A Pursuit Evasion Approach for Avoiding an Inattentive Human in the Presence of a Static Obstacle

Yi Feng Wang Christopher Nielsen Stephen L. Smith

***Abstract*— We consider a mobile robot, modelled as a Dubins' car, following a path. In the vicinity of the path is a human, modelled as an agile point mass, and a static obstacle. The robot's objective is to follow the path unless it is absolutely necessary to deviate so as to avoid collision with the static obstacle or human. We seek to guarantee robot's safety, even if the human is distracted or inattentive, and thus we consider worst-case motions for the human. The resulting problem takes the form of a reversed homicidal chauffeur game, but with the addition of a static obstacle. The static obstacle can interfere with the escape route of the robot, and thus fundamentally changes the form of the game. We propose a navigation algorithm for the robot that provably guarantees safety and that attempts to delay its reaction for as long as possible. We validate the proposed approach in simulation and provide a comparison to existing collision avoidance methods.**

I. Introduction

Robots and self-driving vehicles have become prominent players in modern day logistics. A key challenge is to safely deploy a mobile robot in a dynamic environment with both moving agents, e.g. humans, and static obstacles. In this paper, we consider a Dubins' car-like robot following a path with a human, modelled as a point mass, and a static obstacle in the vicinity. The robot tries to stay on its path unless absolutely necessary in order to avoid a collision with the human or static obstacle. We seek to guarantee safety, even if the human is distracted or inattentive, and thus we consider worst-case motions of the human.

A popular approach to tackle collision avoidance problems is model predictive control (MPC) [1]. MPC-based approaches often model obstacles as state constraints [2]. MPC methods show promising performance but often lack formal safety guarantees in the face of adversarial agents.

An alternative method is to pose collision avoidance as a *pursuit-evasion game* [3]–[5], which is an adversarial game between two groups of opposing players. The solution of the game provides two distinct sets of states known as the "kill zone" and "escape zone." In order to guarantee the robot's safety, the robot control must prevent the game's state from entering the kill zone where the adversarial agent always has a successful strategy to collide with the robot. This approach is useful because it provides a formal way to guarantee safety.

In recent work [6]–[9], the authors have posed the collision avoidance problem as a differential game and have computed the *backward reachable set* (BRS) by solving Hamilton-Jacobi-Isaacs (HJI) partial differential equation [10] numerically. If the game's state enters the BRS and the agent is adversarial, then collision cannot be prevented regardless of the robot's action. The BRS is computed offline and the backward reachability solutions are then utilized by the robot when running online. As the dimension of the game's state-space increases, the memory and the computational time for the BRS increase exponentially [10].

Similar to the work above, we take a pursuit-evasion approach to our problem in order to obtain safety guarantees. The most closely related work to our problem is the *reversed homicidal chauffeur game* (also called the suicidal pedestrian) [11]. In this game, the robot (evader) is a Dubins' vehicle and the human (pursuer) is an agile point object. In [11], the authors completely characterized the solution to this game. Our paper differs from this work in that the environment contains a static obstacle which allows the human to use the obstacle to limit the robot's evasive options. A similar challenge was faced when introducing a circular static obstacle to the lion and man game [12] and static obstacles to the pursuit evasion game [13]. However, both players in [12] and [13] are modelled as point masses.

The main contribution of this paper is to provide an event-triggered controller that guarantees the robot to be collision-free. Our evasion strategy also tries to delay the robot's first evasive reaction for as long as possible. Compared to methods based on numerically computing the BRS and the use of the MPC, our strategy requires much less computational power and memory. Additionally, the methods of numerically computing the BRS do not consider the existence of a obstacle in the surrounding. In simulations, we show the benefit of the proposed approach.

II. The problem setup

The robot (evader) is modelled as a Dubins' vehicle with fixed translation speed $v_e > 0$ and minimum turning radius $R > 0$. The robot's control input u is proportional to the angular velocity of its heading angle and is constrained by its maximum turning radius, $|u(\cdot)| \leq 1$. Following [3], we attach an orthonormal moving frame to the evader whose origin is the robot's position in the plane with respect to a fixed inertial coordinate system. The ordinate axis points in the robot's heading direction; the abscissa results in a positively oriented body frame. The body frame is referred to as the reduced space [3].

The human (pursuer) is the more agile agent and is modelled as a point mass with fixed speed $v_p > 0$. The

This research was supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC)

The authors are with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada {y3438wang;cnielsen;stephen.smith}@uwaterloo.ca

coordinates of the pursuer in the reduced space are denoted by $\boldsymbol{p}(t) = (x_p(t), y_p(t)) \in \mathbb{R}^2$. The equations of motion of the pursuer in the reduced space are

$$\begin{aligned} \dot{x}_p(t) &= -\frac{v_e}{R} y_p(t) u(t) - v_p \sin(\phi(t)) \\ \dot{y}_p(t) &= \frac{v_e}{R} x_p(t) u(t) - v_e - v_p \cos(\phi(t)) \end{aligned} \tag{1}$$

where $\phi \in (\pi, \pi]$ represents the pursuer's relative heading. We assume that $v_e \geq v_p$ so that it is possible for the robot to evade an adversarial human.

