Proceedings of the The 60th Israel Annual Conference on Aerospace Sciences, Tel-Aviv & Haifa, Israel, March 4-5, 2020

Numerical Study of Different Variants of Dubins' Car Model

Nataly V. Munts*

Krasovskii Institute of Mathematics and Mechanics, Russian Academy of Sciences, Yekaterinburg, 620990, Russia

A numerical study of the famous model control problem "Dubins' car" has been performed in the framework of this paper. The variants are considered with symmetric and asymmetric constraints for the turn rates. The main objective is to construct reachability sets upto instant or, what is the same, the level sets for the time optimal result function. The original three-dimensional dynamics has been investigated as well as the reduced two-dimensional one. Examples of numerically computed optimal time result function are given for several typical situations.

I Introduction

One of the most famous models in the mathematical control theory, differential games, and engineering practice is so-called "Dubins' car". It describes a car (an aircraft, a vessel) moving in a horizontal plane and having a constant magnitude of the linear velocity. The maneuvers can be performed by means of rotating the direction of the velocity vector, but the angular velocity of the rotation is bounded. In the classical study, the lower and upper bounds are symmetrical with respect to the zero, that is, the cars turns to left and to right symmetrically.

Despite of the name, the model was formulated at first time in a work [1] by Russian mathematician A.A. Markov (senior). In that paper, he studied a problem of constructing railways, which should connect two given points in the plane, be of the possible minimal length, and have radius of turns bounded from below. At the next time, this problem is explored in a work [2] by L.E. Dubins as a pure geometric problem: of what kind should be a curve with bounded curvature that connects two given points and (possibly) has given tangents at these points. One can easily see that these two formulations are equivalent to each other and to the car model.

This model have appeared in many studies. R. Isaacs used it in his famous problem "homicidal chauffeur" [3,4] where a Dubins' car pursues a pedestrian, which has a simple motion dynamics. The game has been completely solved by A.W. Merz [5,6]. Later, there were several formulations of games, which also used this model [7–10].

In 1990s, on the base of the Dubins' car model, J.A. Reeds and L.A. Shepp suggested a new model [11] where the car can move both forward and backward, that is, the new model includes two controls: of the course direction rotation and of the linear velocity magnitude, which can be negative (what corresponds to the backward motion).

The Dubins' car model is widely used in both theoretical and practical investigations. Reachability sets of this system in the geometric plane have been studied independently in [12,13]. A very deep and detailed investigation of this model for the cases of symmetric, asymmetric, and one-sided turns is set forth in [14–18]. Many applications of this model

*Junior Researcher, natalymunts@gmail.com

in robotics can be found in [19]. Also, there are applications in aircraft control (see, for example, [20–23]). Some of the mentioned works, in particular, deal with situations of one-sided turns of the Dubins' car (an emergency aircraft), that is, with situations when the lower and upper constraints for the turning rate are not symmetrical with respect to the zero. The Dubins' car has its applications in studying motion of naval vessels (see, for example, [24–26]).

The main tool for investigating problems connected to the Dubins' car model is the Pontryagin's Maximum Principle (PMP). Due to relative simplicity of the dynamics, PMP can be easily written and more or less easily solved. But because of non-linearity of the dynamics, PMP provides necessary conditions only, and its solution gives a family of extremal trajectories, which are only candidates to be optimal motions of the system (that is, the motions, which leads the system to the boundary of reachability set). So, afterwards, one needs to exclude from consideration those trajectories, which indeed are not optimal, and this task is much less easier.

Thus, another approach can be more productive. Namely, the approach when the primary object under consideration is not a motion of the system, but a collection of reachability sets. Having a point on the boundary of a reachability set, it is quite simple to construct the trajectory, which attains this point. But to use this approach, one needs to have the reachable sets of the system. And the sets can be obtained numerically.

During several years, the author investigates numerical grid method for solving time-optimal differential games with lifeline. The notion of *lifeline* was introduced by R. Isaacs in his book [4]: this is a set, after reaching which the second player wins unconditionally. These investigations have been stimulated by the fact that practical realizations of grid methods can process a finite number of nodes, which therefore covers a bounded domain only. This leads to necessity to define some boundary conditions on the outer boundary of this domain, which are often defined as plus infinity what gives unconditional win of the second player. The algorithm computes an approximation of the value function (the time of reaching the target set guaranteed by the first player, who tries to minimize this time). The value function is computed as a generalized solution (in minimax [27] or viscous [28–31] sense) of a boundary problem for a partial differential equation of Hamilton – Jacobi type (HJE), which corresponds to the original game.

This method can be also applied to compute reachability sets of a controlled system. To do that, the system is turned to a game by introducing a fictitious second player, whose control is constrained by a single-point set coinciding with the origin and occurs in the dynamics as an additive term. The lifeline in this case appears as the boundary of state constraints, that is, the boundary of a set, in which the system should be kept. Of course, such a problem is not equivalent to a problem without state constraints. In my research [32], a statement is suggested, which helps to characterize the area where the optimal result functions coincide for the problems with and without state constraints.

The paper is organized as follows. In Section II, the Dubins' car model is introduced. Section III describe the game problem, which can be solved by the suggested numerical method, and the method itself. Results of numerically constructed reachability sets for different typical variants of the system parameters are given in Section IV.

II Dubins' Car Model

The Dubins' car model is the simplest model describing an automobile (ship, aircraft) movement on the plane with constant magnitude linear velocity and bounded turn radius.

The dynamics of this system is the following:

$$\dot{\mathbf{x}} = \begin{pmatrix} \dot{X} \\ \dot{Y} \\ \dot{\theta} \end{pmatrix} = f(\mathbf{x}, a) = \begin{pmatrix} \cos\theta \\ \sin\theta \\ a \end{pmatrix},$$

where X and Y are geometric coordinates of the automobile in the plane, θ is the angle between the positive direction of the axis X and direction of the automobile movement, $a \in \mathcal{A} = [a_1, 1]$ is a control input, which is the instantaneous angular velocity of the movement direction rotation. The linear velocity magnitude is assumed to be 1. The parameter a_1 describes the lower constraint for the turn rate. If $a_1 = -1$, then one has the classical formulation of the Dubins' car model when the left and right turns are symmetrical. If $-1 < a_1 < 0$, then the car still can turn to both left and right, but the right turn has a large turn radius. If $a_1 = 0$, then the right turn is prohibited, but the car has capability of a straight-forward motion, And, finally, if $0 < a_1 < 1$, then the car can not neither turn right, nor go straightly, but can only turn left with different turn radii.

This problem allows dimensional reduction suggested by R. Isaacs in his book [4]. A moving coordinate system is introduced, which origin is joined with the automobile, and the axis x_2 is directed along the instantaneous vector of the linear velocity. The axis x_1 is introduced to be orthogonal to the axis x_2. The motion equations in new coordinates are as follows:

$$\dot{x} = \begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} = \mathbf{f}(x, a) = \begin{pmatrix} -x_2 a \\ x_1 a - 1 \end{pmatrix}.$$

Control input $a \in \mathcal{A} = [a_1, 1]$ has the same meaning of the instantaneous angular velocity of movement direction rotation. Note that after this passage to the reduced dynamics, the system is mirrored: the left turn of the original system corresponds to the right turn of the reduced one, and vice versa.

