



**HAL**  
open science

# COMPUTATION OF AVOIDANCE REGIONS FOR DRIVER ASSISTANCE SYSTEMS BY USING A HAMILTON-JACOBI APPROACH

Ilaria Xausa, Robert Baier, Olivier Bokanowski, Matthias Gerdt

► **To cite this version:**

Ilaria Xausa, Robert Baier, Olivier Bokanowski, Matthias Gerdt. COMPUTATION OF AVOIDANCE REGIONS FOR DRIVER ASSISTANCE SYSTEMS BY USING A HAMILTON-JACOBI APPROACH. Optimal Control Applications and Methods, In press, 10.1002/oca.2565 . hal-01123490v3

**HAL Id: hal-01123490**

**<https://hal.science/hal-01123490v3>**

Submitted on 27 Nov 2019

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# COMPUTATION OF AVOIDANCE REGIONS FOR DRIVER ASSISTANCE SYSTEMS BY USING A HAMILTON-JACOBI APPROACH

ILARIA XAUSA, ROBERT BAIER, OLIVIER BOKANOWSKI,  
AND MATTHIAS GERDTS

**ABSTRACT.** We consider the problem of computing safety regions, modeled as nonconvex backward reachable sets, for a nonlinear car collision avoidance model with time-dependent obstacles. The Hamilton-Jacobi-Bellman framework is used. A new formulation of level set functions for obstacle avoidance is given and sufficient conditions for granting the obstacle avoidance on the whole time interval are obtained, even though the conditions are checked only at discrete times. Different scenarios including various road configurations, different geometry of vehicle and obstacles, as well as fixed or moving obstacles, are then studied and computed. Computations involve solving nonlinear partial differential equations of up to five space dimensions plus time with nonsmooth obstacle representations, and an efficient solver is used to this end. A comparison with a direct optimal control approach is also done for one of the examples.

**Keywords:** collision avoidance, Hamilton-Jacobi-Bellman equations, backward reachable sets, level set approach, high dimensional partial differential equations.

## 1. MOTIVATION

The paper investigates optimal control approaches for the detection of potential collisions in car motions. The aim is to identify and to compute safety regions for the car by means of a reachable set analysis. Related techniques for collision detection in real time have been developed in [1] (see also reference therein) using reachability analysis with zonotopes and linearized dynamics. Reachable set approximations have also been obtained through zonotopes in [25], [3], [2] (CORA matlab toolbox), or through polytopes in [25], [24]. For an overview of methods see also [11, 7]. The new contributions in this paper are the ability to approximate nonconvex reachable sets for general nonlinear dynamics accurately while taking into account complicated scenarios with obstacles. The first aim of the paper is to present a verification tool for safety systems, illustrated on a car avoidance model, compare [27, 32]. The proposed method has the advantage to be capable of approximating

---

*Date:* November 27, 2019.

This work was partially supported by the EU under the 7th Framework Programme Marie Curie Initial Training Network “FP7-PEOPLE-2010-ITN”, SADCO project, GA number 264735-SADCO.

the reachable set without relying on convex overestimation or underestimation. It also avoids a linearization of the nonlinear dynamics.

It is not primarily intended to use this techniques in real time, however, a potential approach towards real time computations would be to create a database of solutions for different scenarios and to use some online interpolation techniques.

It is well known that Hamilton-Jacobi (HJ) approach can be used to modelize and to compute reachable sets [26]. A general setting for taking state constraints (or obstacles) into account for the HJ setting is given in [11], the approach we use in the present work. In [14] the collision avoidance of an unmanned aerial vehicle (UAV) is modeled as an infinite horizon control problem with obstacle avoidance and also solved by using an HJ approach.

For a general presentation of HJ equations we refer to the textbook [8]. A large panel of approximation methods are furthermore available. Finite difference methods for the approximation of nonlinear HJ equations were first proposed in [13]. Precise numerical schemes (finite-difference type schemes) were later on developed in [28]. We refer to [30] and [16] for recent surveys on related high-order discretization methods.

In this work, we also apply the HJ framework in order to deal with time dependent state constraints as detailed in [12].

The motion of the car is modeled by the "point mass" model, which is a nonlinear 4-state model (see section 2.2) with two controls for acceleration/deceleration and for the yaw rate. More precise car model exists (as the one in [19]). However, the 4-state model is often used for reference in computations and is more easy to handle numerically by the Hamilton-Jacobi approach.

Once an obstacle has been detected by suitable sensors (e.g. radar, lidar), our approach can be used to decide whether a collision is going to happen or not.

In order to compute the reachable sets, the ROC-HJ solver [10] will be used, considering several different car models and scenarios. The software solves Hamilton-Jacobi Bellman (HJB) equations and can be used for computing the reachable sets as well as optimal trajectories. Moreover, for one scenario, we shall also verify our simulations by the DFOG method of [7] (using the OCPID-DAE1 solver [20]). For solving the HJB equations, a numerical method using a mesh grid is introduced, so numerical errors may occur. However, as the mesh grid is refined, the numerical error will converges to zero. Such errors are analyzed in the first example of the numerical section (see Table 1).

The plan of the paper is the following. In Section 2, we consider the 4-dimensional point mass model for a car and we describe the problem of the backward reachable set computation under different types of nonsmooth state constraints, and also introduce level set functions to represent the reachable sets. In Section 3, the HJB approach is briefly recalled in our setting. In Section 4, a general way to construct explicit level set functions

associated to state constraints is introduced, and a new procedure with respect to collision avoidance is described in more details, with particular focus also for the avoidance of rectangular vehicles in the HJB framework. Section 5 contains several numerical examples for collision avoidance scenarios, showing the relevance of our approach. Finally a conclusion is made in Section 6, where we also outline some ongoing works using the HJB approach for collision avoidance.

## 2. PROBLEM SETTING AND MODELLING

**2.1. Presentation of the problem.** The main tasks in collision avoidance are to reliably indicate future collisions and – if possible – to provide escape trajectories if such exist. In particular once an obstacle has been detected by suitable sensors (e.g. radar, lidar), we want to be able to decide whether a collision is going to happen or not.

Let  $U$  be a nonempty compact subset of  $\mathbb{R}^m$  with  $m \geq 1$ . Let  $n \geq 1$  and  $f : \mathbb{R}^n \times U \rightarrow \mathbb{R}^n$  be Lipschitz continuous with respect to  $(z, u)$ . Let

$$\mathcal{U} := \{u : [0, \infty) \rightarrow U \mid u \text{ measurable}\}$$

be the set of control policies. Given an initial state  $z_0 \in \mathbb{R}^n$ , we denote by  $z_{z_0}^u$  the *absolutely continuous* solution of the following nonlinear dynamical system

$$\dot{z}(s) = f(z(s), u(s)) \quad \text{for } s \geq 0, \quad (1)$$

$$z(0) = z_0. \quad (2)$$

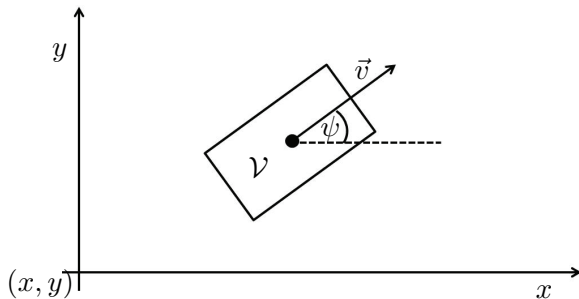
(More precisely, (1) stands only for "almost every"  $s \geq 0$  since the control  $u(\cdot)$  is only a measurable function of the time.) Let  $\Omega$  and  $(\mathcal{K}_s)_{s \geq 0}$  be nonempty closed sets of  $\mathbb{R}^n$ .

**Definition 2.1.** *The backward reachable set associated to the dynamics  $f$ , for reaching the target  $\Omega$  within time  $t$  and respecting the state constraints  $(\mathcal{K}_s)_{s \geq 0}$  is defined as:*

$$\mathcal{R}_{\Omega, (\mathcal{K}_s)_s}^f(t) := \left\{ z_0 \in \mathbb{R}^n \mid \exists u \in L^\infty((0, \infty), U) : z_{z_0}^u(t) \in \Omega \right. \\ \left. \text{and } z_{z_0}^u(s) \in \mathcal{K}_s \text{ for } s \in [0, t] \right\}.$$

*If the context is clear, we use the abbreviation  $\mathcal{R}(t)$ .*

The definition corresponds to the set of points  $z_0$  such that there exists trajectory starting from  $z_0$  ending inside the target area  $\Omega$  at the given time  $t \geq 0$ , while satisfying the state constraints. In the application in the next subsection,  $(\mathcal{K}_s)_s$  will model that the car staying on the road and avoiding the obstacles.

FIGURE 1. vehicle  $\mathcal{V}$ 

**2.2. The 4-dimensional point mass model.** The *point mass model* used hereafter is a simplified four-dimensional nonlinear state model for the car, where the controls are the acceleration/deceleration and yaw rate.

The center of gravity of the car is identified with its coordinates  $(x(t), y(t))$ ,  $v(t)$  denotes the module of the velocity of the car and  $\psi(t)$  is the yaw angle, see Fig. 1. The equations of motion are then given by

$$x' = v \cos(\psi) \quad (3a)$$

$$y' = v \sin(\psi) \quad (3b)$$

$$\psi' = w \quad (3c)$$

$$v' = a \quad (3d)$$

where  $a(t) \in [a_{min}, a_{max}]$  is the control for acceleration (if  $a(t) > 0$ ) or deceleration (if  $a(t) < 0$ ), and  $w(t)$  is the yaw rate, such that  $|w(t)| \leq w_{max}$ .

**2.3. Level set functions for target and state constraints.** The *target set*  $\Omega$  will represent the target area. In our examples we will consider a region in the form of

$$\Omega = \left\{ z = (x, y, \psi, v) \in \mathbb{R}^4 \mid x \geq x_{target}, y \geq y_{target}, |\psi| \leq \varepsilon \right\} \quad (4)$$

for a small threshold  $\varepsilon > 0$ . The conditions on  $x$  and  $y$  realizes that the vehicle will reach a secure end position after the maneuver, while the conditions on the yaw angle assures that the car is driving approximately horizontally so that the car is not heading to leave the road after the maneuver.

An important tool, which will be used in this paper, is the notion of *level set functions* associated to a given set. For the target constraint  $\Omega$ , we will associate a Lipschitz continuous function  $\varphi : \mathbb{R}^n \rightarrow \mathbb{R}$  (here with  $n = 4$ ), such that

$$z \in \Omega \iff \varphi(z) \leq 0.$$

Indeed there always exists such a function, since one can use the *signed distance* to  $\Omega$  ( $\varphi(z) = d(z, \Omega)$  if  $z \notin \Omega$ , and  $\varphi(z) = -d(z, \mathbb{R}^n \setminus \Omega)$  if  $z \in \Omega$ ).

For instance, in the case of (4), we can consider:

$$\varphi(z) := \max \left( -(x - x_{target}), -(y - y_{target}), |\psi| - \varepsilon \right).$$

More complex level set functions will be constructed in Section 4 but they may no more correspond to usual distance functions.

Now for the case of the state constraints, different road geometries will be considered, as well as different fixed and/or moving obstacles that the car must avoid. In all cases, we will manage to model all state constraints by a simple, explicit, level set function  $g$ .

