Parameter Adjustments in POMDP-Based Trajectory Planning for Unsignalized Intersections

Adam Kollarčík[1,2][a] and Zdeněk Hanzálek[2][b]

[1]*Dept. of Control Engineering, Faculty of Electrical Engineering, Czech Technical University in Prague, Czech Republic*

[2]*Czech Institute of Informatics, Robotics and Cybernetics, Czech Technical University in Prague, Czech Republic*

Keywords: Automated Intersection Crossing, Trajectory Planning, POMDP.

Abstract: This paper investigates the problem of trajectory planning for autonomous vehicles at unsignalized intersections, specifically focusing on scenarios where the vehicle lacks the right of way and yet must cross safely. To address this issue, we have employed a method based on the Partially Observable Markov Decision Processes (POMDPs) framework designed for planning under uncertainty. The method utilizes the Adaptive Belief Tree (ABT) algorithm as an approximate solver for the POMDPs. We outline the POMDP formulation, beginning with discretizing the intersection's topology. Additionally, we present a dynamics model for the prediction of the evolving states of vehicles, such as their position and velocity. Using an observation model, we also describe the connection of those states with the imperfect (noisy) available measurements. Our results confirmed that the method is able to plan collision-free trajectories in a series of simulations utilizing real-world traffic data from aerial footage of two distinct intersections. Furthermore, we studied the impact of parameter adjustments of the ABT algorithm on the method's performance. This provides guidance in determining reasonable parameter settings, which is valuable for future method applications.

1 INTRODUCTION

The advent of autonomous vehicles signals a change in urban mobility, promising to enhance efficiency, safety, and the overall driving experience. One of the challenges these vehicles face is crossing an unsignalized intersection, e.g., a roundabout shown in Figure 1. With the absence of traffic signals and especially without the right of way, the autonomous vehicle must independently determine whether and how to proceed, guided only by traffic regulations and the inherently uncertain behavior of other vehicles, perceived by measurements affected by noise. This is the description of the trajectory planning problem for unsignalized intersections we explore in this paper.

In particular, we investigate a method based on Partially Observable Markov Decision Processes (POMDPs), a framework for planning under uncertainty with incomplete information. For a given intersection and 2D measurements of other vehicles, the method outputs the trajectory as a sequence of accelerations along the desired path.

Our contributions are twofold. At first, we verify the capability of the POMDP-based method to plan collision-free trajectories in simulations using real-life traffic data from aerial footage of two intersections with different topologies[1]. In addition, based on those simulations, we analyze the influence of the parameter values on the method's performance.

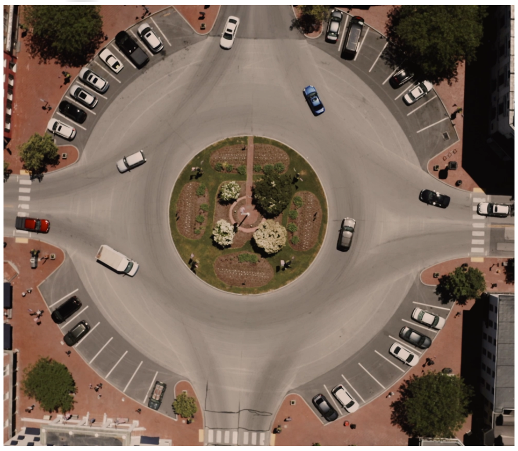
Figure 1: Unsignalized intersection (roundabout).

[a] https://orcid.org/0000-0002-9150-839X

[b] https://orcid.org/0000-0002-8135-1296

[1] See our supplementary YouTube video.

Kollarčík, A. and Hanzálek, Z.
Parameter Adjustments in POMDP-Based Trajectory Planning for Unsignalized Intersections.
DOI: 10.5220/0012742400003702
Paper published under CC license (CC BY-NC-ND 4.0)
In *Proceedings of the 10th International Conference on Vehicle Technology and Intelligent Transport Systems (VEHITS 2024)*, pages 522-529
ISBN: 978-989-758-703-0; ISSN: 2184-495X
Proceedings Copyright © 2024 by SCITEPRESS – Science and Technology Publications, Lda.

2 RELATED WORK

The literature (Eskandarian et al., 2021; Hubmann et al., 2018) broadly divides the planning strategies for automated vehicles into rule-based, reactive, and interactive methods, each with distinct approaches to predicting future events.

The rule-based methods have a predefined set of rules and actions established (Schwarting et al., 2018), e.g., in the form of a finite state machine. Thus their decisions rely only on the current state. The reactive methods treat the motion of vehicles as independent entities. They may introduce simplifying assumptions such as the constant velocity of other vehicles or their set future behavior (Hubmann et al., 2018). Together with the rule-based methods, they lack the ability to consider the interconnected behavior of all vehicles, which is essential for safe interactions in automated driving (Schwarting et al., 2018).

