Obstacle Avoidance for Low Speed Autonomous Vehicles

Yuxiao Chen [*a], Huei Peng [a], Jessy Grizzle [b]
[*] Corresponding author. Phone +7347801768
[a] Dept. of Mechanical Engineering, University of Michigan, {chenyx, hpeng}@umich.edu
[b] Dept. of Mechanical Engineering, University of Michigan, grizzle@eecs.umich.edu

Abstract

This paper presents an obstacle avoidance algorithm for low speed autonomous vehicles (AV), with guaranteed safety. A supervisory control algorithm is constructed based on a barrier function method, which works in a plug-and-play fashion with any lower-level navigation algorithm. When the risk of collision is low, the barrier function is not active; when the risk is high, based on the distance to an "avoidable set", the barrier function controller will intervene, using a mixed integer program to ensure safety with minimal control effort. This method is applied to solve the navigation and pedestrian avoidance problem of a low speed AV. Its performance is compared with two benchmark algorithms: a potential field method and the Hamilton-Jacobi method.

Key words: obstacle avoidance; barrier function; autonomous vehicles

1. Introduction

Automotive companies are actively pursuing autonomous vehicles (AVs) to realize their potential in improved safety and mobility. Some of the efforts targets high speed applications, while others focus on low speed applications, such as airport transport, driverless pods on urban streets, museum tours, etc. In 2013, Hitachi announced their development of a single-passenger mobility-support robot "ROPITS", which uses stereo cameras and multiple laser radar sensors to navigate(IIMURA and YAMAMOTO 2014). In Britain, LUTZ Pathfinder has tested a driverless 'pod' vehicle. (Transport Systems Catapult 2015) Indoor autonomous robots have also been tested, including museum guiding robots Minerva (Thrun et al. 1999) and KAPROS (Yamazaki et al. 2009).

The main contribution of this paper is the following: (1) we propose the polar algorithm, a novel method that automatically computes the avoidable set given the dynamics which provides safety guarantee to the vehicle; (2) we propose a mixed integer programming (MIP) based implementation of control barrier function that impose the safety requirement above the navigation of the vehicle as a supervisory control scheme.

Low speed autonomous vehicles differ from high speed ones in two ways: (1) there are no lane boundaries; (2) they share space with multiple pedestrians and stationary obstacles. A basic problem is to navigate the AV from an initial point to a target point within a reasonable time, while avoiding collision with obstacles. Many methods have been proposed for obstacle avoidance of high speed autonomous vehicles, such as Model Predictive Control (Gray et al. 2012)(Yoon et al. 2009), Fuzzy logic (Fernandez Llorca et al. 2011), and motion primitive method (Frazzoli, Dahleh, and Feron 2002), but they usually cannot guarantee safety. This is partly because of the high operating speed of the vehicles.

On the other hand, when the operating speed is low, the problem of robot navigation with static or moving obstacle has been studied extensively. Cell decomposition was used in Minerva (Thrun et al. 1999) and tested with real tourists interacting with the robot. Bis et al. extended the cell decomposition concept to deal with moving obstacle with known speeds. The potential field method, originally developed for stationary obstacles, was also extended to moving obstacles (Khatib 1985)(Shimoda, Kuroda, and Iagnemma 2005). However, none of these methods provide safety guarantee. Van den Berg et al proposed reciprocal collision avoidance which provides collision avoidance guarantee under the assumption that the vehicle speed can be controlled instantaneously; however, this assumption is not usually valid in the real world. Usually motor torque is the control variable, i.e., acceleration can be directly controlled, but not the speed. Dynamic window Approach (DWA) was proposed in (Fox, Burgard, and Thrun 1997) which guarantees that the vehicle will not collide with static obstacles. The DWA idea was further developed in (Ogren and Leonard 2005) and in (Mitsch, Ghorbal, and Platzer 2013) for moving obstacles, with heavy use of braking. The method we propose can guarantee safety of low speed autonomous vehicles by steering and braking, and steering is preferred if it is not necessary to brake.

The safety assurance is rooted in the concept of reachable sets. The key challenge of this control concept is the computation of the reachable set based on the system dynamics. Multiple methods of computing the reachable set have been proposed in the literature. Prajna et al. proposed a barrier certificate concept that computes an outer approximation of the reachable set via sum of square programming (Prajna et al. 2004). Mitchell et al. proposed a Time-Dependent Hamilton-Jacobi method, which is applicable to two player differential games with given final time (Mitchell, Bayen, and Tomlin 2005). Henrion et al. proposed the occupation measure method to calculate an outer approximation of the reachable set via Moment Programming for polynomial dynamic systems (Henrion

and Korda 2014)(Majumdar et al. 2014). The polar algorithm computes a polytopic avoidable set whose compliment is an outer approximation of the reachable set, and controlled invariant. The conservativeness serves as a safety buffer for the reachable set. Then a control barrier function is constructed based on the avoidable set, and implemented as supervisory controller using MIP.

In the remainder of this paper, Section 2 introduces the dynamic model and defines the problem to be solved; Section 3 presents the supervisory control structure. Section 4 introduces the polar algorithm; Section 5 discusses the implementation of polar algorithm on low speed autonomous vehicles, and the simulation results are shown in Section 6. Finally, conclusions are presented in Section 7.

2. Dynamic Models and Problem Formulation

2.1. Dynamic models

Two dynamic models are used in this study. The first model describes vehicle motion in the Earth-fixed coordinates; the second model records the relative motion between the vehicle and a moving obstacle. When there are multiple obstacles, we will create a copy of the second model for each obstacle. For each type of obstacle, a maximum velocity and a geometric size is defined.

A unicycle model is used to represent the dynamics of the autonomous vehicle:

$$\begin{cases} \dot{X} = v\cos\psi & \dot{v} = a \\ \dot{Y} = v\sin\psi & \dot{\psi} = r \end{cases}, \tag{1}$$

where X and Y are the global Cartesian coordinates, v denotes the vehicle velocity and ψ is the heading angle. The inputs to the vehicle are acceleration a and yaw rate r.

$$u = (a, r). \tag{2}$$

The following constraints are used in this paper:

$$\begin{aligned} v &\in [0, v_{\max}] & r &\in [-r_{\max}, r_{\max}] \\ a &\in [-a_{\max}, a_{\max}] & a^2 + v^2 r^2 &\le \mu^2 g^2 \end{aligned}, \tag{3}$$

where μ is the fiction coefficient and g is the gravitational constant.

The motion relative to a moving obstacle is described by the following model:

$$\begin{bmatrix} \Delta\dot{X} \\ \Delta\dot{Y} \\ \dot{v} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} v_{dx} - v\cos\left(\theta + \tan^{-1}\left(\dfrac{\Delta Y}{\Delta X}\right)\right) \\ v_{dy} - v\sin\left(\theta + \tan^{-1}\left(\dfrac{\Delta Y}{\Delta X}\right)\right) \\ a \\ r + \dfrac{\sin\theta}{\Delta X^2 + \Delta Y^2} - \dfrac{\Delta Y v_{dx} - \Delta X v_{dy}}{\Delta X^2 + \Delta Y^2} \end{bmatrix}, \tag{4}$$

where ΔX and ΔY denote the relative position of the obstacle with respect to the vehicle in the global coordinates:

$$\begin{cases} \Delta X = X_d - X \\ \Delta Y = Y_d - Y \end{cases}, \begin{cases} \dot{X}_d = v_{dx} \\ \dot{Y}_d = v_{dy} \end{cases}, \tag{5}$$

X_d and Y_d are the coordinates of the obstacle. v_d, velocity of the obstacle, is a disturbance input and v_{dx} and v_{dy} are its components in X and Y directions. θ is the difference between the AV heading angle and the yaw angle of the obstacle.

$$\theta = \psi - \tan^{-1}\left(\frac{\Delta Y}{\Delta X}\right) \tag{6}$$

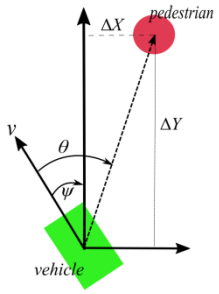

Fig. 1. Coordinate system of the relative dynamic model

Both θ and ψ are restricted between $-\pi$ and π. In the remainder of this paper, we will denote the dynamic system in Eq. (4) as Σ_0, and the set for state, input and disturbance are denoted as S_0, U_0 and D_0, respectively.

2.2. Problem Formulation

The goal of the AV is to reach a destination without colliding with any obstacle. Strictly speaking, under some circumstances, collision is inevitable, for example, when the AV is surrounded by hostile pursuers. Macek et al. proposed the concept of 'passive safety' and 'passive friendly safety' in (Macek et al. 2009). In this paper, we adopt the idea of 'passive friendly safety' and extend it to multiple moving obstacles. We assert the following rules:

(1) When the AV is stopped, any conflict is not considered a collision caused by the AV.