In the particular case of fixed configurations, state constraints are of the form  $z(s) \in \mathcal{K}$  for all  $s \in [0, t]$  (for a given closed set  $\mathcal{K}$  of  $\mathbb{R}^n$ ), and we will construct a Lipschitz continuous function  $g : \mathbb{R}^n \rightarrow \mathbb{R}$  such that

$$z \in \mathcal{K} \iff g(z) \leq 0.$$

In the general case of moving obstacles, the state constraints are of the form

$$z(s) \in \mathcal{K}_s, \quad \forall s \in [0, t],$$

where  $(\mathcal{K}_s)_{s \geq 0}$  is a family of closed sets. Then we need to introduce a time-dependent level set function (still denoted  $g$ )  $g : \mathbb{R}^n \times \mathbb{R} \rightarrow \mathbb{R}$ , such that, for any  $s \geq 0$ ,

$$z \in \mathcal{K}_s \iff g(z, s) \leq 0.$$

The obstacles and corresponding level set functions will be more precisely described and constructed in Section 4.

### 3. HAMILTON-JACOBI-BELLMAN APPROACH

In order to apply the HJB approach of [12], assumptions (H1)-(H3) will be assumed on the objects modelling the car traffic scenario as defined in the previous section. Assumption (H1) is natural for controlled systems and fulfilled for our model. Assumption (H2) and (H3) allow for nonsmooth representation of targets and state constraints.

(H1)  $f : \mathbb{R}^n \times U \rightarrow \mathbb{R}^n$  is a continuous function and Lipschitz continuous in  $z$  uniformly in  $u$ , i.e.,  $\exists L \geq 0, \forall z_1, z_2, \forall u \in U: |f(z_1, u) - f(z_2, u)| \leq L|z_1 - z_2|$ .

(H2)  $\Omega$  is a nonempty closed set of  $\mathbb{R}^n$ . Let  $\varphi : \mathbb{R}^n \rightarrow \mathbb{R}$  be Lipschitz continuous and a level set function for the target, i.e.,

$$\varphi(z) \leq 0 \iff z \in \Omega.$$

(H3)  $(\mathcal{K}_s)_{s \in [0, T]}$  are a family of subsets of  $\mathbb{R}^n$  such that there exists a Lipschitz continuous level set function  $g : \mathbb{R}^n \times \mathbb{R} \rightarrow \mathbb{R}$  with

$$g(z, s) \leq 0 \iff z \in \mathcal{K}_s \quad \text{for all } s \in [0, T], z \in \mathbb{R}^n.$$

The existence of a function  $g$  with the Lipschitz regularity is discussed in [12].

Furthermore, we will assume the following convexity assumption on the dynamics:

(H4) For all  $z \in \mathbb{R}^n$  the velocity set  $f(z, U)$  is convex.

In our case, for the dynamics given in (3) (resp. (9)) this convexity assumption is satisfied since the map  $u \mapsto f(x, u)$  is affine. It allows to have the compactness of the set of trajectories and therefore the existence of a minimum for the value function (6) (or (13)).

However this condition is not mandatory and we could use the Filippov-Ważewski relaxation theorem (see [6, Sec. 2.4, Theorem 2]) to work with a convexified dynamics in the case  $f(x, U)$  is nonconvex.

We now focus on backward reachable sets  $\mathcal{R}_{\Omega, (\mathcal{K}_s)}^f(t)$  associated to a dynamics  $f$  and how to compute such reachable sets.

**Remark 3.1.** *Assume that there is no time dependency in the dynamics, and that  $\mathcal{K}_s = \mathcal{K}$  for all  $s \geq 0$ . Let the capture basin, or backward reachable set until time  $t$ , be defined by*

$$\text{Cap}_{\Omega, (\mathcal{K}_s)}^f(t) := \left\{ z_0 \in \mathbb{R}^n \mid \exists \tau \in [0, t], \exists u \in \mathcal{U}, z_{z_0}^u(\tau) \in \Omega \right. \\ \left. \text{and } z_{z_0}^u(s) \in \mathcal{K}_s \text{ for all } s \in [0, \tau] \right\} \quad (5)$$

(following [5, Subsec. 1.2.1.2]) so that we consider trajectories that can reach the target until time  $t$ . We will also use the abbreviation  $\text{Cap}(t)$ . Then the set (5) is also a reachable set for a modified dynamics  $\hat{f}$ :

$$\text{Cap}_{\Omega, \mathcal{K}}^f(t) = \mathcal{R}_{\Omega, (\mathcal{K}_s)}^{\hat{f}}(t)$$

where  $\tilde{f}(z_0, \hat{u}) := \lambda f(z_0, u)$  for  $\hat{u} = (u, \lambda) \in \hat{U} = U \times [0, 1]$  (see for instance [26]). Here, a new virtual control  $\lambda(\cdot)$  with  $\lambda(s) \in [0, 1]$  is introduced.

We first consider the case of time-invariant state constraints (such as the road and non-moving obstacles), i.e.,

$$\mathcal{K}_s := \mathcal{K} \quad \forall s \geq 0.$$

Let us associate to this problem the following *value function*  $\vartheta$ :

$$\vartheta(z_0, t) := \inf_{u \in \mathcal{U}} \max \left( \varphi(z_{z_0}^u(t)), \max_{s \in [0, t]} g(z_{z_0}^u(s)) \right). \quad (6)$$

Here, we simply denote  $g(z, s) \equiv g(z)$ . Such a value function involving a supremum cost has been studied by Barron and Ishii in [9]. As a consequence of [11], we have:

**Proposition 3.2.** *The value function  $\vartheta$  is a level set function for  $\mathcal{R}_{\Omega, (\mathcal{K}_s)_s}^f(t)$  in the sense that the following holds:*

$$\mathcal{R}_{\Omega, (\mathcal{K}_s)_s}^f(t) = \{z_0 \in \mathbb{R}^n \mid \vartheta(z_0, t) \leq 0\} \quad (7)$$

In particular assumptions (H2) and (H3) are essential for (7) to hold. Furthermore,  $v$  is the unique continuous *viscosity solution* (in the sense of [8, Sec. I.3]) of the following Hamilton-Jacobi (HJ) equation

$$\min(\partial_t \vartheta + \mathcal{H}(z, \nabla_z \vartheta), \vartheta - g(z)) = 0, \quad t > 0, z \in \mathbb{R}^n, \quad (8a)$$

$$\vartheta(z, 0) = \max(\varphi(z), g(z)), \quad z \in \mathbb{R}^n, \quad (8b)$$

where

$$\mathcal{H}(z, p) := \max_{u \in U} (-f(z, u) \cdot p), \quad p \in \mathbb{R}^m,$$

is the *Hamiltonian*.

**Remark 3.3.** *From the point of view of the vehicle obstacle avoidance problem, the value function  $\vartheta$  has the following property :  $\vartheta((x_0, y_0, \psi_0, v_0), t) \leq 0$  if and only if it is possible to drive the vehicle from the initial position  $z_0 = (x_0, y_0, \psi_0, v_0)$  with dynamics (1), towards the target set  $\Omega$ , avoiding the obstacles (such as possible other obstacle vehicles) before time  $t$ .*

We now consider the computation of backward reachable sets for *time dependent state constraints*, and follow the approach of [12]. We consider (3) with the new state variable  $\xi := (z, \tau)$  and the "augmented" dynamics with values in  $\mathbb{R}^5$ :

$$F(\xi, u) := \begin{pmatrix} f(z, u) \\ 1 \end{pmatrix}. \quad (9)$$

This means, in solving the corresponding PDE, we have to consider an additional state variable: the dynamics is augmented by the equation

$$\tau' = 1. \quad (10)$$

Let also be  $\xi_0 := (z_0, 0)$  and trajectories  $\xi_{\tau_0, \xi_0}^u$  associated to  $F$ , fulfilling

$$\dot{\xi}(s) = F(\xi(s), u(s)) \text{ and } \xi(\tau_0) = \xi_0.$$

For a fixed  $T > 0$ , let

$$\tilde{\Omega} := \bigcup_{s \in [0, T]} \Omega \times \{s\} \equiv \Omega \times [0, T] \quad (11)$$

and

$$\tilde{\mathcal{K}} := \bigcup_{s \in [0, T]} \mathcal{K}_s \times \{s\}. \quad (12)$$

Then it holds:

**Proposition 3.4.** *For all  $t \geq 0$ , we have:*

$$\forall s \in [0, t], z_0 \in \mathcal{R}_{\Omega, (\mathcal{K}_s)_s}^f(t) \iff (z_0, 0) \in \mathcal{R}_{\tilde{\Omega}, \tilde{\mathcal{K}}}^F(t).$$



We extend the definition of  $\varphi$  by  $\varphi(z, s) := \varphi(z)$  so that for any  $\xi_0 \in \mathbb{R}^n \times \mathbb{R}$  and  $\tau \geq 0$  we can associate a value  $w$  as follows:

$$w(\xi_0, \tau) := \inf_{u \in \mathcal{U}} \max \left( \varphi(\xi_{\xi_0}^u(\tau)), \max_{s \in [0, \tau]} g(\xi_{\xi_0}^u(s)) \right) \quad (13)$$

Then, one can verify that  $w$  is Lipschitz continuous and the following theorem holds:

**Theorem 3.5.** *Assume (H1)-(H4) and consider  $w$  from (13).*

(i) *For every  $\tau \geq 0$  we have:*

$$\mathcal{R}_{\Omega, (\mathcal{K}_s)_s}^f(\tau) = \{z \in \mathbb{R}^n, w((z, 0), \tau) \leq 0\}.$$

(ii)  *$w$  is the unique continuous viscosity solution of*

$$\min(\partial_\tau w + \mathcal{H}((z, t), (\nabla_z w, \partial_t w)), w((z, t), \tau) - g(z)) = 0, \quad \tau > 0, (z, t) \in \mathbb{R}^{n+1}, \quad (14a)$$

$$w((z, t), 0) = \max(\varphi(z), g(z, t)), \quad (z, t) \in \mathbb{R}^{n+1}. \quad (14b)$$

where for any  $\xi = (z, t)$  and  $(p_z, p_t) \in \mathbb{R}^n \times \mathbb{R}$ :

$$\mathcal{H}((z, t), (p_z, p_t)) := \max_{u \in U} (-f(z, u) \cdot p_z - p_t).$$

Once the backward reachable set is characterized by a viscosity solution of (14), it is possible to use a pde solver to find the solution on a grid. We refer to [11] for the specific approximation of (8) (resp. (14)).

**Minimal time function and optimal trajectory reconstruction.** In the case of fixed state constraints (i.e.,  $\mathcal{K}_s \equiv \mathcal{K}$ ), the *minimal time function*, denoted by  $\mathcal{T}$ , is defined by:

$$\mathcal{T}(z_0) := \inf\{t \in [0, T] \mid \exists u \in \mathcal{U} : z_{z_0}^u(t) \in \Omega \text{ and } z_{z_0}^u(s) \in \mathcal{K} \text{ for all } s \in [0, t]\},$$

and if no such time  $t$  exists then we set  $\mathcal{T}(z_0) = \infty$ . Let  $\vartheta$  be defined as in (6). It is easy to see that the function satisfies

$$\mathcal{T}(z_0) = \inf\{t \in [0, T], \vartheta(z_0, t) \leq 0\}. \quad (15)$$

In particular, if  $\mathcal{T}(z_0) = +\infty$ , then there is no  $t \in [0, T]$  such that  $\vartheta(z_0, t) \leq 0$ . Notice that  $\mathcal{T}$  can be discontinuous even though  $\vartheta$  is always Lipschitz continuous. No controllability assumptions are used in the present approach.

The optimal trajectory reconstruction is then obtained by minimizing the minimal time function along possible trajectories, see for instance Falcone [15], Soravia [31]. From the numerical point of view this allows to store only the minimal time function and not the value function which would have one more variable (the time variable). (Note that precise trajectory reconstruction results from the value function can be found in [29, 4].) The minimal time function can be shown to satisfy a dynamic programming principle of the form:

$$\inf_{u(\cdot)} \mathcal{T}(z_\xi^u(h)) = \mathcal{T}(\xi) - h, \quad \text{for all } 0 < h \leq \mathcal{T}(\xi) < \infty, \quad (16)$$

where the infimum is over all control functions  $u(\cdot) \in \mathcal{U}$  such that the trajectory  $z_\xi^u(\cdot)$  satisfy the state constraints.

