Worst-case Analysis of the Time-To-React Using Reachable Sets

Sebastian Söntges*, Markus Koschi*, and Matthias Althoff

Abstract—Collision mitigation and collision avoidance systems in intelligent vehicles reduce the severity and number of accidents. To determine the optimal point in time at which such systems should intervene, time-based criticality metrics such as the Time-To-React (TTR) are commonly used. The TTR describes the last point in time along the current trajectory at which an evasive trajectory exists. In this paper, we present a novel approach to determine the point in time after which it is guaranteed that no evasive maneuver exists, i.e., by using reachable sets, we over-approximate the TTR. Our deterministic upper bound of the TTR can be used to trigger a collision mitigation system or to find a feasible emergency maneuver which avoids the collision. We demonstrate the efficient computation of the tight over-approximated TTR in different urban and rural traffic scenarios, and compare our results to an estimated TTR using an optimization-based trajectory planner.

I. INTRODUCTION

A. Motivation

Risk assessment is a crucial component of intelligent vehicles to avoid collisions within and beyond the planning horizon [1]. Advanced driver assistant systems (ADAS) have to reliably determine whether the driver is able to avoid potential collisions. If the assumed motion of the vehicle will (most likely) end in a crash, a collision mitigation system can reduce the severity of the impact. Such systems should only intervene if no evasive trajectory exists so that the driver has control of the vehicle as long as possible and to prevent unnecessary interventions (false positives). However, a system also has to detect every unavoidable collision so that no missed intervention occurs (false negatives). Self-driving vehicles, in addition, can use risk assessment to avoid collisions and to obtain optimal trajectories which are the least critical.

B. Related work

We review existing work in the categories a) detecting unavoidable collisions, b) computing the Time-To-Collision, and c) computing the time until the last evasive maneuver.

a) Detecting unavoidable collisions: Collision mitigation systems only intervene at unavoidable collisions, which are often approximately detected by checking a finite set of possible evasive maneuvers [2], [3]. To describe states in which the system eventually collides regardless of what trajectory it follows, the notion of Inevitable Collision States (ICS) was introduced [4]. In order to guarantee that a collision is unavoidable, one has to employ methods which consider the set of all possible trajectories. For this purpose, reachable sets, which are the set of states reachable for a system subject to a set of inputs, are often used. The work in [5] uses backward reachable sets for the example of a lane departure system. The authors of [6] determine all reachable positions while ignoring the velocity domain, which results in overly large reachable regions. In our previous work [7], we compute an over-approximation of the reachable set considering position, velocity, and acceleration constraints. This over-approximation can be used to determine the nonexistence of evasive trajectories [8].

b) Time-To-Collision: For practical employment, one does not only wish to detect whether a collision is unavoidable given the current state, but rather wants to know the time until a collision when continuing the current trajectory. Time-To-Collision (TTC) denotes the time until impact, given a predicted trajectory of the ego vehicle and of each surrounding object [9]. A worst-case analysis of the TTC is described in [10]. To account for uncertainties, one can use stochastic predictions to obtain a probabilistic TTC [11]–[13].

c) Time until last evasive maneuver: The TTC is not sufficient for collision avoidance, since it provides no information about possible evasive maneuvers. For that reason, the Time-To-React (TTR) has been proposed as the remaining time along the current trajectory until which a collision-free and dynamically feasible trajectory still exists [14]. The authors of [14] define the TTR as the maximum of the Time-To-Brake (TTB), Time-To-Steer (TTS), and Time-To-Kickdown (TTK), which correspond to maximum possible braking, steering, and acceleration trajectories, respectively. These time-based metrics are often generalized as Time-To-X (TTX), i.e., the time remaining for an action X to avoid a collision. Since [14] is only designed for restricted traffic situations with one other object, an iterative search strategy using predefined evasive trajectories is proposed in [15] for scenarios with multiple objects. An active safety system for pedestrian avoidance employing the concepts of TTB and TTS is described in [16]. To consider uncertainties when computing the TTR, one can use probabilistic collision detection systems [17]–[19], which are similar to the probabilistic TTC as described above.

