A Pursuit Evasion Game Approach to Obstacle Avoidance

by

Yi Feng Wang

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Electrical and Computer Engineering

Waterloo, Ontario, Canada, 2021

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

We propose an event triggered obstacle avoidance control scheme that guarantees the safety of a mobile robot in a material transport task at a warehouse. This collision avoidance strategy is based on a *pursuer static obstacle* (P.S.O.) game which is a modification of the reversed homicidal chauffeur game. In the PSO game, the evader represents the mobile robot and is modelled by a Dubins vehicle while the pursuer represents the human working at the warehouse and is modelled as an omnidirectional agent. Additionally, we consider the presence of a static obstacle in the work space. Based on our analysis, the evader has two options to avoid both the pursuer and the obstacle. The first option is to turn hard in the opposite direction of both the pursuer and the obstacle. The second option is to go in between the obstacle and the pursuer. The work of the reversed homicidal chauffeur game shows that the first evasive strategy guarantees the mobile robot's safety. However, a clear advantage of implementing the second evasive strategy is that the mobile robot can delay its evasive maneuver and hence the mobile robot can follow its predetermined path for a longer period of time. Therefore, the mobile robot actively checks the feasibility of executing the second evasive strategy rather than executing the first evasive strategy right away.

Based on a numerical study of our collision avoidance strategy, we argue that both the task duration and the path error due to the evasive action are reduced when compared to simply turning hard in the opposite direction of both the pursuer and the obstacle. Our event triggered obstacle avoidance controller is also compared to strategies based on ISO 13482 and the solution of the velocity obstacle set which are commonly used in the current robotic industry. Based on this comparison, we argue that there is a clear trade-off between the reduction in duration and path error and the guarantee of safety during a material transport task.

Acknowledgements

First I would like to thank my supervisors Professor Stephen L. Smith and Professor Christopher Nielsen for their genius insight, constant support and patient guidance. I have learned a lot working and studying under you not only in the field of my research but also ways to conduct research and problem solving skills. Your dedication and work ethics have really inspired me.

I am thankful for the amazing research and learning opportunity provided and facilitated by the outstanding community at the University of Waterloo. I would also like to thank my MASc. thesis readers Professor William Melek and Professor Seyed Majid Zahedi.

I would also like to thank all my colleagues and friends. Your help and support have been immensely valuable. Let us keep in touch and really look forward working with you again. For the past year, it has been challenging with the COVID-19 pandemic. So again I want to thank all the hard working staff at the University of Waterloo to provide all the care and support especially during this challenging time.

Dedication

This thesis is dedicated to my beloved family for your support and encouragement.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

With advances in technologies such as computation, perception and control, autonomous systems such as robots and self-driving vehicles have become the new players in many industries including manufacturing and logistics. One phenomenon is the massive deployment of many mobile robots for delivery tasks in urban logistics [51] and in warehouses [14]. A benefit of using mobile robots is a reduction of labour costs. Also, the deployment of mobile robots can help eliminate simple repetitive and weight lifting tasks in industry that may result in worker injury. However, due to the cost of converting old warehouses into fully autonomous warehouses, many logistics companies may look into the option of partially automating their workforce. Therefore, the mobile robots that operate in warehouses may work alongside humans in the same environment. Motivated by this context, we consider the problem of controlling a mobile robot that is deployed alongside human workers.

For a mobile robot to operate autonomously, its decision making system is commonly hierarchically organized into four levels: route planning, behavioral decision making, motion planning, and control [40]. One of the fundamental challenges for mobile robot operation is collision avoidance because of unpredictable changes in its surrounding environment. Even though the motion planning level considers the presence of permanent structures such as walls and racks when formulating the reference path, the controller often has the additional task of collision avoidance when encountering dynamic obstacles. This thesis proposes a real-time controller that guarantees safe operation of the mobile robot in an environment modelling a warehouse, illustrated in Figure 1.1[1]. The control algorithm guarantees the safety of the mobile robot by maintaining a minimum separation from static and moving

[1]Modified and retrieved from https://www.cisco-eagle.com. Copyright 2014 by Cisco-Eagle.

obstacles under suitable assumptions on the robot's environment. The second objective is to explore the possibility of staying in path following mode for as long as possible.

Figure 1.1: *The mobile robot operates inside a warehouse.*

1.1 Literature Review

In this section, some well known methods for collision avoidance are reviewed.

1.1.1 Model Predictive Control

Model predictive control (MPC) solves a motion planning problem over a short time horizon while satisfying a set of constraints. After solving an optimization problem at each moment, a sequence of control signals are obtained. Usually, the first control action is applied to the system and the rest are discarded. When running in real-time, the above process repeats until the mobile robot reaches its destination. State constraints are usually used for obstacle avoidance. The state constraints prevent the controller from considering controls that can result in collisions.

In [49], the author first derives a linear time-varying model for a mobile robot with bicycle kinematics when operating in low curvature paths, such as highways. The author then uses this model to formulate a linear MPC problem for obstacle avoidance. A benefit of using linear MPC is its ability to reduce the real-time computational complexity when compared to running MPC with a nonlinear model. In [52] and [29], the objective is to develop an optimal tracking controller and the obstacles are treated as constraints in solving the optimization problem. The sensor on board the mobile robot is assumed to have

a limited sensing range and can detect the position of the obstacles near its current location. This information is incorporated online in a nonlinear model predictive framework and the problem is solved in real-time at each instant. The procedure produces a collision-free trajectory up to a certain time window at each sampling instant for the controller to track.

In MPC based approaches, the online constrained optimization problem must be solved at each sampling instant. Overall, the computational power of the mobile robot must be high. Therefore, the main disadvantage of using an online nonlinear MPC is that it is computationally expensive and cannot be easily implemented on real world systems. Even though linear MPC requires much less computational power, it also has a prominent drawback. The linear MPC used in [49] employs on approximation of the nonlinear system and may be prone to errors that can potentially cause collisions in extreme scenarios.

1.1.2 Deep Reinforcement Learning

The reinforcement learning process asks the mobile robot to maximize the cumulative reward based on an objective function [32] [11]. The work in [32] and [11] use reinforcement learning method for collision avoidance and [10] is a modification to the work in [11] which allows the robot to consider social norms in avoiding pedestrians in its proximity. When using reinforcement learning to train the controller to perform obstacle avoidance, the obstacles are treated as constraints in the maximization of the objective function. A clear disadvantage of using the reinforcement learning is the need for a long and tedious offline training process (often the training process is completed over many hours). Also, it is extremely difficult to mathematically prove guaranteed safety of the mobile robot under all circumstances.

1.1.3 Velocity Obstacle Set

In [18], the authors first compute a set which is called the velocity obstacle for each of the moving agents and each agent's trajectory must avoid the combined set of other agents' sets of velocity obstacle. The velocity obstacle is the set of velocities for the mobile robot that will result in a collision between the robot and other agent within a period. A clear advantage of using the collision avoidance strategy based on the velocity obstacle is that the mobile robot can operate closely to other agents if it stays in the vicinity of the predetermined path. Then the mobile robot does not need to put too much effort to come back to the predetermined path and to continue its path following after finishing evading the other agent.

1.1.4 Standard Safety Field

In [22], the evasive strategy based on the ISO 13482 uses a standard safety field where the robot is fixed at the center and a circular space is formed around the robot which is the *stop zone*. If the pursuer is in the stop zone, then the evader stops and waits until the pursuer leaves the stop zone. Commonly, there is an outer ring space which defines a set labelled as the *slow zone* where the robot slows down if the human agent is inside. The guaranteed safety of the robot is based on the idea that no collision can occur before the robot stops and this idea is known as *passive safety* [31]. The work in [30] also uses a modified safety field idea to stop the robot before the other agent can get to its restricted proximity.

1.1.5 Social Navigation Model

In [47], the authors propose a collision avoidance method between humans and a robot based on social navigation model which uses different artificial forces to influence the robot's response when facing humans in its surrounding. The creation of these artificial forces depend on the influences of the humans in the environment of operation and the goal that the robot wants to achieve. To quantify the human influences to the robot, it is critical to first estimate the human motion and then decide how the human motion affects the robot to reach its goal. The human motion prediction depends on the face orientation and body pose of the human perceived by the robot [48]. Also, the prediction in [47] considers the concept of personal space [19]. The personal space is a set of space which prevents the uneasiness of the humans when operating close to a robot in their environment of operation. Different types of personal space can be constructed based on the different scenarios [24]. Based on the predicted human motion, the set of human trajectories is classified into the avoiding, unavoiding, and approaching subsets. After classifying the human motions, the social navigation model allows the robot to smoothly respond to human motions including the evasion of possible collision with the humans.

1.1.6 Game Theory

Since our obstacle avoidance control scheme is based on a pursuit evasion game, a topic in game theory, we first review some studies in pursuit evasion games before discussing the use of game theory for obstacle avoidance. In the mathematical theory of games, players with different motives operate in a game theater and one studies a mathematical model

of the interactions between players. The outcome of solving a game is to identify every player's strategy. A pursuit evasion game, a particular type of game, involves a group of players, known as the pursuers, trying to capture the players from another group, known as the evaders (see Chapter 2 for more details). Pursuit evasion games were first reported by Isaacs in [21] and can be divided into two main categories: *Game of kind* and *Game of Degree*.

In a game of kind, one is interested to find whether or not one of the players can achieve its objective under optimal play with no constraints in the game. The three popular games of kind include the *game of two identical cars* [35] [16], the *homicidal chauffeur* [34] [36] [21] [43] [42], and the *reversed homicidal chauffeur* [15]. In [35], the pursuer and the evader are both Dubins vehicles with different minimum turning radii and speeds. The author shows that the optimal strategies for both the pursuer and the evader can be determined as functions of the three-dimensional state. The three-dimensional state includes changes in their respective positions and headings. In [16], the authors utilize the idea of reachable set, a key notion in mathematical control theory, in [23] [26] and compute the set of initial states where the pursuer can capture the evader regardless of the evader's subsequent actions. Hence, the objective of the pursuer is to reach the reachable set and the objective of the evader is to prevent the pursuer from accessing the reachable set.

The homicidal chauffeur game was first proposed in Isaacs work [21] where a Dubins vehicle tries to capture an evader that is omnidirectional. To make the problem meaningful, the evader's speed must be less or equal to the pursuer's. In [21], Isaacs computes the boundary that separates the game space into the capture zone and the escape zone. The pursuer can always achieve capture if the game is initialized in the capture zone. If the game is initialized in the escape zone, then the evader can avoid being captured under optimal play. Hence, one can determine whether a homicidal chauffeur game results in the capture or escape of the evader under optimal play by only evaluating the initial states of the evader and the pursuer. In [43], [42], and [41], the authors obtain the numerical solutions of computing the capture zone in the homicidal chauffeur game by computing level sets of the value function. The level sets illustrate the entire boundary of the capture zone at different time instants in the game and also illustrates how the capture zone grows as time passes. In the reversed homicidal chauffeur game [15], the pursuer is omnidirectional and the evader is the Dubins vehicle. The authors compute the capture zone and the escape zone between the two players based on the solutions in the homicidal chauffeur game [21]. The boundary between the capture zone and the escape zone can be used by the mobile robot to avoid a human if the human arrives on the boundary.

In a game of degree, the cost function has a running cost and one might be interested to find whether or not one of the players can achieve its objective in minimum time. In [46],

the authors find the saddle point for the cost function and use it as the optimal strategy to play the game of degree for a pursuit evasion game. In the *surveillance evasion game* [28], the pursuer tries to keep the evader within a specified detection zone for as long as possible and the evader tries to escape the detection zone for as soon as possible. The author utilizes the methods in differential geometry to formulate different sets of optimal strategies for both the evader and the pursuer depending on the configurations between players in the game. In the *air combat game of guided missile* [45], the pursuer is a missile and the evader is a targeted aircraft. The capture set is obtained by constructing saddle point trajectories on its boundary. The computation of these trajectories considers the limited fuel that the missile has. The optimal control problems for computing the trajectories are solved iteratively using discretization and nonlinear programming, until the saddle point has been found. Often, collision avoidance schemes based on using the concepts from the pursuit evasion game can be mathematically proved to guarantee safety because it can allow the mobile robot to avoid the other agent that uses its optimal strategy to collide with the mobile robot.

1.1.7 Backward Reachable Set

In [27], the authors treat the human-driven vehicle as the pursuer and the autonomous vehicle as the robot. The authors first compute the *backward reachable set* (BRS) using *Hamilton-Jacobi* (HJ) reachability analysis [9]. The BRS is the set of states from which there does not exist a strategy for the robot to prevent its collision with the pursuer, assuming that the pursuer executes its optimal strategy (worst cast scenario) for some time t into the future. Compared to other methods of computing the BRS, HJ reachability is able to compute the exact BRS for nonlinear dynamics with control and disturbance inputs. However, it is the most computationally expensive and the computation time increases exponentially as the dimension of the system increases. Therefore, the BRS is usually computed offline and is cached via a look-up table in the robot's memory [27].

To reduce the dimension of the system, HJ reachability formulation uses the system relative dynamics. Then, a value function is defined and it obeys the *Hamilton-Jacobi-Isaacs partial differential equation* (HJI PDE) for the overall robot and pursuer system [37]. The BRS is the set of system states where the value function is less or equal to zero. To obtain the value function, HJI PDE is solved starting from the boundary condition of the value function where $t = 0$ and the tool used in this computation is *BEACLS* toolkit [4]. If the system is outside the BRS where the value function is greater than zero, there at least exists a robot policy that keeps the robot safe over a future time period $|t|$ regardless of the pursuer's strategy. In [17] and [3], the evasion strategy is to switch to the optimal

control when the system is near the boundary of the BRS. The optimal control offers the greatest increase of the value function, assuming the worst case scenario. In [27], the MPC controller generates a guaranteed collision free trajectory by considering the BRS as constraints. However, the BRS constraints are considered whenever the system state approaches the boundary of the BRS. Therefore, the robot has the appropriate sensors to continuously check the change of the pursuer's state.

1.2 Contributions

In Chapter 3, we setup a new kind of pursuit evasion game known as the pursuer static obstacle game (PSO) based on the reversed homicidal chauffeur game [15]. Unlike the reversed homicidal chauffeur game, the PSO game allows for the presence of a static obstacle that can hinder the evader's ability to avoid the pursuer. Then we introduce the notion of the default escape route and the idea of the guaranteed evasion where the mobile robot is guaranteed to be safe if its selected evasive strategy can achieve the configuration between all players defined by guaranteed evasion before being captured by the pursuer or colliding into the obstacle. Next, we identify all events that can occur during the PSO game and we present the evader's corresponding response to guarantee mobile robot's safety when encountering each event. Based on these events and their corresponding evader's reactions, an event triggered controller is presented and the proof to guarantee the robot's safety during a simple material transport task is also provided. The mobile robot follows the nominal path while constantly monitoring the occurrence of any of the events in the event triggered controller and it reacts with the corresponding response if an event occurs. Also in Chapter 3, we present an evaluation mechanism to check whether or not the moment of evaluation is the last moment that a feasible evasive action exists. The definition of a feasible evasive action is also provided in Chapter 3. Our evasive strategy wants to allow the mobile robot to stay on the given path for as long as possible until there exists no future feasible evasive action.

In Chapter 4, we conducted a simulation of a simple material transport task using the event triggered controller and analyzed the its performance. The performance indicators include the task duration and the path error due to the evader's execution of an evasive maneuver. In general, there are reductions in both the task duration and the path error for executing the event triggered controller when compared to executing the simple control strategy. Also in Chapter 4, we compare the performance of the event triggered controller with the collision avoidance strategy based on using the velocity obstacle [18]. In this comparison, there is a clear trade-off between achieving guaranteed safety and being able

to operate closely to human in terms of task duration and path error. If the human is cooperative, then the collision avoidance strategy based on using the velocity obstacle results in a less task duration and a lower path error. However, if the human is not cooperative, the mobile robot may collide with the human if using the collision avoidance strategy based on using the velocity obstacle for a material transport task.

1.3 Organization

A detailed review of the fundamental concepts for solving a pursuit evasion game, in particular for a reversed homicidal chauffeur game, is presented in Chapter 2. Next, Chapter 3 discusses the *pursuer static obstacle game*, denoted by *PSO* game; the mathematical concepts for solving the PSO game; and the collision avoidance algorithm based on the solution of the PSO game. The contents in Chapter 4 show the simulation results of our evasive policy and the comparison of our method with some other common methods in the field of mobile robot's collision avoidance. Finally, the conclusions and the potential future work are discussed in Chapter 5. In the appendix, detailed proofs and derivations for concepts discussed in Chapter 3 are provided.

Chapter 2

Pursuit Evasion Games

Research on the theory of games started with the seminal work on mixed-strategy equilibria in zero-sum games by J. Von Neumann and O. Morgenstern [38]. Based on the theory in [38], R. Isaacs and A. Mertz introduced the idea of *pursuit evasion games*. According to Isaacs [21], in a pursuit evasion game there are at least two opposing players; one is called the pursuer and the other is called the evader. All players are in the same environment and are assumed to be rational, which means that the two players will play the game optimally based on their respective objectives. The game terminates when the evader is captured, i.e., the distance between the pursuer and evader is less than a pre-specified value or the game terminates when escape is assured, i.e., the evader has a strategy that guarantees evasion. Our research utilizes the pursuit evasion game framework to solve a collision avoidance problem for a mobile robot operating in the plane which models the floor of a warehouse. This chapter introduces some fundamental concepts for a two player pursuit evasion game.

2.1 Fundamental Concepts

In this section we review a few concepts from a two player pursuit evasion game from a general point of view.

2.1.1 Kinematics and the Reduced Space

The evolution of a player's motion is governed by its kinematic equation which is typically a state-space model. The state-space model for the pursuer is

$$\dot{\boldsymbol{x}}_p(t) = f_p(\boldsymbol{x}_p(t), \boldsymbol{u}_p(t))$$

where $\boldsymbol{x}_p$ is the pursuer's state vector and $\boldsymbol{u}_p$ is its control input. The position of the pursuer is written

$$\boldsymbol{y}_p(t) = h_p(\boldsymbol{x}_p(t)).$$

The state-space model for the evader is denoted

$$\dot{\boldsymbol{x}}_e(t) = f_e(\boldsymbol{x}_e(t), \boldsymbol{u}_e(t)),$$

where $\boldsymbol{x}_e$ is the state vector $\boldsymbol{u}_e$ is its control input. The position of the evader is

$$\boldsymbol{y}_e(t) = h_e(\boldsymbol{x}_e(t)).$$

Following Issacs' notation in [21], the set in which $\boldsymbol{y}_p$ and $\boldsymbol{y}_e$ take values is called the *game theatre* and is denoted by $\mathcal{E}$. In many of the classical games the game theatre is a subset of the plane, i.e., $\mathcal{E} \subseteq \mathbb{R}^2$. The combined vector $(\boldsymbol{x}_e, \boldsymbol{x}_p)$ is called the state of the game, the set in which the state takes values is called the *realistic space*, and

$$\begin{bmatrix} \dot{\boldsymbol{x}}_e(t) \\ \dot{\boldsymbol{x}}_p(t) \end{bmatrix} = \begin{bmatrix} f_e(\boldsymbol{x}_e(t), \boldsymbol{u}_e(t)) \\ f_p(\boldsymbol{x}_p(t), \boldsymbol{u}_p(t)) \end{bmatrix}$$

is called the kinematic equation.

In [21], Issacs argues that it is unnecessary to use such a high dimensional state-space to describe the evolution of the game. By expressing the kinematic equations in a body fixed coordinate system attached to either player, the dimension of the game's state-space can be reduced. This lower dimensional state-space for modelling the evolution of the game is known as the *reduced space*. In addition to reducing the dimension of the state-space, an advantage of working in the reduced game space is in geometric visualization, which can be helpful for analyzing the game and formulating well-defined strategies for its players. In a two player game we will denote by $\boldsymbol{x}$ (with no subscripts) the state of the game in the reduced space and we write the reduced kinematics as

$$\boldsymbol{x}(t) = f(\boldsymbol{x}(t), \boldsymbol{u}_e(t), \boldsymbol{u}_p(t)). \tag{2.1}$$

Our convention is that, without loss of generality, the reduced state is expressed with respect to a body fixed frame attached to the evader and that $\boldsymbol{p}$ denotes the pursuer's relative position in this frame.

2.1.2 Termination of the Game and the Terminal Surface

In a two player pursuit evasion game, the pursuer *captures* the evader if the Euclidean distance between the agents is less than or equal to a pre-defined quantity $c > 0$. The constant c takes into consideration the size and the shape of both players.

Definition 1. *Fix a positive constant $c > 0$. Consider a configuration of the pursuer and evader with associated positions $\boldsymbol{y}_p$ and $\boldsymbol{y}_e$. The evader has been **captured** (with capture radius c) if*

$$\|\boldsymbol{y}_e - \boldsymbol{y}_p\|_2 < c.$$

In the reduced space the set $\mathcal{S} = \{\boldsymbol{p} \in \mathcal{E} : \|\boldsymbol{p}\|_2 = c\}$ is called the *terminal surface*. The terminal surface separates the game theatre into two sets; the *collision space* $\mathcal{C} = \{\boldsymbol{p} : \|\boldsymbol{p}\|_2 < c\}$ and the *game space* $\mathcal{G} = \{\boldsymbol{p} : \|\boldsymbol{p}\|_2 \geq c\}$. In the special case where the game theatre is the plane, the terminal surface is a circle of radius c centered at the origin while the collision space is the interior of this circle.

2.1.3 Important Concepts used to Solve Pursuit Evasion Games

Two common types of two player pursuit evasion games are the *game of degree* and the *game of kind* (see Section 1.1.6). In either type, the general form of the game's payoff is

$$J(\boldsymbol{x}(0), \boldsymbol{u}_e, \boldsymbol{u}_p) = \int_0^{t_f} G(\boldsymbol{x}(t), \boldsymbol{u}_e, \boldsymbol{u}_p)\,\mathrm{d}t + K(\boldsymbol{x}(t_f)),$$

where t_f is the termination time. The value of the payoff J is the quantity that the evader strives to maximize while the pursuer strives to minimize. In the game of degree, the quantity to be optimized is often the termination time, i.e., the time that it takes the pursuer to capture the evader or the time that it takes the evader to escape. However, in a game of kind, the integral payoff (the running reward/cost) is often zero and the terminal payoff $K(\boldsymbol{x}(t_f))$ equals either positive one indicating the evader has escaped or negative one indicating capture.

In [21], Isaacs uses dynamic programming methods with two opposing controls to solve the pursuit evasion game. The solution provides both the value of the game and the optimal strategies for both players. Let $\boldsymbol{u}_p^*$ and $\boldsymbol{u}_e^*$ denote the optimal control strategies for, respectively, the pursuer and the evader. The differential equation (2.1) when both the evader and the pursuer apply their optimal control strategies is referred to as the *path equation* by Issacs [20]. By integrating the path equation, the phase curves corresponding to optimal play can be obtained from a given set of initial conditions to the termination of the game.