More precisely, let  $h > 0$  be a given time step, let  $N \geq 1$  be some integer (a maximal number of iterations). Assume that the starting point  $z_0$  is satisfying  $\mathcal{T}(z_0) < \infty$ , and that we aim to reach  $z_n := z^u(t_n) \in \Omega$  at some future time  $t_n = nh > 0$ , with  $n \leq N$ . This is equivalent to require  $\mathcal{T}(z_n) = 0$ . For a given small threshold  $\eta > 0$  and for a given control discretization of the set  $U$ , say  $(u_k)_{k=1, \dots, N_u} \subset U$ , we consider the following iterative procedure:

while  $n < N$  and  $\mathcal{T}(z_n) \geq \eta$  do:

Find  $u_{k^*} := \operatorname{argmin}_{u_k} \mathcal{T}(\bar{z}_{z_n}^{u_k}(h))$

Set  $z_{n+1} := \bar{z}_{z_n}^{u_{k^*}}(h)$

if  $\mathcal{T}(z_{n+1}) = \infty$  stop, otherwise set  $n \leftarrow n + 1$ .

More precisely, the procedure is stopped if  $\mathcal{T}(z_{n+1}) = +\infty$  (we cannot reach the target from this point), if  $\mathcal{T}(z_{n+1}) < \eta$  (target reached up to the threshold  $\eta$ ), or if the maximal number of iterations is reached. In the algorithm,  $\bar{z}_{z_n}^{u_k}(h)$  denotes a *one-step second-order Runge-Kutta approximation* of the trajectory with fixed control  $u_k$  on  $[t_n, t_{n+1}[$ . For instance, the *Heun scheme with piecewise constant selections* uses the iteration:

$$\bar{z}_{z_n}^{u_k}(h) := z_n + \frac{h}{2}(f(z_n, u_k) + f(z_n + hf(z_n, u_k), u_k)).$$

It is also possible to do smaller time steps between  $[t_n, t_n + h]$  in order to improve the precision for a given control  $u_k$ . Nevertheless, numerical observations show that the approximation is in general more sensitive to the control discretization  $(u_k)_{k=1, \dots, N_u}$  of the set  $U$ .

In the time-dependent case this minimal time function can be defined in a similar way from the value  $w$  (we refer to [12] for details), and the optimal trajectory reconstruction follows the same lines with the "augmented" dynamics (9).

**Remark 3.6.** *Notice that the discretization in the optimal trajectory reconstruction can be done completely independently from the discretization method used for solving the HJ equations as well as the minimal time function.*

When reconstructing the trajectory with the minimal time function in the iteration above, a simple yet important property is:

**Lemma 3.7.** *Assume that  $\mathcal{T}(z_n) < +\infty$ . Then  $g(z_n) \leq 0$ . This means in particular that the state  $z_n$  satisfies the state constraints.*

*Proof.* If  $\tau_n := \mathcal{T}(z_n) < \infty$  then  $\vartheta(z_n, \tau_n) \leq 0$ . On the other hand by definition of  $\vartheta$ :  $g(z_n) \leq \vartheta(z_n, \tau_n)$ . This concludes to the desired result.  $\square$

In particular, if the trajectory reconstruction satisfies  $\mathcal{T}(z_n) < \infty$ , for  $n = 0, \dots, N$  (and this is expected since  $\mathcal{T}(z_0) < +\infty$  and in view of the

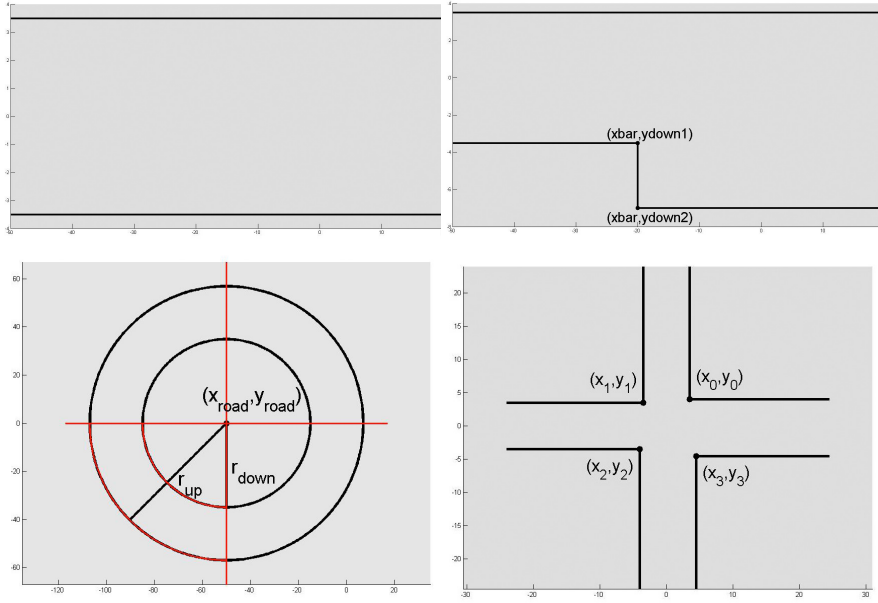


FIGURE 2. different road geometries: straight (top-left), varying width (top-right), curved (bottom-left), crossing (bottom-right)

dynamic programming principle (16)), then we will have

$$g(z_n) \leq 0, \quad \forall n = 0, \dots, N, \quad (17)$$

and therefore all points  $(z_n)_{n=0, \dots, N}$  satisfy the state constraints. In the time-dependent case, the condition  $g(z_n) \leq 0$  must be replaced by  $g(z_n, \tau_n) \leq 0$ .

#### 4. LEVEL SET FUNCTIONS FOR DIFFERENT PROBLEM DATA AND COLLISION AVOIDANCE

In this section we construct Lipschitz level functions to represent obstacle, state constraints and targets satisfying assumptions (H2)-(H3). The aim is also to illustrate how to obtain explicit analytic formula for some specific obstacles and vehicles. In some cases, the avoidance of two convex set may not be easily characterized by the negativity of some analytic function. We will propose a new way to use simplified analytic level set functions in order to obtain a collision avoidance characterization.

**4.1. Road configurations.** Let us first describe different simple road geometries depicted in Fig. 2 which will be used in the next numerical section. The road will be denoted by  $\mathcal{K}_r$ , a subset of  $\mathbb{R}^2$ .

- A straight road with constant width:

$$\mathcal{K}_r = \left\{ (x, y) \in \mathbb{R}^2 \mid y_{down} \leq y \leq y_{up} \right\}, \quad (18)$$

with  $y_{down} = -3.5$  and  $y_{up} = 3.5$  (values are typically in meters). A level set function associated to  $\mathcal{K}_r$  can be given by

$$g(x, y) := \max \left( -(y - y_{down}), -(y_{up} - y) \right). \quad (19)$$

- A straight road with varying widths, modeled as:

$$\mathcal{K}_r := \left\{ (x, y) \in \mathbb{R}^2 \mid y_{down}(x) \leq y \leq y_{up} \right\}, \quad (20)$$

where

$$y_{down}(x) = \begin{cases} y_{down_1} & \text{if } x \leq \bar{x}, \\ y_{down_2} & \text{if } x > \bar{x} \end{cases} \quad (21)$$

and  $y_{up}, y_{down_1}, y_{down_2}, \bar{x}$  are constants. It would be natural to define the level set function of (20) as in (19) where instead of  $y_{down}$  we use the function  $y_{down}(x)$  defined in (21). However, (21) leads to a discontinuous function  $g$ , which is not convenient for our purposes. In this particular case, the following definition is better suited because it is Lipschitz continuous in  $(x, y)$ :

$$g(x, y) = \max \left( \min \left( -(y - y_{down_1}), -(x - \bar{x}) \right), \right. \\ \left. -(y - y_{down_2}), (y - y_{up}) \right). \quad (22)$$

- The circular curve shown in Figure 2 is defined as

$$\mathcal{K}_r = \left\{ (x, y), \theta_{\min} \leq \Theta(x, y) \leq \theta_{\max}, \right. \\ \left. r_{down} \leq \rho(x, y) \leq r_{up} \right\} \quad (23)$$

$\theta_{\min} < \theta_{\max}$  are two limiting angles for the road boundaries (with  $\theta_{\max} \leq \theta_{\min} + 2\pi$ ),  $(x_c, y_c)$  denotes the center of the road circle,  $0 < r_{up} < r_{down}$  the radius of the bigger and smaller circle respectively,

$$r = \rho(x, y) := \sqrt{(x - x_c)^2 + (y - y_c)^2} \\ \theta = \Theta(x, y) := \arctan\left(\frac{y - y_c}{x - x_c}\right) + k_{x,y}\pi, \quad \text{with } k_{x,y} \in \mathbb{Z}$$

denotes a continuous representation on  $\mathcal{K}_r$  such that  $x - x_c = r \cos(\theta)$  and  $y - y_c = r \sin(\theta)$ . The level set function is defined as

$$g(x, y) = \max \left( \rho(x, y) - r_{up}, -(\rho(x, y) - r_{down}), \right. \\ \left. \Theta(x, y) - \theta_{\max}, -(\Theta(x, y) - \theta_{\min}) \right). \quad (24)$$

**Remark 4.1.** A general and simple rule for constructing Lipschitz continuous level set functions is the following. Assume that  $g_1$  (resp.  $g_2$ ) are

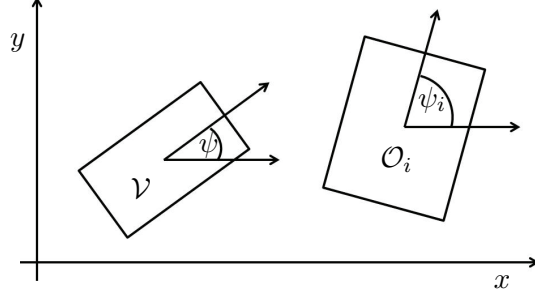


FIGURE 3. Vehicle and obstacles

Lipschitz continuous level set functions for the set  $\mathcal{K}_1$  (resp.  $\mathcal{K}_2$ ), that is,  $g_i(x) \leq 0 \Leftrightarrow x \in \mathcal{K}_i$ , for  $i = 1, 2$ . Then

$$\max(g_1(x), g_2(x)) \leq 0 \Leftrightarrow x \in \mathcal{K}_1 \cap \mathcal{K}_2 \quad (25)$$

$$\min(g_1(x), g_2(x)) \leq 0 \Leftrightarrow x \in \mathcal{K}_1 \cup \mathcal{K}_2. \quad (26)$$

Hence  $\max(g_1, g_2)$  (resp.  $\min(g_1, g_2)$ ) is Lipschitz continuous and it can be used as a level set function for  $\mathcal{K}_1 \cap \mathcal{K}_2$  (resp.  $\mathcal{K}_1 \cup \mathcal{K}_2$ ). Then more complex structures can be coded by combining (25) and (26). This is related to well-known techniques for (signed) distance functions in computational geometry (see e.g. [17, 23])

- A crossing with corner points  $(x_i, y_i)_{i=0, \dots, 3}$ : Let the upper right part be defined as  $\mathcal{K}_0 := \{x - x_0 \leq 0 \text{ or } y - y_0 \leq 0\}$ , and similarly,  $\mathcal{K}_1 := \{y - y_1 \leq 0 \text{ or } -(x - x_1) \leq 0\}$  (upper left part),  $\mathcal{K}_2 := \{-(x - x_2) \leq 0, \text{ or } -(y - y_2) \leq 0\}$  (lower left part),  $\mathcal{K}_3 := \{-(y - y_3) \leq 0, \text{ or } x - x_3 \leq 0\}$  (lower right part).

