IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 8 March 2023, accepted 25 April 2023, date of publication 28 April 2023, date of current version 3 May 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3271592

RESEARCH ARTICLE

Capturing a Differential Drive Robot With a Dubins Car

UBALDO RUIZ
Centro de Investigación Científica y de Educación Superior de Ensenada, Ensenada, Baja California 22860, Mexico
e-mail: uruiz@cicese.mx

This work was supported by CONACYT under Grant A1-S-21934.

ABSTRACT This work addresses the pursuit-evasion problem of capturing a Differential Drive Robot (DDR) with a Dubins Car (DC) in minimum time. We model the problem as a zero-sum differential game, and using differential game theory, we compute the time-optimal motion strategies of the players near the end of the game. We unveil the existence of three singular surfaces: an Evader's Dispersal Surface (EDS), a Transition Surface (TS), where the DDR switches its controls, and a Pursuer's Universal Surface (PUS). A particular set of motion strategies for the players is used at each singular surface. To compute our solution, we assume that both players have the same maximum speed and a bounded turning ratio. Considering the previous setting, the game's outcome only depends on the particular motion capabilities of the players. From previous results in the literature and those presented in the current paper, we can establish that a DDR has an advantage when it plays as a pursuer or as an evader compared to a Dubins Car performing the same role.

INDEX TERMS Differential games, optimal control, Pursuit-evasion, robotics.

I. INTRODUCTION

Several tasks in robotics require dealing with a moving target. For example, an autonomous vehicle following another car operated by a human driver, a robotic guard that wants to capture a malicious agent or seeks to find a moving agent in an environment. These problems can be modeled as pursuit-evasion games [1], [2], where there are two classes of players who have antagonistic goals. If one finds a solution for this worst-case scenario, it is possible to adapt it for more favorable situations.

In particular, we study the problem of capturing a Differential Drive Robot (DDR) with a Dubins Car (DC) as soon as possible. We assume both agents move on the Euclidean plane, and we model them like unitary discs. The DC plays the pursuer's role, and the DDR plays the evader's role. The DC's goal is to capture the DDR in minimum time, while the DDR wants to avoid being captured for as much time as possible. To the best of our knowledge, this is the first work to address this particular game setting. A similar problem involving a Dubins Car and a Differential Drive Robot was studied in [3]. In that work, different from our current formulation, the Differential Drive Robot wants to capture the Dubins Car in minimum time. This *"slight"* change in the players' roles has a major consequence in computing the problem's solution and requires performing a complete analysis to construct it. As will be shown in the current paper, we found that the time-optimal motion strategies in [3] are different from the ones computed in the current paper.

An important aspect of this work is that it considers two of the most popular non-holonomic mobile vehicles in robotics, a DC and a DDR. Dealing with two non-holonomic vehicles in a pursuit-evasion game makes the problem more difficult to model and analyze compared to the existing work [4], [5]. Other works in the literature have studied pursuit-evasion games involving two non-holonomic players. For example, in [6], the capture problem between two identical Dubins Cars was analyzed. In [7], an analogous version considering two identical DDRs was studied. Note that both players have the same motion capabilities in those works, while in our game, the pursuer and the evader have different kinematic models. Having different motion capabilities make us wonder if there is a more favorable role for each player. From the results in [3] and those presented in the current paper, we established that a DDR has an advantage when it plays either of the two roles compared to a Dubins Car that performs the same role.

The associate editor coordinating the review of this manuscript and approving it for publication was Yangming Li.

Additionally, since the players have different kinematic constraints in our game, this also implies that the time-optimal motion strategies used by the players in [6] and [7] are distinct from those found in our current work.

In our problem, as in [3], [6], and [7] to find the players' time-optimal motion strategies, we employ tools from differential game theory [1], [2]. In particular, we model our problem as a zero-sum game, where the cost the players seek to optimize is the capture time. We characterize regions of the playing space containing initial configurations that lead to capture. To compute the solution to our pursuit-evasion problem, we follow the approach developed by Isaacs [1], [8]. The main idea behind Isaacs' methodology is partitioning the playing space into regions where the value function is differentiable. Finding the boundaries of those regions, known as *singular surfaces*, is usually the most difficult part of the process since there is no knowledge ahead of time if the optimal trajectories have singular portions. Usually, when the regular backward construction of candidate trajectories does not cover the entire playing space [8], one assumes the existence of a singular surface and tributary trajectories joining it.

Isaacs' methodology allows computing closed-form solutions [1], [2], [3], [4], [5], [6], [7], [8], [9] for some specific game formulations. In this paper, we find analytical expressions that describe the motion strategies of the players.

A. RELATED WORK

In a differential game [1], [2], [8], [10], the state evolution of two or more agents is dictated by a dynamical system. If the agents have conflicting goals, then it is known as a non-cooperative differential game. Additionally, if a player's gain is balanced by the other's loss, it is known as a zero-sum non-cooperative game. Our problem belongs to the latter class of games. Many works studying non-cooperative zero-sum differential games can be found in the literature [4], [5], [7], [9], [11], [12], [13]. The Homicidal Chauffeur problem [9] is probably the most famous and well-known. In that problem, a Dubins Car has as its task to run over a pedestrian as soon as possible, and it wins if it can accomplish the goal in finite time. On the other hand, the pedestrian wants to delay it as much as possible, and he wins if he can avoid it forever. The car is faster than the pedestrian, but it has a turning ratio constraint which limits its ability to change its motion direction. On the contrary, the pedestrian is more agile than the car, as it can instantly change its direction of motion. The game occurs in an infinite parking lot without obstacles, modeled as an Euclidean plane. Thus, the game's outcome depends solely on the kinematic constraints of the players. Solving the problem consists of finding the motion strategies of the players to achieve their goals and, based on the system's initial configurations, deciding which player wins the game.

The problem addressed in this paper is related to the following works in the literature: [3], [4], [5], [6], [7]. In the following paragraphs, we compare them with our current work. Reference [4] addressed the pursuit-evasion problem of capturing an omnidirectional agent with a DDR. In that work, analogous to the Homicidal Chauffeur problem, the DDR is faster than the omnidirectional agent but can instantaneously change its motion direction at a bounded rate. The DDR wants to capture the omnidirectional agent in the shortest possible time. On the contrary, the omnidirectional agent wants to delay it as much as possible. The time-optimal motion strategies of the players to achieve their goals are computed in [4]. In addition, considering the players' initial configurations, the winner of the game can be decided, i.e., if the DDR captures the omnidirectional agent in finite time or not. Reference [5] studied an analogous problem in which the players' roles are reversed; the omnidirectional agent's goal is capturing the DDR in minimum time. In contrast, the DDR wants to delay it. We also have a DDR agent in our current work, analogous to [4] and [5]. Nevertheless, in our case, the pursuer is a DC with non-holonomic constraints distinct from those of an omnidirectional agent or a DDR. The result of this modification is that the players' motion strategies and the solution's essence differ from those obtained in [4] and [5]. Another key difference is that to solve our game; we require a space representation with an additional dimension compared to the one used in [4] and [5]. Having an additional dimension makes it more difficult to compute and analyze the solution.