The retrogression principle was proposed by Issacs in [21]. It solves the path equation in reverse time for initial conditions on the terminal surface $\mathcal{S}$. The main reason for solving the game backwards in time is because of the availability of data on $\mathcal{S}$; initial conditions on $\mathcal{S}$ represent the end of the game. This allows one to characterize the phase curves that eventually arrive on $\mathcal{S}$. Let $\tau \coloneqq -t$ be the reversed time variable. Then

$$\mathring{\boldsymbol{x}}(\tau) \coloneqq \frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}\tau} = \frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}t}\frac{\mathrm{d}t}{\mathrm{d}\tau} = -\dot{\boldsymbol{x}}(t).$$

The expression for the kinematics of the retrogressive path equations (RPE) which is the kinematics of the path equations written in the retrogressive notation is

$$\mathring{\boldsymbol{x}}(\tau) = -f(\boldsymbol{x}(\tau), \boldsymbol{u}_p^*(\tau), \boldsymbol{u}_e^*(\tau)). \tag{2.2}$$

A subset of the game space is called *semipermeable* if under optimal play by both players and if the game's state is initialized in this subset, then the state remains in this set, i.e., it's a (locally) positively invariant set under optimal play.

2.1.4 Retrogressive Path Equations and the Barrier Surface

Pursuit evasion games such as the homicidal chauffeur in [21] and the reversed homicidal chauffeur in [15] are typical examples of games of kind. Therefore, as mentioned above, the running payoff is taken to be zero and the terminal payoff is

$$K(\boldsymbol{x}(t_f)) = \begin{cases} +1, & \text{escape,} \\ -1, & \text{capture.} \end{cases}$$

Initial conditions on the terminal surface, $\mathcal{S}$ serve as the initial conditions when integrating the retrogressive path equations (2.2). Let $\gamma(\boldsymbol{x})$ be the vector normal to $\mathcal{S}$ at $\boldsymbol{x}$ pointing

towards the game space. According to Isaacs [21], if

$$\min_{\boldsymbol{u}_p} \max_{\boldsymbol{u}_e} \Big(\gamma(\boldsymbol{x})^\top f(\boldsymbol{x}, \boldsymbol{u}_p, \boldsymbol{u}_e) \Big) < 0, \qquad \boldsymbol{x} \in \mathcal{S}, \tag{2.3}$$

then the pursuer can penetrate the terminal surface $\mathcal{S}$ at $\boldsymbol{x}$ regardless of the evader's strategy; the set of points that satisfy (2.3) is called the *usable part* of $\mathcal{S}$. By reversing the inequality in (2.3), one characterizes those points at which there is no strategy for the pursuer to penetrate $\mathcal{S}$ regardless of the evader's strategy; this set is called the *non-usable part* of $\mathcal{S}$. The collection of points separating the above two sets is called the *boundary of the usable part* and is characterized by

$$\min_{\boldsymbol{u}_p} \max_{\boldsymbol{u}_e} \Big(\gamma(\boldsymbol{x})^\top f(\boldsymbol{x}, \boldsymbol{u}_p, \boldsymbol{u}_e) \Big) = 0, \qquad \boldsymbol{x} \in \mathcal{S}.$$

Points on the boundary of the usable part are taken as the initial conditions to (2.2) for constructing the *barrier surface*, denoted by $\mathcal{B}$. The barrier surface $\mathcal{B}$ is a semipermeable surface that separates the game space $\mathcal{G}$ into two open sets. The first open set is known as the collision zone; if the game is initialized in the collision zone, then the pursuer will inevitably capture the pursuer under optimal play. The second open set is known as the evasion zone; if the game is initialized in the evasion zone, then the evader can always avoid capture through optimal play.

If the state is initialized on the barrier surface and both players act optimally, then the pursuer moves tangentially along the barrier surface and the evader avoids capture.

2.2 The Reversed Homicidal Chauffeur Game

The obstacle avoidance algorithm that we have developed builds on the work by I.Exarchos *et al.* [15]. In [15], the authors study a pursuit evasion game that reverses the roles of the players in the classical homicidal chauffeur game. In the reversed homicidal chauffeur game, the evader's kinematic model is a Dubins vehicle with kinematic equations

$$\begin{aligned} \dot{x}_e(t) &= v_e \cos\left(\phi_e(t)\right) \\ \dot{y}_e(t) &= v_e \sin\left(\phi_e(t)\right) \\ \dot{\phi}_e(t) &= -\frac{v_e}{R} u_e(t), \qquad |u_e(t)| \leq 1, \end{aligned} \tag{2.4}$$

where $v_e > 0$ and $R > 0$ are, respectively, the speed and the minimum turning radius for the evader. Let

$$\boldsymbol{x}_e(t) \coloneqq \begin{bmatrix} x_e(t) \\ y_e(t) \\ \phi_e(t) \end{bmatrix} \quad \text{and} \quad \boldsymbol{y}_e(t) \coloneqq \begin{bmatrix} x_e(t) \\ y_e(t) \end{bmatrix}$$

denote, respectively, the state and the position of the evader at time t in the realistic space. In the reversed homicidal chauffeur game, the pursuer is the more agile player whose kinematic equations are given by

$$\begin{aligned} \dot{x}_p(t) &= v_p \cos\left(u_p(t)\right) \\ \dot{y}_p(t) &= v_p \sin\left(u_p(t)\right). \end{aligned} \tag{2.5}$$

The pursuer can instantaneously turn in any direction and $v_p > 0$ is a fixed constant. Let

$$\boldsymbol{x}_p(t) \coloneqq \begin{bmatrix} x_p(t) \\ y_p(t) \end{bmatrix} \quad \text{and} \quad \boldsymbol{y}_p(t) \coloneqq \boldsymbol{x}_p(t)$$

denote, respectively, the state and the position of the pursuer at time t in the realistic space. We assume throughout this thesis that the evader is faster than the pursuer, i.e., $v_e > v_p$. Illustrated in Figure 2.1, the evader turns in the counter clockwise direction when $u_e(t) < 0$ and turns in the clockwise direction when $u_e(t) > 0$.

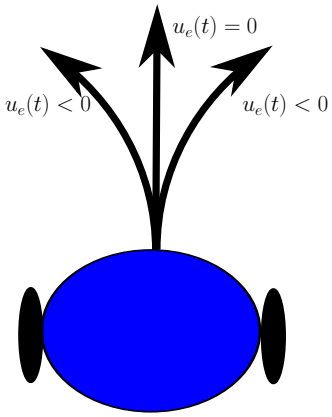


Figure 2.1: *If $u_e(t) < 0$, then the evader turns to the left with respect to its heading. If $u_e(t) > 0$, then the evader turns to the right.*

From (2.4) and (2.5), we see that the state-space of the game, in the realistic space, is 5-dimensional. To obtain a reduced order model, attach a moving frame $\{\boldsymbol{y}_e(t), \{\boldsymbol{b}_x, \boldsymbol{b}_y\}\}$

to the evader whose origin is $\boldsymbol{y}_e(t)$ and where $\boldsymbol{b}_y$ is the unit vector pointing in the heading direction of the evader while $\boldsymbol{b}_x$ is a rotation of $\boldsymbol{b}_y$ by $-\pi/2$. In the reduced space, the origin represents the current position of the evader and the equations of motion of the pursuer are given by

$$\begin{aligned} \dot{x}(t) &= -\frac{v_e}{R} y(t) u(t) - v_p \sin(\phi(t)) \\ \dot{y}(t) &= \frac{v_e}{R} x(t) u(t) - v_e - v_p \cos(\phi(t)) \end{aligned} \tag{2.6}$$

where $u \coloneqq u_e$ and $\phi \coloneqq \pi - u_p + \phi_e$ represents the pursuer's relative heading in this new reference frame. In the reduced space the pursuer's relative position is simply

$$\boldsymbol{p} \coloneqq \begin{bmatrix} x \\ y \end{bmatrix}.$$

Since the game theatre is the plane, the terminal surface is a circle of radius c centered at the origin in the reduced space

$$\mathcal{S} \coloneqq \{\boldsymbol{p} \in \mathbb{R}^2 : \|\boldsymbol{p}\|_2 = c\} \tag{2.7}$$

and the collision space $\mathcal{C}$ is its interior; observe that the terminal surface (2.7) can be expressed as the image of a parameterized curve

$$s \mapsto c \begin{bmatrix} \sin(s) \\ \cos(s) \end{bmatrix}.$$

In [15], the authors compute the boundary of the usable part for the reversed homicidal chauffeur game to be the finite set consisting of just two points on $\mathcal{S}$

$$c \begin{bmatrix} \sin(\bar{s}) \\ \cos(\bar{s}) \end{bmatrix} \quad \text{and} \quad c \begin{bmatrix} -\sin(\bar{s}) \\ \cos(\bar{s}) \end{bmatrix}, \quad \text{where} \quad \bar{s} = \arccos\left(-\frac{v_p}{v_e}\right). \tag{2.8}$$

The usable part of $\mathcal{S}$ is then

$$\left\{ c \begin{bmatrix} \sin(s) \\ \cos(s) \end{bmatrix} : |s| < \bar{s} \right\}$$

and the non-usable part of $\mathcal{S}$ is

$$\left\{ c \begin{bmatrix} \sin(s) \\ \cos(s) \end{bmatrix} : \bar{s} < |s| \leq \pi \right\}.$$

In [15], the authors assume that the pursuer is on the right side with respect to the y-axis in the reduced space and it is shown that the barrier surface can be expressed in terms of the parameterized curve

$$\begin{aligned} x(\tau) &= -R + R\cos\left(\frac{v_e}{R}\tau\right) + (c + v_p\tau)\sin\left(\bar{s} - \frac{v_e}{R}\tau\right) \\ y(\tau) &= R\sin\left(\frac{v_e}{R}\tau\right) + (c + v_p\tau)\cos\left(\bar{s} - \frac{v_e}{R}\tau\right) \end{aligned} \tag{2.9}$$

where the parameter belongs to the interval $\tau \in [0, \overline{\tau}]$ and $\overline{\tau}$ is the solution to the transcendental equation

$$(c + v_p\overline{\tau})\sin\left(\bar{s} - \frac{v_e}{R}\overline{\tau}\right) = R - R\cos\left(\frac{v_e}{R}\overline{\tau}\right). \tag{2.10}$$

Since the game is symmetric with respect to the y-axis in the reduced space, the left side barrier surface is simply the mirror image of (2.9) along the y-axis. The barrier surface is the set

$$\mathcal{B} \coloneqq \left\{ \begin{bmatrix} x(\tau) \\ y(\tau) \end{bmatrix} \in \mathbb{R}^2 : \tau \in [0, \overline{\tau}] \right\} \cup \left\{ \begin{bmatrix} -x(\tau) \\ y(\tau) \end{bmatrix} \in \mathbb{R}^2 : \tau \in [0, \overline{\tau}] \right\} \tag{2.11}$$

and is depicted in Figure 2.2. If $\boldsymbol{p} \in \mathcal{B}$, it is useful to let $\tau(\boldsymbol{p})$ denote the value of the parameter that when plugged into[1] (2.9) returns the point $\boldsymbol{p}$.

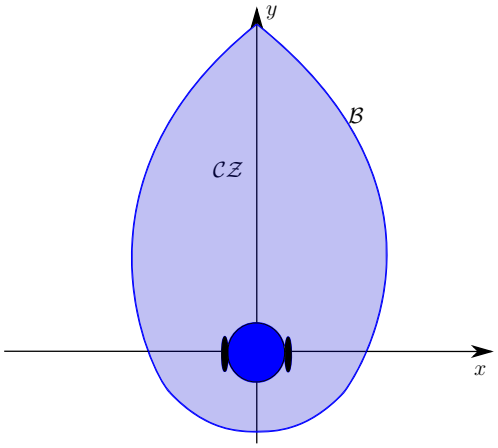


Figure 2.2: *The collision zone $\mathcal{CZ}$ is shaded with the space strictly inside the blue circle as the collision space $\mathcal{C}$.*

[1] Or (2.9) with $x(\tau)$ replaced with $-x(\tau)$.

The set

$$\mathcal{B} \cup \left\{ c \begin{bmatrix} \sin(s) \\ \cos(s) \end{bmatrix} : \bar{s} \leq |s| \leq \pi \right\}$$

is a closed curve in the plane consisting of the union of the barrier surface, the nonusable part of the terminal surface and the boundary of the usable part. The interior of this closed curve is the *collision zone* and is denoted $\mathcal{CZ}$; it is illustrated in Figure 2.2. If $\boldsymbol{p}$ lies in the collision zone, then the evader has either already been captured or capture is unavoidable.

The evader's solution to the reversed homicidal chauffeur game as presented in [15] is to apply maximum steering away from the pursuer when the pursuer arrives on the barrier surface. By doing so, the pursuer can never penetrate the barrier surface; its optimal strategy results in it sliding along the barrier surface.

Remark 1. *If the pursuer and evader move at the same speed, then the entire terminal surface $\mathcal{S}$ except for the point directly behind the evader is usuable [15]. If the pursuer is faster than the evader, then capture is unavoidable.*

Chapter 3

Pursuer Static Obstacle Game

In this chapter we use the results in Section 2.2 to consider a reversed homicidal chauffeur game with a static obstacle and this is known as the *pursuer static obstacle* (PSO) game. Motivated by the theme of autonomous material transport at a warehouse, illustrated in Figure 1.1, the evader in the PSO game is a mobile robot whose nominal behaviour is to follow a pre-determined path in the plane which models the floor of a warehouse. In our formulation of the PSO game, the mobile robot encounters a single human agent whose intention is unknown and is therefore treated as being adversarial. Unlike in the reversed homicidal chauffeur game, a static obstacle is present which can interfere with the evasion strategy. We seek an evasion policy for the mobile robot that guarantees that there are no collisions while trying to keep the mobile robot on its nominal path for as long as possible.

3.1 Problem Formulation

The evader's (mobile robot's) kinematic model is (2.4) while the pursuer's (human's) kinematic model is (2.5). The evader is faster than the pursuer $v_e > v_p$. There is a static obstacle in the game theatre $\mathcal{E} = \mathbb{R}^2$ modelled as a point whose location in the inertial frame is denoted $\boldsymbol{y}_o \in \mathcal{E}$.

Definition 2 (Capture/Collision)**.** *Fix a positive constant $c > 0$. Consider a configuration of the pursuer, evader and obstacle with associated positions $\boldsymbol{y}_p$, $\boldsymbol{y}_e$, and $\boldsymbol{y}_o$. The evader has been **captured** (with capture radius c) if either of the following hold:*

$$\|\boldsymbol{y}_e - \boldsymbol{y}_p\|_2 < c, \qquad \|\boldsymbol{y}_e - \boldsymbol{y}_o\|_2 < c.$$

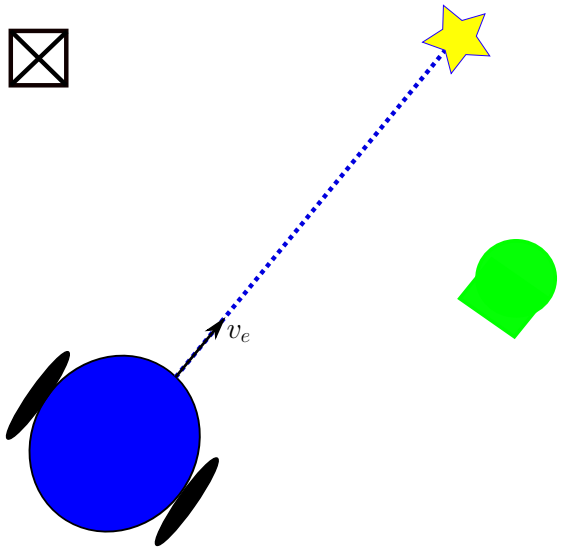


Figure 3.1: *The evader follows a nominal path in a PSO game where the obstacle (crate) and the pursuer (human with a green cap) on either side of the path.*

In Definition 2 we have used the same capture radius for the static obstacle and the pursuer; this simplifies the problem and makes the subsequent exposition easier to follow. From hereon the capture radius c is fixed and satisfies the following assumption.

Assumption 1. *The evader's minimum turning radius is greater then the capture radius,* $R > c$.

In the absence of both the pursuer and the obstacle, the evader is assumed to move along a pre-determined path driven by an associated nominal control signal $u_0 : [0, \infty) \to \mathbb{R}$. The following simplifying assumption is made throughout.

Assumption 2. *The evader's pre-determined path is a line with associated control signal* $u_0(t) = 0$, $t \geq 0$.

Our PSO game is symmetric along the nominal path; for this reason we assume throughout that the pursuer and the obstacle are initially to the right and left of the nominal path respectively. The pursuer knows the location of the static obstacle and may use the obstacle to aid its capture of the evader. Therefore, a viable strategy for the pursuer is to force the evader into a dilemma of either being captured by the pursuer or colliding with the obstacle. We assume the evader has the appropriate sensors to detect the pursuer's and the obstacle's locations.

Under these standing assumptions, we are ready to state the problem considered in this thesis.

Problem 1. *Find an evasion policy for the evader that:*

(i) guarantees that the evader is not captured and,

(ii) maximizes the following quantity

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

Problem 1 asks that the evader stay on its nominal path for as long as possible while still avoiding capture.

Remark 2. *In a warehouse environment, where the role of pursuer is played by an employee, it is unlikely that the pursuer will chase the evader to capture them from behind. For this reason, we will further assume that the pursuer ceases its pursuit if it ever "falls behind" the evader, in a sense to be made precise below.*

3.1.1 Modelling the presence of a static obstacle

The origin of the reduced game space represents the location of the evader. Since the evader moves, the static obstacle appears to move in the reduced game space; the obstacle's kinematic equation in the reduced space is

$$\begin{aligned} \dot{x}_o(t) &= -\frac{v_e}{R} y_o(t) u(t) \\ \dot{y}_o(t) &= \frac{v_e}{R} x_o(t) u(t) - v_e. \end{aligned} \tag{3.1}$$

In the reduced space, the obstacle's relative position is simply:

$$\boldsymbol{o} \coloneqq \begin{bmatrix} x_o \\ y_o \end{bmatrix}$$

and let $\boldsymbol{y}_o$ denote the obstacle's position in the realistic space. The presence of the static obstacle induces sets and surfaces that are analogous to those described in Section 2.2 for the reversed homicidal chauffeur game. In particular the terminal surface associated to the obstacle is the circle:

$$\mathcal{S}_o \coloneqq \{\boldsymbol{o} \in \mathbb{R}^2 : \|\boldsymbol{o}\|_2 = c\}. \tag{3.2}$$

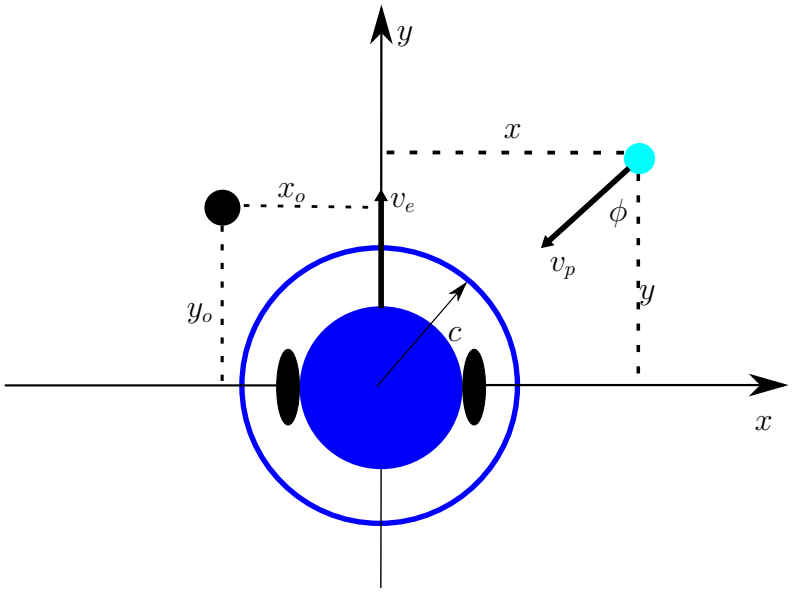


Figure 3.2: *PSO game in the reduced space where the cyan and black dots represent, respectively, the pursuer and the obstacle; the instantaneous ϕ is found in 2.6.*

Clearly, if the obstacle is on the bottom half of the terminal surface,

$$\{\boldsymbol{o} \in \mathcal{S}_o : y_o \leq 0\}$$

then the obstacle can never get inside the collision space. Therefore, the following two points are the boundary of the usable part associated to the obstacle:

$$c \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad \text{and} \quad -c \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

on the terminal surface $\mathcal{S}_o$. The usable part of $\mathcal{S}_o$ equals

$$\left\{ c \begin{bmatrix} \sin(s) \\ \cos(s) \end{bmatrix} : |s| < \frac{\pi}{2} \right\}$$

and the non-usable part of $\mathcal{S}_o$ is

$$\left\{ c \begin{bmatrix} \sin(s) \\ \cos(s) \end{bmatrix} : \frac{\pi}{2} < |s| \leq \pi \right\}.$$

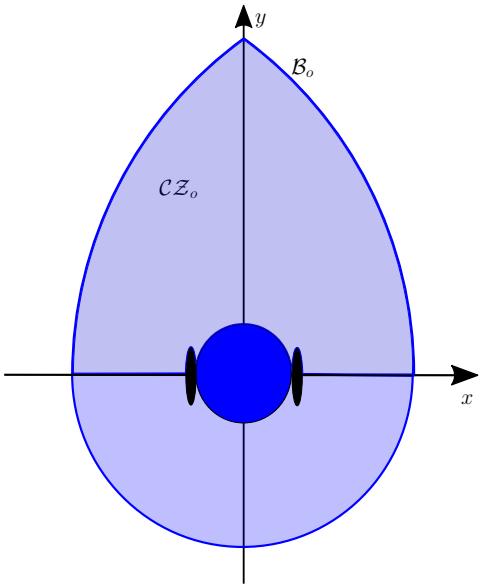


Figure 3.3: *The static collision zone $\mathcal{CZ}_o$ is shaded with the space strictly inside the blue circle as the collision space $\mathcal{C}$.*

Like the barrier surface $\mathcal{B}$ for the reversed homicidal chauffeur game, there is an analogous semipermeable surface associated to the static obstacle. Physically, it represents those points at which the evader must take evasive action in order to avoid colliding with the static obstacle. We call this set the *static barrier surface*. The static barrier surface can be expressed in terms of a parameterized curve

$$\begin{aligned} x_o(\tau_o) &= -R + (c+R)\cos\left(\frac{v_e}{R}\tau_o\right) \\ y_o(\tau_o) &= (c+R)\sin\left(\frac{v_e}{R}\tau_o\right) \end{aligned} \tag{3.3}$$

where the parameter τ_0 belongs to the interval $[0, \overline{\tau}_o]$ and

$$\overline{\tau}_o \coloneqq \frac{R}{v_e}\arccos\left(\frac{R}{c+R}\right). \tag{3.4}$$

Since the game is symmetric with respect to the y-axis in the reduced space, the left side static barrier surface is the reflection of (3.3) along the y-axis. The static barrier surface

is then given by set

$$\mathcal{B}_o \coloneqq \left\{ \begin{bmatrix} x_o(\tau) \\ y_o(\tau) \end{bmatrix} \in \mathbb{R}^2 : \tau \in [0, \overline{\tau}_o] \right\} \cup \left\{ \begin{bmatrix} -x_o(\tau) \\ y_o(\tau) \end{bmatrix} \in \mathbb{R}^2 : \tau \in [0, \overline{\tau}_o] \right\}. \tag{3.5}$$

If $\boldsymbol{o} \in \mathcal{B}_o$, then $\tau_o(\boldsymbol{o})$ denotes the value of the parameter that when plugged into[1] (3.3) returns the point $\boldsymbol{o}$.