Interactive methods can be either centralized or decentralized (Eskandarian et al., 2021). The centralized methods are usually based on the Vehicle-to-Everything (V2X) paradigm (Tong et al., 2019) and assume communication among vehicles to generate a single global strategy. Nevertheless, given predictions that only half of the vehicles will be autonomous by 2060 (Litman, 2023), decentralized approaches are vital for the foreseeable future. Decentralized methods might be further categorized into three main groups according to their way of tackling interactivity. Those are game theory-based, probabilistic, and data-driven approaches (Schwarting et al., 2018).

This paper investigates a probabilistic planning method developed for unsignalized intersection crossing introduced in (Hubmann et al., 2018) based on POMDPs. The method relies on the Adaptive Belief Tree algorithm (Kurniawati and Yadav, 2016), using a particle filter to estimate the probability distribution of non-observable variables, such as the intentions of other vehicles. In (Bey et al., 2021), the authors employed the method for driving at roundabouts. In (Hubmann et al., 2019), occlusion caused both by the environment and by other dynamic objects was included, and in (Schulz et al., 2019), an improved behavior prediction with dynamic Bayesian network was introduced.

3 APPROACH

This section outlines the components of our approach to addressing the unsignalized intersection crossing problem. We begin by providing a formal problem definition in Section 3.1. Key to our strategy is formulating this problem as a POMDP, which we discuss in Section 3.2. To solve the POMDP, we apply an approximate solver called the Adaptive Belief Tree (ABT) algorithm, detailed in Section 3.3. Our method involves discretizing the intersection topology, a process explained in Section 3.4, where we identify all feasible 2D paths without lane changes based on combinations of intersection entrances and exits. Subsequently, we introduce dynamical and observation models in Sections 3.5, and 3.6, respectively, that capture the complexities of the real world in a simplified yet sufficiently accurate manner for trajectory planning. Entities such as *state* and *observation* are defined, and we construct a reward function in Section 3.7. This function is designed to be maximized within the POMDP, ensuring that the resulting trajectory aligns with our expectations, e.g., encompasses crucial factors such as collision avoidance and adherence to velocity limits. To complete the process, we leverage the mentioned ABT solver for our POMDP formulation to derive the desired trajectory as a sequence of longitudinal accelerations along the desired path toward the goal exit of the intersection.

3.1 Formal Problem Definition

To formally define the problem, we seek a collision-free trajectory through a known intersection topology along a predefined path $\boldsymbol{\phi}(t): t \to \mathbb{R}^2$, $t \in [0,\ d]$ where d is the total length of the path. We assume that the position $\boldsymbol{p}_i = [x_i,\ y_i]^\top$, velocity $\boldsymbol{v}_i = [v_{x,i},\ v_{y,i}]^\top$, unit heading vector $\boldsymbol{\theta_i}$, width w_i, and length l_i measurements of all n relevant other (non-ego) vehicles $i \in \{1,\dots,n\}$ are available every sample time k. Thus, we are looking for a sequence of longitudinal accelerations a_0^k of the ego vehicle (denoted with the zero subscript), such that there is no overlap between the bounding rectangles given by $\boldsymbol{p}_0^k$, $\boldsymbol{\theta}_0^k$, w_0, l_0 and $\boldsymbol{p}_i^k$, $\boldsymbol{\theta}_i^k$, w_i, l_i for each i. Lateral accelerations of all vehicles need not to be directly determined, as the curvature of the followed path implicitly controls it, provided that drifting is avoided.

3.2 POMDPs

POMDPs generalize the Markov Decision Process (MDP) framework for decision-making and planning under uncertainty (Kurniawati, 2022). They are defined by a tuple $\langle S, A, O, T, Z, R, \gamma \rangle$, where S is the set of states, A is a set of actions, O is the set of observations, T is a set of conditional transition probabilities between states, Z is a set of conditional observation probabilities, $R: S \times A$ is the reward function, and $\gamma \in (0,\ 1]$ is the discount factor.

Fundamentally, T represents the stochastic dynamics model of the system, and Z represents the stochastic observation (measurement) model with the probabilities $T(s'|a,s)$ of transition from state $s \in S$ to state $s' \in S$ with action $a \in A$, and $Z(o|a,s')$ of observation $o \in O$ in s'. The particular way of modeling the dynamical and measurement properties for our purposes is described Sections 3.5 and 3.6.

Since the current state is unknown, the *belief* $b \in B$ is introduced, representing a distribution over all states from the set B known as the *belief space*:

$$b(s') \propto Z(o\,|\,a,s') \sum_{s \in S} T(s'\,|\,a,s) b(s), \qquad (1)$$

where $\propto$ denotes that the right-hand side probabilities are not normalized. This converts the original POMDP problem into an MDP where beliefs are states, with continuous and observable state space B.

The goal of the POMDP planner is to determine a policy $\pi : B \to A$, that maximize the expected sum of discounted rewards called the *value function* V:

$$V = \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k R(\boldsymbol{s}_k, a_k) \,\middle|\, b, \pi\right]. \qquad (2)$$

Even though the belief space is continuous, exact POMDP solvers exist. They are based on the fact that any optimal value function is piecewise linear and convex and thus can be described as an upper envelope of a finite set of linear functions called the α-*vectors* (Braziunas, 2003). Nevertheless, exact methods are highly intractable and not suitable for online applications (Kurniawati, 2022). Approximate online methods mostly perform a belief tree search combining heuristics, branch and bound pruning, and Monte Carlo sampling (Ye et al., 2017). An example of these approximate methods is the Adaptive Belief Tree algorithm (ABT) implemented with its variants in a publicly available solver called TAPIR (Klimenko et al., 2014), which we use in our work.