Two related works to our problem considering non-holonomic players are [6] and [7]. As in our work, the pursuer's goal is to capture the evader in minimum time. In [7], the pursuer and the evader are two identical DDRs. Note that, similar to this work, the evader is a DDR. Nevertheless, the pursuer is also a DDR, which has a different kinematic model from a DC. That change results in motion trajectories for the players distinct from those in our current work. In [6], an analogous version is studied considering two identical DCs. Again, given that the evader has different kinematic capabilities, the player's motion strategies found in [6] differ from the ones presented in the current work. A more comprehensive solution to the problem in [6] was presented in [14], and a feedback-based solution for particular cases is provided in [15]. However, since our evader is a DDR, our solution differs from those of [14] and [15].

Recently, a similar game of capturing involving a DDR and a DC was studied in [3]. In that work, a DDR wants to capture a DC in minimum time; the DDR plays as a pursuer and the DC as an evader. In our work, the players have opposite roles. This change in the game's formulation has the consequence that the solution of our game is different from the one presented in [3]. In particular, the regions where capture can be achieved differ from those in [3]. In our game, since the DC is restricted to always move forward while the DDR can move forwards or backward, there is a more limited set of configurations where capture can occur compared to [3]. Additionally, the boundaries of the regions (singular surfaces) where the value function is differentiable also change. In [3], four singular surfaces were revealed, two evader's dispersal surfaces, a transition surface, and a pursuer's dispersal surface. In our current game, an evader's dispersal surface, a transition surface, and a pursuer's universal surface were exhibited. The differences in the players' motion strategies at each singular surface will be discussed in more detail in the following sections.

B. CONTRIBUTIONS

We present a list of the main contributions of our work.

- We identify the initial configurations in the playing space from which the DC captures the DDR, and we compute the time-optimal motion strategies of the players to achieve the task from those configurations.
- We unveil the existence of three singular surfaces in this game near the end of the game, an evader's dispersal surface (EDS), a transition surface (TS) where the DDR switches its controls, and a pursuer's universal surface (PUS).
- Numerical simulations of the time-optimal motion strategies of the players in two reference frames are provided.
- From the results in [3] and those presented in the current paper, we establish that a DDR has an advantage when it plays either of the two roles compared to a Dubins Car performing the same role.

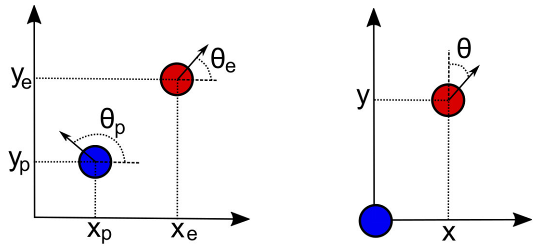
II. PROBLEM FORMULATION

Two non-holonomic vehicles, a Dubins Car (DC) and a Differential Drive Robot (DDR), move on the Euclidean plane. The DC plays as a pursuer and tries to capture the DDR as soon as possible. The DDR, in contrast, plays as an evader and tries to delay capture. Both players are modeled as unitary discs with the same maximum speed of $V^{\max}$. Therefore, the game's outcome is determined by the specific non-holonomic constraints of the players. The game ends when the distance between both players is smaller than the value l_c. We study the problem in a purely kinematic setting.

The problem is initially modeled in the Euclidean plane (see Fig. 1). In our case, we denote the DC pose as (x_p, y_p, θ_p) and the DDR pose as (x_e, y_e, θ_e). The state of the system is represented as $(x_p, y_p, \theta_p, x_e, y_e, \theta_e) \in \mathbb{R}^2 \times S^1 \times \mathbb{R}^2 \times S^1$. In differential games, this representation is called *realistic space*. The equations of motion of the system are

$$\begin{aligned}
\dot{x}_p &= V^{\max}\cos\theta_p, \quad \dot{y}_p = V^{\max}\sin\theta_p,\\
\dot{\theta}_p &= \frac{V^{\max}}{r_p}v\\
\dot{x}_e &= \left(\frac{u_1+u_2}{2}\right)\cos\theta_e, \quad \dot{y}_e = \left(\frac{u_1+u_2}{2}\right)\sin\theta_e,\\
\dot{\theta}_e &= \left(\frac{u_2-u_1}{2b}\right) \qquad (1)
\end{aligned}$$

where $v \in [-1, 1]$ is the DC control and r_p is the maximum turning radius of the DC. The DDR controls are $u_1, u_2 \in [-V^{\max}, V^{\max}]$. They correspond to the left and right wheel velocities, respectively. We denote the distance between the DDR's center and the wheels' location by b. In this game, we assume that both wheels have a unitary radius. Therefore, the rotational and translational speeds are equivalent. The previous equations can be summarized as $\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{v}, \mathbf{u})$ where $v \in [-1, 1]$ and $u = (u_1, u_2) \in [-V^{\max}, V^{\max}] \times [-V^{\max}, V^{\max}]$. In the realistic space, all angles are measured counter-clockwise from the x-axis.



(a) Realistic space (b) Reduced space

FIGURE 1. The DC pursuer corresponds to the blue disc and the DDR evader to the red disc.

In differential game theory, a coordinate system mounted on the pursuer is generally used to simplify the analysis. In this case, we employ a reference frame fixed to the DC body (see Fig. 1b). In this frame, the state of the system is given by $\mathbf{x} = (x, y, \theta)$, which corresponds to the DDR pose relative to the body of the DC. In the literature, this representation is known as the *reduced space*. In this case, all orientations are measured with respect to the positive y-axis in a clockwise sense. The following equations provide a transformation between the reduced and realistic space representations,

$$\begin{aligned}
x &= (x_e - x_p)\sin\theta_p - (y_e - y_p)\cos\theta_p\\
y &= (x_e - x_p)\cos\theta_p + (y_e - y_p)\sin\theta_p\\
\theta &= \theta_p - \theta_e \qquad (2)
\end{aligned}$$

From (2), we obtain the motion equations in the reduced space,

$$\begin{aligned}
\dot{x} &= \left(\frac{V^{\max}}{r_p}\right)vy + \left(\frac{u_1+u_2}{2}\right)\sin\theta\\
\dot{y} &= -\left(\frac{V^{\max}}{r_p}\right)vx - V^{\max} + \left(\frac{u_1+u_2}{2}\right)\cos\theta\\
\dot{\theta} &= \left(\frac{V^{\max}}{r_p}\right)v - \left(\frac{u_2-u_1}{2b}\right) \qquad (3)
\end{aligned}$$

where $v \in [-1, 1]$ is the DC control and $u_1, u_2 \in [-V^{\max}, V^{\max}]$ are the DDR controls. The previous equations can be summarized in the form $\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{v}, \mathbf{u})$, where $\mathbf{x} = (x, y)$, $\mathbf{v} = (v)$ and $\mathbf{u} = (u_1, u_2)$.

