu, J. Wei, J. M. Dolan, H. Zhao, and H. Zha, "A real-time motion planner with trajectory optimization for autonomous vehicles," in *Proc. of the IEEE International Conference on Robotics and Automation*, 2012, pp. 2061–2067.

[47] J. Suh and S. Oh, "A cost-aware path planning algorithm for mobile robots," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012, pp. 4724–4729.

[48] S. J. Anderson and S. C. Peters, "An optimal-control-based framework for trajectory planning, threat assessment, and semi-autonomous control of passenger vehicles in hazard avoidance scenarios," *International Journal of Vehicle Autonomous Systems*, vol. 8, pp. 190–216, 2010.

[49] P. Bender, J. Ziegler, and C. Stiller, "Lanelets: Efficient map representation for autonomous driving," in *Proc. of the IEEE Intelligent Vehicles Symposium*, 2014, pp. 420–425.

[50] J. Ziegler and C. Stiller, "Fast collision checking for intelligent vehicle motion planning," in *Proc. of the IEEE Intelligent Vehicles Symposium*, 2010, pp. 518–522.

[51] M. Koschi and M. Althoff, "SPOT: A tool for set-based prediction of traffic participants," in *Proc. of the IEEE Intelligent Vehicles Symposium*, 2017.

[52] A. Lambert, D. Gruyer, G. S. Pierre, and A. N. Ndjeng, "Collision probability assessment for speed control," in *Proc. of the 11th International IEEE Conference on Intelligent Transportation Systems*, 2008, pp. 1043–1048.

[53] M. Althoff, O. Stursberg, and M. Buss, "Model-based probabilistic collision detection in autonomous driving," *IEEE Transactions on Intelligent Transportation Systems*, vol. 10, no. 2, pp. 299 – 310, 2009.