The systems are considered in a finite time interval $[0, T]$ with the initial positions $X(0) = Y(0) = \theta(0) = 0$ and $x_1(0) = x_2(0) = 0$. Admissible controls are open-loop, that is, the set $\mathbb{A}$ of the feasible controls consists of all measurable (or, which is the same, piecewise-continuous) functions $a(\cdot) : [0, T] \to \mathcal{A}$. Under this assumption, a trajectories $\mathbf{x}\big(\cdot; a(\cdot)\big)$ and $x\big(\cdot; a(\cdot)\big)$ of the systems generated by a control $a(\cdot)$ are just solutions of integral equation

$$\mathbf{x}(t) = \int_0^t \mathbf{f}\big(\mathbf{x}(t), a(t)\big)\, dt$$

and, respectively,

$$x(t) = \int_0^t f\big(x(t), a(t)\big)\, dt.$$

During numerical constructions these equations can be solved approximately by means of the Euler or Runge–Kutta methods.

The objective is to obtain numerically reachability sets of these systems upto instant t, that is, the sets

$$\mathbf{G}(t) = \big\{\mathbf{x} = (X, Y, \theta) \,|\, \exists\, a(\cdot) \in \mathbb{A}\, \exists\, 0 \le \tau \le t : \mathbf{x}\big(\tau; a(\cdot)\big) = \mathbf{x}\big\}$$

and, respectively,

$$G(t) = \big\{x = (x_1, x_2) \,|\, \exists\, a(\cdot) \in \mathbb{A}\, \exists\, 0 \le \tau \le t : x\big(\tau; a(\cdot)\big) = x\big\}.$$

III Numerical Method

III.A Formulation of Game with Lifeline

Consider a conflict controlled system

$$\dot{x} = f(x, a, b), \qquad t \geq 0,\ a \in A,\ b \in B, \tag{1}$$

where $x \in \mathbb{R}^n$ is the phase vector of the system; a and b are the controls of the first and second players constrained by compact sets A and B in their Euclidean spaces. A compact set $\mathcal{T}$ and an open set $\mathcal{W} \subset \mathbb{R}^n$ are given such that $\mathcal{T} \subset \mathcal{W}$ and the boundary $\partial\mathcal{W}$ is bounded. Denote $\mathcal{G} := \mathcal{W} \setminus \mathcal{T}$ and $\mathcal{F} := \mathbb{R}^n \setminus \mathcal{W}$ (see Fig. 1). The game takes place in the set $\mathcal{G}$; the objective of the first player is to guide the system to the set $\mathcal{T}$ as soon as possible keeping the trajectory outside the set $\mathcal{F}$; the objective of the second player is to guide the system to the set $\mathcal{F}$, or if it is impossible, to keep the trajectory inside the set $\mathcal{G}$ forever, or if the latter is impossible too, to postpone the reaching the set $\mathcal{T}$ as long as he can. Control problems also match this formulation, because one can use an additive second player control b constrained by a set B consisting of the zero vector only: $f(x, a, b) = g(x, a) + b$, $b \in B = \{\mathbf{0}\}$.

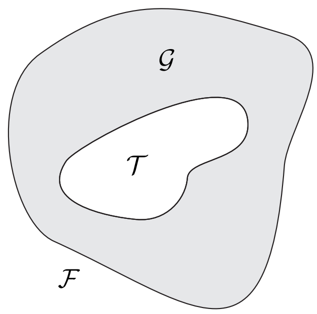

Figure 1. The sets $\mathcal{T}$, $\mathcal{F}$, and $\mathcal{G}$

Such a game can be called a *game with lifeline*; the boundary $\partial\mathcal{F}$ of the set $\mathcal{F}$ is the lifeline where the second player wins unconditionally.

Assume that the following conditions are fulfilled:

C.1 The function $f : \mathbb{R}^n \times A \times B \mapsto \mathbb{R}^n$ is continuous in all variables and Lipschitz continuous in variable x: for all $x^{(1)}$, $x^{(2)} \in \mathbb{R}^n$, $a \in A$, $b \in B$

$$\big\| f(x^{(1)}, a, b) - f(x^{(2)}, a, b) \big\| \leq L \| x^{(1)} - x^{(2)} \|;$$

moreover, it satisfies Isaacs' condition:

$$\min_{a\in A} \max_{b\in B} \big\langle p, f(x, a, b) \big\rangle = \max_{b\in B} \min_{a\in A} \big\langle p, f(x, a, b) \big\rangle \quad \forall p \in \mathbb{R}^n;$$

Here and below, the symbol $\langle \cdot, \cdot \rangle$ stands for scalar product.

C.2 The boundary $\partial\mathcal{G}$ of the set $\mathcal{G}$ (that is, the boundaries $\partial\mathcal{T}$ and $\partial\mathcal{F}$) is smooth, compact, and has a bounded curvature.

The players' aims of this kind can be formalized in the following way. Let the function $x(\cdot; x_0)$ be a trajectory of the system emanated from the initial point $x(0) = x_0$. Two instants are considered:

$$t_* = t_*\big(x(\cdot, x_0)\big) = \min\big\{t \geq 0 : x(t; x_0) \in \mathcal{T}\big\},$$
$$t^* = t^*\big(x(\cdot, x_0)\big) = \min\big\{t \geq 0 : x(t; x_0) \in \mathcal{F}\big\},$$

which are the instants when the trajectory $x(\cdot;x_0)$ hits for the first time the sets $\mathcal{T}$ and $\mathcal{F}$, respectively. If the trajectory doesn't arrive at the set $\mathcal{T}$ $(\mathcal{F})$, then the value t_* (t^*) is equal to $+\infty$. The reasonings can be conducted using either formalization with nonanticipating strategies, or positional formalization of N.N. Krasovskii and A.I. Subbotin [33,34]. In the last case, the feedback strategies of the first and the second player are arbitrary functions $a(\cdot) : \mathbb{R}^n \mapsto A$ and $b(\cdot) : \mathbb{R}^n \mapsto B$, respectively.

Define the result of the game on the trajectory $x(\cdot;x_0)$ as

$$\tau\big(x(\cdot;x_0)\big) = \begin{cases} +\infty, & \text{if } t_* = +\infty \text{ or } t^* < t_*, \\ t_*, & \text{otherwise.} \end{cases} \tag{2}$$

In [35], the author proves that a time-optimal problem with lifeline has the value function:

$$T(x_0) = \inf_{a(\cdot)\in\mathbb{A}} \sup_{b(\cdot)\in\mathbb{B}} \tau\big(x(\cdot;x_0)\big) = \sup_{b(\cdot)\in\mathbb{B}} \inf_{a(\cdot)\in\mathbb{A}} \tau\big(x(\cdot;x_0)\big).$$

Here, the extremes are searched over the classes $\mathbb{A}$ and $\mathbb{B}$ of all admissible strategies of the first and the second player (nonanticipating or feedback). The strategies $a(\cdot)$ and $b(\cdot)$ define the trajectory $x(\cdot;x_0)$ according to the chosen ideology (of nonanticipating or feedback strategies).

The unboundedness of the value function and cost functional can cause some uneasiness in numerical researching game (1), (2). For this reason, one often substitutes the unbounded cost functional with a bounded one by means of *Kruzhkov's transform*:

$$J\big(x(\cdot,x_0)\big) = \begin{cases} 1 - \exp\big(-\tau\big(x(\cdot;x_0)\big)\big), & \text{if } \tau\big(x(\cdot;x_0)\big) < +\infty, \\ 1, & \text{otherwise.} \end{cases} \tag{3}$$

In such a case, the value function also becomes bounded and its magnitude belongs to the range from zero to unit:

$$v(x_0) := \inf_{a(\cdot)\in\mathbb{A}} \sup_{b(\cdot)\in\mathbb{B}} J\big(x(\cdot;x_0)\big) = \sup_{b(\cdot)\in\mathbb{B}} \inf_{a(\cdot)\in\mathbb{A}} J\big(x(\cdot;x_0)\big).$$

III.B Boundary Value Problem for HJE

Generally speaking, numerical scheme construction and justification of its convergence are analogous to paper [31] where the numerical scheme for the classic time-optimal problem is constructed and it's convergence is proved. Herewith, the value function is characterized as the unique generalized (viscosity) solution of the corresponding boundary value problem for HJE:

$$\begin{aligned} &z + H(x, Dz) = 0,\ x \in \mathcal{G}, \\ &z(x) = 0,\ \text{if } x \in \partial\mathcal{T}, \\ &z(x) = 1,\ \text{if } x \in \partial\mathcal{F}. \end{aligned} \tag{4}$$

Here and below, the symbol Dz denoted the gradient of the function z. The function H is called the *Hamiltonian* and in the case of dynamics (1) is defined as follows:

$$H(x,p) = \min_{a\in A} \max_{b\in B} \big\langle -f(x,a,b)\cdot p\big\rangle - 1, \quad x \in \mathcal{G},\ p \in \mathbb{R}^n.$$

A complexity of solving problems of this type is that the necessary condition of the existence of a classical solution of a partial differential equation is double differentiability of the function in the left-hand side of the equality. However, in the case of the time-optimal game, one can only speak of continuity of the Hamiltonian, so, there is no guarantee that HJE has a classical solution.