There is a static obstacle in the workspace modelled as point in the plane and whose coordinates in the reduced space are written $\boldsymbol{o}(t) = (x_o(t), y_o(t)) \in \mathbb{R}^2$. While the obstacle is static, its position changes in the reduced space due to the motion of the robot and its equations of motion are

$$\begin{aligned} \dot{x}_o(t) &= -\frac{v_e}{R} y_o(t) u(t) \\ \dot{y}_o(t) &= \frac{v_e}{R} x_o(t) u(t) - v_e. \end{aligned} \tag{2}$$

In this paper, we propose an evasion policy in the case of a straight line nominal path with associated nominal control signal $u_0 = 0$.

To characterize a collision and to account for the geometry of the obstacle, the robot and the human, we fix a real constant $c > 0$ called the collision radius and we say that the robot has been **captured** or has **collided** if either $\|\boldsymbol{p}\|_2 < c$ or $\|\boldsymbol{o}\|_2 < c$. The collision radius c and the robot's minium turning are assumed to satisfy $R > c$. The setup in the reduced space is illustrated in Figure 1.

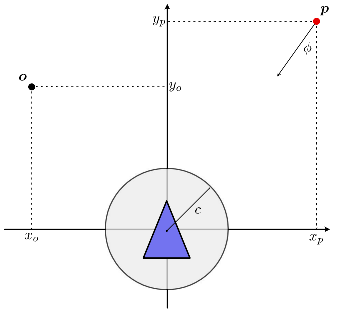


Fig. 1. Problem setup in the reduced space.

An **evasive action** at time t is a piecewise continuous control signal with the property that[1] $u(t^+) \neq u_0(t^+)$, i.e., it results in the evader leaving its nominal path. The class of allowable evasive actions are Dubins paths [14] denoted CSC where $C \in \{R, L\}$ is a hard turn with minimum turning radius either to the right (R) or to the left (L) and S is the straight maneuver. The duration of C and S can be zero so that, for example, a hard right followed by a hard left is an allowable evasive action.

To ensure well-posedness, we assume that the initial conditions of the robot, human and the obstacle are such that there exists a robot strategy for which a collision can be avoided. Under these conditions, in this article we seek a real-time evasion policy for the robot that guarantees that the robot is never captured and which is minimally invasive in the sense that the robot only deviates from its nominal control signal when it is absolutely necessary in order to avoid a collision.

Problem. *Find an evasion policy for the robot that:*

(i) *guarantees that the evader is not captured and,*
(ii) *maximizes the following quantity*

$$\inf\{t \geq 0 : u(t) \neq u_0(t)\}.$$

Remark 1. *We assume that human and the obstacle start on opposite sides of the robot, i.e., $x_o(0)x_p(0) \leq 0$ because otherwise the problem considered herein reduces to that of the reversed homicidal chauffeur [11]. For simplicity of exposition and in light of the reflective symmetry about the ordinate axis in the reduced space, we also assume without loss of generality that $x_o(0) \leq 0$ and $x_p(0) \geq 0$.*

III. Technical Preliminaries

We first introduce some concepts based on the classical homicidal chauffeur problem [3] and the closely related suicidal pedestrian differential game [11]. We use these ideas to define the notions of a default escape and configurations where the robot can guarantee its safety.

A. Collision zones

In two player pursuit evasion games, the notion of a *barrier surface* can be used to define a simple closed curve that divides the plane into two open sets. The first open set is bounded and has the property that if the game is initialized there, then the pursuer will inevitably capture the evader under optimal play. The second open set is unbounded and has the property that if the game is initialized there, then the evader can always avoid capture. If the game is initialized on the barrier surface itself, then the game's state remains in this set for some time under optimal play, i.e., the barrier surface is a (locally) positively invariant set under optimal play.

In [11], it is shown that the robot-human barrier surface can be expressed in terms of the parameterized curve

$$\begin{aligned} x(\tau) &= R\cos\left(\frac{v_e}{R}\tau\right) - R + (c + v_p\tau)\sin\left(\bar{s} - \frac{v_e}{R}\tau\right) \\ y(\tau) &= R\sin\left(\frac{v_e}{R}\tau\right) + (c + v_p\tau)\cos\left(\bar{s} - \frac{v_e}{R}\tau\right) \end{aligned} \tag{3}$$

where $\tau \in [0, \overline{\tau}]$ and $\overline{\tau}$ is the solution to

$$(c + v_p\overline{\tau})\sin\left(\bar{s} - \frac{v_e}{R}\overline{\tau}\right) = R - R\cos\left(\frac{v_e}{R}\overline{\tau}\right).$$

The barrier surface for the robot-human pair is the set

$$\mathcal{B} := \left\{\begin{bmatrix} x(\tau) \\ y(\tau) \end{bmatrix}\right\} \cup \left\{\begin{bmatrix} -x(\tau) \\ y(\tau) \end{bmatrix}\right\}, \quad \tau \in [0, \overline{\tau}], \tag{4}$$

defined using (3). We combine (4) with the boundary of the collision disk to define the **robot-human collision zone**, denoted $\mathcal{CZ}$, as the interior of the simple closed curve

$$\mathcal{B} \cup \left\{c\begin{bmatrix} \sin(s) \\ \cos(s) \end{bmatrix} : \arccos(-v_p/v_e) \leq |s| \leq \pi\right\}. \tag{5}$$

[1] The notation $u(t^+)$ denotes the limit of $u(\tau)$ as $\tau \to t$ from the right.