It is difficult to exactly determine the TTR, since all possible evasive trajectories have to be evaluated. If only a finite number of evasive trajectories is considered, it cannot be guaranteed that one has found the latest possible
trajectory. However, we wish to know when to react at the latest, i.e., the earliest point in time at which an evasive trajectory definitely does not exist.

C. Contribution

We propose an efficient method to over-approximate the TTR. Existing sampling-based methods (e.g., [14]–[16]) under-approximate the TTR, since they determine the time at which they can still obtain a feasible evasive trajectory. In contrast, our novel set-based approach determines an over-approximation of the TTR, since by using reachable sets, we determine the time at which it is guaranteed that no evasive maneuver exists.

Given an assumed motion of the vehicle, our upper bound of the TTR makes it possible for collision mitigation systems to know beforehand when, at the latest, to definitely intervene or warn the driver. Similarly, collision avoidance systems to know beforehand when, at the latest, to definitely intervene or warn the driver. Similarly, collision avoidance systems or autonomous vehicles can use the over-approximated TTR as the upper bound when searching for evasive trajectories, since it is guaranteed that no collision-free trajectory exists after that time.

Using our over-approximated TTR, one can now judge the accuracy of existing TTR computations. We show that our upper bound is a tight over-approximation by estimating the accuracy of existing TTR computations. We show that our method is deterministic, i.e., it always returns the upper bound when searching for evasive trajectories, or warn the driver. Similarly, collision avoidance systems to know beforehand when, at the latest, to definitely intervene or warn the driver. Similarly, collision avoidance systems or autonomous vehicles can use the over-approximated TTR as the upper bound when searching for evasive trajectories, since it is guaranteed that no collision-free trajectory exists after that time.

Using our over-approximated TTR, one can now judge the accuracy of existing TTR computations. We show that our upper bound is a tight over-approximation by estimating the accuracy of existing TTR computations. We show that our method is deterministic, i.e., it always returns the upper bound when searching for evasive trajectories, or warn the driver. Similarly, collision avoidance systems to know beforehand when, at the latest, to definitely intervene or warn the driver. Similarly, collision avoidance systems or autonomous vehicles can use the over-approximated TTR as the upper bound when searching for evasive trajectories, since it is guaranteed that no collision-free trajectory exists after that time.

The remainder of this paper is organized as follows: After defining the problem statement in Sec. II, we present our approach in Sec. III. Sec. IV describes the optimization-based trajectory generation we use for comparison. Examples of traffic scenarios in Sec. V illustrate that we can tightly over-approximate the TTR. We conclude our paper in Sec. VI.

II. DEFINITIONS AND PROBLEM STATEMENT

A. Definitions

We model the motion of the vehicle by a dynamical system

$$\dot{x}(t) = f(x(t), u(t)), \quad (1)$$

where $x(t) \in \mathcal{X}$ is the state within the state space $\mathcal{X} \subseteq \mathbb{R}^n$, $u(t) \in \mathcal{U}$ is the input within the set of admissible control inputs $\mathcal{U} \subseteq \mathbb{R}^m$, and $t$ is the time. The solution of (1) for an input trajectory $u(\cdot)$ and an initial state $x_0$ at time $t_0$ is denoted by the state trajectory $x(t; x_0, u(\cdot))$. We further introduce the planning horizon $T$ and the final time $t_f := t_0 + T$. Since we require that possible trajectories are collision-free, the vehicle must avoid the occupancy of (dynamic) obstacles $\mathcal{O}(t) \subseteq \mathbb{R}^2$. Thus, we define the set of all colliding states by

$$\mathcal{F}(t) := \left\{ x(t) \in \mathcal{X} \mid A(x(t)) \cap \mathcal{O}(t) \neq \emptyset \right\}, \quad (2)$$

where $A(x(t)) \subseteq \mathbb{R}^2$ denotes the occupancy of the vehicle on the road. Using the set of colliding states (obtained from a given prediction), we can assess the risk of the current input trajectory $u_c(\cdot) \in \mathcal{U}$ of the vehicle with initial state $x_0 \notin \mathcal{F}(t_0)$:

