On Integrating POMDP and Scenario MPC for Planning under Uncertainty - with Applications to Highway Driving

Carl Hynén and Daniel Axehill

Hynén, C., Axehill, D., (2022), On Integrating POMDP and Scenario MPC for Planning under Uncertainty - with Applications to Highway Driving, *2022 IEEE INTELLIGENT VEHICLES SYMPOSIUM (IV)*, , 1152-1160. https://doi.org/10.1109/IV51971.2022.9827005

Original publication available at:
https://doi.org/10.1109/IV51971.2022.9827005

http://www.ieee.org/

On Integrating POMDP and Scenario MPC for Planning under Uncertainty – with Applications to Highway Driving

Carl Hynén Ulfsjöö[1] and Daniel Axehill[1]

***Abstract*— Motion planning and decision-making while considering uncertainty is critical for an autonomous vehicle to safely and efficiently drive on a highway. This paper presents a new combined two-step approach for this problem, where a partially observable Markov decision process (POMDP) is tightly coupled with a scenario model predictive control (SCMPC) step. To generate the scenarios in the SCMPC step, the solution to the POMDP is used together with a novel scenario-reduction procedure, which selects a small representative subset of all scenarios considered in the POMDP. The resulting planner is evaluated in a simulation study where the impact of the two-step approach and the scenario-reduction method is shown.**

I. Introduction

The development of advanced driver assistance systems and autonomous vehicles has received much interest over the past decade from industry and academia alike. As the technology develops it has the potential of revolutionizing the transportation sector. However, there are still many technological hurdles to cross before fully autonomous vehicles on public roads will be commonplace. One such hurdle is the autonomous system's ability to reason about the uncertainty in its environment and choose an appropriate action, that considers the uncertainty without becoming overly conservative. Related to this is the decision makers ability to reason about how traffic participants cooperate with each other and with the autonomous vehicle, as this can drastically reduce the uncertainty. This is especially important for heavy vehicles as their slow longitudinal dynamics and large size make it hard to efficiently maneuver if the surrounding vehicles are not cooperative. Investigating this for highway driving such as the situation depicted in Fig. 1, is the main focus of this work.

A. Related work

Many methods have been proposed to solve this planning problem. The more traditional approach is to decompose the problem into separate prediction, behavior planning and trajectory planning stages that operate in a hierarchical fashion [1], [2]. This greatly simplifies the problem and allows the subproblems to be tackled independently, but e.g. makes it hard to explicitly model interactions between the ego and the environment, which can lead to reduced performance [3], [4].

Another common approach is to jointly plan and predict, which means that the planned ego motion is considered while the surrounding agents are predicted. The tightness of the coupling between planning and prediction varies for different methods. One approach typical for partially observable Markov decision process (POMDP) based planning methods [5]–[8] is to predict the surrounding vehicles in closed loop within one planning cycle. An alternative way of coupling the planner and predictor is taken by e.g. [3], where the predictor and planner are separate but the planned ego trajectory is used as an input to the predictor. The largest advantage of the coupled approach is that interactions can be directly included in the plan. The drawback of this is that performing the prediction within one planing cycle, limits which planning and prediction methods that can be used, as it increases the dimension of the configuration space and typically requires lightweight prediction algorithms.

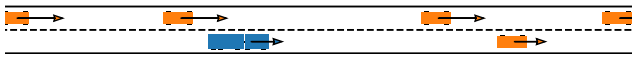


Fig. 1: The investigated driving situation, where the ego vehicle has to overtake a vehicle with faster, high density traffic in the passing lane.

Existing planning methods handle prediction and estimation uncertainty in various ways. The established technique is to only plan while considering one deterministic trajectory for each of the surrounding vehicles, and let feedback and replanning handle disturbances. This can be very effective and is used by many of the early autonomous vehicles [2]. This method is, however, purely reactive as it does not adjust its plan until after a disturbance has affected the system in a measurable way. A more proactive approach is to take the uncertainty into account while predicting and plan while considering that the state of the surrounding vehicles is distributed according to some probability distribution or belongs to some set. This approach is taken by e.g. [3], [9], [10], which use different flavors of model predictive control (MPC) to handle the stochastic prediction. Still, these types of planners do not anticipate that the system will receive more information about the environment in the future, which could lead to conservative behavior. There are many proposed methods to account for future measurements in the plan. In [11] a robust scenario-based approach is taken, where the prediction is represented by multiple possible sets, which have to be avoided. From the time that the sets become disjoint, they are seen as different and are allowed to use different plans. In [12] a similar technique is used, but instead of a robust prediction they use a probabilistic prediction together with stochastic MPC and assume that two modes are distinguishable when the β-confidence ellipsoids are disjoint.

This work was supported by Sweden's Innovation Agency.

[1]Division of Automatic Control, Linköping University, Sweden, (e-mail: {carl.hynen, daniel.axehill}@liu.se).

Finally, there are many existing planners that anticipate future measurements by formulating the problem as a POMDP [5]–[8], [13].

A POMDP is a very natural way of modeling this planning problem as estimation uncertainty, prediction uncertainty and future measurements are all included in the model. POMDPs are generally very hard to solve, however advances in computational power as well as improved online POMDP solvers [14]–[17] have increased their popularity. Despite these developments, large problems are still hard to solve, especially for problems with long planning horizons and large action spaces. For this reason POMDPs typically use a small set of actions and coarse time discretization [7], [8]. However, this can limit the richness of solutions that the solver can find and can make the solution unpleasant to execute. Alternatively, in [5], [6], this is instead alleviated by planning over handcrafted policies and in [18] by using reinforcement learned policies to guide the solver through a larger action space.

B. Contributions

In this work a novel two-step approach to this problem is investigated, where an initial solution to the POMDP is improved by a subsequent scenario MPC (SCMPC) step. The main contributions of this work are:

- A combined two-step POMDP and SCMPC-based planner that uses the solution from the POMDP directly in the SCMPC formulation to tightly couple the two methods.
- A scenario-reduction method that transfers the most important scenarios from the POMDP to the SCMPC.
- An extensive simulation study where the proposed planner is evaluated in a complex highway driving situation, where the advantages of the combined approach and the scenario-reduction method are shown.