III. TERMINAL CONDITIONS

In this section, we find the configurations in which the DC guarantees the capture independently of the controls selected by the DDR [1]. This set is named the usable part (UP). In our case, the DC captures the DDR when the distance between them is less than a value l_c despite any DDR's opposition. For our game, similar to [3], we represent the terminal surface

ζ in the reduced space as a cylinder centered on the origin with a radius l_c and a height 2π. It is convenient to define a parameterization of ζ by two angles ϕ and ψ. ϕ is the angle between the DC's heading and the DDR's position, and ψ is the angle between the headings of both players. We denote l as the distance between the players. In the reduced space, DC guarantees capture when $l = l_c$ and $\dot{l} < 0$. In the UP

$$x = l_c \sin\phi, \quad y = l_c \cos\phi, \quad \theta = \psi, \quad l^2 = x^2 + y^2, \tag{4}$$

This game ends when

$$\min_{v} \max_{u_1,u_2} \dot{l} < 0 \tag{5}$$

From the time derivative of l and substituting (3) into the resulting expression,

$$\begin{aligned} \dot{l} &= \left(\frac{u_1+u_2}{2}\right)(\sin\phi\sin\psi + \cos\phi\cos\psi) - V^{\max}\cos\phi \\ &= \left(\frac{u_1+u_2}{2}\right)\cos(\phi-\psi) - V^{\max}\cos\phi \end{aligned} \tag{6}$$

Considering the optimal controls of the players in the UP,

$$\min_{v} \max_{u_1,u_2} \dot{l} = V^{\max}\,|\cos(\phi-\psi)| - V^{\max}\cos\phi \tag{7}$$

Therefore, from (5),

$$\text{UP} = \{\phi, \psi \mid \cos\phi > |\cos(\phi-\psi)|\} \tag{8}$$

The usable part boundary (BUP) is given by

$$\min_{v} \max_{u_1,u_2} \dot{l} = 0 \tag{9}$$

therefore,

$$\text{BUP} = \{\phi, \psi \mid \cos\phi = |\cos(\phi-\psi)|\} \tag{10}$$

From (7), we conclude that the DDR executes a translation at maximum speed when it is captured by the DC. The DDR translates forward when $\cos(\phi-\psi) > 0$ and backward when $\cos(\phi-\psi) < 0$. We can also observe that the UP and the BUP do not explicitly depend on the DC control v. However, the DC only attains capture when it moves forward, and the DDR is located in front of it, i.e., $\phi \in [-\frac{\pi}{2}, \frac{\pi}{2}]$. This behavior differs from the one observed in [3], in which the DDR can capture the DC moving forward or backward in the reverse pursuit-evasion game. Fig. 2 shows a graphical representation of the UP and its boundary for this game.

IV. OUTLINE OF THE SOLUTION

In this section, we present an outline of the problem's solution to provide the reader with a guide to the process applied to compute and identify the time-optimal motion strategies of the players.

We characterize a portion of the reduced space that contains initial configurations that attain capture. In particular, we identify three regions, each corresponding to specific motion strategies for the players. Those regions are bordered by three types of singular surfaces: transition, universal, and dispersal surfaces. For this game, we found a transition surface (TS) where the DDR switches its controls, a pursuer's universal surface (PUS), and an evader's dispersal surface (EDS). Fig. 3 shows the reduced space characterization.

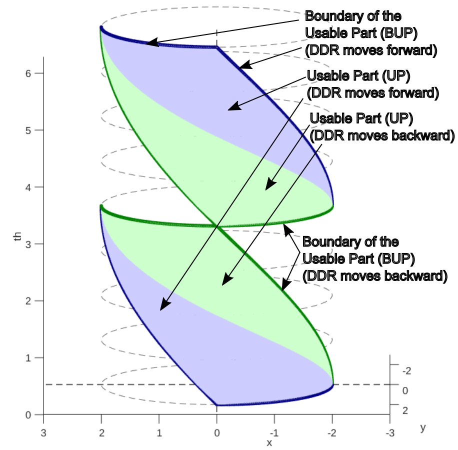


FIGURE 2. Usable Part (UP) and its boundary (BUP) in the reduced space. The purple regions contain configurations where the DC terminates the game while the DDR moves forward. Similarly, the green regions contain the final configurations in which the DDR moves backward.

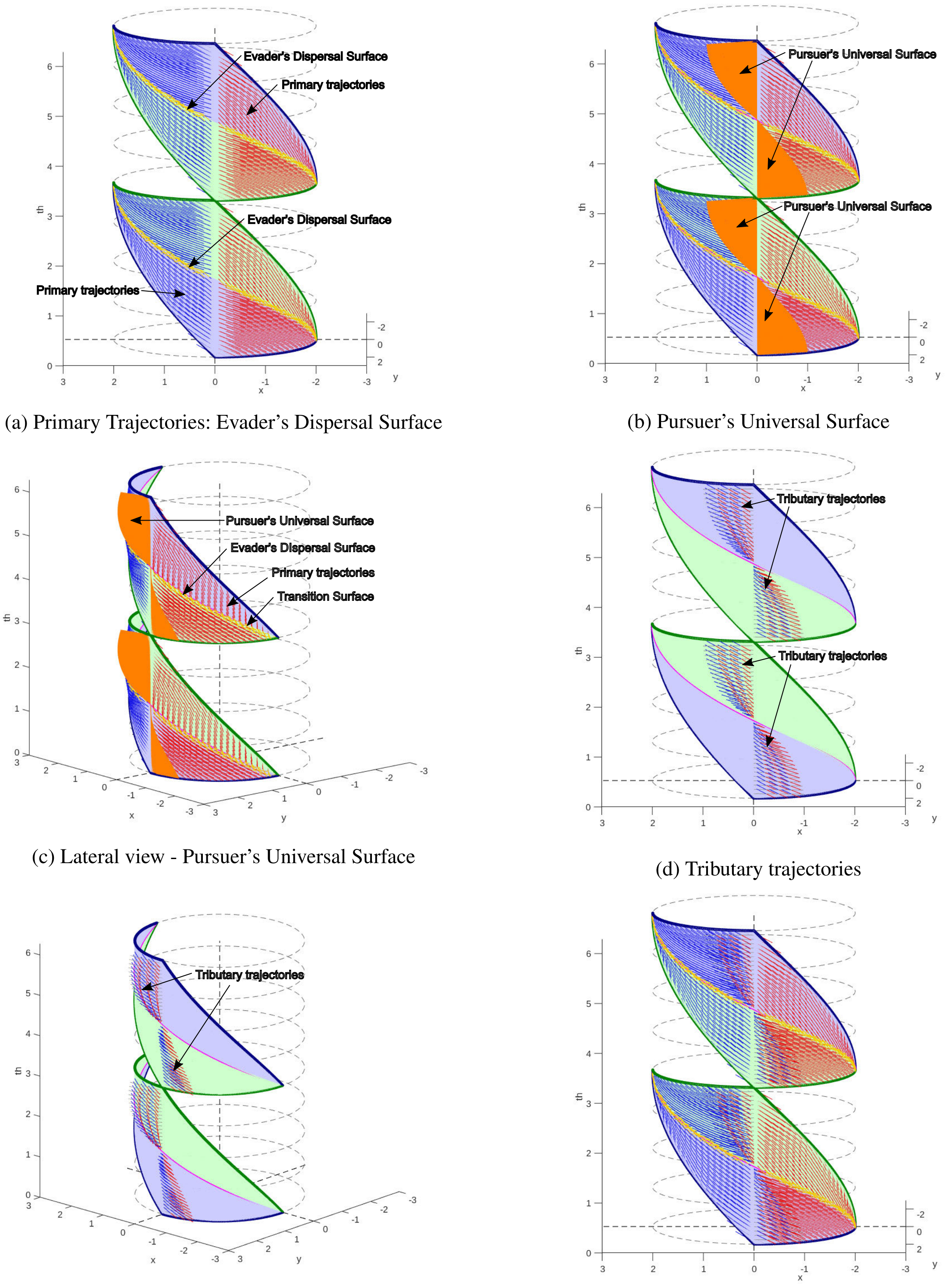
In Fig. 3a, we can observe the trajectories (blue and red curves) leading directly to the capture condition. The red curves correspond to those where the DC pursuer applies a control $v^* = 1$, while the blue curves correspond to those where $v^* = -1$ is used. In the trajectories reaching the purple region, the DDR evader moves forward at a maximum speed, whereas, in those reaching the green region, it moves backward at maximum speed.