The set

$$\mathcal{B}_o \cup \left\{ c \begin{bmatrix} \sin(s) \\ \cos(s) \end{bmatrix} : \frac{\pi}{2} \leq |s| \leq \pi \right\}$$

is a closed curve in the plane consisting of the union of the static barrier surface, the non-usable part of the terminal surface $\mathcal{S}_o$ and the boundary of the usable part. The interior of this closed curve is the *static collision zone* and is denoted $\mathcal{CZ}_o$; it's illustrated in Figure 3.3. If the obstacle enters the static collision zone, then the evader has either already collided with the obstacle or a collision is unavoidable. Similar to the solution of the reversed homicidal chauffeur game, the solution of the game involving only the obstacle is for the evader to apply maximum steering away from the obstacle when the obstacle arrives on the static barrier surface.

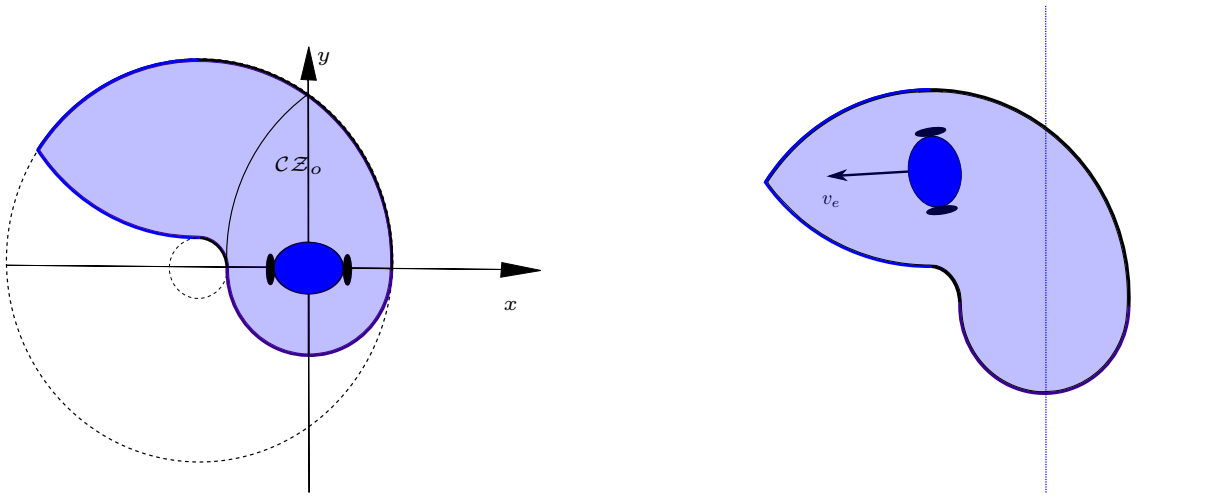
3.1.2 The Default Escape Route

In the reversed homicidal chauffeur game, when the pursuer arrives on the barrier surface, $\boldsymbol{p} \in \mathcal{B}$, the evader must make a hard turn for a duration of $\tau(\boldsymbol{p})$. By executing this evasive maneuver, an area is traced by the static collision zone and we call this area the *default escape route*. To prevent a collision between the evader and the obstacle, the obstacle cannot be in the interior of the default escape route during such an evasive action.

Illustrated in Figure 3.4, the boundary of the default escape route consists of three segments. In purple, the first segment is the non-usable part of the terminal surface $\mathcal{S}_o$ given in (3.2). In black, the second segment is the boundary of the area between the two concentric circles and it is expressed by the two parametric curves of $\tau \in [0, \tau(\boldsymbol{p})]$:

$$\begin{bmatrix} x_{outer}(\tau) \\ y_{outer}(\tau) \end{bmatrix} = \begin{bmatrix} -R + (c+R)\cos\left(\frac{v_e}{R}\tau\right) \\ (c+R)\sin\left(\frac{v_e}{R}\tau\right) \end{bmatrix}, \tag{3.6}$$

[1] Or (3.3) with $x_o(\tau_o)$ replaced with $-x_o(\tau_o)$.



(a) *The default escape route in the reduced space.*

(b) *The default escape route in the realistic space where the nominal path is the vertical blue dashed line.*

Figure 3.4: *The default escape route is the area traced out by the static collision zone.*

and

$$\begin{bmatrix} x_{inner}(\tau) \\ y_{inner}(\tau) \end{bmatrix} = \begin{bmatrix} -R + (R-c)\cos\left(\frac{v_e}{R}\tau\right) \\ (R-c)\sin\left(\frac{v_e}{R}\tau\right) \end{bmatrix}. \tag{3.7}$$

The value of $\tau(\boldsymbol{p})$ is upper bounded by $\frac{2R\pi}{v_e}$. Let

$$\begin{bmatrix} x_e(\tau(\boldsymbol{p})) \\ y_e(\tau(\boldsymbol{p})) \\ \theta_e(\tau(\boldsymbol{p})) \end{bmatrix}$$

denote the state of the evader at the end of turning hard left (See Figure 3.4b). In blue, the third segment is the static barrier surface which is first translated by $(x_e(\tau(\boldsymbol{p})), y_e(\tau(\boldsymbol{p})))$ and then rotated by $\theta_e(\tau(\boldsymbol{p}))$:

$$\begin{bmatrix} x_o(\tau(\boldsymbol{p}), \tau_o) \\ y_o(\tau(\boldsymbol{p}), \tau_o) \end{bmatrix} = \begin{bmatrix} \cos\left(\theta_e(\tau(\boldsymbol{p}))\right) & -\sin\left(\theta_e(\tau(\boldsymbol{p}))\right) \\ \sin\left(\theta_e(\tau(\boldsymbol{p}))\right) & \cos\left(\theta_e(\tau(\boldsymbol{p}))\right) \end{bmatrix} \begin{bmatrix} x_o(\tau_o) \\ y_o(\tau_o) \end{bmatrix} + \begin{bmatrix} x_e\left(\tau(\boldsymbol{p})\right) \\ y_e\left(\tau(\boldsymbol{p})\right) \end{bmatrix}, \tag{3.8}$$

where $(x_o(\tau_o), y_o(\tau_o))$ is the expression of the static barrier surface in (3.3) and $\tau_o \in [0, \overline{\tau}_o]$. The left side branch of the transformed static barrier surface is also given by (3.8) where $x_o(\tau_o)$ is replaced by $-x_o(\tau_o)$.

The size of the default escape route depends on $\tau(\boldsymbol{p}) \in [0, \bar{\tau}]$ where $\bar{\tau}$ is the solution to (2.10) and the maximum value of τ_o, $\bar{\tau}_o$, which can be found by (3.4). It can be shown that $\bar{\tau} \leq \frac{R}{v_e}\pi$ and $\bar{\tau}_o \leq \frac{R}{3v_e}\pi$. The resulting default escape route that results from $\tau(\boldsymbol{p}) = \frac{R}{v_e}\pi$ and $\bar{\tau}_o = \frac{R}{3v_e}\pi$ is shown in Figure 3.5.

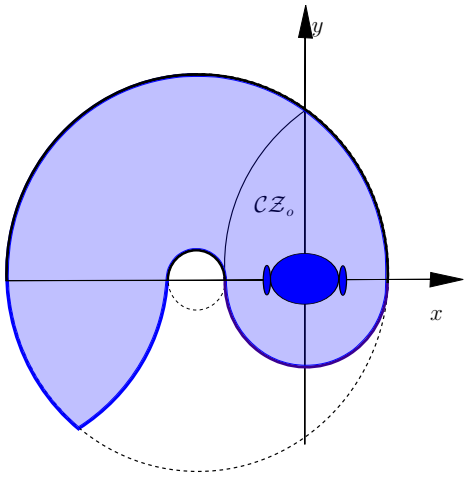


Figure 3.5: *The largest default escape route.*

3.1.3 Guaranteed evasion configuration

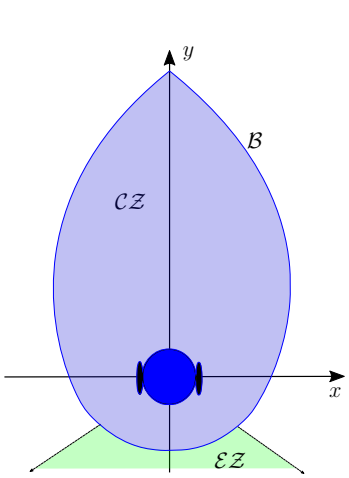
To simplify the forthcoming discussion, it is convenient to formally characterize configurations of the PSO game in which one of the adversarial players (pursuer or obstacle) is eliminated from the game and safety is easy to ensure by using the strategy of the reversed homicidal chauffeur game. We call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

The boundary of the usable part for the reversed homicidal chauffeur game consists of the two points (2.8). Using these points, define

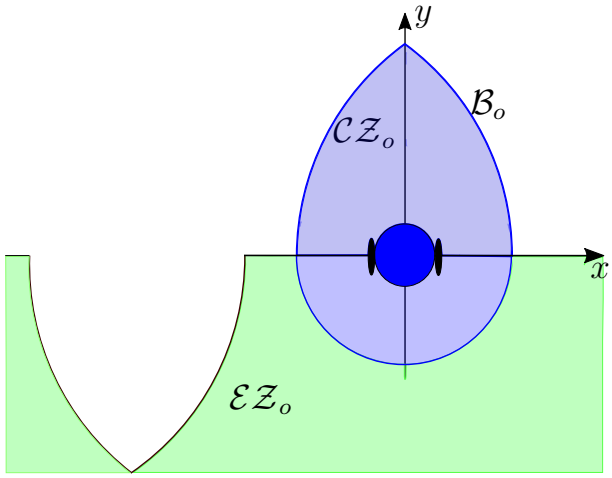
$$\mathcal{EZ} := \text{cone}\left(\left\{\begin{bmatrix}\sin(\bar{s})\\ \cos(\bar{s})\end{bmatrix}, \begin{bmatrix}-\sin(\bar{s})\\ \cos(\bar{s})\end{bmatrix}\right\}\right) \setminus \left\{\boldsymbol{p} \in \mathbb{R}^2 : \|\boldsymbol{p}\|_2 < c\right\}$$

to be the *evasion zone with respect to the pursuer*. It is coloured in green in Figure 3.6a. We use this evasion zone to clarify the statements in Remark 2.

Assumption 3. *If the pursuer enters the evasion zone $\mathcal{EZ}$, then the pursuer ceases its pursuit of the evader.*



(a) *Capture is inevitable if the pursuer enters collision zone $\mathcal{CZ}$. The pursuer stops pursuing if it enters the evasion zone $\mathcal{EZ}$.*



(b) *Collision is inevitable if the obstacle enters the static collision zone $\mathcal{CZ}_o$. The obstacle exits the game if it enters the static evasion zone $\mathcal{EZ}_o$.*

Figure 3.6: *Sets associated with the PSO game.*

Similarly, we define the *evasion zone with respect to the obstacle*, $\mathcal{EZ}_o$ as:

$$\left\{\boldsymbol{o} \in \mathbb{R}^2 : y_o \leq 0\right\}$$

set minus the largest default escape route, illustrated in Figure 3.8. This set is colored in green in Figure 3.6b.

The PSO game terminates if the pursuer is captured (Definition 2). In addition, we consider the game to have terminated if the players enter a guaranteed evasion configuration.

Definition 3 (Guaranteed Evasion Configuration)**.** *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 PSO game reduces to the reversed homicidal chauffeur and the work in [15] gives the strategy for evading the pursuer. In guaranteed

evasion configuration (ii), the evader applies maximum steering away from the obstacle when the obstacle arrives on its static barrier surface.

3.2 Overview of the Evasion Strategy

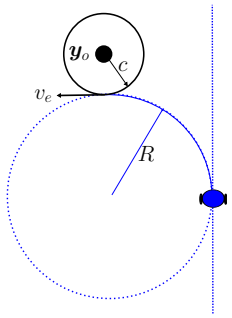
Initially, the evader follows a nominal path in the plane. In the absence of a static obstacle, the solution to the reversed homicidal chauffeur game [15] is for the evader to wait until the pursuer arrives on the barrier surface and then take a hard left at the "last moment." We call the area traced out by the robot while performing this evasive maneuver the *default escape route* (see Figure 3.4 and Section 3.1.2). Therefore, as long as the static obstacle is not in the interior of the default escape route, the evader can successfully avoid the pursuer by simply employing the reversed homicidal chauffeur strategy.

If the static obstacle arrives at the boundary of the default escape route and the pursuer is not on the barrier surface, then the evader must decide whether or not to allow the static obstacle to enter the default escape route and thereby (temporarily) interfere with the reversed homicidal chauffeur strategy. In order to make this decision, the first evaluation is to determine whether or not the pursuer can reach the barrier surface prior to the obstacle leaving the default escape route assuming the evader continues along its path. If this evaluation is false, then the static obstacle won't interfere with the reversed homicidal chauffeur strategy and the evader can continue to follow its path.

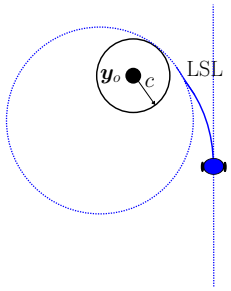
Otherwise, if the evaluation is true, then the evader must decide whether to go "under" the static obstacle by immediately taking the default escape route (Figure 3.7a and Figure 3.7c) or to delay the evasive maneuver by "threading the needle" and going between the obstacle and the pursuer in the future (Figure 3.7b and Figure 3.7d); the latter maneuver is sometimes referred to as going "above" the static obstacle. The complexity of the problem arises from this decision.

An evasive action at time t is a piece-wise continuous control signal with the property that[2] $u(t^+) \neq 0$, i.e., it results in the evader leaving its nominal path. When going "under" the obstacle, the class of admissible evasive actions is $|u(\cdot)| = 1$, i.e., a hard left or right turn. When going "above" the obstacle, the class of admissible evasive actions are those control signals that result in the evader tracing out one of four Dubins paths: LS, LSL, RL, and RSL. In [13] and [33], the optimal path between any two poses for a Dubins vehicle is in the set {CCC,CSC} where C $\in$ {R, L} is a circular turning maneuver with minimum turning radius either to the right R or to the left L and S is the straight maneuver.

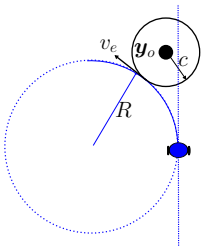
[2]The notation $u(t^+)$ denotes the limit of $u(\tau)$ as $\tau \to t$ from the right.



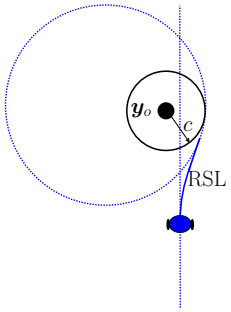
(a) *The obstacle is not on the nominal path and the evader turns below it.*



(b) *The obstacle is not on the nominal path and the evader turns above it.*



(c) *The obstacle is on the nominal path and the evader turns below it.*



(d) *The obstacle is on the nominal path and the evader turns above it.*

Figure 3.7: *Methods for the evader with minimum turning radius, R to evade the obstacle (black dot) starting from a position on the nominal path (blue dashed line).*

Definition 4 (Feasible evasive action)**.** *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:*

(i) for every $\tau \in [t, t+T]$, $\boldsymbol{o}(\tau) \notin \mathcal{CZ}_o$ and $\boldsymbol{p}(\tau) \notin \mathcal{CZ}$

(ii) $(\boldsymbol{o}(t+T), \boldsymbol{p}(t+T))$ is a guaranteed evasion configuration.

In our setup, the obstacle may or may not lie on the nominal path. When the obstacle does not lie on the nominal path, there is a family of feasible LSL Dubins paths where the circle corresponding to the last L maneuver contains the obstacle's collision disc at each moment before the obstacle is inside the static evasion zone. Any of these LSL evasive actions which encompasses the obstacle like the one illustrated in Figure 3.7b, allows the

evader to evade the obstacle by just turning above it. In some cases, the guaranteed evasion configuration is achieved during the evader's S maneuver. Hence, the LS evasive action may be used in the case where the obstacle does not lie on the nominal path.

When the obstacle lies on the nominal path, there is a family of feasible RSL Dubins paths where the circle corresponding to the last L maneuver also contains the obstacle like the one illustrated in Figure 3.7d. In some cases, the guaranteed evasion configuration is achieved with the evader only executing the RL maneuver. In either scenario, our strategy is to try and go "above" the obstacle whenever possible by continuously searching for the availability of feasible LS/LSL/RL/RSL maneuvers as the evader follows the nominal path.

Our collision avoidance algorithm is event based. An evasive maneuver is triggered by the events listed in Table 3.1. The obstacle avoidance algorithm continuously monitors these events and if an event occurs, then the evader responds with the corresponding evasive action.

Table 3.1: *The events that trigger an evasive action.*

Event	Event Description	Evader's Responses
1	The pursuer is on the barrier surface.	Turn hard left.
2	The obstacle is on the boundary of the default escape route and there is no feasible LS/LSL/RL/RSL.	Turn hard left.
3	The obstacle is in the default escape route and does not interfere the nominal path; the last moment in time for which a feasible LS/LSL exists.	Execute the feasible LS/LSL.
4	The obstacle is in the default escape route and interferes the nominal path; the last moment in time for which a feasible RL/RSL exists.	Execute the feasible RL/RSL.

Remark 3. *Events 3 and 4 in Table 3.1 merit more explanation. In particular, the term "last moment" means that at time t there exists a feasible evasive action but at t^+ no feasible evasive actions exist. In practice, these events are monitored by checking at time t whether or not a feasible evasive action exists at time $t + \delta t$ where $\delta t > 0$.*

3.2.1 Algorithms

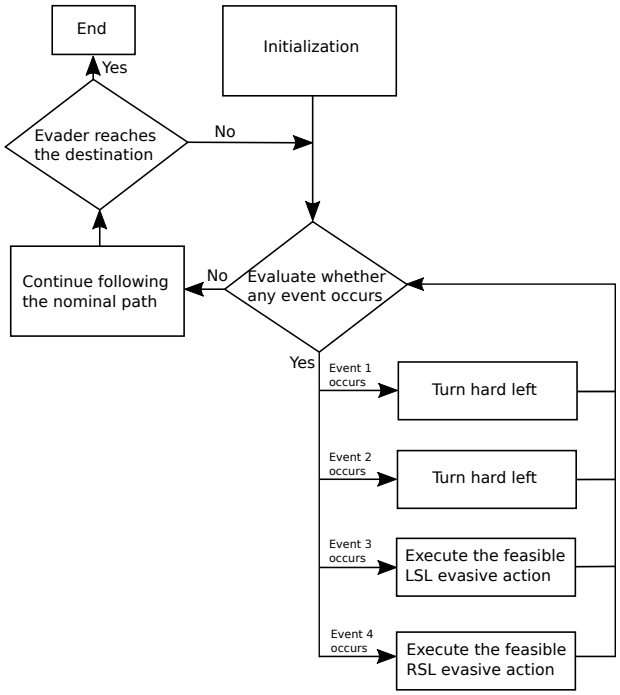


Figure 3.8: *Flowchart of our event triggered controller, using the events in Table 3.1.*

Our algorithm is summarized by the flow chart in Figure 3.8 and is based on the events listed in Table 3.1.

Remark 4. *The algorithm described by Figure 3.8 assumes that, initially, the pursuer is to the right of the nominal path while the static obstacle is to the left. Since the game is symmetric about the nominal path, the algorithm can be easily modified to fit the situation where the obstacle and the pursuer swap sides. If the pursuer and the obstacle are on the same side of the evader, then the solution of the PSO game is to wait until either the pursuer or obstacle reaches each of their respective barrier surface and then the evader turns away from them.*

Theorem 1. *Suppose that initially the pursuer and the obstacle are outside their respective collision zones and that the obstacle is not in the interior of the default escape route. Then, under Assumptions 1 through 3, the execution of the algorithm in Figure 3.8 guarantees that the evader is not captured.*

Proof. Let $T \geq 0$ denote the time at which the first event in Table 3.1 occurs.

Case 1: (Event 1 triggers first) By the definition of Event 1, the pursuer 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 pursuer is not in its collision zone. Therefore, by turning hard left, the PSO 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 pursuer and the obstacle do not enter their respective collision zones while the evader executes this evasive action (see Definition 4). After the evader avoids the obstacle by going "above" it, the PSO game enters guaranteed evasion configuration (see Definition 3).

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 pursuer 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 evader 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 evader 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. □

In order to implement our algorithm, various subroutines must be developed. First, we must be able to determine whether or not the pursuer can arrive on the barrier surface before the obstacle leaves the default escape route; this subroutine is described in Section 3.4.2. In order to monitor Events 2, 3, and 4, we must be able to search the space of feasible LS/LSL/RL/RSL maneuvers and determine the last moment in time which a feasible maneuver exists. Much of the remainder of this chapter is dedicated to solving this issue. Detailed descriptions of how to check for feasible LSL and RSL evasive actions are presented in Section 3.4.3 and Section 3.4.4, respectively.

With respect to part (ii) of Problem 1, we were unable to prove that the algorithm in Figure 3.8 delays the evasive action as much as possible. However, it is clear that by "threading the needle" and going "above" the obstacle whenever possible, our collision avoidance strategy results in delayed evasions compared to the strategy of always taking a hard left whenever the obstacle might potentially interfere with the reversed homicidal chauffeur strategy.

3.3 Technical Preliminaries

In this section we explain how to compute two time values that are needed to implement our evasion strategy. The first is the amount of time it takes the obstacle to leave the default escape route. The second is the minimum time for the pursuer to arrive on the barrier surface assuming the evader moves in a straight line.