Following Remark 4.1, a level set function for  $\mathcal{K} := \bigcap_{i=0, \dots, 3} \mathcal{K}_i$  can be obtained by

$$g(x, y) = \max \left( \min(x - x_0, y - y_0), \min(y - y_1, -(x - x_1)), \right. \\ \left. \min(-(x - x_2), -(y - y_2)), \min(-(y - y_3), x - x_3) \right), \quad (27)$$

More general ways to construct level set functions for roads delimited by polygonal lines could be obtained following similar ideas. Also let us mention that the modeling of the road boundaries via piecewise cubic polynomials or B-splines can be obtained as in [18].

**4.2. Obstacles and corresponding level set functions.** Next,  $k \geq 1$  additional obstacles are considered (obstacles which are different from the road). There are modeled as disks or rectangles that the car has to avoid (the car being itself modeled in the form of a disk or a rectangle), therefore defining another type of state constraint.

Let  $X_i = (x_i, y_i)$  denote the center of obstacle  $i$ , which may depend on the time  $s$ . Let  $X = (x, y)$  denote the center of gravity of the vehicle.

**Circular obstacles.** We first consider the simpler case of circular obstacles and vehicle, where the vehicle is approximated by a closed ball  $B(X, r)$  for a given radius  $r > 0$ , and the obstacles by closed balls  $B(X_i(s), r_i)$  centered at  $X_i(s)$  and with given fixed radius  $r_i > 0$ . Then it holds  $B(X_i(s), r_i) \cap B(X, r) = \emptyset \Leftrightarrow \|X - X_i(s)\|_2 > r + r_i$ , hence there is no collision at time  $s$  if  $B(X, r) \cap \bigcup_{1 \leq i \leq k} B(X_i(s), r_i) = \emptyset$ , which amounts to saying

$$g(z, s) := \min_{1 \leq i \leq k} -(\|X - X_i(s)\|_2 - r - r_i) < 0, \quad (28)$$

where  $z = (x, y, \psi, v)$ .

**Rectangular obstacles.** We now turn on the more realistic case of rectangular obstacles and vehicle (see Fig. 3).

Let us first remark that the Gilbert-Johnson-Keehrsti distance algorithm [22] can compute the signed distance function for two convex obstacles in two dimensions and therefore can be used to detect collisions. However, since the obstacle function will have to be computed on each grid point of the discretization method, in order to be more efficient, we will look for a more straightforward analytic way to detect collision avoidance.

We assume the vehicle  $\mathcal{V}$  is a rectangle centered at  $X$  and with half lengths  $\ell = (\ell_x, \ell_y)^T$  rotated by the angle  $\psi$ , i.e.,

$$\mathcal{V} = X + R_\psi([- \ell_x, \ell_x] \times [- \ell_y, \ell_y])$$

where  $R_\psi$  denotes the rotation matrix  $R_\psi := \begin{pmatrix} \cos(\psi) & -\sin(\psi) \\ \sin(\psi) & \cos(\psi) \end{pmatrix}$ . We assume that each obstacle  $\mathcal{O}_i$  is also a rectangle with center  $X^i = X^i(s)$  and half lengths  $\ell^i = (\ell_x^i, \ell_y^i)^T$ . Then the four corners  $(X_j)_{1 \leq j \leq 4}$  of the vehicle with state  $z = (x, y, \psi, v)$  are determined by

$$X_j = X + R_\psi T_j \ell, \quad T_j = \begin{pmatrix} (-1)^{j-1} & 0 \\ 0 & (-1)^{\lfloor \frac{j-1}{2} \rfloor} \end{pmatrix}, \quad 1 \leq j \leq 4,$$

where  $\lfloor x \rfloor$  denotes the integer part of a real  $x$ , and in the same way the four corners  $(X_j^i)_{1 \leq j \leq 4}$  of obstacle  $\mathcal{O}_i$  (determined by its center  $X^i(s)$  and orientation  $\psi_i(s)$ ) are given by

$$X_j^i := X^i + R_{\psi_i} T_j \ell^i, \quad 1 \leq j \leq 4.$$

Furthermore, for a given  $X = (x, y)^T$ , let

$$d_\ell(X) := \max(\ell_x - |x|, \ell_y - |y|).$$

The function  $d_\ell$  is a level set function for the avoidance of  $[- \ell_x, \ell_x] \times [- \ell_y, \ell_y]$ , since  $d_\ell(X) < 0 \Leftrightarrow X \notin [- \ell_x, \ell_x] \times [- \ell_y, \ell_y]$ . For an arbitrary point  $Y \in \mathbb{R}^2$ , the following function

$$d_{\mathcal{V}}(Y) := d_\ell(R_{-\psi}(Y - X))$$

is a level set function for the avoidance of the vehicle, in the sense that  $d_{\mathcal{V}}(Y) < 0 \Leftrightarrow Y \notin \mathcal{V}$ .

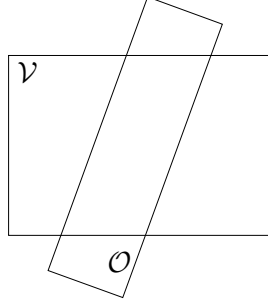


FIGURE 4. Vehicle and obstacle are not necessarily disjoint even if the condition (30) holds, i.e., even if each corner point of  $\mathcal{V}$  (resp.  $\mathcal{O}$ ) lies outside  $\mathcal{O}$  (resp.  $\mathcal{V}$ ).

In the same way,

$$d_{\mathcal{O}_i}(Y) := d_{\ell^i}(R_{-\psi_i}(Y - X^i))$$

satisfies  $d_{\mathcal{O}_i}(Y) < 0 \Leftrightarrow Y \notin \mathcal{O}_i$ .

Now we consider the following function, for  $z = (x, y, \psi, v) \in \mathbb{R}^4$  and  $s \geq 0$ :

$$g(z, s) := \max_{1 \leq i \leq k} \left( \max_{j=1, \dots, 4} d_{\mathcal{V}(z)}(X_j^i(s)), \max_{j=1, \dots, 4} d_{\mathcal{O}_i(s)}(X_j(z)) \right). \quad (29)$$

If the positions of the obstacles do not depend of time, we can define  $g(z)$  in the same way without time dependency. Here we have denoted  $\mathcal{V}(z)$  and  $X_j(z)$  for the center and the corners of the vehicle to stress the dependance on the state variable  $z$ , and  $X_j^i(s)$  to emphasize the time dependency of the obstacle corners.

The function  $g$  will serve as a level set function for obstacle avoidance. Presently, from the definition of the  $g$  function, it holds:

**Lemma 4.2.** *The function  $g$  is Lipschitz continuous and*

$$g(z, s) < 0 \Leftrightarrow \forall i, j : X_j(z) \notin \mathcal{O}_i(s) \text{ and } \forall i, j : X_j^i(s) \notin \mathcal{V}(z) \quad (30)$$

However, we aim to characterize the fact that the obstacle and vehicle are disjoint, i.e.,

$$\mathcal{V}(z(s)) \cap \left( \bigcup_{1 \leq i \leq k} \mathcal{O}_i(s) \right) = \emptyset. \quad (31)$$

In general, the condition  $g(z(t_n), t_n) < 0$  at a given time  $t_n$ , i.e., condition (30), is not sufficient to ensure that (31) holds at time  $t_n$ , as shown by the counter-example illustrated in Fig. 4.

In order to use condition (30) as a sufficient condition, we need furthermore the following result.

**Lemma 4.3.** *We consider that the vehicle as well as all obstacles  $\mathcal{O}_i$  are rectangles. Let  $\underline{d}$  be the minimal car and obstacle half-lengths*

$$\underline{d} := \min(\ell_x, \ell_y, \min_{1 \leq i \leq k} (\min(\ell_x^i, \ell_y^i))) \quad (32)$$

and let  $\bar{v}$  be an upper bound for all relative velocities between the car corners and the obstacle corners involved in the computations

$$\bar{v} := \max_{s \in [0, T]} \max_{1 \leq j, j' \leq 4} \max_{1 \leq i \leq k} \|\dot{X}_j(s) - \dot{X}_{j'}^k(s)\|. \quad (33)$$

Assume that

- (i) at time  $s = t_n$ , the vehicle is disjoint from the obstacles (i.e., (31) holds);
- (ii) at time  $s = t_{n+1}$  condition (30) is fulfilled (i.e.,  $g(z(t_{n+1}), t_{n+1}) < 0$ );
- (iii) the time step satisfies

$$\Delta t_n := t_{n+1} - t_n < \underline{d}/\bar{v}. \quad (34)$$

Then at time  $s = t_{n+1}$  the vehicle is also disjoint from the obstacles (i.e., (31) holds with  $s = t_{n+1}$ ).

**Remark 4.4.** *An upper bound of (33), for  $\bar{v}$ , can be obtained on a given computational bounded domain  $\Omega$  and for given obstacle parameters.*

*Proof.* of Lemma 4.3. Assume, to the contrary, that an intersection occurs at time  $t_{n+1}$ , i.e., (31) is not satisfied for a given obstacle. while (30) is fulfilled at time  $t_{n+1}$  due to assumption (ii). Let us show, in that case, that one point  $M(s)$  of the obstacle (or, resp., the vehicle) will run through the vehicle (or, resp., the obstacle) for a distance of at least  $d \geq \underline{d}$ . This will imply in particular that

$$\bar{v} \Delta t_n \geq \int_{t_n}^{t_{n+1}} v(s) ds \geq \underline{d} \quad (35)$$

(where  $v(s)$  denotes the relative velocity of the considered point  $M(s)$ ), which will contradict (34).

Precisely, assume that one obstacle (say  $\mathcal{O} := \mathcal{O}^i$ ) intersects the vehicle  $\mathcal{V}$  at time  $t_{n+1}$ . Let  $\ell_{\mathcal{V}} := \min(\ell_x, \ell_y)$  denote the minimum half-length of the edges of the vehicle, as well as  $\ell_{\mathcal{O}} := \min(\ell_x^i, \ell_y^i)$ .

Because  $\mathcal{O}$  and  $\mathcal{V}$  will play a symmetric role from now on, we can also assume that  $\ell_{\mathcal{O}} \geq \ell_{\mathcal{V}}$  (otherwise we may exchange  $\mathcal{O}$  and  $\mathcal{V}$  in the forthcoming argument).

Let  $I_{\mathcal{V}}$  denote the set of points  $x \in \mathcal{V}$  such that  $d(x, \mathbb{R}^2 \setminus \mathcal{V})$  is maximal. We notice that  $I_{\mathcal{V}}$  is a segment and that

$$I_{\mathcal{V}} := \{x \in \mathbb{R}^2 \mid d(x, \mathbb{R}^2 \setminus \mathcal{V}) = \ell_{\mathcal{V}}\}.$$

( $I_{\mathcal{V}}$  may be reduced to a point for a squared vehicle.) This set is illustrated in Figure 5.

In the same way for the given obstacle  $\mathcal{O} = \mathcal{O}^i$ , let

$$I_{\mathcal{O}} := \{x \in \mathbb{R}^2 \mid d(x, \mathbb{R}^2 \setminus \mathcal{O}) = \ell_{\mathcal{O}}\}.$$



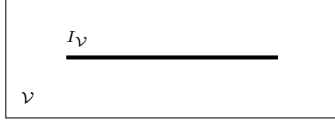


FIGURE 5. The interval  $I_{\mathcal{V}}$  associated to the vehicle  $\mathcal{V}$ .