II. Problem Formulation

A. Overview

The proposed planning approach consists of two separate steps algorithmic. In the first step a POMDP is solved. In a typical online POMDP solver [16] sampled scenarios are used to represent the uncertainty, and the output from the solver is sequences of optimal action-observation pairs for each scenario. The solution can be represented by a belief tree drawn in Fig. 2, where the optimal solution is highlighted. In the investigated driving situation, the different colored scenarios could e.g. represent uncertainty realizations where an overtake is possible and not possible. Notice that the solver can only choose which action to perform, while the resulting observation is dependent on the realization of the stochastic process, which means that the optimal solution must include all possible observations for a selected action.

The POMDP is able to plan a discrete sequence of actions that considers the uncertainty in the problem, however there are several weak points of this method. Since state-of-the-art online POMDP solvers use particles to represent the uncertainty, a large number of particles can be needed to model the uncertainty sufficiently well. As the planner has to consider each particle during planning, the computational requirements are therefore typically high. This leads to a low planning rate in common POMDP-based planners [7], [13]. Another related problem is that if the action space is large and planning horizon is long, then the belief tree will be large and computationally demanding to fully explore. As the typical way of avoiding this is by simply discretizing time and the control signals coarsely, this leads to a solution that will also be coarse.

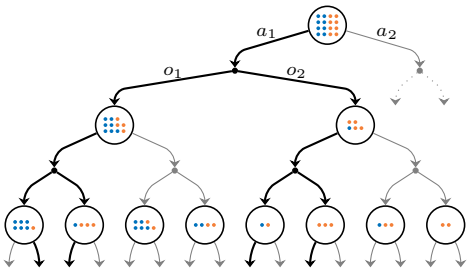


Fig. 2: An example of a belief tree that a tree-search based POMDP solver outputs. The solution is highlighted in black and the colored dots represent different scenarios.

To remedy this problem, a second planning step is proposed in this work that uses SCMPC to improve the solution. The SCMPC problem can use continuous control variables and a finer time discretization than the POMDP, which can improve the rough POMDP solution, while at the same time including beneficial information from it. The SCMPC integrates naturally with the POMDP on three fronts:

- The sampled scenarios from the POMDP can directly be transferred to the MPC problem and information from the POMDP solution can be used to judiciously select the important scenarios.
- The SCMPC problem can be initialized with the solution to the POMDP, which can improve performance and guarantees that the initial solution is feasible.
- Information about which moment in time an uncertainty is known can be transferred from the POMDP, and be modeled using so-called nonanticipatory constraints in the SCMPC. This can be seen in Fig. 2, where certain scenarios are grouped together and are forced to perform the same action.

B. Partially Observable Markov Decision Process

A POMDP is defined by the tuple $\langle \mathcal{X}, \mathcal{A}, \mathcal{O}, T, Z, R, b_0, \gamma \rangle$. The state of the world $\boldsymbol{x}$ belongs to the state space $\mathcal{X}$. At each time-step the ego vehicle can perform an action $\boldsymbol{a}$ from $\mathcal{A}$, and will then receive an observation $\boldsymbol{o}$ from $\mathcal{O}$ together with a reward $R(\boldsymbol{x}, \boldsymbol{a})$. The transition function $T(\boldsymbol{x}', \boldsymbol{x}, \boldsymbol{a}) = p(\boldsymbol{x}'|\boldsymbol{x}, \boldsymbol{a})$ describes the probability of ending up in the state $\boldsymbol{x}'$, and the observation function $Z(\boldsymbol{x}', \boldsymbol{a}, \boldsymbol{o}) = p(\boldsymbol{o}|\boldsymbol{x}', \boldsymbol{a})$ then gives the probability of receiving the observation $\boldsymbol{o}$ after executing the action $\boldsymbol{a}$ in the state $\boldsymbol{x}$. Finally, b_0 represents the initial belief $p(\boldsymbol{x}(t_0))$, and $\gamma \in [0, 1)$ is a discount factor

that balances how much short-term rewards should be favored relative future ones.

The solution to a POMDP is a policy $\pi\colon b \rightarrow \boldsymbol{a}$ that maximizes the expected discounted reward $V_\pi(b)$ of executing that policy from the belief b over an infinite horizon.

$$\pi^*(b_0) = \arg\max_\pi \underbrace{\mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R(\boldsymbol{x}(t), \pi(b(t)))\,|b(0) = b_0\right]}_{V_\pi(b_0)}, \quad (1)$$

In the following sections the different parts of the POMDP model are described.

1) State representation: To include interaction effects in the planning, the states of the nearby vehicles are included in the state representation. The full state vector is then described as

$$\boldsymbol{x} = \begin{bmatrix}\boldsymbol{x}_e^T & \boldsymbol{x}_1^T & \dots & \boldsymbol{x}_M^T\end{bmatrix}^T, \quad (2)$$

where $\boldsymbol{x}_e$ is the state of the ego vehicle, $\boldsymbol{x}_j$, $j \in \{1, \dots, M\}$ is the state of the j:th surrounding vehicle and M is the number of surrounding vehicles considered during planning. The state of the ego vehicle is in turn modeled as

$$\boldsymbol{x}_e = \begin{bmatrix}x_e & y_e & \psi_e & v_e & r_e\end{bmatrix}^T, \quad (3)$$

where x, y, and ψ represent the pose of the vehicle, v the longitudinal velocity, and $r \in \mathbb{Z}$ is the lane that the vehicle tracks (zero at the rightmost lane and increasing to the left). Each surrounding vehicle is modeled similarly

$$\boldsymbol{x}_j = \begin{bmatrix}x_j & y_j & \psi_j & v_j & r_j & \boldsymbol{\theta}_j^T\end{bmatrix}^T, \quad (4)$$

where the parameter vector $\boldsymbol{\theta}$ has been added, which contains the model parameters of that specific vehicle.

2) Action representation: To efficiently solve the POMDP problem the action space must be discrete and its size should be as small as possible to avoid unnecessary branching. The actions are modeled as separate longitudinal actions ($a_{\text{lon}} \in \mathcal{A}_{\text{lon}}$) and lateral lane-change actions ($a_{\text{lat}} \in \mathcal{A}_{\text{lat}}$). Finally, the full action space is defined as $\mathcal{A} = \mathcal{A}_{\text{lon}} \times \mathcal{A}_{\text{lat}}$. The lateral actions are defined as discrete lane-changing decisions, $\mathcal{A}_{\text{lat}} = \{-1, 0, 1\}$, which represent lane change right (LCR), lane keep (LK) and lane change left (LCL) actions respectively. The longitudinal actions are then defined as low-level desired accelerations, $\mathcal{A}_{\text{lon}} = \{-2\,\text{m/s}^2, 0\,\text{m/s}^2, 0.5\,\text{m/s}^2\}$, which represent braking, cruising and accelerating. To further limit the action space only valid lane changes are permitted and lane-change actions can only be combined with the cruising action.