3.3 Adaptive Belief Tree Algorithm

The ABT algorithm (Kurniawati and Yadav, 2016) is an online and anytime POMDP solver that uses the Augmented Belief Tree for finding well-performing policies. The tree $\mathcal{T}$ is constructed by sampling initial belief b_0 with n_{par} particles and propagating them with actions according to the observation and dynamics models T and Z until the set depth of the optimization horizon N or a terminal state is reached. At each step, the particles are assigned a reward $r = R(s,a)$.

Sequences of quadruples (s_i, a_i, o_i, r_i) representing a path in the tree where i is the depth of each contained node are sampled state trajectories also referred to as *episodes*. At last, each episode is assigned a quadruple $(s_N, -, -, r_N)$ where s_N is a sampled next state and r_N is the expected future reward for the following states. This is computed directly for the terminal states; non-terminal states are heuristically estimated, in our case, with a 3-step lookahead. An example of an Augmented Belief Tree is illustrated in Figure 2.

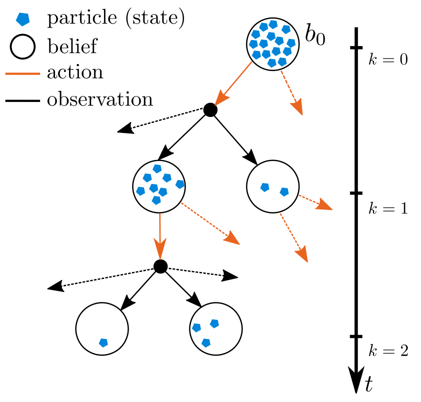

Figure 2: Illustrative example of an augmented belief tree.

With the tree containing n_{ep} episodes, the policy based on a $Q(b,a)$ (*Q-value*) estimate $\hat{Q}(b,a)$ for each belief-action pair is obtained:

$$V(b,\pi) = \max_{a \in \mathcal{T}(b)} \hat{Q}(b,a), \qquad (3)$$

$$\pi(b) = \arg\max_{a \in \mathcal{T}(b)} \hat{Q}(b,a), \qquad (4)$$

where $Q(b,a)$ is the reward for performing an action a in the first step and then behaving optimally. The estimate is computed as an average reward of all episodes $h \in H(b,a)$ containing the pair (b,a) in depth l:

$$\hat{Q}(b,a) = \frac{1}{|H(b,a)|} \sum_{h \in H(b,a)} \left(\sum_{i=l}^{N} \gamma^{i-l} r_i \right). \qquad (5)$$

After the action from $\pi(b)$ is applied and new observation observed, a particle filter update according to (1) is performed, and relevant parts of the tree $\mathcal{T}$ are reused in the next step.

The estimation $\hat{Q}(b,a)$ is used not only for determining the policy $\pi(b)$ but also during the construction of the tree $\mathcal{T}$ to select the actions for propagating particles. First, actions that have not been picked already for the belief are chosen randomly with uniform distribution. When there are no unused actions left, the *upper confidence bound* (UCB) is employed to address the exploration-exploitation trade-off:

$$a_{\text{sel}} = \arg\max_{a \in A} \left(\hat{Q}(b,a) + c \sqrt{\frac{\log \sum_a |H(b,a)|}{|H(b,a)|}} \right), \qquad (6)$$

where c is the tunning UCB parameter.

The current belief is updated after the action $\pi(b_0)$ is applied, and new measurements are obtained. The corresponding branches of the belief tree are reused, with the current belief being the new root of the tree. This process is repeated until the goal is reached. If needed, new particles are generated to maintain the n_{par} number of particles to avoid particle depletion.

3.4 Topology Discretization

To simplify the representation of the intersection within the POMDP and to reduce computational demands, we adopt an approach from (Hubmann et al., 2018). The intersection is condensed into a set of all possible paths, as illustrated in Figure 3. Particularly effective for unsignalized intersections, where only a few paths are reasonable to assume, this approach reduces the description of any vehicle to a triplet of values $\boldsymbol{s}_i = [p_i\,,\ v_i\,,\ \mu_i]^\top$, where $\mu \in \{1,\dots,m\}$ is the intended path out of the m possible paths, p is the position along the path p, and v is the longitudinal velocity. The state of our POMDP is then represented by a vector $\boldsymbol{s}$ containing these values for all $n+1$ vehicles, including the ego vehicle.