This collision zone is depicted as the light blue region in Figure 2a. If $\boldsymbol{p}$ lies in the collision zone, then the robot has either already been captured or capture is inevitable. The optimal strategy for the robot when there is no obstacle is to take a hard turn (in the appropriate direction) whenever the human $\boldsymbol{p}$ lands on the barrier curve (4) and keep turning until $\boldsymbol{p}$ slides off of $\mathcal{B}$. From hereon, we call this the *reversed homicidal chauffeur strategy*.

An analogous concept of a collision zone can be defined for the robot-obstacle pair. Introduce the parameterized curves

$$\begin{aligned} x(\tau_o) &= -R + (c+R)\cos\left(\frac{v_e}{R}\tau_o\right) \\ y(\tau_o) &= (c+R)\sin\left(\frac{v_e}{R}\tau_o\right) \end{aligned} \tag{6}$$

where $\tau_o \in [0, \overline{\tau}_o]$ and

$$\overline{\tau}_o := \frac{R}{v_e}\arccos\left(\frac{R}{c+R}\right).$$

The barrier surface for the robot-obstacle pair is the set

$$\mathcal{B}_o := \left\{\begin{bmatrix} x(\tau_o) \\ y(\tau_o)\end{bmatrix}\right\} \cup \left\{\begin{bmatrix} -x(\tau_o) \\ y(\tau_o)\end{bmatrix}\right\}, \quad \tau \in [0, \overline{\tau}_o], \tag{7}$$

defined using (6). Combining (6) with the boundary of the collision disk we define the **robot-obstacle collision zone**, denoted $\mathcal{CZ}_o$, as the interior of the simple closed curve

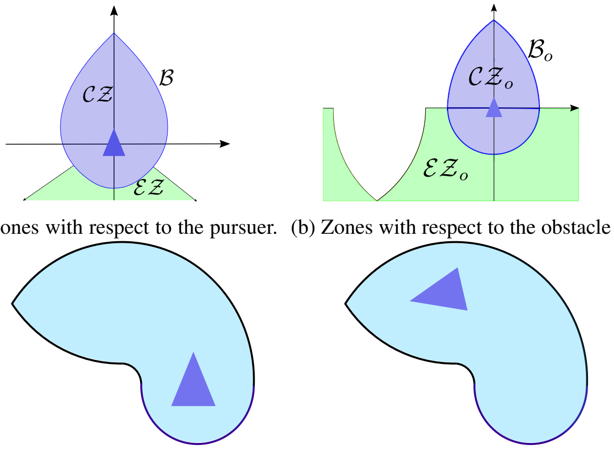
$$\mathcal{B}_o \cup \left\{ c\begin{bmatrix} \sin(s) \\ \cos(s)\end{bmatrix} : \pi/2 \le |s| \le \pi \right\} \tag{8}$$

which is depicted as the blue region in Figure 2b. If $\boldsymbol{o}$ lies in the collision zone, then the robot has either already collided with the obstacle or a collision is unavoidable. The optimal strategy for the robot when there is no human is to take a hard turn (in the appropriate direction) whenever the obstacle lands on the barrier curve (7) up until $\boldsymbol{o}$ slides off of $\mathcal{B}_o$.

Remark 2. *Since the obstacle is static, the robot-obstacle collision zone $\mathcal{CZ}_o$ is a proper subset of the robot-human collision zone $\mathcal{CZ}$.*

B. Default escape route

As discussed above, if there is no obstacle and the robot applies the reversed homicidal chauffeur strategy, then under optimal pursuit the human slides along the barrier surface until leaving it. The duration of this maneuver depends on where $\boldsymbol{p}$ lands on $\mathcal{B}$ and is denoted $\tau(\boldsymbol{p}) \in \mathbb{R}_{\ge 0}$. We call the area traced out by the closure of the static collision zone $\mathcal{CZ}_o$ the **default escape route** shown in Figures 2c and 2d. To avoid a collision between the robot and the obstacle while executing reversed homicidal chauffeur strategy, the obstacle must not be in the interior of the default escape route. The *largest default escape route* occurs when the the duration of the robot's hard turn is the longest, i.e., the human is on the ordinate axis lands on the tip of the barrier curve.



(a) Zones with respect to the pursuer. (b) Zones with respect to the obstacle

(c) Initial robot position before taking the default escape route.

(d) Resulting robot position after taking the default escape route.

Fig. 2. The default escape route of the robot, collision, and evasion zones with respect to both the pursuer and the obstacle.