(2) When a obstacle runs into the AV from behind, it is not considered a collision caused by the AV.

The definition of collision from behind is

$$|\theta| \ge \frac{\pi}{2} \tag{7}$$

The second rule is not needed for single obstacle case, since the vehicle can accelerate and avoid the collision. But in the case of multiple moving obstacles, when an AV is threatened by an approaching obstacle from behind, neither accelerating (risk to others in front) nor slowing down (escalating the situation) is safe. These two rules are applied for all simulations in this paper.

For simplicity, it is assumed that all obstacles are pedestrians, and both the vehicle and pedestrians are assumed to have a round shape with radius R_v and R_p,

respectively. In simulations, collision is detected by the following rule:

$$\left(\sqrt{\Delta X^2+\Delta Y^2}\le R_v+R_p\right)\wedge\left(v>0\right)\wedge\left(\left|\theta\right|\le\frac{\pi}{2}\right) \quad (8)$$

In this study, the on-board sensors are assumed to measure all states accurately. The speed of the pedestrians is assumed to be bounded:

$$\left\|v_d\right\|=\sqrt{{v_{dx}}^2+{v_{dy}}^2}\le v_{d\max} \quad (9)$$

3. Supervisory Control and Avoidable Set

3.1. Supervisory Control

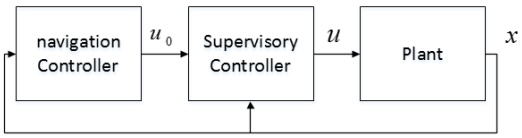

Fig. 2. Block diagram of supervisory control

The overall control system consists of two parts: the navigation controller and the supervisory controller. The navigation controller guides the AV to the destination, and its design is not the focus of this paper. A model predictive control algorithm will be used in the simulation part. The goal of the supervisory controller (the focus of this paper) is safety and safety only. It modifies the input of the navigation controller with minimum interference to guarantee safety. This means the supervisory controller is inactive if there is no threat, and action is taken only to keep the vehicle safe. This concept is realized through the following formulation

$$minimize\ \left\|u-u_0\right\|_Q, s.t.\ x\notin P_B, \quad (10)$$

where P_B is the "avoidable set" to be defined later in this paper. Q is the matrix defining the norm:

$$\left\|u-u_0\right\|_Q^{\ 2}=\left(u-u_0\right)^T Q\left(u-u_0\right), \quad (11)$$

which is tuned for desired performance.

3.2. Definition of avoidable set

The goal of the supervisory control is to avoid collision under all possible disturbance input, i.e., Eq. (8) is not violated by any obstacle at any time. We first define three sets.

(1) **Infeasible set**: As shown in Fig. 3, the infeasible set is the region where collision is not always preventable, i.e., there exists a disturbance under which a collision will happen for any control input. However note that if the pedestrian is not hostile or not shrewd enough to choose this disturbance value, collision may not occur. Once the AV is inside the infeasible set, it should slow down and stop immediately. The infeasible set is denoted as X_{In} in the remainder of the paper.

(2) **Avoidable set**: To guarantee that the AV never enters the infeasible set, we will construct a superset, which encloses the infeasible set, and its complement is controlled invariant. For a state in the avoidable set and outside of the infeasible set, collision can be avoided by taking emergency action, to be explained later. If possible, the state should be driven out away from the avoidable set.

The controlled invariant set: The complement of the avoidable set. Because it is control invariant, any state outside of the avoidable set can stay outside under all possible disturbances.

Denote the system dynamics as

$$\dot{x}=f\left(x,u,d\right),\ x\in S,u\in U,d\in D \quad (12)$$

where x, u and d are the state, control input and the disturbance input, respectively. The mathematical conditions for a set P to be avoidable is then

$$\begin{gathered}\forall x(0)\notin P_B,\forall t>0,\forall s\in[0,t],\forall d(s)\in D,\\ \exists u(s)\in U,\ s.t.\ x(t)\notin P\end{gathered} \quad (13)$$

If set P_B is the zero level set of a real-valued function $b(x)$ (i.e., P_B is characterized as $b(x)<0$), then the set invariance condition becomes a boundary condition:

$$\begin{gathered}b(x)\ge 0,\forall x\notin P_B; b(x)<0,\forall x\in P_B\\ \forall x, b(x)=0\rightarrow\forall d\in D,\exists u\in U,\\ s.t.\ \ \dot{b}(x)=\left(\nabla_x b\right)^T f\left(x,u,d\right)\ge 0\end{gathered} \quad (14)$$

Suppose at a given point x_0 is on ∂P_B, i.e., the boundary of P_B, the normal vector exists. Denote the normal vector pointing outwards from P_B as $\vec{n}_{x_0}$, then the geometric condition of (14) is equivalent to

$$\forall d\in D,\exists u\in U\ s.t.\left\langle f\left(x_0,u,d\right),\vec{n}_{x_0}\right\rangle\ge 0 \quad (15)$$

where $\left\langle\bullet,\bullet\right\rangle$ denotes the inner product of two vectors.

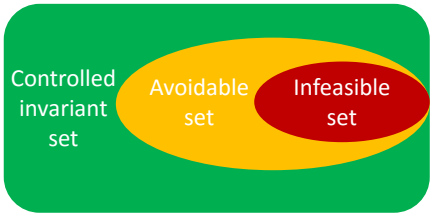

Fig. 3. Different stages of obstacle avoidance.

4. Polar Algorithm

The key challenge of the supervisory control is to find an avoidable set that contains the infeasible set. To solve this problem, a polar algorithm is proposed. The infeasible set is represented by a bounded polytope containing the collision set (Eq. (8)), then the polar algorithm solves for another polytope that contains the infeasible set and satisfies the boundary condition (Eq. (13)). The polar algorithm is applicable to the dynamics in the following form

$$\dot{x}=Eu+Gd, x\in S,u\in U,d\in D, \quad (16)$$

where S, U and D are polytope; E and G are constant matrices. Note that there is no state dependent term in the

state derivative. The dynamic model in Section 2 is simplified to this form by viewing all state dependent terms as disturbance, which is explained in detail in Section 5.

4.1. Polar of a polytope

In this section, we introduce the polar algorithm, which is built based on the dual property of polytope. A polytope P has two important elements: vertices and facets. For simplicity, only bounded polytopes with a finite number of vertices and bounding hyperplanes are considered. Given a vector space $X=\mathbb{R}^n$, a bounded polytope P with the origin in its interior can be represented as set of the convex combination of its vertices:

$$P=\left\{x \mid x=\sum_i \lambda_i v_i, \sum_i \lambda_i = 1, \lambda_i \geq 0, \forall i\right\}, \tag{17}$$

where $\{v_i\}=P.V$ is the set of vertices of P. It can also be represented as an intersection of finitely many closed half spaces:

$$P=\bigcap_{H_i}\left\{x \mid {H_i}^T x \leq 1\right\}, \tag{18}$$

where $\left\{x \mid {H_i}^T x \leq 1\right\}$ are the bounding half spaces. We further denote $\{H_i\} \subset X^{\#}$ as the set of linear functionals that corresponds to the half spaces:

$$H_i(x)={H_i}^T x. \tag{19}$$

$X^{\#}=\{f: \mathrm{X} \rightarrow \mathbb{R} \mid f \text{ linear}\}$ is the algebraic dual of the vector space X. Since the algebraic dual of a Euclidean space is also an Euclidean space, we treat the linear functionals in $X^{\#}$ as vectors in the dual space and denote the set $\{H_i\}$ as $P.H$. In addition, half spaces and polytopes can also be defined in $X^{\#}$. The bounding hyperplane corresponding to a bounding half space $\left\{x \mid H^T x \leq 1\right\}$ is defined as

$$\left\{x \mid {H_i}^T x = 1\right\}. \tag{20}$$

And the corresponding facet is defined as

$$F(H_i)=P \cap \left\{x \mid {H_i}^T x = 1\right\}. \tag{21}$$

For polytopes, the normal vector of facet $F(H)$ is simply H, which simplifies the condition in Eq. (15).

The polar of P is a polytope in the dual space $X^{\#}$ defined as

$$P^{\Delta}=\left\{H \mid H^T x \leq 1, \forall x \in P\right\}. \tag{22}$$

The vertices of the original polytope are mapped to the facets of the polar, and it is easy to check that the facets of the original polytope are mapped to the vertices of the polar.

$$P^{\Delta}.H=P.V \qquad P^{\Delta}.V=P.H \tag{23}$$

Polar provides a convenient way of imposing the inclusion of polytopes:

$$P_1 \subseteq P_2 \leftrightarrow P_2.H \subseteq P_1^{\Delta} \tag{24}$$

And we will use this property in the polar algorithm.