To identify possible paths, we utilize the *lanelet2* (Poggenhans et al., 2018) map in the OSM XML format as a source for the full intersection topology. Using the lanelet2 library, we generate a route graph, searching for all possible paths as a sequence of $\mathbb{R}^2$ points. Subsequently, each path is stored as two separate cubic splines for the x and y coordinates. This provides an efficient mapping from the path position to the global coordinates, denoted as $\boldsymbol{\phi}(p) : p \to \mathbb{R}^2$, and mapping to the unit heading $\boldsymbol{\psi}(p) : p \to \boldsymbol{y} \in \mathbb{R}^2 : \|\boldsymbol{y}\| = 1$. For the inverse mapping converting global coordinates back to the position on a path, denoted as $\varepsilon(\boldsymbol{p}) : \boldsymbol{p} \to \mathbb{R}$, we employ a search algorithm to determine the nearest point on the selected path.

3.5 Dynamics Model

As we proceed with the formulation of our POMDP, it is crucial to introduce a dynamical model describing how the state, defined in the previous section, evolves over time. While various mathematical models of different complexities exist for vehicles, we aim for simplicity and reduced computational demand. Thus, we follow the approach from (Hubmann et al., 2018) and adopt a discrete linear point mass model for each vehicle $i \in \{0,\dots,n\}$:

$$\boldsymbol{s}_i^{k+1} = \boldsymbol{A}\boldsymbol{s}_i^k + \boldsymbol{B}u_i^k(\boldsymbol{s}) + \boldsymbol{\eta}\,, \quad \boldsymbol{\eta} \sim \mathcal{N}(\boldsymbol{0},\boldsymbol{Q}), \tag{7}$$

$$\boldsymbol{A} = \begin{bmatrix} 1 & \Delta t & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \qquad \boldsymbol{B} = \begin{bmatrix} \frac{1}{2}\Delta t^2 \\ \Delta t \\ 0 \end{bmatrix}. \tag{8}$$

Here, Δt represents the time between subsequent samples k and $k+1$, and $u(\boldsymbol{s})$ denotes the acceleration of the vehicle. The Gaussian noise $\boldsymbol{\eta}$ with zero mean and covariance matrix $\boldsymbol{Q}$ accounts for model inaccuracies.

The first two rows of the model matrices $\boldsymbol{A}$, $\boldsymbol{B}$ represent the change of position p_i and velocity v_i in time. Upon further inspection, it is also noticeable from their last row that the intention μ_i remains constant. Yet, it is unknown for all vehicles except for the ego. The same applies to acceleration values, which need to be predicted to estimate the future movement of the other vehicles.

In (Hubmann et al., 2018), the authors used a heuristically chosen deceleration when a crash is anticipated for other vehicles, otherwise, the reference velocity is followed. Alternatively, a dynamic Bayesian network prediction is presented in the follow-up paper (Schulz et al., 2019). Another approach (Bey et al., 2021) introduces a version of the intelligent driver model (IDM) (Treiber et al., 2000).

We implemented the IDM model for its simplicity:

$$u_i^k = a_{\max}\rho_{\text{IDM}} + \omega_1\,, \quad \omega_1 \sim \mathcal{N}(0,\sigma_{\omega_1}), \tag{9}$$

where $a_{\max}$ is the maximal acceleration, and ω_1 introduces randomness into the model with variance σ_{ω_1}. The factor ρ_{IDM} is given by the IDM:

$$\rho_{\text{IDM}} = 1 - \left(\frac{v_i^k}{v_{\text{des}}}\right)^{\delta} - \left(\frac{d^*(v_i^k, v_{\text{lead}})}{d_{\text{lead}}}\right)^2, \tag{10}$$

$$d^*(v_i^k, v_{\text{lead}}) = d_{\min} + v_i^k\tau + \frac{v_i^k(v_i^k - v_{\text{lead}})}{2\sqrt{a_{\max}|a_{\min}|}}, \tag{11}$$

where v_{des} is the desired velocity, τ is the time headway, δ is the acceleration exponent, v_{lead} is the velocity of the leading (approached) vehicle, and $a_{\min}$ is the minimal acceleration (maximal deceleration).

The acceleration of the ego vehicle is obtained from the resulting POMDP policy according to (4). The model (7), and (8) together with other vehicles acceleration predictions (9) represent the dynamics model referred to as T in Section 3.2.

3.6 Observation Model

Having established the dynamics model, our focus shifts to the observation model, describing how the observations are obtained from the state of the system. In this context, observations should align with measurements, encompassing variables such as position, velocity, and orientation.

Let vector $\boldsymbol{z}_i^k$ denote the observation of road user i at time step k. These observations are acquired

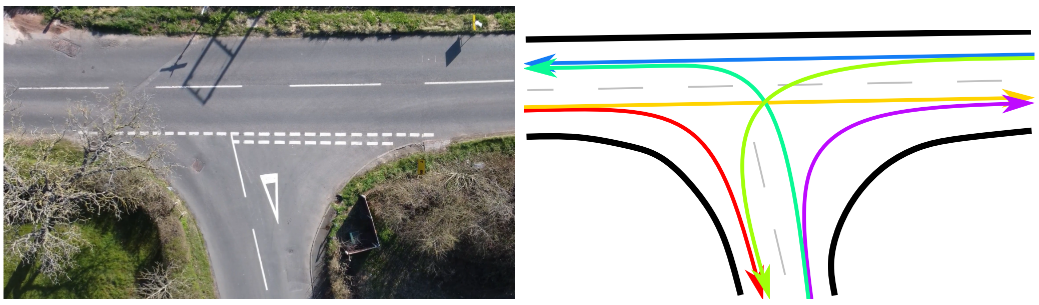

Figure 3: Visualisation of topology discretization – an intersection (left) with all possible simple paths (right).