Fig. 3a also shows a TS (the boundary between red and yellow trajectories). The DDR switches its controls on the TS, and from translating at maximum speed, it starts rotating in place in retro-time.

On a dispersal surface (DS), one player can select between two controls producing two trajectories of the same cost. Following [8], this player dominates the singular surface. The second player must respond according to the pick of the first player to avoid favoring it. To accomplish that, it must execute a non-admissible strategy, i.e., his choice of controls is based on the knowledge of his opponent's control selection. In contrast, an admissible strategy in differential game theory does not demand additional information on the players' controls. It is based just on the knowledge of the system state.

The EDS in Fig. 3a is formed by some of the primary trajectories coming from different regions of the usable part. On that dispersal surface, the DDR evader has two choices for its optimal control: rotating clockwise or counter-clockwise; however, the associated trajectories to each control have the same cost.

When we construct the trajectories that emanate from the usable part (primary solution), and we discover *void* regions, sometimes we can solve this problem by assuming that there is a universal surface (US) in the void [8]. In such cases, the void is covered by two fields of tributaries that emanate



(a) Primary Trajectories: Evader's Dispersal Surface

(b) Pursuer's Universal Surface

(c) Lateral view - Pursuer's Universal Surface

(d) Tributary trajectories

(e) Lateral view - Tributary trajectories

(f) Reduced space partition

FIGURE 3. Outline of the solution in the reduced space. The first row shows the primary trajectories, the Evader's Dispersal Surface, and the Universal Surface appearing in this game. The second row presents a lateral view of the singular surfaces and the tributary trajectories joining the Pursuer's Universal Surface. Finally, the third row shows a lateral view of the tributary trajectories reaching the Pursuer's Universal Surface and illustrates the set of trajectories filling the reduced space.

from both sides of the singular surface. Tributary trajectories from both sides join transversely on a universal surface (US).

In this game, the PUS is located at $\phi = 0$ in Fig. 3b. In the PUS, the DC applies a control $v^* = 0$. The tributaries that join the PUS on each side are computed by setting $v^* = -1$ or $v^* = 1$, see Fig. 3d.

We present a description of the properties of the regions found in this paper.

- **Region I** corresponds to the trajectories in the primary solution. The system's trajectories in this region (red and blue curves) reach the UP and correspond to a DDR's translation at maximum speed in the realistic space, while the DC translates and rotates at maximum speed.
- **Region II** contains the system's trajectories (yellow curves), departing from the transition surface in retro-time. In all cases, the trajectories correspond to a DDR's rotation in place at maximum speed in the realistic space. They end at the EDS.
- **Region III** corresponds to the tributaries joining the PUS. For these trajectories, the DDR performs a translation at maximum speed while the DC translates and rotates at maximum speed. On one side of the PUS, the DC is rotating clockwise, and on the other counterclockwise. Once a trajectory reaches the PUS, the DC follows a straight-line trajectory until it captures the DDR.

V. TIME-OPTIMAL MOTION STRATEGIES

We proceed to compute the players' time-optimal motion strategies. In differential game theory, a retro-time integration of the motion equations starting at the ending configurations is performed. To do that, first, we need to find the optimal controls of the players used to perform their tasks.

Following the methodology in [1], we proceed to construct the Hamiltonian of the system, which is given by

$$H(\mathbf{x}, \lambda, \mathbf{v}, \mathbf{u}) = \lambda^T \cdot f(\mathbf{x}, \mathbf{v}, \mathbf{u}) + L(\mathbf{x}, \mathbf{v}, \mathbf{u}) \quad (11)$$

where λ^T are the costate variables and $L(\mathbf{x}, \mathbf{v}, \mathbf{u})$ is the cost function. For our problem $L(\mathbf{x}, \mathbf{v}, \mathbf{u}) = 1$, therefore,

$$\begin{aligned} H(\mathbf{x}, \lambda, \mathbf{v}, \mathbf{u}) = &\lambda_x \left(\frac{V^{\max}}{r_p}\right) vy + \lambda_x \left(\frac{u_1 + u_2}{2}\right) \sin\theta \\ &- \lambda_y \left(\frac{V^{\max}}{r_p}\right) vx - \lambda_y V^{\max} \\ &+ \lambda_y \left(\frac{u_1 + u_2}{2}\right) \cos\theta + \lambda_\theta \left(\frac{V^{\max}}{r_p}\right) v \\ &- \lambda_\theta \left(\frac{u_2 - u_1}{2b}\right) + 1 \end{aligned} \quad (12)$$

where $\lambda^T = (\lambda_x, \lambda_y, \lambda_\theta)$. For problems of minimum time [1], as in this work,

$$\begin{aligned} &\min_{\mathbf{v}} \max_{\mathbf{u}} H(\mathbf{x}, \lambda, \mathbf{v}, \mathbf{u}) = 0 \\ &\mathbf{v}^* = \arg\min_{\mathbf{v}} H(\mathbf{x}, \lambda, \mathbf{v}, \mathbf{u}) \\ &\mathbf{u}^* = \arg\max_{\mathbf{u}} H(\mathbf{x}, \lambda, \mathbf{v}, \mathbf{u}) \end{aligned} \quad (13)$$

where $\mathbf{v}^*$ and $\mathbf{u}^*$ denote the optimal controls of the players. From (12) and (13), we have that the DC control is given by

$$v^* = -\operatorname{sgn}\left(\lambda_x y - \lambda_y x + \lambda_\theta\right) \quad (14)$$

and the DDR controls are given by

$$\begin{aligned} u_1^* &= \operatorname{sgn}\left(\frac{\lambda_x \sin\theta}{2} + \frac{\lambda_y \cos\theta}{2} + \frac{\lambda_\theta}{2b}\right) V^{\max} \\ u_2^* &= \operatorname{sgn}\left(\frac{\lambda_x \sin\theta}{2} + \frac{\lambda_y \cos\theta}{2} - \frac{\lambda_\theta}{2b}\right) V^{\max} \end{aligned} \quad (15)$$