The notion of the *generalized viscosity solution* introduced in [36] has been used to deal with this problem. In book [27], an alternative method of obtaining a generalized solution of HJE was introduced. It is called *generalized minimax solution*. Also in book [27], it is proved that viscosity and minimax solutions coincide in the points of continuity.

In [37, 38], the author proves that the value function of game (1), (3) is a viscosity solution of problem (4). The proof was performed under the assumption of the dynamical advantage of each player on the boundaries of the corresponding sets:

$$\forall x \in \partial\mathcal{T} \ \min_{a\in A}\max_{b\in B} \big\langle n_{\mathcal{T}}(x), f(x,a,b)\big\rangle < 0,$$
$$\forall x \in \partial\mathcal{F} \ \max_{b\in B}\min_{a\in A} \big\langle n_{\mathcal{F}}(x), f(x,a,b)\big\rangle < 0.$$

Here, $n_{\mathcal{T}}(x)$ $\big(n_{\mathcal{F}}(x)\big)$ is a normal vector to the boundary $\partial\mathcal{T}$ $(\partial\mathcal{F})$ of the set $\mathcal{T}$ $(\mathcal{F})$ at the point x directed outward the corresponding set or (what is the same) inward the set $\mathcal{G}$. The sense of these relations is that if the system is at the boundary of the set $\mathcal{T}$ $(\mathcal{F})$, then the first (second) player can guarantee leading the trajectory of the system inside the corresponding set despite of the action of the opponent. Combination of these assumptions results in the continuity of the value function inside the set $\mathcal{G}$.

III.C Discrete Scheme

Replace the continuous dynamics with a discrete one with the time step $h > 0$:

$$x_n = x_{n-1} + hf(x_{n-1}, a_{n-1}, b_{n-1}),\ n = 1,\dots,N, \quad x_0 \text{ is set},$$

where $a_n \in A$ and $b_n \in B$.

Now, it is the turn of the space discretization. Consider a grid $\mathcal{L}$ with the step k covering the whole space $\mathbb{R}^n$ and consisting of nodes $q_{i_1,\dots,i_n} = (x_{i_1},\dots,x_{i_n})$, $i_1,\dots,i_n \in \mathbb{Z}$, $x_{i_j} = ki_j$. (Generally speaking, steps along different axes can differ, but this fact doesn't change the main idea of numerical scheme construction.) Here and below, mostly, a linear indexation q_ν, $\nu \in \mathbb{Z}$ for the nodes of the grid $\mathcal{L}$ is used. The symbol $\mathcal{L}_{\mathcal{T}}$ stands for the set of those nodes of the grid $\mathcal{L}$, which belong to the set $\mathcal{T}$, the symbol $\mathcal{L}_{\mathcal{G}}$ denotes the collection of nodes falling into the set $\mathcal{G}$, and the symbol $\mathcal{L}_{\mathcal{F}}$ stands for the set of nodes from the set $\mathcal{F}$. In theoretical constructions, the grid is assumed infinite.

For every point $x \in \mathbb{R}^n$, one can find a simplex $S(x)$ with vertices $\big\{q_l(x)\big\}$ from $\mathcal{L}$ such that the point x belongs to the simplex $S(x)$ and $S(x)$ does not contain other nodes of the grid. It is assumed that with choosing the grid $\mathcal{L}$, a separation of the game space to simplexes with vertices at nodes of the grid is also chosen. On the basis of $S(x)$, one can obtain the *barycentric (local) coordinates* $\lambda_l(x)$ of the point x with respect to the vertices $q_l(x)$ of the simplex $S(x)$:

$$x = \sum_{l=1}^{n+1} \lambda_l(x) q_l(x), \qquad \lambda_l(x) \geq 0,\ \sum_{l=1}^{n+1} \lambda_l(x) = 1.$$

Sometimes, the arguments of the coefficients λ and vertices q will be omitted if they are clear from the context.

Substitute the function $w_h(\cdot)$ with a new one $w(\cdot)$, which magnitudes $w(q_\nu)$ at the nodes q_ν of the grid $\mathcal{L}$ form an infinite vector $W = \big(w(q_\nu)\big)_{s\in\mathbb{Z}}$. The magnitude $w(x)$

at some point x, which is not a node of the grid, can be reconstructed by means of piecewise-linear approximation based on local coordinates of the point x:

$$w_{loc}(x, W) = \sum_{l=1}^{n+1} \lambda_l \, w(q_l). \tag{5}$$

The value function of discretized game can be characterized by means of Discrete Dynamic Programming Principle:

$$w(q_\nu) = \begin{cases} \gamma \max\limits_{b \in B} \min\limits_{a \in A} w_{loc}\big(z(q_\nu, a, b), W\big) + 1 - \gamma, & \text{if } q_\nu \in \mathcal{L}_{\mathcal{G}}, \\ 0, & \text{if } q_\nu \in \mathcal{L}_{\mathcal{T}}, \\ 1, & \text{if } q_\nu \in \mathcal{L}_{\mathcal{F}}. \end{cases}$$

Here $\gamma = e^{-h}$, $z(x, a, b) = x + hf(x, a, b)$.

This characterization is of recursive kind, because the magnitude $w(q_\nu)$ at some node q_ν depends on the magnitude of local reconstruction w_{loc}, which again depends on the magnitudes of the function $w(\cdot)$ at nodes of the grid, which may include the node q_ν. Such a kind of relations obtained is typical for dynamic programming principle. In the following, on the basis of this formula, an iterative numerical method of construction of vector W and function w is proposed. Moreover, from the definition of $w(\cdot)$, one can see that in a practical realization of the numerical method it is necessary to keep values of this function on nodes from $\mathcal{L}_{\mathcal{G}}$. If the set $\mathcal{G}$ is bounded, then $\mathcal{L}_{\mathcal{G}}$ contains only finite number of nodes and can be represented in a computer.

For the chosen grid $\mathcal{L} = \{q_\nu\}_{\nu \in \mathbb{Z}}$, denote by $\mathcal{M}$ the set of infinite vectors with elements $W = \big(w(q_\nu)\big)_{\nu \in \mathbb{Z}}$. Denote by $\mathcal{M}_1$ those vectors in the set $\mathcal{M}$, which elements $w(q_\nu)$ satisfy the inequality $0 \le w(q_\nu) \le 1$. For every $s \in \mathbb{Z}$, define an operator $F_s : \mathcal{M} \to \mathbb{R}$ using a vector $W = \big(w(q_\nu)\big)_{\nu \in \mathbb{Z}}$ in the following way:

$$F_s(W) = \begin{cases} \gamma \max\limits_{b \in B} \min\limits_{a \in A} w_{loc}\big(z(q_s, a, b), W\big) + 1 - \gamma, & \text{if } q_s \in \mathcal{L}_{\mathcal{G}}, \\ 0, & \text{if } q_s \in \mathcal{L}_{\mathcal{T}}, \\ 1, & \text{if } q_s \in \mathcal{L}_{\mathcal{F}}. \end{cases} \tag{6}$$

Here, $w_{loc} : \mathbb{R}^n \times \mathcal{M} \to \mathbb{R}$ is the local reconstruction (5) of the function $w(\cdot)$ corresponding to the vector W. The manifold of values of the operators F_s over all indices s (over all nodes q_s) defines an operator $F : \mathcal{M} \to \mathcal{M}$.

