Safe, Learning-Based MPC for Highway Driving under Lane-Change Uncertainty: A Distributionally Robust Approach[⋆,⋆⋆]

Mathijs Schuurmans[a,*], Alexander Katriniok[b], Christopher Meissen[d], H. Eric Tseng[c], Panagiotis Patrinos[a]

[a]*Department of Electrical Engineering (ESAT-STADIUS), KU Leuven, Kasteelpark Arenberg 10, Leuven, 3001, Belgium*
[b]*Ford Research & Innovation Center, Aachen, 52072, Germany*
[c]*Ford Research & Innovation Center, Dearborn, 48124, Michigan, USA*
[d]*Ford Greenfield Labs, Palo Alto, 52072, California, USA*

Abstract

We present a case study applying learning-based distributionally robust model predictive control to highway motion planning under stochastic uncertainty of the lane change behavior of surrounding road users. The dynamics of road users are modelled using Markov jump systems, in which the switching variable describes the desired lane of the vehicle under consideration and the continuous state describes the pose and velocity of the vehicles. We assume the switching probabilities of the underlying Markov chain to be unknown. As the vehicle is observed and thus, samples from the Markov chain are drawn, the transition probabilities are estimated along with an ambiguity set which accounts for misestimations of these probabilities. Correspondingly, a distributionally robust optimal control problem is formulated over a scenario tree, and solved in receding horizon. As a result, a motion planning procedure is obtained which through observation of the target vehicle gradually becomes less conservative while avoiding overconfidence in estimates obtained from small sample sizes. We present an extensive numerical case study, comparing the effects of several different design aspects on the controller performance and safety.

Keywords: Model predictive control, Risk measures, Distributionally robust optimization, Automated driving, Path planning

⋆This work was supported by the Ford KU Leuven Research Alliance Project (KUL00024, KUL0075). The work of P. Patrinos was supported by KU Leuven internal funding: C14/18/068; Fonds de la Recherche Scientifique (FNRS) and Fonds Wetenschappelijk Onderzoek (FWO) Vlaanderen under EOS Project no G0F6718N (SeLMA); FWO: G0A0920N, G086318N, G086518N; European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 953348.

⋆⋆Supplementary videos are available online [1].

*Corresponding author
Email address: mathijs.schuurmans@kuleuven.be (Mathijs Schuurmans)

2010 MSC: 93E20, 93C10, 93E35, 91G70

1. Introduction

The prospect of significantly increasing safety and efficiency of traffic have motivated considerable research efforts towards increasingly advanced driving assistance systems and even (fully) autonomous driving. Due to their strong theoretical foundations and the ability to naturally include safety constraints in the control design, systems based on model predictive control (MPC) provide promising avenues towards this goal. However, one of the key challenges in planning and following safe trajectories among neighboring traffic participants is the large amounts of uncertainty present in forecasts of the traffic state in the near future [2].

This central challenge has led to the growing popularity of stochastic MPC formulations using models obtained through techniques from machine learning (ML) and statistics [3]. However, among all domains where ML techniques provide great promise, it is clear that autonomous driving is one where interpretability, safety and reliability of the employed models are of vital importance. This is a concern with many commonly used ML models, in particular at smaller sample sizes. Indeed, it is well-known that sample-average approximations for minimizing the expected value of random cost functions under unknown distributions often leads to poor out-of-sample performance. This approach, which is closely related to overfitting in machine learning, is known in the optimization and operations research community as the optimizer's curse [4, 5].

Motivated by these concerns, robust approaches are traditionally considered as a safe alternative [6, 7]. Here, one would replace a probability distribution over realizations of an uncertain influence on the system by the worst-case realization over a prediction horizon. As such, only the knowledge of the support of the disturbance distribution (i.e., the uncertainty set) is required to provide *a priori* safety guarantees. The obvious drawback of such formulations is that, in practice, requiring robustness to all possible realization may be too strict. The resulting control formulation may be overly conservative, and therefore lead to severe loss of driving efficiency or infeasibility (situations in which no action or policy satisfying the safety constraints exists). Furthermore, empirical evidence suggests that overly cautious behavior may even reduce safety [8].

In order to capture the benefits of both approaches, while mitigating their respective drawbacks, we propose a distributionally robust control methodology [9, 10]. To do so, we consider a stochastic model of the joint dynamics of the agents in the immediate traffic environment assuming that the underlying distribution is unknown. In every time step, the proposed controller receives a sample from this distribution through observing the current state of the system. Using these gathered samples, the parameters of a stochastic model are estimated, along with statistical bounds on the possible misestimation of these parameters. Combining these two quantities yields a set of statistically plausible distributions over the stochastic process in the next time step—these are

commonly referred to as *ambiguity sets*.

The controller then plans a trajectory assuming the worst-case probability distribution within this ambiguity set, resulting in a formulation equivalent to a risk-averse optimal control problem (OCP) [11, 12]. Initially, all ambiguity sets are expected to contain (almost) all distributions (leading to a robust OCP), whereas over time, the uncertainty on the estimated distributions decreases and accordingly, the conservatism of the corresponding control decisions is systematically reduced.

In this work, we focus on the task of highway path planning, considering stochastic behavior of the surrounding vehicles (target vehicles). We model the target vehicle states as autonomous Markov jump systems; At every time step, the driver of the target vehicle makes a choice of semantically meaningful driving behaviors, which we refer to as the *mode* of the system. In particular, each mode could correspond to a desired lane of the vehicle. Given the (randomly) selected mode, the continuous state of the vehicle is updated using a kinematic vehicle model. Repeating this over multiple time-steps yields a stochastic, yet dynamically feasible state trajectory.

The choice of this model class is motivated by the typical multimodal nature of distributions of vehicle trajectories on highways [13]. Indeed, the majority of lateral position trajectories will be concentrated around the lane center lines, whereas only a small minority of trajectories will run along the separating lines between the lanes. By contrast, many popular models for representing stochasticity in control systems, such as Gaussian processes, will fail to represent this bimodality.

Furthermore, under this model, the resulting optimal control problem can be represented on a scenario tree, which allows us to optimize over all causal closed-loop policies without requiring a parametric approximation, reducing conservatism with respect to alternative approaches, such as tube-based formulations or affine state-feedback parametrization [14, 15].

1.1. Related work

Within the field of automated driving and advanced driver assistance systems, the same class of switching models was used in earlier work [16, 17] for an adaptive cruise control (ACC) setting. In [17], this model is employed in a stochastic MPC framework. In [16], this was extended to a distributionally robust setting, where the transition probabilities were no longer required to be known *a priori*. In this work, we subsequently extend the work of [16] to a full path planning use case, which includes both longitudinal and lateral control as well as possibly nonlinear vehicle dynamics and (inherently nonconvex) collision avoidance constraints.

To ensure safety of the generated trajectories, we impose collision avoidance constraints in the form of conditional *chance constraints*. For numerical reasons, however, we replace chance constraints by risk constraints [18, §3.1.1]. Risk measures been used for collision avoidance in the broader context of autonomous navigation under uncertainty [19, 20, 21, 22, 23, 24, 25, 26, 15, 27].

In [20, 21, 22, 25, 26, 27], risk measures are constructed with respect to some known reference distribution, treating the risk level as a tuning parameter. In contrast to these approaches, we adopt a *distributionally robust* approach, in which risk measures are implicitly defined through statistical ambiguity sets (see §3 for more details). As a result, risk parameters are automatically calibrated based on the sample size of the observed dataset [28].

This distributionally robust perspective is shared with [24, 15, 29, 30], which approximate chance constraints using concentration inequalities. Although these approaches are similar in spirit, they differ significantly in the remaining modeling choices and solution methodologies. For instance, contrary to our scenario tree formulation, which models discrete probabilistic choices, the cited works consider the uncertainty to originate from additive disturbances [15, 31], or from the uncertainty of a Gaussian process model of the environment dynamics [24]. These modeling choices in turn dictate the use of different classes of ambiguity sets (e.g., moment-based ambiguity sets [15], Wasserstein balls [24]), and corresponding optimal control formulations and solution methods.

On the other hand, in the model of [30], qualitative (discrete) driver actions can be taken into account, as they allow multimodal mixture models to describe the distributions of future target vehicle states. However, in contrast to our work, it is assumed that the distributions are known (i.e., provided by some given learning system). The purpose of the concentration inequalities in [30] – which depend on these distributions – is therefore to ease computations, rather than to ensure robustness against ambiguity. Furthermore, several techniques to predict future trajectories of surrounding vehicles, based on continuous/discrete hybrid models, e.g., based on Hidden Markov Models [32] or Long Short-Term Memory (LSTM) models [33, 34] have been proposed in the literature. However, it is very difficult to derive practically useful and statistically valid concentration bounds on the obtained probability distributions for the purpose of constructing ambiguity sets. Furthermore, making predictions of future states using e.g., the method of [33] requires performing inference on the LSTM model. By consequence, trajectory prediction and planning must typically be carried out separately, since otherwise, solving the optimal control problem involves running inference of the LSTM, including its gradient at every iteration. By contrast, with the model class used in this work, it is easy to introduce dependence of the target vehicle policy on the ego vehicle states, which affords additional modeling freedom for the purpose of integration into an MPC scheme. We provide an example of this in Section 4.3.1.

1.2. Contributions and organization

We now summarize the main contributions of this work:

1. We propose a simple, interpretable and extendable model for the behavior of target vehicles in highway scenarios. This model takes the form of a Markov jump system, which naturally models both the uncertainty inherent to the tactical driving decisions of the driver (represented by

the discrete modes) and the more predictable nature of vehicle kinematics. Compared to existing, more sophisticated discrete/continuous hybrid models from the literature (e.g., [33, 34]), these have the benefit of being easy to encorporate into an MPC scheme, and allowing for dependence of their dynamics on the ego vehicle state.

2. We present a distributionally robust MPC scheme for closed-loop path planning under probabilistic collision avoidance constraints. The true underlying distributions are assumed to be unknown. As the target vehicle behavior is observed online, the nominal transition probabilities are estimated, along with an *ambiguity set* which models the statistical uncertainty on the estimated distribution.

3. We carry out an extensive numerical case study, illustrating the effects of the involved design parameters on the behavior of the control system. This includes a comparison of several formulations of the collision avoidance constraints.

The remainder of the article is organized as follows: In Section 2, we describe the set-up of the case study. We describe the used vehicle models, and introduce several possible formulations describing collision-avoidance constraints. In Section 3, we briefly review some technical preliminaries and introduce the MPC design. Section 4 presents an extensive numerical case study of the resulting control scheme. Finally, in Section 5, we discuss the most salient conclusions and suggest possible extensions and improvements for future work.

Notation

Let $\mathbb{R}$ and $\mathbb{N}$ denote the set of real and natural numbers, respectively. $\mathbb{R}_+$ ($\mathbb{R}_-$) denotes the set of positive (negative) real numbers. I_n denotes the $n \times n$ identity matrix. Given $a, b \in \mathbb{N}, a \leq b$, we write $\mathbb{N}_{[a,b]} := \{n \in \mathbb{N} \mid a \leq n \leq b\}$. We use $\mathbf{1}_d$ to denote a vector in $\mathbb{R}^d$ with all elements equal to 1. For some $i \in \mathbb{N}_{[1,d]}$, we denote by $\mathbf{e}_i \in \mathbb{R}^d$ the i'th standard basis vector, taking the value 1 in its i'th components and 0 everywhere else. Given a matrix P, we denote its ith row by $P_{i:}$. We denote by $[x]_+$ the positive part of x, i.e., $[x]_+ = \mathbf{max}\{0, x\}$, where $\mathbf{max}$ is evaluated element-wise.

2. Modelling for highway motion planning

The following section describes the set-up and the models used in our case study. This involves the dynamics of the host (or ego) vehicle, of *target vehicles* (i.e., surrounding vehicles) and the considered formulations for the obstacle avoidance constraints. Overall, we will consider driving scenarios on a multi-lane highway. For simplicity, we consider straight road segments. As our final controller does not rely on convexity or linearity of the involved dynamics, the set-up can be straightforwardly extended to more general curved roads by transforming all involved coordinates to a Frenét frame [35].

We will denote the state of the i'th vehicle in the scene by $x^{(i)}$. We will assume the convention that vehicle 0 is the ego vehicle. The augmented state vector obtained by combining the states of both the ego vehicle and the target vehicles will be denoted as $\boldsymbol{x} = \begin{bmatrix} x^{(0)} & \dots & x^{(n)} \end{bmatrix}^\top \in \mathbb{R}^{n_x}$. This allows us to compactly write the dynamics of the full concatenated system as

$$\boldsymbol{x}_{t+1} = f(\boldsymbol{x}_t, u_t, \theta_{t+1}), \tag{1}$$

where $u_t \in \mathbb{R}^{n_u}$ denotes the control actions applied to the ego vehicle at time t and $\{\theta_t\}_{t\in\mathbb{N}}$ is a stochastic process, representing the uncertain behavior of the surrounding vehicles. We will assume that θ_t takes values on a finite set $W := \{1, \dots, d\}$ and is governed by a time-homogeneous Markov chain with *unknown* transition probability matrix P. In Section 3.1, we describe how we use online observations to obtain estimates of P, while remaining aware of potential estimation errors. In the next sections (§2.1 and 2.2), we provide example implementations for f, which we have used for our numerical experiments. However, we do emphasize that many other modeling choices for f could be made without any modification to the remainder of the control formulation, which is described in Section 3.

2.1. Host vehicle Dynamics

The host vehicle dynamics are considered to be deterministic and known. We utilize the well-known kinematic bicycle model [36], which is governed by continuous-time dynamics of the form

$$\dot{p_x} = v\,\mathbf{cos}(\psi + \beta) \tag{2a}$$
$$\dot{p_y} = v\,\mathbf{sin}(\psi + \beta) \tag{2b}$$
$$\dot{\psi} = \tfrac{v}{l_\mathrm{r}}\,\mathbf{sin}(\beta) \tag{2c}$$
$$\dot{v} = a \tag{2d}$$
$$\beta = \mathbf{tan}^{-1}\left(\tfrac{l_\mathrm{r}}{l_\mathrm{f}+l_\mathrm{r}}\,\mathbf{tan}\,\delta\right) \tag{2e}$$

with state vector $x = \begin{bmatrix} p_x & p_y & \psi & v \end{bmatrix}^\top$ (longitudinal and lateral position, heading angle and longitudinal velocity) and input vector $u = \begin{bmatrix} a & \delta \end{bmatrix}^\top$ (longitudinal acceleration and steering angle). In our experiments, we discretize these equations using a simple forward Euler scheme, although of course, more sophisticated integration schemes could readily be used for our control scheme, if additional accuracy were required. For simplicity, we will set $l_\mathrm{r} = l_\mathrm{f} = \frac{l}{2}$.