**Definition 1 (Time-To-Collision)** The Time-To-Collision (TTC) is the maximum time we can continue the current trajectory $u_c(\cdot) \in \mathcal{U}$ before we enter the set of colliding states $\mathcal{F}(\cdot)$:

$$\text{TTC} := \sup_{t_s \in \mathbb{R}} \left\{ t_s - t_0 \mid t_s \in [t_0, t_f], \forall t \in [t_0, t_s] : x(t; x_0, u_c(\cdot)) \notin \mathcal{F}(t) \right\}.$$

*Remark 2 (Existence of collision-free trajectory)* From Def. 2, it immediately follows that a collision-free trajectory exists if and only if the reachable set of the current state $x_0$ is nonempty at the final time $t_f$:

$$\mathcal{R}(t; x_0, t_0) \neq \emptyset \Rightarrow \exists u(\cdot) : \forall t \in [t_0, t_f] : x(t; x_0, u(\cdot)) \notin \mathcal{F}(t).$$

Thus, the TTR can also be expressed in terms of $\mathcal{R}$:

**Proposition 1 (Time-To-React using reachable sets)** The TTR is the last point in time along the current trajectory from which the reachable set is nonempty at the end of the planning horizon:

$$\text{TTR} := \sup_{t_s \in \mathbb{R}} \left\{ t_s - t_0 \mid t_s \in [t_0, t_f], \forall t \in [t_0, t_s] : x(t; x_0, u_c(\cdot)) \notin \mathcal{F}(t) \right\}.$$
Proof: Prop. 1 directly follows from Def. 2 and Def. 3.

In order to efficiently search for the TTR, we want to know the time interval in which the TTR is monotonic with respect to time. Using the TTC from Def. 1 we can express the monotonicity:

**Proposition 2 (Monotonicity of the TTR)** Given a set of colliding states \( \mathcal{F}(\cdot) \) and the current trajectory \( u_c(\cdot) \) of the vehicle starting at \( x_0 \). If there is no emergency trajectory starting from \( u_c(\cdot) \) at \( t_1 \geq t_0 \), there cannot be any trajectory starting from a later point in time \( t_2 \in [t_1, t_0 + \text{TTC}] \):

\[
\begin{align*}
t_0 & \leq t_1 \leq t_2 \leq t_0 + \text{TTC} \leq t_f : \\
\mathcal{R}(t_f ; x(t_1 ; x_0, u_c(\cdot)), t_1) & = \emptyset \Rightarrow \\
\mathcal{R}(t_f ; x(t_2 ; x_0, u_c(\cdot)), t_2) & = \emptyset.
\end{align*}
\]

Proof: From Def. 3 it follows that \( x(t_2 ; x_0, u_c(\cdot)) \in \mathcal{R}(t_2 ; x(t_1 ; x_0, u_c(\cdot)), t_1) \) and thus \( \forall t \in [t_2, t_f] : \mathcal{R}(t ; x(t_2 ; x_0, u_c(\cdot)), t_2) \subseteq \mathcal{R}(t ; x(t_1 ; x_0, u_c(\cdot)), t_1) \).

**B. Problem statement**

In this paper, we want to estimate the Time-To-React according to Prop. 1 for a given obstacle prediction \( \mathcal{O}(t) \) by a strict and tight over-approximation \( \mathcal{TTR}_\text{max} \geq \text{TTR} \), so that we know the maximum time we have to avoid a collision.

To model the motion of the vehicle, we use a velocity- and acceleration-bounded point mass:

\[
f(x, u) = \begin{bmatrix} s_x \\ s_y \\ u_x \\ u_y \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} s_x \\ s_y \\ u_x \\ u_y \end{bmatrix}.
\]

(3)

\( v\text{min}, x \leq v_x \leq v\text{max}, x, \quad |u_x| \leq a\text{max}, \)

\( v\text{min}, y \leq v_y \leq v\text{max}, y, \quad |u_y| \leq a\text{max}, \)

and assume that the occupancy of the vehicle on the road is a circle with radius \( r_\text{ego} \):

\[
\mathcal{A}(x(t)) = \left\{ y \mid y \in \mathbb{R}^2, \left\| \begin{bmatrix} s_x \\ s_y \end{bmatrix} \cdot y \right\|_2 \leq r_\text{ego} \right\}.
\]

(4)

The dynamical system (3) is deliberately simple and cannot accurately model a vehicle in emergency situations. However, with the two basic assumptions of bounded velocity and acceleration, it is a valid abstraction of more accurate vehicle models, i.e., the reachable set of the abstract model contains the reachable set of more accurate models, and we may use Prop. 1 to find a valid over-approximation of the TTR.