3.3.1 Time for the Obstacle to Leave the Default Escape Route

Given $\tau \in [0, \bar{\tau}]$ which defines the extent of the default escape route, and given an obstacle position $\boldsymbol{o}(0) = (x_o(0), y_o(0))$ in the default escape route, and assuming the evader's control signal u is either $u(\cdot) = 0$ (move straight) or $u(\cdot) = 1$ (turn hard right), we seek to compute the smallest non-zero time at which the obstacle reaches the boundary of the default escape route.

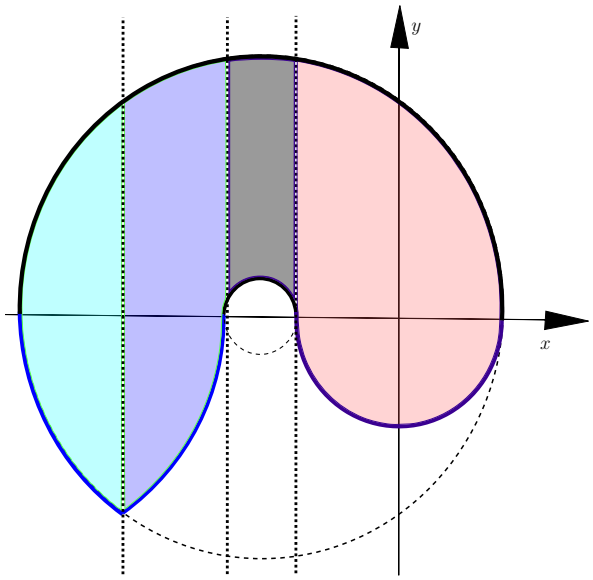


Figure 3.9: *The segmented default escape route with different T_S.*

Time to Exit the Default Escape Route when the Evader Moves Straight

Illustrated in Figure 3.9, the default escape route can be partitioned into four subsets. As long as the obstacle is initially not in the red shaded region where $x_o(0) \leq -c$, the evader's continued S maneuver results in the obstacle leaving the default escape route. Let t_S be the duration of the evader's straight maneuver. Assuming the obstacle is inside the default escape route, let T_S be the required time that the obstacle first reaches the boundary of the default escape route as result of the evader going straight. To solve for the corresponding T_S, we need to find where the obstacle leaves the default escape route.

If the obstacle is initially in the cyan shaded region, the evader's continued straight maneuver eventually results in the obstacle leaving the default escape route through the right side branch of the transformed static barrier surface (3.8). The right side branch of the transformed static barrier surface is an extension of the outer boundary for the second segment of the default escape route (3.6). Since τ is the given value that defines the extent of the default escape route, then let $\tilde{\tau}$ be the variable in the expression (3.6). To solve for the corresponding T_S, we first solve the following system of equation with the given τ and the obstacle's initial position:

$$x_o(0) = x_{outer}(\tilde{\tau})$$

for $\tilde{\tau} \in [\tau, \tau + \overline{\tau}_o]$. After finding the value for $\tilde{\tau}$, then substitute it into $y_{outer}(\tilde{\tau})$ and

$$T_S = \frac{y_o(0) - y_{outer}(\tilde{\tau})}{v_e}.$$

If the obstacle is in the blue shaded region, the evader's continued straight maneuver eventually results in the obstacle leaving the default escape route through the left side branch of the transformed static barrier surface (3.8). To solve for the corresponding t_S, we first solve the following system of equation with the given τ and the obstacle's initial position:

$$x_o(0) = x_o(\tau, \tau_o)$$

for $\tau_o \in [0, \overline{\tau}_o]$. After finding the value for τ_o, then substitute it into $y_o(\tau, \tau_o)$ and

$$T_S = \frac{y_o(0) - y_o(\tau, \tau_o)}{v_e}.$$

If the obstacle is in the grey shaded region, the evader's continued straight maneuver eventually results in the obstacle leaving the default escape route through the lower boundary of the second segment given by (3.7). The procedure to find the corresponding T_S follows the similar steps as for the obstacle inside the cyan shaded region. We first solve the following system of equation with the given τ and the obstacle's initial position:

$$x_o(0) = x_{inner}(\tilde{\tau})$$

for $\tilde{\tau} \in [0, \tau]$. After finding the value for $\tilde{\tau}$, then substitute it into $y_{inner}(\tilde{\tau})$ and

$$T_S = \frac{y_o(0) - y_{inner}(\tilde{\tau})}{v_e}.$$

If the pursuer can get inside the collision zone while the evader moves straight for $t_S \in [0, T_S]$, then the capture can occur while the evader tries to get the obstacle outside of the default escape route as a result of its straight maneuver. Otherwise, the capture does not occur. This evaluation is used to determine whether or not a feasible LS/LSL or RSL exists in Section 3.4.3 and Section 3.4.4.

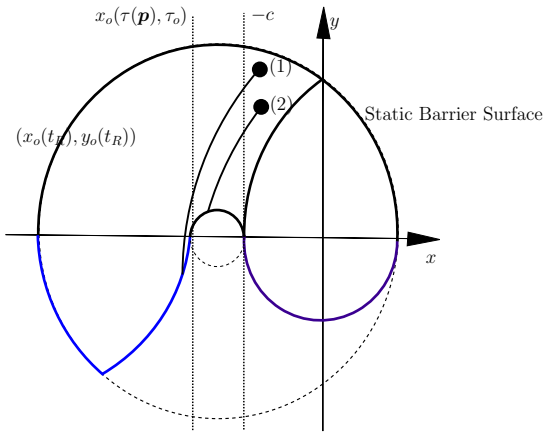


Figure 3.10: *The motion of the obstacle inside the default escape route due to the evader turning hard right.*

Time to Exit the Default Escape Route when the Evader Turns Hard Right

Illustrated in Figure 3.10, the obstacle leaves the default escape route either through the inner concentric circle (3.7) (see the path starting from point 1) or the left side branch of the transformed static barrier surface (3.8) (see the path starting from point 2). If the evader turns hard right, the obstacle moves in a concentric circle in the reduced space. Let t_R be the duration of the evader's hard right turn. Based on (3.1), the obstacle's position in the reduced space at the end of the right turn is

$$\begin{aligned} x_o(t_R) &= R + r_{R_o} \cos\left(\frac{v_e}{R} t_R + \alpha_{R_o}\right) \\ y_o(t_R) &= r_{R_o} \sin\left(\frac{v_e}{R} t_R + \alpha_{R_o}\right), \end{aligned} \tag{3.9}$$

where $r_{R_o} = \sqrt{(x_o(0) - R)^2 + y_o(0)^2}$ and $\alpha_{R_o} = \arctan\left(\frac{y_o(0)}{x_o(0) - R}\right)$.

Assuming the obstacle is inside the default escape route, let T_R be the required time that the obstacle first reaches the boundary of the default escape route as a result of the evader turning hard right. If the obstacle leaves the default escape route through the left side branch of the transformed static barrier surface (3.8), then solve the following system of equations for $t_R \in [0, \frac{R}{v_e}\pi]$ and $\tau_o \in [0, \overline{\tau}_o]$ with given τ and the initial obstacle's position:

$$x_o(t_R) = x_o(\tau, \tau_o) \quad \text{and} \quad y_o(t_R) = y_o(\tau, \tau_o).$$

Since τ is the given value that defines the extent of the default escape route, then let $\tilde{\tau}$ be the variable in the expression (3.7). If the obstacle leaves the default escape route through the inner concentric circle (3.7), then solve the following system of equations for $t_R \in [0, \frac{R}{2v_e}\pi]$ and $\tilde{\tau} \in [0, \tau]$ with given τ and the initial obstacle's position:

$$x_o(t_R) = x_{inner}(\tilde{\tau}) \quad \text{and} \quad y_o(t_R) = y_{inner}(\tilde{\tau}).$$

After solving the above system of equations, T_R is found. If the pursuer can get inside the collision zone while the evader turns hard right for $t_R \in [0, T_R]$, then the capture can occur while the evader tries to get the obstacle outside of the default escape route as a result of its hard right turn. Otherwise, the capture does not occur. This evaluation is used to determine whether or not a feasible RL exists in Section 3.4.4.

3.3.2 Time for Pursuer to Arrive on Barrier Surface

Given the initial pursuer's position $\boldsymbol{y}_p(0)$ and assuming the evader's control signal is $u(\cdot) = 0$ (move straight), we seek to compute the minimum time for the pursuer to arrive on the barrier surface and the longest duration of the hard left turn to evade the pursuer after the pursuer arrives on the barrier surface. To simplify the analysis, the nominal path is assumed to be a vertical line fixed along the positive y-axis and the evader's initial heading is fixed at $\frac{\pi}{2}$.

Time for Pursuer to Arrive at a General Point on Barrier Surface

Before deriving the minimum time for the pursuer to arrive on the barrier surface, we first look at the pursuer's strategy to arrive at a point on the barrier surface given by τ in the least amount of time. In the realistic space, the motion of each point on the barrier surface depends on the evader's motion and state, $\boldsymbol{x}_e(t)$:

$$\begin{bmatrix} x(\boldsymbol{x}_e(t), \tau) \\ y(\boldsymbol{x}_e(t), \tau) \end{bmatrix} = \begin{bmatrix} \cos\left(\phi_e(t) - \frac{\pi}{2}\right) & -\sin\left(\phi_e(t) - \frac{\pi}{2}\right) \\ \sin\left(\phi_e(t) - \frac{\pi}{2}\right) & \cos\left(\phi_e(t) - \frac{\pi}{2}\right) \end{bmatrix} \begin{bmatrix} x(\tau) \\ y(\tau) \end{bmatrix} + \begin{bmatrix} x_e(t) \\ y_e(t) \end{bmatrix}, \tag{3.10}$$

where $(x(\tau), y(\tau))$ is given by (2.9) and $\tau \in [0, \overline{\tau}]$. Illustrated in Figure 3.11, given the initial location,

$$\begin{bmatrix} x(\boldsymbol{x}_e(0), \tau) \\ y(\boldsymbol{x}_e(0), \tau) \end{bmatrix} \in \mathcal{B} \subset \mathbb{R}^2, \tag{3.11}$$

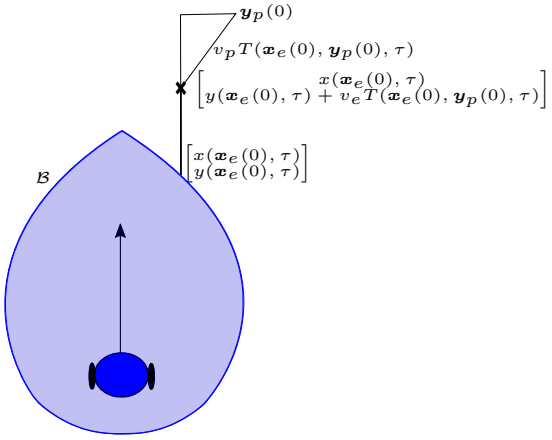


Figure 3.11: *The evader moves straight. The pursuer reaches a point on the barrier surface under the constant bearing scheme and the diagonal cross indicates the moment of capture.*

a general point on the barrier surface moves along a vertical path as the evader moves straight. Based on [44] [5], the pursuer's strategy that takes the least amount of time for it to arrive at the point given by τ on the barrier surface is to use the *constant bearing control*, assuming that the evader moves straight.

Illustrated in Figure 3.11, the pursuer's strategy to arrive at the point parameterized by τ on the barrier surface in the least amount of time is to move towards the point given by

$$\begin{bmatrix} x(\boldsymbol{x}_e(0), \tau) \\ y(\boldsymbol{x}_e(0), \tau) + v_e T(\boldsymbol{x}_e(0), \boldsymbol{y}_p(0), \tau) \end{bmatrix},$$

where

$$\begin{aligned} T(\boldsymbol{x}_e(0), \boldsymbol{y}_p(0), \tau) = & \frac{\sqrt{v_p^2\big(\left(x(0) - x(\tau)\right)^2 - \left(y(0) - y(\tau)\right)^2 \big) - v_e^2 \left(x(0) - x(\tau)\right)^2}}{(v_e^2 - v_p^2)} \\ & - \frac{v_e \left(y(\tau) - y(0)\right)}{(v_e^2 - v_p^2)}, \end{aligned} \tag{3.12}$$

is the minimum time for the pursuer to arrive on the barrier surface at the point given by (3.11), (see Appendix A for detailed derivation of $T(\boldsymbol{x}_e(0), \boldsymbol{y}_p(0), \tau)$). As the evader moves along the vertical path fixed along the y-axis, with a slight abuse of notation, taking $(x(\tau), y(\tau))$ as the initial position of a general point on the barrier surface that the pursuer tries to catch and $(x(0), y(0))$ as the initial pursuer's position, the rest of the analysis in

this section uses these notations (see Appendix A for the reasoning).

Max Duration to Execute Hard Left Turn to Evade the Pursuer

As the evader moves straight, one can derive the point on the barrier surface that results in the maximum $\tau \in [0, \overline{\tau}]$ where the pursuer can reach. This value indicates the maximum duration for the evader to execute the hard left turn to evade the pursuer after it reaches the barrier surface. For $T(\boldsymbol{x}_e(0), \boldsymbol{y}_p(0), \tau)$ to be a real number, the following condition must be satisfied with regard to the initial pursuer's position:

$$y(0) \geq \frac{\sqrt{v_e^2 - v_p^2}}{v_p} \left(x(0) - x(\tau)\right) + y(\tau), \tag{3.13}$$

and see Appendix A for detailed derivation of this linear relationship between the y-coordinate and the x-coordinate of the initial pursuer's position. Given a $\tau \in [0, \overline{\tau}]$, a point on the barrier surface, if the initial pursuer's position does not satisfy the condition in (3.13), then the pursuer can never arrive at this point on the barrier surface.

Remark 5. *Given the pursuer's initial position, the equality in* (3.13) *can be used to find the upper bound for* $\tau \in [0, \overline{\tau}]$ *where the pursuer can arrive on the barrier surface, assuming that the evader continuously moves straight.*

In regard to Remark 5, the upper bound for $\tau \in [0, \overline{\tau}]$ can be found by solving the system of equations between the initial pursuer's position and (3.13). This upper bound is the maximum duration that is required for the evader to turn hard left to evade the pursuer once the pursuer arrives on the barrier surface. Also, this upper bound is used to construct the default escape route which is used in Section 3.4.

The Minimum Time for the Pursuer to Arrive on the Barrier Surface

Let T_{BRS} be minimum time for the pursuer to arrive on the barrier surface, assuming that the evader continuously moves straight. Based on (3.12), T_{BRS} can be found by solving the following minimization problem parameterized by the pursuer's initial position:

$$\begin{aligned} \min_{\tau} \quad & T(\boldsymbol{x}_e(0), \boldsymbol{y}_p(0), \tau) \\ \text{subject to} \quad & \tau \in [0, \overline{\tau}], \end{aligned} \tag{3.14}$$

where each τ must satisfy the condition in (3.13).

The Series of Closed form Equations to Solve for T_{BRS}

We can compute a series of closed-form equations to solve for T_{BRS} instead of solving the minimum problem (3.14). The series of closed-form equations define the boundary of a set that consists the pursuer's initial positions where the pursuer can arrive on the barrier surface within a time, T. Known as the *capture set*, this set consists of all the individual capture sets for all points on the barrier surface, $\tau \in [0, \overline{\tau}]$. Therefore to find the series of closed-form equations to solve for T_{BRS}, we need to first find the individual set for a general point on the barrier surface associated to τ and then we can obtain the boundary of the capture set.

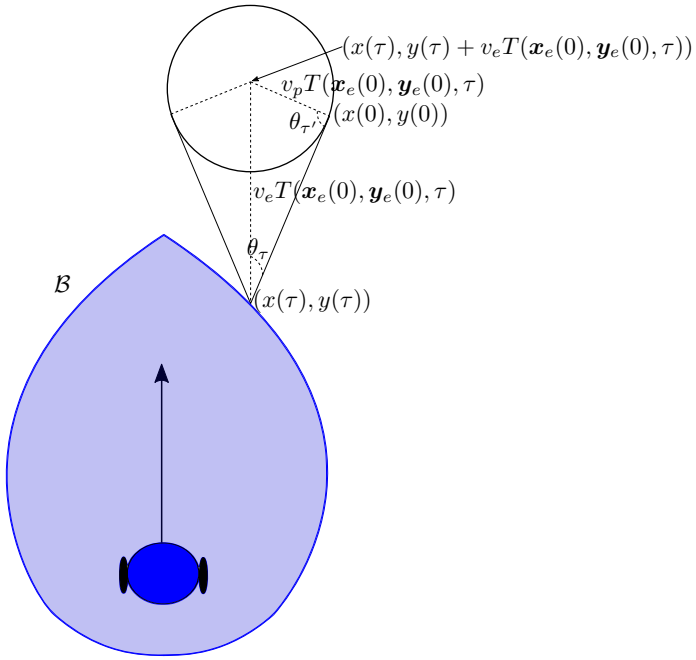


Figure 3.12: *The capture set where the pursuer can catch the point on the barrier surface, parameterized by τ within $T(\boldsymbol{x}_e(0), \boldsymbol{y}_p(0), \tau)$ under constant bearing scheme.*

Illustrated in Figure 3.12, the set that resembles an "ice cream cone" is the individual capture set for a general point on the barrier surface associated to τ. We know that the condition in (3.13) must be satisfied between the initial pursuer's position and the point on the barrier surface in order for the pursuer to be able to arrive on the point as the evader moves along a straight line. In Appendix A, the y-coordinate and the x-coordinate of the initial pursuer's position in the linear relationship (3.13) can be individually expressed with

the term $T(\boldsymbol{x}_e(0), \boldsymbol{y}_p(0), \tau)$ which is the time that the pursuer arrives at the point given by τ under constant bearing scheme[3]:

$$\begin{aligned} x(0) &= \left(\frac{v_p}{v_e}\sqrt{v_e^2 - v_p^2}\right) T(\boldsymbol{x}_e(0), \boldsymbol{y}_p(0), \tau) + x(\tau) \\ y(0) &= \left(\frac{v_e^2 - v_p^2}{v_e}\right) T(\boldsymbol{x}_e(0), \boldsymbol{y}_p(0), \tau) + y(\tau) \end{aligned} \tag{3.15}$$

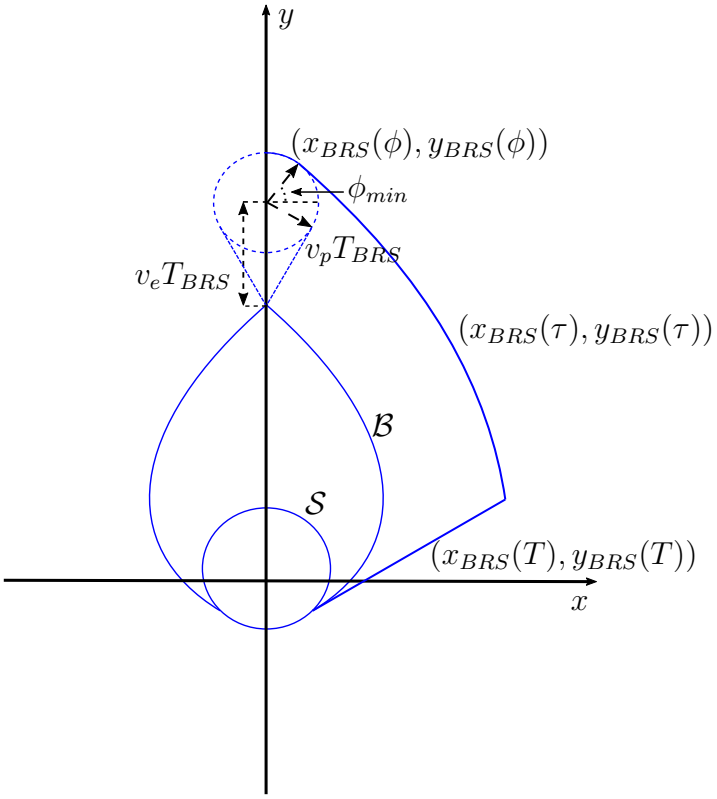
The linear relationship becomes the linear boundary which is the "cone" that holds the "ice cream" in Figure 3.12. Also illustrated in Figure 3.12, the point of capture is

$$(x(\tau), y(\tau) + v_e T(\boldsymbol{x}_e(0), \boldsymbol{y}_p(0), \tau))$$

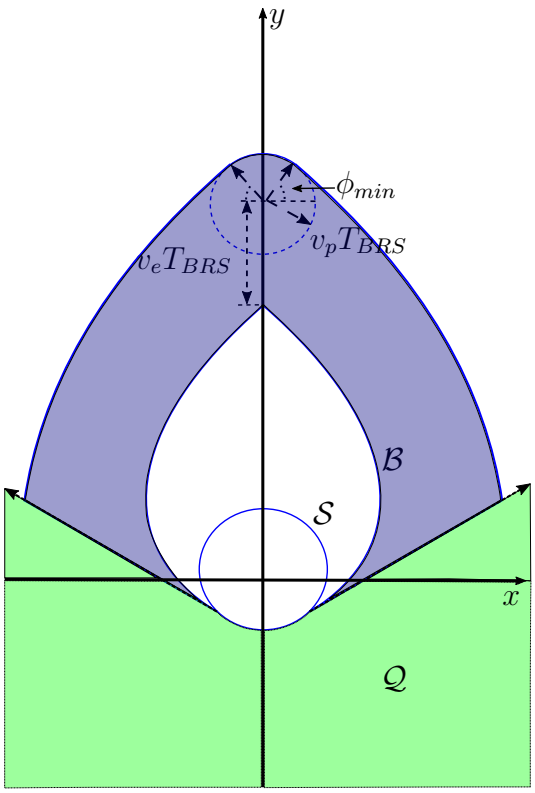
where the pursuer arrives on the barrier surface at a given τ, assuming that the arrival time for the pursuer is $T(\boldsymbol{x}_e(0), \boldsymbol{y}_p(0), \tau)$. A circle centered at the point of capture with radius $v_p T(\boldsymbol{x}_e(0), \boldsymbol{y}_p(0), \tau)$ can be formed which shows the set of the initial pursuer's positions where the pursuer can arrive at the point associated to τ within $T(\boldsymbol{x}_e(0), \boldsymbol{y}_p(0), \tau)$. In Appendix A, we show that the left and right branches of the linear boundary (3.15) are tangent to this circle at the two point of contacts. The "ice cream" part of the capture set is the circular boundary between the two point of contacts. Therefore, the set of space that allows the pursuer to arrive on the barrier surface at the point given by τ, within the interval from 0 to $T(\boldsymbol{x}_e(0), \boldsymbol{y}_p(0), \tau)$ is enclosed by both the right and left branches of the linear boundary (3.15) and the arc of the circular boundary in between the two points of contact.