C. Guaranteed evasion configurations

To simplify the forthcoming discussion, it is convenient to formally characterize configurations in which one of the adversarial players (human or obstacle) is eliminated from the game and safety is easy to ensure by using the reversed homicidal chauffeur strategy. We call such configurations guaranteed evasion configurations.

The **robot-human evasion zone** is denoted $\mathcal{EZ}$ and is the green area in Figure 2a. It is formed by that portion of the cone passing through the points $(\pm x(0), y(0))$ obtained from (3) that is disjoint from the collision disk. We now make the following reasonable assumption.

Assumption 1. *If the human enters the robot-human evasion zone, then the human ceases their pursuit of the robot.*

The **robot-obstacle evasion zone** is denoted $\mathcal{EZ}_o$ and is the green area in Figure 2b. It is formed by that portion of the lower half plane that is disjoint from the largest default escape route.

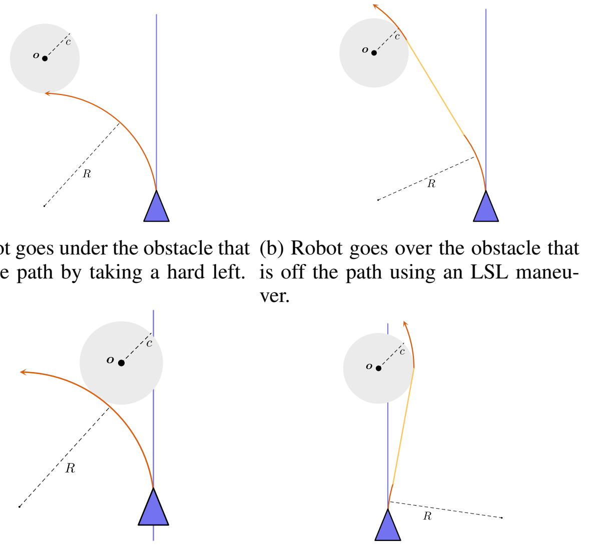
Definition 3. *Consider a configuration of the pursuer, obstacle, and evader, defined by the relative positions $\boldsymbol{o}$ and $\boldsymbol{p}$ in the reduced space. We say this is a **guaranteed evasion configuration** if either*

(i) $\boldsymbol{o} \in \mathcal{EZ}_o$ and $\boldsymbol{p} \notin \mathcal{CZ}$, or (ii) $\boldsymbol{o} \notin \mathcal{CZ}_o$ and $\boldsymbol{p} \in \mathcal{EZ}$.

In guaranteed evasion configuration (i), the game reduces to the reversed homicidal chauffeur. In guaranteed evasion configuration (ii), the evader applies maximum steering away from the obstacle when the obstacle arrives on its barrier surface.

IV. Event-Triggered Evasion

In this section we present an event-triggered evasion strategy for the robot whose high level principle is to only use the reversed homicidal chauffeur strategy when necessary in order to delay the evasive maneuver as much as possible.



(a) Robot goes under the obstacle that is off the path by taking a hard left.

(b) Robot goes over the obstacle that is off the path using an LSL maneuver.

(c) Robot goes under the obstacle that is on the path by taking a hard left.

(d) Robot goes over the obstacle that is on the path using an RSL maneuver.

Fig. 3. Robot's evasive maneuvers under the event-triggered controller.

A. *Overview of the strategy*

Initially, the robot follows its nominal path. If the obstacle is not in the interior of the default escape route, then the robot can evade the human using the reversed homicidal chauffeur strategy. If the obstacle arrives at the boundary of the default escape route and the human is not on its barrier surface, then the robot must decide whether or not to allow the obstacle to enter the default escape route and (temporarily) interfere with the reversed homicidal chauffeur strategy.

In this scenario, the proposed strategy first determines if the human can reach its barrier curve while the obstacle is in the default escape route. If so, then the robot must decide whether to go "under" the static obstacle by immediately taking the default escape route (Figure 3a and Figure 3c) or to delay the evasive maneuver by "threading the needle" and going between the obstacle and the human in the future (Figure 3b and Figure 3d); the latter maneuver is sometimes referred to as going "above" the static obstacle. The complexity of the problem arises from this decision.

To describe the proposed strategy in more detail, we first define the notion of a feasible evasive action.

Definition 4. *Consider a configuration of the evader, pursuer, and obstacle at time t defined by the relative positions $\boldsymbol{p}(t)$, $\boldsymbol{o}(t)$ in the reduced game space. An evasive action at time t is* ***feasible*** *if there exists a time $T > 0$ such that both of the following conditions hold:*

1) *for every $\tau \in [t, t+T]$, $\boldsymbol{o}(\tau) \notin \mathcal{CZ}_o$ and $\boldsymbol{p}(\tau) \notin \mathcal{CZ}$ and,*
2) *$(\boldsymbol{o}(t+T), \boldsymbol{p}(t+T))$ is a guaranteed evasion configuration.*

An evasive maneuver is triggered by the events listed in Table I. The obstacle avoidance algorithm continuously monitors these events and if an event occurs, then the evader responds with the corresponding evasive action.

TABLE I

EVENTS THAT TRIGGER AN EVASIVE ACTION.

