

# Determining the Maximum Time Horizon for Vehicles to Safely Follow a Trajectory

Silvia Magdici, Zhenzhang Ye, and Matthias Althoff

**Abstract**—Dealing with the unknown future behavior of other traffic participants is one of the main challenges when generating safe trajectories for autonomous vehicles. When the ego vehicle (i.e., the vehicle to be controlled) follows a given trajectory, an emergency maneuver should be kept available for all times in order to avoid collisions. However, generating an emergency maneuver for each time step is computationally expensive and often not required. In this paper, we propose an algorithm for determining the maximum time horizon under which the ego vehicle can safely follow a given trajectory. First, an upper and a lower bound of this time horizon are computed. Then, binary search is used to find the maximum time horizon for which safety is still guaranteed. Our algorithm reduces the frequency of generating emergency maneuvers while still guaranteeing collision-free trajectories. The approach is tested on real traffic data, and it is shown that our algorithm indeed reduces the frequency of generating emergency maneuvers compared to previous work.

## I. INTRODUCTION

According to the National Highway Traffic Safety Administration<sup>1</sup>, more than 90% of crashes are caused by human error. Fully autonomous vehicles have a huge potential to reduce crashes by taking over driving duties. However, in order to avoid collisions, safety must be guaranteed when generating trajectories to be followed by autonomous vehicles.

Trajectory planning with obstacle avoidance is already a mature field, whose approaches can be classified as follows: 1) planning in discrete space and 2) planning in continuous space.

1) *Planning in discrete space*: Sampling-based motion planning algorithms such as Rapidly-Exploring Random Trees (RRT) [1]–[3] or Probabilistic Road Maps (PRM) [4]–[8] are particularly efficient for path planning in high-dimensional, non-convex state spaces. To decrease the computational cost required by sampling, predefined and parametrized trajectories, referred to as motion primitives [9], [10] (e.g., turn left, right turn, go straight, etc.) can be used. The construction of formally verified maneuver automata using reachability analysis is investigated in [11]. To find a feasible trajectory, the authors of [12] use a heuristic graph search within the maneuver automaton. In [13], a high level path planner based on motion primitives is used. Then, the planned trajectory is tracked using nonlinear model predictive control (MPC). However, this approach limits the motion of the ego vehicle to a subset of all possible maneuvers.

2) *Planning in continuous space*: To generate trajectories directly in continuous space, elastic bands have been introduced [14]. Elastic bands are paths which can be deformed in order to react to changes in the environment. This approach can be used for emergency maneuver generation [15], trajectory planning [16], or adaptive cruise control [17]. Since a single elastic band might fail to describe a desired path, several elastic bands are generated in [18]. Then, a single path is selected based on a given cost function. However, due to changes in the environment, which determine deformation of the elastic bands, the corresponding trajectory might become unfeasible, and the ego vehicle might not be able to follow it anymore.

For computing optimal trajectories which consider constraints, different approaches such as optimal control or MPC can be used [19]–[21]. In [19], MPC is utilized for lane departure prevention. An algorithm which generates collision-free trajectories in a static environment is proposed in [20]. In [21], collision avoidance is achieved through steering and braking, under the assumption that the obstacles move with constant velocity.

Since most of the real traffic scenarios are dynamic, replanning must be considered in order to avoid collisions. Although most previous work considers this mechanism, it has not analyzed the moment at which replanning should be performed in order to guarantee safety. Instead, replanning is done on the fly when a dangerous situation might already be inevitable. The authors of [22] propose an algorithm that computes an adaptive time horizon which dictates when replanning should be performed. The so-called “time to potential failure” determines how long the current trajectory is safe under some given assumptions. However, if no further safe trajectory is found, a collision might be imminent. To cope with the dynamic environment, the authors of [23] propose an adaptive planning horizon computation based on the changing rate of the environment configuration. However, this approach cannot guarantee that another feasible maneuver exists after the computed time horizon.

The above methods cannot ensure safety for every scenario due to possible unexpected maneuvers of other traffic participants. Therefore, in our previous work [24], a fail-safe motion planner is proposed, which is recalled in Sec. IV. The proposed fail-safe motion planner in [24] can ensure safety regardless of the maneuvers performed by other traffic participants. Nonetheless, generating an emergency maneuver for each time step is computationally expensive and typically not required. In this paper, we propose an extension to our fail-safe motion planner: *an algorithm for determining the*

Silvia Magdici, Zhenzhang Ye, and Matthias Althoff are with the Department of Computer Science, Technische Universität München, 85748 Garching, Germany, {silvia.magdici, zhenzhang.ye, matthias.althoff}@tum.de

<sup>1</sup><http://www.nhtsa.gov>

maximum time horizon  $t^*$  to safely follow a given trajectory. A fail-safe maneuver starting at  $t^*$  is guaranteed to exist.

The remainder of this paper is organized as follows: In Sec. II the system dynamics is presented, together with the considered constraints. Preliminaries and the problem statement are introduced in Sec. III. In Sec. IV, the proposed algorithm is described and the computation of  $t^*$  is explained in detail. Numerical simulations are presented in Sec. V together with discussions. Finally, the conclusions are given in Sec. VI.