2.2. Target vehicle behavior

We model the target vehicle behavior as an autonomous Markov jump system, which includes a separate driving mode corresponding to each lane on the road. Within each mode, the dynamics are governed by an autonomous system that drives the lateral component of the vehicle position to the centerline of the lane corresponding to the current mode and the longitudinal component

to a preferred velocity (given for each lane). Additionally, it is possible for the policy to take into account relative positions and velocities with respect to other vehicles. The state vector $x^{(i)} = \left[\, p_x^{(i)} \; p_y^{(i)} \; v_x^{(i)} \; v_y^{(i)} \,\right]^\top$ of vehicle i consists of the position and velocity in the (x, y)-plane. For simplicity, the longitudinal and lateral dynamics are decoupled for these vehicles into two double-integrator systems, leading to dynamics of the form

$$x_{t+1}^{(i)} = Ax_t^{(i)} + B\pi^{(i)}\left(\boldsymbol{x}_t, \theta_{t+1}\right), \text{ with } A = \begin{bmatrix} 1 & T_\mathrm{s} & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T_\mathrm{s} \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 0 \\ T_\mathrm{s} & 0 \\ 0 & 0 \\ 0 & T_\mathrm{s} \end{bmatrix}, \tag{3}$$

and $\pi^{(i)} : \mathbb{R}^{n_x} \times W \to \mathbb{R}^2$ is the mode-dependent control policy for target vehicle i. Unless otherwise specified, we will assume that the policy is given as,

$$\pi^{(i)}(\boldsymbol{x}, \theta) = \begin{bmatrix} 0 & 0 & k_{v,x} & 0 \\ 0 & -k_y & 0 & -k_{v,y} \end{bmatrix} x^{(i)} + \begin{bmatrix} k_{v,x} v_{\mathrm{ref},\theta} \\ k_y y_{\mathrm{lane},\theta} \end{bmatrix}. \tag{4}$$

See Table 1 for a description of the involved parameters. Note that under (4), the target vehicle dynamics are described as an autonomous Markov jump linear system (MJLS) [37]. However, we emphasize that the presented control framework allows for sophisticated models for the target vehicle behavior, as we illustrate in §4.3.1.

The decoupling of the nonlinear vehicle dynamics into longitudinal and lateral motion is a common modeling choice for highway driving applications (e.g., [38, 39, 40]). It is motivated by the fact that for highway traffic, (i) vehicles tend to be closely aligned with the direction of the road, so that relative heading angles are close to zero; and (ii) velocities tend to vary slowly around the reference speed. Therefore, a linearization of the nonlinear kinematic bicycle model around the desired velocity and heading angle $\psi = 0$, leading to (3) is typically quite accurate.

We assume here that the parameters involved in (3) are given. In practice, data-driven estimates of these parameters would need to be obtained. However, this is beyond the scope of the current paper and considered for future work. We assume that at each time t, the mode θ_t can be measured, for example by means of the indicator lights of the target vehicle, its lateral motion, or a combination of both. See, e.g., [41] for a procedure to determine mode observations from the continuous state.

2.3. Collision avoidance formulations

In order to ensure safety of the obtained trajectories, we impose collision avoidance constraints in the optimal control problem. Suppose we are given the augmented state $\boldsymbol{x} \in \mathbb{R}^{n_x}$ of the vehicles in the scene. Then, the obstacle avoidance constraint will be of the form

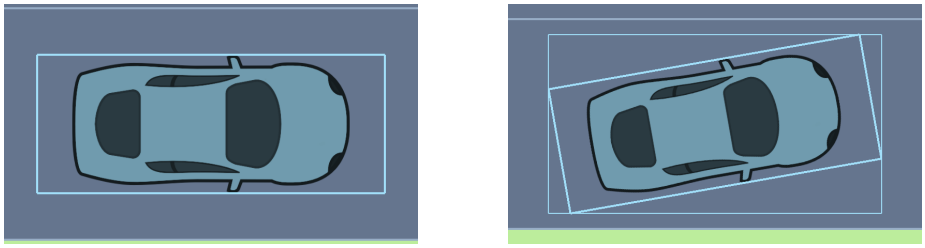
$$h(\boldsymbol{x}) \leq 0, \tag{5}$$

where, in the case of a single pair of vehicles, the function $h : \mathbb{R}^{n_x} \to \mathbb{R}$ represents a negative distance between the vehicles. Below, we propose several

Table 1: Symbols used in modeling the lane change scenario.

Parameter	Description
$k_y, k_{v,x}, k_{v,y}$	Internal control gains used to model target vehicle behavior
v_{ref}	Reference velocity
$y_{\text{lane},i}$	Lateral position of the lane center of lane i
T_{s}	Sampling rate
l_{r}, l_{f}	Longitudinal distance from center of gravity and rear/front wheels
l, w	length ($l = l_r + l_f$) and width of a vehicle

alternatives for the choice of this function h, assuming a single "obstacle" vehicle. In the case where more vehicles are present in the scene, the pointwise maximum of the resulting functions can be used. It is important to emphasize at this point that in the optimal control problem, the future states cannot be exactly predicted. Therefore, a probabilistic counterpart to (5) is enforced in practice (see (23)). This is discussed in more detail in Section 3. For now, however, we focus on the possible instances of the function h.



(a) Heading angle $\psi = 0$.

(b) Heading angle $\psi \neq 0$.

Figure 1: Bounding boxes around a vehicle. For a rotated vehicle, an axis-aligned bounding box induces some additional conservatism.

For the purpose of collision avoidance, we model vehicles using rectangular bounding boxes, as illustrated in Fig. 1. The bounding boxes can either be aligned with the longitudinal axis of the road or with the longitudinal axis of the vehicle. The former will typically lead to simplified collision avoidance constraints at the cost of some additional conservatism which scales with $|\mathbf{sin}\,\psi|$. Since $\mathbf{sin}\,\psi$ is typically a small value in highway situations, this is unlikely to cause significant losses in performance.

2.3.1. Axis-aligned bounding boxes

Computation of axis-aligned bounding boxes. An axis-aligned rectangle can be represented as the set

$$\mathcal{B} = \{p \in \mathbb{R}^2 \mid \underline{p} \leq p \leq \overline{p}\},$$

where the inequalities are taken element-wise. If we are given a vehicle with length l, width w and state $\begin{bmatrix} p_x & p_y & \psi & v \end{bmatrix}^\top$, then $\Delta_y = \frac{l}{2}|\sin\psi| + \frac{w}{2}|\cos\psi|$, $\Delta_x = \frac{l}{2}|\cos\psi| + \frac{w}{2}|\sin\psi|$ and

$$\overline{p} = \begin{bmatrix} p_x+\Delta_x \\ p_y+\Delta_y \end{bmatrix}, \quad \underline{p} = \begin{bmatrix} p_x-\Delta_x \\ p_y-\Delta_y \end{bmatrix}. \tag{6}$$

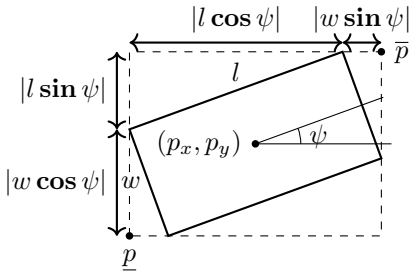


Figure 2: Axis-aligned bounding box dimensions

Due to the absolute values involved, these quantities are nonsmooth functions of the state, which is undesirable for the purposes of numerical optimization. Assuming that each vehicle respects the driving direction of the road, we have that $\cos\psi > 0$, so that we may omit the absolute value. For $\sin\psi$, however, this is not the case, hence we will introduce an approximation.

In particular, we will overapproximate $\phi : x \mapsto |x|$ by slightly modifying the result in [42]:

$$\tilde{\phi}_k(x) = b_k + x\tanh(kx), \quad b_k = \frac{W_{\mathrm{L}}(1)}{k}\left(1 - \tanh(W_{\mathrm{L}}(1))\right) \tag{7}$$

where k is a parameter controlling the smoothness as illustrated in Fig. 3 and W_{L} denotes the Lambert W-function [43]. The higher k, the better the approximation, yet the larger the second derivatives around 0. In all experiments below, a value of $k = 20$ is used. Replacing now $|\sin\psi|$ with $\tilde{\phi}_k(\sin\psi)$ in expressions (6) for $\underline{p}$ and $\overline{p}$, these quantities become smooth functions of the augmented state $\boldsymbol{x}$. Note, finally, that b_k is chosen such that $\tilde{\phi}_k(x) \geq |x|$ holds for all $x \in \mathbb{R}$ (see Appendix A), so that the full vehicle is always fully contained within the bounding rectangle, regardless of the approximation quality.

Given two vehicles ($i \in \{0, 1\}$) with axis-aligned bounding boxes of lengths and widths $l^{(i)}, w^{(i)}$, it will be convenient to construct a single bounding box around the vehicle serving as the "obstacle" (by convention, vehicle 1). For a position $p^{(1)} = \begin{bmatrix} p_x^{(1)} & p_y^{(1)} \end{bmatrix}^\top$ of vehicle 1, this bounding box $\mathcal{B}$ is defined as before, with

$$\underline{p} = p^{(1)} - \frac{1}{2}\begin{bmatrix} l^{(0)}+l^{(1)} \\ w^{(0)}+w^{(1)} \end{bmatrix}, \quad \overline{p} = p^{(1)} + \frac{1}{2}\begin{bmatrix} l^{(0)}+l^{(1)} \\ w^{(0)}+w^{(1)} \end{bmatrix}.$$

Collision avoidance constraints now amount to the statement $p^{(0)} \notin \mathcal{B}$. This can be expressed in several ways. We discuss and explicitly compare two alternatives, which we refer to as the *projection* and *ellipse* formulation.

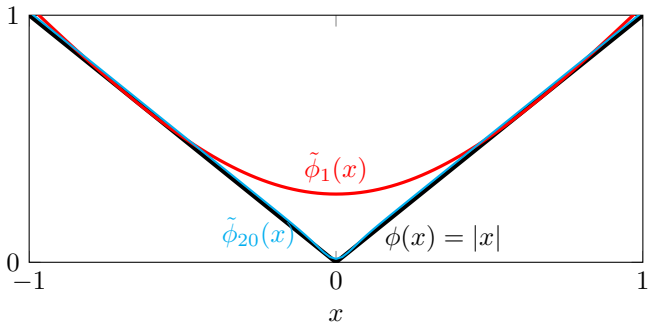


Figure 3: Smooth approximation (7) of $|x|$ for different values of the smoothness parameter k.

Projection formulation. A direct formulation of the aforementioned constraints is simply to express the distance from the current position $p^{(0)}$ to the expanded bounding box $\mathcal{B}$, i.e.,

$$\mathbf{dist}_{\mathcal{B}}(p^{(0)}) = \|p^{(0)} - \mathbf{max}\{\underline{p}, \mathbf{min}\{p^{(0)}, \overline{p}\}\}\|_2. \tag{8}$$

Then the collision avoidance constraint will amount to $\mathbf{dist}_{\mathcal{B}}(p^{(0)}) > 0$. For numerical reasons, however, it is necessary to add some positive constant $\epsilon > 0$ and instead impose (5) with

$$h(\boldsymbol{x}) = h_{\text{proj}}(\boldsymbol{x}) := \epsilon - \mathbf{dist}_{\mathcal{B}}(p^{(0)})^2. \tag{9}$$

Since this squared distance is only once differentiable, it may be numerically beneficial to approximate this formulation by replacing the **min** and **max** operator in this constraint with a smooth function as follows. Note that $\mathbf{max}\{x, y\} = x\mathbf{I}_{x\geq y} + y(1 - \mathbf{I}_{x\geq y})$, where

$$\mathbf{I}_{x\geq y} = \begin{cases} 1 & \text{if } x \geq y \\ 0 & \text{otherwise} \end{cases}$$

is the indicator of the predicate $x \geq y$, which can be approximated using a sigmoid function, resulting in $\tilde{\mathbf{I}}_{x\geq y} = \sigma_k(x-y) = (1+\mathbf{exp}(-k(x-y)))^{-1}$, with $k > 0$ again some smoothness parameter (the larger, the better the approximation, but the worse the conditioning, due to increased curvature). We will refer to this approximated distance formulation as the **sigmoid approximation**. Experimentally (using the IPOPT solver [44]), this has been observed to improve the reliability with which the online optimization problem is solved (see Experiment 4.1).

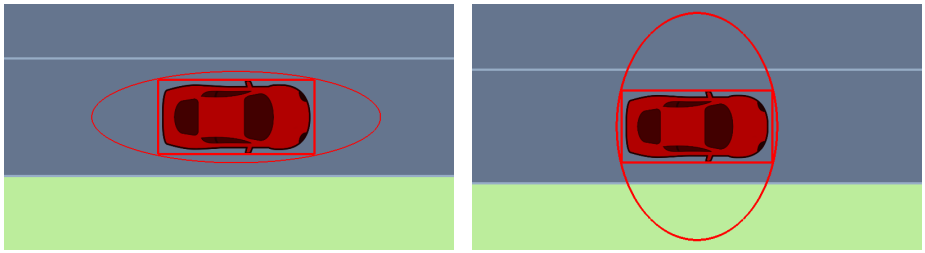
Ellipse formulation. As an approximation of the collision avoidance constraint, we fit an ellipse

$$\mathcal{E} = \left\{ x \in \mathbb{R}^2 \;\middle|\; \left(x - p^{(1)}\right)^\top E \left(x - p^{(1)}\right) \leq 1 \right\}$$

around the bounding box $\mathcal{B}$. Here, $E = \mathbf{diag}(\lambda_1, \lambda_2)$ is a positive definite diagonal matrix with eigenvalues λ_1 and $\lambda_2 = \gamma\lambda_1$, where λ_1 is computed such that the vertices of $\mathcal{B}$ lie on the boundary of $\mathcal{E}$, i.e., $\overline{p}^\top E\overline{p} = 1$, and $\gamma > 0$ is a freely chosen parameter controlling the elongation of the ellipse (see Fig. 4). The resulting collision avoidance constraint then uses the function

$$h(\boldsymbol{x}) = h_{\text{ellipse}}(\boldsymbol{x}) = 1 - (p^{(0)} - p^{(1)})^\top E(p^{(0)} - p^{(1)}). \tag{10}$$

Note that since $\overline{p}$ and $\underline{p}$ are functions of the vehicle heading angle, E is a



(a) Elongated ellipse ($\gamma = 7$) (b) Widened ellipse ($\gamma = 0.5$)

Figure 4: Ellipsoidal sets around a given bounding box $\mathcal{B}$, for different values of the elongation parameter γ.

function of the state $\boldsymbol{x}$. However, for modest heading angles of the vehicle, E could be precomputed for the bounding box that accounts for the largest allowed heading angle, at the cost of only modest conservatism. Nevertheless, since a closed-form expression for the parameters of E exists, their computational cost is negligible, so we compute them online during our experiments, allowing us to account for the instantaneous orientation of the vehicles. In scenarios where tighter maneuvers at larger relative heading angles may be needed, a similar but less conservative formulation can be obtained by modeling a single target vehicle by several ellipses [30].