We included more detailed properties of polar in Section A.1 in the appendix. For more information, see (Ziegler 1995).

In our problem, i.e., finding an avoidable set for the infeasible set X_{In}, the set inclusion condition is enforced by the following constraint:

$$P_B.H \in X_{In}^{\Delta}, \tag{25}$$

4.2. Hyperplane Orientation and boundary condition

The boundary condition of the avoidable set is interpreted as an orientation condition for the bounding hyperplanes. For each facet of a polytope P, suppose that the corresponding bounding hyperplane is $H^T x=1$, the normal vector that points outwards from P is simply H.

For the dynamic system in Eq. (16), the avoidable set boundary condition in Eq. (15) becomes

$$\begin{aligned} &\forall H \in P.H, \forall d \in D, \\ &\exists u \in U \ s.t. \langle Eu+Gd, H \rangle \geq 0 \end{aligned}. \tag{26}$$

In order to find all bounding hyperplane orientations that are valid for an avoidable set, first we fix input u. For a bounding hyperplane $H^T x=1$ to satisfy the boundary condition of the avoidable set with fixed input u, the following inequality must hold:

$$H^T(Eu+Gd) \geq 0, \forall d \in D. \tag{27}$$

Since D is a polytope and the system dynamics are linear, Eq. (27) is simplified to checking only its vertices:

$$H^T(Eu+Gd) \geq 0, \forall d \in D.V. \tag{28}$$

Eq. (28) defines a polytope in $X^{\#}$ and it is easy to check that ${P_{Hs}}^u=\left\{H \mid (Eu+Gd)^T H \geq 0, \forall d \in D.V\right\}$ contains all the functionals corresponding to the bounding hyperplanes valid under input u.

Define

$$P_{Hs}=\bigcup_{\forall u \in U.V} P_{Hs}^u \tag{29}$$

This union may not be convex, but using the linearity of the dynamics, we claim the following:

Claim 1. *P_{Hs} is the maximal set of linear functionals that satisfies the boundary condition of an avoidable set for the dynamic system shown in Eq. (16).*

The proof is shown in the A.1 in Appendix.

Recall that there are two requirements for the avoidable set: set inclusion and boundary conditions. The set inclusion requirement is simplified to picking bounding hyperplanes from the polar of the infeasible set X_{In}; the boundary conditions requirement is simplified to picking bounding hyperplanes from P_{Hs}. Therefore it is natural to intersect the two sets. Define

$$P_H = P_{Hs} \cap X_{In}^{\Delta}. \tag{30}$$

This set may not be convex since it is the intersection of a convex polytope and a nonconvex union of polytopes. Further define

$$P_B = Conv(P_H)^{\Delta}, \tag{31}$$

where $Conv(\cdot)$ denotes the convex hull of a polytope. Then P_B possess the following properties:

Claim 2. *P_B is an avoidable set that contains X_{In}.*

Claim 3. *P_B is the minimal avoidable set.*

Claim 4. *If the origin is in the interior of X_{In} and $\overline{X_{In}} = X_{In} + c$, where c is a constant shifting vector, then $\overline{P_B} = P_B + c$ where $\overline{P_B}$ is the constructed avoidable set based on $\overline{X_{In}}$.*

The above claims mean that P_B is the minimal avoidable set that satisfies both set inclusion condition and boundary condition, and it is invariant with respect to the change of origin position.

The proofs of all claims are shown in A.3, A.4 and A.5 in the Appendix.

5. Avoidable Set for Low Speed Autonomous Vehicles

In this section, the avoidable set for a single moving obstacle is solved using the polar algorithm.

5.1. Infeasible set

The obstacle avoidance problem with a single moving obstacle can be formulated as a pursuit-evasion problem. In general, the vehicle can use both steering and braking to avoid collision (and both are used in the control implementation). In the computation of the infeasible set, however, we only use braking. This allows easy extension to multiple pedestrians cases.

For a given initial condition, an obstacle's future position is bounded by:

$$\left(X_d(t) - X_d(0)\right)^2 + \left(Y_d(t) - Y_d(0)\right)^2 \le \left(v_{d\max} t\right)^2 \tag{32}$$

Then the vehicle's position is calculated by applying the maximum brake. By checking whether a collision happens before the vehicle stops, the points in the state space can be then identified as either feasible (safety is guaranteed) or infeasible (safety cannot be guaranteed). We select a grid with certain resolution in the state space and compute whether collision happens or not for each grid point. A polytope X_{In} is then found that contains all the infeasible grid points by computing their convex hull.

5.2. Avoidable set for autonomous vehicle

Consider the relative dynamics shown in Eq. (4). It is rewritten as:

$$\begin{bmatrix} \Delta\dot{X} \\ \Delta\dot{Y} \\ \dot{v} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \bar{d}_1 \\ \bar{d}_2 \\ a \\ r + \dfrac{\sin\theta}{\Delta X^2 + \Delta Y^2} + \bar{d}_3 \end{bmatrix} \tag{33}$$

where

$$\begin{aligned} \bar{d}_1 &= v_{dx} - v\cos\left(\theta + \tan^{-1}\left(\Delta Y / \Delta X\right)\right) \\ \bar{d}_2 &= v_{dy} - v\sin\left(\theta + \tan^{-1}\left(\Delta Y / \Delta X\right)\right) \\ \bar{d}_3 &= \frac{\Delta X v_{dy} - \Delta Y v_{dx}}{\Delta X^2 + \Delta Y^2} \end{aligned} \tag{34}$$

It is then easy to find the bounds of the disturbance:

$$\begin{aligned} \bar{d}_1^{\,2} + \bar{d}_2^{\,2} &\le \left(v_{d\max} + v_{\max}\right)^2 + v_{\max}^{\,2} \\ \left|\bar{d}_3\right| &\le \frac{v_{d\max}}{\sqrt{\Delta X^2 + \Delta Y^2}} \end{aligned} \tag{35}$$

From the collision avoidance condition,

$$\left(\sqrt{\Delta X^2 + \Delta Y^2} \ge R_v + R_p\right) \Rightarrow \left(\left|\bar{d}_3\right| \le \frac{v_{d\max}}{R_v + R_p}\right) \tag{36}$$

Therefore, the dynamic is simplified as

$$\begin{aligned} \dot{x} &= Eu + G\bar{d} + k \\ k &= \begin{bmatrix} 0 & 0 & 0 & \dfrac{\sin\theta}{\Delta X^2 + \Delta Y^2} \end{bmatrix}^{\cdot} \end{aligned} \tag{37}$$

where

$$\begin{aligned} \bar{d}_1^{\,2} + \bar{d}_2^{\,2} &\le \left(v_{d\max} + v_{\max}\right)^2 + v_{\max}^{\,2} \\ \left|\bar{d}_3\right| &\le \frac{v_{d\max}}{R_v + R_p} \end{aligned}^{\cdot} \tag{38}$$

Although the constraint for $\bar{d}_1$ and $\bar{d}_2$ is a circle and nonlinear, it can be outer approximated by a polygon. As the number of vertices grows, the approximation becomes more accurate. Therefore, the disturbance $\bar{d} = \begin{bmatrix} \bar{d}_1 & \bar{d}_2 & \bar{d}_3 \end{bmatrix}^T$ is bounded by a polytope.

The input constraints are

$$\begin{aligned} a &\in \left[-a_{\max}, a_{\max}\right] \\ r &\in \left[-r_{\max}, r_{\max}\right] \\ a^2 + v_{\max}^{\,2} r^2 &\le \left(\mu g\right)^2 \end{aligned}, \tag{39}$$

where μ is the tire-road friction coefficient, and g is the gravitational constant. A polygon is again used to approximate the circular constraint, so that the set to constrain U is also approximated by a polytope.

Denote the dynamic system in Eq.(37) by $\bar{\Sigma}$, and the set of disturbance is denoted by $\bar{D}$.

Claim 5. *If a set P is an avoidable set for system $\bar{\Sigma}$, then it is an avoidable set for Σ_0 in Eq. (4).*

The proof is shown in A.6 in the Appendix.

The last term k in Eq. (37) provides additional safety margin. When $\theta > 0$, the obstacle is on the right hand side of the vehicle. To avoid collision, θ must increase. Similarly, when $\theta < 0$, θ must decrease for safety. Both of these conditions are helped by k.

Denote the dynamic system that ignores the last term k as $\tilde{\Sigma}: \dot{x} = E_1 u + G_1 \bar{d}$

Claim 6. *Suppose a polytope P is an avoidable set for $\tilde{\Sigma}$. Additionally, for any point x on the boundary of P where $x_4 = \theta > 0$, the bounding hyperplane $H^T x = \begin{bmatrix} H_1 & H_2 & H_3 & H_4 \end{bmatrix} x = 1$ at that point satisfies $H_4 \geq 0$, and for any point where $x_4 = \theta < 0$, $H_4 \leq 0$ holds, then P is an avoidable set for $\bar{\Sigma}$.*

The proof is shown in A.7 in the Appendix.