From (15) and (14), we can observe that the optimal controls of the players depend on the values of λ^T. Those values are obtained using the costate equations, which are computed from the Hamiltonian's partial derivatives with respect to the state variables [1]. We define the retro-time as $\tau = t_f - t$, where t_f is the game's termination time. The *retro-time derivative* of a variable x is denoted as $\overset{\circ}{x}$. The costate equation in its retro-time version is

$$\overset{\circ}{\lambda} = \frac{\partial}{\partial \mathbf{x}} H(\mathbf{x}, \lambda, \mathbf{v}^*, \mathbf{u}^*) \quad (16)$$

For our game,

$$\begin{aligned} &\overset{\circ}{\lambda}_x = -\lambda_y \left(\frac{V^{\max}}{r_p}\right) v^*, \quad \overset{\circ}{\lambda}_y = \lambda_x \left(\frac{V^{\max}}{r_p}\right) v^* \\ &\overset{\circ}{\lambda}_\theta = \left(\frac{u_1^* + u_2^*}{2}\right) \left(\lambda_x \cos\theta - \lambda_y \sin\theta\right) \end{aligned} \quad (17)$$

We proceed to compute the players' retro-time trajectories close to the game's end. As the first step, we need to determine the initial conditions of the costate and motion equations in the reduced space. From (4), we have that at the game's end, $x_f = l_c \sin\phi$, $y_f = l_c \cos\phi$, and $\theta = \psi$. Thus,

$$\begin{aligned} &\frac{\partial x}{\partial \phi} = l_c \cos\phi, \quad \frac{\partial y}{\partial \phi} = -l_c \sin\phi, \quad \frac{\partial \theta}{\partial \phi} = 0 \\ &\frac{\partial x}{\partial \psi} = 0, \quad \frac{\partial y}{\partial \psi} = 0, \quad \frac{\partial \theta}{\partial \psi} = 1 \end{aligned} \quad (18)$$

Given that $\lambda(\mathrm{x}) = 0$ on the UP, then λ_ϕ and λ_ψ are

$$\begin{aligned} \lambda_\phi &= \frac{\partial \lambda}{\partial \phi} = \frac{\partial \lambda}{\partial x}\frac{\partial x}{\partial \phi} + \frac{\partial \lambda}{\partial y}\frac{\partial y}{\partial \phi} + \frac{\partial \lambda}{\partial \theta}\frac{\partial \theta}{\partial \phi} \\ &= \lambda_x \cos\phi - \lambda_y \sin\phi = 0 \\ \lambda_\psi &= \frac{\partial \lambda}{\partial \psi} = \frac{\partial \lambda}{\partial x}\frac{\partial x}{\partial \psi} + \frac{\partial \lambda}{\partial y}\frac{\partial y}{\partial \psi} + \frac{\partial \lambda}{\partial \theta}\frac{\partial \theta}{\partial \psi} = \lambda_\theta = 0 \end{aligned} \quad (19)$$

From (19), we get

$$\lambda_x \cos\phi = \lambda_y \sin\phi, \quad \lambda_\theta = 0 \quad (20)$$

Thus, in the UP,

$$\lambda_x = \sin\phi, \quad \lambda_y = \cos\phi, \quad \lambda_\theta = 0 \quad (21)$$

From Section III, near the end of the game, the DDR executes a translation at maximum speed. Thus, the solution of (17) is given by

$$\begin{aligned} \lambda_x &= \sin\left(\phi - \frac{V^{\max}}{r_p} v^* \tau\right) \\ \lambda_y &= \cos\left(\phi - \frac{V^{\max}}{r_p} v^* \tau\right) \\ \lambda_\theta &= \left(\frac{u_1^* + u_2^*}{2}\right) \sin(\phi - \psi)\tau \end{aligned} \quad (22)$$

The previous solutions are valid in the UP and if the players' controls do not change. We compute numerically the retro-time instant when the DDR switches its controls.

The next step is to integrate the motion equations in the reduced space. The retro-time version of (3) in the given by

$$\begin{aligned}\mathring{x} &= -\left(\frac{V^{\max}}{r_p}\right)vy - \left(\frac{u_1+u_2}{2}\right)\sin\theta \\ \mathring{y} &= \left(\frac{V^{\max}}{r_p}\right)vx + V^{\max} - \left(\frac{u_1+u_2}{2}\right)\cos\theta \\ \mathring{\theta} &= -\left(\frac{V^{\max}}{r_p}\right)v + \left(\frac{u_2-u_1}{2b}\right)\end{aligned} \tag{23}$$

Solving the previous equations with the initial conditions $x_f = l_c \sin\phi$, $y_f = l_c \cos\phi$ and $\theta = \psi$, and also with the optimal controls u_1^*, u_2^* and v^*, we obtain

$$\begin{aligned}x &= -\frac{r_p}{v^*} + \left(x_f + \frac{r_p}{v^*}\right)\cos\left(\frac{V^{\max}}{r_p}v^*\tau\right) \\ &\quad - y_f \sin\left(\frac{V^{\max}}{r_p}v^*\tau\right) \\ &\quad + \left(\frac{u_1^*+u_2^*}{2}\right)\sin\left(\frac{V^{\max}}{r_p}v^*\tau - \psi\right)\tau \\ y &= \left(x_f + \frac{r_p}{v^*}\right)\sin\left(\frac{V^{\max}}{r_p}v^*\tau\right) \\ &\quad + y_f \cos\left(\frac{V^{\max}}{r_p}v^*\tau\right) \\ &\quad - \left(\frac{u_1^*+u_2^*}{2}\right)\cos\left(\frac{V^{\max}}{r_p}v^*\tau - \psi\right)\tau \\ \theta &= \psi - \frac{V^{\max}}{r_p}v^*\tau\end{aligned} \tag{24}$$

v^* is provided by (14) and u_1^*, u_2^* by (15). Note that the switch function

$$S = \lambda_x y - \lambda_y x + \lambda_\theta \tag{25}$$

in (14) is zero at the moment of capture, i.e., the DC is following a straight line at that time instant. To compute the value v^* the time instant before capture we need to analyze the value of $\mathring{S}$ (the retro-time derivative of the switch function S). From that analysis, we have that $v^* = -\text{sgn}(\sin\phi)$. Also note that (24) delivers the players' trajectories in the reduced space. We need to apply a coordinate transformation to find the corresponding trajectories in the realistic space. The trajectories in (24) are called the *primary solution*.