Let us first show that (at time  $t_{n+1}$ ):

$$\mathcal{O} \cap I_{\mathcal{V}} = \emptyset.$$

If, to the contrary,  $\mathcal{O} \cap I_{\mathcal{V}} \neq \emptyset$  (at time  $t_{n+1}$ ), then the point  $M(t_{n+1}) \in \mathcal{O} \cap I_{\mathcal{V}}$ , attached to  $\mathcal{O}$ , has run through the vehicle for a distance of at least  $\ell_{\mathcal{V}}$  between  $t_n$  and  $t_{n+1}$  (since at time  $t_n$ , by assumption (i),  $M(t_n) \in \mathcal{O}(t_n)$  was outside of  $\mathcal{V}(z(t_n))$ ). By using the fact that  $\ell_{\mathcal{V}} \geq \underline{d}$ , and (35), we arrive to the contradiction  $\bar{v}\Delta t \geq \underline{d}$ .

In the same way, we can conclude to

$$\mathcal{V} \cap I_{\mathcal{O}} = \emptyset.$$

Let us now consider an edge of the obstacle (say an edge  $[A; B]$  of  $\mathcal{O}$ ) that overlaps the vehicle  $\mathcal{V}$ , but with  $A, B \notin \mathcal{V}$ , as illustrated in Figs. 9-10. Furthermore we notice that  $[A; B] \cap I_{\mathcal{V}} \subset \mathcal{O} \cap I_{\mathcal{V}} = \emptyset$ , hence  $[A; B] \cap I_{\mathcal{V}} = \emptyset$ .

Notice that for any edge of  $\mathcal{O}$  that overlaps the vehicle, and since the corner points are outside of  $\mathcal{V}$ , there can be only two generic situations: either the edge intersects two parallel edges of the vehicle, or the edge intersects two neighboring edges of the vehicle. (The particular cases when one edge overlaps the vehicle and intersects another edge, or overlaps one corner point cannot happen, since each corner point of each considered object is assumed to be strictly outside the other object.)

Also, it is not possible that  $I_{\mathcal{V}}$  lies in between two parallel edges, say between the edges  $(A, B)$  and  $(A', B')$  as in Fig. 6, since otherwise the segment  $I_{\mathcal{V}}$  would be fully included in the object  $\mathcal{O} = (ABB'A')$  and this case has already been excluded. Some forbidden situations are also depicted in Fig. 6–8.

FIGURE 6. Forbidden situation:  $\mathcal{O} \cap I_{\mathcal{V}} \neq \emptyset$

We are left with three possible situations, depicted in Figures 9 and 10 (left and right):

- case 1: each one of two parallel edges  $[A; B]$  and  $[A'; B']$  of  $\mathcal{O}$  intersect two parallel edges of  $\mathcal{V}$ , as in Fig. 9;
- case 2: only one edge  $[A; B]$  of  $\mathcal{O}$  intersects two parallel edges of  $\mathcal{V}$ , as illustrated in Fig. 10 (left);

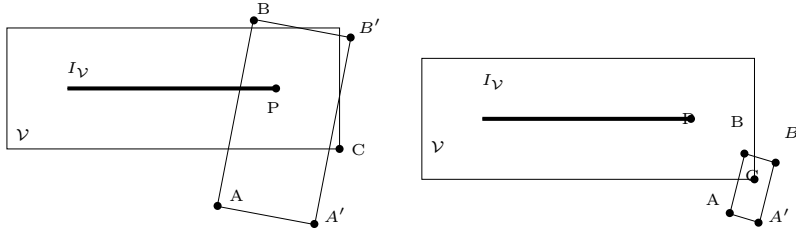


FIGURE 7. Forbidden situations:  $\mathcal{O} \cap I_{\mathcal{V}} \neq \emptyset$  (left), in the right figure, (30) does not hold since  $B \in \mathcal{V}$  and  $C \in \mathcal{O}$

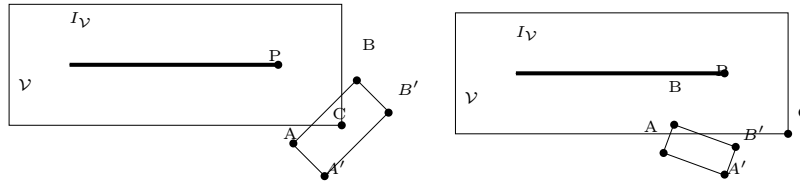


FIGURE 8. Forbidden situations: (30) does not hold, since  $C \in \mathcal{O}$  (left), or  $B \in \mathcal{V}$  (right)

- case 3: no edge of  $\mathcal{O}$  intersect two parallel edges of  $\mathcal{V}$ ; rather, the two edges  $[A; B]$  and  $[A'; B']$  intersects two neighboring edges of  $\mathcal{V}$ , as in Fig. 10 (right).

In all these cases, the distance between the parallel edges  $[A; B]$  and  $[A', B']$  of  $\mathcal{O}$  (which is greater or equal to  $2\ell_{\mathcal{O}}$ ), is smaller than the distance from  $P$  (an extremal point of  $I_{\mathcal{V}}$ ) to a corner point  $C$  of  $\mathcal{V}$  (see Figures 9-10), with  $d(P, C) = d(C, I_{\mathcal{V}}) = \sqrt{2}\ell_{\mathcal{V}}$ . Therefore

$$2\ell_{\mathcal{O}} \leq d(P, C) = \sqrt{2}\ell_{\mathcal{V}}.$$

This is in contradiction with the fact that  $\ell_{\mathcal{V}} \leq \ell_{\mathcal{O}}$  (and  $\ell_{\mathcal{V}} > 0$ ), and concludes the proof of the Lemma.  $\square$

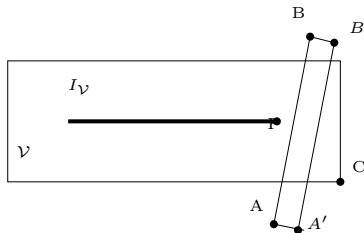


FIGURE 9. Case 1: two parallel edges of  $\mathcal{O}$  intersect two parallel edges of  $\mathcal{V}$ .

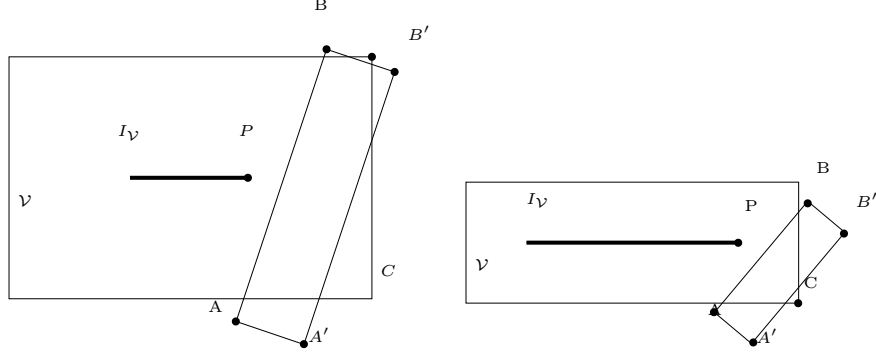


FIGURE 10. Case 2 (left): only one edge of  $\mathcal{O}$  (the segment  $[A; B]$ ) intersects two parallel edges of  $\mathcal{V}$ . Case 3 (right): no edge of  $\mathcal{O}$  intersects two parallel edges of  $\mathcal{V}$ .

From Lemma 4.2 and Lemma 4.3 we deduce, in other words, assuming that  $\Delta t_n \leq \overline{\Delta t} := \underline{d}/\bar{v}$ , that if there is no collision at time  $t_n$ , then (30) characterizes the fact that there is no collision at time  $t_{n+1}$ .

**Corollary 4.5.** *In particular, considering a trajectory reconstruction with time steps  $t_n = n\Delta t$ , if at the initial time  $t_0 = 0$  the vehicle and the obstacles are disjoint (i.e. condition (31) holds) and if  $0 \leq \Delta t < \overline{\Delta t}$  where  $\overline{\Delta t} := \bar{d}/\bar{v}$  and  $\bar{d}$  and  $\bar{v}$  are as in (32)-(33), and if*

$$g(z(t_n), t_n) < 0, \quad \forall n = 1, \dots, N \quad (36)$$

(i.e., condition (30) holds at each successive time step) then there is no collision at all discrete time steps  $t_n$ , i.e.:

$$\mathcal{V}(z(t_n)) \cap \left( \bigcup_{1 \leq i \leq k} \mathcal{O}_i(t_n) \right) = \emptyset \quad \forall n = 1, \dots, N.$$

**Remark 4.6.** *By using Lemma 3.7, the condition (36) will be fulfilled along all time steps of the trajectory reconstruction.*

*Therefore the obstacle function  $g$  can be used for collision avoidance in our HJB framework as long as the time-step condition (34) holds.*

**Moving obstacles.** There are many possible motions for the obstacles, which can be considered. One of them is a linear motion along a straight path:

$$x_i(s) = x_{i0} + v_{ix} s, \quad (37a)$$

$$y_i(s) = y_{i0} + v_{iy} s, \quad (37b)$$

$$\psi_{i0} = \arctan \left( \frac{v_{iy}}{v_{ix}} \right), \quad (37c)$$

where  $(v_{ix}, v_{iy})$  is the constant velocity of the obstacle  $i$  and  $\theta_{i0}$  is the initial angle which is constant during the motion. Another possibility is a motion

along a curved road, e.g. a rotation around a center  $(c_x, c_y)$  with constant angular velocity  $\omega_i$  starting with the angle  $\theta_{i0}$ , i.e.

$$x_i(s) = a \cos(\theta_{i0} + w_i s) + c_x, \quad (38a)$$

$$y_i(s) = a \sin(\theta_{i0} + w_i s) + c_y, \quad (38b)$$

$$\psi_{i0}(s) = (\theta_{i0} + w_i s) - \frac{\pi}{2}. \quad (38c)$$

**Collision avoidance between time steps.** It is still possible that a violation of collision avoidance appears between time steps  $t_n$  and  $t_{n+1}$ . In order to control this error or to avoid it, we first start with the following result.

Let  $d(A, B)$  be the signed distance between two sets  $A, B$ , defined by

$$d(A, B) := \min(\mu(A, B), \mu(B, A))$$

where  $\mu(A, B) := \min_{x \in A} d_B(x)$  and  $d_B(\cdot)$  is the signed distance to  $B$ . We notice that if  $A \cap B = \emptyset \Leftrightarrow d(A, B) > 0$ , and in that case  $d(A, B) = \min_{x \in A, y \in B} \|x - y\|_2$ .

Let  $d(t) := d(\mathcal{V}(z(t)), \cup_i \mathcal{O}_i(t))$  be the signed distance between the vehicle and the set of obstacles.