After finding the individual capture set for a general point on the barrier surface associated to τ, the overall capture set for the entire barrier surface is the union of all capture sets for $\tau \in [0, \overline{\tau}]$, shown in Figure 3.13. If the pursuer is on the boundary of the overall capture set, then the minimum time for it to arrive on the barrier surface under the constant bearing scheme is T_{BRS}. The boundary of the overall capture set consists of three segments. Let $(x_{BRS}(\cdot), y_{BRS}(\cdot))$ be the function for the boundary of the set and the function takes in different inputs for different segments of the boundary. The first segment is based on the linear boundary of the individual capture set with regard to the point on the

[3]This is the linear boundary of the "ice cream cone" in Figure 3.12 and it consists of the left and right branches. The left branch is found by negating the expression for $x(0)$.



(a) *The overall capture set is the union of all individual capture sets for all points on the barrier surface given by* $\tau \in [0, \overline{\tau}]$.



(b) *The blue shaded region is the set where the pursuer can catch the barrier surface if the evader applies the S maneuver for a period of* T_{BRS}. *The green shaded region is under the linear boundary in* (3.16) *where the pursuer can never reach the barrier surface if the evader continues to apply the S maneuver.*

Figure 3.13: *The overall capture set resulting from the constant bearing scheme.*

barrier surface associated to $\tau = 0$ and the first segment is given by

$$\begin{aligned} x_{BRS}(T) &= \frac{v_p}{v_e}\sqrt{v_e^2 - v_p^2}T + x(\tau) \\ y_{BRS}(T) &= \frac{v_e^2 - v_p^2}{v_e}T + y(\tau), \end{aligned} \tag{3.16}$$

where $T \in [0, T_{BRS}]$. The second segment is given by

$$\begin{aligned} x_{BRS}(\tau) &= \sin\left(\bar{s} - \frac{v_e}{R}\tau\right) v_p T_{BRS} + x(\tau) \\ y_{BRS}(\tau) &= \cos\left(\bar{s} - \frac{v_e}{R}\tau\right) v_p T_{BRS} + v_e T_{BRS} + y(\tau), \end{aligned} \tag{3.17}$$

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

$$\begin{bmatrix} \sin\left(\bar{s} - \frac{v_e}{R}\tau\right) \\ \cos\left(\bar{s} - \frac{v_e}{R}\tau\right) \end{bmatrix}$$

is the normal vector of unit length for each point on the barrier surface extending into the opposite direction of the collision zone, $\mathcal{CZ}$. The third segment is based on the circular boundary of the individual capture set for the point on the barrier surface associated to $\tau = \overline{\tau}$ and the third segment is given by

$$\begin{aligned} x_{BRS}(\phi) &= v_p \cos(\phi) T_{BRS} + x(\tau) \\ y_{BRS}(\phi) &= v_p \sin(\phi) T_{BRS} + y(\tau) + v_e T_{BRS} \end{aligned} \tag{3.18}$$

where $\phi \in [\phi_{min}, \frac{\pi}{2}]$ and ϕ_{min}[4] is illustrated in Figure 3.13a. The boundary of the overall capture set is symmetrical along the positive y-axis. Therefore, if the pursuer is initially on the right side with respect to the y-axis, the right side boundary of the overall capture set is considered.[5]

The linear boundary of (3.16) is aligned with the outward extending vector for the point on the barrier surface where $\tau = 0$. If the pursuer falls on or under this linear boundary, which means that $y(t) \leq y_{BRS}(T)$ given in (3.16), then the pursuer can never catch the barrier surface if the evader continuously travels along a straight path. Therefore, illustrated in Figure 3.13b, if $\boldsymbol{p} \in \mathcal{Q}$, then the evader can force the pursuer to get inside the set $\mathcal{EZ}$ if it continues to travel along a straight path. By solving the systems of equations for (3.16), (3.17), and (3.18) individually with the initial pursuer's position, the time value can be found. The minimum time value obtained from solving the three individual systems of equations is the minimum time that the pursuer takes to arrive on the barrier surface under the constant bearing scheme.

[4] ϕ_{min} can be found by solving the system of equations for (3.17) and (3.18).

[5] Illustrated in Figure 3.13b, the left side boundary of the overall capture set is found by negating the x-coordinate value in (3.16), (3.17), and (3.18).

3.4 Evaluation of Triggering Conditions

In this section, we show how to evaluate whether or not Event 2, Event 3 or Event 4 from Table 3.1 is triggered in a PSO game. The evaluation to check any event in Table 3.1 may involve solving systems of nonlinear equations. To solve these nonlinear systems of equations, we use a nonlinear solver [7], [6], [50] based on *Newton's method* with a bounded variable space.

3.4.1 Pursuer's Action Circle

The intention of the pursuer cannot be determined and the pursuer may use the obstacle to aid its capture of the evader. Therefore, the evader must consider the reachable set of the pursuer. In the realistic space, the pursuer's reachable set is an expanding circle with respect to time. This expanding circle is denoted as the *pursuer's action circle*, $\mathcal{C}_p$ centered at the pursuer's initial position and its radius is $v_p t$ where t is the time duration that the evader executes the evasive strategy. In the reduced space, the pursuer's action circle, $\mathcal{C}_p$ is an expanding circle and its center moves relative to the evader.

3.4.2 Evaluation of Obstacle inside Default Escape Route

Let t_{reach} be the moment that the obstacle arrives on the upper boundary of the default escape route. At t_{reach}, the evader evaluates whether or not the pursuer can reach the barrier surface prior to the obstacle leaving the default escape route. If the pursuer cannot reach the barrier surface while the obstacle is strictly inside the default escape route, then Event 2, 3, or 4 in Table 3.1 cannot occur. Hence, the evader only needs to monitor if Event 1 in Table 3.1 occurs and reacts accordingly. To evaluate whether or not the pursuer can get onto the barrier surface while the obstacle is inside the default escape route, the following procedure is used.

Construction of the Default Escape Route

First we construct the default escape route. By using (3.15), the evader knows where the pursuer can arrive at the point on the barrier surface that results in the maximum τ, i.e., the maximum duration to turn hard left to evade the pursuer after arriving on the barrier

surface.[6] To construct the default escape route, we use the maximum τ value as the input for (3.6) and (3.8). The default escape route at each moment in time can be obtained and the whereabouts of the obstacle relative to the default escape route can be found.

Time for Pursuer to Arrive on Barrier Surface

Before any Event in Table 3.1 is triggered, the evader keeps following the nominal path. Based on the discussion in Section 3.3.2, the pursuer's strategy to arrive on the barrier surface in the least amount of time if the evader keeps following the nominal path is to follow the constant bearing strategy. Mentioned at the end of the Section 3.3.2, T_{BRS} can be found by solving the system of equations involving (3.16), (3.17), and (3.18) with the pursuer's position at t_{reach} as the initial pursuer's position. T_{BRS} is the minimum required time for the pursuer to arrive on the barrier surface.

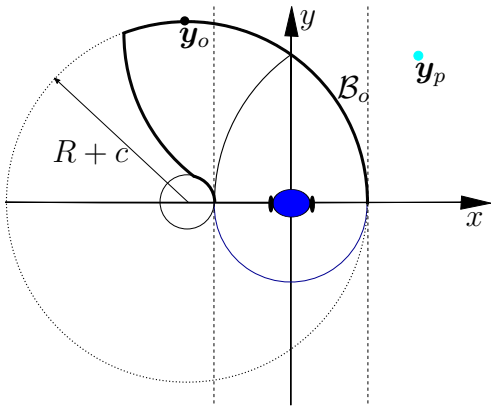
Check whether or not Event 2, 3, or 4 Occurs

With the default escape route and the obstacle's position at t_{reach}, the time for the obstacle to leave the interior of the default escape route if the evader keeps following the nominal path can be found by following Section 3.3.1. Now by comparing the time for the obstacle to leave the default escape route with T_{BRS}, the evader can decide whether the obstacle is outside of the default escape route at the moment when the pursuer is on the barrier surface. If the pursuer arrives on the barrier surface prior to the obstacle leaving the default escape route, the evader must have a feasible evasive action at any moment while the obstacle is inside the default escape route so that the guaranteed evasion can be achieved. The following sections in this chapter show how to determine whether or not a feasible evasive action exists at a moment in time while the obstacle is inside the default escape route.

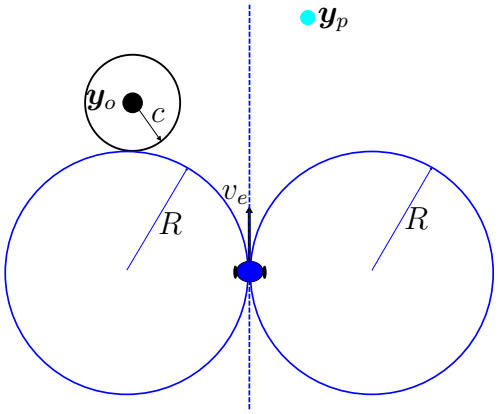
3.4.3 Determine the Existence of Feasible LS/LSL

In this section, we discuss how to evaluate the existence of a feasible LS/LSL evasive action in the scenario that the obstacle does not interfere the nominal path, illustrated in Figure 3.14. The triggering of Event 2 or Event 3 in Table 3.1 depends on the existence of a feasible LS/LSL evasive action. A LS/LSL evasive action is defined by how long the

[6] We can solve a system of equations for the variable $\tau \in [0, \overline{\tau}]$ with the pursuer's position at the moment of evaluation as the initial pursuer's position in (3.15).



(a) *Based on Figure 3.4a, the configuration of all players in the reduced space at* t_{reach}.



(b) *The configuration of all players in the realistic space at* t_{reach} *when the collision disc centered at the obstacle touches the left minimum turning circle with radius* R.

Figure 3.14: *The configuration of all players at* t_{reach} *when the obstacle (black dot) is on the boundary of the default escape route and does not interfere the nominal path.*

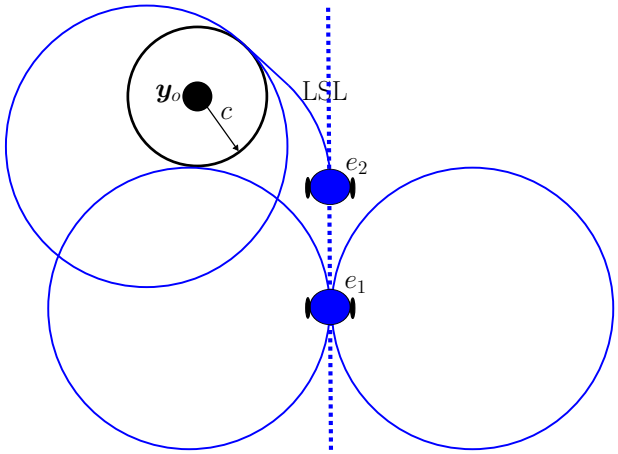


Figure 3.15: *After executing the LS maneuver starting at* e_2, *the evader can turn hard left to evade the pursuer with obstacle outside the default escape route.*

evader applies the first L maneuver and the second S maneuver depends on the duration of the first L maneuver.

Illustrated in Figure 3.15, after executing the first L and the second S maneuver, the evader's left minimum turning circle encompasses the obstacle so that the threat of colliding into the obstacle is eliminated. Then the evader can either turn hard left immediately after the second S maneuver if the pursuer is on the barrier surface or can wait until the pursuer is on the barrier surface to turn hard left. Therefore, the PSO game becomes the reversed homicidal chauffeur game after the second S maneuver and the guaranteed evasion can be achieved.

The Motion of the Obstacle due to the L maneuver

Let t_L be the duration for the evader to execute the first L maneuver of a LS/LSL evasive action. Similar to the obstacle's motion due to the R maneuver (3.9), the obstacle moves in a circle in the reduced space due to the L maneuver:

$$\begin{aligned} x_o(t_L) &= -R + r_{L_o} \sin\left(\frac{v_e}{R} t_L + \alpha_{L_o}\right) \\ y_o(t_L) &= r_{L_o} \cos\left(\frac{v_e}{R} t_L + \alpha_{L_o}\right), \end{aligned} \tag{3.19}$$

where $r_{L_o} = \sqrt{(x_o(0)+R)^2 + y_o(0)^2}$ and $\alpha_{L_o} = \arctan\left(\frac{x_o(0)+R}{y_o(0)}\right)$. The above expression is obtained by first solving the differential equation (3.1) with $u(\cdot) = -1$[7].

Duration of S maneuver for each LS/LSL Evasive Action

Let $T_S(t_L)$ denote the duration of the S maneuver after switching from the initial L maneuver so that the obstacle eventually leaves the interior of the default escape route. With a defined t_L, $(x_o(t_L), y_o(t_L))$ is the obstacle's initial relative position after the evader executing the first L maneuver and before the S maneuver. For each LS/LSL evasive action, its corresponding $T_S(t_L)$ is found by following Section 3.3.1. If the obstacle is in the grey shaded region, illustrated in Figure 3.9, we can derive that

$$T_S(t_L) = \frac{r_{L_o} \cos\left(\frac{v_e}{R} t_L + \alpha_{L_o}\right) - \sqrt{(R-c)^2 - r_{L_o}^2 \sin^2\left(\frac{v_e}{R} t_L + \alpha_{L_o}\right)}}{v_e},$$

[7]The resulting expression from solving the differential equation can be further simplified by using harmonic addition theorem.

which is based on (3.19) and (3.7). This derivation is based on the substitution of the value for τ and the rearrangement of the expression $y_{inner}(\tau)$:

$$y_{inner}(\tau) = \sqrt{(R-c)^2 - r_{L_o}^2 \sin^2\left(\frac{v_e}{R}t_L + \alpha_{L_o}\right)}.$$

Max and Min Turning Time for the First L

As the evader continues to follow the nominal path after t_{reach}, there is a family of LS evasive actions that prevents the evader from colliding into the obstacle at each position along the nominal path like the one illustrated in Figure 3.15 which encompasses the obstacle. Let $T_{L_{max}}$ be the required duration for the evader to apply the first L maneuver of a LS/LSL evasive action so that the obstacle is directly above the point $(-c, 0)$ in the reduced space. Shown in Figure 3.9, the vertical line with its x-coordinate as $-c$ separates the red segment of a default escape route with the rest of the segments. If the evader continues to apply the first L maneuver beyond $T_{L_{max}}$, then it must turn towards the pursuer at some future moment in order to avoid the collision with the obstacle. By doing so, it defeats the purpose of our avoidance strategy which is to keep the evader from turning in the direction towards the pursuer. Therefore, $T_{L_{max}}$ is the upper limit for the evader to apply the first L maneuver amongst the family of LS/LSL evasive actions at any instant while the evader continues to follow the nominal path with the obstacle inside the default escape route. Let $T_{L_{min}}$ be the lower limit for the evader to apply the first L maneuver of a LS/LSL evasive action. Based on (3.19),

$$\begin{aligned} T_{L_{min}} &= 0 \\ T_{L_{max}} &= \frac{R}{v_e}\left(\arcsin\left(\frac{R-c}{\sqrt{(x_o(0)+R)^2 + (y_o(0))^2}}\right) - \arctan\left(\frac{x_o(0)+R}{y_o(0)}\right)\right), \end{aligned} \tag{3.20}$$

where $(x_o(0), y_o(0))$ is the obstacle's initial position right before the evader executing the first L maneuver.

Change of Default Escape Route and Duration of S Maneuver

Now, we know that

$$t_L \in [T_{L_{min}}, T_{L_{max}}].$$

According to the reversed homicidal chauffeur game [15], the pursuer must eventually be inside the evasion zone regardless of its action if the evader keeps turning hard left,

assuming that the pursuer is initially not in the collision zone. Therefore, as the evader turns hard left, the required duration for the evader to keep applying the L maneuver so that the pursuer gets inside the evasion zone decreases. The actual default escape route shrinks in size as the evader continues turning hard left.

Remark 6. *The default escape route before the execution of a LS/LSL evasive action is larger in size compared to the corresponding default escape route during a LS/LSL evasive action. Since we do not know about the pursuer's subsequent action during a LS/LSL evasive action prior to the LS/LSL, we will use the default escape route right before the execution of a LS/LSL evasive action to determine whether or not the LS/LSL evasive action is feasible.*

Conditions for Capture during a LS/LSL Maneuver

If the pursuer is on the right side of the evader and outside the collision zone at the moment that the evader turns hard left, based on the solution for the reversed homicidal chauffeur game [15], the pursuer cannot get inside the collision zone during the L maneuver.

Claim 1. *Consider the evader starts executing a LS/LSL evasive action at a moment t when the pursuer is on the right side of the nominal path and outside the collision zone. Then the pursuer can only enter the collision zone while the evader is on the S segment during its execution of the LS/LSL action.*

We know that the pursuer cannot enter the collision zone during the first L maneuver. While the evader executes the last L maneuver, the pursuer can only enter the collision zone if it is already in the collision zone at the moment when the evader starts executing the last L maneuver. If so, the pursuer must enter the collision zone during the second S maneuver of a LS/LSL. If the pursuer is not yet inside the collision zone at the end of the S segment, then the last L maneuver keeps the pursuer from entering the collision zone.

Based on Claim 1, two possibilities arise. One is that the evader can be captured during the S segment and the other is to have the pursuer inside the collision zone but not yet in the collision space at the end of the S segment. Therefore, the following two procedures are used to check the feasibility of a family of LS/LSL evasive actions at each instant while the evader continues to follow the nominal path with the obstacle inside the default escape route.

Conditions for Capture along the S Segment

In the realistic space, the evader's motion as it executes the first L maneuver of a LS/LSL evasive action is given by

$$\begin{aligned} x_e(t_L) &= R\sin\left(\phi_e(0) + \frac{v_e}{R}t_L\right) - R\sin(\phi_e(0)) + x_e(0) \\ y_e(t_L) &= -R\cos\left(\phi_e(0) + \frac{v_e}{R}t_L\right) + R\cos(\phi_e(0)) + y_e(0) \\ \phi_e(t_L) &= \phi_e(0) + \frac{v_e}{R}t_L, \end{aligned} \tag{3.21}$$

where

$$\begin{bmatrix} x_e(0) \\ y_e(0) \\ \phi_e(0) \end{bmatrix}$$

is the initial evader's state prior to the execution of the first L maneuver and $t_L \in [T_{L_{min}}, T_{L_{max}}]$. Let $t_S \in [0, T_S(t_L)]$ be the time variable for the evader to apply the second S maneuver of a LS/LSL evasive action. Based on (3.21), the evader's motion due to its second S maneuver is given by

$$\begin{aligned} x_e(t_S, t_L) &= v_e\cos\left(\phi_e(0) + \frac{v_e}{R}t_L\right)t_S + x_e(t_L) \\ y_e(t_S, t_L) &= v_e\sin\left(\phi_e(0) + \frac{v_e}{R}t_L\right)t_S + y_e(t_L). \end{aligned} \tag{3.22}$$

To check whether or not the pursuer can capture the evader during the second S maneuver, an evaluation is conducted to see if the pursuer's action circle, $\mathcal{C}_p$ overlaps the moving collision space as the evader moves along the S segment of a LS/LSL. Based on (3.22), the following expression is used for this evaluation:

$$(x_p(0) - x_e(t_S, t_L))^2 + (y_p(0) - y_e(t_S, t_L))^2 \leq (v_p(t_L + t_S) + c)^2, \tag{3.23}$$

where $(x_p(0), y_p(0))$ is the initial pursuer's location prior to the execution of the LS/LSL, $t_L \in [T_{L_{min}}, T_{L_{max}}]$, and $t_S \in [0, T_S(t_L)]$. A feasible LS/LSL evasive action amongst the family of LS/LSL evasive actions given by $t_L \in [T_{L_{min}}, T_{L_{max}}]$ must ensure that the inequality given in (3.23) does not occur.

The evaluation given by the expression in (3.23) can be solved as an optimization problem. Based on (3.23), the optimization problem is defined by the following max-min

problem:

$$\begin{aligned} \max_{t_L} \min_{t_S} \quad & (x_p(0) - x_e(t_S, t_L))^2 + (y_p(0) - y_e(t_S, t_L))^2 - (v_p(t_L + t_S) + c)^2 \\ \text{subject to} \quad & t_L \in [T_{L_{min}}, T_{L_{max}}] \\ & t_S \in [0, T_S(t_L)]. \end{aligned} \tag{3.24}$$

If the objective value in the above max-min problem is non-negative with the obtained t_L and t_S from solving the optimization problem in (3.24), then the obtained t_L defines a LS/LSL evasive action that may be feasible. Otherwise, there is no feasible LS/LSL evasive action in the family of LS/LSL evasive actions given by $t_L \in [T_{L_{min}}, T_{L_{max}}]$. To solve this problem numerically, we have used the grid search approach by first discretizing the values for both $t_L \in [T_{L_{min}}, T_{L_{max}}]$ and $t_S \in [0, T_S(t_L)]$ and to test each t_L with the condition (3.23). If there is a t_L such that all corresponding $t_S \in [0, T_S(t_L)]$ does not satisfy the condition given in (3.23), then the LS/LSL evasive action defined by this t_L may be feasible. Otherwise, the LS/LSL evasive action corresponding to this t_L must be infeasible. To fully determine whether a LS/LSL evasive action is feasible or not, the evaluation in the next subsection must be conducted to check whether or not there exists a configuration that the capture of the evader is inevitable in the future at the end of the S maneuver of a LS/LSL evasive action.

Conditions for Inevitable Capture at the end of the S Segment

In the realistic space, the motion of the transformed barrier surface is relative to the evader's movement and it is defined by (3.10) at a moment t. With $T_S(t_L)$ for each LS/LSL found in Section 3.3.1, the resulting collision zone and the corresponding pursuer's action circle at the end of the S segment can be obtained. Let $(x(T_S(t_L), \tau), y(T_S(t_L), \tau))$ be the expression for the transformed barrier surface at the end of the S segment of a LS/LSL evasive action where $\tau \in [0, \overline{\tau}]$ is the parameter of each point on the barrier surface.

After completing the evaluation given in the previous subsection, the candidate LS/LSL evasive actions are selected and they do not satisfy the condition in (3.23). The next process is to exhaustively search through the candidate LS/LSL evasive actions and to evaluate whether or not there is an overlap between the resulting collision zone and the corresponding pursuer's action circle at the end of the evader's S maneuver for each candidate LS/LSL evasive action. If the closest point on the barrier surface to the center of the pursuer's action circle is inside the circle, then this overlap exists. Otherwise, the overlap does not exist. With a defined $T_S(t_L)$, the expression below is the distance formula between each

point on the barrier surface[8] and the center of the pursuer's action circle at the end of the evader's S maneuver for each LS/LSL:

$$f(t_L, \tau) = (x_p(0) - x(T_S(t_L), \tau))^2 + (y_p(0) - y(T_S(t_L), \tau))^2 - v_p^2(t_L + T_S(t_L))^2, \quad (3.25)$$

where $(x_p(0), y_p(0))$ is the initial pursuer's position in the realistic space prior to the execution of the LS/LSL, $t_L \in [T_{L_{min}}, T_{L_{max}}]$, and $\tau \in [0, \overline{\tau}]$. After taking the partial derivative of (3.25) with respect to τ, the closest point on the transformed barrier surface at the end of the S segment of a LS/LSL evasive action to the center of the pursuer's action circle is obtained:

$$\frac{\partial f(t_L, \tau)}{\partial \tau} = 0,$$

where the relationship between t_L and the variable τ which represents the closest point on the barrier surface can be found. With a defined t_L, a corresponding τ value can be obtained. By utilizing this relationship and (3.25), the search[9] through the candidate LS/LSL evasive actions can be conducted to check whether or not there is a LS/LSL evasive action where its corresponding closest point on the barrier surface is outside the resulting pursuer's action circle. If the evaluation is true, then the corresponding LS/LSL evasive action is feasible.