## II. SYSTEM MODELING

In this section, the system dynamics of the ego vehicle taken from [21] is introduced:

$$\begin{aligned} \dot{s}_x &= v \cos\theta, \quad \dot{s}_y = v \sin\theta, \\ \dot{\theta} &= \frac{v \delta}{l \left[ 1 + \left( \frac{v}{v_{ch}} \right)^2 \right]}, \\ \dot{\delta} &= u_1, \quad \dot{v} = u_2, \end{aligned} \quad (1)$$

where  $s_x, s_y$  are the coordinates of the ego vehicle's position,  $\theta$  is the orientation,  $\delta$  is the steering angle, and  $v$  is the velocity. The system's inputs  $u_1$  and  $u_2$  are the steering rate and the acceleration, respectively. In addition, two more parameters are used: the wheel base  $l$  and the characteristic velocity  $v_{ch}$ . The state of the ego vehicle at time  $t$  is a vector  $x = [s_x, s_y, \theta, \delta, v]^T$ , and the controlled variable at time  $t$  is  $u = [u_1, u_2]^T$ .

In order to consider the physical limitations of a vehicle, the following constraints are imposed:

$$0 \leq v \leq v_{\max}, \quad (2)$$

$$\delta_{\min} \leq \delta \leq \delta_{\max}, \quad (3)$$

$$\dot{\delta}_{\min} \leq u_1 \leq \dot{\delta}_{\max}, \quad (4)$$

$$a_{\min} \leq u_2 \leq a_{\max}, \quad (5)$$

where the boundary values are assumed to be given. In the following, we denote by  $\mathcal{X} \subset \mathbb{R}^5$  the set of states which satisfies the inequalities (2)-(3);  $\mathcal{U} \subset \mathbb{R}^2$  represents the set of inputs that satisfy the inequalities (4)-(5).

## III. PRELIMINARIES AND PROBLEM STATEMENT

Let us first provide some preliminaries and the notation used throughout this paper. In the following, the area defined by the left and right boundary of a road is referred to as *lane*. Let  $\Gamma(x(t)) : \mathbb{R}^5 \rightarrow P(\mathbb{R}^2)$  be the occupancy of a vehicle at time  $t$ , defined as the rectangle which encloses the vehicle body, where  $P(\cdot)$  denotes the power set. The occupancy  $\Gamma(x(t))$  is computed based on the position, orientation, length, and width of the ego vehicle. To ensure that the vehicle is not driving outside the lane boundaries, an additional constraint is added to the previous set of constraints (2)-(5):

$$\forall t \Gamma(x(t)) \subset \textit{lane}. \quad (6)$$

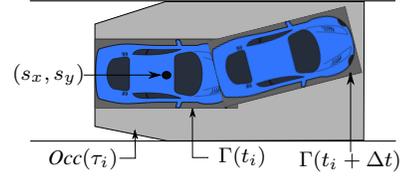


Fig. 2: Occupancy and overapproximative occupancy of a vehicle.

Let  $p$  be the number of the considered surrounding vehicles. Here, we only consider the surrounding vehicles which are in front of the ego vehicle, in the same lane or in adjacent lanes. The vehicles which are behind the ego vehicle are not considered, since each vehicle should avoid collisions with the vehicles driving ahead.

To cope with uncertainties introduced by the unknown behavior of other traffic participants and by the measurements of the environment (e.g. static obstacles, surrounding vehicles), an algorithm for computing the overapproximation of the future occupancies of other traffic participants is proposed in [25] and implemented by the tool SPOT [26]. We denote by  $Occ_k(\tau_i)$ ,  $k \in \{1, 2, \dots, p\}$  the overapproximative predicted occupancy of a surrounding vehicle  $k$ . This is computed for a given time interval  $\tau_i = [t_i, t_i + \Delta t]$ , which encloses all possible occupancies complying with traffic rules, so that  $\Gamma(\tau_i) \subset Occ(\tau_i)$ , where  $\Gamma(\tau_i) := \bigcup_{t \in \tau_i} \Gamma(t)$ , as illustrated in Fig. 2.

Let us assume that an optimal trajectory  $[x_{\text{opt}}(t_0), x_{\text{opt}}(t_0 + \Delta t), \dots, x_{\text{opt}}(T_h)]$  is given, where  $\Delta t$  is the time step and  $T_h$  is the time horizon,  $T_h = m \cdot \Delta t$ ,  $m \in \mathbb{N}$ . Similarly, we write for the emergency maneuver initiated at time  $t$ ,  $\mathcal{X}_{\text{emg}}(t) = [x_{\text{emg}}(t), x_{\text{emg}}(t + \Delta t), \dots, x_{\text{emg}}(T_{\text{emg}})]$ , where  $T_{\text{emg}} = n \cdot \Delta t$ ,  $n \in \mathbb{N}$ , represents the time horizon for generating the emergency maneuver.

Recall that our goal is to determine the maximum time horizon  $t^*$  for which the ego vehicle can follow a given optimal trajectory while guaranteeing safety. To guarantee safety, the following two constraints are considered:

- 1) The occupancy associated with states within the optimal trajectory  $\Gamma(x_{\text{opt}}(\tau_i))$ ,  $\forall i \in \{0, \dots, m\}$ ,  $t_i \leq t^*$  should not intersect with the corresponding overapproximative occupancy set of other traffic participants  $Occ_k(\tau_i)$ .
- 2) There exists an emergency maneuver initiated at  $t^*$  whose corresponding occupancy  $\Gamma(x_{\text{emg}}(\tau_j))$ ,  $\forall j \in \{i, \dots, n\}$  does not intersect with the corresponding overapproximative occupancy set of other traffic participants  $Occ_k(\tau_j)$ ,  $t^* \leq t_j \leq T_{\text{emg}}$ ,  $k \in \{1, \dots, p\}$ , as illustrated in Fig. 1d.