3) Transition function: The vehicles are modeled using a bicycle model in which the dynamics are defined as

$$\dot{x} = v\cos(\psi) \quad (5a)$$

$$\dot{y} = v\sin(\psi) \quad (5b)$$

$$\dot{\psi} = v\kappa \quad (5c)$$

$$\dot{v} = a_{\text{lon}}, \quad (5d)$$

where κ is the signed curvature and a_{lon} is the longitudinal acceleration. The curvature is calculated by assuming that the vehicle tries to follow the center of its current lane using pure pursuit [19]. Additionally, the dynamics of the route state r are defined as $r(t+1) = r(t) + a_{\text{lat}}$, where a_{lat} represents the discrete lane-change decision. To include the bicycle model in the POMDP it is solved numerically using a Runge-Kutta method.

TABLE I: Description and values of all parameters in the prediction model.

Parameter	Description	Value
IDM		
v_0	desired speed	$[22, 28]$
T	minimum safe time ahead	$[1.5, 3.00]$
s_0	minimum distance ahead	5
a	maximum acceleration	1.25
b	comfortable deceleration	2
δ	acceleration exponent	4
MOBIL		
p	politeness factor	$[0, 1]$
b_{safe}	maximum safe deceleration	4
a_{thr}	threshold to consider lane change preferable	0.2
a_{bias}	bias to the rightmost lane	0.25
Yield classifier		
α_{lc}	threshold of lane-change classifier	$[0.2, 1]$

The dynamical behavior is the same for both the ego and the surrounding vehicles, what differs between them is how a_{lon} and a_{lat} are selected. For the ego vehicle they are selected from the action set in Section II-B.2 during the tree search of the POMDP solver, while for the other vehicles they are selected by prediction models.

Prediction of surrounding vehicles: For simplicity, two typical traffic simulation models are used for the prediction of the surrounding traffic participants, namely the intelligent driver model (IDM) [20] for the longitudinal prediction and MOBIL [21] for prediction of lane changes. In addition to these, a simple lane-changing classifier is used to determine if a vehicle will actively yield to a merging vehicle. The parameters of these three models are described in Table I, where the assumed value or range of possible values are also shown. For the ranges it is assumed that the true parameter value is uniformly distributed between the bounds.

IDM is a commonly used traffic simulation model that predicts the longitudinal acceleration of a vehicle by assuming that the vehicle tries to balance driving at its desired velocity and keeping a, according to its own parameters, reasonable distance to the vehicle ahead. With Δs defined as the distance and Δv as the speed difference between the predicted vehicle and the vehicle ahead of it, the desired distance to the vehicle ahead is defined as

$$s^* = s_0 + vT + \frac{v\Delta v}{2\sqrt{ab}}.$$

The predicted acceleration is then given by

$$a_{\text{IDM}} = a\left(1 - \left(\frac{v}{v_0}\right)^\delta - \left(\frac{s^*}{\Delta s}\right)^2\right).$$

MOBIL is traffic simulation model that predicts if a vehicle will perform a lane change. It does this by weighting

the egotistical improvement of a lane change against the slowdown it will cause the rest of the traffic. If this is greater than a threshold and the lane change is safe, it will perform the maneuver. This can be defined by denoting the predicted and following vehicles x_p and x_f, respectively. The acceleration of a vehicle is denoted $a_{\text{idm}}(\cdot)$ if the lane change does not take place, and $a'_{\text{idm}}(\cdot)$ if it does. A lane change is considered preferable for a vehicle if the following conditions are true

$$a'_{\text{idm}}(x_f) \leq b_{\text{safe}} \tag{6a}$$

$$\Delta_{x_p} - p\Delta_{x_f} \geq a_{\text{thr}} - \begin{cases} a_{\text{bias}} & \text{if } r(t+1) = 0, \\ 0 & \text{otherwise,} \end{cases} \tag{6b}$$

where $\Delta_{x_p} = a'_{\text{idm}}(x_p) - a_{\text{idm}}(x_p)$ is the increase in acceleration of the predicted vehicle and $\Delta_{x_f} = a_{\text{idm}}(x_f) - a'_{\text{idm}}(x_f)$ is the decrease in acceleration for the following vehicle.

To model if a vehicle will yield or not, a lane-change classifier is used that predicts the probability that a vehicle is changing lanes to another vehicle's current lane, called $\hat{P}(\text{LC}|\boldsymbol{z})$, where $\boldsymbol{z}$ is a vector that consists of the current pose and velocity of all vehicles.The classifier uses a logistic model trained on the NGSIM I-80 and US-101 freeway driving datasets using the following features: signed lateral distance to the lane center, signed lateral velocity relative the lane center and difference in longitudinal velocity aw well as distance to the vehicle ahead and the two closest vehicles in the adjacent merging lane. Note that this ignores possible usage of turn signals, but they could be added in the classifier. Every vehicle is then assumed to do its best to avoid a collision, which is done by always selecting the vehicle in front of the predicted vehicle as the target vehicle in the IDM. What differentiates different drivers from each other is how they treat potential merging vehicles during the merging procedure. Each driver estimates the probability that a candidate vehicle will merge, and if this probability is above the driver-specific threshold, that is $\hat{P}(\text{LC}|\boldsymbol{z}) > \alpha_{\text{lc}}$, then that vehicle is classified as an additional target vehicle that the IDM should yield for. Depending on the value of this threshold the cautiousness of the driver will vary from the extremes of yielding for every adjacent vehicle at $\alpha_{\text{lc}} = 0$, to only yielding for vehicles directly in front of the predicted vehicle at $\alpha_{\text{lc}} = 1$.