through spline mappings, as detailed in Section 3.4:

$$z_i^k = \begin{bmatrix} \boldsymbol{\phi}_{\mu_i^k}(p_i^k) \\ v_i^k \boldsymbol{\psi}_{\mu_i^k}(p_i^k) \\ \boldsymbol{\psi}_{\mu_i^k}(p_i^k) \end{bmatrix} + \boldsymbol{\zeta}, \quad \boldsymbol{\zeta} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{R}), \tag{12}$$

where μ_i^k signifies the specific path for the mappings, and $\boldsymbol{\zeta}$ represents Gaussian observation noise with zero mean and covariance matrix $\boldsymbol{R}$.

For particle generation within the particle filter, the estimation of the intended path μ_i^k from global measurements is essential. As the inverse mapping ε has already been defined, the focus is on estimating the intended path μ_i^k. However, this might not be determined uniquely due to potential path overlaps or shared segments, making paths indistinguishable in certain instances. Consequently, a probability distribution is employed to assess the likelihood of each path intention. This involves combining the likelihoods f_1 and f_2 of two features shown in Figure 4:

$$f_1(i,\mu) = e^{\left(-0.05 D_i^k(\mu)^4 + 1\right)}, \tag{13}$$

$$f_2(i,\mu) = e^{\left(3(\alpha_i^k(\mu) - 1)\right)}. \tag{14}$$

The features are the distance from the closest point projection $D_i^k(\mu) = \|\boldsymbol{p}_i^k - \boldsymbol{\phi}_\mu(\varepsilon_\mu(\boldsymbol{p}_i^k))\|$, and a dot product $\alpha_i^k(\mu) = \boldsymbol{\theta}_i^k \cdot \boldsymbol{\psi}_\mu(\varepsilon_\mu(\boldsymbol{p}_i^k))$ of the path heading and measured heading of a vehicle.

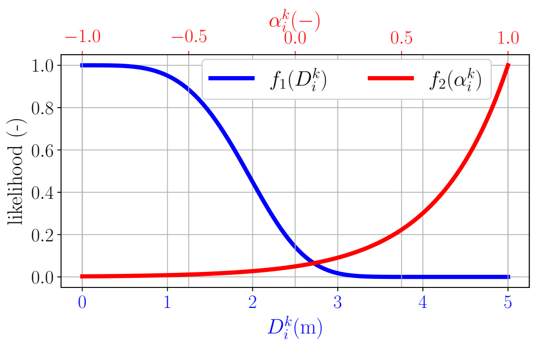

Figure 4: Likelihood functions of features.

The combination of these likelihoods results in the distribution over all path intentions μ:

$$P(i,\mu) = \frac{q(\mu) f_1(i,\mu) f_2(i,\mu)}{\sum_{j=1}^{m} q(j) f_1(i,j) f_2(i,j)}, \tag{15}$$

where $q(\mu)$ is the overlap coefficient, adjusting the probabilities for situations involving overlapping paths, as commonly encountered in roundabouts.

3.7 Reward Function

The reward function is a crucial component that defines the behavior of the ego vehicle within the intersection. It is responsible for aligning the vehicle's actions with our desired objectives: efficient navigation through the intersection while prioritizing safety, avoiding excessive acceleration, and preventing collisions. This multi-goal task is formulated as

$$r^k = r_{\text{vel}}(\boldsymbol{s}_0^k) + r_{\text{acc}}(a_0^k) + r_{\text{crash}}(\boldsymbol{s}^k). \tag{16}$$

The reward at each time step, denoted as r^k, is composed of three fundamental terms: r_{vel} representing velocity, r_{acc} for acceleration, and r_{crash} pertaining to collision avoidance.

The velocity reward encourages the ego vehicle to maintain an appropriate speed throughout its trajectory. It is determined based on the deviation from the reference velocity $\Delta v = v_{\text{des}} - v_0^k$:

$$r_{\text{vel}}(v_0^k) = \begin{cases} -R_{\text{vel}} \Delta v & \text{if } \Delta v \geq 1, \\ -R_{\text{vel}} \Delta v^2 & \text{otherwise}, \end{cases} \tag{17}$$

where R_{vel} is a positive weight. The desired velocity v_{des} is influenced by factors such as the curvature $\kappa_\mu(p_0^k)$ at the vehicle's position p_0^k along its intended path μ_0^k. The desired velocity is determined as:

$$v_{\text{des}} = \min\left(v_{\text{curve}}, v_{\text{lim}}\right), \tag{18}$$

where $v_{\text{curve}} = \sqrt{a_{\text{lat,max}} / \kappa_\mu(p_0^k)}$ is the velocity limited by the maximal acceptable lateral acceleration $a_{\text{lat,max}}$, max, and v_{lim} maximal velocity limit of the intersection.