Using the above notations, the addressed problem can be formulated as follows:

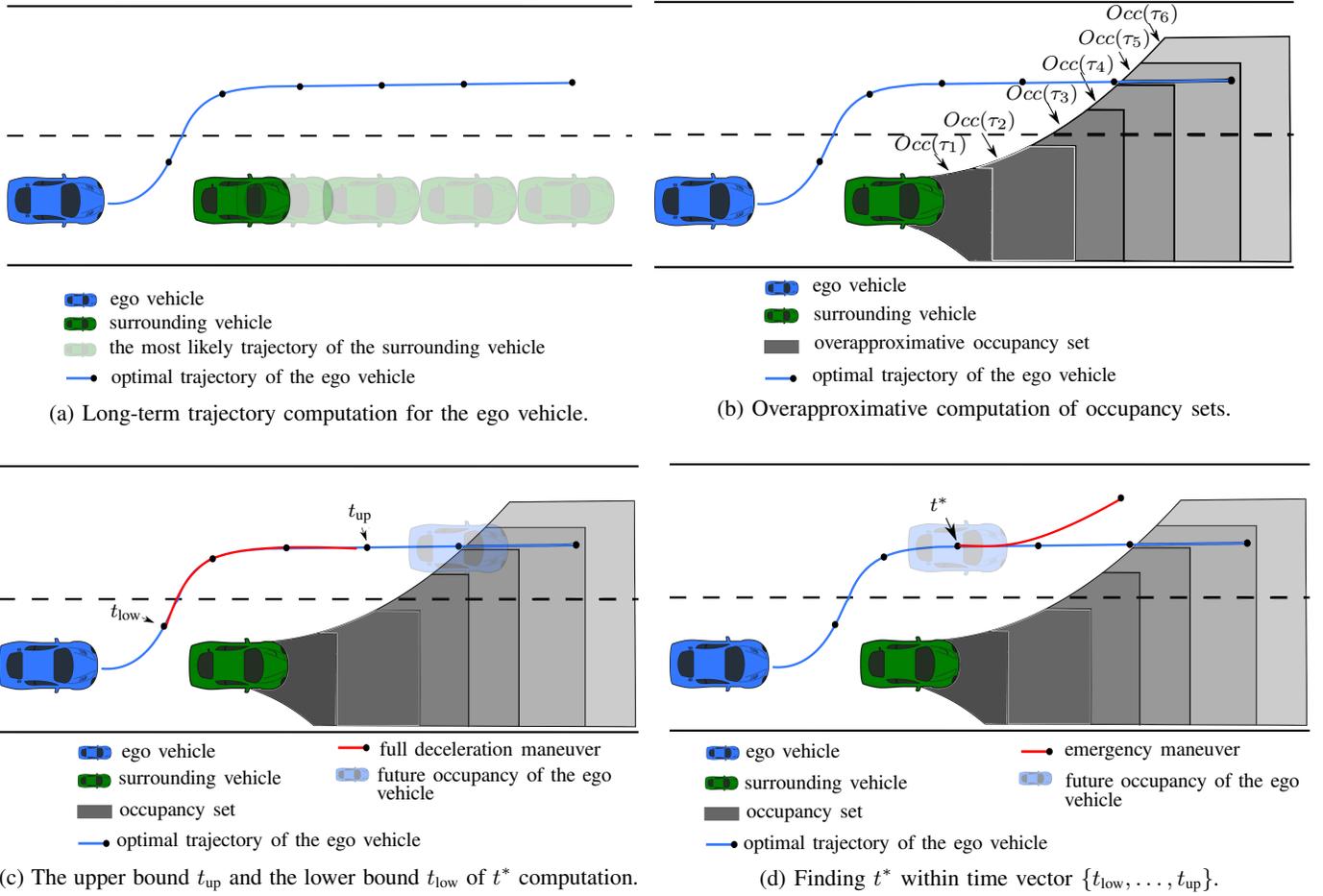


Fig. 1: The main steps of the proposed approach.

$$\begin{aligned}
 t^* &= \max_{0 \leq i \leq m} t_i, \\
 \text{subject to } & \forall k \in \{1, \dots, p\} \forall r \in \{0, \dots, i-1\}, \\
 & \forall j \in \{i, \dots, n\}, \exists \mathcal{X}_{\text{emg}}(t_i) : \\
 & \Gamma(x_{\text{opt}}(\tau_r)) \cap Occ_k(\tau_r) = \emptyset \quad \wedge \\
 & \Gamma(x_{\text{emg}}(\tau_j)) \cap Occ_k(\tau_j) = \emptyset.
 \end{aligned} \tag{7}$$

#### IV. COMPUTING THE MAXIMUM TIME HORIZON $t^*$

To solve (7), we propose an algorithm which can be summarized in three main steps, as illustrated in Fig. 3. After new measurements are collected, the most likely trajectories of surrounding vehicles are predicted by utilizing one of various existing approaches (e.g. constant yaw rate and acceleration [27], maneuver recognition module [28]). In this work, the maneuver recognition module (MRM) [28] is used to generate the most likely trajectory of other vehicles.

Then, a long-term trajectory  $x_{\text{opt}}(t_i), \forall i \in \{1, \dots, m\}$  based on the most likely trajectory of other traffic participants is generated (see Fig. 1a). To generate a long-term plan for the ego vehicle, an RRT-based method [29] is applied. Other trajectory generating algorithms can be used as well.