With Claim 6, the problem is simplified to finding an avoidable set for $\tilde{\Sigma}$, and check whether the condition in Claim 6 holds.

By applying the procedure described in Section 4, the avoidable set P_B is obtained and shown in Fig. 4. Because the dimension of the state space is 4, P_B is projected to three 3-D plots. The parameters used in generating the avoidable set are listed in TABLE III.

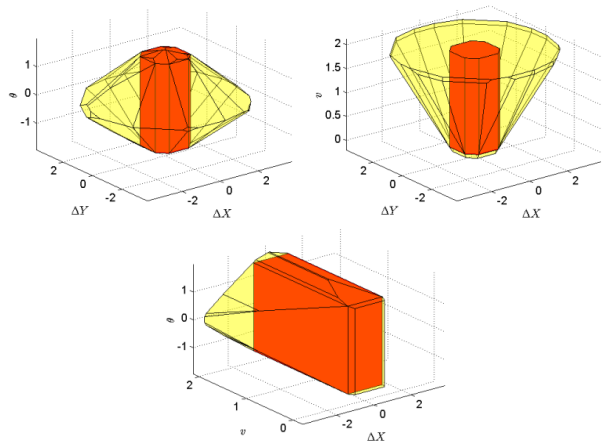

Fig. 4. The avoidable set (yellow) and the infeasible set (red)

5.3. Control Barrier Function

Aaron et al. proposed the concept of control barrier function (CBF) in (Ames, Grizzle, and Tabuada 2014), which guarantees that the state does not enter a given set, and it works with a control Lyapunov function through a quadratic program. The control barrier function idea forms the basis of the supervisory control of this paper. First, a control barrier function is applied to a single half-space generated from a facet of the avoidable set, a Mixed Integer Program (MIP) is used to deal with multiple facets.

Let's focus on one of the half-space of the polytope of the avoidable set. If the state must stay outside of the half-space $\{x \mid H^T x \leq 1\}$ to ensure safety, the control barrier function is defined as

$$b(x) = H^T x - 1, \quad B(x) = -\log \frac{b(x)}{1 + b(x)}. \tag{40}$$

It can be seen that $B(x)$ goes to infinity as x approaches the bounding hyperplane. When x is far from the boundary, $B(x)$ is positive but small. If the derivative of $B(x)$ is always finite, then $B(x)$ is finite, and x remains outside of the half-space. In particular, the following constraint is enforced by the supervisory controller:

$$\dot{B}(x) \leq c_1 / B(x), \tag{41}$$

where c_1 is a positive constant. This constraint is loose when x is far away from the boundary, and becomes tighter as x approaches the half-space. Consider the influence of sampling, the condition is modified to

$$H^T f(x,u,d) \geq -\frac{c_1 b}{B(x) + c_1 T_s} \tag{42}$$

where T_s is the sampling time. The derivation of (42) is shown in A.8 in the Appendix.

5.4. Mixed Integer Program

For each bounding hyperplane H of P_B, if the state satisfies $H^T x \leq 1$, then H is said to be active, otherwise, it is said to be inactive. Since avoidable set P_B is the intersection of all bounding half-spaces, the state x is outside P_B if and only if there exists at least one active bounding hyperplane, i.e.,

$$x \notin P_B \Leftrightarrow \exists H_0 \in P_B.H, H_0{}^T x > 1 \tag{43}$$

When $x(t)$ is outside the avoidable set, each active bounding hyperplane generates a linear constraint, in the form of Eq. (42), and we require at least one constraint is satisfied. Since this is a logic disjunction, combining the supervisory control structure in Eq. (10), we form an MIP with slack variables:

$$minimize \left\| u - u_0 \right\|_Q, \; s.t.$$

$$H_j{}^T f(x,u,d) \geq \frac{-c_1 b_{H_j}}{B_{H_j}(x) + c_1 T_s} - s_j c_2, \forall H_j \in H_a, \tag{44}$$

$$\sum_{j=1}^{N_P} s_j \leq N_P - 1, \; s_j \in \{0,1\}$$

where $\{s_j\}$ are slack variables, and c_2 is a large positive number. When $s_j = 1$, the corresponding inequality is automatically satisfied; when $s_j = 0$, the original barrier inequality is enforced. H_a is the set of all active bounding hyperplanes and N_P is its cardinality. The condition $\sum_{j=1}^{N_P} s_j \leq N_P - 1$ ensures that at least one of the original barrier inequalities is satisfied.

5.5. From single obstacle to multiple obstacles

When there are multiple moving obstacles, one can follow the same approach to calculate the avoidable set by expanding the state space to

$$x = \begin{bmatrix} \Delta X_1 & \Delta Y_1 & \theta_1 & \dots & \Delta X_n & \Delta Y_n & \theta_n & v \end{bmatrix}^T. \tag{45}$$

Because the computation complexity grows exponentially with the dimension of the state space, this naive approach is not scalable to a large number of

pedestrians. Therefore, we need a simpler way. The key innovation is that the AV is only allowed to brake when calculating the infeasible set, as mentioned in Section 5.1. With this assumption, the infeasible set can be computed for one obstacle, and then applied to multiple obstacles since the emergency action to avoid all obstacles are the same. If steering is allowed, this will no longer be the case.

Recall the concept of responsibility in Section 2.2, let X_{In} be the set outside of which the vehicle can come to a full stop before hitting the obstacle for all obstacle movement. In the multiple-obstacle case, as long as all obstacles are outside of X_{In}, the vehicle is always able to stop before hitting any one of the obstacles.

For each pedestrian, there is a set of state as described in Eq. (4) representing the relative dynamics between the pedestrian and the AV. Now the task is to keep multiple states outside of a single avoidable set.

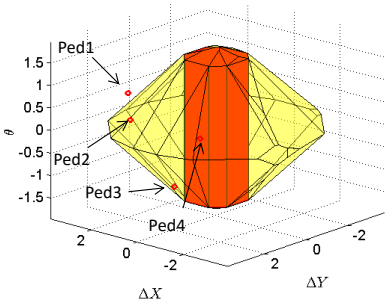

Fig. 5. Multiple-pedestrian cases with a single infeasible set (red) and avoidable set (yellow)

When there are multiple obstacles, the avoidable set is not always avoidable. Nevertheless, the supervisory control developed for a single pedestrian can still be used with the following modifications: (i) the constraint in Eq.(44) must be checked for each of the pedestrians; and (ii) when any obstacle breaches the "avoidable set," braking is applied until either the states for all pedestrians are outside the avoidable set, or the AV comes to a complete stop. Although braking is used to guarantee safety in multiple pedestrian case, the simulation shows that the supervisory control does not rely on braking heavily. The vehicle only comes to full stop when it's trapped by multiple pedestrians.

6. Simulation Results

6.1. Simulation results

The goal of the AV is to reach a destination from a fixed starting point without colliding with any pedestrian. The initial positions and velocities of the pedestrians are random, and they walk randomly. All objects stay in a predefined rectangular region $[-X_{\lim}, X_{\lim}]\times[-Y_{\lim}, Y_{\lim}]$. The random-walk is generated with Gaussian distributed acceleration in X and Y directions:

$$\begin{aligned} \dot{v}_{px} &= a_{px} \qquad \dot{v}_{py} = a_{py} \\ a_{px} &\sim N(0,\sigma_{ax}) \quad a_{py} \sim N(0,\sigma_{ay}) \end{aligned}. \tag{46}$$

The velocity must also satisfy the boundedness constraint in Eq. (9). To keep the pedestrians inside the rectangular region, the following (reflection) rule is used

$$\begin{aligned} v_{px} &= -|v_{px}|, \text{ if } X_p \geq X_{\lim}; \\ v_{px} &= |v_{px}|, \text{ if } X_p \leq -X_{\lim}; \\ v_{py} &= -|v_{py}|, \text{ if } Y_p \geq Y_{\lim}; \\ v_{py} &= |v_{py}|, \text{ if } Y_p \leq -Y_{\lim}. \end{aligned} \tag{47}$$

A greedy Model Predictive Controller is used as the navigation controller. The detail of MPC is shown in A.10 in the Appendix. The control structure is shown below:

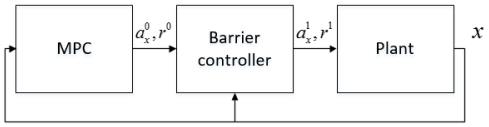

Fig. 6 Control structure for simulation

The simulation parameters are listed in TABLE I.