Event	Event Description	Robot's Response
1	Human is on its barrier surface.	Turn hard away from human.
2	Obstacle is on the boundary of the default escape route and there is no feasible LS/LSL/RL/RSL.	Turn hard away from human.
3	Obstacle in the interior of default escape route and off the path and the last moment in time at which a feasible LS/LSL exists is reached.	Execute feasible LS/LSL.
4	Obstacle in the interior of default escape route and on the path and the last moment in time at which a feasible RL/RSL exists is reached.	Execute feasible RL/RSL.

Theorem 5. *Under the conditions described in Section II and Assumption 1, the execution of the algorithm in Table I guarantees that the robot is not captured.*

Proof. Let $T \geq 0$ denote the time at which the first event in Table I occurs.

Case 1: (Event 1 triggers first) By the definition of Event 1, the human is outside the collision zone at T. If the obstacle is outside the default escape route or on its boundary, then the response to Event 1 results in the game entering guaranteed evasion configuration. (ii) (see Definition 3). We now show that the obstacle cannot be inside the default escape route at time T. Suppose, by way of contradiction, that the obstacle is strictly inside the default escape route at T. Let $\bar{t}$, $0 \leq \bar{t} < T$ be the moment when the obstacle first arrives on the boundary of the default escape route. Since Event 2 did not trigger at $\bar{t}$, there must have existed a feasible LS/LSL/RL/RSL at $\bar{t}$. Since the pursuer is on the barrier surface at time T, the only feasible evasion maneuver is a hard left maneuver; there is no feasible LS/LSL/RL/RSL at time T. Therefore, there is a time in the interval $(\bar{t}, T)$ which was the last moment of which a LS/LSL/RL/RSL maneuver was feasible. Since Event 3 and 4 did not trigger, no such moment existed and therefore the obstacle cannot be in the interior of the default escape route at time T.

Case 2: (Event 2 triggers first) By the definition of Event 2, the obstacle is on the boundary of default escape route. Since Event 1 has not triggered, the human is not in its collision zone. Therefore, by turning hard left, the game enters guaranteed evasion configuration (i) (see Definition 3).

Case 3: (Event 3 triggers first) This case follows directly from the triggering condition. The evader has a feasible LS/LSL evasive action because the triggering condition is that T is the last moment for which a feasible LS/LSL exists. Since the LS/LSL is feasible, the human and the obstacle do not enter their respective collision zones while the evader executes this evasive action. After the evader avoids the obstacle by going "above" it, the game enters guaranteed evasion configuration (i).

Case 4: (Event 4 is triggered first at T) The proof is similar to Case 3.

Case 5: (No events trigger) Let t be an arbitrary time, and suppose no event has occurred in the interval $[0,t]$. Since Event 1 did not trigger in $[0,t]$, the human cannot be inside the collision zone at t. If the obstacle is completely outside or on the boundary of the default escape route, then the configuration at t is safe because the robot can turn hard left to guarantee evasion. Suppose the obstacle is strictly inside the default escape route at t and let $\bar{t}$, $0 \leq \bar{t} < t$, be the moment when the obstacle is first on the boundary of the default escape route. Since Event 2 did not trigger at $\bar{t}$, there existed a feasible LS/LSL/RL/RSL at $\bar{t}$. Suppose by way of contradiction that no feasible LS/LSL/RL/RSL exists at t. Then there must be a last moment in time in the interval $(\bar{t},t)$ for which a feasible LS/LSL/RL/RSL exists, and Event 3 or 4 will be triggered at this time. This contradicts that the assumption that no event is triggered for $[0,t]$. Therefore, if the obstacle is strictly inside the default escape route at t, there must still exist a feasible LS/LSL/RL/RSL at t and the robot is in a safe configuration. Since t is arbitrary, we see that the evader is safe for all times t before the first event occurs. □

V. Supporting results

To implement our algorithm, various subroutines are needed. First, we must be able to determine whether or not the human can arrive on its barrier surface before the obstacle leaves the default escape route. Second, to monitor Events 2, 3, and 4, we search the space of feasible evasion maneuvers seeking the last moment at which a feasible maneuver exists. This section describes how we resolve these issues.

A. *Determining if obstacle might interfere with evasion*

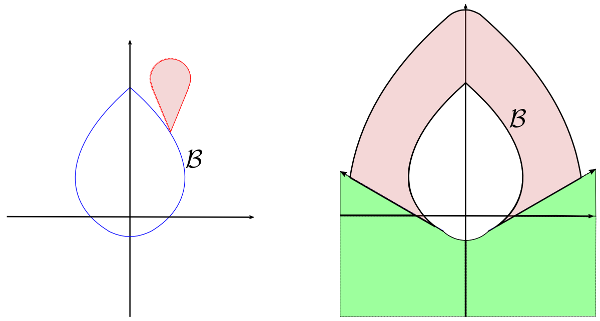
If the obstacle arrives at the boundary of the default escape route, then it is routine to compute how long it takes for it to leave the default escape route if the robot keeps following the path. To determine if the obstacle might interfere with evasion of the human during this period, we use *constant bearing control* [15] to find the minimum time for the human to reach its barrier surface (4). Given $\boldsymbol{p}(t) = (x_p(t), y_p(t))$ and $\boldsymbol{q}(\tau) := (x(\tau), y(\tau))$ from (3), if