In our game, we found that after following some of the primary trajectories, the DDR switches its controls, and it starts to rotate in place at maximum speed. However, given that (24) have transcendental functions, we cannot compute an analytical function to determine the retro-time instant τ_s when that change occurs, and numerical analysis is required to find τ_s. Once τ_s is determined, we need to perform a new integration of the costate and motion equations. Since the DDR rotates in place at maximum speed, the solution of (17) is

$$\begin{aligned}\lambda_x &= \sin\left(\phi - \frac{V^{\max}}{r_p}v^*\tau\right) \\ \lambda_y &= \cos\left(\phi - \frac{V^{\max}}{r_p}v^*\tau\right) \\ \lambda_\theta &= \lambda_{\theta_s}\end{aligned} \tag{26}$$

where λ_{θ_s} is computed substituting τ_s into the third expression in (22).

Integrating (23) with x_s, y_s and θ_s (the values of x, y and θ at τ_s), and considering that the DDR rotates in place at maximum speed, we obtain,

$$\begin{aligned}x &= \left(x_s + \frac{r_p}{v^*}\right)\cos\left(\frac{V^{\max}}{r_p}v^*(\tau-\tau_s)\right) \\ &\quad - y_s \sin\left(\frac{V^{\max}}{r_p}v^*(\tau-\tau_s)\right) - \frac{r_p}{v^*} \\ y &= \left(x_s + \frac{r_p}{v^*}\right)\sin\left(\frac{V^{\max}}{r_p}v^*(\tau-\tau_s)\right) \\ &\quad + y_s \cos\left(\frac{V^{\max}}{r_p}v^*(\tau-\tau_s)\right) \\ \theta &= \theta_s - \frac{V^{\max}}{r_p}v^*(\tau-\tau_s) + \left(\frac{u_2^*-u_1^*}{2b}\right)(\tau-\tau_s)\end{aligned} \tag{27}$$

The trajectories in the previous equation are valid until the system reaches the EDS.

In this game, we found the existence of a void region at $\phi = 0$. The switch function S and its first retro-time derivative $\mathring{S}$ vanishes for these configurations. Examining its second retro-time derivative $\mathring{\mathring{S}}$, we obtain

$$v^* = \text{sgn}(v^*) \tag{28}$$

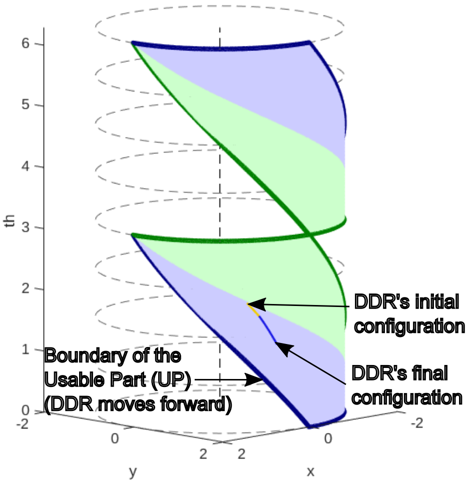
Thus, v^* can take the values -1, 0, or 1. As in the previous work [1], [6], [9], this points to the existence of a universal surface (US). On that surface, the DC pursuer applies $v^* = 0$ which can be verified using the Isaacs' necessary condition for the existence of universal surfaces [1]. The retro-time versions of the motion equations in the US are

$$\begin{aligned}\mathring{x} &= -\left(\frac{u_1+u_2}{2}\right)\sin\theta \\ \mathring{y} &= V^{\max} - \left(\frac{u_1+u_2}{2}\right)\cos\theta \\ \mathring{\theta} &= \left(\frac{u_2-u_1}{2b}\right)\end{aligned} \tag{29}$$

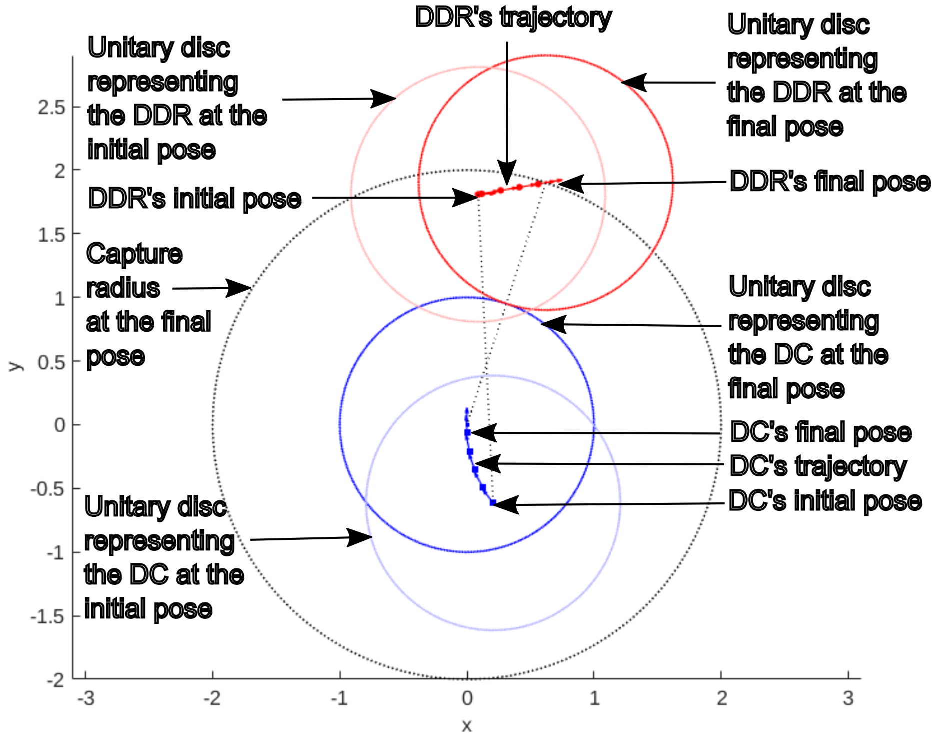
Solving the previous equations with the initial conditions $x_f = 0$, $y_f = l_c$ and $\theta = \psi$, and also with the DDR optimal controls u_1^* and u_2^*, we obtain,

$$\begin{aligned}x &= \mp\tau V^{\max}\sin\psi \\ y &= l_c + \tau(V^{\max} \mp V^{\max}\cos\psi) \\ \theta &= \psi\end{aligned} \tag{30}$$

the sign - is taken when the DDR moves forward at the end of the game and the sign + is taken if the DDR moves backward.



(a) The blue and yellow curves show the trajectory followed by the DDR in the reduced space.