**III. OVER-APPROXIMAITON OF THE TTR**

To determine an over-approximative \( \mathcal{TTR}_\text{max} \) according to Prop. 1 we have to compute the reachable set of (3). Since the computation of the exact reachable set is often computationally not feasible, we resort to a method computing an over-approximation (superset) of the reachable set, i.e., \( \forall t \geq 0: \mathcal{R}^\triangledown(t ; x_0, t_0) \supseteq \mathcal{R}(t ; x_0, t_0) \), as described in Sec. III-A.

![Fig. 1. Reachable set approximation for five time steps. Bold arrows indicate which subsets \( B_i^q \) contribute to which subsets \( B_i \) in the succeeding time step \( t_i \).](image-url)

Given a method to compute an over-approximation of the reachable set, we can determine an upper bound of the TTR using Prop. 1. If for a candidate \( t_s \), our over-approximation \( \mathcal{R}^\triangledown(t ; x(t_s ; x_0, u(\cdot)), t_s) \) vanishes at \( t = t_f \), there is no evasive trajectory starting from \( t_s \), and \( t_s - t_0 \) is a valid over-approximative \( \mathcal{TTR}_\text{max} \). To minimize \( \mathcal{TTR}_\text{max} \geq \text{TTR} \), we search the set of candidates \( t_s \in [t_0, t_f] \) to find the earliest at which the reachable set vanishes, as described in Sec. III-B.

**A. Over-approximation of the reachable set**

We compute an over-approximation of the reachable set by using [7], which is briefly described in the following. Our method computes \( \mathcal{R}^\triangledown \) iteratively at discrete points in time \( t_i \). In each iteration, we first propagate \( \mathcal{R}^\triangledown \) one time step forward to obtain the set of states \( \mathcal{X}_i \):

\[
\mathcal{X}_i \supseteq \bigcup_{x_{i-1} \in \mathcal{R}^\triangledown} \bigcup_{u(\cdot) \in \mathcal{U}} x(t_i ; x_{i-1}, u(\cdot)).
\]

(5)

Then, we remove the set of colliding states:

\[
\mathcal{R}^\triangledown \supseteq \mathcal{X}_i \setminus \mathcal{F}(t_i).
\]

(6)

Note that this approach uses several approximations which are necessary for an efficient numerical computation. These are due in particular to an efficient set representation (convex polytopes) and required set operations (e.g., set difference and union). For further details, we refer the reader to [7].

The resulting set \( \mathcal{R}^\triangledown \) is represented by the union of four-dimensional convex polytopes \( B^q_i \) (in the position/velocity domain):

\[
\mathcal{R}^\triangledown = \bigcup_{q \in \mathcal{Q}} B^q_i.
\]

(7)

As an example, Fig. 1 shows the computed reachable set at different time steps in the position domain. Each of the polytopes \( B^q_i \) originates from one or more parents \( B^q_{i-1} \). The relationships between each \( B^q_{i-1} \) and its succeeding \( B^q_i \) can be represented as a directed acyclic graph. Each trajectory which visits the set \( B^q_i \) must have visited one of its parents \( B^q_{i-1} \) and must visit one of its children \( B^q_{i+1} \), as is illustrated in Fig. 1 for \( i = 1, \ldots, 5 \).
B. Search for the minimum TTR$_{\text{max}}$