2.3.2. Arbitrarily oriented bounding boxes

If the bounding boxes of either of the vehicles are not mutually aligned, then the projection requires the following modifications.

Point-wise projection formulation. We define $\mathcal{O} = \{v \mid \underline{p} \leq v \leq \overline{p}\}$, where $\overline{p} = -\underline{p} = 1/2\,[\,l\ w\,]^\top$. Now the vehicle-aligned bounding box in a state $[\,p\ \psi\ v\,]^\top$ is defined as $\mathcal{B} = \{p + R(\psi)v \mid v \in \mathcal{O}\}$, where

$$R(\psi) = \begin{bmatrix} \mathbf{cos}(\psi) & -\mathbf{sin}(\psi) \\ \mathbf{sin}(\psi) & \mathbf{cos}(\psi) \end{bmatrix},$$

as illustrated in Fig. 5. An expression for the distance is then obtained from (8) as follows. If $\mathcal{O}' = \{Rv \mid v \in \mathcal{O}\}$, then for any point $q \in \mathbb{R}^2$,

$$\mathbf{dist}^2_{\mathcal{B}}(q) = \mathbf{dist}^2_{\mathcal{O}'}(q - p).$$

Due to the orthogonality of R:

$$\begin{aligned}\mathbf{dist}^2_{\mathcal{O}'}(q) &= \mathbf{dist}^2_{\mathcal{O}}(R^\top q)\\ \Rightarrow \mathbf{dist}^2_{\mathcal{B}}(q) &= \mathbf{dist}^2_{\mathcal{O}}(R^\top(q-p)).\end{aligned}$$

Thus, the squared distance between a point q and the rectangle $\mathcal{B}$ is

$$\mathbf{dist}^2_{\mathcal{B}}(q) = \|R^\top(q-p) - \mathbf{min}\{\mathbf{max}\{R^\top(q-p), \underline{p}\}, \overline{p}\}\|_2^2.$$

Similarly to [45], we conclude that collision occurs whenever at least one corner of at least one vehicle is at distance zero from the other vehicle. For each target vehicle in the environment, this leads to 8 constraints similar to (5) with functions h of the form (9); one for every corner of every vehicle. The reduction of the approximation error thus comes at the cost of a moderate increase in computational demand of the constraint.

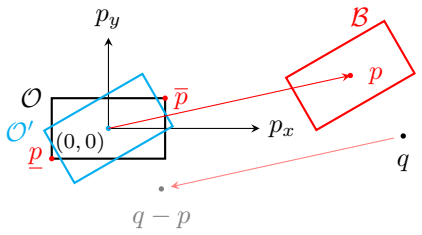


Figure 5: Illustration of the distance to a generally oriented rectangle.

It is worthwhile to note that many other collision constraint formulations exist [46, 47, 48]. In particular, preliminary experiments suggested that the method proposed in [47] performs similarly to the projection formulation. However, both [47] and the well-known signed distance formulation [48] introduce additional auxiliary variables for every node in the scenario tree, inducing a visible computational overhead. For this reason, they are less suitable for planning over scenario trees and we have not included them in our final experimental comparison.

2.4. Physical constraints

We impose several constraints which enforce physically consistent or reasonable behavior and should therefore not be imposed probabilistically. The advantage of such constraints is that they can be imposed directly on the variables residing in the nodes of the scenario tree (see (24) below), without requiring a risk measure and the corresponding risk reformulation described in [12]. This reduces the number of auxiliary variables and constraints introduced in the final optimal control problem. We impose the following constraints in this manner.

Road boundary constraints. We impose that the lateral position of the ego vehicle remains between the road boundaries, i.e., constraints of the form $p_y^{(0)} - \overline{c} \leq 0$ and $-p_y^{(0)} + \underline{c} \leq 0$, where $\overline{c}, \underline{c}$ depend on the width of the road and the lateral dimensions of the axis-aligned bounding box.

Velocity constraints. We enforce that all (longitudinal) velocities remain bounded from above by some positive limit $v_{\max} > 0$, and from below by 0.

For ease of notation, we concatenate these constraints and compactly write them as

$$g_{\text{phys}}(\boldsymbol{x}) \leq 0,$$

where $g_{\text{phys}} : \mathbb{R}^{n_x} \to \mathbb{R}^3$ is an affine function.

2.5. Control objective

We conclude this section by specifying the control objective of the controller. The goal of the controller is to maintain a (given) constant velocity v_{ref}, preferably in the rightmost lane. To this end, we define a desired state x_{ref} and specify the quadratic cost function $\ell : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}_+$ of the form

$$\ell(\boldsymbol{x}, u) := (x^{(0)} - x_{\text{ref}})^\top Q (x^{(0)} - x_{\text{ref}}) + u^\top R u,$$

where $Q \succeq 0$ and $R \succ 0$ (given in Table C.6) are determined to penalize in every time step (i) the deviation of the lateral position of the host vehicle from the right lane center; (ii) the longitudinal velocity of the host vehicle from some target value; and (iii) the total control effort. In other words, we set $x_{\text{ref}} = \begin{bmatrix} 0 & 0 & 0 & v_{\text{ref}} \end{bmatrix}^\top$. The relative weighting was decided primarily to compensate for differences in scale of the different state/input variables, but could of course be tuned more specifically to reflect desired driving behavior. We choose the cost to be a function of the host vehicle state $x^{(0)}$ only, but allow for more general costs as well, allowing to penalize high relative velocities between vehicles in the environment, for instance, which may help reduce congestions.

Finally, we remark that in general, the stage cost may also depend on the discrete mode θ, i.e., we have $\ell : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times W \to \mathbb{R}_+$. This could be used to model changing preferences depending on the maneuver of surrounding vehicles. This poses no theoretical difficulties or additional computational cost. However, as this addition is not particularly meaningful for the case study at hand, we opt to simplify notation and disregard this dependence.

3. Control methodology

We will now formally describe the motion planning formulation. As mentioned before, we formulate the decision-making problem as a distributionally robust optimal control problem [28], taking into account ambiguity in estimated transition probabilities. In the following, recall that $f : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times W \to \mathbb{R}^{n_x}$ defined in (1) denotes the joint (discrete-time) vehicle dynamics.

We will begin by introducing the learning method in Section 3.1. In Section 3.3, we describe how the learning task is embedded in the final control/decision-making problem that is solved online.

3.1. Learning ambiguity sets from data

3.1.1. Independent data

We start by describing the learning problem in the simpler case of i.i.d. data. Let $(W, \mathcal{F}, \mathbb{P})$ denote the probability space defined by the finite sample space $W = \{1, \dots, d\}$, the set of all its subsets $\mathcal{F}$ and probability measure $\mathbb{P}$. A random variable $Z : W \to \mathbb{R}$ on this space can be identified with a vector $z = (z_i)_{i=1}^d \in \mathbb{R}^d$, where $z_i := Z(i)$. Similarly, the distribution over Z can be fully characterized by a probability vector $p \in \Delta_d$ with $p_i = \mathbb{P}\{i\}$, $i \in W$ the atomic probabilities, where $\Delta_d := \{p \in \mathbb{R}^d \mid p \geq 0, \mathbf{1}^\top p_i = 1\}$ denotes the d-dimensional probability simplex. For a stochastic process $(\theta_t)_{t=1}^N$, a similar notation is valid, taking the sample space to be W^N.

For ease of exposition, suppose that $L : \mathbb{R}^n \times W \to \mathbb{R}$ represents some random cost, and we are interested in minimizing its expected value (this will be extended to the multi-stage case in §3.3):

$$\underset{u \in \mathbb{R}^n}{\textbf{minimize}}\ \mathbb{E}_p[L(u, \theta)] = \sum_{i \in W} p_i L(u, i)\,. \tag{11}$$

Of course, solving this problem requires knowledge of the probability distribution p, which we have assumed to be unavailable. If instead, we have access to a sample $\{\widehat{\theta}_k\}_{t=1}^t$, drawn i.i.d. from p, we can compute the empirical estimate $\widehat{p}_t := t^{-1} \sum_{k=1}^t \mathbf{e}_{\theta_k}$ and use this as an approximation for p. However, it is well-known that this approach tends to result in severe underestimations of the true expected costs, in particular for small sample sizes [49, 5]. This phenomenon, which is akin to overfitting, can be mitigated in a systematic manner by additionally constructing a set of probability vectors which account for potential misestimations. Such a set is commonly referred to as an ambiguity set and can be constructed in a variety of manners, each valid for different settings and underlying assumptions on the underlying system and data-generating distributions [50, 49, 51, 52, 16].

In this work, we build ambiguity sets by considering all probability distributions with some distance from the empirical distribution $\widehat{p}_t$, expressed in the total variation (TV) metric. To do so, we leverage the following well-known concentration inequality.

Lemma 3.1 (Total variation bounds [53, Thm. A.6.6])**.** *Let $\{\theta_k\}_{k=1}^t$ denote an i.i.d. sample drawn from a distribution $p \in \Delta_d$, and let $\widehat{p}_t := t^{-1} \sum_{k=1}^t \mathbf{e}_{\theta_k}$ denote the empirical distribution. Then, for any confidence level $\beta \in (0, 1)$,*

$$\mathbb{P}[\|p - \widehat{p}_t\|_1 \leq r_{\beta,t}] \geq 1 - \beta, \ \text{with } r_{\beta,t} = \sqrt{\tfrac{d \log 2 - \log \beta}{t}}. \tag{12}$$

This leads to the ambiguity set

$$\mathcal{A}_t(\beta) = \{\pi \in \Delta_d \mid \|\pi - \widehat{p}_t\|_1 \leq r_{\beta,t}\}\,, \tag{13}$$

which has the desirable property that the true distribution p is captured by $\mathcal{A}_t(\beta)$ with high probability, i.e., $\mathbb{P}[p \in \mathcal{A}_t(\beta)] \geq 1 - \beta$. Therefore, solving the

distributionally robust counterpart of (11)

$$\widehat{L} := \min_{u \in \mathbb{R}^n} \max_{p \in \mathcal{A}_t(\beta)} \mathbb{E}_p[L(u, \theta)], \tag{14}$$

to obtain a minimizer $u_t^\star$, ensures that with high probability,

$$\mathbb{E}_p[L(u_t^\star, \theta)] \leq \widehat{L}. \tag{15}$$

That is, the true, out-of-sample, expected cost will be no larger than the predicted cost. Moreover, observe from (12) that the radius of the ambiguity set reduces at a rate of $\mathcal{O}(t^{-1/2})$, which matches the expected convergence rate of the maximum likelihood estimator $\widehat{p}_t$.

Remark 3.2 (Recursive computation of (13)). We highlight that since $\mathcal{A}_t(\beta)$ is fully described by its center $\widehat{p}_t$ and its radius $r_{\beta,t}$, it can be updated online at a negligible computational cost. Indeed, given $\widehat{p}_t$ and a new data point $\theta_{t+1} \in W$, it is well-known that $\widehat{p}_{t+1}$ can be computed as $\widehat{p}_{t+1} = \frac{t\widehat{p}_t + \mathbf{e}_{\theta_{t+1}}}{t+1}$, and $r_{\beta,t+1}$ is provided in closed form (12).

3.1.2. Markovian data

Let us consider now the case where $(\theta_t)_{t \in \mathbb{N}}$ is a Markov chain, defined on a probability space $(\Omega, \mathcal{F}, \mathbb{P})$, which can be constructed as the product space of the probability space introduced in §3.1.1. We assume that the initial mode is known so that the Markov chain is fully specified by its transition probability matrix $P = (P_{ij})_{i,j \in W} \in \mathbb{R}^{d \times d}$ with $P_{ij} := \mathbb{P}[\theta_k = j \mid \theta_{k-1} = i]$, $\forall k \in \mathbb{N}$. Given a sample sequence $\{\widehat{\theta}_k \in W\}_{k=1}^t$ from the Markov chain, the sets $W_{t,i} := \{\widehat{\theta}_k \mid \widehat{\theta}_{k-1} = i, k \in \mathbb{N}_{[2,t]}\}$ for $i \in W$ define i.i.d. data samples from the distribution $P_{i:}$, i.e., the i'th row of the transition matrix. Using the procedure described in §3.1.1, we can for each mode $i \in W$, construct an ambiguity set $\mathcal{A}_{t,i}(\beta) \subseteq \Delta_d$ defined as

$$\mathcal{A}_{t,i}(\beta) = \{p \in \Delta_d \mid \|p - \widehat{P}_{t,i:}\|_1 \leq r_{\beta,|W_{t,i}|}(\beta)\}, \tag{16}$$

with $\widehat{P}_{t,i:}$ the empirical distribution over the dataset $W_{t,i}$ and $r_{\beta,|W_{t,i}|}$ given by (12), replacing t with the cardinality of $W_{t,i}$. Note that for all $t \in \mathbb{N}$ and $i \in W$, $\mathcal{A}_{t,i}(\beta)$ is a random variable, since it is a function of the data sample $(\widehat{\theta}_k)_{k=1}^t$.

Remark 3.3 (Sample complexity). It is possible to directly use specialized concentration bounds derived for Markov chains (e.g., [54]), instead of applying lemma 3.1 for each individual row. However, without further assumptions on the Markov chain (e.g., known mixing time), these bounds provide no improvement over lemma 3.1 and furthermore, they would yield a single radius for all rows, whereas the proposed approach allows the radii for the rows corresponding to more frequently active modes to decrease more rapidly.