In [39], it is proved that if the time and space steps h and k are fixed, then the operator F is a contraction map and as a consequence, one can obtain that there exists a unique fixed point $\mathbf{W}$ of the operator F, which determines a function $\mathbf{w}(\cdot)$ in $\mathbb{R}^n$. This function depends on the time h and space k discretization steps. In [40], it is proved that if sequences of time and space steps $\{h_n\} \to 0$ and $\{k_n/h_n\} \to 0$ as $n \to \infty$, then the sequence of functions $\big\{\mathbf{w}_n(\cdot)\big\}$ converges to the value function after the Kruzhkov's transform.

III.D Realization of Numerical Procedure

So, the computer realization of this numerical procedure implies a realization of some storage for a grid, which covers some bounded parallelepipedal domain or bounded domain of some arbitrary shape. Since, the procedure is of iterational kind, two values should be stored in each node: the current value and value for the next iteration. This storage should provide an operation of computation of an approximation of the function in any point, both in a node and in some intermediate one. The nodes should cover the set $\mathcal{G}$ where the system moves; values in nodes from $\mathcal{L}_\mathcal{T}$ and $\mathcal{L}_\mathcal{F}$ equal constantly to 0 or 1, therefore, no need to allocate cell to keep this information.

As the initial values all nodes get 1. The operator monotonically decreases, so the initial values should be not less than the values to be computed.

Then for each node q_s, formula (6) is applied. Application of this formula to all nodes makes one iteration of the algorithm. Entire algorithm either consists of some prescribed number of iterations, or repeats the iteration procedure until some condition becomes true (for example, the values of the function in nodes stabilize).

The described algorithm and corresponding data structures have been realized as a program in C# 4.6 for computational environments .Net and .NetCore, therefore, the obtained program is cross-platform and can be started as under Microsoft Windows, as under Linux, and under macOS.

Note that the Dubins' car problem is of control type, not a game. Despite this, one can state that it can be considered as a game with a fictitious second player having a control constrained by a one-point set coinciding with the origin and embedded into dynamics as an additive term. Formally, the coincidence of the reachable sets for the control problem and level sets of the value function for the game can be explained in the following way. Stepwise motions of the game system written according to the formalization by N.N. Krasovskii and A.I. Subbotin will coincide with the Darboux sums of the integrals of the trajectories of the original controlled system (without the second player). When the diameter of the time partition tends to zero and the stepwise motions converge to the constructive motions (the Krasovskii motions), the Darboux sums converge to the value of the corresponding integral, that is, to the trajectories of the original control problem. Therefore, the bundles of motions in the systems with and without second player coincide, and the times of reaching the target set are the same for these systems. So, one can say about equivalence of the results of the controlled system and the game with a fictitious second player and applicability of the suggested computational algorithm for constructing reachable sets upto instant.

IV Examples of Reachable Sets

Four variants of the problem have been considered for computations:

1. for $a_1 = -1.0$, the case of two-sided symmetrical turn;
2. for $a_1 = -0.5$, the case of two-sided asymmetrical turn;
3. for $a_1 = 0.0$, the case of one-sided turn, the straight-forward motion is allowed;
4. for $a_1 = 0.5$, the case of one-sided turn, the straight-forward motion is prohibited.

For each variant, the optimal result function has been computed for two problems: for the original 3-dimensional problem in the coordinates X, Y, θ and for the reduced 2-dimensional one in the coordinates x_1, x_2.

The results of computation for each variant are given in 4 figures: the picture of level sets of the optimal result function for the reduced system and two views of 3-dimensional reachability sets $G(t)$ upto moment for three time instants: $t = 1.0$, $t = 2.5$, $t = 3.5$. It would be also interesting to compute the reachability sets for larger times in the range from 1.5π up to 6π.

The computations have been performed under Linux Kubuntu 18.04 with .NetCore 3.0 on a PC having CPU Intel Core i7 4702MQ, 2.2GHz, and 16Gb of RAM. (However, for three-dimensional problems the program takes about 1.8Gb of RAM only.) This processor has 4 cores with HyperThreading, so, the internal capabilities of C#/.NetCore for parallel processing of arrays have been used, what gives approximately 4.5-times acceleration in comparison with sequential 1-threaded computations.

Two-dimensional problems have been computed using the following parameters of discretization: 70 iterations, the time step $h = 0.15$, the space step for all axis $k = 0.025$, the state constraint set $\mathcal{G} = [-5.0, +5.0] \times [-3.0, +6.0]$. So, the grid contains $401 \cdot 361 = 144761$ nodes (the nodes from the part $\mathcal{L}_\mathcal{T}$ are also included to the stored grid.) The initial set $\mathcal{T} = [-0.05, +0.05] \times [-0.05, +0.05]$. The abstract minimization over the control set $A = [a_1, 1]$ of the controller has been changed by minimization over a grid with the step 0.05. The computation time was about 13 minutes for problems 1 and 2, and (somewhy) about 6 minutes for the the problems 3 and 4.

For three-dimensional problems, the following discretization parameters have been used: 50 iterations, the time step $h = 0.15$, the space step for all axis $k = 0.05$, the state constraint set $\mathcal{G} = [-5.0, +5.0] \times [-3.0, +6.0] \times [-3.25, +3.25]$. So, the grid contains $201 \cdot 181 \cdot 131 = 4765911$ nodes (the nodes from the part $\mathcal{L}_\mathcal{T}$ are also included to the stored grid.) The initial set $\mathcal{T} = [-0.2, +0.2] \times [-0.2, +0.2] \times [-0.2, +0.2]$. It is sufficiently non-point-like. Also, it would be interesting to organize more delicate computations with a more fine space grid and with a less-sized initial set. The abstract minimization over the control set $A = [a_1, 1]$ of the controller has been changed by minimization over a grid with the step 0.05. The computation time was a bit less than 2 hours for problems 1 and 2, and (somewhy) about 1 hour and 40 minutes for the the problems 3 and 4.

The decreasing of computational time for problems 3 and 4 can be explained by the fact that major part of the grid did not changed the stored values and kept units until the end of computations what saves writings to memory and, therefore, the computational time.

Figure 2 shows level set of the optimal result function for the problem with symmetrical turns ($a_1 = -1$). Figures 6, 10, and 14 show them for the cases of asymmetric ($a_1 = -0.5$) and one-sided ($a_1 = 0.0$ and $a_1 = 0.5$) turns. In these Figures, the blue color corresponds to small times and to the initial set $\mathcal{T}$, the red color corresponds to the time $+\infty$, that is, to the area, which cannot be achieved either at all because of the state constraints, or earlier some time. This time can be estimated as $70\,\text{iterations} \cdot 0.15\,(\text{time step})/3\,(\text{iterations per cell}) \approx 3.5$ (several iterations are necessary to get exact values in all vertices of a cell even the neighbor cell already have exact values). To obtain exact results for a wider area, one needs to enlarge the set $\mathcal{G}$ and to increase the number of iterations.