TABLE I SETTINGS OF THE SIMULATION RUNS

Number of pedestrians	7
Region of pedestrian movement	$[-5,5]\times[-5,5]$
Initial Position of the vehicle	$(1,-7)$
Initial velocity of the vehicle	2m/s
Initial yaw angle	$\pi/2$
Destination	(0,5)

A sample simulation is illustrated in Fig. 7. The blue circles are snapshots of the position of the vehicle; the green circles show the positions of the pedestrians; and the red square is the destination. The color of the snapshots changes from light to thick as time flows.

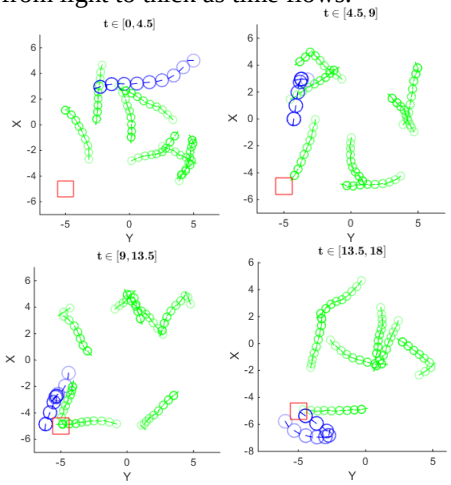

Fig. 7. Sample simulation results

The control inputs from the navigation controller and the supervisory controller are shown in the first 2 subplots in Fig. 8, where a_x^0 and r^0 are the acceleration and yaw rate command from the navigation controller; a_x^1 and r^1 are the command from the supervisory controller. $d_{\max}$ denotes the distance from the state to the avoidable set, and was plotted

in the third subplot. Different colors correspond to $d_{\max}$ for different pedestrians.

$$d_{\max} = \max_{H \in P.H} \frac{H^T x - 1}{\|H\|} \tag{48}$$

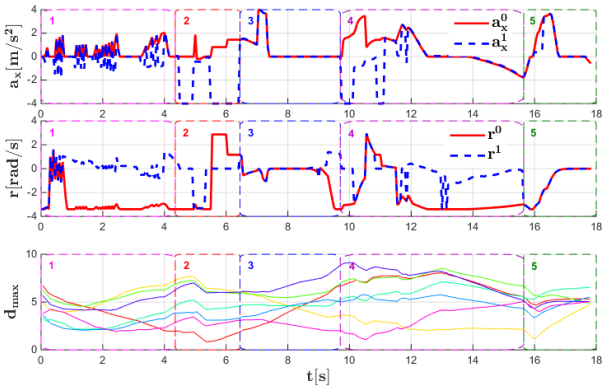

Fig. 8 Control input and minimum distance to avoidable set

Based on the difference between MPC input and Barrier input, the simulation was divided into 5 stages. In stage 3 and 5, the supervisory controller detects little danger, so the two control inputs stay close; in stage 1 and 4, the supervisory controller manage to follow similar acceleration command, but use different yaw rate to avoid collision. This is because we put more weight on acceleration difference than yaw rate difference when defining Q matrix in Eq. (11). In stage 2, both acceleration and steering are changed by the supervisory control to ensure safety. Compare the input plot and $d_{\max}$ plot, the time when supervisory control changes the MPC input corresponds to the time when the smallest $d_{\max}$ among 7 pedestrians is small, which indicates danger.

6.2. Comparison to two benchmark methods

The performance of the polar algorithm is compared to two benchmark methods: the potential field method (Shimoda, Kuroda, and Iagnemma 2005) and the Hamilton Jacobi method (Mitchell, Bayen, and Tomlin 2005). The details of both benchmark methods are included in the Appendix.

In order to compare the performance, we repeat the simulation 1000 times with the same setting shown in TABLE I. In each trial, the vehicle starts at the same location and tries to reach the same destination, while the pedestrians appear at random position and does random walk. A simulation trial is marked as "stuck" if the AV failed to reach the destination within 25 seconds; and a trial is marked as "crash" if Eq.(8) is satisfied at any time.

TABLE II KEY PERFORMANCE INDICES OF THE THREE METHODS IN 1000 SIMULATION TRIALS

Method	Average time	Collision	Stuck trips
Polar Method	10.88s	0	25
Hamilton Jacobi	14.93s	0	171
Potential Field	8.47s	436	0

The statistics of the 1000 trials are shown in TABLE II. Both the polar method and the Hamilton Jacobi method can ensure safety, i.e. no crash, while the potential field method crashes in about half of the trials. The Polar method reaches the destination in a much shorter time, and with fewer "stuck" cases compared with the Hamilton Jacobi method, indicating that the proposed method is as safe as, but much less conservative than the Hamilton Jacobi method.

7. Conclusion and Discussion

This paper proposed a polar algorithm to design collision avoidance algorithms for low speed autonomous vehicles. The concept is based on the construction of an "avoidable set," which is an extension of the commonly used concept of a controlled invariant set. A Mixed Integer Programming based supervisory control structure is proposed to implement this algorithm. Safety can be guaranteed for both single moving obstacle and multiple moving obstacles while liveness is maintained. The safety guarantee was verified with simulations.

Acknowledgment

The work is supported by NSF Contract #CNS-1239037. It is part of the Cyber-Physical System project participated in by the University of Michigan; University of California, Los Angeles; Georgia Institute of Technology; and Carnegie Mellon University.

References

Ames, Aaron D., Jessy W. Grizzle, and Paulo Tabuada. 2014. "Control Barrier Function Based Quadratic Programs with Application to Adaptive Cruise Control." In *53rd IEEE Conference on Decision and Control*, 6271–78. IEEE.

Evans, L C, and P E Souganidis. 1983. "Differential Games and Representation Formulas for Solutions of Hamilton-Jacobi-Isaacs Equations.," March.

Fernandez Llorca, David, Vicente Milanes, Ignacio Parra Alonso, Miguel Gavilan, Iván Garcia Daza, Joshué Perez, and Miguel Ángel Sotelo. 2011. "Autonomous Pedestrian Collision Avoidance Using a Fuzzy Steering Controller." *IEEE Transactions on Intelligent Transportation Systems* 12 (2): 390–401.

Fox, Dieter, Wolfram Burgard, and Sebastian Thrun. 1997. "The Dynamic Window Approach to Collision Avoidance." *IEEE Robotics and Automation*

Magazine 4 (1): 23–33

Frazzoli, Emilio, Munther A. Dahleh, and Eric Feron. 2002. "Real-Time Motion Planning for Agile Autonomous Vehicles." *Journal of Guidance, Control, and Dynamics*

Gray, Andrew, Yiqi Gao, Theresa Lin, J Karl Hedrick, H Eric Tseng, and Francesco Borrelli. 2012. "Predictive Control for Agile Semi-Autonomous Ground Vehicles Using Motion Primitives." In *American Control Conference (ACC), 2012*, 4239–44.

Henrion, Didier, and Milan Korda. 2014. "Convex Computation of the Region of Attraction of Polynomial Control Systems." *IEEE Transactions on Automatic Control* 59 (2): 297–312.

IIMURA, Taiki, and Kenjiro YAMAMOTO. 2014. "2A1-H03 Development of Single-Passenger Mobility-Support Robot"

Khatib, O. 1985. "Real-Time Obstacle Avoidance for Manipulators and Mobile Robots." *Proceedings. 1985 IEEE International Conference on Robotics and Automation*

Macek, Kristijan, Dizan Alejandro Vasquez Govea, Thierry Fraichard, and Roland Siegwart. 2009. "Towards Safe Vehicle Navigation in Dynamic Urban Scenarios."

Majumdar, A., R. Vasudevan, M. M. Tobenkin, and R. Tedrake. 2014. "Convex Optimization of Nonlinear Feedback Controllers via Occupation Measures." *The International Journal of Robotics Research* 33 (9): 1209–30.

Mitchell, I.M., A.M. Bayen, and C.J. Tomlin. 2005. "A Time-Dependent Hamilton-Jacobi Formulation of Reachable Sets for Continuous Dynamic Games." *IEEE Transactions on Automatic Control* 50 (7): 947–57.

Mitsch, Stefan, Khalil Ghorbal, and André Platzer. 2013. "On Provably Safe Obstacle Avoidance for Autonomous Robotic Ground Vehicles." *Roboticsproceedings.Org*.

Ogren, P., and N.E. Leonard. 2005. "A Convergent Dynamic Window Approach to Obstacle Avoidance." *IEEE Transactions on Robotics* 21 (2): 188–95.

Prajna, Stephen, Stephen Prajna, Ali Jadbabaie, and Ali Jadbabaie. 2004. "Safety Verification of Hybrid Systems Using Barrier Certificates." *Hybrid Systems: Computation and Control* 2993: 477–92.