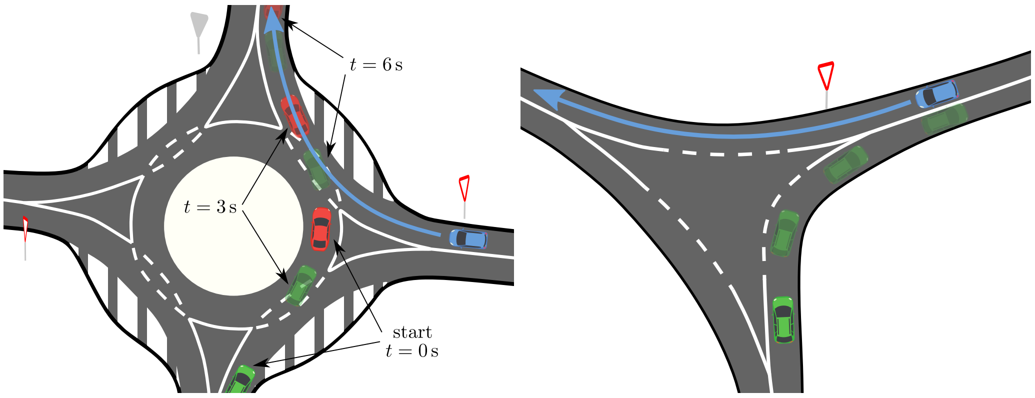

Figure 5: Simulation scenarios – roundabout (left), threeway junction (right). The ego car is represented with a blue color.

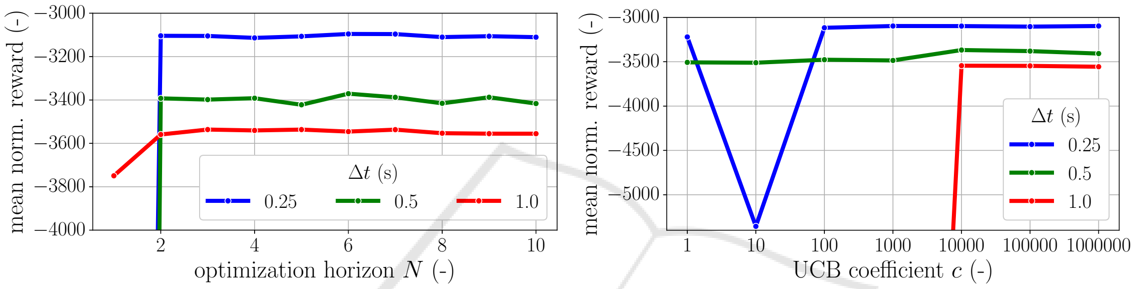

Figure 6: Roundabout rewards – varying optimization horizon N (left), and varying UCB coefficient c (right).

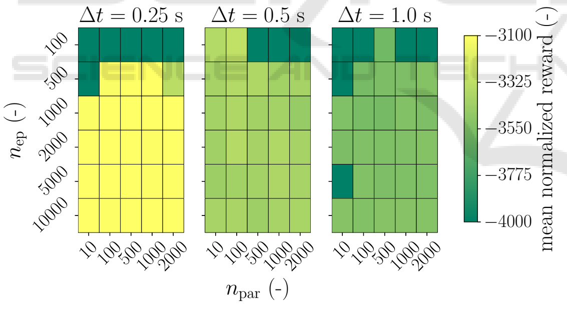

Figure 7: Roundabout rewards for varying n_{par} and n_{ep}.

An acceleration penalty is imposed to prevent abrupt changes in velocity, thus increasing driving comfort. It is computed using a quadratic penalty function, weighted by $R_{\mathrm{acc}} \geq 0$:

$$r_{\mathrm{acc}}(a_0^k) = -R_{\mathrm{acc}} {a_0^k}^2. \tag{19}$$

The collision avoidance component is designed to prevent crashes. It assigns a negative reward $-R_{\mathrm{crash}}$ when a collision is detected and 0 otherwise:

$$r_{\mathrm{crash}}(\boldsymbol{s}^k) = \begin{cases} -R_{\mathrm{crash}} & \text{if ego crashed,} \\ 0 & \text{otherwise.} \end{cases} \tag{20}$$

Collision detection relies on the overlap of bounding rectangles of vehicles, as explained in Section 3.1.

4 RESULTS

In this section, we test the investigated method's collision-free planning capabilities and the impact of parameter adjustments on the planning algorithm's effectiveness. For these proposes, we ran simulations based on data collected from aerial observations of two specific intersections as shown in Figure 5. In the simulations, building upon the results of (Maňour, 2021), we replaced a yielding vehicle with our ego vehicle to mimic real-life situations. Even though the use of such offline data does not include direct reactions of other vehicles to the ego vehicle, we deem it appropriate for testing situations where the ego is expected to yield and adequately respond to the behavior of others. Also, our simulations assume that the ego vehicle adheres to the point mass model (7).