$$y_p(t) - y(\tau) \geq \frac{1}{v_p}\big(x_p(t) - x(\tau)\big)\sqrt{v_e^2 - v_p^2}, \tag{9}$$

then the minimum time for the human to arrive at the point $\boldsymbol{q}(\tau) \in \mathcal{B}$ equals

$$T(\boldsymbol{p},\tau) = \frac{\sqrt{v_p^2\|\boldsymbol{p}-\boldsymbol{q}\|_2^2 - v_e^2\left(x_p - x\right)^2}}{(v_e^2 - v_p^2)} - \frac{v_e\left(y - y_p\right)}{(v_e^2 - v_p^2)}. \tag{10}$$

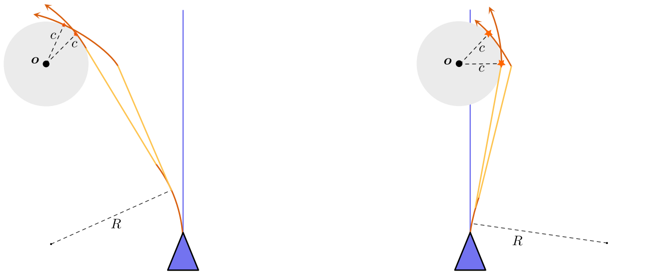
Otherwise, the human cannot get to the barrier curve. So for a given time T and a given $\boldsymbol{q}(\tau) \in \mathcal{B}$, we use (9) and (10) to characterize the set of human positions that can reach $\boldsymbol{q}(\tau)$ in T seconds or less; this is the "ice cream cone" shaped set illustrated in Figure 4a. Next, by varying the point $\boldsymbol{q}(\tau)$ on the barrier curve, we characterize the set of all human positions that can reach $\mathcal{B}$ in T seconds or less (Figure 4b).



(a) Configurations (red set) that reach a particular point.

(b) Configurations (red set) that reach any point.

Fig. 4. Human configurations that can reach its barrier surface in T or less seconds using constant bearing control.



(a) LSL maneuvers varied by the first L duration.

(b) RSL maneuvers varied by the first R duration.

Fig. 5. *Family of LSL/RSL maneuvers that graze the boundary of the obstacle's disk.*

Finally, letting the role of T be played by the time it would take for the obstacle to leave the default escape route, we can use the associated set in Figure 4b to determine whether the human can arrive on $\mathcal{B}$ prior to the obstacle leaving the default escape route. Additional details are given in [16].

B. *Existence of feasible evasive actions*

We discuss the subroutines used to check Events 3 and 4.

1) Obstacle off the nominal path: In Table I, the robot's evasive maneuver corresponding to Event 3 is to first move to a position where $\boldsymbol{o} \in \mathcal{EZ}_o$, illustrated in Figure 3b. In Figure 5a, this evasive action consists of the robot's first and second LS maneuver. The robot's minimum turning circle at the end of the second S segment encompasses the obstacle and forms a common tangent between both circles. At the end of the LS maneuver, the obstacle's threat can be eliminated and the robot can turn hard left to evade the human if necessary by using the last L maneuver of a LSL.

Let t_L denote the duration of the robot's first L maneuver. Since the obstacle is inside the default escape route, the robot can only turn till its heading is aligned with the common tangent line between the minimum turning circle of the first L maneuver and the collision disc fixed at the obstacle. Therefore, t_L is bounded by, say, $t_L^{\max}$. The S segment depends on the duration of the first L maneuver. Figure 5a illustrates the feasibility check for Event 3, which evaluates whether or not there exists a LSL maneuver with $t_L \in [0, t_L^{\max}]$ such that the human cannot get inside $\mathcal{CZ}$ during the execution of the LS maneuver.

By the results in [11], the human can only enter its capture zone on the S portion of an LSL maneuver. Thus we only need to check whether or not the human can get inside $\mathcal{CZ}$ during the S segment of each LSL with $t_L \in [0, t_L^{\max}]$ to determine feasibility. Based on the human's point mass dynamics, the area it can reach is bounded by an expanding circle centered at its initial position.

2) Obstacle on the nominal path: In this case, as Illustrated in Figure 3d, the robot must first turn right to avoid the obstacle. The check to determine a feasible RSL closely resembles the feasibility check of an LSL. The RS maneuver corresponding to Event 4 in Table I also tries to get the circle of the last L to encompass the obstacle (see Figure 5b). Unlike the LSL maneuver, the R portion of the RSL can be used to fully evade the obstacle by forcing the obstacle out of the default escape route and hence the last L maneuver can then be used to evade the human if necessary. The duration of the R maneuver is bounded and its max duration is to turn hard right until reaching the moment either the obstacle is out of the default escape route or the human can arrive on the barrier surface[2].