The reachable set $R^\sim$ allows us to determine whether a candidate $t_* \in [t_0, t_f]$ yields a valid TTR$_{\text{max}}$. To efficiently find the minimum TTR$_{\text{max}}$ in the set of candidates $t_* \in [t_0, t_f]$, we propose to use binary search. Alg. 1 gives an outline of the discrete-time binary search for TTR$_{\text{max}}$. As the upper bound for the search, we use the TTC, since $u_c(\cdot)$ is only collision-free from $t_0$ until $t_0 + $TTC$ (cf. Def. 1 and Prop. 2). The TTC is easily determined using $u_c(\cdot)$ and $F(\cdot)$. We do not refine the lower bound of the search, since this requires additional computing resources; however, one could use the Time-To-Brake as a lower bound. To compute this time, one can start at the TTC and apply the maximum feasible acceleration backwards along the path of $u_c(\cdot)$ until a state $x(t; x_0, u_c(\cdot))$ is reached [20].

### Algorithm 1 Discrete-time binary search for TTR$_{\text{max}}$

**Input:** $x_0$ at $t_0$, $\Delta t$, $t_f$, TTC, $F(t)$, $u_c(\cdot)$

**Output:** TTR$_{\text{max}}$

1. low = 0, high = $\lceil \text{TTC}/\Delta t \rceil$
2. while low < high do
3. mid ← $\lfloor (\text{low} + \text{high})/2 \rfloor$
4. $t_* ← \text{mid} \cdot \Delta t + t_0$
5. if $R^\sim(t_f; x(t_*; x_0, u_c(\cdot)), t_*)$ is not $\emptyset$ then
6. low ← mid + 1
7. else
8. high ← mid
9. end if
10. end while
11. return TTR$_{\text{max}} ← \text{low} \cdot \Delta t$

Usually, the TTR is computed online, i.e., during runtime of the vehicle with regular updates of $x_0$, $u_c(\cdot)$, and $F(t)$. Thus, we can use the TTR$_{\text{max}}$ based on the current information. By refining the previously obtained TTR$_{\text{max}}$, we can enhance the search and save computation time.

IV. ESTIMATION OF THE TTR THROUGH OPTIMIZATION-BASED TRAJECTORY GENERATION

In order to evaluate the tightness of the over-approximation of our proposed set-based algorithm, we compare the upper bound TTR$_{\text{max}}$ (computed with Alg. 1) with an estimate TTR$_c$. We determine TTR$_c$ by searching for the latest point in time from which we can explicitly generate a valid evasive trajectory using the same vehicle model [3]. Suppose we have a set of possible TTR$_c$ candidates. For each candidate, we try to find a feasible trajectory starting from $t_* = t_0 + $TTR$_c$. Finally, we choose the longest TTR$_c$ for which a feasible trajectory can be found. Each trajectory is generated by iteratively solving a convex optimization problem around an initial trajectory guess, as described below.

To generate a collision-free initial trajectory, we use a depth-first search in the reachable set. The initial trajectory guess $[\hat{x}_0, \ldots, \hat{x}_s, \ldots, \hat{x}_n]^T$ is constructed so that it matches the intended trajectory from time $t_0$ until time $t_*$ and lies in the reachable set for the remaining time steps. The optimized trajectory is obtained by displacing the states at the points in time $t_{s+1}, \ldots, t_f$. As shown in Fig. 2(a), we constrain the displacement to be smaller than $r_{s+1}, \ldots, r_n$ to ensure that the initial trajectory is only locally optimized, smooth, and collision-free. Instead of directly using the center points $\hat{x}_i$ for the optimization, we introduce the positions $y_i$, since if $\hat{x}_i$ is close to an obstacle, the allowed displacement $r_i$ would be small and there would only be little space for optimization. We determine $y_i$ and $r_i$ by a local search starting from $\hat{x}_i$, as shown in Fig. 2(b).