Shimoda, Shingo, Yoji Kuroda, and Karl Iagnemma. 2005. "Potential Field Navigation of High Speed Unmanned Ground Vehicles on Uneven Terrain." *Proceedings - IEEE International Conference on Robotics and Automation* 2005 (April): 2828–33.

Thrun, S., M. Bennewitz, W. Burgard, A.B. Cremers, F. Dellaert, D. Fox, D. Hahnel, et al. 1999. "MINERVA: A Second-Generation Museum Tour-Guide Robot." *Proceedings 1999 IEEE International Conference on Robotics and Automation*

Transport Systems Catapult. 2015. "Autonomous Vehicles Hit the Streets in the UK's First 'Driverless Pod' Trial- The Inquirer."

Yamazaki, Keiichi, Akiko Yamazaki, Mai Okada, Yoshinori Kuno, Yoshinori Kobayashi, Yosuke Hoshi, Karola Pitsch, Paul Luff, Dirk Lehn, and Christian Heath. 2009. "Revealing Gauguin : Engaging Visitors in Robot Guide ' S Explanation in an Art Museum." In *CHI '09 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 1437–46.

Yoon, Yongsoon, Jongho Shin, H. Jin Kim, Yongwoon Park, and Shankar Sastry. 2009. "Model-Predictive Active Steering and Obstacle Avoidance for Autonomous Ground Vehicles." *Control Engineering Practice* 17 (7): 741–50.

Ziegler, Günter M. 1995. *Lectures on Polytopes*.

Appendix

A.1. Additional properties of Polar

The polar of a polytope $P \subset X$ is a polytope in the algebraic dual space $X^{\#}$ defined as

$$P^{\Delta} = \left\{ H \mid H^T x \le 1, \forall x \in P \right\}. \quad \text{(A.1)}$$

Because of the convexity and linearity of polytopes, a simpler definition is

$$P^{\Delta} = \left\{ H \mid H^T v_i \le 1, \forall v_i \in P.V \right\}. \quad \text{(A.2)}$$

The vertices of P are mapped to the facets of P^{Δ}, and the facets of the P are mapped to the vertices of P^{Δ}, as shown in the following example.

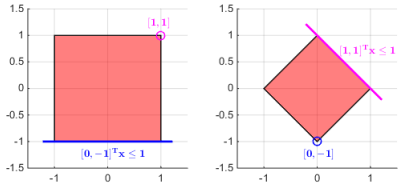

Fig. 9. Example of polar

For a bounded polytope P with the origin in its interior, the following hold:

(1) The polar of P, denoted as P^{Δ}, is a bounded polytope with the same dimension as P and containing the origin in the interior.

(2) The polar of P^{Δ}, denoted as $P^{\Delta\Delta}$, is the original polytope, i.e., $P^{\Delta\Delta} = P$.

(3) For any $H \in P^{\Delta}$, P is completely contained in the half space $\left\{ x \mid H^T x \le 1 \right\}$, i.e.,

$$\forall H \in P^{\Delta}, \forall x \in P, H^T x \le 1 \quad \text{(A.3)}$$

(4) For any point H outside P^{Δ}, P is not completely contained in the half-space $\left\{ x \mid H^T x \le 1 \right\}$

$$\forall H \notin P^{\Delta}, \exists x \in P, H^T x > 1 \quad \text{(A.4)}$$

Because of the properties above, the polar concept provides a clear condition for polytope inclusion:

$$P_1 \subseteq P_2 \leftrightarrow P_2.H \subseteq P_1^{\Delta} \tag{A.5}$$

A.2. Proof of Claim 1:

For any bounding half space $H^T x \le 1$ that satisfies Eq. (27), since U is a convex polytope, it can be rewritten as a convex combination of the vertices of U :

$$\begin{aligned} &\forall d \in D, \exists u = \sum \lambda_i u_i, u_i \in U.V, \\ &\lambda_i \ge 0, \sum \lambda_i = 1 : H^T \left(Eu + Gd\right) \ge 0 \end{aligned} \tag{A.6}$$

Note that $U.V$ is a set with finite elements, so $\max_{u_i \in U.V} \left\{H^T E u_i\right\}$ exists, and

$$\max_{u_i \in U.V} \left\{H^T E u_i\right\} \ge H^T E u \tag{A.7}$$

Therefore, let $u_m = \arg\max_{u \in U.V} \left\{H^T E u\right\}$,

$$H \in P_{Hs}^{u_m} \subseteq P_{Hs} \tag{A.8}$$

Eq. (A.8) proves that P_{Hs} contains all linear functionals corresponding to bounding half spaces satisfying the boundary conditions.

□

A.3. Proof of Claim 2:

First, write $Conv\left(P_H\right)$ as

$$\begin{aligned} Conv\left(P_H\right) &= Conv\left(P_{Hs} \cap X_{In}^{\Delta}\right) \\ &= Conv\left(\bigcup_{u \in U.V} \left(P_{Hs}^u \cap X_{In}^{\Delta}\right)\right) \end{aligned} \tag{A.9}$$

Recall the fact that for any union of polytopes $\bigcup_i P_i$, the vertices of its convex hull is a subset of the union of the vertices of all polytope components.

$$Conv\left(\bigcup_i P_i\right).V \subseteq \bigcup_i \left(P_i.V\right) \tag{A.10}$$

Therefore,

$$Conv\left(P_H\right).V \subseteq \bigcup_{u \in U.V} \left(P_{Hs}^u \cap X_{In}^{\Delta}\right).V \tag{A.11}$$

Recall the definition of a polar; it follows that

$$P_B.H = Conv\left(P_H\right).V \subseteq \bigcup_{u_i \in U.V} \left(P_{Hs}^{u_i} \cap X_{In}^{\Delta}\right).V \tag{A.12}$$

Therefore the bounding hyperplanes of P_B lie inside the intersection of X_{In}^{Δ} and P_{Hs} , and P_B is an avoidable set that contains X_{In} .

□

A.4. Proof of Claim 3:

First, we show that for all $H \notin P_H$, $H^T x = 1$ is not a valid bounding hyperplane for the avoidable set containing X_{In} . Then we show that for all $H \in P_H$, the half space $H^T x \le 1$ contains P_B . In other words, P_B is minimal.

For all $H \notin P_H$, by definition, $H \notin X_{In}^{\Delta} \vee H \notin P_{Hs}$. If $H \notin X_{In}^{\Delta}$, then from Eq.(A.4), $H^T x \le 1$ is a half space that does not contain X_d . If $H \notin P_{Hs}$, from Claim 1, H does not satisfy the boundary conditions.

For all $H \in P_H$, $H \in P_H \Rightarrow H \in Conv\left(P_H\right)$, so

$$H = \sum \lambda_i H_i, \lambda_i \ge 0, \sum \lambda_i = 1, H_i \in Conv\left(P_H\right).V \tag{A.13}$$

For all $x_0 \in P_B$, $H^T x_0 = \sum \lambda_i {H_i}^T x_0$. Since $x_0 \in P_B$, it follows that

$$\begin{aligned} &{H_i}^T x_0 \le 1, \forall H_i \in Conv\left(P_H\right).V \\ &\Rightarrow H^T x_0 = \sum \lambda_i {H_i}^T x_0 \le 1 \end{aligned} \tag{A.14}$$

Eq. (A.14) implies that adding $H^T x \le 1$ as a half space to P_B will not reduce any point from P_B , so P_B is minimal.

□

A.5. Proof of Claim 4:

Let P_H be the polytope consisting of all feasible bounding hyperplanes H for an avoidable set P_B . P_H may not contain the origin in its interior. Normalize the expression of its bounding hyperplanes to

$$\begin{cases} {v_i}^T H \le 1, i = 1, \ldots N_v \\ {\alpha_j}^T H \le 0, j = 1, \ldots N_\alpha \end{cases} \tag{A.15}$$

where the first category denotes bounding hyperplanes corresponding to facets that do not contain the origin, and the second category denotes those facets that contain the origin. N_v and N_α are the number of linear constraints for each category.