4) Observation function: The key part of the POMDP is the partial observability, i.e., that we have to plan without exactly knowing the state of the system, while at the same time knowing that more information about the state might be available in the future. This information is available in the form of a noisy partial observation of the state. In the POMDP formulation it is modeled as

$$\begin{aligned} \boldsymbol{o} &= \begin{bmatrix} \boldsymbol{o}_1^T & \dots & \boldsymbol{o}_M^T \end{bmatrix}^T \\ \boldsymbol{o}_j &= \begin{bmatrix} x_j & y_j & v_j \end{bmatrix}^T + e_j \\ e_j &\sim \mathcal{N}(0, R_j). \end{aligned}$$

That is, we only directly measure the position and velocity of the vehicles, and the distribution of the remaining state vector has to be inferred from these noisy measurements. Note that only the nearest M vehicles, within $200\,\text{m}$, are included in the state and observation vector.

TABLE II: The weights used by the cost function.

w_{vel}	w_{acc}	w_{brake}	w_{jerk}	w_{lc}	w_{abort}	w_{risk}	w_{bias}	w_{crash}	w_{invalid}
5	4	50	25	25	250	2500	5	10^6	10^6

TABLE III: The parameter values used in the RSS model for the ego and the surrounding vehicles.

	Reaction time	a_{max}	b_{min}	b_{max}
Ego	1.00	0.5	4.0	8.0
Other	1.25	2.0	6.0	8.0

5) Reward function: The reward function in the POMDP is modeled using four main parts, one to reward progress toward the goal, one to penalize non-smooth behavior, one to penalize risky situations and finally one to model hard constraints. These parts are defined as costs, i.e. the goal is to minimize them. Because of this, their signs are flipped in the reward function. The individual parts are defined as

$$J_{\text{vel}} = w_{\text{vel}}|v_e - v_*| \tag{7a}$$

$$J_{\text{smooth}} = \underbrace{w_{\text{acc}} a_{\text{lon}}^2}_{J_{\text{acc}}} + \underbrace{w_{\text{brake}} a_{\text{lon}}^-}_{J_{\text{brake}}} + \underbrace{w_{\text{jerk}} \dot{a}_{\text{lon}}^2}_{J_{\text{jerk}}} + J_{\text{lc}} \tag{7b}$$

$$J_{\text{risk}} = w_{\text{risk}} \left(\frac{(d_{\text{safe,f}} - d_{\text{f}})^+}{d_{\text{safe,f}}} + \frac{(d_{\text{safe,r}} - d_{\text{r}})^+}{d_{\text{safe,r}}} \right) \tag{7c}$$

$$J_{\text{const}} = w_{\text{crash}} \mathbf{1}_{\mathcal{X}_{\text{obs}}}(\boldsymbol{x}_e) + w_{\text{invalid}} \mathbf{1}_{\mathcal{A}_{\text{invalid}}}(\boldsymbol{a}), \tag{7d}$$

where v_* is the desired speed, $(\cdot)^-, (\cdot)^+$ denote the negative and positive part, $d_{\text{safe,f}}$, $d_{\text{safe,r}}$ are the safe front and rear distances, d_f, d_r are the distances to the vehicle ahead and behind, finally $\mathbf{1}_{\mathcal{A}}(x)$ is an indicator function, which is 1 if $x \in \mathcal{A}$ and otherwise 0. The sets $\mathcal{X}_{\text{obs}}$ and $\mathcal{A}_{\text{invalid}}$ represent the states occupied with obstacles and the invalid actions defined in Section II-B.2, respectively. J_{lc} penalizes the solution with w_{lc} when the LCR or LCL action is taken, and with w_{abort} if a lane change is aborted. In addition to these parts, an extra cost of w_{bias} is added when not driving in the rightmost lane to enforce driving norms. The weights of the cost function are presented in Table II and have been selected to roughly rank the importance of the different parts of the cost function.

The safety distances are defined using the Responsibility-Sensitive Safety (RSS) model [22] using the parameters in Table III. The parameters are different for the ego and the surrounding vehicles, which is done to model the fact that the autonomous ego vehicle has a faster minimum reaction time than a human driver. However, as ego is a heavy vehicle, while the surrounding vehicles are assumed to be passenger cars, it has slower longitudinal dynamics. In RSS a vehicle is not responsible for keeping a safe distance to the vehicle behind it, $d_{\text{safe,r}}$ is therefore only used to model the risk introduced by a lane change and is otherwise set to 0. These distances should not be seen as the true RSS distances that we have to enforce to guarantee safety, but as comfortable safety distances for nominal driving. If the true RSS distance is violated the proper response defined by [22] should still be followed, but this is assumed to be enforced by a separate safety system.

C. Scenario Model Predictive Control

Scenario model predictive control uses the typical MPC technique where a finite-horizon optimal control problem (FHOCP) is solved in each sample, and then only the first portion of the solution is executed. It uses multiple scenarios to represent the uncertainty in the system and jointly solves the FHOCPs for the different scenarios, with the initial part of the control signal common for all scenarios.

To reduce the computational complexity of the SCMPC step only the longitudinal state is improved, while the lateral action from the POMDP is assumed to be fixed. In addition to this, the states of the surrounding vehicles are not included in the state space. Instead, they are seen as fixed predictions taken from the solution to the POMDP, and not affected by the ego vehicle's deviation from the POMDP solution. The SCMPC problem is formulated as

$$\min_{\tilde{x},u} \sum_{i=1}^{K} p_i \left(\Phi(\tilde{x}_N^i, p_N^i) + \sum_{k=0}^{N-1} l(\tilde{x}_k^i, u_k^i, p_k^i) \right) \tag{8a}$$