3.2. Distributionally robust and risk-averse optimization

Besides robustifying against estimation errors, the formulation (14) can also be interpreted as replacing the expectation in the cost function of the example problem (11) by a more general operator, called a *risk measure*. Given the

finite sample space W with cardinality $|W| = d$, the outcomes of a random variable $Z : W \to \mathbb{R}$ can be represented by a vector $z \in \mathbb{R}^d$. A risk measure $\rho : \mathbb{R}^d \to \mathbb{R}$ then represents a mapping from an uncertain realization of Z to an *a priori* measure of its value. Trivial examples of risk measures are the expectation (corresponding to a *risk neutral* attitude) and the **max** operation (corresponding to a *robust* or fully *risk-averse* attitude). In particular, the class of *coherent* risk measures are of interest for most problems involving decision-making [55, §6.3]. These risk measures satisfy four additional properties, namely (i) Monotonicity; (ii) Convexity; (iii) Translational invariance; and (iv) Positive homogeneity, which ensure that they exhibit characteristics that one would intuitively expect, e.g., if a random variable X is larger than Y with probability 1, then the risk of X must also be larger than the risk of Y; For more details and intuition regarding these axiomatic properties, see, for instance [56, 57, 58]. Furthermore, these properties endow coherent risk measures with sufficient structure to allow tractable reformulations (e.g., [12]) of optimization problems involving them, despite their typical nonsmooth nature.

Coherent risk measures were originally used to represent the asymmetry in the preferences of decision makers. In most cases, an incurred cost that is larger than the expected value carries more weight than one which is smaller by the same margin. This notion, which originated in finance and operations research, but has recently received increased interest in the control and robotics community [21, 59, 60, 61], considers the selection of the risk measure and its parameters as a degree of freedom for the designer. Alternatively, however, coherent risk measures are intimately related to *distributionally robust* optimization through a well-known result, which we refer to as the *dual risk representation* [55, Thm. 6.4]. This result states that one can associate with each coherent risk measure ρ, a nonempty, closed, and convex subset of the probability simplex (i.e., an ambiguity set) $\mathcal{A}_\rho \subseteq \Delta_d$ such that

$$\rho[z] = \max_{p \in \mathcal{A}_\rho} \mathbb{E}_p[Z] = \max_{p \in \mathcal{A}_\rho} p^\top z.$$

In other words, the risk measure ρ can be written as a worst-case expectation over a set of probability distributions as in (14). We refer to $\mathcal{A}_\rho$ as the ambiguity set *induced by* the risk measure ρ. As a result of this equivalence, (14) inherits the favorable properties and intuitive interpretation of risk-averse optimization, but by virtue of Lemma 3.1, it has the added benefit that the risk measure is calibrated automatically based on the amount of data that is available, resulting in the out-of-sample certificate (15). This in turn allows us to establish theoretical guarantees such as stability and constraint satisfaction of the closed-loop control system described in the next section (see [28] for details).

3.3. Distributionally robust MPC over scenario trees

3.3.1. Scenario tree notation

We now integrate the data-driven ambiguity sets obtained as described in Section 3.1 to a multi-stage predictive control problem. Since W is a finite set, the possible realizations of a mode sequence $\theta_{[0,N]}$ can be enumerated and

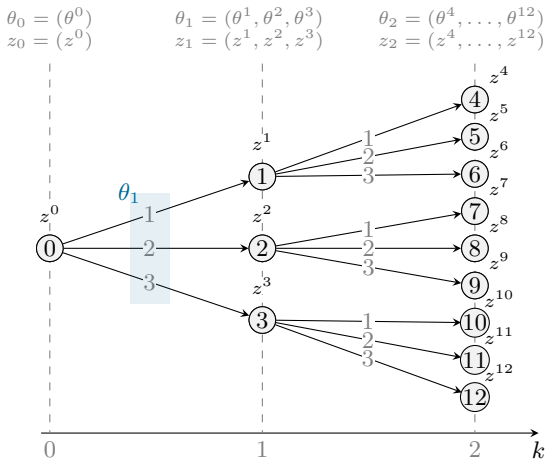


Figure 6: Scenario tree representation of some stochastic process $(z_t)_{t=0}^2$ adapted to the filtration generated by the Markov chain $(\theta_t)_{t=0}^2$, with $W = \{1, 2, 3\}$.

represented on a *scenario tree* [62, 28, 63]. Similarly, any stochastic process (z_t) adapted to the filtration induced by (θ_t) can be represented on the resulting scenario tree. We denote the value of z_t corresponding to a node ι in the tree as z^ι, as illustrated in Fig. 6. The set of nodes in the tree are partitioned into time steps or *stages*. The set of nodes at a stage k is denoted by $\mathbf{nod}(k)$, and similarly, for $k_0, k_1 \in \mathbb{N}_{[0,N]}$, with $k_1 > k_0$, $\mathbf{nod}([k_0, k_1]) = \bigcup_{k=k_0}^{k_1} \mathbf{nod}(k)$. Correspondingly, we have $z_t = (z^\iota)_{\iota \in \mathbf{nod}(t)}$. For a given node $\iota \in \mathbf{nod}(t)$, $t \in \mathbb{N}_{[0,N-1]}$, we call a node $\iota_+ \in \mathbf{nod}(t+1)$ that can be reached from ι in one step a *child* node, denoted $\iota_+ \in \mathbf{ch}(\iota)$. Conversely, we denote the (unique) *ancestor* node of a node $\iota \in \mathbf{nod}(t)$, $t \in \mathbb{N}_{[1,N]}$ by $\mathbf{anc}(\iota) \in \mathbf{nod}(t-1)$. We define an n-step *ancestor* of a node ι recursively by $\mathbf{anc}^n(\iota) := \mathbf{anc}\left(\mathbf{anc}^{n-1}(\iota)\right)$, with $\mathbf{anc}^0(\iota) := \iota$. The nodes $\iota \in \mathbf{nod}(N)$ have no child nodes and are called *leaf nodes*. The unique node at stage 0 has no ancestor and is called the *root node*. Finally, we will slightly abuse notation to write $z^{\mathbf{ch}(\iota)} := (z^j)_{j \in \mathbf{ch}(\iota)}$ for any non-leaf node ι.

3.3.2. Multi-stage risk cost

Using this construction, we can proceed to formulate the optimal control problem for the ego vehicle. This problem, which will take the form of a *distributionally robust optimal control problem*, will be solved in receding horizon. At every time step, the realized value of θ_t is observed, and the involved ambiguity sets are updated accordingly, giving rise to a learning closed-loop control scheme.

Consider some N-stage scenario tree with $M = \sum_{k=0}^{N-1} |\mathbf{nod}(k)|$ non-leaf nodes, and let

$$\mathbf{u} = \{u^\iota \in \mathbb{R}^{n_u} \mid \iota \in \mathbf{nod}([0, N-1])\} \in \mathbb{R}^{M n_u}$$

denote a sequence of control actions over all non-leaf nodes the tree. It is worthwhile to remark that the set of all such sequences suffice to represent all possible closed-loop N-step policies, without requiring a predetermined parametrization. Consider the corresponding stochastic process $x_t \in \mathbb{R}^{n_x}$, $t \in \mathbb{N}$, representing the system state and satisfying dynamics $x_{t+1} = f(x_t, u_t, \theta_{t+1})$, or, in scenario tree notation, $x^\iota = f(x^{\mathbf{anc}(\iota)}, u^{\mathbf{anc}(\iota)}, \theta^\iota)$, for all $\iota \in \mathbf{nod}([1, N])$. Secondly, let $\ell : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}_+$ and $V_\mathrm{f} : \mathbb{R}^{n_x} \to \mathbb{R}_+$ denote some appropriate stage cost and terminal cost functions. To ease further notation, we define the mappings $\ell_t : \mathbb{R}^{n_u M} \to \mathbb{R}_+^{|\mathbf{nod}(t)|}$

$$\ell_t(\boldsymbol{u}) := (\ell(x^\iota, u^\iota))_{\iota \in \mathbf{nod}(t)} \text{ for } t = 0, \dots, N-1, \tag{17}$$

$$\ell_N(\boldsymbol{u}) := (V_\mathrm{f}(x^\iota))_{\iota \in \mathbf{nod}(N)}, \tag{18}$$

where for any $\iota \in \mathbf{nod}(t)$, $t \in \mathbb{N}_{[1,N]}$, x^ι depends on the (known) initial state x_0 and the control actions $\{u^{\mathbf{anc}^k(\iota)}\}_{k=1}^t$. Following the procedure described in §3.1, we can for each mode $\theta \in W$, construct an ambiguity set $\mathcal{A}_{t,\theta}(\beta)$ containing possible distributions over its successor modes, see (16). Furthermore, since they can be precomputed at negligible computational cost (cf. remark 3.2), we can propagate future ambiguity sets (which are fully defined by their centers and radii) and store them on the scenario tree. Indeed, a node $\iota \in \mathbf{nod}(t)$, $t \in [0, N-1]$, corresponds exactly to the event that the mode sequence $(\theta^{\mathbf{anc}^k(\iota)})_{k=0}^t$ is observed in the next t time steps. Using these "virtual" data points, the center and the radius of the ambiguity set at node ι are uniquely determined. We denote the resulting ambiguity sets by $\mathcal{A}^\iota$ for all $\iota \in \mathbf{nod}([0, N-1])$.

In doing so, we define at every time step, a *conditional risk mapping* $\rho_{|t}[z_{t+1}] : \mathbb{R}^{|\mathbf{nod}(t+1)|} \to \mathbb{R}^{|\mathbf{nod}(t)|}$, defined as [55, 12]

$$\rho_{|t}[z_{t+1}] := \left(\rho_{\mathcal{A}^\iota}[z^{\mathbf{ch}(\iota)}]\right)_{\iota \in \mathbf{nod}(t)}. \tag{19}$$

Intuitively, the conditional risk mapping $\rho_{|t}$ computes the risk the random outcome of a process at stage $t+1$, conditioned on its value at stage t. Using this construction, we define a multi-stage risk cost

$$V(\mathbf{u}) := \ell_0(\boldsymbol{u}) + \rho_{|0}\Big[\ell_1(\boldsymbol{u}) + \rho_{|1}\big[\dots \ell_{N-1}(\boldsymbol{u}) + \rho_{|N-1}[\ell_N(\boldsymbol{u})]\dots\big]\Big], \tag{20}$$

with ℓ_t, $t \in \mathbb{N}_{[0,N]}$ as in (17). This cost represents the worst-case expected cost over the scenario tree of predicted N-step future scenarios.

Example 3.4 (Cost function). To illustrate the practical computation of the cost (20), consider a prediction horizon of $N = 2$ steps and $W = \{1, 2\}$, resulting in the scenario tree depicted in Fig. 7. In this case, the cost function (20) reduces to

$$V(\mathbf{u}) = \ell_0(\boldsymbol{u}) + \rho_{|0}\Big[\ell_1(\boldsymbol{u}) + \rho_{|1}\big[\ell_2(\boldsymbol{u})\big]\Big], \tag{21}$$

The cost function can be evaluated through a backward substitution: The conditional risk mapping of the terminal cost $\ell_2(\boldsymbol{u}) = (V_\mathrm{f}(x^\iota))_{\iota=3}^6 \in \mathbb{R}^4$ is

represented on this particular scenario tree as

$$\begin{aligned} z_1 &:= \begin{bmatrix} z^1 \\ z^2 \end{bmatrix} = \rho_{|1}[\ell_2(\boldsymbol{u})] \\ &= \begin{bmatrix} \rho_{\mathcal{A}^1}\left[V_{\mathrm{f}}(x^3), V_{\mathrm{f}}(x^4)\right] \\ \rho_{\mathcal{A}^2}\left[V_{\mathrm{f}}(x^5), V_{\mathrm{f}}(x^6)\right] \end{bmatrix} \quad \text{(cf. (19))}. \end{aligned}$$

The values z^1, z^2 can then be assigned to the corresponding nodes 1, 2 on the tree. Next, plugging these quantities into (21), we may define another auxiliary variable,

$$z^0 := \rho_{|0}\left[\ell_1(\boldsymbol{u}) + z_1\right] = \rho_{\mathcal{A}^0}\left[\ell\left(x^1, u^1\right) + z^1, \ell\left(x^2, u^2\right) + z^2\right],$$

which is assigned to the root node 0, so it can finally be combined with $\ell_0(\boldsymbol{u})$ to obtain $V(\boldsymbol{u}) = \ell(x^0, u^0) + z^0$, as annotated on the bottom left of Fig. 7. △

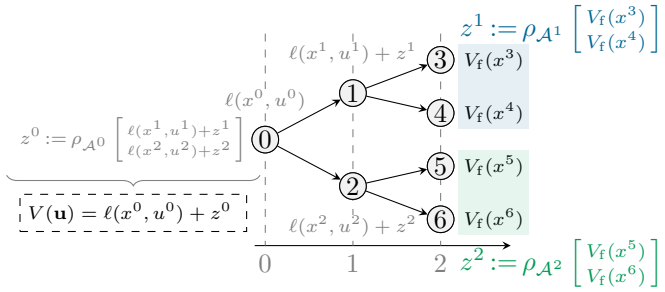


Figure 7: Graphical illustration of the computation of the cost in example 3.4 on a scenario tree for a prediction horizon $N = 2$, and $W = \{1, 2\}$.

3.3.3. Ambiguous chance constraints

Finally, following [16], collision avoidance constraints are imposed as *ambiguous* conditional chance constraints. To this end, we utilize a popular coherent risk measure known as the *average value-at-risk* [20, 12, 19]. As before, let Z denote a random variable, represented by its vector of realizations $z \in \mathbb{R}^d$, and probability vector $p \in \Delta_d$. For a given risk level $\alpha > 0$, the average value-at-risk of z is defined as [55, §6.2.4]

$$\begin{aligned} \mathsf{AV@R}_\alpha^p[z] &= \min_{t \in \mathbb{R}} t + \tfrac{1}{\alpha}\,\mathbb{E}_p\left[[Z - t]_+\right] \\ &= \min_{t \in \mathbb{R}} t + \tfrac{1}{\alpha}\langle p, [z - t]_+ \rangle. \end{aligned} \tag{22}$$

It can be shown that, $\mathsf{AV@R}_\alpha^p[z] \leq 0 \implies \mathbb{P}[Z > 0] \leq \alpha$[1], and furthermore, the former inequality can be exactly reformulated as a smooth constraint by

[1] See Appendix B for additional details.

introducing some additional auxiliary variables [12]. Besides resulting in a safe, continuous surrogate of the original chance constraint, the average value-at-risk has the additional practical benefit of penalizing larger constraint violations and thus providing more informative gradients to local optimization solvers.