Next, an upper bound  $t_{up}$  of  $t^*$  is determined, which represents the maximal time for which the ego vehicle can follow the long-term trajectory without intersecting with the corresponding occupancy set of other vehicles. To further prune the search interval of  $t^*$ , a lower bound  $t_{low}$  is computed, which represents the latest time at which full braking can be initiated so that standstill is reached before or at  $t_{up}$ . Both  $t_{up}$  and  $t_{low}$  are illustrated in Fig. 1c. Finally, the maximum time horizon  $t^*$  is calculated using binary search [30] within the interval  $[t_{low}, t_{up}]$ . In the following, a detailed description of every step is provided.

##### ① Compute the upper bound $t_{up}$

The upper bound  $t_{up}$  is the maximum time for which the planned trajectory of the ego vehicle does not intersect with the corresponding occupancy prediction of surrounding vehicles, which is formalized as follows:

$$\begin{aligned}
 t_{up} &= \max_{0 \leq i \leq m} t_i \\
 \text{subject to } & \forall k \in \{1, \dots, p\}, \forall r \in \{0, \dots, i-1\} : \\
 & \Gamma(x_{\text{opt}}(\tau_r)) \cap Occ_k(\tau_r) = \emptyset.
 \end{aligned} \tag{8}$$

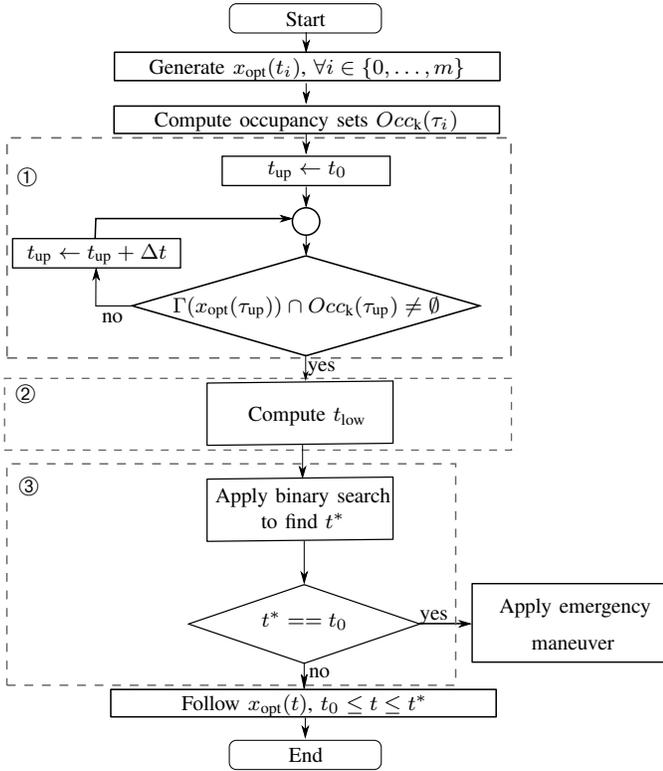


Fig. 3: Algorithm for computing the maximum time horizon  $t^*$ .

More specifically, this problem is solved by checking collisions for each time step  $\Delta t$  starting at  $t_0$ . If there exists no intersection, we further check for  $t_{up} := t_{up} + \Delta t$  until  $t_{up} = T_h$ . If  $t_{up} = T_h$  and no collision is found, each value within the time interval  $[t_0, T_h]$  is a possible candidate for the maximum time  $t^*$ . However, to further prune this time interval, a lower bound is computed.

### ② Compute the lower bound $t_{low}$

The lower bound  $t_{low}$  is determined by computing the latest time when the ego vehicle can initiate an emergency maneuver along the optimal trajectory to safely stop before  $t_{up}$ . This represents a lower bound, since it is safe, but not necessarily optimal. Therefore, every state  $x_{opt}(t), t_0 \leq t \leq t_{low}$  is safe, since there exists at least one safe emergency maneuver starting at  $t_{low}$ . Hence, those states can be discarded from the search interval of  $t^*$ .

To compute  $t_{low}$ , we first assume that  $v(t_{up}) = 0$ . Then, we compute the optimal velocity profile along the given path backwards in time. This velocity profile minimizes the trip time, starting with standstill at the position of  $x_{opt}(t_{up})$ , towards the position of  $x_{opt}(t_0)$ . To generate this optimal velocity profile which provides the shortest trip time for curved paths, the method presented in [31] can be used. However, if the path curvature is negligible, full deceleration  $a_{min}$  provides the fastest emergency maneuver. Finally, the time where the velocity profile of the optimal trajectory  $x_{opt}$  intersects with the optimal velocity for decelerating to the position of  $x_{opt}(t_{up})$ ,

where standstill is reached, represents  $t_{low}$ , as illustrated in Fig. 4.

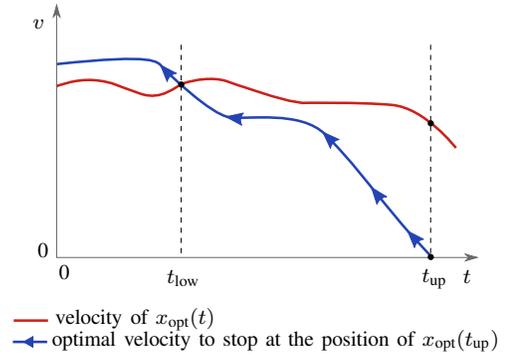


Fig. 4: Computation of  $t_{low}$ .