$$\text{s.t. } \tilde{x}_{k+1}^i = f(\tilde{x}_k^i, u_k^i) \tag{8b}$$

$$\tilde{x}_{k+1}^i \in \mathbb{X}_{\text{free}}(p_{k+1}^i) \tag{8c}$$

$$u_{\min} \leq u_k^i \leq u_{\max} \tag{8d}$$

$$\dot{u}_{\min} \leq \dot{u}_k^i \leq \dot{u}_{\max} \tag{8e}$$

$$u_l^i = u_l^j,\ l = 0, \ldots, n_{i,j} \tag{8f}$$

$$\tilde{x}_0^i = \tilde{x}_0, \tag{8g}$$

where (8b) to (8g) should hold for the whole time horizon N ($k \in \{0, \ldots N-1\}$) and all K scenarios ($i, j \in \{1, \ldots K\}$). Subscripts are used to index time and superscripts are used to index the different scenarios. The controlled state for scenario i with the time index k is defined as $\tilde{x}_k^i = \begin{bmatrix} x_k^i & y_k^i & \psi_k^i & v_k^i \end{bmatrix}^T$, the corresponding control signal, u_k^i, is defined as the longitudinal acceleration and p_k^i is the state of the surrounding vehicles, which is seen as a parameter predicted by the POMDP. The probability of a scenario is described by p_i, where $\sum_{i=1}^{K} p_i = 1$. The cost function $l(\tilde{x}_k^i, u_k^i, p_k^i)$ is formulated to be equivalent with the reward function in the POMDP, described in Section II-B.5. However, as the SCMPC only performs longitudinal planning and does not have to enforce constraints with the cost function, it is defined to only include J_{vel}, J_{acc}, J_{brake}, J_{jerk} and J_{risk}. The terminal cost $\Phi(\tilde{x}_N^i, p_N^i)$ is defined similarly but only includes J_{jerk} and J_{risk}. The system dynamics (8b) are defined by numerically simulating the bicycle model defined in Section II-B.3 using a Runge-Kutta method. $\mathbb{X}_{\text{free}}(p_{k+1}^i)$ is the obstacle-free set and is described by a convex polytope. This polytope is estimated by iteratively performing a line search to expand an object-oriented bounding box of the ego vehicle, see [23] for details. The bounds on the control signal are set to $u_{\min} = -2\,\text{m/s}$, $u_{\max} = 0.5\,\text{m/s}$ and $\dot{u}_{\max} = -\dot{u}_{\min} = 1\,\text{m/s}^2$. Finally, x_0 is the initial state of the ego vehicle and (8f) defines the nonanticipatory constraints, where $n_{i,j}$ is a parameter predicted by the POMDP and denotes the time-step after which scenario i and j are distinguishable.

D. Scenario reduction

The size of the SCMPC problem grows as more scenarios are included. One method of keeping the solution time manageable would be to use a distributed algorithm to solve the problem [24]. Another approach, which is taken in this work and can be combined with the former method, is to instead select a smaller representative subset of all the scenarios. This is called scenario reduction and is an important technique to simplify stochastic programs [25], [26]. In this work, it is applied in a novel way, where the large number of scenarios in the sampling-based POMDP solver is reduced to a small user-defined number of scenarios in the SCMPC formulation. The SCMPC formulation differs from the typical problems where scenario reduction is used, which tend to be stochastic linear programs. In addition to this, more information is associated with a scenario from the solution to the POMDP than a traditional scenario. For these reasons, this work focuses more on the practical side of these algorithms.

Let $J = \begin{bmatrix} \xi^1 \ldots \xi^M \end{bmatrix}$ denote the set of all scenarios, where a scenario can be seen as a sequence of states $\boldsymbol{x}$. The scenario-reduction algorithm should then find a subset I with K elements that best represents the set. This can be formulated as

$$\min_{I} \sum_{\xi_1 \in J \setminus I} p(\xi_1) \min_{\xi_2 \in I} d(\xi_1, \xi_2) \tag{9a}$$

$$\text{s.t. } I \subset J,\ |I| = K, \tag{9b}$$

where $p(\xi_1)$ is the probability of scenario ξ_1 and $d(\cdot, \cdot)$ is a function, which gives the distance between two scenarios. This is a combinatorial problem and instead of directly solving it, a forward heuristic is used that greedily adds scenarios from J to the set I that maximizes the reduction of the distance between the sets. For more information, see [25].

To determine the distance between scenarios a probability distribution is associated with each scenario, which represents the estimated distribution of the surrounding vehicles if the scenario were to take place. The distribution is estimated using a particle filter with the scenario as the observation. The distance between two scenarios is then defined as the distance between the associated probability distributions, which is calculated using a symmetrized Kullback-Leibler (KL) divergence. To simplify calculations the probability distributions are approximated as Gaussian distributions, for which the KL divergence admits a closed-form solution. The result of using KL divergence to measure the distance is that two scenarios will be considered more distant if they are different in a state where there is a low level of uncertainty than in a state with large level of uncertainty. For highway driving this means that variation in lateral position is penalized much more than variation in longitudinal position, as the latter is typically more certain.

III. Simulations

The proposed two-step planner is evaluated in the highway driving scenario shown in Fig. 1, where there is a slow vehicle

TABLE IV: The parameters of the POMDP-SCMPC planner.

Parameter	Value
Planning frequency	1 Hz
Tracked vehicles (filter)	10
Filter rate	20 Hz
POMDP	
Number of scenarios	30
Maximum run time	900 ms
Search depth	15
Time discretization	1 s
Discount factor	0.95
Predicted vehicles	4
SCMPC	
Maximum run time	100 ms
Time discretization	0.33 s
Time horizon	7 s

ahead of the ego vehicle with a speed of 60 km/h. The rest of the traffic is traveling at the speed defined in Table I, i.e. an average of 90 km/h, which is also the desired ego vehicle speed. The parameters are randomized for each simulation and the initial position of each vehicle is also randomized but with a constant traffic density of 15 vehicles per km. The simulation environment is based on CARLA and SUMMIT [27], [28].

The states of the nearest vehicles, within 200 m, are tracked using a marginalized particle filter. DESPOT [16] is used to solve the POMDP defined in Section II-B using the nearest vehicles. DESPOT uses an upper and a lower bound to guide the search. The upper bound is calculated by assuming that all surrounding vehicles suddenly disappear, and the lower bound is estimated by performing rollouts using an IDM. To formulate the SCMPC problem, CasADi [29] is used. It is then solved using the nonlinear solver IPOPT [30] with the linear solver MA27. To compensate for the planing delay, the initial belief is simulated forward with the previous plan. The parameters of the planner are detailed in Table IV.

Seven different configurations of the planner are evaluated. To generate a baseline, a configuration without the SCMPC step is used. Furthermore, five different variations of the POMDP-SCMPC planner are evaluated with 1, 3, 5, 8 and 30 scenarios selected by the scenario-reduction algorithm. Finally, to assess the impact of the scenario-reduction algorithm a version of the POMDP-SCMPC planner is used with 5 randomly sampled scenarios. As the vehicle parameters and the initial state are randomized, the different planners are evaluated using simulations with the same initial seed. This reduces the variation, however, note that the planner-simulator system is still not deterministic. The planners are each evaluated on 50 different initial seeds with a simulation time of 60 s. All configurations apart from the POMDP-SCMPC with 30 scenarios can run in real time with a maximum of 100 ms of the planning time allotted to the SCMPC stage. With 30 scenarios, on the other hand, the solution time ranges from 0.5 s to 1 s. For this configuration, the simulation is therefore slowed down to wait for the SCMPC stage to finish.