**Lemma 4.7.** *Let  $\bar{v}$  be the maximum relative velocity between the vehicle and the obstacles (as in (33)), and  $\Delta t_n := t_{n+1} - t_n$ .*

(i) *It holds*

$$d(t) \geq \min(d(t_n), d(t_{n+1})) - \bar{v} \frac{\Delta t_n}{2}, \quad \forall t \in [t_n, t_{n+1}]. \quad (39)$$

(ii) *In the same way, in the case of the vehicle and the obstacles are modeled as rectangles, using (29), it holds*

$$g(z(t), t) \leq \max(g(z(t_n), t_n), g(z(t_{n+1}), t_{n+1})) + \bar{v} \frac{\Delta t_n}{2} \quad \forall t \in [t_n, t_{n+1}]. \quad (40)$$

*Proof.* We only give a sketch of the proof of (i). The maximum distance that a point of the vehicle can cover during a time step  $\tau$  is  $\tau \bar{v}$ . Taking into account the relative velocity, we obtain that on the time interval  $t \in [t_n, t_n + \Delta t_n/2]$ ,  $d(t) \geq d(t_n) - \bar{v}(\Delta t_n/2)$ . In the same way one can prove that for  $t \in [t_n + \Delta t_n/2, t_{n+1}]$ ,  $d(t) \geq d(t_{n+1}) - \bar{v}(\Delta t_n/2)$ . Hence the desired formula (39) follows.  $\square$

*Minimal distance between time steps.* As a first result, if

$$\min(d(t_n), d(t_{n+1})) \geq 0$$

(no collision at time steps  $t_n$  and  $t_{n+1}$ ), then

$$\min_{t \in [t_n, t_{n+1}]} d(t) \geq -\bar{v} \frac{\Delta t_n}{2}.$$

*Secure collision avoidance.* On the other hand if we want to secure collision avoidance, we can consider a small number  $\varepsilon > 0$  and require that

$$\min(d(t_n), d(t_{n+1})) > \varepsilon, \quad (41)$$

Then using (39) it holds:

$$\frac{\Delta t_n}{2} < \varepsilon/\bar{v} \quad \Rightarrow \quad \forall t \in [t_n, t_{n+1}], \quad d(t) > 0. \quad (42)$$

**Remark 4.8.** Consider the case of vehicle and obstacle being rectangles, let  $g$  be defined as in (29),  $\tilde{g}_\varepsilon$  the following obstacle function

$$\tilde{g}_\varepsilon := g + \varepsilon,$$

and assume that

$$\max(\tilde{g}_\varepsilon(z(t_n), t_n), \tilde{g}_\varepsilon(z(t_{n+1}), t_{n+1})) \leq 0, \quad (43)$$

then using (40), it holds

$$\frac{\Delta t_n}{2} \leq \varepsilon/\bar{v} \quad \Rightarrow \quad \max_{t \in [t_n, t_{n+1}]} g(z(t), t) \leq -\varepsilon + \bar{v} \frac{\Delta t_n}{2} \leq 0.$$

Therefore, if there is collision avoidance neither at time  $t_n$  nor at time  $t_{n+1}$  within a margin of length  $\varepsilon > 0$ , (i.e., if (43) holds), and if  $\Delta t_n/2 \leq \varepsilon/\bar{v}$ , then collision avoidance will also hold on  $[t_n, t_{n+1}]$ .

## 5. NUMERICAL SIMULATIONS

Here we plot some numerical simulations for different scenarios. Throughout the paper, we use the 4-dimensional point mass model (3). In the present work and for solving the HJ equation (14), we have used an Essentially Non Oscillating (ENO) finite difference scheme of second order for the spatial discretization of the Hamilton-Jacobi-Bellman partial differential equation. It is coupled with an Euler forward scheme (RK1) in time (see [28]). The computations have been performed by using the ROC-HJ parallel solver [10].

**5.1. Scenario 1: straight road with a fixed rectangular obstacle.** We first consider a straight road configuration with only one fixed obstacle, as in Figure 11. The aim for the vehicle (represented by the red box) is to minimize the time to reach the target area (the target, delimited by a blue line) and to avoid the obstacle (blue box). More precisely the parameters of the problem are:

- Vehicle parameters and initial position:
  - half lengths  $\ell_x = \ell_y = 1.0$ ,
  - initial value  $z_0 = (x_0, y_0, \theta_0, v_0) = (-40.0, -1.5, 0.0, 35.0)$
- Target:  $\Omega = \{(x, y), x \geq 0, |y| \leq 3.5\}$ .
- Obstacle parameters: one fixed obstacle with half lengths  $\ell_x^1 = 1.0$  and  $\ell_y^1 = 1.0$ , centered at  $X^1 = (-10.0, -1.5)$
- Road parameters: straight road as described in (18).

In Fig. 12 the capture basin is represented in blue and for different times. At time  $t = 0$  only the points which are in the target area (and away from the road boundary) are thus represented. Then the evolution of the union of (backward) reachable sets is represented for different times. At time  $t_i$ ,

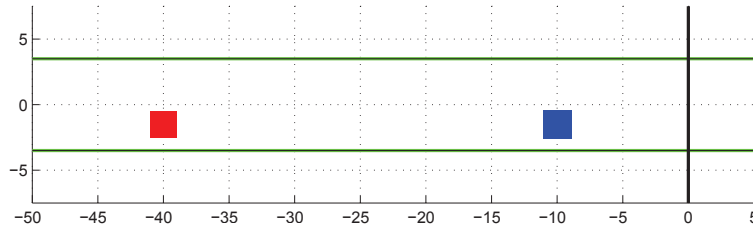


FIGURE 11. Car traffic scenario: the red car represents the controlled vehicle, the blue car is an obstacle (and is fixed in this scenario)

the region  $Cap(t_i)$  represents the set of starting points which can reach the target avoiding the obstacle within  $t_i$  seconds.

Next, in Fig. 13, we have represented the initial position of the vehicle as well as the reconstruction of the optimal trajectory (as black line): the car (red rectangle) drives from left to right in Fig. 13 and it overtakes the fixed obstacle.

The uncolored region (white part) in Fig. 13 corresponds to the area of starting points from which the blue car cannot reach the target, whatever maneuver is undertaken (mainly, its velocity is too high to avoid a collision with the blue car). Hence the trajectories from these points are infeasible, and such starting points do not belong to the backward reachable sets.

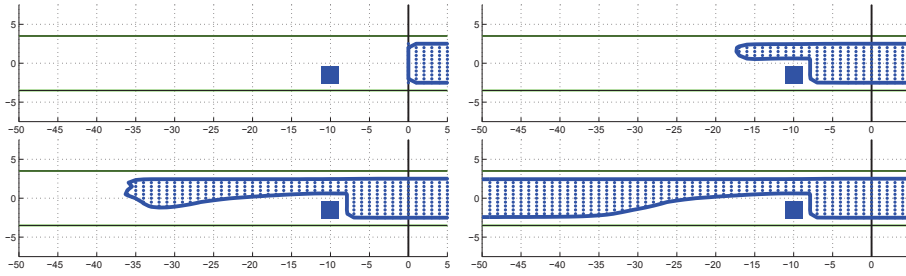


FIGURE 12. (Scenario 1) capture basins  $Cap(t_i)$  for different times ( $t_1 = 0s, t_2 = 0.5s, t_3 = 1s, t_4 = 1.5s$ )

**Convergence test (scenario 1).** For testing the stability of the HJ approach we first perform a convergence analysis with respect to mesh grid refinement. We define a grid on the state space (or computational domain)

$$(x, y, \psi, v) \in [-50, 10] \times [-4, 4] \times [-1, 1] \times [5, 65]. \quad (44)$$

A variable number of grid points in the  $(x, y)$  variables is used, given by

$$N_x = 35 \cdot 2^m \text{ and } N_y = 4 \cdot 2^m,$$

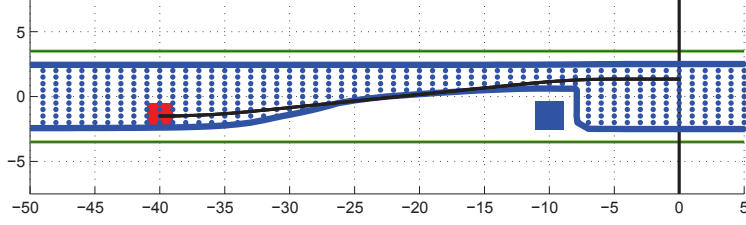


FIGURE 13. (Scenario 1) capture basin  $Cap(2)$  and optimal trajectory.

depending on an integer parameter  $m \in \{1, \dots, 5\}$ . The number of grid points in the  $\psi$  and  $v$  variables are fixed and given by

$$N_\psi = 20 \quad \text{and} \quad N_v = 6.$$

The errors are computed by using a reference value function  $v_{ex}$  obtained for  $m = 5$  (i.e.,  $N_x = 1120$ ,  $N_y = 128$ ). Furthermore the following CFL (Courant-Friedrich-Levy) restriction

$$\Delta t \sum_{j=1}^4 \frac{\|f_j\|_\infty}{\Delta x_j} \leq \frac{1}{2} \quad (45)$$

is used for the stability of the finite difference scheme (where  $f_j$  are the components of the dynamics  $f$  and  $\|f_j\|_\infty$  is the supremum norm of  $f_j$  on the computational domain,  $\Delta z := (\Delta x_j)_{1 \leq j \leq 4} \equiv (\Delta x, \Delta y, \Delta \psi, \Delta v)$  are the different grid mesh steps).

The results are given in Table 1. For a given grid mesh  $(z_i)$  and corresponding vector of mesh sizes  $\Delta z$ , the local error at grid point  $z_i$  is  $e_i := v(z_i) - v_{ex}(z_i)$  and the  $L^\infty$ ,  $L^1$  and  $L^2$  errors are defined as follows:

$$e_{L^\infty} := \max_i |e_i|, \quad e_{L^1} := |\Delta z| \sum_i |e_i|, \quad e_{L^2} := |\Delta z|^{1/2} \left( \sum_i e_i^2 \right)^{1/2} \quad (46)$$

where  $|\Delta z| := \prod_{1 \leq j \leq 4} |\Delta x_j|$ .

In Table 1, in order to evaluate numerically the order of convergence for a given  $L^p$  norm, the estimate  $\alpha_m := \frac{\log(e^{(m-1)}/e^{(m)})}{\log(2)}$  is used for corresponding values  $N_x = 35 \cdot 2^m$  and  $N_y = 4 \cdot 2^m$  (i.e., the mesh steps  $N_x$  and  $N_y$  are refined by 2 between two successive computations).

We observe a convergence of order roughly 2 even for the ENO2-RK1 scheme which in principle is only first order in time. This is due to the fact that the dynamics is close to a linear one in this case (we have also tested a similar RK2 scheme, second order in time, which gives similar convergence results on this example).

**Comparison with a direct method (scenario 1).** For this particular scenario, we validate the results by comparing with a direct optimal control approach for calculating the reachable set. The simulations are obtained by using the OCPID-DAE1 Software [20], and following the approach described

$N_x$	$N_y$	$\Delta t$	$e_{L^\infty}$	order	$e_{L^1}$	order	$e_{L^2}$	order	CPU time (in s)
70	8	3.97 E-3	0.489	–	1.126	–	6.769	–	0.34
140	16	2.03 E-3	0.078	2.64	0.337	1.73	2.255	1.58	1.50
280	32	1.02 E-3	0.026	1.60	0.118	1.52	0.795	1.50	9.20
560	64	0.51 E-3	0.006	2.07	0.030	1.98	0.207	1.94	69.40

TABLE 1. (Scenario 1) error table for varying  $(N_x, N_y)$  parameters

in [7] and [21, 33]. The resulting capture basin for time  $T = 2$  is plotted in Fig. 14 (upper graph) and it is in good correspondence with the set obtained by the HJ approach (Fig. 14, lower graph). Notice that we can only expect that both computed capture basins are equal up to some accuracy of the order  $O(\Delta x)$  (with  $\Delta x = 1$  in this figure).

**Remark 5.1.** The advantage of using a direct optimal control approach (like the OCPID-DAE1 Software) is that it is able to deal with a greater number of states variables, which is necessary whenever we need a precise car-model, close to the behavior of a real car. However the handling of state constraints (in particular obstacles with nonsmooth boundaries) sometimes leads to numerical difficulties in order to compute feasible trajectories.

On the other hand the pde solver (like ROC-HJ for solving Hamilton-Jacobi equations) is limited, in practice, by the number of state variables, because it requires to solve a pde with as many dimensions as the number of state variables. However if the dimension can be processed on a given computer system, then the HJ approach requests only the Lipschitz property of the functions describing the dynamics and the state constraints. In particular, there is no problem for dealing with nonsmooth obstacles such as rectangular obstacles, crossing roads scenarios (more generally, it could handle polygonal roads or more complex polytopial obstacles).