### ③ Binary search of $t^*$

To generate an emergency maneuver, an optimal control-based method is used. The considered objective function to be minimized includes four weighted terms: The first term minimizes the velocity  $v(t)$ , in order to reach standstill. Additionally, the steering rate  $u_1(t)$ , the acceleration  $u_2(t)$  and, the steering angle  $\delta(t)$  are minimized. Thus, to generate the emergency maneuver, the following constrained optimization problem is constructed:

$$\begin{aligned} \min_u \quad & \sum_{t_i} \gamma_1 v(t_i)^2 + \gamma_2 u_1(t_i)^2 + \gamma_3 u_2(t_i)^2 + \gamma_4 \delta(t_i)^2, \\ \text{subject to} \quad & \forall k \in \{1, \dots, p\} \quad \forall \tau_i \in \{\tau^*, \dots, \tau_{emg}\} : \\ & \Gamma(x(\tau_i)) \cap Occ_k(\tau_i) = \emptyset, \\ & \text{eq. (1) - (6)}, \end{aligned} \tag{9}$$

where  $\tau = [t, t + \Delta t]$  and  $\gamma_1, \gamma_2, \gamma_3, \gamma_4$  are weighting factors.

To solve (9), we apply Sequential Quadratic Programming (SQP) [32], since this method provides an efficient solution of constrained nonlinear optimization problems.

To find  $t^*$  within the interval  $[t_{low}, t_{up}]$ , any search algorithm can be used. Here, binary search [30] is applied due to its efficiency ( $\mathcal{O}(\log n)$  complexity) compared to the sequential search ( $\mathcal{O}(n)$  complexity).

Let us introduce the operator  $idx(e, v)$  which provides the index of the element  $e$  within a vector  $v$ ; additionally, the following notation is used:  $timeVect = [t_{low}, t_{low} + \Delta t, \dots, t_{up}]$ . Alg. 1 presents in detail how the computation of  $t^*$  is performed. First, an initialization is done by adding a label  $visit_i \leftarrow 0$ , which is a boolean variable, for each element  $timeVect_i$ . An element  $timeVect_i$  gets the label  $visit_i \leftarrow 1$  if it has been already checked whether an emergency maneuver starting with that element exists or not. Then, binary search is applied to find  $t^*$ ; instead of comparing the target value with the elements within the array, we check at which element an emergency maneuver starts.

## V. NUMERICAL EXPERIMENTS

To demonstrate the efficiency of our presented algorithm, our approach is tested against real traffic data. This data

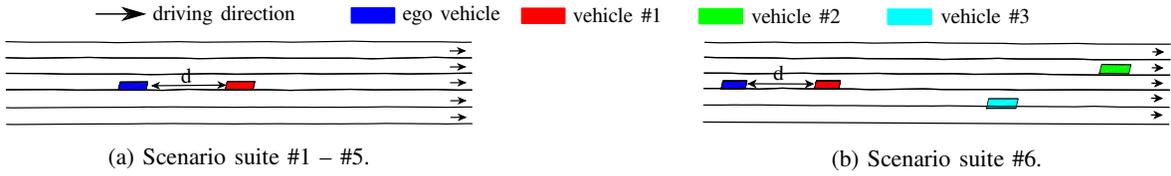


Fig. 5: Initial setting for the considered scenarios suite.

---

**Algorithm 1:** Search of  $t^*$  within  $[t_{low}, t_{up}]$ .

---

**Input:**  $t_{up}, t_{low}, \Delta t, timeVect$   
**Output:**  $t^*$

```

1 forall the  $t_i \in timeVect$  do
2    $visit_i \leftarrow 0$ 
3 while true do
4    $i \leftarrow idx(t_{low}, timeVect), j \leftarrow idx(t_{up}, timeVect);$ 
5    $t_{mid} \leftarrow timeVect(\lfloor \frac{i+j}{2} \rfloor);$ 
6   if  $\exists(\mathcal{X}_{emg}(t_{mid}))$  then
7      $s \leftarrow idx(t_{mid} + \Delta t, timeVect);$ 
8     if  $visit_s == 1$  then
9        $t^* \leftarrow t_{mid}$ 
10      break
11    else
12       $t_{low} \leftarrow t_{mid}$ 
13       $visit_{\lfloor \frac{i+j}{2} \rfloor} \leftarrow 1$ 
14    else
15       $s \leftarrow idx(t_{mid} - \Delta t, timeVect);$ 
16      if  $visit_s == 1$  then
17         $t^* \leftarrow t_{mid} - \Delta t$ 
18        break
19      else
20         $t_{up} \leftarrow t_{mid}$ 
21         $visit_{\lfloor \frac{i+j}{2} \rfloor} \leftarrow 1$ 
22 return  $t^*$ 

```

---

set was collected within the Next Generation Simulation (NGSIM)<sup>2</sup> program, on the Hollywood Freeway, in Los Angeles, CA, on June 15th, 2005. Acceleration, velocity, orientation, and position of each vehicle after each time step  $\Delta t = 0.1$  are included in the data set. Additionally, the length and the width of each traffic participant are provided.

#### A. Simulation setup

To validate our method, six scenario suites are analyzed: the first five consider only one surrounding vehicle, whereas the last one considers three surrounding vehicles. First, we select an initial frame from the data set and set it as the initial simulation time. Then, at each future time step  $t_i$ , we extract the corresponding information regarding surrounding vehicles.