The results from this are summarized in Table V, where the

TABLE V: Average cost per second of using the POMDP and the POMDP-SCMPC with 1, 3, 5, 8 and 30 scenarios. The POMDP-SCMPC* uses randomly sampled scenarios instead of the scenario-reduction algorithm.

	J	J_{vel}	J_{acc}	J_{brake}	J_{jerk}	J_{risk}
POMDP	40.1	31.8	0.34	1.50	1.18	5.24
POMDP-SCMPC (1)	43.4	31.6	0.25	1.68	1.82	8.13
POMDP-SCMPC (3)	40.0	31.4	0.19	0.80	1.01	6.60
POMDP-SCMPC (5)	37.7	31.4	0.17	0.63	0.81	4.47
POMDP-SCMPC (8)	37.5	31.8	0.16	0.50	0.66	4.43
POMDP-SCMPC (30)	37.2	32.4	0.16	0.56	0.77	3.32
POMDP-SCMPC* (5)	40.1	31.7	0.18	0.76	1.02	6.45

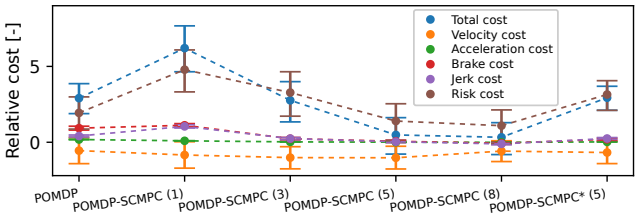


Fig. 3: Difference in cost between the different planner configurations and the POMDP-SCMPC (30) with 95 % confidence intervals. A positive value denotes worse performance compared to POMDP-SCMPC (30).

average cost per second is presented as well as its different parts. In Fig. 3 the average difference in cost between the different planner configurations and the POMDP-SCMPC using all scenarios is plotted together with its 95 % confidence intervals. For conciseness, only the parts related to the longitudinal state of the vehicle are presented, as all versions of the planner are equally successful with merging into the passing lane (31 successes within 60 s with no crashes). Since all configurations use the same POMDP formulation and solver parameters, this is quite natural and indicates that no version of the SCMPC step negatively affects the POMDP's ability to find good lateral actions. In the figures it can be seen that by using the SCMPC with only one scenario from the scenario reduction algorithm, the overall performance degrades compared with the pure POMDP. However, as more scenarios are added to the SCMPC the performance gradually improves, and with five scenarios the cost is significantly lower than with the pure POMDP. The trend continues, and the lowest cost is achieved by the POMDP-SCMPC with all scenarios. This shows that the increased fidelity available to the SCMPC allows it to improve the overall performance, given that enough scenarios are used. By more carefully looking at the different parts of the cost function, more information is revealed. For the POMDP-SCMPC with one scenario it has a lower J_{acc} than the POMDP planner, while J_{brake}, J_{jerk} and J_{risk} are significantly larger. One reason for this is that by only using one scenario, as opposed to the pure POMDP with its 30 scenarios, it is not aware of how uncertain the future prediction actually is. This means that it likely has to use a more reactive approach to handle the uncertainty,

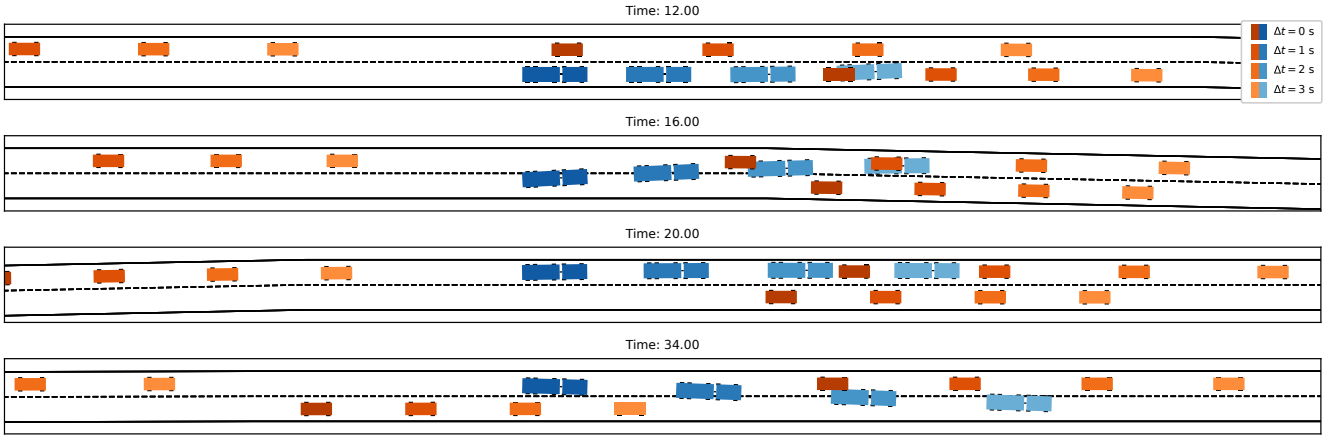


Fig. 4: The position of the ego vehicle plotted in blue and the surrounding vehicles in orange during a merge into the passing lane and during the return to the original lane.

leading to more risky situations, which require more braking. This becomes even more clear when the POMDP-SCMPC with one scenario is compared with the POMDP-SCMPC with more scenarios. As more scenarios are added, J_{acc}, J_{brake}, J_{jerk} and J_{risk} continue to decrease, but after five scenarios all parts of the cost function apart from J_{risk} start to flatten out. This overall pattern is sensible, by using more scenarios the SCMPC is more aware of how uncertain the future is and by being more aware of the uncertainty, it is able to select a control signal that better anticipates the possible future scenarios.

Finally, comparing the POMDP-SCMPC using the scenario-reduction algorithm with using the randomly sampled scenarios, shows some of its advantages. Using five random scenarios results in a comparable total cost to just using three scenarios from the scenario-reduction procedure. Compared with using five scenarios from the reduction algorithm, it performs worse in all regards. This is rather logical, with few random scenarios the solution will vary significantly between planning cycles, leading to rougher indecisive plans.