The evaluation of (22), however, requires knowledge of p, but our controller only has access to the estimated ambiguity sets (16). For this reason, we impose constraints of this type robustly with respect to misestimation of p. In particular, this leads to the collision constraints

$$\bar{\rho}^{\iota}\left[\left(h\left(x^{i}\right)\right)_{i\in\mathbf{ch}(\iota)}\right] := \max_{p\in\mathcal{A}^{\iota}} \mathsf{AV@R}_{\alpha}^{p}\left[\left(h\left(x^{i}\right)\right)_{i\in\mathbf{ch}(\iota)}\right] \leq 0, \tag{23}$$

for all $\iota \in \mathbf{nod}\left([0, N-1]\right)$, where $h : \mathbb{R}^{n} \to \mathbb{R}$ is some function of the state that represents a negative distance. Section 2.3 describes several possible choices for the function h.

3.3.4. Full optimal control problem

For a given state x, we can now combine these components together with the previously introduced constraints in Section 2 into the optimal control problem

$$\begin{aligned}
\underset{\mathbf{u}}{\textbf{minimize}} \quad & V(\mathbf{u}) \\
\textbf{subj. to} \quad & x^{0} = x, \\
& x^{\iota} = f(x^{\mathbf{anc}(\iota)}, u^{\mathbf{anc}(\iota)}, \theta^{\iota}),\ \forall \iota \in \mathbf{nod}\left([1, N]\right), \\
& \bar{\rho}^{\iota}\left[\left(h\left(x^{i}\right)\right)_{i\in\mathbf{ch}(\iota)}\right] \leq 0,\ \forall \iota \in \mathbf{nod}\left([0, N-1]\right), \\
& \underline{u} \leq u \leq \bar{u},\ \forall \iota \in \mathbf{nod}\left([0, N-1]\right), \\
& g_{\text{phys}}(x^{\iota}) \leq 0,\ \forall \iota \in \mathbf{nod}\left([0, N]\right).
\end{aligned} \tag{24}$$

Remark 3.5 (Feasibility). In order to guarantee that a control action is computed at all times, it is common in practice to relax the constraints by introducing slack variables (see e.g., [48]), resulting in a soft-constrained version of (24). An alternative option is to hand-craft a back-up controller, which gets invoked whenever the optimal control problem is infeasible [64, 65]. However, since the aim of this work is to study the behavior of (24) in closed-loop, we consider the design of additional safeguards to be beyond the scope of this work.

In typical MPC fashion, our proposed controller involves repeated application of the following steps: (i) solve (24) for the current state x; (ii) apply the resulting control action in the root node; (iii) measure/observe the new state and mode; and (iv) update the ambiguity set parameters and repopulate the scenario tree.

The ambiguity set (13) belongs to the general set of *conic-representable* risk measures, which allows us to tractably reformulate the OCP (24) as a standard nonlinear program, using duality-based techniques described in [12, 28, 52].

4. Numerical case studies

4.1. Exploratory experiments

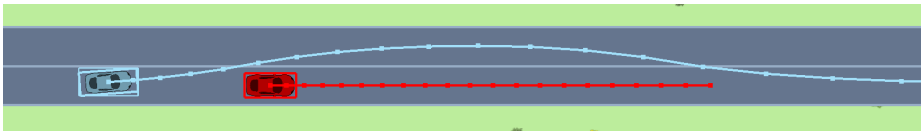


Figure 8: Scene layout for the experiments in Sections 4.1 and 4.2. (Blue: ego vehicle; red: target vehicle)

This section provides some exploratory and qualitative simulation results which motivate some of the decisions made for the configurations of further experiments. Unless stated otherwise, the sampling time used for these experiments is $T_\mathrm{s} = 0.2\,\mathrm{s}$. The reference velocity was kept at $v_\mathrm{ref} = 30\,\mathrm{m\,s^{-1}}$. For other settings, we refer to Appendix C.

All nonlinear programs are solved using IPOPT [44], using the CasADI [66] interface in Python.

Experiment 4.1 — Collision avoidance formulations.
As a first comparison of the effectiveness of the above collision-avoidance constraints, we run a closed-loop simulation of an overtaking scenario with a single target vehicle, shown in Fig. 8. For the time being, we employ **deterministic** model predictive control, i.e., we keep the mode of the (single) target vehicle fixed to the right lane, both for the predictive model and the simulation model. This allows us to isolate the effects of the choices regarding collision avoidance formulations.

Figure 9 gives an overview of the resulting trajectories of the ego vehicle with the different formulations. For the projection formulation, the solver occasionally returned an infeasible solution. In this experiment, we accepted the solution regardless, allowing the controller to correct using feedback in the next time instances.

Table 2: Infeasible instances per configuration (out of 50).

		Projection			
Formulation	Ellipse	AA* (exact)	AA* (smooth)	exact	smooth
Failure cases	0	19	0	1	1

* *Axis-Aligned*

In Table 2, the number of infeasible instances are found for the different collision avoidance constraints. This leads to several conclusions: (i) The ellipsoidal and smoothened, axis-aligned projection-type collision avoidance constraints perform the most reliably, yielding no infeasible solutions. We

conclude that this formulation is most numerically well-behaved, as infeasibilities are not caused by model mismatch or other sources of uncertainty; (ii) The sigmoid approximation for the projection-type constraints decreases the number of infeasible instances significantly; (iii) The exact projection-type constraint leads to infeasible solutions at a rate that prohibits corrections using feedback, leading to a collision in the closed-loop trajectories; (iv) From Fig. 9, the conservatism introduced by either the ellipsoidal constraints or the smoothing approximation seems insignificant.

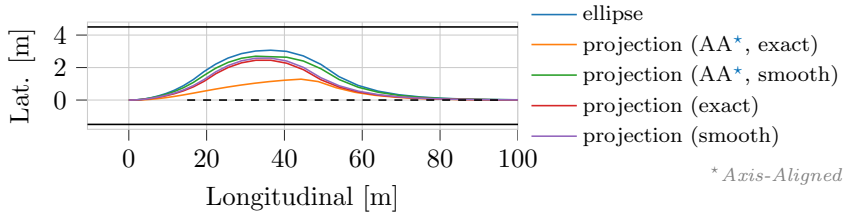


Figure 9: Overtaking trajectories for different collision-avoidance formulations. The colored lines indicate the ego vehicle trajectories for different collision avoidance constraints. The dashed line indicates the target vehicle positions. (Best viewed in color.)

This experiment, combined with similar test runs yielding observations consistent with the above conclusions motivates the use of the **ellipsoidal constraints** by default.

Experiment 4.2 — Ellipsoidal parameter tuning. A remaining degree of freedom of the ellipsoidal constraint is the choice of the tuning parameter γ, which determines the "elongation" of the ellipsoidal shape. The lower this value, the wider the ellipse. If it is chosen too low, an overtaking maneuver will be rendered impossible, as illustrated in Fig. 11a. On the other hand, if an excessively large value is chosen, the time spent in the non-preferred lane during an overtaking maneuver increases, with an increased cost as a result. This trade-off is illustrated in Fig. 10. For the current cost function, values between 7 and 15 were observed to provide the best performance.

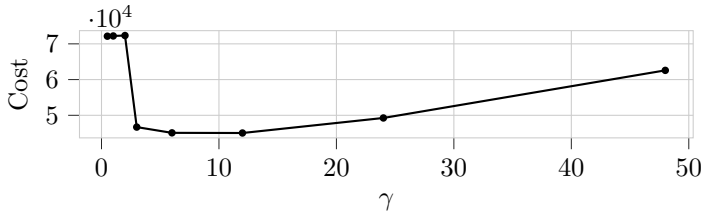
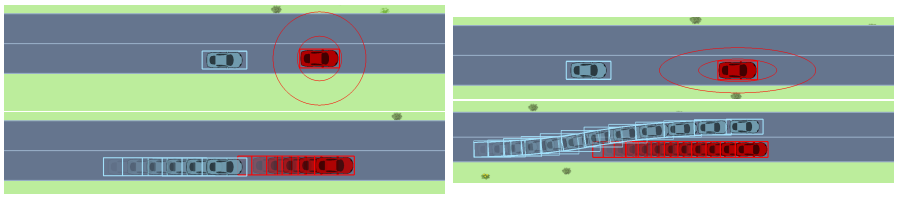


Figure 10: Closed-loop cost versus the elongation parameter γ.



(a) $\gamma = 1$ (b) $\gamma = 12$

Figure 11: Illustration of the ellipsoidal constraint sets and closed-loop trajectories for different values of the elongation parameter γ. The outer ellipse corresponds to the extended rectangle and delineates the constraint set for the center point of the ego vehicle. (Blue: ego vehicle; red: target vehicle.)

4.2. Effects of sample-size and scenario tree topology on conservatism

We now move on to the setting where a (nontrivial) scenario tree is built for the predictive model and the full learning distributionally robust MPC framework is used for control. The following experiments illustrate the effects of two important parameters that control risk-aversion in this setting.

4.2.1. Deterministic target vehicle

We first study the behavior of the controller in a situation where the true behavior of the target vehicle is rather safe and predictable, i.e., it regulates to a fixed reference velocity in the right lane and performs no lane-change maneuvers, i.e., the transition kernel governing the target vehicle driving style in this experiment is $P = \left[\begin{smallmatrix} 1 & 0 \\ 1 & 0 \end{smallmatrix}\right]$. Recall that the modes in this set-up represent a target lane, i.e., element P_{ij} in P represents the probability that at any time, the target vehicle will change its desired lane to lane j given that it is currently driving towards lane i. Thus, with this instance of P, the target vehicle will drive towards the center of the right lane with probability 1, regardless of the initialization of the Markov chain. The overall setup of the scenario is the same as in Experiment 4.1. This scenario is representative of a large fraction of expected driving situations and is therefore an interesting case to consider.

Under these conditions, it is evident that a deterministic MPC scheme would be optimal, since the mode sequence is fixed a priori and therefore, the deterministic predictions are exact. In these experiments, we will therefore refer to the deterministic controller as *prescient*. Since by construction of the driving scenario, the behavior of the target vehicle is predictable and safe for an overtaking maneuver, the interesting question is: *"How conservative is the learning distributionally robust MPC scheme in an inherently safe scenario, and does this lead to impractical behavior?"*

Experiment 4.3 — Conservatism versus branching horizon. The current experiment investigates the effect of the following two quantities on the conservatism of the collision avoidance constraints (23): (i) the amount of observed data from the target vehicle; and (ii) the branching horizon used in the scenario tree.

We simulate the MPC scheme in closed loop for the overtaking scenario and record its achieved cost by evaluating the stage cost at every time step and summing over the simulation run. To simulate the behavior of the controller after observing the target vehicle for some time, we sample a sequence of length m from the governing Markov chain to train the learning controller. Note that since we initialize the empirical distribution to the uniform distribution and the experiment is constructed such that $\theta_t = 1, \forall t \in \mathbb{N}$, the estimated probability matrix will be identically $\widehat{P} = \left[\begin{smallmatrix} 1 & 0 \\ 0.5 & 0.5 \end{smallmatrix}\right]$, for any sample size larger than 1. Therefore, any difference in the behavior stems from the confidence on these estimates, as measured by the radii of the ambiguity sets.

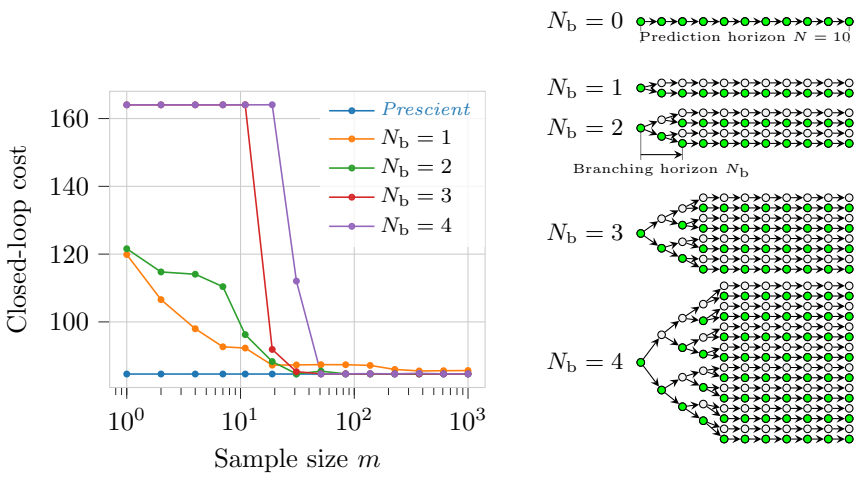
Fig. 12 shows estimated closed-loop costs over a simulation versus this sample size m. As the sample size grows and the radii of the ambiguity sets shrink according to (12), the controllers become more confident about the prediction that a sudden lane change will not occur and decide to perform an overtaking maneuver, which results in the sudden decrease in costs in the figure. Note that the smaller the branching horizon, the more quickly this threshold is reached, as fewer potentially dangerous future situations are taken into account during prediction.

However, after roughly 50 data points – corresponding to 10 seconds of observation – even the most conservative design is able to perform the overtaking maneuver without violating its constraints, achieving the same closed-loop cost as the *prescient* controller. Meanwhile, this controller does plan with respect to a considerably more detailed model, which robustifies it against use cases where a sudden lane change would spuriously occur. In contrast, a robust formulation, where the collision avoidance constraints are imposed in the worst-case realization, an overtaking maneuver is not feasible, as illustrated by the behavior of the distributionally robust (DR) controller at sample size $m = 0$.

Experiment 4.4 — Prediction breadth versus depth. In Experiment 4.3, an increasing branching horizon was demonstrated to rather quickly yield optimal behavior in the case where the true underlying system is not exhibiting low-probability switching behavior, even though it improves resilience of the control scheme to unexpected behaviors.

Naturally, however, an increased branching horizon also induces larger scenario trees, and therefore higher computational costs. For this reason, we repeat Experiment 4.3 with (approximately) constant scenario tree sizes. That is, the prediction horizon (*"depth"* of the prediction) is traded off with the branching horizon (*"breadth"* of the prediction) such that the total size of the scenario tree does not change significantly — see Fig. 13b for an illustration of the resulting scenario trees. In doing so, we investigate whether it is beneficial to increase the detail of the prediction at a given stage at the cost of a reduced prediction horizon, given a fixed computational budget.

Fig. 13 shows that in this setup a degradation in the performance can indeed be observed whenever the trade-off is made too heavily in favor of *breadth* rather than *depth*. Although at sufficiently large sample sizes, the



(a) Cost versus sample size.

(b) Scenario tree topologies.