The initial set-up for the considered traffic scenario suites is illustrated in Fig. 5. The constraints for the velocity limitation, steering rate, and acceleration, as well as the length, the width, and characteristic velocity used for the ego vehicle, are shown in Tab. I. Additionally, the weighting parameters used in the optimization problem are set with the following values:  $\gamma_1 = 80$ ,  $\gamma_2 = 10$ ,  $\gamma_3 = 1$ ,  $\gamma_4 = 0$ . The initial velocity  $v$  of the ego vehicle and the initial smallest distance  $d$  to any surrounding vehicle are defined by the user. In the simulation, different initial values of  $v$  and  $d$  are considered. In scenario suite #1 and #2,  $v$  is set as 20m/s and  $d$  is 18m and 12m, respectively, while in scenario suite #3 and #4,  $v$  is set as 16m/s and  $d$  is 18m and 12m. The initial velocity of the surrounding vehicle is 12.19m/s in scenario suites #1 – #4. In scenario suite #5,  $v$  is 20m/s and  $d$  is 40m.

For scenario suite #6, the initial velocity  $v$  of the ego vehicle is 20m/s. The initial velocity of surrounding vehicles #1, #2, and #3 is taken from the dataset, and has the following values: 12.19m/s, 14.62m/s, and 10.92m/s, respectively. The distance between the ego vehicle and the closest surrounding vehicle is 18m. Figs. 6 and 7 show the results of the simulation for each considered scenario suite. For illustration purposes, the position of the surrounding vehicle is shown only for every 0.5 seconds, and not for every  $\Delta t$ .

#### B. Simulation results

The values of  $t^*$  computed for each scenario suite are shown in Fig. 8, as the time when an emergency maneuver must be generated. Taking scenario suite #3 as an example, the ego vehicle can safely follow the long-term trajectory for the following time interval  $[0, 2]$  and an emergency maneuver is kept available at  $t_1^* = 2$ s. Meanwhile, new measurements are collected, and a new safe interval is computed as  $[2, 3.4]$ . A safe emergency maneuver is pre-computed after  $t_2^* = 1.4$ s. Based on new information regarding the behavior of the surrounding vehicles, a new safe time horizon is computed, together with an evasive maneuver starting at the end of the computed time interval.

Using our previous work [24], 95 emergency maneuvers would have been generated, one for each time step. However, with the method proposed in this paper, only 7 emergency maneuvers are needed for the same scenario suite. Therefore, our algorithm is indeed capable of significantly reducing the required number of emergency maneuvers. Similarly, in scenario suite #6, we only need to generate 8 emergency maneuvers, whereas when using our previous approach, an emergency maneuver is generated for each time step, requiring 56 maneuvers.

<sup>2</sup><https://ops.fhwa.dot.gov/trafficanalysistools/ngsim.htm>

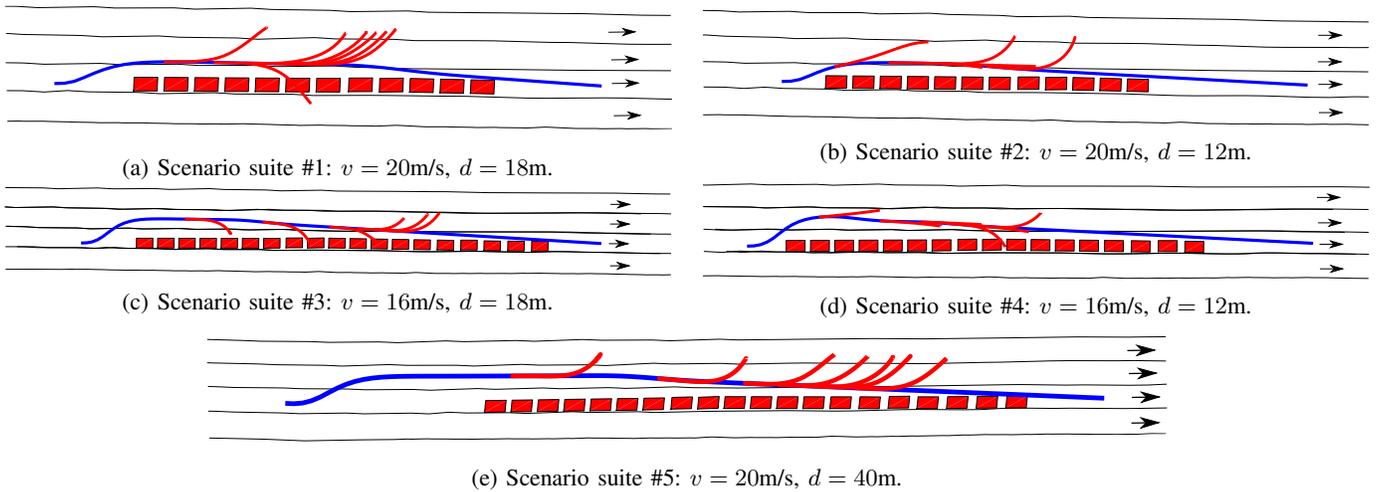


Fig. 6: Simulation results for scenario suite #1-#5.

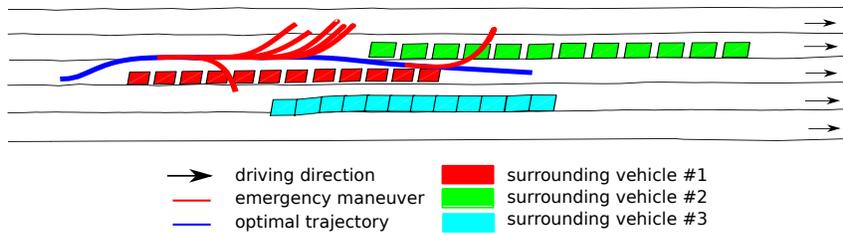


Fig. 7: Simulation result for scenario suite # 6.

TABLE I: Parameters.