An example of a successful overtake maneuver is shown in Fig. 4, where a POMDP-SCMPC with five scenarios has been used. In Fig. 5 the ego state for this exact simulation is plotted together with a simulation using the same seed but with only the POMDP. This shows some typical characteristics of the different planners. The POMDP can only use the limited action set, which gives the control signal its step-like shape and does not allow it to accelerate during the lane changes. In contrast to this, the POMDP-SCMPC outputs a smoother acceleration profile as it has a finer time discretization and is not limited to three discrete levels of acceleration. Despite the differences, they share a similar lane action, which mostly differs because the limited action set leads to a slightly lower speed.

The effect of using multiple scenarios in the SCMPC for this exact situation is presented in Fig. 6a, where the ego state is plotted for all scenarios from the POMDP with the five most important scenarios highlighted. Note that this does

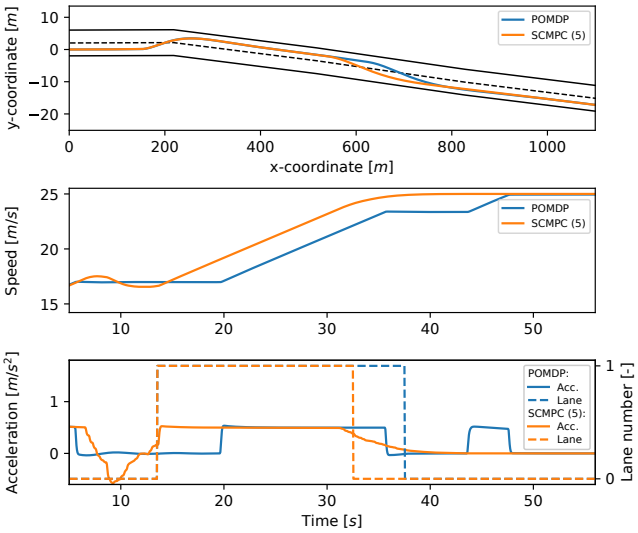
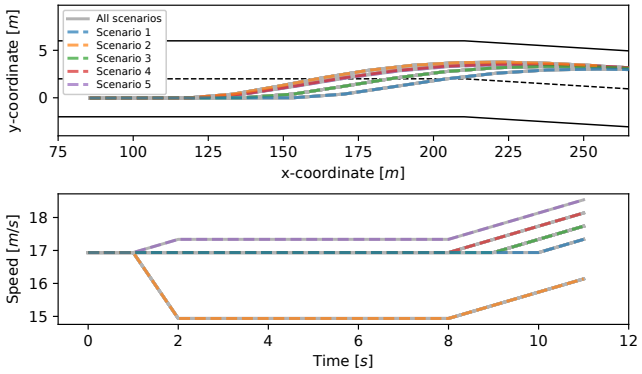
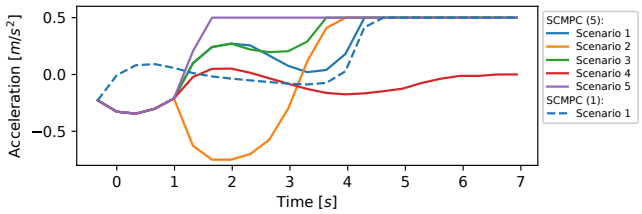


Fig. 5: The position, speed, acceleration and lane action during an overtake for the POMDP and POMDP-SCMPC with five scenarios.

not show the states of the surrounding vehicles, which are used to perform the scenario reduction, but only the state of the ego vehicle. In Fig. 6b the corresponding SCMPC solutions from just using the first scenario and from using the first five scenarios are shown. It can be seen that for the SCMPC with five scenarios, all scenarios have the same control signal during the first second of the plan. After that all scenarios but 1 and 3 are considered to be distinguishable by the POMDP, and are allowed to use different control signals. Finally, after $t = 2\,\text{s}$ all scenarios are distinguishable can use different control signals. This forces the SCMPC to compromise between the different scenarios and find an initial control signal that works for all scenarios. When only one scenario is used the optimal solution is instead to immediately stop the slight braking and only adapt to the first scenario. If any of the other scenarios were to happen this would then



(a) The ego vehicle position and velocity from the POMDP solution with the five most important scenarios highlighted.



(b) The predicted acceleration signal from the SCMPC with five scenarios and with only one.

Fig. 6: The predicted ego state from the POMDP solution and the acceleration signal from the SCMPC in a lane-change maneuver.

result in a reactive response during the next planning cycle, potentially leading to worse performance.

IV. Conclusions

We proposed a new combined two-step POMDP and scenario MPC approach to uncertainty-aware planning. It combines the POMDP's ability to handle general uncertainty and ability to find a global solution, with the SCMPC's ability to use continuous control actions. The two methods are coupled together in a novel way, where a scenario-reduction algorithm extracts the important scenarios from the solution to the POMDP and transfers them to the SCMPC. This allows the SCMPC problem to be solved efficiently with a small number of scenarios. Simulations show that using the combined approach with the scenario-reduction algorithm improves the overall performance. Additionally, by including more scenarios from the POMDP in the SCMPC, it can better anticipate the uncertainty and reduce the risk.

For future work, the long-term goal is to implement the planner on a full-scale test vehicle, however before that some areas need to investigated and improved. Other methods of performing the scenario reduction could be investigated to incorporate more domain knowledge into the algorithm. Additionally, a limitation of the proposed planner is the large computational requirements, and thus the low running frequency. This problem could be alleviated by using a parallel POMDP solver such as HyP-DESPOT [17] and a distributed approach to the SCMPC [24]. Finally, integrating more data-driven prediction models in the POMDP is crucial to handle driving situations where the other traffic participants behave in a less structural manner.

References

[1] B. Paden, M. Čáp, S. Z. Yong, D. Yershov, and E. Frazzoli, "A Survey of Motion Planning and Control Techniques for Self-Driving Urban Vehicles," *IEEE Transactions on Intelligent Vehicles*, vol. 1, no. 1, pp. 33–55, Mar. 2016.

[2] J. Ziegler, P. Bender, M. Schreiber, H. Lategahn, T. Strauss, C. Stiller, Thao Dang, U. Franke, N. Appenrodt, C. G. Keller, E. Kaus, R. G. Herrtwich, C. Rabe, D. Pfeiffer, F. Lindner, F. Stein, F. Erbs, M. Enzweiler, C. Knoppel, J. Hipp, M. Haueis, M. Trepte, C. Brenk, A. Tamke, M. Ghanaat, M. Braun, A. Joos, H. Fritz, H. Mock, M. Hein, and E. Zeeb, "Making Bertha Drive—An Autonomous Journey on a Historic Route," *IEEE Intelligent Transportation Systems Magazine*, vol. 6, no. 2, pp. 8–20, 2014.