Checking Existence of Feasible LS/LSL at time t^+

With regard to Remark 3, to check for the future feasible evasive action, both $x_o(0)$ and $y_o(0)$ in (3.20) and in Section 3.3.1 for finding $T_S(t_L)$ of a LS/LSL are the obstacle's position at t^+. Also the evader's initial position becomes the evader's position at t^+ but the pursuer's initial position stays the same. The resulting pursuer's action circle expands because it needs to consider the space where the pursuer can reach within δt. Therefore, the expression in (3.23) that is used to analyze whether or not the pursuer's action circle overlaps the moving collision space at t^+ changes to

$$(x_p(0) - x_e(t_S, t_L))^2 + (y_p(0) - y_e(t_S, t_L))^2 \leq (v_p(t_L + t_S + \delta t) + c)^2, \quad (3.26)$$

[8]Since the pursuer is on the right side of the evader before executing a LS/LSL, the pursuer must enter $\mathcal{CZ}$ through the right side barrier surface with respect to the evader's heading.

[9]To solve this problem numerically, we have used the grid search again to go through the t_L value for all candidate LS/LSL evasive actions.

where the initial evader's and the obstacle's positions correspond to their positions at t^+. Also, the expression in (3.25) changes

$$f(t_L, \tau) = (x_p(0) - x(T_S(t_L), \tau))^2 + (y_p(0) - y(T_S(t_L), \tau))^2 - v_p^2(t_L + T_S(t_L) + \delta t)^2, \tag{3.27}$$

where $x(T_S(t_L), \tau)$ and $y(T_S(t_L), \tau)$ change based on the evader's and obstacle's positions at t^+. The evaluation procedure to determine the existence of future feasible LS/LSL at t^+ stays the same as the evaluation procedure for the feasible LS/LSL at the current moment by following the previous two subsections.

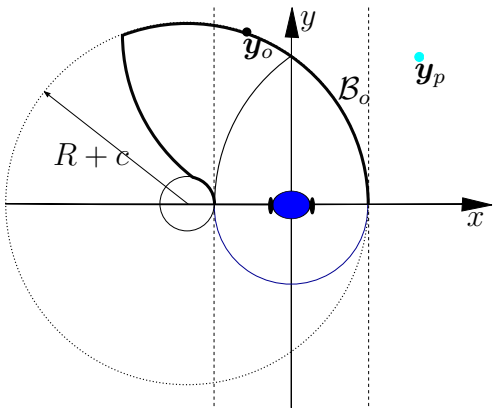
Triggering of Event 2 or 3 in Table 3.1

If there is no feasible LS/LSL amongst the family of LS/LSL evasive actions at t_{reach} or t^+_{reach} by following the evaluation procedures in the above three subsections, then Event 2 is triggered. Otherwise, the evader keeps following the nominal path. At each subsequent moment t after t_{reach} while the obstacle is still inside the default escape route, the evaluation of feasible LS/LSL amongst the family of LS/LSL evasive actions at t^+ is conducted. If there is no feasible LS/LSL, then Event 3 is triggered.

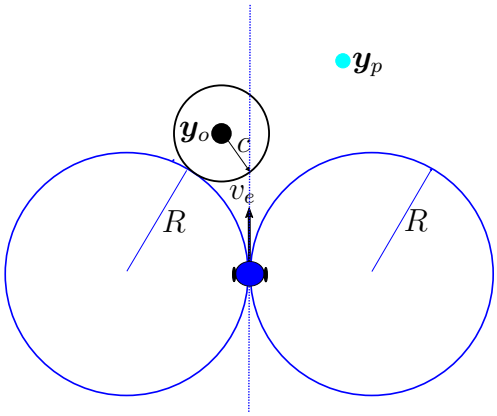
3.4.4 Determine the Existence of Feasible RL/RSL

Our thesis mainly focuses on evaluating the existence of feasible LS/LSL with the obstacle not interfering the nominal path, illustrated in Figure 3.14, in Section 3.4.3. With the obstacle interfering the nominal path, the evaluation of the existence of feasible RL/RSL is related to determine whether or not Event 2 or 4 in Table 3.1 is triggered and it follows a similar manner as for the evaluation of feasible LS/LSL in Section 3.4.3.

The first R maneuver and the second S maneuver are used to avoid the obstacle by forcing the obstacle to leave the default escape route. Since the pursuer is on the right side of the evader at the beginning of a RL/RSL, the evasive action should avoid turning towards the pursuer for too long. Therefore, the S maneuver is used to let the obstacle get out of the default escape route so that the last L maneuver of a RSL evasive action is safe to use for evading the pursuer if necessary. However, the first R maneuver can force the pursuer to get onto the same side as the obstacle. As a result, the evader can keep executing this R maneuver to eventually evade both the pursuer and the obstacle since they are now all on the left side with respect to the evader. The last L maneuver of a RL evasive action is used to bring the evader back to the nominal path.



(a) *Based on Figure 3.4a, the configuration of all players in the reduced space at t_{reach}; The x-coordinate value of the obstacle is in between $-c$ and 0.*



(b) *The configuration of all players in the realistic space at t_{reach} when the collision disc centered at the obstacle touches the evader's left minimum turning circle and the nominal path (blue dashed line) intersects the collision disc.*

Figure 3.16: *The configuration of all players at t_{reach} when the obstacle (black dot) is on the boundary of the default escape route and interferes the nominal path.*

Conditions for Capture while Executing a RL

Illustrated in Figure 3.10, the first R maneuver of the RL evasive action forces the obstacle out of the default escape route. In the realistic space, the obstacle's motion as it executes the first R maneuver of a RL/RSL is given by

$$\begin{aligned} x_e(t_R) &= R\sin\left(\frac{v_e}{R}t_R - \phi_e(0)\right) + R\sin(\phi_e(0)) + x_e(0) \\ y_e(t_R) &= R\cos\left(\frac{v_e}{R}t_R - \phi_e(0)\right) - R\cos(\phi_e(0)) + y_e(0) \\ \phi_e(t_R) &= \phi_e(0) - \frac{v_e}{R}t_R, \end{aligned} \tag{3.28}$$

where

$$\begin{bmatrix} x_e(0) \\ y_e(0) \\ \phi_e(0) \end{bmatrix}$$

is the initial evader's state prior to the execution of the first R maneuver and t_R is the duration of this R maneuver. Based on (3.10), let $(x(t_R, \tau), y(t_R, \tau))$ be the expression for the barrier surface in the realistic space during the evader's R maneuver. The distance formula between each point on the barrier surface and the center of the pursuer's action circle while the evader executes the first R maneuver is given by

$$f(t_R, \tau) = (x_p(0) - x(t_R, \tau))^2 + (y_p(0) - y(t_R, \tau))^2 - (v_p t_R)^2, \tag{3.29}$$

where $(x_p(0), y_p(0))$ is the initial pursuer's position prior to the execution of the RL evasive action. By taking the partial derivative of (3.29) with respect to τ, the closest point on the barrier surface to the center of the resulting pursuer's action circle is found:

$$\frac{\partial f(t_R, \tau)}{\partial \tau} = 0,$$

where the relationship between t_R and the variable τ which now represents the closest point on the barrier surface can be found. With a defined t_R, a corresponding τ value can be obtained. We know that $\tau \in [0, \overline{\tau}]$ and the range for t_R is found in the following subsection. After obtaining the range for t_R, a search can be conducted to check whether or not there is an overlap between the resulting collision zone and the pursuer's action circle. If there is an overlap, then the closest point on the barrier surface to the center of the pursuer's action circle must be inside the circle due to the evader's R maneuver. Otherwise, the evader continues to apply the R maneuver until the obstacle is outside of the default escape route and the evader can then switch to the L maneuver to get back onto the nominal path.

Max and Min Turning Time for the First R

When the evader turns hard right, the pursuer may get to anywhere on the barrier surface. Therefore, the max bound of t_R is the time for the evader to turn hard right so that the obstacle is out of the interior of the largest possible default escape route found by using $\overline{\tau}$ and let $T_{R_{max}}$ be this max bound. Any duration that is required for the evader to turn hard right to force the obstacle to get out of the default escape route is less than $T_{R_{max}}$. $T_{R_{max}}$ can be found by following the procedure in Section 3.3.1 and using the current pursuer's position at the moment of evaluation.

Let $T_{R_{min}}$ be the required duration for the evader to turn hard right so that the obstacle is directly above the point $(-c, 0)$ in the reduced space. Only when the x-coordinate value of the obstacle is less or equal to $-c$, the evader can switch to the S maneuver so that the

obstacle can leave the default escape route by arriving on the inner concentric or inside the static evasion zone. In either situation, the threat of colliding into the obstacle is eliminated at the end of the S segment so that the game becomes a reversed homicidal chauffeur. Therefore, the last hard left turn can be used to achieve the guaranteed evasion. $T_{R_{min}}$ is found by the expression below which is based on (3.9):

$$T_{R_{min}} = \frac{R}{v_e}\left(\arccos\left(\frac{-R-c}{r_{R_o}}\right) - \alpha_{R_o}\right), \tag{3.30}$$

and $T_{R_{min}}$ must satisfy the following condition

$$\sin\left(\frac{v_e}{R}T_{R_{min}} + \alpha_{R_o}\right) = 0,$$

where r_{R_o} and α_{R_o} are found by using the exact procedure as in (3.9). Conclusively,

$$t_R \in [T_{R_{min}}, T_{R_{max}}].$$

Duration of S maneuver for each RSL Evasive Action

Let $T_S(t_R)$ denote the duration of the S maneuver after switching from the initial R maneuver so that the obstacle eventually leaves the interior of the default escape route. With a defined t_R, $(x_o(t_R), y_o(t_R))$ is the obstacle's initial relative position after the evader executing the first R maneuver and before the S maneuver. For each RSL evasive action, its corresponding $T_S(t_R)$ is found by following the derivation for time to leave the default escape route with the S maneuver in Section 3.3.1.

Conditions for Capture during a RSL Maneuver

After executing the first R and the second S maneuvers of a feasible RSL, the evader wants to avoid the situation of having the obstacle on the left side static barrier surface while the pursuer is on the right side barrier surface. If the pursuer and the obstacle are on their respective barrier surfaces, then the capture by the pursuer or the collision with the obstacle becomes inevitable.

When determining the feasibility of a RL, if there is an overlap between the collision zone and the resulting pursuer's action circle due to the R maneuver, then there must exist an upper limit for the duration of the R maneuver amongst $t_R \in [T_{R_{min}}, T_{R_{max}}]$ for which the pursuer's action circle just touches the barrier surface in the realistic space.

This upper limit can be found by exhaustively search through $t_R \in [T_{R_{min}}, T_{R_{max}}]$. Let $T_{R_{up}} \in [T_{R_{min}}, T_{R_{max}})$ be the corresponding value for t_R at which the pursuer's action circle just touches the barrier surface. If there still exists a feasible evasive action with the obstacle inside the default escape route, then the feasible evasive action must be RSL where its $t_R \in [T_{R_{min}}, T_{R_{up}}]$. To evaluate whether or not there exists a feasible RSL in the family of RSL given by $t_R \in [T_{R_{min}}, T_{R_{up}}]$ at the moment when the obstacle is on the nominal path, the procedures in the following two subsections are used.

Conditions for Capture along the S Segment

The evaluation for checking whether or not the pursuer can enter the collision zone during the S maneuver follows the method of evaluating capture along the S segment in Section 3.4.3. Based on (3.28), the evader's motion as it executes the second S maneuver of a RSL evasive action is given by

$$\begin{aligned} x_e(t_S, t_R) &= v_e \cos\left(\phi_e(0) + \frac{v_e}{R} t_R\right) t_S + x_e(t_R) \\ y_e(t_S, t_R) &= v_e \sin\left(\phi_e(0) + \frac{v_e}{R} t_R\right) t_S + y_e(t_R), \end{aligned} \tag{3.31}$$

where $t_R \in [T_{R_{min}}, T_{R_{up}}]$ and $t_S \in [0, T_S(t_R)]$.

Based on (3.31), the expression below is used to evaluate whether or not the pursuer's action circle intersects the collision space during the S maneuver:

$$(x_p(0) - x_e(t_S, t_R))^2 + (y_p(0) - y_e(t_S, t_R))^2 \leq (v_p(t_R + t_S) + c)^2, \tag{3.32}$$

where $(x_p(0), y_p(0))$ is the initial pursuer's position prior to the execution of the first R maneuver, $t_R \in [T_{R_{min}}, T_{R_{up}}]$ and $t_S \in [0, T_S(t_R)]$. The evaluation in (3.32) can be solved by the following optimization problem:

$$\begin{aligned} \max_{t_R} \min_{t_S} \quad & (x_p(0) - x_e(t_S, t_R))^2 + (y_p(0) - y_e(t_S, t_R))^2 - (v_p(t_R + t_S) + c)^2 \\ \text{subject to} \quad & t_R \in [T_{R_{min}}, T_{R_{up}}] \\ & t_S \in [0, T_S(t_R)]. \end{aligned} \tag{3.33}$$

If the objective value in the above optimization problem is non-negative with the obtained t_R and t_S from solving the optimization problem in (3.33), then the obtained t_R defines a RSL evasive action that may be feasible. Otherwise, there is no feasible RSL evasive action amongst the candidates given by $t_R \in [T_{R_{min}}, T_{R_{up}}]$. To solve this problem numerically, we

have used the grid search approach by discretizing the values for both $t_R \in [T_{R_{min}}, T_{R_{up}}]$ and $t_S \in [0, T_S(t_R)]$ and to test whether or not all of the values for both t_R and t_S satisfy the condition given in (3.32). For each $t_R \in [0, T_S(t_R)]$ such that the pursuer's action circle does not ever intersect the collision zone during the S maneuver, it is a candidate RSL and each candidate RSL is then tested by following the next subsection in order to fully determine whether or not this candidate RSL is feasible.

Conditions for Inevitable Capture at the end of the S Segment

The motion of the transformed barrier surface is relative to the evader's motion under the RSL evasive action and it is defined by (3.10) at a moment t. Let $(x(T_S(t_R), \tau), y(T_S(t_R), \tau))$ be the expression for the transformed barrier surface at the end of the S segment of a RSL evasive action where $\tau \in [0, \overline{\tau}]$ is the parameter for each point on the barrier surface. By following what has been discussed about the inevitable capture at the end of the S segment in Section 3.4.3, the next process is to exhaustively search through all candidate RSL evasive actions from the previous evaluation. This search[10] checks whether or not there is an overlaps between the resulting pursuer's action circle and the corresponding collision zone at the end of the evader's S maneuver for each candidate RSL.

With a defined t_R and by following the procedure in Section 3.3.1, $T_S(t_R)$ is found. The distance formula between each point on the barrier surface and the center of the resulting pursuer's action circle at the end of the evader's S maneuver for each RSL is given by

$$f(t_R, \tau) = (x_p(0) - x(T_S(t_R), \tau))^2 + (y_p(0) - y(T_S(t_R), \tau))^2 - v_p^2(t_R + T_S(t_R))^2, \quad (3.34)$$

where $(x_p(0), y_p(0))$ is the initial pursuer's position prior to the execution of the first R maneuver of the RSL evasive action, candidate $t_R \in [T_{R_{min}}, T_{R_{up}}]$, and $\tau \in [0, \overline{\tau}]$. After taking the partial derivative of (3.34) with respect to τ, the closest point on the transformed barrier surface at the end of the S segment to the center of the pursuer's action circle is obtained:

$$\frac{\partial f(t_R, \tau)}{\partial \tau} = 0,$$

where the relationship between candidate t_R and the variable τ which represents the closest point on the barrier surface can be found. If there is a t_R value amongst the candidate RSL evasive actions where its corresponding closest point on the barrier surface at the end of the S segment is outside the resulting pursuer's action circle, then RSL that corresponds to this t_R value is feasible. Otherwise, it is infeasible.

[10]To solve this problem numerically, we have used the grid search again by checking all candidate t_R.

Checking Existence of Feasible RL/RSL at time t^+

With regard to Remark 3, the initial positions of both the evader and the obstacle are their respective positions at t^+. However, the pursuer's initial position stays the same. These changes affect both $T_{R_{max}}$ and $T_{R_{min}}$. With the change in the obstacle's and the evader's initial positions to their positions at t^+, (3.28), (3.29), (3.31), (3.32), and (3.34) need to be adjusted accordingly. For the new expressions for (3.29), (3.32), and (3.34), the expansion of the resulting pursuer's action circle must consider the space where the pursuer can reach within δt. The new expressions for (3.29), (3.32), and (3.34) are respectively given by:

$$f(t_R, \tau) = (x_p(0) - x(t_R, \tau))^2 + (y_p(0) - y(t_R, \tau))^2 - (v_p(t_R + \delta t))^2, \tag{3.35}$$

$$(x_p(0) - x_e(t_S, t_R))^2 + (y_p(0) - y_e(t_S, t_R))^2 \leq (v_p(t_R + t_S + \delta t) + c)^2, \tag{3.36}$$

and

$$f(t_R, \tau) = (x_p(0) - x(T_S(t_R), \tau))^2 + (y_p(0) - y(T_S(t_R), \tau))^2 - (v_p(t_R + T_S(t_R) + \delta t))^2, \tag{3.37}$$

where the initial evader's and the obstacle's positions are their positions at t^+. The evaluation procedure to determine the existence of future feasible RL/RSL at t^+ stays the same as the evaluation procedure for the feasible RL/RSL at the current moment by following the previous subsections in Section 3.4.4.

Triggering of Event 2 or 4 in Table 3.1

Except the changing of the wording from LS/LSL to RL/RSL, the triggering of Event 2 or 4 follows the same logic as the triggering of Event 2 or 3 in Section 3.4.3.

Chapter 4

Simulation Studies

In this chapter, we present simulation results based on the evasive strategy from Chapter 3 and a comparison study between our event based collision avoidance strategy and the collision avoidance strategy based on the velocity obstacle set [18]. The collision avoidance strategy based on the ISO 13482 [22] is also discussed in the comparison study. These two collision avoidance strategies are commonly used in the current robotic industry. All simulations are implemented in MATLAB. We used functions such as $vpasolve(\cdot)$ and $fmincon(\cdot)$ to solve nonlinear system of equations.

According to the study in [8], the average pedestrian walking speed is 1.4 m/s. However, considering that usually a worker inside a warehouse either carries a load or drags a pallet jack, the walking speed should be lower than what is reported in [8]. For this simulation study, we assume that the pursuer's constant speed is 1.2 m/s. The evader's speed is assumed to be 2.0 m/s based on information of some of the mobile robots such as OTTO 100 [2] and Omron LD series [1]. The minimum turning radius of the evader is set to 0.8 m and the radius of the collision space is set to 0.6 m. By setting the speed of the evader as the unit value 1, the following shows the resulting value for the speed of the pursuer, the minimum turning radius of the evader, and the radius of the collision space used in the simulation:

$$v_e = 1, \quad v_p = 0.6, \quad R = 0.8, \quad c = 0.6.$$

In the simulation, the pursuer is on the right side with respect to the nominal path and the obstacle is on the left side with respect to the nominal path at the start of the PSO game. The evader follows the nominal path with a pure pursuit controller [12] and the evader switches to our evasive control strategy when one of the events from Table 3.1 is triggered. After achieving the guaranteed evasion and hence the PSO game terminates,

the evader switches back to the pure pursuit controller in order to get back on the nominal path. Based on the common behaviours of the human agents inside a warehouse, we create three scenarios for how the pursuer moves in a PSO game: diagonal cross (first move downwards and then move diagonally from its position on the right side of the nominal path to the left side of the nominal path), straight cross (first move downwards and then move from its position on the right side of the nominal path to the left side of the nominal path along a straight line), and downward move (move downwards along the path that is parallel to the nominal path on the right side). Since the nominal path is assumed to be a line, the evader's initial position is fixed at the origin in the realistic space. The nominal path is fixed along the positive y-axis in the realistic space and the evader's heading is initially fixed at $\frac{\pi}{2}$. In the simulations, the nominal path is set to 6.5 unit length.

4.1 Simulation Results

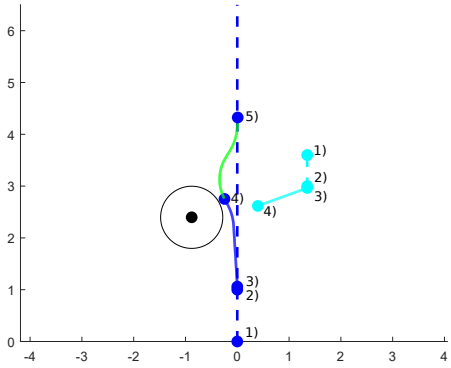
In [25], the authors identify some of the metrics that can be used to evaluate the performance of a mobile robot in a navigation task. Our evaluation is based on the metrics such as the mission duration and the path error due to the deviation from the nominal path for either evading the obstacle or the pursuer. The mission duration is the time that the evader takes to get from its initial position to the destination. Define the path error as

$$err = \int_0^{t_f} |x_e(t)|\, dt, \tag{4.1}$$

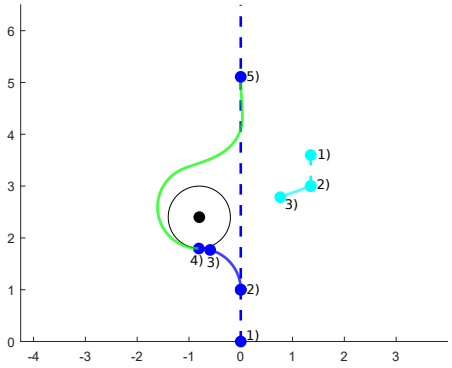
where t_f is the instant that the evader arrives at the destination point and the nominal straight path is assumed to be aligned with the y-axis in the realistic space.