Parameter (unit)	$\Delta t$ (s)	$v$ (m/s)	$\delta$ (rad)	$\dot{\delta}$ (rads)	$a$ (m/s <sup>2</sup> )	$l$ (m)	$w$ (m)	$v_{ch}$ (m/s)
Value/Interval	0.1	[0,30]	$[-\pi/2, \pi/2]$	$[-\pi/8, \pi/8]$	[-10,10]	4	2	20

When comparing scenario suites #1 to #2 and #3 to #4, we can see that our algorithm automatically adapts the frequency of generating emergency maneuvers, based on the current traffic situation. When the ego vehicle gets closer to a surrounding vehicle,  $t^*$  becomes smaller. More frequent emergency maneuvers are generated in order to account for the future possible behavior of other traffic participants. Scenario suite #5 shows this more obviously - when the ego vehicle is approaching the surrounding vehicle in longitudinal direction, more frequent emergency maneuvers are generated.

The simulation results show that our algorithm can indeed reduce the frequency of generating emergency maneuvers. Moreover, our proposed method can automatically adapt the frequency of generating emergency maneuvers to changing traffic conditions.

## VI. CONCLUSION

In this paper, an algorithm for computing the maximum time horizon during which a vehicle can safely follow a given trajectory is proposed. Instead of generating emergency maneuvers at each time step, we determine the maximum time horizon during which the ego vehicle can follow a trajectory

without needing to apply an emergency maneuver. To prune the search interval for finding this maximum time horizon, first a lower and an upper bound are computed. After the search interval is pruned, we apply binary search to find the maximum time horizon for which a given trajectory can be safely followed. At the end of this time horizon, only one emergency maneuver is generated.

The proposed method is tested against real traffic data and can highly reduce the frequency of generating emergency maneuvers, while still guaranteeing safety. The simulation results show that our approach performs well under different traffic scenarios, adapting the frequency of generating emergency maneuvers according to the current traffic situation.

## ACKNOWLEDGMENT

The authors would like to gratefully acknowledge financial support by the German Research Foundation (DFG) grant Graduiertenkolleg 1480 (PUMA).

## REFERENCES

- [1] J. Bruce and M. Veloso, "Real-time randomized path planning for robot navigation." in *Proc. of the IEEE Conference on Intelligent Robots and Systems*, vol. 3, 2002, pp. 2383 – 2388.

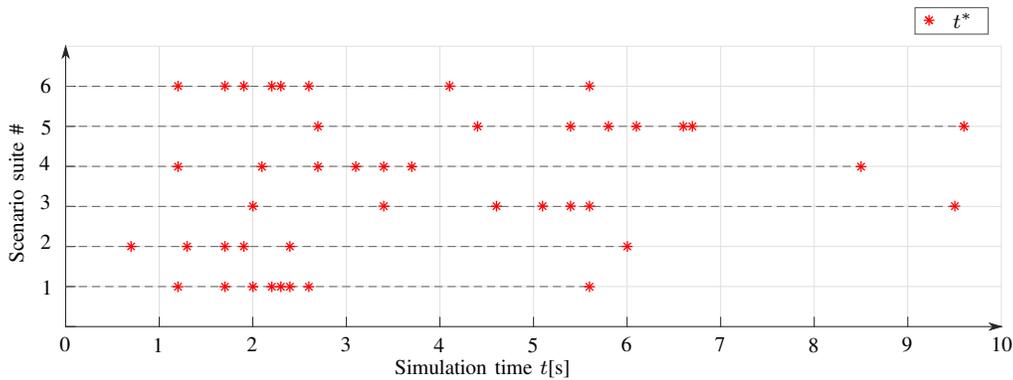


Fig. 8: Values of  $t^*$  for the considered scenarios suite. After each  $t^*$ , the current traffic scenario is updated and it is checked if the current optimal trajectory is still feasible. If it is not, and there exists no other new optimal trajectory, the precomputed emergency maneuver is engaged until a new optimal trajectory is found. Otherwise, a new  $t^*$  is computed, during which the ego vehicle can safely follow the optimal trajectory.