[3] B. Zhou, W. Schwarting, D. Rus, and J. Alonso-Mora, "Joint Multi-Policy Behavior Estimation and Receding-Horizon Trajectory Planning for Automated Urban Driving," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, May 2018, pp. 2388–2394.

[4] Y. Luo, P. Cai, A. Bera, D. Hsu, W. S. Lee, and D. Manocha, "PORCA: Modeling and Planning for Autonomous Driving Among Many Pedestrians," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3418–3425, Oct. 2018.

[5] E. Galceran, A. G. Cunningham, R. M. Eustice, and E. Olson, "Multipolicy decision-making for autonomous driving via changepoint-based behavior prediction: Theory and experiment," *Autonomous Robots*, vol. 41, no. 6, pp. 1367–1382, Aug. 2017.

[6] L. Zhang, W. Ding, J. Chen, and S. Shen, "Efficient Uncertainty-aware Decision-making for Automated Driving Using Guided Branching," *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020.

[7] C. Hubmann, J. Schulz, M. Becker, D. Althoff, and C. Stiller, "Automated Driving in Uncertain Environments: Planning With Interaction and Uncertain Maneuver Prediction," *IEEE Transactions on Intelligent Vehicles*, vol. PP, pp. 1–1, Jan. 2018.

[8] Z. N. Sunberg, C. J. Ho, and M. J. Kochenderfer, "The value of inferring the internal state of traffic participants for autonomous freeway driving," in *2017 American Control Conference (ACC)*, May 2017, pp. 3004–3010.

[9] T. Brüdigam, M. Olbrich, M. Leibold, and D. Wollherr, "Combining Stochastic and Scenario Model Predictive Control to Handle Target Vehicle Uncertainty in an Autonomous Driving Highway Scenario," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, Nov. 2018, pp. 1317–1324.

[10] G. Schildbach and F. Borrelli, "Scenario model predictive control for lane change assistance on highways," in *2015 IEEE Intelligent Vehicles Symposium (IV)*, June 2015, pp. 611–616.

[11] 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, July 2021.

[12] S. H. Nair, V. Govindarajan, T. Lin, C. Meissen, H. E. Tseng, and F. Borrelli, "Stochastic MPC with Multi-modal Predictions for Traffic Intersections," *arXiv:2109.09792 [cs, eess]*, Sept. 2021.

[13] H. Bai, S. Cai, N. Ye, D. Hsu, and W. S. Lee, "Intention-aware online POMDP planning for autonomous driving in a crowd," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, May 2015, pp. 454–460.

[14] D. Silver and J. Veness, "Monte-Carlo planning in large POMDPs," in *NIPS*, Jan. 2010, pp. 2164–2172.

[15] H. Kurniawati and V. Yadav, "An Online POMDP Solver for Uncertainty Planning in Dynamic Environment," in *Robotics Research*, M. Inaba and P. Corke, Eds. Cham: Springer International Publishing, 2016, vol. 114, pp. 611–629.

[16] N. Ye, A. Somani, D. Hsu, and W. S. Lee, "DESPOT: Online POMDP Planning with Regularization," *Journal of Artificial Intelligence Research*, vol. 58, pp. 231–266, Jan. 2017.

[17] P. Cai, Y. Luo, D. Hsu, and W. S. Lee, "HyP-DESPOT: A hybrid parallel algorithm for online planning under uncertainty," *The International Journal of Robotics Research*, vol. 40, no. 2-3, pp. 558–573, Feb. 2021.

[18] P. Cai, Y. Luo, A. Saxena, D. Hsu, and W. S. Lee, "LeTS-Drive: Driving in a Crowd by Learning from Tree Search," *Robotics: Science & Systems XV (RSS)*, 2019.

[19] R. C. Coulter, "Implementation of the Pure Pursuit Path Tracking Algorithm," Carnegie-Mellon UNIV Pittsburgh PA Robotics INST, Tech. Rep., Jan. 1992.

[20] 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.

[21] A. Kesting, M. Treiber, and D. Helbing, "General Lane-Changing Model MOBIL for Car-Following Models," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1999, no. 1, pp. 86–94, Jan. 2007.

[22] S. Shalev-Shwartz, S. Shammah, and A. Shashua, "On a Formal Model of Safe and Scalable Self-driving Cars," *arXiv:1708.06374 [cs, stat]*, Oct. 2018.

[23] K. Bergman, O. Ljungqvist, J. Linder, and D. Axehill, "An Optimization-Based Motion Planner for Autonomous Maneuvering of Marine Vessels in Complex Environments," in *2020 59th IEEE Conference on Decision and Control (CDC)*, Dec. 2020, pp. 5283–5290.

[24] C. Phiquepal and M. Toussaint, "Control-Tree Optimization: An approach to MPC under discrete Partial Observability," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. Xi'an, China: IEEE, May 2021, pp. 9666–9672.

[25] W. Römisch, "Scenario Reduction Techniques in Stochastic Programming," in *Stochastic Algorithms: Foundations and Applications*, ser. Lecture Notes in Computer Science, O. Watanabe and T. Zeugmann, Eds. Berlin, Heidelberg: Springer, 2009, pp. 1–14.

[26] H. Heitsch and W. Roemisch, "Scenario tree modeling for multistage stochastic programs," *Math. Program.*, vol. 118, pp. 371–406, May 2009.

[27] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An Open Urban Driving Simulator," in *Proceedings of the 1st Annual Conference on Robot Learning*. PMLR, Oct. 2017, pp. 1–16.

[28] P. Cai, Y. Lee, Y. Luo, and D. Hsu, "SUMMIT: A Simulator for Urban Driving in Massive Mixed Traffic," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, May 2020, pp. 4023–4029.

[29] 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*, vol. 11, no. 1, pp. 1–36, Mar. 2019.

[30] A. Wächter and L. Biegler, "On the Implementation of an Interior-Point Filter Line-Search Algorithm for Large-Scale Nonlinear Programming," *Mathematical programming*, vol. 106, pp. 25–57, Mar. 2006.