Next, we consider more complex scenarios and different road geometries.

**5.2. Scenario 2: straight road with varying width.** We shall consider a highway road with varying width :

$$\begin{cases} y_{up} = 3.5 \text{ m} \\ y_{down} = -3.5 \text{ m}, \text{ if } z_1 \leq -15.0 \\ y_{down} = -7.0 \text{ m}, \text{ if } z_1 > -15.0. \end{cases}$$

This is illustrated in Figs. (20) and (21) (the road is represented with green lines). This can be interpreted as an additional exit lane appearing only for a short part of the considered road.

Two obstacles (blue rectangles) are moving with linear motion (see (37)) in the same direction as the reference vehicle (red rectangle). All object widths and lengths are here equal to  $1 \text{ m}$ . The set of blue points depicted in



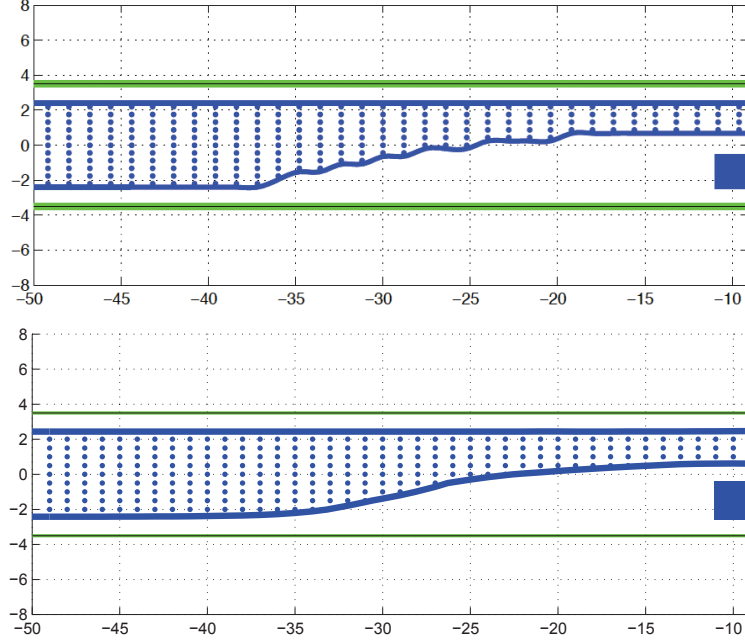


FIGURE 14. (Scenario 1) comparison of capture basins obtained with a direct method (up) and the HJ approach (down)

Figure 15 and Figure 16 is the projection on the  $(x, y)$ -panel of the capture basin for  $t_f = 2$  s, with yaw angle  $\psi(t_0) = 0$  and velocity  $v(t_0) = 35$  m s<sup>-1</sup>.

*Scenario 2a:* (see Fig. 15) In this example an overtaking maneuver is considered with one obstacle (first car) in front of the vehicle, moving forward with velocity  $10$  m s<sup>-1</sup> and to be overtaken, and a second obstacle (second car) next to the vehicle also moving forward but with higher velocity  $20$  m s<sup>-1</sup> and blocking the maneuver. In Fig. 15, the initial position of the vehicle and of the obstacle cars are depicted at the initial time  $t_0 = 0$ . The parameters used in the computation for this figure are the grid with  $(N_x, N_y) = (70, 12)$ ; the trajectory (black line) is starting from  $(x(0), y(0)) = (-40.0, -1.5)$ ,  $\psi(0) = 0$  and  $v(0) = 35$  m s<sup>-1</sup>, a first obstacle car takes initial values  $(x(0), y(0)) = (-10, -1.5)$ , with  $\psi(0) = 0$  and a constant velocity  $v(0) = 10$  m s<sup>-1</sup>; a second obstacle car takes initial values  $(x(0), y(0)) = (-40, 1.5)$ , with  $\psi(0) = 0$  and a constant velocity  $v(0) = 20$  m s<sup>-1</sup>.

*Scenario 2b:* (see Fig. 16) This example is similar to the previous one except for the fact that the second car is next to the first one at initial time. The second obstacle car now takes initial values  $(x(0), y(0)) = (-10, 1.5)$ , and other parameters are otherwise unchanged.

The capture basin is the set of initial points of  $\mathbb{R}^4$  (according to the model (3)) for which the collision can be avoided and the target can be reached

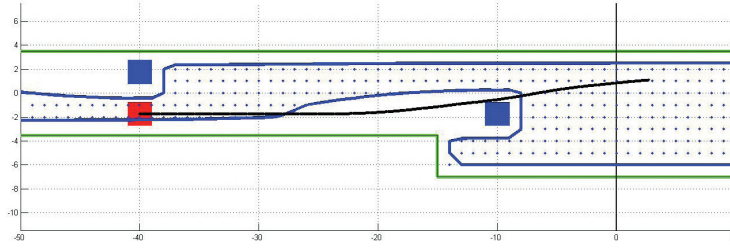


FIGURE 15. (Scenario 2a) capture basin  $Cap(2)$ . Here the blue car above the vehicle (red car) is moving forward (to the right) at constant velocity  $20 \text{ m s}^{-1}$ , the second blue car also moves forward at velocity  $10 \text{ m s}^{-1}$ , while the controlled vehicle (red car) has an initial velocity of  $35 \text{ m s}^{-1}$ .

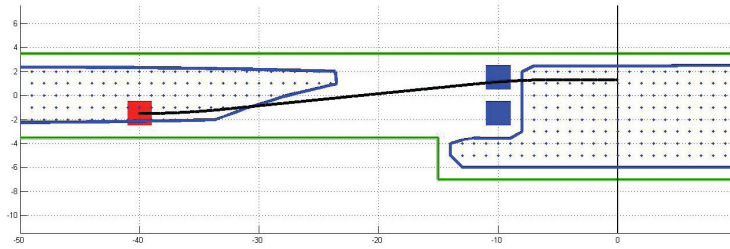


FIGURE 16. (Scenario 2b) capture basin  $Cap(2)$ , non-connected, in contrary to Fig. 15

within  $t_f$  seconds (so that the final conditions  $x(t_f) \geq 0$  and  $\psi(t_f) = 0$  are satisfied). The set of blue points depicted in Figure 16 (also in Figure 15) is the projection on the  $(x, y)$ -plane of the capture basin, where the yaw angle and the velocity are fixed to the values  $\psi(t_0) = 0$  and  $v(t_0) = 35 \text{ m s}^{-1}$  (which corresponds to the initial values of the car when the maneuver starts).

By starting in the blue region, the vehicle (in red) can avoid a collision. On the other hand, starting from a point in the white area will lead to infeasibility, i.e., the vehicle will either go outside the road or will collide with the obstacle before being able to reach the target area.

In both figures we notice an extra part of the backward reachable set which lays below the second obstacle. This is due to the varying road width, and it means that the vehicle may also start from the exit lane.

In Figure (16) the capture basin is not connected which means that the vehicle (red rectangle) can avoid a collision by starting the maneuver either leaving the obstacles behind (since it is faster no crash will occur), by bypassing the second car from the exit lane or by starting from a sufficiently large distance behind the two obstacles depending on its  $y$  position (about

24 m if the reference vehicle starts in the first lane and 14 m if it starts in the second lane).

The optimal trajectory (black line) seems to overlap the obstacles and their trajectories before reaching the target set. This is because the blue rectangles only show the initial position of the obstacles at time  $t_0$ , and not the evolution of their linear motions in the time interval  $[t_0, t_f]$ .

**5.3. Scenario 3: curved road with fix or moving obstacles.** The road shape is now described by the set of equations (23) with a 7 m width and a road radius of 50 m; the boundaries of the road as shown by green lines in Figure 17. The initial velocity of the vehicle is set to  $v(t_0) = 30 \text{ ms}^{-1}$ .

*Scenario 3a:* Two fixed obstacles (blue rectangles with different width and length parameters) have to be avoided by the vehicle (red square), see Fig. 17, and the target  $x \geq 0$ ,  $\psi = 0$  has to be reached, if possible, in the time interval  $[0, 5]$  measured in seconds. The first obstacle has dimensions 0.5 m and it is positioned in  $(-5, 48.25)$ . The second obstacle has width 0.5 m and length 1 m, its position is  $(-25, 45)$ .

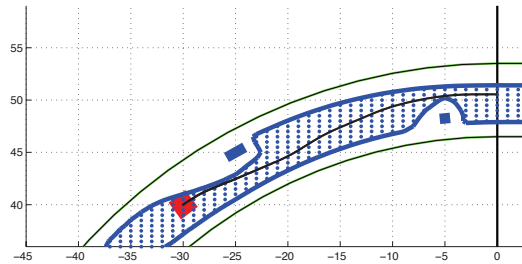


FIGURE 17. (Scenario 3a) capture basin  $Cap(5)$  for a curved road and two fixed obstacles

*Scenario 3b:* In this scenario depicted in Figure 18, the road parameters are similar, there is now only one obstacle but it is furthermore moving with a circular motion at speed of  $5 \text{ ms}^{-1}$ .

**5.4. Scenario 4: crossing road and moving obstacles.** The following scenario involves a crossing. As in (27) the width of the four streets involved in the crossing can be different.

Here, the horizontal lower and upper road bounds and the vertical limits are different, as illustrated in Fig.19. An object (blue rectangle) of dimensions 1 m is traveling from left to right from position  $(-10.0, -2.0)$  meters with speed  $5 \text{ ms}^{-1}$  and deceleration  $5 \text{ ms}^{-2}$  (until a stop at  $t = 1 \text{ s}$ ). A second obstacle (length 1.0 m and width 2.0 m) starting from position  $(-18, 4)$  is traveling from top to bottom with speed  $5 \text{ ms}^{-1}$  and deceleration  $5 \text{ ms}^{-2}$  (until a stop at  $t = 1 \text{ s}$ ). Within time  $t_f = 2.5 \text{ s}$  the red square of dimensions 1.0 m has to reach one of the three targets at the end of each road: top (with yaw angle  $\frac{\pi}{2}$ ), bottom (with yaw angle  $-\frac{\pi}{2}$ ) or right (with yaw angle 0). At

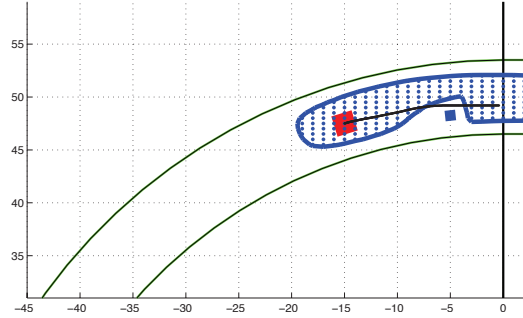


FIGURE 18. (Scenario 3b) capture basin  $Cap(5)$  for a curved road and one moving obstacle

the initial speed  $v(t_0) = 20.0 \text{ m s}^{-1}$  and for an initial position as in Fig.19, the red vehicle is able to leave the crossing before the second obstacle enters the center of the crossing. In this example the optimal trajectory (black line) will steer to overtake the obstacle in the front and it will also decelerate to avoid the second obstacle.

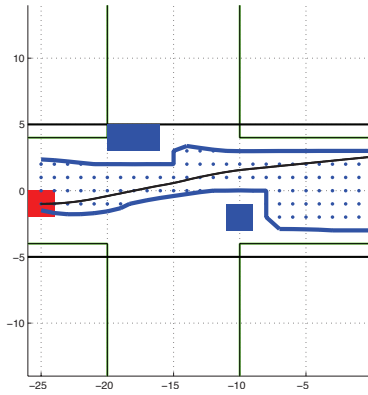


FIGURE 19. (Scenario 4) capture basin  $Cap(2.5)$  for a crossing with one fixed and one moving obstacle

## 6. CONCLUSION

In this paper, we have shown the feasibility of the HJB approach for computing target regions for vehicle collision avoidance problems.