4.1.1 Obstacle Not Interfering with the Path

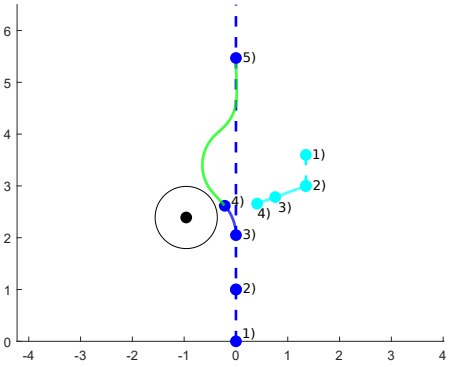
In Figure 4.1, the pursuer performs the diagonal cross maneuver and tries to use the obstacle to catch the evader by cutting through the nominal path ahead of the evader and thus closing the gap between the pursuer and the obstacle. Figure 4.1 also presents the three types of evasive actions in the scenario that the obstacle does not interfere with the nominal path. The blue and cyan dots numbered 1 and 2 illustrate, respectively, the initial positions of the evader and the pursuer and the positions of the evader and the pursuer at t_{reach} when the obstacle is on boundary of the default escape route given by (3.6) and (3.8) (see Section 3.4.2 for the definition of t_{reach}).



(a) *LS/LSL evasive action.*



(b) *Turn hard left under the obstacle.*



(c) *Turn hard left above the obstacle.*

Figure 4.1: *For obstacle not interfering the path, the simulation result shows how each evasive maneuver achieves the guaranteed evasion where the blue and cyan dots represent respectively the evader and the pursuer at different timestamps.*

Under each scenario which is based on the pursuer's movement, we tested our event based collision avoidance strategy up to 1000 iterations. For each test, the obstacle and the pursuer were randomly placed in the proximity of the evader. In Table 4.1, we present the number of occurrences for the evader to execute each type of evasive action, the task duration, and the path error for each type of the pursuer's movement.

Both LS/LSL and turning hard left above the obstacle based on the solution for the reversed homicidal chauffeur game avoid the obstacle by going above it. Both actions prevent the pursuer from getting inside the collision zone, $\mathcal{CZ}$. By going above the obstacle,

Table 4.1: *With the obstacle not interfering the nominal path, our event based collision avoidance strategy is tested with different pursuer's movement.*

Pursuer Movement	Type of Evasive Action	Number of Occurrences	Task Duration		Path Error	
			Mean	STD	Mean	STD
Diagonal cross maneuver	Turn hard left under the obstacle	322	9.34	2.04	3.61	2.50
	LS/LSL evasive action	16	7.96	1.73	2.33	2.47
	Turn hard left above the obstacle	662	6.94	0.77	0.72	1.14
Straight cross maneuver	Turn hard left under the obstacle	314	10.37	2.96	4.81	3.51
	LS/LSL evasive action	25	8.23	1.70	3.05	2.69
	Turn hard left above the obstacle	661	7.15	0.88	1.24	1.51
Downwards maneuver	Turn hard left under the obstacle	290	8.14	0.88	2.14	1.15
	LS/LSL evasive action	19	6.60	0.10	0.46	0.39
	Turn hard left above the obstacle	691	6.55	0.22	0.11	0.37

Table 4.2: *The difference in task duration and path error between taking the default escape route and executing the final feasible LS/LSL for games where a feasible future LS/LSL exists.*

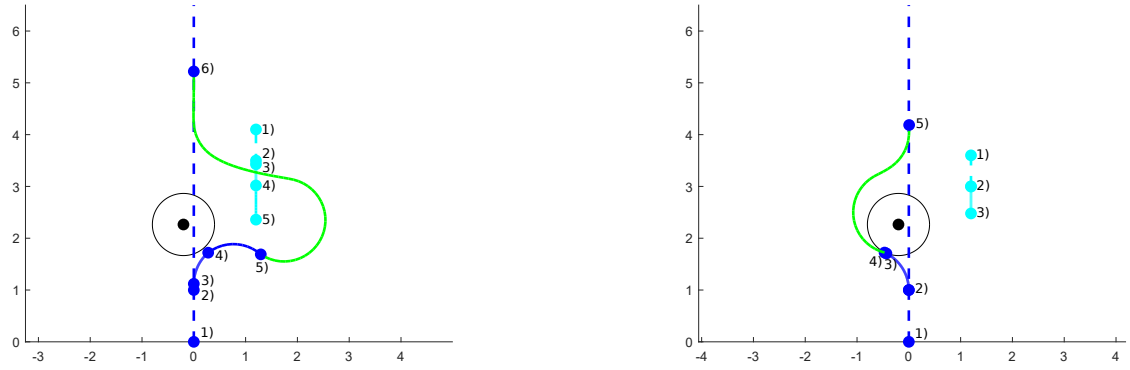
Pursuer Movement	Average Wait Time by LS/LSL	Task Duration		Path Error	
		Mean Diff	% Reduction	Mean Diff	% Reduction
Diagonal cross maneuver	0.75	0.84	9.55	2.44	51.15
Straight cross maneuver	0.73	0.64	7.22	2.30	42.99
Downwards maneuver	0.20	1.65	20.00	2.29	83.27

the evader is in between the obstacle and the nominal path after successfully evading the pursuer. Therefore, the evader takes much less time for it to get back on the nominal path after the PSO game terminates with the pursuer inside the evasion zone compared to the evader's action of turning hard left below the obstacle at t_{reach}. This observation is demonstrated in Table 4.1 where both the average task duration and the average path error for evading the pursuer by going above the obstacle are significantly lower than the average task duration and the path error for turning hard left under the obstacle. Also, Table 4.2 demonstrates the reduction in both the average task duration and the average path error resulting from the evader's execution of the LS/LSL evasive action compared to the earlier hard left under the obstacle maneuver in the case that the future feasible LS/LSL evasive action exists. The reasoning behind the significant differences in both the average task duration and the path error is that the evader needs to turn around the obstacle in order to get back to the nominal path after turning hard left under the obstacle

at t_{reach} and this results in the increase of both the task duration time and path error, illustrated in Figure 4.1b.

Illustrated in Figure 4.1, the blue and cyan dots with label 4, respectively, represent the locations of the evader and the pursuer at where the guaranteed evasion occurs. The solid blue curve shows each respective path taken by the evader to achieve the guaranteed evasion. The green solid curve is the path taken by the evader to get back on the nominal path as result of using the pure pursuit controller after achieving the guaranteed evasion.

4.1.2 Obstacle Interfering the Path



(a) *RL/RSL evasive action.* (b) *Turn hard left under the obstacle.*

Figure 4.2: *For obstacle interfering the path, the simulation result shows how each evasive maneuver achieves the guaranteed evasion where the blue and cyan dots represent respectively the evader and the pursuer at different timestamps.*

As for the case where the obstacle interferes with the nominal path, the evader must avoid the obstacle by either turning just under the obstacle to the left of the nominal path or using the RL/RSL evasive action, illustrated in Figure 4.2. The evader always tries to execute the "latest" RL evasive action which is to wait until the obstacle arrives on the static barrier surface, then to turn hard right to avoid the obstacle at the "last" moment, and to turn hard left to avoid the pursuer after successfully avoiding the obstacle. However, the final RL maneuver may be infeasible and therefore cannot be used as an evasive strategy to achieve the guaranteed evasion. In Figure 4.2, the second blue and cyan dots illustrate, respectively the positions of the evader and the pursuer at t_{reach} which is the last moment for the evader to be able to turn hard left below the obstacle.

Table 4.3: *With the obstacle interfering the nominal path, our event based collision avoidance strategy is tested with different pursuer's movement.*

Pursuer Movement	Type of Evasive Action	Number of Occurrences	Task Duration		Path Error	
			Mean	STD	Mean	STD
Diagonal cross maneuver	Turn hard left under the obstacle	363	7.97	0.71	2.09	0.66
	RL/RSL evasive action	608	6.80	0.59	0.90	0.89
	Final RL evasive action	29	6.78	0.55	0.94	0.76
Downwards maneuver	Turn hard left under the obstacle	369	7.89	0.65	1.78	0.39
	RL/RSL evasive action	596	7.30	1.05	1.58	1.45
	Final RL evasive action	35	7.67	1.19	2.25	1.61

Table 4.4: *The difference in task duration and path error between taking the default escape route and executing the final feasible RL/RSL for games where a feasible future RL/RSL exists.*

Pursuer Movement	Average Wait Time by RL/RSL	Task Duration		Path Error	
		Mean Diff	% Reduction	Mean Diff	% Reduction
Diagonal Cross Maneuver	0.18	0.79	10.43	1.71	65.55
Downwards Maneuver	0.21	-0.12	-1.63	-0.13	-8.62

As the evader keeps following the nominal path after t_{reach}, Figure 4.2a demonstrates that the evader executes a RL maneuver to avoid both the obstacle and the pursuer. If it continues to follow the nominal path beyond the moment that it should execute the RL maneuver, the pursuer can "squeeze" the evader to collide with the obstacle. Under each scenario which is based on the pursuer's movement, we tested our event based collision avoidance strategy in the same manner as for the obstacle not interfering the path. In Table 4.3, we present the number of occurrences for the evader to execute each type of evasive action, the task duration, and the path error for each type of the pursuer's movement.

In Table 4.3 and Table 4.4, both the average task duration and the average path error for the RL/RSL evasive action are significantly lower compared to the hard left maneuver at t_{reach} if the pursuer executes the diagonal cross maneuver. By performing the diagonal cross maneuver, the pursuer seems to be evading the evader as the evader executes the RL/RSL evasive action. Apparently, the pursuer passes along the side of the evader by moving in the opposite direction. Therefore, the PSO game soon terminates with the pursuer arriving in the evasion zone, $\mathcal{EZ}$ with respect to the evader. Hence, the evader can switch back to the pure pursuit controller to get back onto the nominal path as soon as the PSO game terminates. If the pursuer executes the straight cross maneuver, the pursuer

also seems to be evading the evader. Therefore, similar observations of both the average task duration and the average path error for the evader executing the RL/RSL evasive action can be made.[1]

Depending on the pursuer's maneuver, the task duration and the path error may become larger for the evader executing the RL/RSL evasive action compared to executing the hard left maneuver at t_{reach} if the pursuer keeps moving downwards. The downward maneuver resembles the pursuer's action of actively trying to catch the evader as the evader executes the RL/RSL evasive action. Hence, the opposing hard turning maneuver, R, prevents the pursuer from entering the collision zone, $\mathcal{CZ}$. If the pursuer performs a diagonal cross maneuver, Table 4.4 demonstrates the reduction in both the average task duration and the average path error resulting from the evader's execution of the RL/RSL evasive action compared to the earlier hard left under the obstacle maneuver in the case that the future feasible RL/RSL evasive action exists. However, if the pursuer performs a downwards maneuver, the hard left maneuver starting at t_{reach} gives better performance in a simple material transport task than the execution of the future feasible RL/RSL evasive action.

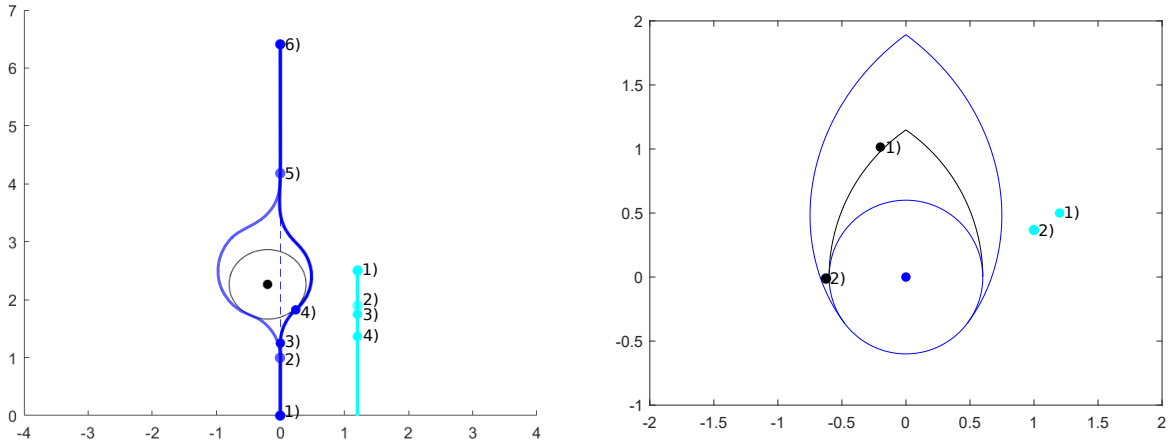
In Figure 4.2a, the blue solid curve represents the evader's movement during its performance of an evasive action and the green solid curve shows the evader's movement to get back on the nominal path. The fifth blue and cyan dots illustrate, respectively, the positions of the evader and the pursuer at the moment that the PSO game terminates. Then the evader switches back to the pure pursuit controller to get back on the nominal path and to continue with the material transporting task.

4.2 Comparison Studies

In this section, we will be comparing the effectiveness of our evasive policy with the work presented in [18] and [22].

The benefit of being able to let the evader operate closely to the pursuer is that the path error is expected to be small and hence the task duration is expected to be close to the task duration of just following the nominal path. Illustrated in Figure 4.3, the pursuer executes the downwards maneuver and is not actively trying to pursue or capture the evader. Therefore, the trajectory of the evader is not affected by the velocity obstacle set of the pursuer and therefore can wait till the "last moment" to make the evasive action when the evader's trajectory is affected by the velocity obstacle set. This clearly shows

[1]The results for the simulation of the pursuer's straight cross maneuver in a PSO game where the obstacle interferes the nominal path are neglected.



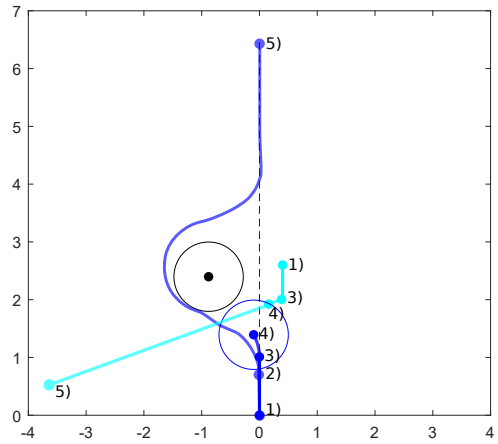
(a) *Compare our evasive strategy (light blue path) and the evasive strategy [18] (dark blue path) to evade both the pursuer and the obstacle.*

(b) *The positions of the pursuer (cyan) and the obstacle (black) in the reduced space that correspond to the start (1) of executing the evasive strategy [18] and the end (2) of the evasive strategy.*

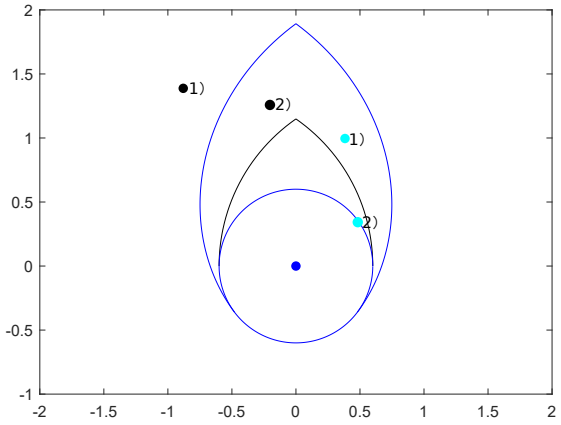
Figure 4.3: *The first comparison example between the evasive strategies based on the velocity obstacle and solving the PSO game.*

that the evasive strategy based on checking the velocity obstacle set results in a lower value for the task duration and has a lower path error compared to our evasive policy in which the evader reacts earlier shown at the time stamp numbered 2 by turning hard left. Comparing results in Table 4.1 and Table 4.3 with Table 4.5 and Table 4.6, the average task duration and the path error are significantly lower for the mobile robot that uses the evasive strategy based on checking the velocity obstacle set. Even for cases where the obstacle does not affect the evader from avoiding the pursuer by turning hard left until the pursuer is on the barrier surface, both the average task duration and the path error of using our strategy are significantly higher compared to using the strategy of checking the velocity obstacle set.

A clear disadvantage of using the evasive strategy based on checking the velocity obstacle set is that the pursuer may not be cooperative and therefore can collide with the evader. This is observed in the case presented in Figure 4.4 where the evader eventually collides with the pursuer by following the evasive procedure [18]. In Figure 4.4b, it clearly shows that the pursuer is already inside the collision zone at the moment when the evader



(a) *Compare our evasive strategy (light blue path) and the evasive strategy [18] (dark blue path).*

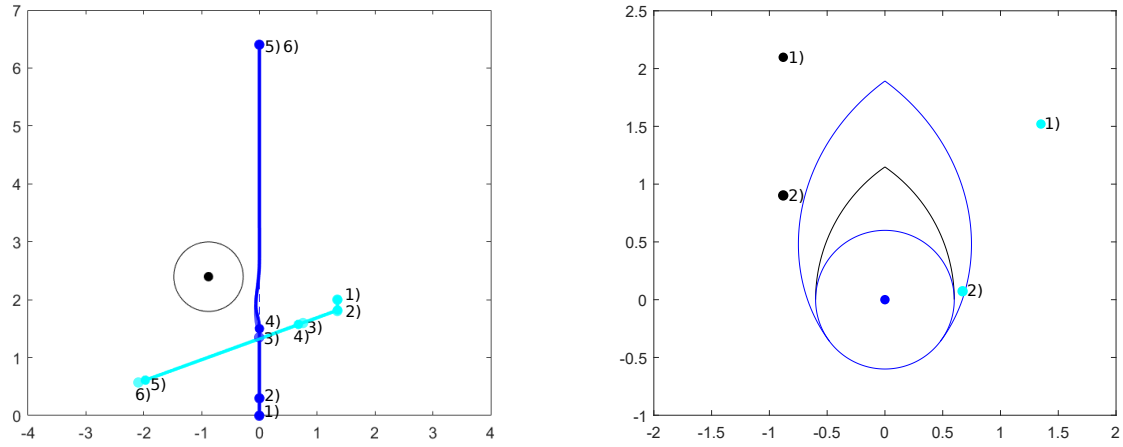


(b) *The positions of the pursuer (cyan) and the obstacle (black) in the reduced space that correspond to the start (1) of executing the evasive strategy [18] and the eventual capture (2) by the pursuer.*

Figure 4.4: *The second comparison example between the evasive strategies based on the velocity obstacle and solving the PSO game.*

starts to avoid the pursuer. Based on the work in [15], there is a strategy that the pursuer can use to capture the evader regardless of what the evader does to avoid the pursuer. Even though in general our evasive strategy causes a much higher time duration and path error, it guarantees the safety of the evader, illustrated in Figure 4.4a with the light blue path. In a warehouse, the human worker may have no real intention of colliding with the mobile robot. However, the diagonal cross maneuver may be an unintentional move such as the human worker walking backwards with his back to the robot. In Table 4.5, and Table 4.6, the success rate of the evader to reach the destination without colliding into either the obstacle or the pursuer is not 100% and it depends on whether the pursuer's action also tries to avoid colliding with the evader. This indicates the level of cooperation that the pursuer has when facing the evader.

In Figure 4.5, the LSL evasive strategy that the evader executes is identical to the evasive strategy [18]. Hence, their task duration and the path error are similar. However, the LSL evasive strategy is executed at an earlier time (the blue dot with label 3) compared to the moment of executing the strategy [18] (the blue dot with label 4). In Figure 4.5b, the



(a) *Compare our evasive strategy (light blue path) and the evasive strategy [18] (dark blue path).*

(b) *The positions of the pursuer (cyan) and the obstacle (black) in the reduced space that correspond to the start (1) of the pursuer crossing diagonally to the opposite side of the nominal path and the start (2) of the evader executing the evasive strategy [18].*

Figure 4.5: *The third comparison example between the evasive strategies based on the velocity obstacle and solving the PSO game.*

pursuer's position at the moment that the evader is about to execute the evasive strategy [18] is shown by the label 2 and it is clearly inside the collision zone, $\mathcal{CZ}$. However, the pursuer does not actively pursue the evader and thus the strategy based on checking the velocity obstacle set is able to successfully evade the pursuer. This illustrates that the pursuer's action is cooperative in avoiding the collision with the evader.

Comparing the two evasive strategies, the evasive strategy that continuously checks the velocity obstacle sets allows the evader to follow the nominal path better. Therefore, both the task duration and the path error are lower compared to our evasive strategy of solving a PSO game. However, the evasive strategy [18] cannot guarantee that the collision between the pursuer and the evader will not happen and our evasive strategy guarantees that the evader will not be captured by the pursuer or collide into the obstacle (see Theorem 1). Therefore, there is a clear trade off between ensuring the safety of the robot during the material transport task and the performance of shortening the time to get to the final destination and as well reducing the path error. If the robot can determine whether or

not the human worker notices the robot, then the collision avoidance controller can be a simple switch case model where both the evasive strategy based on checking the velocity obstacle set and our evasive strategy of solving a PSO game can be set up as two cases. If the human agent notices the robot, then it is likely to cooperate with the robot. Therefore, the robot can use the evasive strategy based on checking the velocity obstacle set to keep its safety. If the human agent does not notice the robot, then the cooperation with the robot is not guaranteed to take place. Therefore, to guarantee safety of the robot, our evasive policy of solving a PSO game can be used.

Table 4.5: *The result of using the evasive strategy [18] in the PSO game where the obstacle does not interfere the path.*

Pursuer Movement	Success Rate	Task Duration		Path Error	
		Mean	STD	Mean	STD
Diagonal Cross Maneuver	63.67	6.52	0.04	0.09	0.22
Straight Cross Maneuver	59.00	6.52	0.02	0.09	0.17
Downwards Maneuver	85.28	6.55	0.05	0.09	0.25

Table 4.6: *The result of using the evasive strategy [18] in the PSO game where the obstacle interferes the path.*

Pursuer Movement	Success Rate	Task Duration		Path Error	
		Mean	STD	Mean	STD
Diagonal Cross Maneuver	52.67	6.67	0.21	1.21	0.82
Downwards Maneuver	73.67	6.64	0.18	1.03	0.75

See Section 1.1.4 for the description of the evasive policy of [22]. A clear advantage of using this evasive strategy is its simplicity in the implementation. The other advantage is that the path error can be neglected if using the standard safety field to avoid the collision with human. The guaranteed safety of the robot is based on the idea that no collision can occur before the robot stops, *passive safety* in [31]. Therefore, human is assumed to be cooperative and avoids the mobile robot after seeing it. However, if the human worker does not notice the presence of the robot, then there is still possibility that it can collide with the mobile robot unintentionally. Another potential problem of using this evasive policy is that the robot may get stuck at a location along the nominal path if the human agent stays on the boundary of the stop zone. This greatly increases the task duration and therefore affects the overall performance of the robot during a material transport task.