We focus on the Adaptive Belief Tree (ABT) algorithm's parameters: the optimization horizon N, the number of particles n_{par}, the number of episodes n_{ep}, and the Upper Confidence Bound (UCB) parameter c influencing the exploration-exploitation trade-off. These parameters are crucial to the ABT algorithm, and their individual impacts on the performance are not immediately apparent. The default values of all parameters used in our simulations are provided in

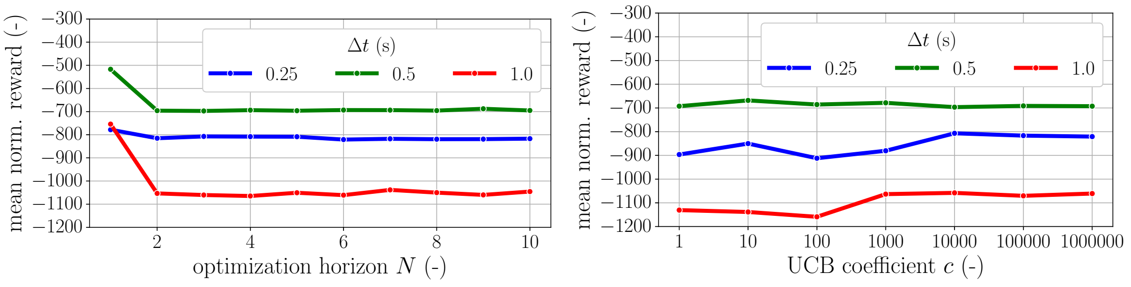

Figure 8: Threeway rewards – varying optimization horizon N (left), and varying UCB coefficient c (right).

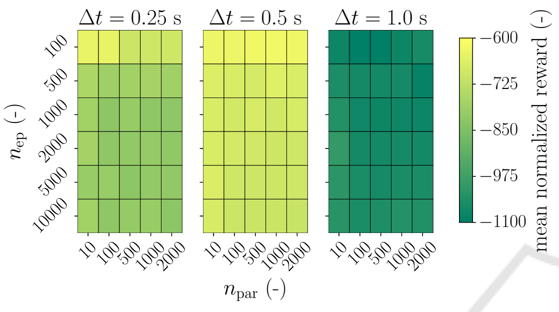

Figure 9: Threeway rewards for varying n_{par} and n_{ep}.

Table 1: Default parameter values.

	Description	Value
A	action set	$\{-2,-1,0,1\}$ m s^{-2}
c	UCB parameter	20000
N	opt. horizon	5
n_{ep}	num. of episodes	3000
n_{par}	num. of particles	300
γ	discount fact.	1
$\boldsymbol{Q}$	dyn. covariance	$\mathbf{0}$
$\boldsymbol{R}$	obs. covariance	$\mathrm{diag}(10^{-2},10^{-2},0)$
σ_{ω_1}	IDM st. dev.	1 m s^{-2}
τ	time headway	2 s
δ	acc. exponent	4
d_{min}	IDM min. dist.	1 m
R_{acc}	acc. reward coef.	1
R_{vel}	vel. reward coef.	100
R_{crash}	crash reward	10000
$a_{lat,max}$	max. lat. acc.	0.5 m s^{-2}

Table 1. We obtained those values by tuning or by following the works they were used in.

For each scenario, we tested every parameter combination in 50 simulations, employing three different time step lengths Δt. The chosen parameter settings significantly influence the simulation runtime, except for the UCB factor c, which does not alter the runtime. To provide a clearer perspective on how these settings impact the simulation duration, Table 2 presents a comparison of roundabout scenario runtime estimates

Table 2: Runtime estimate with realtime percentages.

Δt (s)	Runtime (s) default settings	Runtime (s) $n_{ep} = 10^4$, $n_{par} = 2 \cdot 10^3$
1	6 (75%)	14 (175%)
0.5	10 (125%)	42 (525%)
0.25	24 (300%)	118 (2350%)

for two distinct parameter configurations alongside their corresponding real-time percentages. Our simulation software was implemented in C++, utilizing the TAPIR toolbox together with ROS 1, and we ran the simulations on a laptop with an Intel i5-11500H CPU.

4.1 Roundabout Scenario

The first scenario involves the ego vehicle approaching a roundabout at 8 m s^{-1}, interacting with two vehicles, as seen in the left image of Figure 5. This scenario tests the ability to yield and avoid crashes.

The results, depicted in Figures 6 and 7, show the mean of normalized reward $\Delta t \sum_i r^i$ achieved for various settings. Settings without crashes had values between -3200 and -3600. Mean reward below -4000 represents occurring crashes. The reward is also based on Δt, as the smaller the time step, the higher the reward because the planning algorithm has more samples to react.

Our key findings include identifying parameter threshold values, where their further changes have minimal impact on performance yet significantly increase computation time. The threshold for N is 2; since for $N = 1$, vehicles frequently crash at sample times $\Delta t \leq 0.5$ s. For c, the threshold appears to be 10000, with regular crashes occurring at lower values, mainly for $\Delta t = 1$ s. Crashes also happened commonly for $n_{par} \leq 100$ and for $n_{ep} \leq 500$.