- [2] J. Kuffner and S. LaValle, "RRT-connect: An efficient approach to single-query path planning." in *Proc. of the IEEE Conference on Robotics and Automation*, vol. 2, 2000, pp. 995 – 1001.
- [3] Y. Kuwata, J. Teo, G. Fiore, and S. Karaman, "Real-time motion planning with applications to autonomous urban driving." *IEEE Trans. on Control Systems Technology*, vol. 17, no. 5, pp. 1105 – 1118, 2009.
- [4] Z. Sun, D. Hsu, T. Jiang, H. Kurniawati, and J. H. Reif, "Narrow passage sampling for probabilistic roadmap planning." *IEEE Trans. on Robotics*, vol. 21, no. 6, pp. 1105 – 1115, 2005.
- [5] P. Svestka and M. H. Overmars, "Coordinated motion planning for multiple car-like robots using probabilistic roadmaps." in *Proc. of the IEEE Conference on Robotics and Automation*, vol. 2, 1995, pp. 1631 – 1636.
- [6] L. E. Kavraki, P. Svestka, J. C. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces." *IEEE Trans. on Robotics and Automation*, vol. 12, no. 4, pp. 566 – 580, 1996.
- [7] K. Belghith, F. Kabanza, L. Hartman, and R. Nkambou, "Anytime dynamic path-planning with flexible probabilistic roadmaps." in *Proc. of the IEEE Conference on Robotics and Automation*, 2006, pp. 2372 – 2377.
- [8] L. E. Kavraki, M. N. Kolountzakis, and J. C. Latombe, "Analysis of probabilistic roadmaps for path planning." *IEEE Trans. on Robotics and Automation*, vol. 14, no. 1, pp. 166 – 171, 1998.
- [9] E. Frazzoli, "Robust hybrid control for autonomous vehicle motion planning." Ph.D. dissertation, Massachusetts Institute of Technology, 2001.
- [10] E. Frazzoli, M. A. Dahleh, and E. Feron, "Maneuver-based motion planning for nonlinear systems with symmetries." *IEEE Trans. on Robotics*, vol. 21, no. 6, pp. 1077–1091, 2005.
- [11] D. Heß, M. Althoff, and T. Sattel, "Formal verification of maneuver automata for parameterized motion primitives." in *Proc. of the IEEE International Conference on Intelligent Robots and Systems*, 2014, pp. 1474 – 1481.
- [12] C. B. Neas and M. Farhood, "A hybrid architecture for maneuver-based motion planning and control of agile vehicles." in *Proc. of the IFAC World Congress*, vol. 18, 2011, pp. 3521–3526.
- [13] A. Gray, Y. Gao, T. Lin, J. K. Hedrick, H. E. Tseng, and F. Borrelli, "Predictive control for agile semi-autonomous ground vehicles using motion primitives." in *Proc. of the IEEE American Control Conference*, 2012, pp. 4239 – 4244.
- [14] S. Quinlan and O. Khatib, "Elastic bands: Connecting path planning and control." in *Proc. of the IEEE Conference on Robotics and Automation*, vol. 2, 1993, pp. 802 – 807.
- [15] J. Hilgert, K. Hirsch, T. Bertram, and M. Hiller, "Emergency path planning for autonomous vehicles using elastic band theory." in *Proc. of the IEEE Conference on Advanced Intelligent Mechatronics*, 2003, pp. 1390 – 1395.
- [16] T. Gu, J. Atwood, C. Dong, J. M. Dolan, and J.-W. Lee, "Tunable and stable real-time trajectory planning for urban autonomous driving." in *Proc. of the IEEE Conference on Intelligent Robots and Systems*, 2015, pp. 250 – 256.
- [17] S. K. Gehrig and F. J. Stein, "Elastic bands to enhance vehicle following." in *Proc. of the IEEE Conference on Intelligent Transportation Systems*, 2001, pp. 597 – 602.
- [18] T. Sattel and T. Brandt, "Ground vehicle guidance along collision-free trajectories using elastic bands." in *Proc. of the IEEE American Control Conference*, 2005, pp. 4991 – 4996.
- [19] S. J. Anderson, S. C. Peters, T. E. Pilutti, and K. Iagnemma, "An optimal-control-based framework for trajectory planning, threat assessment, and semi-autonomous control of passenger vehicles in hazard avoidance scenarios." *IEEE Trans. on International Journal of Vehicle Autonomous Systems*, vol. 8, no. 2/3/4, pp. 190 – 216, 2010.
- [20] J.-M. Park, D.-W. Kim, Y.-S. Yoon, H. Kim, and K.-S. Yi, "Obstacle avoidance of autonomous vehicles based on model predictive control." *Proc. of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, vol. 223, no. 12, pp. 1499–1516, 2009.
- [21] M. Werling and D. Liccardo, "Automatic collision avoidance using model-predictive online optimization." in *Proc. of the IEEE Conference on Decision and Control*, 2012, pp. 6309 – 6314.
- [22] K. Hauser, "Adaptive time stepping in real-time motion planning," in *Algorithmic Foundations of Robotics IX*, 2010, pp. 139–155.
- [23] E. Frew, J. Langelaan, and M. Stachura, "Adaptive planning horizon based on information velocity for vision-based navigation," in *Guidance, Navigation and Control Conference*, 2007, pp. 3822–3841.
- [24] S. Magdici and M. Althoff, "Fail-safe motion planning of autonomous vehicles," in *Proc. of the IEEE International Conference on Intelligent Transportation Systems*, 2016, pp. 452–458.
- [25] M. Althoff and S. Magdici, "Set-based prediction of traffic participants on arbitrary road networks," *IEEE Trans. on Intelligent Vehicles*, vol. 1, no. 2, pp. 187–202, 2016.
- [26] M. Koschi and M. Althoff, "SPOT: A tool for set-based prediction of traffic participants," in *Proc. of the IEEE Intelligent Vehicles Symposium*, 2017, pp. 1679–1686.
- [27] A. Barth and U. Franke, "Where will the oncoming vehicle be the next second?" in *Proc. of the IEEE Intelligent Vehicles Symposium*, 2008, pp. 1068 – 1073.
- [28] A. Houenou, P. Bonnifait, V. Cherfaoui, and W. Yao, "Vehicle trajectory prediction based on motion model and maneuver recognition." in *Proc. of the IEEE Conference on Intelligent Robots and Systems*, 2013, pp. 4363 – 4369.
- [29] Y. Kuwata, G. A. Fiore, J. Teo, E. Frazzoli, and J. P. How, "Motion planning for urban driving using rrt," in *Proc. of the IEEE International Conference on Intelligent Robots and Systems*, 2008, pp. 1681–1686.
- [30] D. E. Knuth, *The Art Of Computer Programming*. Addison-Wesley Professional, 2005, vol. 3.
- [31] E. Velenis and P. Tsiotras, "Optimal velocity generation for given acceleration limits: Theoretical analysis," in *Proc. of the IEEE American Control Conference*, 2005, pp. 1478–1483.
- [32] J. Nocedal and S. J. Wright, *Numerical Optimization*. Springer, 2006.