The triples of Figures 3–5, 7–9, 11–13, and 15–17 shows evolution of the 3-dimensional reachability sets for each of 4 problems. The reachability sets $G(t)$ are shown for the instants $t = 1.0$, $t = 2.5$, and $t = 3.5$. Each figure contains two views of the 3-dimensional sets.

One can see as the 2-dimensional sets lose symmetry and size along the axis x_2 as the turn capability of the controller decreases. In the case of 3-dimensional sets, the sets decrease their lengths along the axis θ what corresponds to decreasing the range of possible course angles of the car at the respective instant. This can be evidently seen in the case $a_1 = 0.0$ when the sets are just halves of the corresponding sets for $a_1 = -1.0$.

Some "ripple" of the pictures of 2-dimensional level sets of the optimal result functions and "angularity" of the 3-dimensional reachability sets are caused by "granularity" of the initial set. These effects can be decreased by decreasing the space discretization step k.

The considered examples have been taken from the works [15,18]. In general, one can

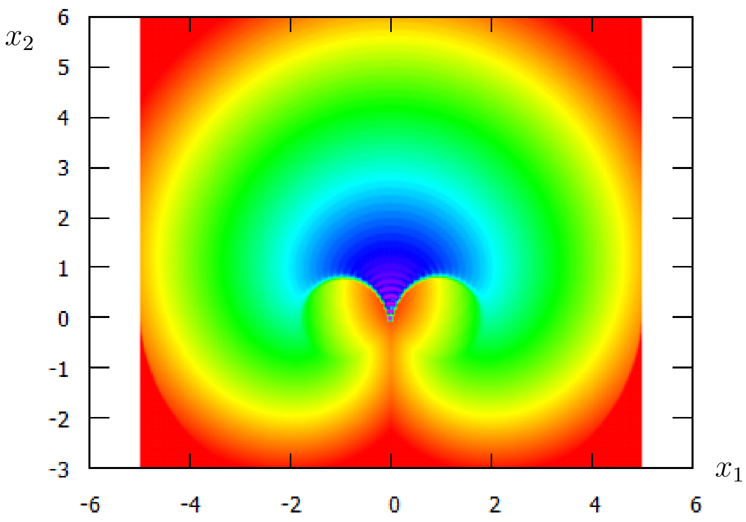

Figure 2. **The level sets of the optimal result function for the reduced dynamics,** $a_1 = -1$

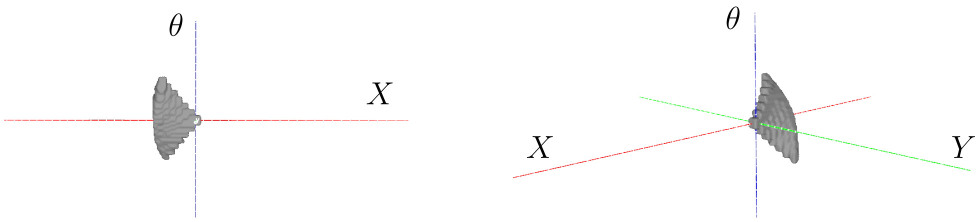

Figure 3. $a_1 = -1$, **the reachability set** $G(t)$ **for** $G(t)$ **for** $t = 1$

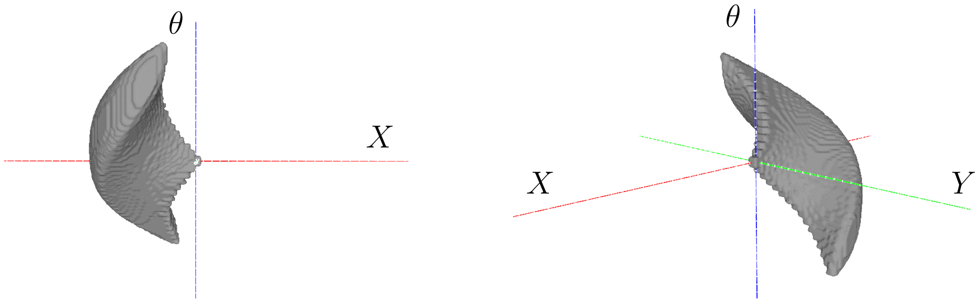

Figure 4. $a_1 = -1$, **the reachability set** $G(t)$ **for** $G(t)$ **for** $t = 2.5$

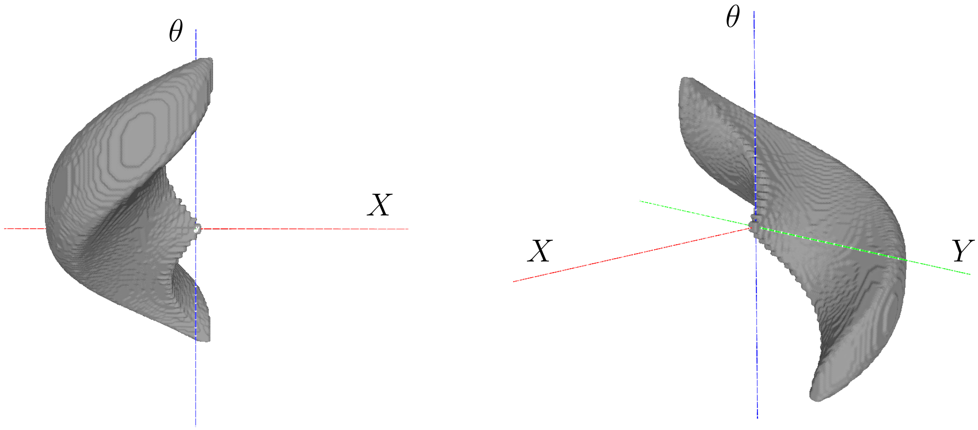

Figure 5. $a_1 = -1$, **the reachability set** $G(t)$ **for** $G(t)$ **for** $t = 3.5$

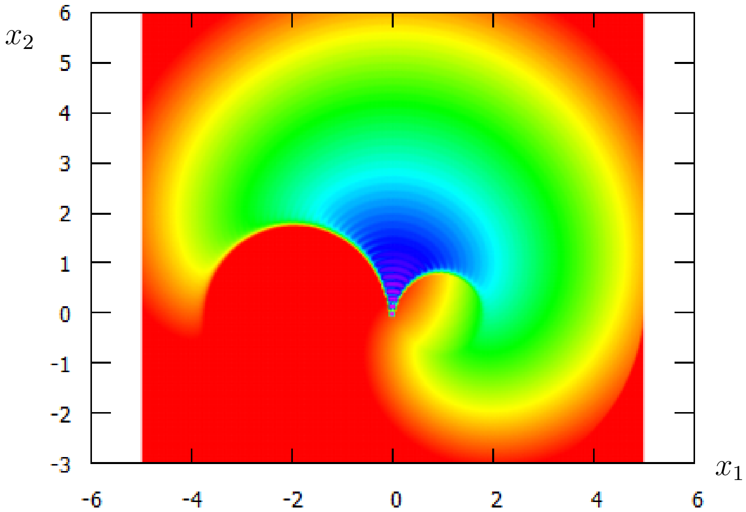

Figure 6. The level sets of the optimal result function for the reduced dynamics, $a_1 = -0.5$

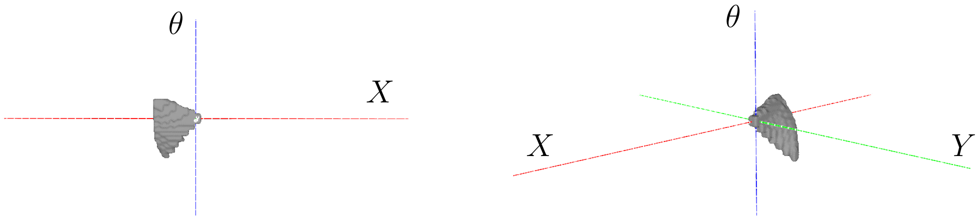

Figure 7. $a_1 = -0.5$**, the reachability set** $G(t)$ **for** $G(t)$ **for** $t = 1$