The optimization problem is to minimize the absolute maximum acceleration:

\[
\begin{align*}
\text{minimize} & \quad \left\| [u_s, \ldots, u_{n-1}]^T \right\|_\infty \\
\text{subject to} & \quad x_i = Ax_{i-1} + Bu_{i-1}, \\
& \quad x_s = \hat{x}_s, \\
& \quad \begin{bmatrix} v_{\min, x} \\ v_{\min, y} \end{bmatrix} \leq C_1 x_i, \\
& \quad \begin{bmatrix} v_{\max, x} \\ v_{\max, y} \end{bmatrix} \geq C_1 x_i, \\
& \quad \|C_2 x_i - y_i\|_2 \leq r_i, \quad i = s + 1, \ldots, n
\end{align*}
\]

where

\[
A = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} \Delta t^2/2 & 0 \\ 0 & \Delta t^2/2 \\ \Delta t & 0 \\ 0 & \Delta t \end{bmatrix}
\]

and

\[
C_1 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad C_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}
\]
If we obtain a trajectory with \( \| [u_n, \ldots, u_{n-1}]^T \|_\infty \leq a_{\text{max}} \), we accept the trajectory as a valid solution; otherwise, we repeat the optimization with this trajectory as the initial trajectory and with a new set of displacement constraints until the optimization objective cannot be further reduced.

V. Numerical Examples

We demonstrate our computation of the TTR_{\text{max}} and TTR_{\infty} in three scenarios, which are included in the CommonRoad benchmarks\footnote{CommonRoad ID: S=Z_Overtake_1a; based on [23, Fig. 3]} [21]. Tab. I lists the parameters of the numerical examples. We generate the intended trajectory \( u_\text{c}(\cdot) \) such that the ego vehicle follows the center of its current lane with constant velocity. To obtain the set of occupied points of other traffic participants \( \mathcal{O}(t) \) within the planning horizon, we use our prediction tool SPOT [22], which assumes that other vehicles have limited velocity and acceleration and abide by the traffic rules.

A. Two-lane road (Scenario I)

The first, deliberately simple scenario is a rural two-lane road\footnote{commonroad.in.tum.de}. Fig. 3(a) illustrates the initial position of the ego vehicle, its current intended trajectory, and a static obstacle in the lane of the ego vehicle.

We obtain TTR_{\text{max}} = 0.8 s, since it is the first time along the intended trajectory at which the reachable set becomes empty at \( t_f \). Fig. 3(b) depicts the reachable set which is initialized at \( t_s = \text{TTR}_{\text{max}} - \Delta t \). Using the optimization-based trajectory planner, we obtain an evasive trajectory which branches off at TTR_{\infty} = 0.7 s, as shown in Fig. 3(c). The maximum acceleration of this trajectory almost requires the maximum allowed acceleration \( a_{\text{max}} \). When decreasing the time step size to \( \Delta t = 0.01 \) s, we obtain TTR_{\text{max}} = 0.79 s and TTR_{\infty} = 0.72 s, which shows that TTR_{\text{max}} is a rather tight upper bound.

B. Intersection (Scenario II)

Scenario II features an urban intersection, where the ego vehicle intends a left turn\footnote{CommonRoad ID: S=GER_Fbb_1c; based on [15, Sec. IV.-A]}. As shown in Fig. 4(a), an approaching vehicle is predicted to continue straight and another vehicle, whose initial position is located outside of the figure, is predicted to turn right.

Our over-approximation results in TTR_{\text{max}} = 1.1 s and our estimation in TTR_{\infty} = 1.0 s. The reachable set and the optimized trajectory starting at state \( x(t_s; x_0, u_\text{c}(\cdot)) \) along the intended trajectory are depicted in Fig. 4(b)-(d) for \( t_s = \text{TTR}_{\text{max}} - \Delta t \) and different time intervals \( t \). As shown in Fig. 4(b), the reachable set is very small in early time steps, and thus, we do not have much time to react. Once we have evaded the approaching vehicle on the right, we have much space on the road, as shown in Fig. 4(d). Note that we can restrict the reachable set to certain lanes, e.g., lanes with same driving direction, by adding further position constraints.