Since the duration of the R portion is bounded by, say $t_R^{\max}$, and since the duration of the S portion depends on that time, we search for RSL maneuvers whose R portion lasts between 0 and $t_R^{\max}$. The robot searches through the family of RSL maneuvers and checks whether or not there exists a RS maneuver for which the human cannot enter $\mathcal{CZ}$. If the evaluation is false, then Event 4 is triggered.

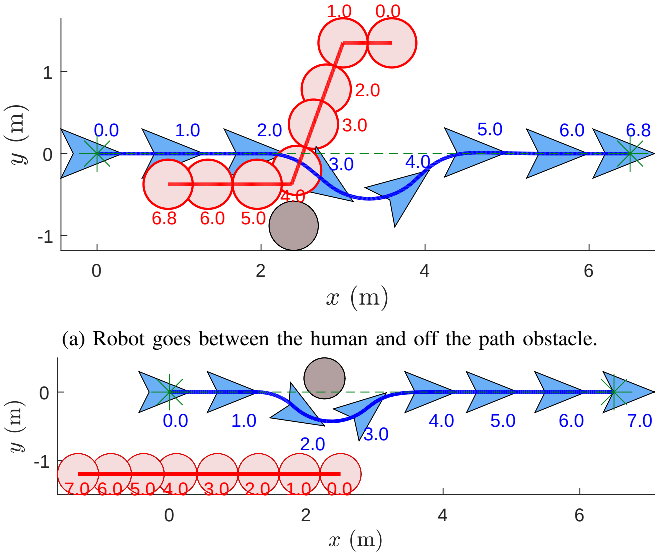
VI. Numerical simulations

We present numerical simulations and compare our event-triggered strategy to other collision avoidance methods.

A. Simulation setup

The following simulation parameters are used throughout: $v_e = 1$, $v_p = 0.6$, $R = 0.8$, $c = 0.6$. In the light of Remark 1, the human is initially placed to the right of the nominal path and the obstacle is to the left. The robot follows its nominal path using a pure pursuit controller [17] and switches to our evasive control strategy when an event occurs. Afterwards, upon achieving a guaranteed evasion, the robot switches back to the pure pursuit controller.

We consider three human motions: (i) diagonal cross (human moves diagonally across the path), (ii) straight cross (human move orthogonally towards the path), and (iii) downward move (human moves downwards parallel to the path). We exclude the straight cross simulations when the obstacle is on the nominal path because there is no significant difference in terms of robot's behaviour compared to the diagonal cross simulation. The nominal path is set to 6.5 unit length. To evaluate the performance of our control scheme we present the mission duration and cumulative path error from [18] in our quantitative analysis. For each human behaviour, we tested up to 1000 cases where the human and the obstacle were randomly placed.



(a) Robot goes between the human and off the path obstacle.

(b) Robot goes between the human and on the path obstacle.

Fig. 6. Robot (blue), human (red), and obstacle (grey) positions at the labeled time in evasive maneuver simulations with stars marking the start and end of nominal path.

TABLE II

Comparison between the event-triggered and simple approaches

Human Motions	t^a_{wait}	Task Duration (% Reduction)	Path Error (% Reduction)
Obstacle not on Nominal Path			
Diagonal cross maneuver	0.75	9.55	51.15
Straight cross maneuver	0.73	7.22	42.99
Downward maneuver	0.20	20.00	83.27
Obstacle on Nominal Path			
Diagonal cross maneuver	0.18	10.43	65.55
Downward maneuver	0.21	-1.63	-8.62

[a] After the obstacle arrives on the default escape route, the time that the robot continues following the nominal path before executing an evasive maneuver.

B. Comparison Studies

First, we compare our strategy to the simple strategy of going "under" the obstacle with a hard left whenever the obstacle hits the boundary of the default escape route or the pursuer is on its barrier surface. An example of our simulation is shown in Figure 6 for both the obstacle off the path and the obstacle on the path. In both scenarios, the robot tries to go in between the human and the obstacle if a feasible LSL/RSL evasive maneuver is available. In the case where the obstacle is not on the path, our strategy successfully delays the evasion when possible and outperforms the simple strategy for all human motions, see Table II. In the case where the obstacle is on the nominal path, our event-triggered strategy still outperforms the simple strategy when the human performs the diagonal cross maneuver. However the simple strategy slightly outperforms the event-triggered strategy when the human executes the downward maneuver. This is due to scenarios where the robot tries to evade

[2] The set from which the human can arrive is enclosed by an expanding circle. By checking whether or not this circle intersects with $\mathcal{CZ}$, we can find the duration of the R maneuver.

the obstacle by first turning hard towards the human and then immediately turning in the opposite direction to evade the human after evading the obstacle. Overall, our strategy significantly outperforms the simple evasion strategy.

Second, in the comparison between the velocity obstacle algorithm [19] and our evasion strategy, the former allows the robot to stay relatively closer to the nominal path when making evasive maneuvers. However, it cannot guarantee its safety if the human tries to "squeeze" the robot with the obstacle.

Third, we studied the evasion policies under ISO 13482 [20] and [21]. Both use the idea of *safety fields* and *passive safety* and assume that moving agents do not collide with a stationary robot. If the human enters a neighbourhood of the robot, then the robot stops and waits for the human to leave before resuming its path following. If the human is distracted or inattentive, then the human may unintentionally collide with the robot that has stopped. Our evasive maneuver deals with these worst-case scenarios and guarantees the safety of the robot.