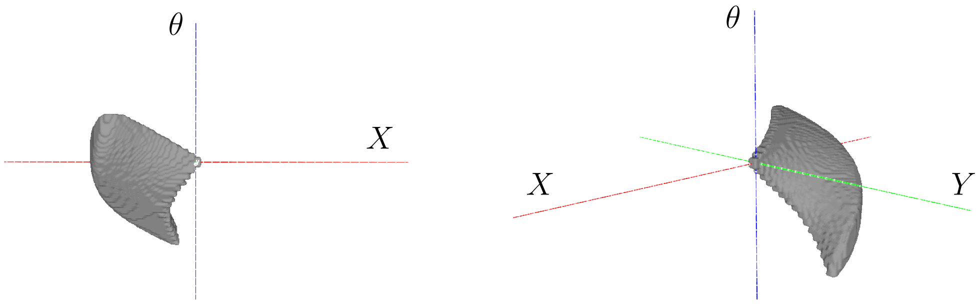

Figure 8. $a_1 = -0.5$**, the reachability set** $G(t)$ **for** $G(t)$ **for** $t = 2.5$

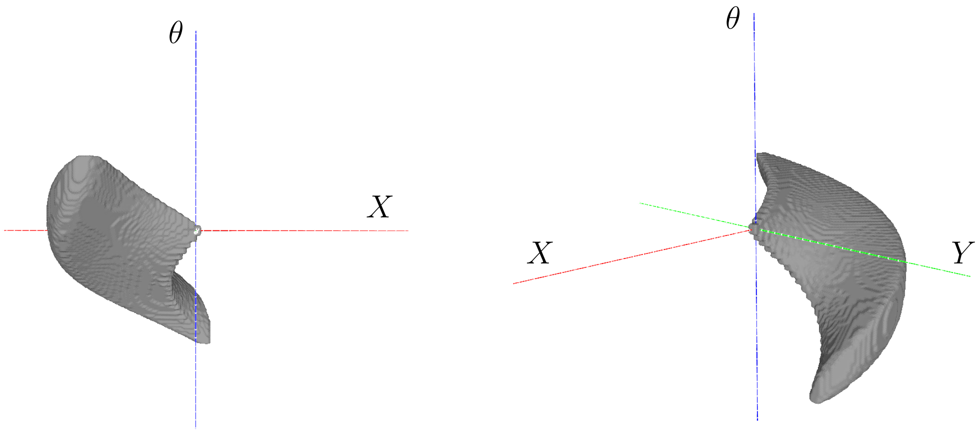

Figure 9. $a_1 = -0.5$**, the reachability set** $G(t)$ **for** $G(t)$ **for** $t = 3.5$

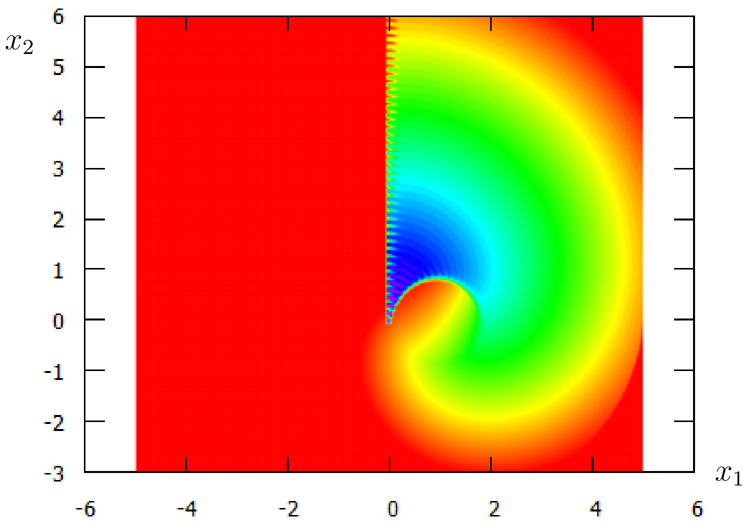

Figure 10. The level sets of the optimal result function for the reduced dynamics, $a_1 = 0$