C. T-Intersection (Scenario III)

Fig. 5(a) illustrates the next urban traffic scenario, where the current trajectory of the ego vehicle continues straight with constant velocity, while three other traffic participants are detected at the T-intersection ahead\footnote{CommonRoad ID: S=GER_Fbb_2b; based on [2, Sec. VI.C]}. Since we are uncertain about the intended maneuver of the other vehicles, the

\begin{table}[h]
\centering
\begin{tabular}{ll}
\hline
Parameter of the Scenarios (S.) I to III. & Value \\
\hline
Initial speed (S. I) & \( v_0 = 20.0 \text{ m/s} \) \\
Initial speed (S. II) & \( v_0 = 7.0 \text{ m/s} \) \\
Initial speed (S. III) & \( v_0 = 14.0 \text{ m/s} \) \\
Minimum velocity (S. I) & \( v_{\text{min},x} = 0.0 \text{ m/s} \) \\
Minimum velocity (S. I) & \( v_{\text{min},y} = -10.0 \text{ m/s} \) \\
Maximum velocity (S. I) & \( v_{\text{max},x} = 25.0 \text{ m/s} \) \\
Maximum velocity (S. I) & \( v_{\text{max},y} = 10.0 \text{ m/s} \) \\
Minimum velocity (S. II, III) & \( v_{\text{min},x/y} = -14.0 \text{ m/s} \) \\
Maximum velocity (S. II, III) & \( v_{\text{max},x/y} = 14.0 \text{ m/s} \) \\
Absolute maximum acceleration & \( a_{\text{evo}} = 10.0 \text{ m/s}^2 \) \\
Radius of vehicle & \( r_{\text{ego}} = 0.9 \text{ m} \) \\
\hline
Parameter of simulation & Value \\
\hline
Initial time & \( t_0 = 0 \text{ s} \) \\
Time horizon & \( T = 3.0 \text{ s} \) \\
Time step size & \( \Delta t = 0.1 \text{ s} \) \\
\hline
\end{tabular}
\caption{Parameters of the Scenarios (S.) I to III.}
\end{table}
occupancy prediction includes full acceleration and braking, and, for the vehicle approaching the intersection, turning left and right.

We obtain $TTR_{\text{max}} = 0.5\text{s}$ and $TTR_{\approx} = 0.3\text{s}$. Fig. 5(b)-(d) depicts the reachable set and optimized trajectory starting at different TTR candidates $t_*$. It can be seen that the reachable set is very small, and thus, only a few evasive maneuvers exist. Note that, as shown in Fig. 5(d), the optimized trajectory starting at $t_* = 0.4\text{s}$ leaves the reachable set, and its maximum acceleration $\|u\|_{\infty} = 10.9\text{m/s}^2$ is larger than $a_{\text{max}}$; thus, this trajectory is not dynamically feasible for our vehicle model, and $TTR_{\approx} = 0.3\text{s}$.

D. Computation times

Next, we examine the computation times required to determine the reachable set for all times from the current candidate $t_*$ until the final time $t_f$ (i.e., line 5 of Alg. 1). Table II compares the computation times of all presented scenarios for different starting times $t_*$. We can see that the computation time of our method drastically decreases for a smaller solution space of the ego vehicle, which is beneficial when trying to efficiently determine the TTR. The computation times have been obtained using a Python/C++
implementation on a machine with a 2.6 GHz Intel Core i7 processor with 20 GB 1600 MHz DDR3 memory. (SPOT requires around 30 ms to compute the occupancy of one traffic participant for the whole planning horizon.)

VI. CONCLUSION AND FUTURE WORK

We present a novel approach to tightly over-approximate the Time-To-React for risk assessment. The proposed method provides an upper bound of the TTR by iteratively computing the set of states reachable by the ego vehicle starting at states along the current trajectory. As soon as the reachable set becomes empty within the planning horizon, an evasive maneuver definitely does not exist. The novelty of our approach is that we obtain a guaranteed over-approximation of the TTR for arbitrary traffic scenarios. Our deterministic approach is independent of the prediction of other objects but can consider uncertainties in their unknown future behavior. Our experiments show that the computation times of our proposed over-approximation of the TTR are very short (below 100 ms) and decrease for more critical situations; thus, our approach is promising for real-time application.

As future work, we wish to define terminal states which can be considered safe for an infinite time horizon so that no finite planning horizon is required.

REFERENCES