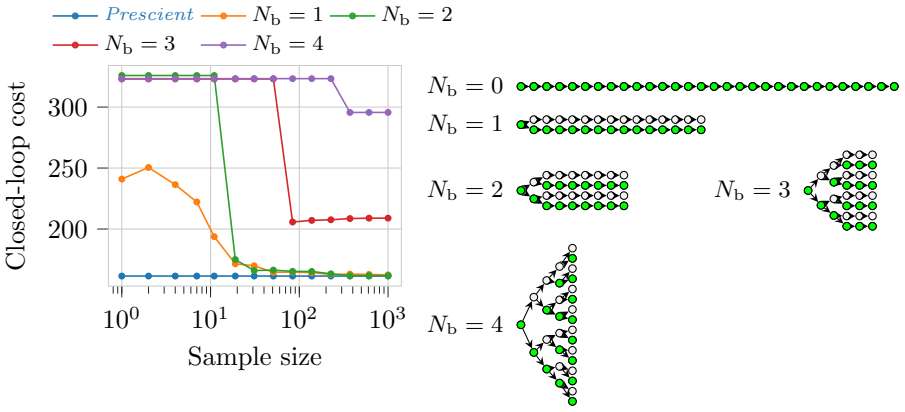
Figure 12: Closed-loop costs versus sample size and scenario tree topologies corresponding to different branching horizons N_b and fixed prediction horizon $N = 10$ in the deterministic setting of Experiment 4.3. The colored nodes indicate modes where the target vehicle is predicted to move towards the right lane.

controllers with short prediction horizons are still able to perform an overtaking maneuver (note the sudden decrease in closed loop costs), they do so at a significantly slower pace, as only a part of the maneuver can be predicted at each time instance. This naturally leads to higher costs. Note, however, that until a branching horizon of 2, there does not yet seem to be a significant degradation in performance as a result of the shortened horizon.

This suggests that there exists a minimal prediction horizon that is required to allow the prediction of the optimal maneuver. As long as the prediction horizon is longer than this value, the optimal behavior is obtained (given enough time to learn the governing distributions). Since for a fixed number of time steps, the length of the preview window (expressed in true seconds) is proportional to the sampling time, it is expected that the required prediction horizon to predict the full overtaking decreases as T_s increases. Indeed, by increasing the sampling horizon to $T_\mathrm{s} = 0.3\,\mathrm{s}$, as presented in Fig. 14, the discrepancy between the results for different values of N_b is reduced considerably. It is however important to emphasize that despite the observed trend, the closed-loop cost need not increase monotonically with the branching horizon, as one may suspect from Fig. 13 alone. Indeed, in the case of Fig. 14 for instance, the controller using $N_\mathrm{b} = 2$ outperforms the one using $N_\mathrm{b} = 1$ at smaller sample sizes. It is therefore advisable in practice to fine-tune the exact choice of N_b to the application at hand.

We conclude that depending on the timescale of the use case, a minimal prediction horizon may be required in order to achieve satisfactory performance. In such cases, it may be beneficial to reduce the branching horizon by

a number of steps to make this possible. An important caveat here, is that if a terminal constraint is imposed, it needs to be imposed at stage N_b for it to guarantee recursive feasibility of the scheme. As a result, N_b should be as large as possible in order to maximally enlarge the region of attraction.



(a) Cost versus sample size

(b) Scenario tree topologies.

Figure 13: Closed-loop costs versus sample size and scenario tree topologies corresponding to different branching horizons N_b in the deterministic setting of Experiment 4.3.

4.2.2. Stochastic target vehicle

Until now, the true behavior of the target vehicle was taken to be deterministic. In that case, there is no benefit to a learning-based stochastic controller. Of course, in reality, we cannot assume to have perfect predictions of the target vehicle behavior. This experiment aims to investigate the degree to which adding a nonzero switching probability between the lanes leads to hazardous situations.

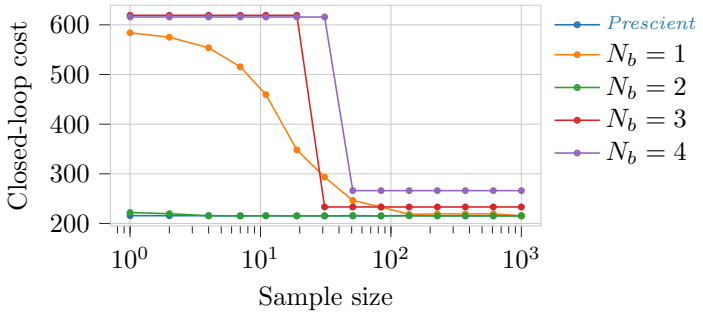


Figure 14: Cost versus sample size for the scenario tree topologies shown in Fig. 13b, for a sample size of $T_\mathrm{s} = 0.3\,\mathrm{s}$ (compared to $T_\mathrm{s} = 0.2\,\mathrm{s}$ in Fig. 13).

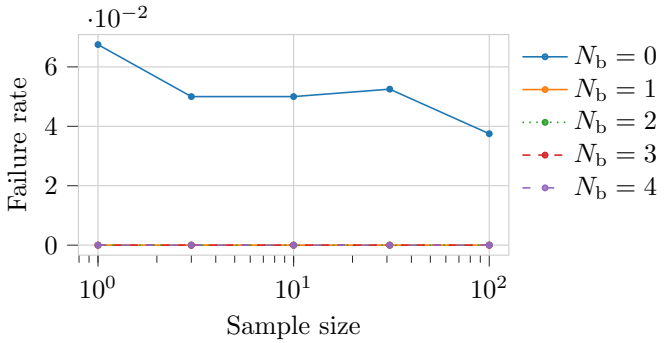


Figure 15: Fraction of infeasible problem instances over 10 random simulations of 40 time steps. Note that the lines of all nonzero branching horizon results overlap at zero.

Experiment 4.5 — Safety comparison with deterministic MPC.

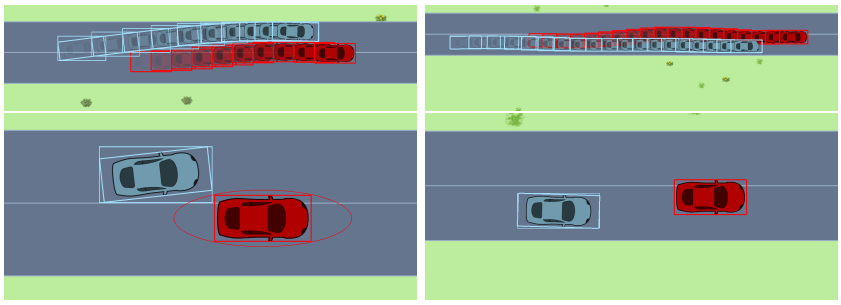
For the sake of illustration, we model a target vehicle that switches lanes rather often by taking $P = \left[\begin{smallmatrix} 0.7 & 0.3 \\ 0.3 & 0.7 \end{smallmatrix}\right]$. We repeat the previous overtaking scenario with stochastic target vehicle behavior for 10 times and record the number of times the solver runs into infeasibility for each of the scenario tree topologies and sample sizes.

Figure 15 presents the fraction of instances of the optimal control problem that were detected to be infeasible. Since the deterministic MPC controller does not anticipate any of the lane changes, it overconfidently attempts to perform an overtaking maneuver in most simulations, leading to many dangerous situations, and infeasibility in approximately 5% of the simulated time steps, on average. By contrast, the learning distributionally robust MPC controllers more accurately assess the situation and do not run into infeasibilities. An example of a trajectory leading up to infeasibility of the deterministic controller is given in Fig. 16.

The random seed for all simulations was kept constant so that one would expect the failure rate of the deterministic controller to be independent of the sample size. The apparent dependence of the results for the deterministic controller on the sample size in Fig. 16 are simply due to numerical errors. Indeed, visual inspection of the trajectories confirms that in some runs, the predicted state sequence lies very close to the boundary of the feasible set, so that a very slight deviation in solutions up to this time step may make it infeasible.

Finally, it is important to note that by careful selection of a terminal constraint set, recursive feasibility can be guaranteed for the distributionally robust controller, whereas this is not possible for the deterministic variant in this situation. This was demonstrated for an ACC use case described in [16]. There furthermore exists a significant body of literature on the problem of constructing invariant sets in the context of motion planning and obstacle avoidance, although most are developed with slightly different use cases in mind [67, 68, 69]. Comparing such approaches on their computational

efficiency, conservatism etc., is however beyond the scope of the current work.



(a) Deterministic (b) Distributionally robust

Figure 16: Snapshot of an infeasible state that occurred during the simulation of the deterministic controller (a) and the corresponding state (and trajectory) of the distributionally robust controller (b). The latter correctly anticipates the dangerous maneuver of the target vehicle and stays behind it. (Blue: ego vehicle; red: target vehicle.)

4.3. Complexity benchmarks

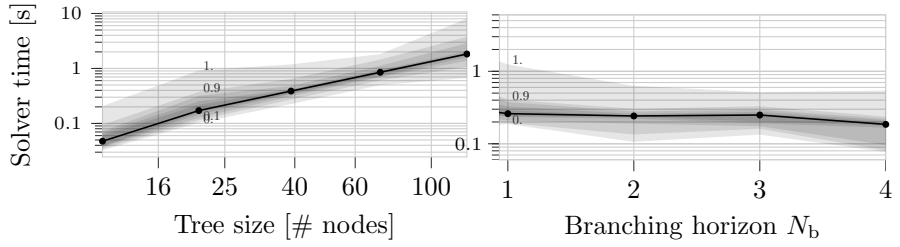
Experiment 4.6. Figure 17 presents a summary of the computation times measured during Experiments 4.3 and 4.4 described above. In Fig. 17a, the runtimes are shown as a function of the tree size. Analogously to the convex case [16], the complexity of the optimal control problem is linear in the number of nodes in the scenario tree. Given a fixed branching horizon, it consequently grows linearly in the prediction horizon as well. However, the complexity does grow exponentially in the branching horizon. As a result, the choice of the branching horizon will therefore largely be dictated by the available computational budget.

As demonstrated by Fig. 17b, trading of the prediction horizon with the branching horizon in such a way that the total tree size (measured in the number of nodes) is preserved (as in Experiment 4.4) indeed has no significant influence on the computational load.

4.3.1. Multi-vehicle scenario

In order to assess the capabilities of the proposed control strategy in a more complex scenario, we consider the set-up illustrated in Fig. 18, involving multiple target vehicles, and furthermore featuring a more sophisticated target vehicle driving model. In this set-up, we assume, as before, that the front vehicle (shown in red) behaves according to the model introduced in (4). Essentially, this model assumes that interaction with the ego vehicle is negligible. This is reasonable since the ego vehicle is driving behind the target vehicle.

For a target vehicle approaching from the rear (shown in orange), this assumption is less easily justified. Instead, we consider the longitudinal control of this vehicle to be given by a slightly modified version of the Intelligent driver



(a) Fixed prediction horizon.

(b) Fixed tree size (approx. 25 nodes).

Figure 17: Solver time (IPOPT[44]) versus (a) the number of nodes in the scenario tree, for a fixed prediction horizon (see Fig. 12b); and (b) the branching horizon in the scenario tree, for a fixed number of nodes (see Fig. 13b). The solid line depicts the median solver time over a closed-loop MPC run with the setup of Experiment 4.3, the shaded regions (from light to dark) represent the quantiles at increments of 0.1.

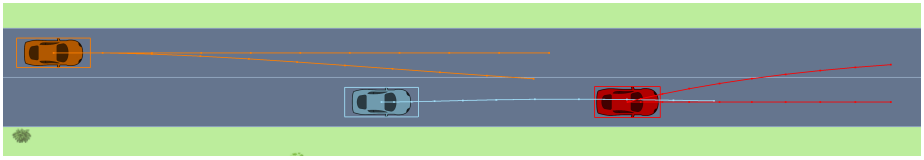


Figure 18: Scene layout for the experiment in §4.3.1, including 2 sample trajectories for each vehicle. (Blue: ego vehicle, red/orange: target vehicles.)

model (IDM) [70], which is a very commonly used model for simulating longitudinal driving behavior on highways. Specifically, assigning the index r to the rear vehicle, we define its driving policy as

$$\pi^{(\mathrm{r})}(\boldsymbol{x},\theta) = \begin{bmatrix} a_x(\boldsymbol{x},\theta) \\ k_y\left(y_{\mathrm{lane},\theta} - p_y^{(\mathrm{r})}\right) - k_{v,y} v_y^{(\mathrm{r})} \end{bmatrix}, \tag{25}$$

where the lateral component is identical to (4), and the longitudinal control is given by the following model[2]

$$a_x(\boldsymbol{x},\theta) = \overline{a}\,\mathbf{tanh}\left(\frac{a_0}{\overline{a}}\left(1 - \left(\frac{v}{v_{\mathrm{ref},\theta}}\right)^{\gamma} - \varphi\left(p_y^{(0)} - p_y^{(\mathrm{r})}\right)\left(\frac{s^\star(\boldsymbol{x})}{s(\boldsymbol{x})}\right)^2\right)\right), \tag{26}$$

where

$$s(\boldsymbol{x}) := p_x^{(0)} - p_x^{(\mathrm{r})} \text{ and } s^\star(\boldsymbol{x}) := s_0 + v_x^{(\mathrm{r})} T + \frac{v_x^{(\mathrm{r})}\left(v_x^{(\mathrm{r})} - v^{(0)}\right)}{2\sqrt{ab}},$$

are adapted closely from IDM. The functions $\varphi(z) := \mathbf{exp}\left(-z^2\right)$ and $\overline{a}\,\mathbf{tanh}\left(\frac{\cdot}{\overline{a}}\right)$ were respectively introduced to (i) downweigh the interaction term between vehicle r and the ego vehicle based on their lateral distance; and (ii) ensure that

[2] see Table C.5 for a description and numerical values for the parameters in the IDM policy.

the target vehicle accelerations remain bounded to some physically meaningful interval $[-\overline{a}, \overline{a}]$.

Note that (26) introduces interaction between the stochastic forecasts of the rear target vehicle and the ego vehicle, which is a desirable property of a high-level motion planning system, as a fully exogenous target vehicle predictions tend to result in unreasonably conservative behavior [71].

Experiment 4.7. Table 3 compares the solver times for several closed-loop simulations of 30 time steps[3] of the multi-vehicle setup illustrated in Fig. 18 for different scenario tree structures. We compare (i) a *full uncertainty* model, where both target vehicles are modeled using uncertain lane selections, resulting in a Markov chain with $d = 4$ modes (2 lanes per vehicle); and (ii) a *simplified* model, where the rear vehicle is assumed to remain in the leftmost lane, and only uncertain lane change behavior is modeled for the frontal target vehicle. This results in a scenario tree with $d = 2$ modes, as we had in the previous experiments.