Chapter 5

Conclusions, Limitations, and Future Work

5.1 Conclusions

In this thesis, we proposed the PSO game which is the reversed homicidal chauffeur game [15] with the inclusion of a static obstacle and developed an event triggered collision avoidance strategy, illustrated in Figure 3.8. With a defined default escape route, the events listed in Table 3.1 are based on the pursuer's location at each instant relative to the default escape route and an event is triggered at the last moment at which there exists a feasible evasive action. This is easy if the obstacle is outside the default escape route. Before the obstacle enters the default escape route, the event triggered control evaluates whether or not there is a future feasible evasive action so that the evader's safety is guaranteed with the obstacle inside the default escape route. If the evaluation is false, then the evader turns hard left under the obstacle to avoid both the pursuer and the obstacle. Otherwise, when the obstacle is inside the default escape route, the evaluation for the existence of feasible LS/LSL or RL/RSL must be conducted continuously as the evader keeps following the nominal path by following steps in Section 3.4.3 or Section 3.4.4. Theorem 1 and its associating proof demonstrate that the use of our event triggering controller guarantees the evader's safety.

Based on the simulations in Section 4.1, the evader should continue to follow the nominal path for as long as possible because both the task duration and the path error resulting from the execution of a later feasible evasive action are lower than the corresponding values resulting from the execution of an earlier evasive action. The only exception is when the

pursuer executes the downwards maneuver in the scenario where the obstacle interferes the nominal path. However, we can still conclude that a mobile robot can generally perform better in a simple material transport task if it can delay its first evasive action for as long as possible.

We also conducted a comparison study between our evasive strategy with the two commonly used collision avoidance methods in the industry: the evasive strategy based on continuously checking the velocity obstacle set [18] and the evasive strategy based on the ISO 13482 [22]. Even though both the average task duration and the average path error are lower using the evasive strategy [18] compared to our evasive strategy, it cannot guarantee the safety of the evader. Clearly, there is a trade off between the mobile robot's performance during a material transport task and the evader's safety when selecting the avoidance mechanism. For the evasive strategy based on the ISO 13482, the evader may deadlock due to the pursuer's behaviour, e.g., stopping on the boundary of the stop zone. This can greatly increase the task duration and hinders the robot's performance in a material transport task. Based on the idea presented in [31] about passive safety, no collision can occur before the robot stops. However, the human worker may walk backwards and may be unaware of the mobile robot. This can cause potential collision if the mobile robot stops and does nothing.

5.1.1 Limitations in Our Evasive Policy

In this thesis, the evader only follows a straight nominal path. A clear limitation is to use our evasive policy in situations where the evader follows a curved path. At any time instant, the ideas presented in both Section 3.4.3 and Section 3.4.4 can still be used to determine if there exists a feasible LS/LSL or RL/RSL because the computations only need to know the initial states of all players right before the evader executing an evasive action. The biggest challenge is to compute the capture set which shows the set of pursuer's initial positions for it to arrive on the barrier surface within a finite amount of time if the evader continues to follow the nominal path. The result in Section 3.3.2 cannot be directly applied since the pursuer's strategy that results in the minimum time to arrive on the barrier surface is only used under the condition that the nominal path must be straight. Another limitation of our event triggering controller is to guarantee the evader's safety if the game space contains multiple obstacles and multiple pursuers simultaneously.

5.2 Future Work

Future work of the research presented in this thesis include solving the limitations mentioned in Section 5.1.1. The simulations in Section 4.1 suggest that both the task duration and path error are generally reduced if the evader maximizes its wait time to execute the first evasive action while guaranteeing the safety of the evader. A solid proof can show that we have realized the objective of maximizing the wait time for the evader to execute its first evasive action by executing it only at the last moment for which the evader's safety can be guaranteed. Also an interesting research topic is to possibly find another evasive strategy that ensures the evader to guarantee its safety while the time for the evader to execute its first evasive action can be further delayed compared to our event triggered controller. All the proposed collision policies in this thesis could be further validated by implementing them in a real environment with an actual mobile robot as the evader, a static obstacle, and a human as the pursuer.

References

[1] Omron ld series. http://www.ia.omron.com/products/family/3664/. Accessed: 2020-12-26.

[2] Otto 100. https://ottomotors.com/amrs. Accessed: 2020-12-26.

[3] Andrea Bajcsy, Somil Bansal, Eli Bronstein, Varun Tolani, and Claire J Tomlin. An efficient reachability-based framework for provably safe autonomous navigation in unknown environments. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, pages 1758–1765. IEEE, 2019.

[4] Somil Bansal, Mo Chen, Ken Tanabe, and Claire J Tomlin. Provably safe and scalable multivehicle trajectory planning. *IEEE Transactions on Control Systems Technology*, 2020.

[5] Shaunak D Bopardikar, Stephen L Smith, Francesco Bullo, and Joao P Hespanha. Dynamic vehicle routing for translating demands: Stability analysis and receding-horizon policies. *IEEE Transactions on Automatic Control*, 55(11):2554–2569, 2010.

[6] Richard H Byrd, Jean Charles Gilbert, and Jorge Nocedal. A trust region method based on interior point techniques for nonlinear programming. *Mathematical programming*, 89(1):149–185, 2000.

[7] Richard H Byrd, Mary E Hribar, and Jorge Nocedal. An interior point algorithm for large-scale nonlinear programming. *SIAM Journal on Optimization*, 9(4):877–900, 1999.

[8] Nick Carey. Establishing pedestrian walking speeds. *Portland State University*, 1(2):4, 2005.

[9] Mo Chen and Claire J Tomlin. Hamilton–jacobi reachability: Some recent theoretical advances and applications in unmanned airspace management. *Annual Review of Control, Robotics, and Autonomous Systems*, 1:333–358, 2018.

[10] Yu Fan Chen, Michael Everett, Miao Liu, and Jonathan P How. Socially aware motion planning with deep reinforcement learning. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1343–1350. IEEE, 2017.

[11] Yu Fan Chen, Miao Liu, Michael Everett, and Jonathan P How. Decentralized non-communicating multiagent collision avoidance with deep reinforcement learning. In *2017 IEEE international conference on robotics and automation (ICRA)*, pages 285–292. IEEE, 2017.

[12] R Craig Coulter. Implementation of the pure pursuit path tracking algorithm. Technical report, Carnegie-Mellon UNIV Pittsburgh PA Robotics INST, 1992.

[13] Lester 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*, 79(3):497–516, 1957.

[14] Hobart R Everett, Douglas W Gage, Gary A Gilbreath, Robin T Laird, and Richard P Smurlo. Real-world issues in warehouse navigation. In *Mobile Robots IX*, volume 2352, pages 249–259. International Society for Optics and Photonics, 1995.

[15] Ioannis Exarchos, Panagiotis Tsiotras, and Meir Pachter. On the suicidal pedestrian differential game. *Dynamic Games and Applications*, 5(3):297–317, 2015.

[16] Andrey Fedotov, Valerii Patsko, and Varvara Turova. Reachable sets for simple models of car motion. *Recent advances in mobile robotics*, pages 147–172, 2011.

[17] Jaime F Fisac, Anayo K Akametalu, Melanie N Zeilinger, Shahab Kaynama, Jeremy Gillula, and Claire J Tomlin. A general safety framework for learning-based control in uncertain robotic systems. *IEEE Transactions on Automatic Control*, 64(7):2737–2752, 2018.

[18] Stephen J Guy, Jatin Chhugani, Changkyu Kim, Nadathur Satish, Ming Lin, Dinesh Manocha, and Pradeep Dubey. Clearpath: highly parallel collision avoidance for multi-agent simulation. In *Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 177–187, 2009.

[19] Edward Twitchell Hall. *The hidden dimension*, volume 609. Garden City, NY: Doubleday, 1966.

[20] Rufus Isaacs. Differential games iii: The basic principles of the solution process. Technical report, Rand Corp Santa Monica, 1954.

[21] Rufus Isaacs. *Differential games: a mathematical theory with applications to warfare and pursuit, control and optimization*. Courier Corporation, 1999.

[22] Theo Jacobs and Gurvinder Singh Virk. Iso 13482-the new safety standard for personal care robots. In *ISR/Robotik 2014; 41st International Symposium on Robotics*, pages 1–6. VDE, 2014.

[23] Richard E Kopp. Pontryagin maximum principle. In *Mathematics in Science and Engineering*, volume 5, pages 255–279. Elsevier, 1962.

[24] Chi-Pang Lam, Chen-Tun Chou, Kuo-Hung Chiang, and Li-Chen Fu. Human-centered robot navigation—towards a harmoniously human–robot coexisting environment. *IEEE Transactions on Robotics*, 27(1):99–112, 2010.

[25] Alexandre Lampe and Raja Chatila. Performance measure for the evaluation of mobile robot autonomy. In *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, pages 4057–4062. IEEE, 2006.

[26] Ernest Bruce Lee and Lawrence Markus. Foundations of optimal control theory. Technical report, Minnesota Univ Minneapolis Center For Control Sciences, 1967.

[27] Karen Leung, Edward Schmerling, Mo Chen, John Talbot, J Christian Gerdes, and Marco Pavone. On infusing reachability-based safety assurance within probabilistic planning frameworks for human-robot vehicle interactions. In *International Symposium on Experimental Robotics*, pages 561–574. Springer, 2018.

[28] J Lewin and JV Breakwell. The surveillance-evasion game of degree. *Journal of Optimization Theory and Applications*, 16(3-4):339–353, 1975.

[29] Jiechao Liu, Paramsothy Jayakumar, Jeffrey L Stein, and Tulga Ersal. A study on model fidelity for model predictive control-based obstacle avoidance in high-speed autonomous ground vehicles. *Vehicle System Dynamics*, 54(11):1629–1650, 2016.

[30] Stefan B Liu, Hendrik Roehm, Christian Heinzemann, Ingo Lütkebohle, Jens Oehlerking, and Matthias Althoff. Provably safe motion of mobile robots in human environments. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1351–1357. IEEE, 2017.

[31] Kristijan Macek, Dizan Alejandro Vasquez Govea, Thierry Fraichard, and Roland Siegwart. Towards safe vehicle navigation in dynamic urban scenarios. 2009.

[32] Kristijan Macek, Ivan PetroviC, and N Peric. A reinforcement learning approach to obstacle avoidance of mobile robots. In *7th International Workshop on Advanced Motion Control. Proceedings (Cat. No. 02TH8623)*, pages 462–466. IEEE, 2002.

[33] Satyanarayana G Manyam, David Casbeer, Alexander L Von Moll, and Zachariah Fuchs. Shortest dubins path to a circle. In *AIAA Scitech 2019 Forum*, page 0919, 2019.

[34] Antony W Merz. The homicidal chauffeur-a differential game. Technical report, Stanford University Dept of Aeronautics and Astronautics, 1971.

[35] AW Merz. The game of two identical cars. *Journal of Optimization Theory and Applications*, 9(5):324–343, 1972.

[36] AW Merz. The homicidal chauffeur. *AIAA Journal*, 12(3):259–260, 1974.

[37] Ian M Mitchell, Alexandre M Bayen, and Claire J Tomlin. A time-dependent hamilton-jacobi formulation of reachable sets for continuous dynamic games. *IEEE Transactions on automatic control*, 50(7):947–957, 2005.

[38] Oskar Morgenstern and John Von Neumann. *Theory of games and economic behavior*. Princeton university press, 1953.

[39] Nils J Nilsson. A mobile automaton: An application of artificial intelligence techniques. Technical report, Sri International Menlo Park Artificial Intelligence Center, 1969.

[40] Brian Paden, Michal Čáp, Sze Zheng Yong, Dmitry Yershov, and Emilio Frazzoli. A survey of motion planning and control techniques for self-driving urban vehicles. *IEEE Transactions on intelligent vehicles*, 1(1):33–55, 2016.

[41] Valery S Patsko and Varvara L Turova. Families of semipermeable curves in differential games with the homicidal chauffeur dynamics. *Automatica*, 40(12):2059–2068, 2004.

[42] Valery S Patsko and VL Turova. Homicidal chauffeur game: computation of level sets of the value function. In *Advances in Dynamic Games and Applications*, pages 295–322. Springer, 2001.

[43] VS Patsko and VL Turova. Numerical study of the homicidal chauffeur game. In *8th Int. conf. On Differential Equation*, pages 363–371, 1998.

[44] K Peppe. Constant bearing, decreasing range: The navy is on a collision course with reality. *Proceedings - United State Naval Institute*, 122:41–43, 1996.

[45] Tuomas Raivio. Capture set computation of an optimally guided missile. *Journal of Guidance, Control, and Dynamics*, 24(6):1167–1175, 2001.

[46] Tuomas Raivio and Harri Ehtamo. On the numerical solution of a class of pursuit-evasion games. In *Advances in dynamic games and applications*, pages 177–192. Springer, 2000.

[47] Photchara Ratsamee, Yasushi Mae, Kenichi Ohara, Masaru Kojima, and Tatsuo Arai. Social navigation model based on human intention analysis using face orientation. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1682–1687. IEEE, 2013.

[48] Photchara Ratsamee, Yasushi Mae, Kenichi Ohara, Tomohito Takubo, and Tatsuo Arai. Human–robot collision avoidance using a modified social force model with body pose and face orientation. *International Journal of Humanoid Robotics*, 10(01):1350008, 2013.

[49] Valerio Turri, Ashwin Carvalho, Hongtei Eric Tseng, Karl Henrik Johansson, and Francesco Borrelli. Linear model predictive control for lane keeping and obstacle avoidance on low curvature roads. In *16th international IEEE conference on intelligent transportation systems (ITSC 2013)*, pages 378–383. IEEE, 2013.

[50] Richard A Waltz, José Luis Morales, Jorge Nocedal, and Dominique Orban. An interior algorithm for nonlinear optimization that combines line search and trust region steps. *Mathematical programming*, 107(3):391–408, 2006.

[51] Yuan Wang, Dongxiang Zhang, Qing Liu, Fumin Shen, and Loo Hay Lee. Towards enhancing the last-mile delivery: An effective crowd-tasking model with scalable solutions. *Transportation Research Part E: Logistics and Transportation Review*, 93:279–293, 2016.

[52] Yongsoon Yoon, Jongho Shin, H Jin Kim, Yongwoon Park, and Shankar Sastry. Model-predictive active steering and obstacle avoidance for autonomous ground vehicles. *Control Engineering Practice*, 17(7):741–750, 2009.

APPENDICES

Appendix A

Derivations in Section 3.3.2

Derivation for $T(\boldsymbol{x}_e(0), \boldsymbol{y}_e(0), \tau)$. Based on (3.10) and Figure 3.11, the following can be derived:

$$\begin{aligned}
&(x_p(0) - x(\boldsymbol{x}_e(0), \tau))^2 + (y_p(0) - y(\boldsymbol{x}_e(0), \tau) - v_e T(\boldsymbol{x}_e(0), \boldsymbol{y}_e(0), \tau))^2 = (v_p T_\tau)^2 \\
&(v_e^2 - v_p^2) T(\boldsymbol{x}_e(0), \boldsymbol{y}_e(0), \tau)^2 + 2v_e(y(\boldsymbol{x}_e(k), \tau) - y_p(k)) T(\boldsymbol{x}_e(0), \boldsymbol{y}_e(0), \tau) \\
&\qquad + (x_p(k) - x(\boldsymbol{x}_e(k), \tau))^2 + (y_p(k) - y(\boldsymbol{x}_e(k), \tau))^2 = 0.
\end{aligned} \tag{A.1}$$

$T(\boldsymbol{x}_e(0), \boldsymbol{y}_e(0), \tau)$ can be found by solving the above quadratic equation in terms of other terms in (A.1).

Since the evader moves along the vertical path fixed along the y-axis, the initial configurations of all players in both realistic space and reduced space are the same. Thus, we can abuse the notation by replacing the following terms with their corresponding terms in the reduced space:

- $x(\boldsymbol{x}_e(0), \tau) \quad \rightarrow \quad x(\tau)$
- $y(\boldsymbol{x}_e(0), \tau) \quad \rightarrow \quad y(\tau)$
- $x_p(0) \quad \rightarrow \quad x(0)$
- $y_p(0) \quad \rightarrow \quad y(0).$

Based on the list above, the old notations are replaced by the new notations in (A.1). By solving $T(\boldsymbol{x}_e(0), \boldsymbol{y}_e(0), \tau)$ again with the new notations, (3.12) can be obtained. □

Derivation of the Linear Boundary for the Capture Set. Since $T(\boldsymbol{x}_e(0), \boldsymbol{y}_e(0), \tau)$ is found by solving the quadratic equation in (A.1), the condition below must hold for $T(\boldsymbol{x}_e(0), \boldsymbol{y}_e(0), \tau)$ to be feasible:

$$v_p^2[(x(0) - x(\tau))^2 - (y(0) - y(\tau))^2] - v_e^2(x(0) - x(\tau))^2 \geq 0. \tag{A.2}$$

After expanding and rearranging the above condition, the relationship below regarding for the initial pursuer's position must hold for $T(\boldsymbol{x}_e(0), \boldsymbol{y}_e(0), \tau)$ to be feasible:

$$y(0) \geq \frac{\sqrt{v_e^2 - v_p^2}}{v_p}(x(0) - x(\tau)) + y(\tau). \tag{A.3}$$

The expression (A.3) presents a linear relationship between the pursuer's position in y-coordinate and x-coordinate. Therefore, the slope of this relationship is given by

$$\frac{\sqrt{v_e^2 - v_p^2}}{v_p}.$$

The linear relationship is the linear boundary of the set of space where the pursuer can get to the point on the barrier surface given by τ within $T(\boldsymbol{x}_e(0), \boldsymbol{y}_e(0), \tau)$ under the constant bearing scheme. With the nominal path as a vertical line, if the pursuer is initially on the linear boundary, then the travel time under the constant bearing scheme for it to arrive at the point on the barrier surface given by τ is

$$T(\boldsymbol{x}_e(0), \boldsymbol{y}_e(0), \tau) = \frac{v_e(y(0) - y(\tau))}{v_e^2 - v_p^2}. \tag{A.4}$$

Illustrated in Figure 3.12, the pursuer arrives on the barrier surface at the point given by τ after the evader travels along the vertical path for $T(\boldsymbol{x}_e(0), \boldsymbol{y}_e(0), \tau)$. Therefore, the point of capture in the realistic space is found. A circle can be formed surrounding the point of capture and the space contained inside the circle is the initial pursuer's position where the pursuer can arrive on the barrier surface at the point $(x(\tau), y(\tau))$ within the period $T(\boldsymbol{x}_e(0), \boldsymbol{y}_e(0), \tau)$. The linear boundary (A.3) touches the circle and it has the left and the right branch.

Let

$$d \coloneqq \sqrt{(x(0) - x(\tau))^2 - (y(0) - y(\tau))^2}, \tag{A.5}$$

be the euclidean distance between the initial pursuer's position and the initial position of the barrier surface at the point τ. The angle between the linear boundary and the vertical path that the point on the barrier surface travels can be found through the following expressions:

$$\cos(\theta_\tau) = \frac{y(0) - y(\tau)}{d}, \qquad \sin(\theta_\tau) = \frac{x(0) - x(\tau)}{d}, \qquad m = \frac{y(0) - y(\tau)}{x(0) - x(\tau)} = \frac{\cos(\theta_\tau)}{\sin(\theta_\tau)},$$

where m is the slope relationship for the linear boundary between

$$\begin{bmatrix} x(\tau) \\ y(\tau) \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} x(0) \\ y(0) \end{bmatrix}.$$

Treating (A.3) as an equality expression and through rearranging, we obtain

$$y(0) - y(\tau) = \frac{\sqrt{v_e^2 - v_p^2}}{v_p}(x(0) - x(\tau)),$$

From the above relations, we see that

$$d\cos(\theta_\tau) = \frac{\sqrt{v_e^2 - v_p^2}}{v_p} d\cos(\theta_\tau),$$

and through elimination of d on both sides we conclude that the slope of the linear boundary can also be expressed as

$$m = \frac{\sqrt{v_e^2 - v_p^2}}{v_p} = \frac{\cos(\theta_\tau)}{\sin(\theta_\tau)}.$$

Treating (A.2) as an equality expression and through rearranging, we obtain

$$v_p^2[(x(0) - x(\tau))^2 - (y(0) - y(\tau))^2] = v_e^2(x(0) - x(\tau))^2,$$

and by substituting the value d into the above expression we can conclude that

$$v_p d = v_e(x(0) - x(\tau)).$$

By using the expression $\sin(\theta_\tau) = \frac{x(0)-x(\tau)}{d}$, we see that

$$\sin(\theta_\tau) = \frac{v_p}{v_e}, \tag{A.6}$$

and by using the alternative expression for the slope of the linear boundary above we also see that

$$\cos(\theta_\tau) = \frac{\sqrt{v_e^2 - v_p^2}}{v_e}. \tag{A.7}$$

With (A.2) as an equality and substituting (A.6), (A.7), and (A.8) into (A.3), the expression for d can be alternatively given by

$$d = \left(\sqrt{v_e^2 - v_p^2}\right) T(\boldsymbol{x}_e(0), \boldsymbol{y}_e(0), \tau). \tag{A.8}$$

By substituting (A.6), (A.7), and (A.8) into (A.3), the following can be derived for the relationship between the initial pursuer's position in terms of $T(\boldsymbol{x}_e(0), \boldsymbol{y}_e(0), \tau)$ and the initial position on the barrier surface that the pursuer tries to pursue:

$$x(0) = \left(\frac{v_p}{v_e}\sqrt{v_e^2 - v_p^2}\right) T(\boldsymbol{x}_e(0), \boldsymbol{y}_e(0), \tau) + x(\tau), \qquad y(0) = \left(\frac{v_e^2 - v_p^2}{v_e}\right) T(\boldsymbol{x}_e(0), \boldsymbol{y}_e(0), \tau) + y(\tau).$$

The above relation tells us the change of the linear boundary with respect to $T(\boldsymbol{x}_e(0), \boldsymbol{y}_e(0), \tau)$. □

Derivation for $\theta_{\tau'}$ in Figure 3.12. In Figure 3.12, a line can be formed in between the initial pursuer's location and the point of capture for the point on the barrier surface given by τ. By using the *law of sines*, the angle between this line and the linear boundary (A.3), $\theta_{\tau'}$ in Figure 3.12 can be found by the following expression:

$$\left(\frac{v_p}{\sin(\theta_\tau)}\right) T(\boldsymbol{x}_e(0), \boldsymbol{y}_p(0), \tau) = \left(\frac{v_e}{\sin(\theta_{\tau'})}\right) T(\boldsymbol{x}_e(0), \boldsymbol{y}_p(0), \tau).$$

By rearranging the above expression, we see that

$$\sin(\theta_{\tau'}) = \frac{v_e}{v_p}\sin(\theta_\tau),$$

and by substituting $\sin(\theta_\tau) = \frac{v_p}{v_e}$ in (A.6) into this equality, we conclude

$$\sin(\theta_{\tau'}) = 1.$$

Therefore, $\theta_{\tau'} = \frac{\pi}{2}$. □