VII. Conclusion and Future Work

We studied a collision avoidance problem for a robot to avoid a static obstacle and an adversarial human-like agent. We developed an event-triggered controller based on the ideas in [11] to guarantee the robot's safety while attempting to delay the evasive maneuver as long as possible. Simulations demonstrate the advantages of our strategy compared to other evasion methods.

A future directions are to solve the problem when the robot's path is not a line and to formally show that the proposed algorithm indeed maximally delays the evasive maneuver.

References

[1] J. Liu, P. Jayakumar, J. L. Stein, and T. Ersal, "A study on model fidelity for model predictive control-based obstacle avoidance in high-speed autonomous ground vehicles," *Vehicle System Dynamics*, vol. 54, no. 11, pp. 1629–1650, 2016.

[2] V. Turri, A. Carvalho, H. E. Tseng, K. H. Johansson, and F. Borrelli, "Linear model predictive control for lane keeping and obstacle avoidance on low curvature roads," in *IEEE conference on intelligent transportation systems (ITSC)*. IEEE, 2013, pp. 378–383.

[3] R. Isaacs, *Differential Games: A Mathematical Theory with Applications to Warfare and Pursuit, Control and Optimization*. Wiley, 1957.

[4] A. Merz, "The game of two identical cars," *Journal of Optimization Theory and Applications*, vol. 9, no. 5, pp. 324–343, 1972.

[5] A. W. Merz, *The Homicidal Chauffeur–a differential game*. Stanford University, 1971.

[6] K. Leung, E. Schmerling, M. Chen, J. Talbot, J. C. Gerdes, and M. Pavone, "On infusing reachability-based safety assurance within probabilistic planning frameworks for human-robot vehicle interactions," in *International Symposium on Experimental Robotics*. Springer, 2018, pp. 561–574.

[7] W. L. Scott and N. E. Leonard, "Optimal evasive strategies for multiple interacting agents with motion constraints," *Automatica*, vol. 94, pp. 26–34, 2018.

[8] A. Bajcsy, S. Bansal, E. Bronstein, V. Tolani, and C. J. Tomlin, "An efficient reachability-based framework for provably safe autonomous navigation in unknown environments," in *IEEE Conference on Decision and Control (CDC)*, 2019, pp. 1758–1765.

[9] S. L. Herbert, S. Bansal, S. Ghosh, and C. J. Tomlin, "Reachability-based safety guarantees using efficient initializations," in *IEEE Conference on Decision and Control (CDC)*, 2019, pp. 4810–4816.

[10] I. M. Mitchell, A. M. Bayen, and C. J. Tomlin, "A time-dependent Hamilton-Jacobi formulation of reachable sets for continuous dynamic games," *IEEE Transactions on Automatic Control*, vol. 50, no. 7, pp. 947–957, 2005.

[11] I. Exarchos, P. Tsiotras, and M. Pachter, "On the suicidal pedestrian differential game," *Dynamic Games and Applications*, vol. 5, no. 3, pp. 297–317, 2015.

[12] N. Karnad and V. Isler, "Lion and man game in the presence of a circular obstacle," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009, pp. 5045–5050.

[13] D. W. Oyler, P. T. Kabamba, and A. R. Girard, "Pursuit–evasion games in the presence of obstacles," *Automatica*, vol. 65, pp. 1–11, 2016.

[14] L. E. Dubins, "On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents," *American Journal of mathematics*, vol. 79, no. 3, pp. 497–516, 1957.

[15] S. D. Bopardikar, S. L. Smith, F. Bullo, and J. P. Hespanha, "Dynamic vehicle routing for translating demands: Stability analysis and receding-horizon policies," *IEEE Transactions on Automatic Control*, vol. 55, no. 11, pp. 2554–2569, 2010.

[16] Y. F. Wang, "A pursuit evasion game approach to obstacle avoidance," Master's thesis, University of Waterloo, 2021.

[17] R. C. Coulter, "Implementation of the pure pursuit path tracking algorithm," Carnegie-Mellon UNIV Pittsburgh PA Robotics INST, Tech. Rep., 1992.

[18] A. Lampe and R. Chatila, "Performance measure for the evaluation of mobile robot autonomy," in *IEEE International Conference on Robotics and Automation*, 2006, pp. 4057–4062.

[19] P. Fiorini and Z. Shiller, "Motion planning in dynamic environments using velocity obstacles," *The International Journal of Robotics Research*, vol. 17, no. 7, pp. 760–772, 1998.

[20] T. Jacobs and G. S. Virk, "ISO 13482-the new safety standard for personal care robots," in *VDE International Symposium on Robotics*, 2014, pp. 1–6.

[21] K. Macek, D. A. V. Govea, T. Fraichard, and R. Siegwart, "Towards safe vehicle navigation in dynamic urban scenarios," *Automatika–Journal for Control, Measurement, Electronics, Computing and Communications*, 2009.