Besides the potentially larger number of modes, remaining sources of additional complexity in the current set-up are the increased state dimension n_x (12 instead of 8), the nonlinearity and interaction in the rear target vehicle policy (cf. (25)) as compared to the exogenous, affine policy (4) used in previous experiments.

When comparing Table 3 with Figure 17a, we find that despite these additional complexities, the solver time for a comparably sized scenario tree is not much different from in the simpler single-target vehicle case considered in Experiment 4.6, and as we concluded before, the most important factor in determining the computation time seems to be the size of the scenario tree.

However, this indicates that when considering a full-uncertainty model for multiple vehicles, only a relatively short prediction (or branching) horizon can be afforded to obtain an acceptable computational cost. In real-life practical implementations of this control method, an appropriate trade-off between robustness and computation time can thus be made by the designer, through the selection (i) the number of modes for each vehicle—and thus, the branching factor of the scenario tree; (ii) the number of branching steps N_b; (iii) the total horizon length N; and (iv) the sampling rate T_s. Furthermore, it is reasonable to assume that in practice, a higher level planning module would be used to select a small number (1 or 2) of target vehicles to include in building the scenario tree, while the remaining road users in the environment are taken into account using a simpler model. We consider a more detailed study of such a hierarchical design for future work.

[3] We refer to the online supplementary material [1] for videos of the closed-loop trajectories.

Table 3: Median (± std. dev.) solver times for Experiment 4.7.

	Tree size (N, N_b)	#nodes	Solver time [ms]
Simplified ($d = 2$)	(5,1)	11	38.17 (± 12.70)
	(5, 2)	19	70.78 (± 26.61)
	(7, 2)	27	124.47 (± 77.25)
Full uncertainty ($d = 4$)	(5, 1)	21	66.23 (± 18.05)
	(5, 2)	69	271.05 (± 110.42)
	(7, 2)	101	612.68 (± 590.17)

5. Conclusion and future work

We have presented a case study on the use of a learning, distributionally robust MPC controller for closed-loop path planning in a highway driving set-up. We have highlighted the most important design parameters and illustrated empirically how they affect controller performance, safety and computational load, among which a suitable trade-off is to be found, based on the concrete application.

This study has highlighted some of the desirable behavioral properties of the controller related to its ability to autonomously and systematically trade-off performance with caution. Combined with its strong theoretical underpinning [28], these empirical results provide a promising foundation for future developments of distributionally robust MPC methodologies for autonomous navigation and other automated driving applications. Finally, we highlight several interesting directions for further extensions and improvements to the methodology.

5.1. Suggestions for future work

We identify several possible directions for further research on the topic, aimed primarily at extending the utility of the developed methodology to more practical, non-academic settings. We divide these directions into the following categories.

5.1.1. Computational

Custom (parallelized) solvers. As demonstrated in the case studies, a significant bottleneck in the practicality of distributionally robust MPC techniques over scenario trees remains the computational load. However, in this work, only an off-the-shelf, general-purpose solver has been used, without any exploitation of the underlying scenario tree structure. For instance, the increasingly widespread availability of GPUs, even in embedded applications, makes an appealing case for a massively parallelized approach towards solving optimal control problems over the individual scenarios within a tree. (See, e.g, [72, 73].)

5.1.2. Methodological

Observers and parameter estimation. In the case where observations are made from continuous data (state or output measurements), with or without additive noise, an important question is how to determine the most likely mode to explain such an observation [41]. This is particularly challenging in the case of nonlinear dynamics and additive measurement noise. Furthermore, it would be interesting to investigate how the misestimation at this level would translate into possible system-theoretic guarantees of the closed loop.

Time-varying distributions. we have considered static Markov chains to govern the stochastic processes. However, in real driving scenarios, the underlying distributions would likely be time-varying. In this case, the derived ambiguity bounds are no longer valid. However, empirical bounds based on cross-validation or bootstrapping could be considered, in order to obtain online estimates of the uncertainty, which could track relatively slow-varying distributions. A trade-off would then likely arise between the time window/forgetting factor in the learning scheme and the expected rate of change of the distributions. If the distribution is rapidly changing, the time window in which the observations are valid is expected to be short, which will likely lead to rather large uncertainty bounds in steady state. By contrast, for a slowly changing distribution, a long window of observations can be gathered, and therefore, the estimates are expected to converge to a closer estimate of the true distribution.

References

[1] "Safe, Learning-Based MPC for Highway Driving under Lane-Change Uncertainty: A Distributionally Robust Approach." [Online]. Available: https://mathijssch.github.io/DRMPC-lane-change-supp/

[2] A. Carvalho, S. Lefévre, G. Schildbach, J. Kong, and F. Borrelli, "Automated driving: The role of forecasts and uncertainty—A control perspective," *European Journal of Control*, vol. 24, pp. 14–32, Jul. 2015.

[3] S. Lefèvre, D. Vasquez, and C. Laugier, "A survey on motion prediction and risk assessment for intelligent vehicles," *ROBOMECH Journal*, vol. 1, no. 1, Dec. 2014.

[4] J. E. Smith and R. L. Winkler, "The Optimizer's Curse: Skepticism and Postdecision Surprise in Decision Analysis," *Management Science*, vol. 52, no. 3, pp. 311–322, Mar. 2006.

[5] B. P. G. Van Parys, P. M. Esfahani, and D. Kuhn, "From Data to Decisions: Distributionally Robust Optimization Is Optimal," *Management Science*, Nov. 2020.

[6] J. Lofberg, "Approximations of closed-loop minimax MPC," in *42nd IEEE International Conference on Decision and Control (IEEE Cat. No.03CH37475)*, vol. 2, Dec. 2003, pp. 1438–1442 Vol.2.

[7] I. Batkovic, U. Rosolia, M. Zanon, and P. Falcone, "A Robust Scenario MPC Approach for Uncertain Multi-Modal Obstacles," *IEEE Control Systems Letters*, vol. 5, no. 3, pp. 947–952, Jul. 2021.

[8] B. Schoettle and M. Sivak, "A Preliminary Analysis of Real-World Crashed Involving Self-Driving Vehicles," The University of Michigan Transportation Research Institute, Ann Arbor, Michigan, Tech. Rep. UMTRI-2015-34, Oct. 2015. [Online]. Available: http://umich.edu/~umtriswt/PDF/UMTRI-2015-34.pdf

[9] H. Rahimian and S. Mehrotra, "Distributionally Robust Optimization: A Review," *arXiv:1908.05659 [cs, math, stat]*, Aug. 2019.

[10] J. Dupačová, "The minimax approach to stochastic programming and an illustrative application," *Stochastics*, vol. 20, no. 1, pp. 73–88, Jan. 1987.

[11] P. Sopasakis, D. Herceg, A. Bemporad, and P. Patrinos, "Risk-averse model predictive control," *Automatica*, vol. 100, pp. 281–288, Feb. 2019.

[12] P. Sopasakis, M. Schuurmans, and P. Patrinos, "Risk-averse risk-constrained optimal control," in *2019 18th European Control Conference (ECC)*, Jun. 2019, pp. 375–380.

[13] T. Phan-Minh, E. C. Grigore, F. A. Boulton, O. Beijbom, and E. M. Wolff, "CoverNet: Multimodal Behavior Prediction Using Trajectory Sets," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Seattle, WA, USA: IEEE, Jun. 2020, pp. 14 062–14 071.

[14] A. D. Bonzanini, J. A. Paulson, and A. Mesbah, "Safe Learning-based Model Predictive Control under State- and Input-dependent Uncertainty using Scenario Trees," in *2020 59th IEEE Conference on Decision and Control (CDC)*, Dec. 2020, pp. 2448–2454.

[15] T. Summers, "Distributionally Robust Sampling-Based Motion Planning Under Uncertainty," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2018, pp. 6518–6523.

[16] M. Schuurmans, A. Katriniok, H. E. Tseng, and P. Patrinos, "Learning-based risk-averse model predictive control for adaptive cruise control with stochastic driver models," in *2020 21st IFAC World Congress*, Jul. 2020.

[17] M. Bichi, G. Ripaccioli, S. D. Cairano, D. Bernardini, A. Bemporad, and I. V. Kolmanovsky, "Stochastic model predictive control with driver behavior learning for improved powertrain control," in *49th IEEE Conference on Decision and Control (CDC)*, Dec. 2010, pp. 6077–6082.

[18] A. Nemirovski, "On safe tractable approximations of chance constraints," *European Journal of Operational Research*, vol. 219, no. 3, pp. 707–718, Jun. 2012.

[19] A. Hakobyan, G. C. Kim, and I. Yang, "Risk-Aware Motion Planning and Control Using CVaR-Constrained Optimization," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3924–3931, Oct. 2019.

[20] M. P. Chapman, J. P. Lacotte, K. M. Smith, I. Yang, Y. Han, M. Pavone, and C. J. Tomlin, "Risk-sensitive safety specifications for stochastic systems using Conditional Value-at-Risk," *arXiv:1909.09703*, Sep. 2019.

[21] A. Dixit, M. Ahmadi, and J. W. Burdick, "Risk-Sensitive Motion Planning using Entropic Value-at-Risk," in *2021 European Control Conference (ECC)*, Jun. 2021, pp. 1726–1732.

[22] M. Ahmadi, M. Ono, M. D. Ingham, R. M. Murray, and A. D. Ames, "Risk-Averse Planning Under Uncertainty," in *2020 American Control Conference (ACC)*, Jul. 2020, pp. 3305–3312.

[23] M. Ono, M. Pavone, Y. Kuwata, and J. Balaram, "Chance-constrained dynamic programming with application to risk-aware robotic space exploration," *Autonomous Robots*, vol. 39, no. 4, pp. 555–571, Dec. 2015.

[24] A. Hakobyan and I. Yang, "On Improving the Distributional Robustness of Risk-Aware Controllers in Learning-Enabled Environments," in *60th IEEE Conference on Decision and Control*, Austin, Texas, USA, 2021, pp. 6017–6024.

[25] H. Nishimura, B. Ivanovic, A. Gaidon, M. Pavone, and M. Schwager, "Risk-Sensitive Sequential Action Control with Multi-Modal Human Trajectory Forecasting for Safe Crowd-Robot Interaction," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2020, pp. 11 205–11 212.

[26] S. Samuelson and I. Yang, "Safety-Aware Optimal Control of Stochastic Systems Using Conditional Value-at-Risk," *arXiv:1802.07903 [cs, math]*, Feb. 2018.

[27] Y. Chen, U. Rosolia, W. Ubellacker, N. Csomay-Shanklin, and A. D. Ames, "Interactive Multi-Modal Motion Planning With Branch Model Predictive Control," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 5365–5372, Apr. 2022.

[28] M. Schuurmans and P. Patrinos, "A General Framework for Learning-Based Distributionally Robust MPC of Markov Jump Systems," *arXiv:2106.00561*, Jun. 2021. [Online]. Available: http://arxiv.org/abs/2106.00561

[29] A. Hakobyan and I. Yang, "Wasserstein Distributionally Robust Motion Control for Collision Avoidance Using Conditional Value-at-Risk," *arXiv:2001.04727 [cs, eess]*, Jan. 2020. [Online]. Available: http://arxiv.org/abs/2001.04727

[30] A. Wang, A. Jasour, and B. C. Williams, "Non-Gaussian Chance-Constrained Trajectory Planning for Autonomous Vehicles Under Agent Uncertainty," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6041–6048, Oct. 2020.

[31] A. Hakobyan and I. Yang, "Wasserstein Distributionally Robust Motion Control for Collision Avoidance Using Conditional Value-at-Risk," *arXiv:2001.04727*, Jan. 2020.

[32] S. Liu, K. Zheng, L. Zhao, and P. Fan, "A Driving Intention Prediction Method Based on Hidden Markov Model for Autonomous Driving," Feb. 2019.

[33] N. Deo and M. M. Trivedi, "Multi-Modal Trajectory Prediction of Surrounding Vehicles with Maneuver based LSTMs," in *2018 IEEE Intelligent Vehicles Symposium (IV)*, Jun. 2018, pp. 1179–1184.

[34] X. Huang, G. Rosman, I. Gilitschenski, A. Jasour, S. G. McGill, J. J. Leonard, and B. C. Williams, "HYPER: Learned Hybrid Trajectory Prediction via Factored Inference and Adaptive Sampling," Oct. 2021.

[35] M. Werling, J. Ziegler, S. Kammel, and S. Thrun, "Optimal trajectory generation for dynamic street scenarios in a Frenét Frame," in *2010 IEEE International Conference on Robotics and Automation*, May 2010, pp. 987–993.

[36] R. Rajamani, *Vehicle Dynamics and Control*, 2nd ed., ser. Mechanical Engineering Series. Springer US, 2012.

[37] O. L. d. V. Costa, M. D. Fragoso, and R. P. Marques, *Discrete-Time Markov Jump Linear Systems*, ser. Probability and Its Applications. London: Springer, 2005.

[38] E. Leurent, D. Efimov, and O.-A. Maillard, "Robust-Adaptive Control of Linear Systems: Beyond Quadratic Costs," *arXiv:2002.10816 [cs, eess, stat]*, Oct. 2020.

[39] J. Nilsson, M. Brännström, E. Coelingh, and J. Fredriksson, "Lane Change Maneuvers for Automated Vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 5, pp. 1087–1096, May 2017.

[40] J. Nilsson, Y. Gao, A. Carvalho, and F. Borrelli, "Manoeuvre generation and control for automated highway driving," *IFAC Proceedings Volumes*, vol. 47, no. 3, pp. 6301–6306, 2014.

[41] M. Schuurmans and P. Patrinos, "Data-driven distributionally robust control of partially observable jump linear systems," *arXiv:2105.02511*, May 2021. [Online]. Available: http://arxiv.org/abs/2105.02511

[42] Y. J. J. Bagul, "A smooth transcendental approximation to $|x|$," *International J. of Math. Sci. & Engg. Appls. (IJMSEA)*, vol. Vol. 11, no. II, pp. 213–217, Aug. 2017.

[43] R. M. Corless, G. H. Gonnet, D. E. G. Hare, D. J. Jeffrey, and D. E. Knuth, "On the LambertW function," *Advances in Computational Mathematics*, vol. 5, no. 1, pp. 329–359, Dec. 1996.

[44] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical Programming*, vol. 106, no. 1, pp. 25–57, Mar. 2006.