Let $H_v = \left\{v_i\right\}$ and $H_\alpha = \left\{\alpha_i\right\}$ denote the two groups of bounding hyperplanes in Eq. (A.15). Since P_H is constructed by intersecting P_{Hs} , where every facet contains the origin, and X_{In}^{Δ} , where every facet does not contain the origin, it is clear that $H_v \subseteq X_{In}^{\Delta}.H = X_{In}.V$. Recall that shifting a polytope is equivalent to shifting all the vertices. It follows that

$$\overline{X_{In}}^{\Delta}.H = \left\{v + c, \forall v \in X_{In}.V\right\} \tag{A.16}$$

Since the construction of P_{Hs} is an invariant of the shifting operation in the state space, P_{Hs} remains the same after shifting X_{In} . It then follows that $\overline{P_H}$ has the following bounding half spaces:

$$\begin{cases} \left(v + c\right)^T H \le 1, \forall v \in H_v \\ \alpha^T H \le 0, \forall \alpha \in H_\alpha \end{cases} \tag{A.17}$$

For any vertex H_0 of $\overline{P_H}$, it is the intersection of N facets, where N is the dimension of the state space. Assume the linear equations corresponding to H_0 is

$$AH_0 = a,$$
$$A = \begin{bmatrix} v_{p1} & \dots & v_{pM} & \alpha_{q1} & \dots & \alpha_{qN-M} \end{bmatrix}^T \quad \text{(A.18)}$$
$$a = \begin{bmatrix} 1 & \dots & 1 & 0 & \dots & 0 \end{bmatrix}^T$$

Since Eq. (A.18) has a unique solution H_0, A is invertible and $H_0 = A^{-1}a$. In addition, since $H_0 \in P_H$,

$$\begin{aligned} &\forall v \in H_v, v \neq v_{p1}, \dots v_{pM}, v^T H_0 < 1 \\ &\forall \alpha \in H_\alpha, \alpha \neq \alpha_{q1}, \dots \alpha_{qN-M}, \alpha^T H_0 < 0 \end{aligned} \quad \text{(A.19)}$$

Now consider the corresponding vertex $\overline{H_0}$ of $\overline{P_H}$. If it exists, it satisfies

$$\left(A + ac^T\right)\overline{H_0} = a \quad \text{(A.20)}$$

If the matrix $\left(A + ac^T\right)$ is invertible, and $\left(A + ac^T\right)^{-1} a$ satisfies all inequalities of $\overline{P_H}$, then $\overline{H_0}$ exists and is a vertex of $\overline{P_H}$.

From the matrix inversion lemma, if A and $1 + c^T A^{-1} a$ are invertible, then $\left(A + ac^T\right)$ is invertible and

$$\left(A + ac^T\right)^{-1} = A^{-1} - A^{-1}a\left(1 + c^T A^{-1} a\right)^{-1} c^T A^{-1} \quad \text{(A.21)}$$

Clearly A is invertible; $1 + c^T A^{-1} a$ is a scalar. Since $\overline{X_{In}} = X_{In} + c$ contains the origin, $-c$ is inside X_{In}. H_0 is a vertex of P_H, so $H_0{}^T(-c) < 1$. It follows that

$$1 + c^T A^{-1} a = 1 + c^T H_0 > 0 \quad \text{(A.22)}$$

Therefore

$$\overline{H_0} = \left(1 + c^T A^{-1} a\right)^{-1} A^{-1} a \quad \text{(A.23)}$$

Now we need to determine whether $\overline{H_0}$ is inside $\overline{P_H}$. For all $v \in H_v, v \neq v_{p1}, \dots v_{pM}$:

$$(v + c)^T \overline{H_0} = \left(1 + c^T A^{-1} a\right)^{-1} \left(v^T A^{-1} a + c^T A^{-1} a\right) \quad \text{(A.24)}$$

From Eq., $v^T A^{-1} a = v^T H_0 < 1$. It follows that

$$(v + c)^T \overline{H_0} < \frac{1 + c^T A^{-1} a}{1 + c^T A^{-1} a} = 1 \quad \text{(A.25)}$$

For all $\alpha \in H_\alpha, \alpha \neq \alpha_{q1}, \dots \alpha_{qN-M}$,

$$\alpha_i{}^T \overline{H_0} = \left(1 + c^T A^{-1} a\right)^{-1} \alpha_i{}^T H_0 < 0 \quad \text{(A.26)}$$

Therefore $\overline{H_0}$ is a vertex of $\overline{P_H}$. Since the mapping from the vertices of P_H to the vertices of $\overline{P_H}$ is established, the only thing left to check is whether $\overline{P_B} = \overline{P_H}^{\Delta} = P_B + c$. Note that

$$\begin{aligned} &\overline{H_0}^T (x + c) \leq 1 \\ &\Leftrightarrow \left(1 + c^T A^{-1} a\right)^{-1} a^T \left(A^{-1}\right)^T (x + c) \leq 1 \\ &\Leftrightarrow \frac{a^T \left(A^{-1}\right)^T x + a^T \left(A^{-1}\right)^T c}{1 + c^T A^{-1} a} \leq 1 \\ &\Leftrightarrow \frac{a^T \left(A^{-1}\right)^T x}{1 + c^T A^{-1} a} \leq \frac{1}{1 + c^T A^{-1} a} \\ &\Leftrightarrow H_0{}^T x \leq 1 \end{aligned} \quad \text{(A.27)}$$

Eq. (A.27) proves that for each facet of P_B, a corresponding facet of $\overline{P_B}$ exists by shifting the facet with vector c, i.e., $\overline{P_B} = P_B + c$.

□

A.6. Proof of Claim 5:

$\bar{\Sigma}$ is actually a differential inclusion of the original system Σ_0, which implies:

$$\begin{aligned} &\forall x \in S_0, d \in D_0, \forall u \in U, \exists \bar{d} \in \bar{D}, \\ &s.t.\ \dot{x} = f(x, u, d) = \bar{f}(x, u, \bar{d}) \end{aligned} \quad \text{(A.28)}$$

This means for any x, u, d in Σ_0, there exists $\bar{d} \in \bar{D}$ that reproduce the state derivative in system $\bar{\Sigma}$ with the same state and input. Therefore any avoidable set under dynamic $\bar{\Sigma}$ is also an avoidable set under dynamic Σ_0.

It follows that P is also an avoidable set for Σ_0.

□

A.7. Proof of Claim 6:

Only the case when $\theta > 0$ is proved. The case when $\theta < 0$ follows the same reasoning. For any point x at the boundary of P, suppose $\theta > 0$, then $H_4 \geq 0$. P is an avoidable set w.r.t. the dynamic system $\tilde{\Sigma}$. Therefore,

$$\forall \bar{d} \in \bar{D}, \exists u \in U : H^T \left(E_1 u + G_1 \bar{d}\right) \geq 0 \quad \text{(A.29)}$$

Since

$$\begin{aligned} &H^T \left(E_1 u + G_1 \bar{d} + k\right) \\ &= H^T \left(E_1 u + G_1 \bar{d}\right) + \frac{H_4 \sin\theta}{\Delta X^2 + \Delta Y^2} \\ &\geq H^T \left(E_1 u + G_1 \bar{d}\right) \geq 0 \end{aligned} \quad \text{(A.30)}$$

It follows that P is also an avoidable set w.r.t. $\bar{\Sigma}$.

□

A.8. Derivation of Eq. (42)

From the definition of the barrier function in Eq. (40), the derivative of $B(x)$ is

$$\dot{B}(x) = \frac{\partial B}{\partial b} \frac{\partial b}{\partial x} \dot{x} \quad \text{(A.31)}$$

Assuming that $\dot{x}$ is constant within the sampling time T_s,

$$\frac{d\left(\frac{\partial b}{\partial x}\dot{x}\right)}{dt}=\frac{d\left(H^T\dot{x}\right)}{dt}=0 \tag{A.32}$$

It follows that

$$B^{(n)}(x)=\frac{\partial^n B}{\partial b^n}\left(H^T\dot{x}\right)^n \tag{A.33}$$

For this certain function, $\frac{\partial^n B}{\partial b^n}$ has an explicit expression:

$$\frac{\partial^n B}{\partial b^n}=\frac{(-1)^n(n-1)!\left[(1+b)^n-b^n\right]}{(1+b)^n b^n} \tag{A.34}$$

Given a sampling time T_s, the barrier function at the next time step can be calculated using the Taylor expansion:

$$\begin{aligned}B(t+T_s)&=B(t)+\sum_{i=1}^{\infty}\frac{1}{n!}B^{(i)}(t)T_s{}^i\\&=B(t)+\sum_{i=1}^{\infty}\frac{1}{n}\frac{(-1)^i\left[(1+b)^i-b^i\right]}{(1+b)^i b^i}\left(H^T\dot{x}\right)^i T_s{}^i\end{aligned} \tag{A.35}$$

It follows that

$$B(t+T_s)-B(t)\le\sum_{i=1}^{\infty}\left(\frac{-H^T\dot{x}T_s}{b}\right)^i=\frac{-H^T\dot{x}T_s}{b+H^T\dot{x}T_s} \tag{A.36}$$

Recall Eq. (41), we have

$$\frac{-\frac{H^T\dot{x}T_s}{b}}{1+\frac{H^T\dot{x}T_s}{b}}\le\frac{c_1}{B}T_s\Rightarrow H^T\dot{x}\ge-\frac{c_1 b}{c_1T_s+B} \tag{A.37}$$

A.9. Simulation setup

The parameters for simulations are listed in TABLE III.