We have modeled the vehicle avoidance problem by using a 4-dimensional "point mass" model to describe the vehicle, with different road geometries as well as different obstacles (fixed or moving obstacle vehicles). We make use of level-set functions in order to represent the reachable sets, the obstacles

(road boundaries), the obstacle avoidance (between possibly moving vehicles). This complex situation leads in general to non-convex reachable sets. These sets can be represented by using level sets of functions encoded on a grid mesh. The HJB approach turns out to be a powerful tool especially for complicated road geometries and multiple obstacles, and can handle general nonlinear dynamics. An avoidance procedure for the avoidance of rectangular vehicles has been justified in detail within the HJB framework.

A next and challenging step would be to analyze the present approach using more precise models, such as the 7-dimensional "single track" model in [33, 32]. Ongoing works also concern the sensitivity of the secure region with respect to small disturbances of the data.

#### REFERENCES

- [1] M. Althoff. *Reachability Analysis and its Application to the Safety Assessment of Autonomous Cars*. PhD thesis, Fakultät für Elektrotechnik und Informationstechnik, TU München, Germany, Aug. 2010. <http://nbn-resolving.de/urn/resolver.pl?urn:nbn:de:bvb:91-diss-20100715-963752-1-4>.
- [2] M. Althoff. An introduction to cora 2015. In *Proc. of the Workshop on Applied Verification for Continuous and Hybrid Systems*, 2015.
- [3] M. Althoff and B. H. Krogh. Zonotope bundles for the efficient computation of reachable sets. In *Proc. of the 50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC 2011), Orlando, FL, USA, December 12–15, 2011*, pages 6814–6821. Institute of Electrical and Electronics Engineers (IEEE), 2011.
- [4] M. Assellaou, O. Bokanowski, A. Désilles, and H. Zidani. Value function and optimal trajectories for a maximum running cost control problem with state constraints. Application to an abort landing problem. Preprint, HAL Id: hal-01484190, March 2017. <https://hal.archives-ouvertes.fr/hal-01484190>.
- [5] J.-P. Aubin, A. M. Bayen, and P. Saint-Pierre. *Viability theory. New directions*. Springer, Heidelberg, second edition, 2011. First edition: J.-P. Aubin in *Systems & Control: Foundations & Applications*, Birkhäuser Boston Inc., Boston, MA, 2009.
- [6] J.-P. Aubin and A. Cellina. *Differential Inclusions*, volume 264 of *Grundlehren der mathematischen Wissenschaften*. Springer-Verlag, Berlin–Heidelberg–New York–Tokyo, 1984.
- [7] R. Baier, M. Gerdts, and I. Xausa. Approximation of reachable sets using optimal control algorithms. *Numer. Algebra Control Optim.*, 3(3):519–548, 2013.
- [8] M. Bardi and I. Capuzzo-Dolcetta. *Optimal Control and Viscosity Solutions of Hamilton-Jacobi-Bellman Equations*. Systems & Control: Foundations & Applications. Birkhäuser Boston Inc., Boston, MA, 1997. With appendices by Maurizio Falcone and Pierpaolo Soravia.
- [9] E. N. Barron and H. Ishii. The Bellman equation for minimizing the maximum cost. *Nonlinear Anal.*, 13(9):1067–1090, 1989.
- [10] O. Bokanowski, A. Désilles, H. Zidani, and Jun-Yi Zhao. User’s guide for the ROC-HJ solver: Reachability, Optimal Control, and Hamilton-Jacobi equations, May 2017. <http://uma.ensta-paristech.fr/soft/ROC-HJ/>.
- [11] O. Bokanowski, N. Forcadel, and H. Zidani. Reachability and minimal times for state constrained nonlinear problems without any controllability assumption. *SIAM J. Control Optim.*, 48(7):4292–4316, 2010.
- [12] O. Bokanowski and H. Zidani. Minimal time problems with moving targets and obstacles. In S. Bittanti, A. Cenedese, and S. Zampieri, editors, *Proceedings of the 18th*

- IFAC World Congress, August 28–September 02, 2011 in Milano, Italy*, volume 18, Part 1, pages 2589–2593, Milano, Italy, 2011. International Federation of Automatic Control.
- [13] M. G. Crandall and P.-L. Lions. Two approximations of solutions of Hamilton-Jacobi equations. *Math. Comp.*, 43(167):1–19, 1984.
- [14] A. Désilles, H. Zidani, and E. Crück. Collision analysis for an UAV. In *AIAA Guidance, Navigation, and Control Conference 2012 (GNC-12), 13–16 August 2012 in Minneapolis, Minnesota*, pages 1–23. American Institute of Aeronautics and Astronautics (AIAA), August 2012. AIAA 2012-4526.
- [15] M. Falcone. Numerical solution of dynamic programming equations. In M. Bardi and I. Capuzzo-Dolcetta, editors, *Optimal Control and Viscosity Solutions of Hamilton-Jacobi-Bellman Equations*, Systems & Control: Foundations & Applications, pages 471–504 in Appendix A. Birkhäuser Boston Inc., Boston, MA, 1997.
- [16] M. Falcone and R. Ferretti. *Semi-Lagrangian Approximation Schemes for Linear and Hamilton-Jacobi Equations*, volume 133 of *Applied Mathematics*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, USA, 2014.
- [17] P.-A. Fayolle, A. Pasko, and B. Schmitt. SARDF: Signed approximate real distance functions in heterogeneous objects modeling. In *Heterogeneous Objects Modelling and Applications: Collection of Papers on Foundations and Practice*, volume 4889 of *Lecture Notes in Comput. Sci.*, pages 118–141. Springer-Verlag, Berlin–Heidelberg, 2008.
- [18] M. Gerdt. Solving mixed-integer optimal control problems by branch&bound: a case study from automobile test-driving with gear shift. *Optimal Control Appl. Methods*, 26(1):1–18 (2005), 2005.
- [19] M. Gerdt. A variable time transformation method for mixed-integer optimal control problems. *Optimal Control Appl. Methods*, 27(3):169–182, 2006.
- [20] M. Gerdt. User’s Guide for OCPID-DAE1 – Optimal Control and Parameter Identification with Differential-Algebraic Equations of Index 1, March 2013. <http://www.optimal-control.de/>.
- [21] M. Gerdt and I. Xausa. Avoidance trajectories using reachable sets and parametric sensitivity analysis. In *System Modeling and Optimization. 25th IFIP TC 7 Conference on System Modeling and Optimization (CSMO 2011), Berlin, Germany, September 12–16, 2011. Revised Selected Papers*, volume 391 of *IFIP Adv. Inf. Commun. Technol.*, pages 491–500. Springer, Heidelberg–Dordrecht–London–New York, 2013.
- [22] E. G. Gilbert, D. W. Johnson, and S. Sathya Keerthi. A fast procedure for computing the distance between complex objects in three-dimensional space. *IEEE J. Robot. Autom.*, 4(2):193–203, 1988.
- [23] J. C. Hart. Sphere tracing: a geometric method for the antialiased ray tracing of implicit surfaces. *The Visual Computer*, 12(10):527–545, 1996.
- [24] R. Kianfar, P. Falcone, and J. Fredriksson. Reachability analysis of cooperative adaptive cruise controller. In Yin Hai Wang, J. Sanchez-Medina, and Guohui Zhang, editors, *Proceedings of the 2012 15th International IEEE Conference on Intelligent Transportation Systems (ITSC 2012) in Anchorage, Alaska, USA, September 16–19, 2012*, pages 1537–1542. Institute of Electrical and Electronics Engineers (IEEE), 2012.
- [25] C. Le Guernic. *Reachability Analysis of Hybrid Systems with Linear Continuous Dynamics*. PhD thesis, École Doctorale Mathématiques, Sciences et Technologies de l’Information, Informatique, Grenoble, France, October 2009. HAL id: tel-00422569.
- [26] I. M. Mitchell, A. M. Bayen, and C. J. Tomlin. A time-dependent Hamilton-Jacobi formulation of reachable sets for continuous dynamic games. *IEEE Trans. Automat. Control*, 50(7):947–957, 2005.
- [27] J. Nilsson, J. Fredriksson, and A. C. E. Ödblom. Verification of collision avoidance systems using reachability analysis. In E. Boje and Xiaohua Xia, editors, *Proceedings of the 19th IFAC World Congress, August 24–29, 2014 in Cape Town, South Africa*,

- volume 47, Issue 3, pages 10676–10681. International Federation of Automatic Control (IFAC), 2014.
- [28] S. Osher and Chi-Wang Shu. High-order essentially nonoscillatory schemes for Hamilton-Jacobi equations. *SIAM J. Numer. Anal.*, 28(4):907–922, 1991.
- [29] J. D. L. Rowland and R. B. Vinter. Construction of optimal feedback controls. *Systems Control Lett.*, 16(5):357–367, 1991.
- [30] Chi-Wang Shu. Survey on discontinuous Galerkin methods for Hamilton-Jacobi equations. In Jichun Li, Hongtao Yang, and E. Machorro, editors, *Recent Advances in Scientific Computing and Applications*, volume 586 of *Contemp. Math.*, pages 323–330. American Mathematical Society (AMS), Providence, RI, USA, 2013.
- [31] P. Soravia. Nonlinear  $\mathcal{H}_\infty$  control. In M. Bardi and I. Capuzzo-Dolcetta, editors, *Optimal Control and Viscosity Solutions of Hamilton-Jacobi-Bellman Equations*, Systems & Control: Foundations & Applications, pages 505–531 in Appendix B. Birkhäuser Boston Inc., Boston, MA, 1997.
- [32] I. Xausa. *Verification of Collision Avoidance Systems using Optimal Control and Sensitivity Analysis*. PhD thesis, Fakultät für Luft- und Raumfahrttechnik, Universität der Bundeswehr München, Munich, Germany, December 2015.
- [33] I. Xausa, R. Baier, M. Gerdts, M. Gonter, and C. Wegwerth. Avoidance trajectories for driver assistance systems via solvers for optimal control problems. In *Proceedings on the 20th International Symposium on Mathematical Theory of Networks and Systems (MTNS 2012), July 9–13, 2012, Melbourne, Australia*, page 8 p., Melbourne, Australia, 2012. Think Business Events. Paper No. 294, full paper.

INSTITUT FUER MATHEMATIK UND RECHNERANWENDUNG, FAKULTAET FUER LUFT- UND RAUMFAHRTTECHNIK, UNIVERSITAET DER BUNDESWEHR MUENCHEN, D-85577 NEUBIBERG/MUENCHEN, GERMANY. EMAIL: ILARIAXAUSA@GMAIL.COM

CHAIR OF APPLIED MATHEMATICS, UNIVERSITY OF BAYREUTH, 95440 BAYREUTH, GERMANY. EMAIL: ROBERT.BAIER@UNI-BAYREUTH.DE

UNIVERSITÉ DE PARIS, LABORATOIRE JACQUES-LOUIS LIONS (LJLL), F-75005 PARIS, FRANCE, & SORBONNE UNIVERSITÉ, LJLL, F-75005 PARIS, FRANCE EMAIL: OLIVIER.BOKANOWSKI@MATH.UNIV-PARIS-DIDEROT.FR

INSTITUTE OF APPLIED MATHEMATICS AND SCIENTIFIC COMPUTING, DEPARTMENT OF AEROSPACE ENGINEERING, BUNDESWEHR UNIVERSITY MUNICH, WERNER-HEISENBERG-WEG 39, 85577 NEUBIBERG, GERMANY. EMAIL: MATTHIAS.GERDTS@UNIBW.DE