[45] B. Evens, M. Schuurmans, and P. Patrinos, "Learning MPC for Interaction-Aware Autonomous Driving: A Game-Theoretic Approach," *arXiv:2111.08331 [math]*, Nov. 2021. [Online]. Available: http://arxiv.org/abs/2111.08331

[46] B. Hermans, P. Patrinos, and G. Pipeleers, "A Penalty Method Based Approach for Autonomous Navigation using Nonlinear Model Predictive Control," *IFAC-PapersOnLine*, vol. 51, no. 20, pp. 234–240, Jan. 2018.

[47] F. Debrouwere, W. Van Loock, G. Pipeleers, M. Diehl, J. De Schutter, and J. Swevers, "Time-optimal path following for robots with object collision avoidance using lagrangian duality," in *9th International Workshop on Robot Motion and Control*, Jul. 2013, pp. 186–191.

[48] X. Zhang, A. Liniger, and F. Borrelli, "Optimization-Based Collision Avoidance," *arXiv:1711.03449*, Jun. 2018.

[49] P. Mohajerin Esfahani and D. Kuhn, "Data-driven distributionally robust optimization using the Wasserstein metric: performance guarantees and tractable reformulations," *Mathematical Programming*, vol. 171, no. 1, pp. 115–166, Sep. 2018.

[50] I. Yang, "Wasserstein Distributionally Robust Stochastic Control: A Data-Driven Approach," *arXiv:1812.09808*, Dec. 2018.

[51] E. Delage and Y. Ye, "Distributionally Robust Optimization Under Moment Uncertainty with Application to Data-Driven Problems," *Operations Research*, vol. 58, no. 3, pp. 595–612, Jun. 2010.

[52] P. Coppens and P. Patrinos, "Data-driven distributionally robust MPC for constrained stochastic systems," *arXiv:2103.03006*, Mar. 2021. [Online]. Available: http://arxiv.org/abs/2103.03006

[53] A. W. van der Vaart and J. A. Wellner, *Weak Convergence and Empirical Processes: With Applications to Statistics*. New York: Springer, 2000.

[54] G. Wolfer and A. Kontorovich, "Minimax Learning of Ergodic Markov Chains," in *Algorithmic Learning Theory*, Mar. 2019, pp. 903–929.

[55] A. Shapiro, D. Dentcheva, and A. P. Ruszczyński, *Lectures on Stochastic Programming: Modeling and Theory*, ser. MPS-SIAM Series on Optimization. Philadelphia: Society for Industrial and Applied Mathematics : Mathematical Programming Society, 2009, no. 9.

[56] A. Majumdar and M. Pavone, "How Should a Robot Assess Risk? Towards an Axiomatic Theory of Risk in Robotics," in *Robotics Research*, ser. Springer Proceedings in Advanced Robotics, N. M. Amato, G. Hager, S. Thomas, and M. Torres-Torriti, Eds. Cham: Springer International Publishing, 2020, pp. 75–84.

[57] S. T. Rachev, F. J. Fabozzi, and S. V. Stoyanov, *Advanced stochastic models, risk assessment, and portfolio optimization: The ideal risk, uncertainty, and performance measures*. John Wiley & Sons, 2008.

[58] S. Y. Chun, A. Shapiro, and S. Uryasev, "Conditional Value-at-Risk and Average Value-at-Risk: Estimation and Asymptotics," *Operations Research*, vol. 60, no. 4, pp. 739–756, Aug. 2012.

[59] C. A. Hans, P. Sopasakis, J. Raisch, C. Reincke-Collon, and P. Patrinos, "Risk-Averse Model Predictive Operation Control of Islanded Microgrids," *IEEE Transactions on Control Systems Technology*, vol. 28, no. 6, pp. 2136–2151, Nov. 2020.

[60] S. Carpin, Y. Chow, and M. Pavone, "Risk aversion in finite Markov Decision Processes using total cost criteria and average value at risk," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, May 2016, pp. 335–342.

[61] M. P. Chapman, J. Lacotte, A. Tamar, D. Lee, K. M. Smith, V. Cheng, J. F. Fisac, S. Jha, M. Pavone, and C. J. Tomlin, "A Risk-Sensitive Finite-Time Reachability Approach for Safety of Stochastic Dynamic Systems," in *2019 American Control Conference (ACC)*, Jul. 2019, pp. 2958–2963.

[62] G. C. Pflug and A. Pichler, *Multistage Stochastic Optimization*, ser. Springer Series in Operations Research and Financial Engineering. Cham: Springer International Publishing, 2014.

[63] P. Sopasakis, M. Schuurmans, and P. Patrinos, "Risk-averse risk-constrained optimal control," in *2019 18th European Control Conference (ECC)*. Naples, Italy: IEEE, Jun. 2019, pp. 375–380.

[64] T. Brüdigam, R. Jacumet, D. Wollherr, M. Leibold, and F. Borrelli, "Safe Stochastic Model Predictive Control," *arXiv:2204.06207 [cs, eess]*, Apr. 2022.

[65] S. H. Nair, E. H. Tseng, and F. Borrelli, "Collision Avoidance for Dynamic Obstacles with Uncertain Predictions using Model Predictive Control," Aug. 2022.

[66] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "CasADi – A software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, In Press, 2018.

[67] K. Berntorp, R. Bai, K. F. Erliksson, C. Danielson, A. Weiss, and S. D. Cairano, "Positive Invariant Sets for Safe Integrated Vehicle Motion Planning and Control," *IEEE Transactions on Intelligent Vehicles*, vol. 5, no. 1, pp. 112–126, Mar. 2020.

[68] R. Soloperto, J. Köhler, F. Allgöwer, and M. A. Müller, "Collision avoidance for uncertain nonlinear systems with moving obstacles using robust Model Predictive Control," in *2019 18th European Control Conference (ECC)*. Naples, Italy: IEEE, Jun. 2019, pp. 811–817.

[69] J. Köhler, M. A. Müller, and F. Allgöwer, "A Nonlinear Model Predictive Control Framework Using Reference Generic Terminal Ingredients," *IEEE Transactions on Automatic Control*, vol. 65, no. 8, pp. 3576–3583, Aug. 2020.

[70] M. Treiber, A. Hennecke, and D. Helbing, "Congested traffic states in empirical observations and microscopic simulations," *Physical Review E*, vol. 62, no. 2, pp. 1805–1824, Aug. 2000.

[71] P. Trautman and A. Krause, "Unfreezing the robot: Navigation in dense, interacting crowds," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*. Taipei: IEEE, Oct. 2010, pp. 797–803.

[72] A. K. Sampathirao, P. Sopasakis, A. Bemporad, and P. Patrinos, "Distributed solution of stochastic optimal control problems on GPUs," in *2015 54th IEEE Conference on Decision and Control (CDC)*. Osaka: IEEE, Dec. 2015, pp. 7183–7188.

[73] A. K. Sampathirao, P. Patrinos, A. Bemporad, and P. Sopasakis, "Massively parallelizable proximal algorithms for large-scale stochastic optimal control problems," *arXiv:2107.01745*, Jul. 2021. [Online]. Available: http://arxiv.org/abs/2107.01745

[74] "The Intelligent-Driver Model and its Variants," https://traffic-simulation.de/info/info_IDM.html.

Appendix A. Computation of smooth approximations

Lemma. Let $\tilde{\phi}_k : \mathbb{R} \to \mathbb{R}$ be defined as in (7). Then, $\tilde{\phi}_k(x) \geq |x|$ for all $k > 0$ and $x \in \mathbb{R}$.

Proof. We need to show that $b_k \geq \Delta(x) := |x| - x\,\mathbf{tanh}\,(kx), \forall x \in \mathbb{R}$. Since for $k \geq 0$, both $|x|$ and $x\,\mathbf{tanh}(kx)$ are symmetric, so is $\Delta(x)$. Therefore, $\mathbf{sup}_{x\in\mathbb{R}}\,\Delta(x) = \mathbf{sup}_{x\in\mathbb{R}_+}\,\Delta(x)$, i.e., we may assume that $x \geq 0$ and so $|x| = x$. Furthermore, since for all $x \in \mathbb{R}$, $x - x\,\mathbf{tanh}(kx) \leq |x| - x\,\mathbf{tanh}(kx)$,

the maximum of the left-hand side will be attained for nonnegative x. The constraint $x \geq 0$ is therefore redundant. Thus, it suffices to show that $b_k \geq \mathbf{sup}_x\, x - x\,\mathbf{tanh}(kx)$. Let x be a stationary point of $x \mapsto x - x\,\mathbf{tanh}(kx)$, then

$$\begin{aligned} & 1 - \mathbf{tanh}(kx) - \tfrac{kx}{\mathbf{cosh}(kx)} = 0 \\ \Leftrightarrow \quad & \mathbf{cosh}(kx) - \mathbf{sinh}(kx) = kx \\ \Leftrightarrow \quad & x = \tfrac{W_{\mathrm{L}}(1)}{k}. \end{aligned}$$

It is easy to show that the unique stationary point $\frac{W_{\mathrm{L}}(1)}{k}$ is indeed a local maximizer. Since, furthermore, $x - x\,\mathbf{tanh}(kx)$ is smooth and bounded above, it is also the global maximizer. Since $b_k \geq \frac{W_{\mathrm{L}}(1)}{k}$ by construction, it holds that $x\,\mathbf{tanh}(kx) + b_k \geq |x|$. □

Appendix B. Background on ambiguous chance constraints

This section provides background on the usage of the average-value as surrogates for chance constraints. For more details, we refer the reader to [18, 55].

Suppose that nominally (under full knowledge of the distribution p), we would impose chance constraints of the form $\mathbb{P}_p[Z > 0] \leq \alpha$. We can equivalently denote this by $\mathbb{E}_p\left[1_{(0,\infty)}(Z)\right] \leq \alpha$, as illustrated in Fig. B.19. Since this function is nonconvex and discontinuous at zero, it is often approximated by its smallest convex upper bound. It is easy to see that for any $t > 0$, we have $[1 + tZ]_+ \geq 1_{(0,\infty)}(Z)$, which holds with equality for $Z = 1$. Therefore,

$$\alpha \geq \min_{t>0} \mathbb{E}_p\left[[1 + tZ]_+\right] \geq \mathbb{E}_p\left[1_{(0,\infty)}(Z)\right].$$

With some straightforward manipulations, this can be reformulated as

$$\begin{aligned} & \min_{t>0} \mathbb{E}_p\left[[1 + tZ]_+\right] \leq \alpha \\ \iff & \min_{t>0} \tfrac{1}{\alpha}\,\mathbb{E}_p\left[[1 + tZ]_+\right] - 1 \leq 0 \\ \iff & \min_{t>0} \tfrac{1}{\alpha}\,\mathbb{E}_p\left[\left[t^{-1} + Z\right]_+\right] - t^{-1} \leq 0 \\ \iff & \min_{\gamma<0} \tfrac{1}{\alpha}\,\mathbb{E}_p\left[[Z - \gamma]_+\right] + \gamma \leq 0 \qquad \text{(Change of variables } t = -\gamma^{-1}\text{)} \\ \iff & \mathsf{AV@R}_p^{\alpha}[Z] \leq 0, \end{aligned}$$

where the last equivalence follows from the fact that $\mathsf{AV@R}_p^{\alpha}[Z] = \mathbf{min}_{\gamma\in\mathbb{R}}\, \frac{1}{\alpha}\,\mathbb{E}_p\left[[Z - \gamma]_+\right] + \gamma \leq 0$ implies that $\gamma \leq 0$, so the negativity constraint on γ is inactive [55, §6.2.4].

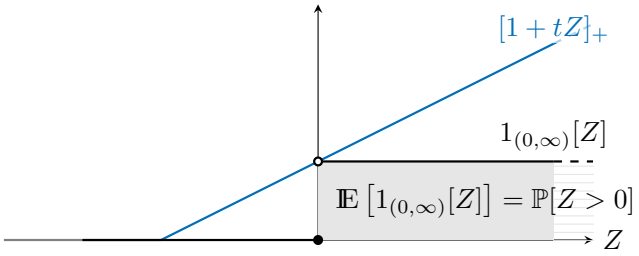


Figure B.19: Graphical interpretation of the AV@R risk measure as a convex overapproximation for chance constraints. The gray area represents $\mathbb{P}[Z > 0]$, which can be equivalently written as the expected value of $1_{(0,\infty)[Z]}$, drawn in black. Drawn in blue is the convex overapproximation $[1 + tZ]_+$ for some $t \in \mathbb{R}_+$. By minimizing over t, the tightest convex overapproximation is obtained [55, 18], which leads to the definition of the average value-at-risk.

Appendix C. Detailed simulator settings

Table C.4: Default settings for the scenario

Setting	**Value**
Sample time T_s [s]	0.2
Transition matrix P	$\left[\begin{smallmatrix} 0.99 & 0.01 \\ 0.05 & 0.95 \end{smallmatrix}\right]$
Ego vehicle length [m]	4.5
Ego vehicle width [m]	1.8
Target vehicle lengths [m]	4
Target vehicle widths [m]	1.9
Max. velocity v_max [m/s]	40.0
Horizon length N	10
Branching horizon N_b	3
Ellipse elongation γ	7
(Nominal) violation rate α	0.05
confidence parameter β	0.05
$k_y, k_{v,x}, k_{v,y}$	1.65, 1.83, 2.62

Table C.5: Parameter values for the IDM model (see [74])

Setting	**Description**	**Value**
T	Time headway with respect to the leading vehicle [s]	0.5
s_0	Desired spatial distance with the leading vehicle [m]	0.5
a_0	Idle acceleration [m/s^2]	0.3
b	Braking deceleration (absolute value) [m/s^2]	0.5
γ	exponent of the free road term	4
$\overline{a}$	Physical limits on acceleration/deceleration [m/s^2]	5

Table C.6: Default settings for the optimal control problem

Setting	**Value**
Min. controls $\underline{u}$	$\begin{bmatrix} -6.4\,\mathrm{m/s^2} & -3^\circ \end{bmatrix}^\top$
Max. controls $\overline{u}$	$\begin{bmatrix} 5.4\,\mathrm{m/s^2} & 3^\circ \end{bmatrix}^\top$
Stage cost state weights Q	$\mathbf{diag}(0, 2, 100, 5)$
Stage cost input weights R	$\begin{bmatrix} 1 & \\ & 10 \end{bmatrix}$
Reference state x_{ref}	$\begin{bmatrix} 0 & 0 & 0 & 30 \end{bmatrix}^\top$