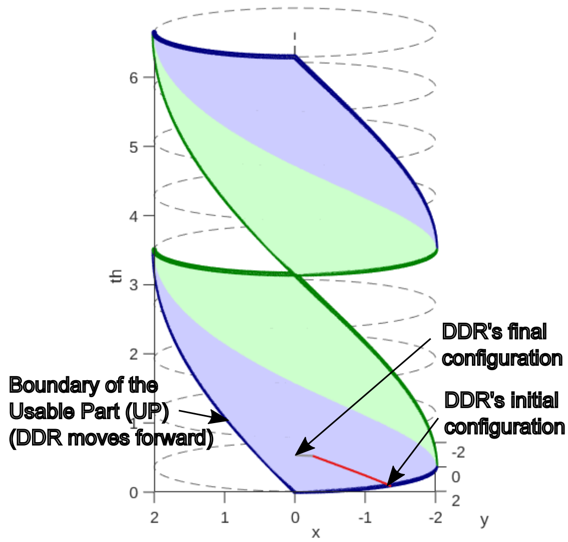


(b) Trajectory of the players in the realistic space. The blue curve indicates the trajectory followed by the DC pursuer, and the red curve corresponds to the trajectory followed by the DDR evader. Arrows indicate the directions of the players.

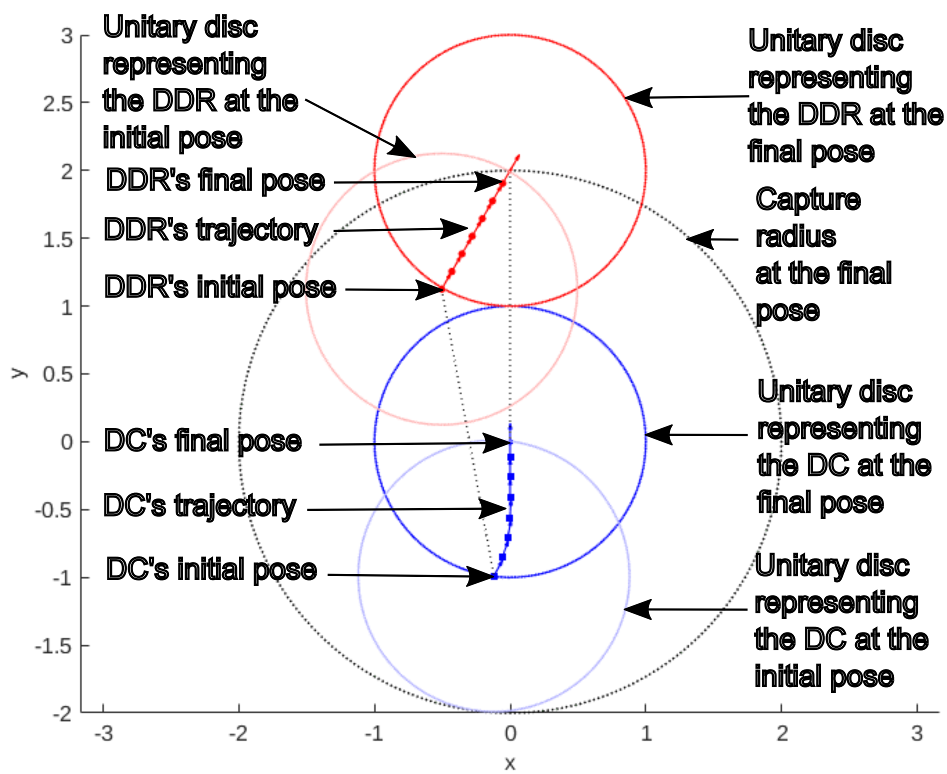
FIGURE 4. Results of the first simulation. The evader starts at the EDS.

The tributary trajectories joining the PUS at each point are computed using the values $v^* = -1$ or $v^* = 1$. The equations that describe them are obtained by integrating the retro-time motion equations in (23) and considering, as initial conditions (x_{US}, y_{US}), the point in the US where the tributary joins it,

$$\begin{aligned} x &= -\frac{r_p}{v^*} + \left(x_{US} + \frac{r_p}{v^*}\right)\cos\left(\frac{V^{\max}}{r_p}v^*\tau\right) \\ &\quad - y_{US}\sin\left(\frac{V^{\max}}{r_p}v^*\tau\right) \\ &\quad + \left(\frac{u_1^* + u_2^*}{2}\right)\sin\left(\frac{V^{\max}}{r_p}v^*\tau - \psi\right)\tau \\ y &= \left(x_{US} + \frac{r_p}{v^*}\right)\sin\left(\frac{V^{\max}}{r_p}v^*\tau\right) \\ &\quad + y_{US}\cos\left(\frac{V^{\max}}{r_p}v^*\tau\right) \\ &\quad - \left(\frac{u_1^* + u_2^*}{2}\right)\cos\left(\frac{V^{\max}}{r_p}v^*\tau - \psi\right)\tau \\ \theta &= \psi - \frac{V^{\max}}{r_p}v^*\tau \end{aligned} \tag{31}$$



(a) The gray and red curves show the trajectory followed by the DDR in the reduced space.



(b) Trajectory of the players in the realistic space. The blue curve indicates the trajectory followed by the DC pursuer, and the red curve corresponds to the trajectory followed by the DDR evader. Arrows indicate the directions of the players.

FIGURE 5. Results of the second simulation. The evader starts on a tributary trajectory of the PUS.

The tributary trajectories of the PUS join smoothly with the corresponding trajectories in the primary solution, see Fig. 3f. Note that (24) and (31) share a similar structure, and evaluating both equations with $(x_f, y_f, \psi) = (x_{US}, y_{US}, \psi) = (0, l_c, \psi)$ produces the same trajectories.

VI. DECISION PROBLEM

Part of the solution process of a pursuit-evasion game involves finding the initial conditions that allow winning for each player, in our case, that make capture possible for the DC or escape for the DDR. An important concept in differential game theory is the barrier, which separates the set of initial configurations into two disjoint regions. One is containing those that result in capture, and another with those that result in escape. As described in [1] and [8], to construct the barrier, a similar approach to the one used to compute the time-optimal motion strategies and their corresponding trajectories is applied. To answer the capture-escape question, we need to find out if the barrier divides the playing space into two parts.

The barrier trajectory is computed by performing a backward integration of the motion (23) and adjoint equations (17), taking as initial conditions the configurations on the terminal surface that belong to the BUP, defined in (10). However, similar to [3], in this problem, the construction of the barrier trajectories and validating if they define a closed region becomes very difficult, and we could not succeed in such a task. In particular, the solution involves using transcendental functions that do not allow us to compute the intersections of the trajectories analytically.

Although we were unable to compute the barrier analytically, in this game, we observed that capture is only attained from initial configurations near the UP. One can easily realize that given that both players have the same maximum speed, the DC cannot decrease the distance to the DDR once they have reached an alignment condition.

VII. SIMULATIONS

This section presents two simulations (see Figs. 4 and 5) of the players' motion strategies. In the first, the evader starts at the EDS shown in Fig. 4a. In the second case, the evader begins at one of the tributaries of the PUS; see Fig. 5a. The general parameters of the simulations were $V^{\max} = 1$ m/s, $l_c = 2$ m, $b = 1$ m, $r_p = 1$ m.