TABLE III SIMULATION PARAMETERS

Parameter	Value	Meaning
$v_{d\max}$	$1.2m/s$	Maximum obstacle speed
$r_{\max}$	$3.4rad/s$	Maximum yaw rate
$a_{\max}$	$4m/\sec^2$	Maximum acceleration
$v_{\max}$	$2m/s$	Maximum vehicle speed
R_v	$0.5m$	Radius of AV
R_p	$0.3m$	Radius of pedestrians
T_s	$0.05s$	Sampling time
μ	0.7	Friction coefficient
σ_{ax}, σ_{ay}	1	Standard deviation of pedestrian acceleration on X and Y direction

The simulations are conducted using Matlab. The toolboxes used are shown in Table II.

TABLE IV TOOLBOXES USED

Polytope Calculation	Multi-Parametric Toolbox 3
Mixed Integer Programming	Gurobi 6.0.4
Hamilton Jacobi Calculation	Level Set Toolbox 1.1.1

A.10. MPC design

The greedy MPC navigates the vehicle to the destination without any knowledge of the pedestrians. The nonlinear unicycle model of the vehicle is linearized for MPC.

$$\begin{bmatrix}\dot{X}\\\dot{Y}\\\dot{v}\\\dot{\bar{\psi}}\end{bmatrix}=\begin{bmatrix}0&0&\cos(\psi_0)&-v\sin(\psi_0)\\0&0&\sin(\psi_0)&v\cos(\psi_0)\\0&0&0&0\\0&0&0&0\end{bmatrix}\begin{bmatrix}X\\Y\\v\\\bar{\psi}\end{bmatrix}+\begin{bmatrix}0\\0\\a\\r\end{bmatrix} \tag{A.38}$$

where

$$\psi_0=\psi(t_0)\qquad\bar{\psi}=\psi-\psi_0 \tag{A.39}$$

This linearized model is then discretized and used for the MPC, with prediction horizon $N_{pred}=5$ and control horizon $N_{con}=1$.

A.11. Potential field controller design

We adapted the method proposed by Shimoda et al. in (Shimoda, Kuroda, and Iagnemma 2005) and tune the parameter to enhance its performance. The potential field is built on the trajectory space, which is the Cartesian product of velocity and yaw rate (different from the original trajectory space in (Shimoda, Kuroda, and Iagnemma 2005) because of the different inputs of the dynamic model). The potential field function used is

$$\begin{aligned}f(v,r)&=K_g(r-r_{des})^2+K_{v1}(v-v_{des})^{k_{v2}}+\\&\sum_{i=1}^{N_{ped}}\frac{K_o(K_{ov}v+1)}{(K_{oa}A_d{}^i+1)\exp\left(\frac{-(r-r_o^i)^2}{2\sigma^i}-K_{od}d^i/v^2\right)}\end{aligned} \tag{A.40}$$

where A_d, r_o, d and σ are defined in the following

$$\begin{aligned}A_d&=\left|\arctan\frac{Y_{des}-Y}{X_{des}-X}-\arctan\frac{Y_d-Y}{X_d-X}\right|\\r_o&=\frac{1}{2}(r_1+r_2)\\d&=\sqrt{(X_d-X)^2+(Y_d-Y)^2}-R_v-R_p\\\sigma&=|r_2-r_1|\end{aligned} \tag{A.41}$$

r_1 and r_2 are the maximum and minimum yaw rate that will lead to collision, as demonstrated in Fig. 10. See (Shimoda, Kuroda, and Iagnemma 2005) for more details.

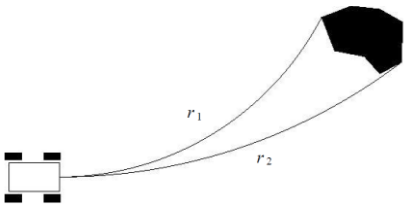

Fig. 10. Maximum and minimum yaw rates leading to collision (original figure in (Shimoda, Kuroda, and Iagnemma 2005))

The parameters for the potential field method are listed in TABLE V.

TABLE V PARAMETER OF THE POTENTIAL FIELD CONTROLLER

K_g	5
K_{v1}	15
K_{v2}	2
K_o	100
K_{oa}	0.1
K_{ov}	0.1
K_{od}	2

A.12. Hamilton Jacobi controller design

The control problem is formed and solved as a pursue-evade game. First, we solve a reachability set for a single obstacle and restrict the vehicle's input to only braking, then apply this reachability set to multiple obstacles (the same "trick" used in our polar method). It should be noted that the coordinate for relative dynamics in the Hamilton Jacobi method differs from that for the polar algorithm.

$$x=\begin{bmatrix}\Delta X_L\\ \Delta Y_L\\ v\end{bmatrix}=\begin{bmatrix}\Delta X\cos\psi+\Delta Y\sin\psi\\ -\Delta X\sin\psi+\Delta Y\cos\psi\\ v\end{bmatrix} \tag{A.42}$$

where ΔX_L and ΔY_L are the relative position of the obstacle in the local frame attached to the vehicle. Theoretically, the three states in Eq. are sufficient for describing the relative dynamics between the AV and the obstacle. The reason for using four states are used in polar algorithm is that the model shown in Eq. (4) is more similar to double integrators, which makes the simplification in the polar algorithm easier. The dynamic equation for the states in Eq. (A.42) is

$$\dot{x}=\begin{bmatrix}-v+\Delta Y_L r+v_{Ldx}\\ -\Delta X_L r+v_{Ldy}\\ a\end{bmatrix} \tag{A.43}$$

where v_{Ldx} and v_{Ldy} are the longitudinal and lateral projection of the obstacle's speed in the local coordinates. The dynamics are then reversed in time to calculate the backwards reachable set.

The value function is defined as

$$\varphi(x,\mathrm{t})=\min_{a\in\mathcal{A}}\max_{b\in\mathcal{B}}\int_{t_0}^{t}0dt+\mathcal{G}\left(x(t)\right) \tag{A.44}$$

and $\mathcal{G}(x)$ satisfies:

$$\mathcal{G}(x)\begin{cases}<0, & \Delta X_L{}^2+\Delta Y_L{}^2<\left(R_v+R_p\right)^2\\ =0, & \Delta X_L{}^2+\Delta Y_L{}^2=\left(R_v+R_p\right)^2\\ >0, & \Delta X_L{}^2+\Delta Y_L{}^2>\left(R_v+R_p\right)^2\end{cases} \tag{A.45}$$

The Hamiltonian of the reversed dynamics allowing only brake is solved analytically:

$$\begin{aligned}H&=\min_{b\in\mathcal{B}}\max_{a\in\mathcal{A}}p^T f\left(x,t,a(x,t),b(x,t)\right)\\&=p_1 v+k(p_3,v)-\sqrt{\frac{p_1{}^2+p_2{}^2}{2}}v_{d\max}\\ k(p_3,v)&=\begin{cases}\min\{0,p_3a_{\max}\}, & v=0\\ -|p_3a_{\max}|, & 0<v<v_{\max}\\ \min\{0,-p_3a_{\max}\}, & v=v_{\max}\end{cases}\end{aligned} \tag{A.46}$$

where $a\in\mathcal{A}$ and $b\in\mathcal{B}$ are the strategy for the vehicle and obstacle, respectively. $p=\dfrac{\partial\phi}{\partial x}$ is the conjugate momenta; $k(p_3,v)$ ensures that the vehicle speed is within the limit. To enforce the responsibility rules presented in Section 2.2,

$$\left(\Delta X_L\le 0\vee v=0\right)\rightarrow H=0 \tag{A.47}$$

The final time is chosen as $T=10s$ .The calculated zero level set is shown in Fig. 11.

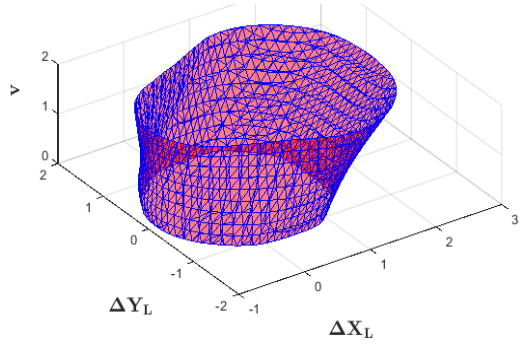

Fig. 11. Hamilton Jacobi reachability set

The result is calculated numerically and stored in a look-up table. In the implementation, local regression is performed to obtain the value and the gradient of φ . Then the following constraint is enforced:

$$\nabla_\varphi(x)^T f(x,u,d)\ge-\alpha\varphi(x) \tag{A.48}$$

where ∇_φ is the gradient at that point and α is a positive constant. For more details, please refer to (Evans and Souganidis 1983) and (Mitchell, Bayen, and Tomlin 2005).