4.2 Threeway Junction Scenario

In the second scenario, the ego vehicle approaches a three-way junction at 8 m s^{-1}, while another vehicle with the right of way also approaches the intersection as depicted in the right image of Figure 5. There is no danger of crashing because the other vehicle turns in a non-collision direction, yet this is an unknown behavior initially. This scenario examines the ego vehicle's information-gathering capability.

Note that the reward here does not indicate the total quality of the solution; rather, it says how well the vehicle follows the reference velocity. Slowing down information gathering is reflected as a lower reward.

This is reflected in our results shown in Figures 8, and 9, where parameter combinations of N, n_{par}, and n_{ep} that caused crashes in the roundabout scenario yield higher rewards as they ignore the possible danger of crashing. The reference velocity is not followed properly for the c values below the threshold, making the rewards lower, especially for $\Delta t = 1$ s, and $\Delta t = 0.5$ s. Also, the rewards for $\Delta t = 0.25$ s are generally lower than for $\Delta t = 0.5$ s, indicating that the cost of delaying a decision when more samples are available is lower relative to the total reward. This behavior might be fine-tuned by parameters R_{acc}, R_{crash} and R_{vel}.

5 CONCLUSION

In this paper, we successfully verified the planning capabilities of the investigated trajectory planning POMDP-based method for unsignalized intersection crossing. We executed a series of simulations based on real-life data from aerial recordings of two distinct intersections. The method proved capable of handling the uncertain aspects of the problem, such as the unknown intention of other vehicles.

Additionally, we investigated the influence of parameter adjustment on the method's performance. Our results indicate that there are certain well-performing threshold values. Exceeding them yields little to no gains and increased computation time. This finding is beneficial with regard to the future application of this method and its derivations.

ACKNOWLEDGEMENTS

This work was co-funded by the European Union under the project ROBOPROX (reg. no. CZ.02.01.0100/22_0080004590) and by the Technology Agency of the Czech Republic under the project Certicar CK03000033.

REFERENCES

Bey, H., Sackmann, M., Lange, A., and Thielecke, J. (2021). POMDP Planning at Roundabouts. In *2021 IEEE IV Workshops*, pages 264–271.

Braziunas, D. (2003). POMDP solution methods.

Eskandarian, A., Wu, C., and Sun, C. (2021). Research Advances and Challenges of Autonomous and Connected Ground Vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 22(2):683–711.

Hubmann, C., Quetschlich, N., Schulz, J., Bernhard, J., Althoff, D., and Stiller, C. (2019). A POMDP Maneuver Planner For Occlusions in Urban Scenarios. In *2019 IEEE IV*, pages 2172–2179.

Hubmann, C., Schulz, J., Becker, M., Althoff, D., and Stiller, C. (2018). Automated Driving in Uncertain Environments: Planning With Interaction and Uncertain Maneuver Prediction. *IEEE Transactions on Intelligent Vehicles*, 3(1):5–17.

Klimenko, D., Song, J., and Kurniawati, H. (2014). TAPIR: A software toolkit for approximating and adapting POMDP solutions online. In *IEEE ICRA*.

Kurniawati, H. (2022). Partially Observable Markov Decision Processes and Robotics. *Annual Review of Control, Robotics, and Autonomous Systems*, 5(1):253–277.

Kurniawati, H. and Yadav, V. (2016). An Online POMDP Solver for Uncertainty Planning in Dynamic Environment. In Inaba, M. and Corke, P., editors, *Robotics Research: The 16th International Symposium ISRR*, Springer Tracts in Advanced Robotics, pages 611–629. Springer International Publishing, Cham.

Litman, T. (2023). Autonomous Vehicle Implementation Predictions: Implications for Transport Planning.

Maňour, Š. (2021). Simulation environment for testing of planning algorithms. Master's thesis, CTU in Prague.

Poggenhans, F., Pauls, J.-H., Janosovits, J., Orf, S., Naumann, M., Kuhnt, F., and Mayr, M. (2018). Lanelet2: A high-definition map framework for the future of automated driving. In *2018 IEEE ITSC*, pages 1672–1679.

Schulz, J., Hubmann, C., Morin, N., Löchner, J., and Burschka, D. (2019). Learning Interaction-Aware Probabilistic Driver Behavior Models from Urban Scenarios. In *2019 IEEE IV*, pages 1326–1333.

Schwarting, W., Alonso-Mora, J., and Rus, D. (2018). Planning and Decision-Making for Autonomous Vehicles. *Annual Review of Control, Robotics, and Autonomous Systems*, 1(1):187–210.

Tong, W., Hussain, A., Bo, W. X., and Maharjan, S. (2019). Artificial Intelligence for Vehicle-to-Everything: A Survey. *IEEE Access*, 7:10823–10843.

Treiber, M., Hennecke, A., and Helbing, D. (2000). Congested traffic states in empirical observations and microscopic simulations. *Physical Review E*, 62(2):1805–1824.

Ye, N., Somani, A., Hsu, D., and Lee, W. S. (2017). DESPOT: Online POMDP Planning with Regularization. *Journal of Artificial Intelligence Research*, 58:231–266.