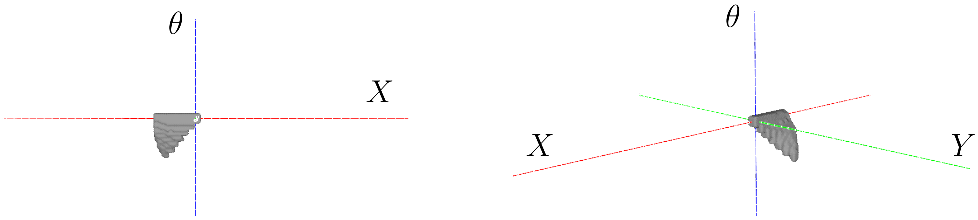

Figure 11. $a_1 = 0$, the reachability set $G(t)$ for $G(t)$ for $t = 1$

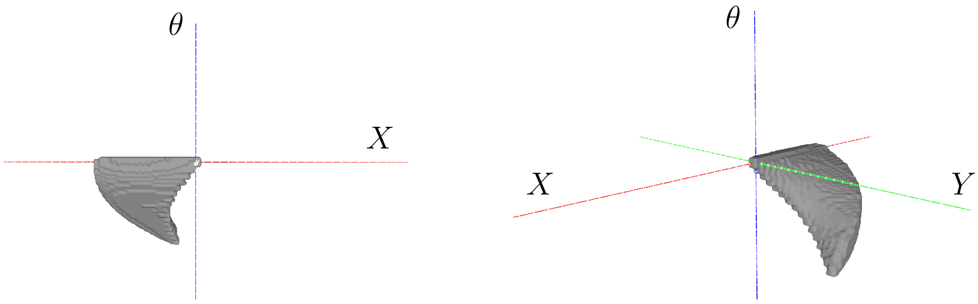

Figure 12. $a_1 = 0$, the reachability set $G(t)$ for $G(t)$ for $t = 2.5$

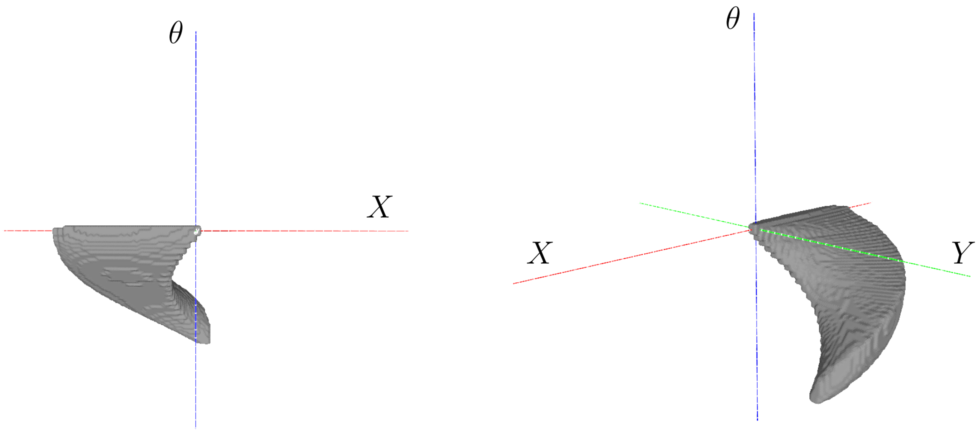

Figure 13. $a_1 = 0$, the reachability set $G(t)$ for $G(t)$ for $t = 3.5$

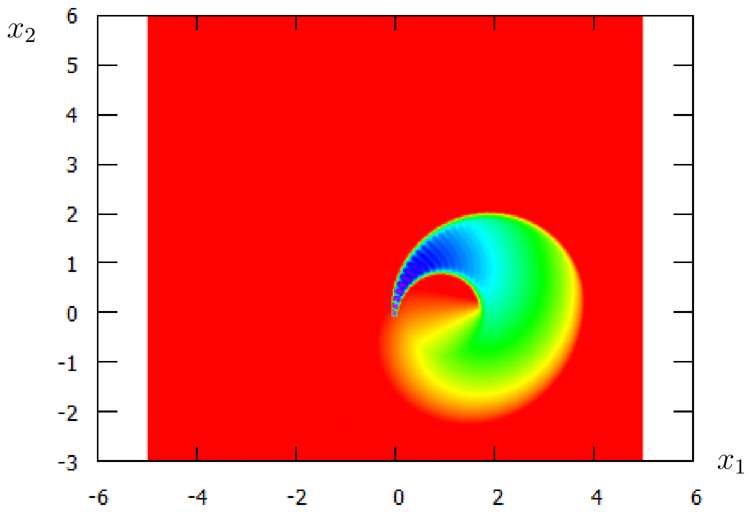

Figure 14. The level sets of the optimal result function for the reduced dynamics, $a_1 = 0.5$

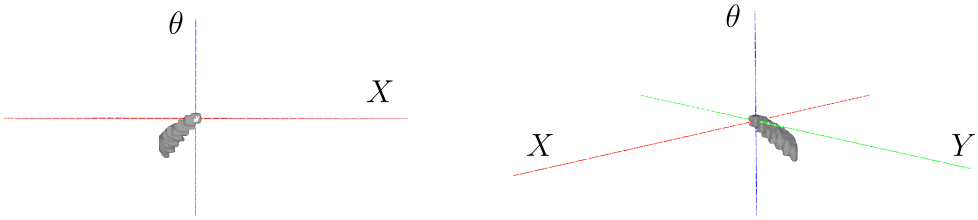

Figure 15. $a_1 = 0.5$, the reachability set $G(t)$ for $G(t)$ for $t = 1$

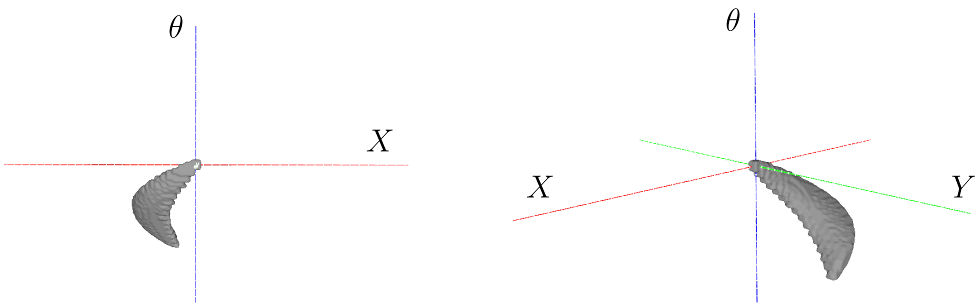

Figure 16. $a_1 = 0.5$, the reachability set $G(t)$ for $G(t)$ for $t = 2.5$

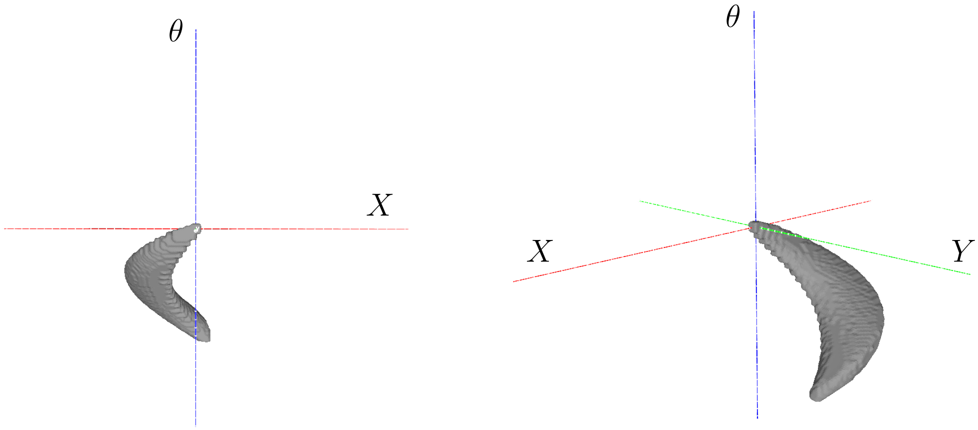

Figure 17. $a_1 = 0.5$, the reachability set $G(t)$ for $G(t)$ for $t = 3.5$

state that there is a quite good coincidence of our results and the ones obtained by those authors. However, we performed computations for such parameters of the grid, which provided more or less fast production of the result. Therefore, the grid is not too large and covers not too wide part of the game space. Also, level sets of the value function coinciding with the ones for the problem without lifeline correspond to times, which are not too large; namely, a bit larger than π, whereas in those works there are level sets for times 3π, 4π and larger. Potentially, these level sets can be computed, but it requires more time and memory.

V Conclusion

In the framework of this paper, a numerical algorithm for constructing the level sets for the optimal result function (reachability sets upto instant) is suggested. The computational procedure is iterative and recomputes values in some spacial grid, which converge to the magnitudes of the optimal result function. The algorithm has been implemented as a cross-platform computational program in C# under the environment .Net Core. By means of this program, level sets of the optimal result function of the Dubins' car model have been computed. Situations of different capabilities of the controller have been considered. Along with the classical formulation when the left and right turns are symmetric, variants of the problem are considered when the turns are asymmetrical or the turn is one-sided.

The used computational procedure is created originally oriented to solving time-optimal differential games with lifeline. Control problems with state constraints can be quite easily injected into this class of games, therefore, this procedure can solve control problems of the mentioned type.

The reachability sets obtained as a result of this study coincide more or less with those obtained analytically earlier.

References

[1] Markov, A. A., "Some examples of the solution of a special kind of problem on greatest and least quantities," *Soobscenija Charkovskogo matematiceskogo obscestva (Proceedings of Kharkov Mathematical Society)*, Vol. 2, No. 1(5, 6), 1889, pp. 250–276. (in Russian).

[2] Dubins, L. E., "On curves of minimal length with a constraint on average curvature and with prescribed initial and terminal positions and tangents," *Amer. J. Math.*, Vol. 79, No. 3, 1957, pp. 497–516.

[3] Isaacs, R., "Games of pursuit," Tech. rep., RAND Corporation, Santa Monica, 1951.

[4] Isaacs, R., *Differential games*, John Wiley and Sons, New York, 1965.

[5] Merz, A. W., *The homicidal chauffeur — a differential game*, Ph.D. thesis, Stanford University, 1971.

[6] Merz, A. W., "The homicidal chauffeur," *AIAA Journal*, Vol. 12, No. 3, 1974, pp. 259–260.

[7] Lewin, J. and Breakwell, J. V., "The surveillance-evasion game of degree," *JOTA*, Vol. 16, No. 3–4, 1975, pp. 339–353.

[8] Lewin, J. and Olsder, G.-J., “Conic surveillance evasion,” *JOTA*, Vol. 27, No. 1, 1979, pp. 107–125.

[9] Cardaliaguet, P., Quincampoix, M., and Saint-Pierre, P., “Numerical methods for optimal control and differential games,” Ceremade CNRS URA 749, University of Paris — Dauphine, 1995.

[10] Cardaliaguet, P., Quincampoix, M., and Saint-Pierre, P., “Set-valued numerical analysis for optimal control and differential games,” in Bardi, M., Raghavan, T. E. S., and Parthasarathy, T., eds., “Stochastic and Differential Games: Theory and Numerical Methods,” Birkhäuser, Boston, Vol. 4 of *Annals of the Int. Soc. of Dynamic Games*, pp. 177–247, 1999.