In the first simulation, we set $\phi = 0.3156$ rad and $\psi = 1.4$ rad. The trajectory followed by the DDR evader in the reduced space is shown in Fig. 4a. We can observe that the DDR travels a trajectory that departs from the EDS. Initially, it follows the yellow curve by performing a rotation in place at maximum speed; later, the DDR switches controls and follows the blue curve moving forward at maximum speed. The corresponding trajectories of the players in the realistic space are presented in Fig. 4b. In that figure, we can observe that while the DDR performs the previous strategy, the DC pursuer moves forward at maximum speed and performs a clockwise rotation until it captures the DDR.

In the second simulation, we set $\phi = 0$ rad and $\psi = \frac{\pi}{6}$ rad. The trajectory followed by the DDR evader in the reduced space is shown in Fig. 5a. Initially, it follows the red curve (tributary trajectory of the PUS), and later, it follows the gray curve (PUS) until capture is achieved. The corresponding trajectories of the players in the realistic space are presented in Fig. 4b. In that figure, we can observe that the DDR translates at maximum speed when the system travels both the tributary and the US trajectories. On the other hand, the DC pursuer moves forward and rotates counterclockwise when it travels the tributary trajectory of the US. Once that surface is reached, it moves following a straight line until it captures the DDR.

VIII. CONCLUSION

This paper addressed the problem of capturing a Differential Drive Robot with a Dubins Car in minimum time. Both players have the same maximum speed and a bounded turning ratio; hence, the game's outcome only depends on the particular motion capabilities of the players. The time-optimal motion strategies of the players to achieve their tasks were computed. In particular, we unveil the existence of three singular surfaces: an evader's dispersal surface (EDS), a pursuer's universal surface (PUS), and a transition surface (TS) where the DDR switches its controls. Some numerical examples of the players' trajectories in both the reduced and realistic space were presented.

In [3], the reverse game was studied; a Differential Drive Robot wants to capture a Dubins Car in minimum time. However, despite the similarities in the game formulation with [3], changing the players' roles in our current work produces a completely different solution. In particular, we reveal the existence of a pursuer's universal surface that has no counterpart in the solution presented in [3]. This result resembles those presented in [9] and [6] that consider a DC pursuer; however, note that the evader's motion constraints (DDR) in our game were not considered in those works.

Comparing the set of final configurations and the regions of the playing space covered with trajectories in [3], for the case where a DDR pursuer wants to capture a DC evader, and the corresponding ones found in this current work, for the case where a DC pursuer wants to capture a DDR evader, we can establish that a DDR pursuer has an advantage over a DC pursuer. A similar conclusion can be drawn between DDR and DC evaders. This advantage is provided mainly by the fact that the DDR can move forward or backward while the DC is always forced to move forward. An interesting future problem is considering a Reeds-Shepp Car (it can move forward and backward) instead of a Dubins Car player. In that case, the solution will depend solely on the rotational capabilities of both players.

In future work, we also plan to compute motion strategies for several DCs cooperating to capture the DDR. Additionally, we are interested in computing time-optimal strategies considering visibility constraints, such as a limited field of view or a bounded range.

REFERENCES

[1] R. Isaacs, *Differential Games*. New York, NY, USA: Wiley, 1965.

[2] T. Başar and G. J. Olsder, *Dynamic Noncooperative Game Theory* (SIAM Series in Classics in Applied Mathematics), 2nd ed. Philadelphia, PA, USA: Academic, 1999.

[3] U. Ruiz, "Capturing a Dubins car with a differential drive robot," *IEEE Access*, vol. 10, pp. 81805–81815, 2022.

[4] U. Ruiz, R. Murrieta-Cid, and J. L. Marroquin, "Time-optimal motion strategies for capturing an omnidirectional evader using a differential drive robot," *IEEE Trans. Robot.*, vol. 29, no. 5, pp. 1180–1196, Oct. 2013.

[5] U. Ruiz and R. Murrieta-Cid, "A differential pursuit/evasion game of capture between an omnidirectional agent and a differential drive robot, and their winning roles," *Int. J. Control*, vol. 89, no. 11, pp. 2169–2184, Feb. 2016.

[6] A. W. Merz, "The game of two identical cars," *J. Optim. Theory Appl.*, vol. 9, no. 5, pp. 324–342, May 1972.

[7] L. Bravo, U. Ruiz, and R. Murrieta-Cid, "A pursuit–evasion game between two identical differential drive robots," *J. Franklin Inst.*, vol. 357, no. 10, pp. 5773–5808, Jul. 2020.

[8] J. Lewin, *Differential Games: Theory and Methods for Solving Game Problems With Singular Surfaces*, London, U.K.: Springer, 2012.

[9] A. W. Merz, "The homicidal chauffeur—A differential game," Dept. Aeronaut. Astronaut., Ph.D. dissertation, Stanford Univ., Stanford, CA, USA, 1971.

[10] A. Friedman, *Differential Games*. New York, NY, USA: Wiley, 1971.

[11] U. Ruiz, "A game of surveillance between an omnidirectional agent and a differential drive robot," *Int. J. Control*, vol. 95, no. 6, pp. 1694–1706, Jan. 2021.

[12] U. Ruiz, "Time-optimal escape of an omnidirectional agent from the field of view of a differential drive robot," *Int. J. Control, Autom. Syst.*, vol. 21, no. 1, pp. 292–305, Jan. 2023.

[13] N. Karnad and V. Isler, "Lion and man game in the presence of a circular obstacle," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, St. Louis, MO, USA, Oct. 2009, pp. 5045–5050.

[14] R. Bera, V. R. Makkapati, and M. Kothari, "A comprehensive differential game theoretic solution to a game of two cars," *J. Optim. Theory Appl.*, vol. 174, no. 3, pp. 818–836, Jul. 2017.

[15] A. Chaudhari and D. Chakraborty, "A time-optimal feedback control for a particular case of the game of two cars," *IEEE Trans. Autom. Control*, vol. 67, no. 4, pp. 1806–1821, Apr. 2022.

[16] D. J. Balkcom and M. T. Mason, "Time optimal trajectories for bounded velocity differential drive vehicles," *Int. J. Robot. Res.*, vol. 21, no. 3, pp. 199–217, Mar. 2002.

[17] L. S. Pontryagin, V. G. Boltyanskii, R. V. Gamkrelidze, and E. F. Mishchenko, *The Mathematical Theory of Optimal Processes*. New York, NY, USA: Wiley, 1962.

UBALDO RUIZ received the Ph.D. degree in computer science from Centro de Investigación en Matemáticas (CIMAT), Guanajuato, Mexico, in 2013. From 2013 to 2014, he was a Postdoctoral Fellow with the Computer Science Department, University of Minnesota. From 2014 to 2022, he was a CONACYT Research Fellow. He is currently a Researcher with Centro de Investigación Científica y de Educación Superior de Ensenada (CICESE), Baja California, Mexico. His research interests include robotics, differential games, optimal control, and motion planning.