[11] Reeds, J. A. and Shepp, L. A., “Optimal paths for a car that goes both forwards and backwards,” *Pacific J. Math.*, Vol. 145, No. 2, 1990, pp. 367–393.

[12] Cockayne, E. J. and Hall, G. W. C., “Plane Motion of a Particle Subject to Curvature Constraints,” *SIAM Journal on Control*, Vol. 13, 1975, pp. 197–220.

[13] Berdyshev, Y. I., “Nelinejnye zadachi posledovatel’nogo upravleniya i ikh prolozhenie (Nonlinear Problems in Sequential Control and Their Application),” Ekaterinburg, Russia, 2015. (in Russian).

[14] Patsko, V. S., Pyatko, S. G., and Fedotov, A. A., “Three-Dimensional Reachability Set for a Nonlinear Control System,” *Journal of Computer and Systems Sciences International*, Vol. 42, No. 3, 2003, pp. 320–328.

[15] Fedotov, A. A., Patsko, V. S., and Turova, V. L., “Reachable Sets for Simple Models of Car Motion,” in Topalov, A. V., ed., “Recent Advances in Mobile Robotics,” InTech, Rijeka, pp. 147–172, 2011.

[16] Fedotov, A. A. and Patsko, V. S., “Investigation of Reachable Set at Instant for the Dubins’ Car,” in “Proceedings of the 58th Israel Annual Conference on Aerospace Sciences,” Technion, Israel, pp. 1655–1669, 2018.

[17] Patsko, V. S. and Fedotov, A. A., “Reachable Set for Dubins’ Car in the Case of One-Sided Turn,” *Trudy Instituta matematiki i mekhaniki UrO RAN (Proceedings of Institute of Mathematics and Mechanics UrB RAS)*, Vol. 24, No. 1, 2018, pp. 143–155. (in Russian).

[18] Patsko, V. S. and Fedotov, A. A., “Reachable Set for Dubins Car and Its Application to Observation Problem with Incomplete Information,” in “Preprints of the 27th Mediterranean Conference on Control and Automation (MED’19), Akko, Israel, July 1-4,” IEEE, pp. 483–488, 2019.

[19] Laumond, J.-P., ed., *Robot motion planning and control*, Vol. 229 of *Lect. Notes in Contr. and Inform. Sci.*, Springer, New York, 1998.

[20] Chitsaz, H. and LaValle, S. M., “Time-optimal Paths for a Dubins airplane,” in “Proceedings of the 46th IEEE Conference on Decision and Control New Orleans, LA, USA, Dec. 12-14, 2007,” , 2007, pp. 2379–2384.

[21] Meuleau, N., Plauntand, C., Smith, D. E., and Smith, T., "An Emergency Landing Planner for Damaged Aircraft," in "Proceedings of the 21st Innovative Applications of Artificial Intelligence Conference," AAAI Press, Pasadena, California, 2009, pp. 114–121.

[22] Choi, H., *Time Optimal Paths for a Dubins Car and Dubins Airplane with a Unidirectional Turning Constraint*, Ph.D. thesis, University of Michigan, 2014.

[23] Owen, M. and Beard, R. W., "Implementing Dubins Airplane Paths on Fixed-Wing UAVs," in Valavanis, K. P. and Vachtsevanos, G. J., eds., "Handbook of Unmanned Aerial Vehicles," Springer, New York, pp. 1677–1701, 2014.

[24] Parlangeli, G. and Indiveri, G., "Dubins inspired 2D smooth paths with bounded curvature and curvature derivative," in Indiveri, G. and Pascoal, A. M., eds., "Proceedings of the 7th IFAC Symposium on Intelligent Autonomous Vehicles," Elsevier, Vol. 43 of *IFAC Proceedings Volumes (IFAC PapersOnline)*, pp. 252–257, 2010.

[25] Hvamb, K., *Motion Planning Algorithms for Marine Vehicles*, Master's thesis, Norwegian University of Science and Technology, 2015.

[26] Lekkas, A. M., Roald, A. L., and Breivik, M., "Online Path Planning for Surface Vehicles Exposed to Unknown Ocean Currents Using Pseudospectral Optimal Control," in Hassani, V., ed., "Proceedings of the 10th IFAC Conference on Control Applications in Marine Systems," Elsevier, Vol. 43 of *IFAC Proceedings Volumes (IFAC PapersOnline)*, pp. 1–7, 2016.

[27] Subbotin, A. I., *Generalized Solutions of First Order PDEs: the Dynamical Optimization Perspective*, Birkhäuser, Boston, 1995.

[28] Crandall, M. G., Evans, L. C., and Lions, P.-L., "Viscosity solutions of Hamilton – Jacobi equations," *Transactions of the American Mathematical Society*, Vol. 277, No. 1, 1983, pp. 1–42.

[29] Crandall, M. G., Evans, L. C., and Lions, P.-L., "Some properties of viscosity solutions of Hamilton – Jacobi equations," *Transactions of the American Mathematical Society*, Vol. 282, No. 2, 1984, pp. 487–502.

[30] Bardi, M. and Capuzzo-Dolcetta, I., *Optimal Control and Viscosity Solutions of Hamilton – Jacobi – Bellman Equations*, Birkhäuser, Boston, 1997.

[31] Bardi, M., Falcone, M., and Soravia, P., "Numerical Methods for Pursuit-Evasion Games via Viscosity Solutions," in Bardi, M., Parthasarathy, T., and Raghavan, T. E. S., eds., "Annals of the International Society of Dynamic Games, Vol. 6: Stochastic and Differential Games," Birkhäuser, Boston, pp. 105–175, 1999.

[32] Munts, N. V. and Kumkov, S. S., "Convergence Of Numerical Method For Time-Optimal Differential Games With Lifeline," in Renault, J. and Ramsey, D., eds., "Annals of the Int. Soc. of Dynamic Games," Birkhäuser, Boston, 2020. (to be published).

[33] Krasovskii, N. N. and Subbotin, A. I., *Positional Differential Games*, Nauka, Moscow, 1974. (in Russian).

[34] Krasovskii, N. N. and Subbotin, A. I., *Game-Theoretical Control Problems*, Springer-Verlag, New York, 1988.

[35] Munts, N. V. and Kumkov, S. S., "Existence of value function in time-optimal game with life line," in "Proceedings of the 47th International Youth School-conference "Modern Problems in Mathematics and its Applications"," Yekaterinburg, Russia, 2016, pp. 94–99. (in Russian).

[36] Crandall, M. G., Evans, L. C., and Lions, P.-L., "Some properties of viscosity solutions of Hamilton – Jacobi equations," *Transactions of the American Mathematical Society*, Vol. 282, No. 2, 1984, pp. 487–502.

[37] Munts, N. V. and Kumkov, S. S., "On coincidence of minimax solution and value function of time-optimal problem with lifeline," *Trudy Instituta Matematiki i Mekhaniki UrO RAN (Proceedings of Institute of Mathematics and Mechanics UrB RAS)*, Vol. 24, No. 2, 2018, pp. 200–214. (in Russian).

[38] Munts, N. V. and Kumkov, S. S., "On Time-Optimal Problems with Lifeline," *Dynamic Games and Applications*, Vol. 9, No. 3, 2019, pp. 751–770.

[39] Munts, N. V. and Kumkov, S. S., "Numerical method for time-optimal differential games with life line," in "Proceedings of the 55th Israel Annual Conference on Aerospace Sciences," Technion, Israel, 2015, pp. 1253–1266.

[40] Munts, N. V. and Kumkov, S. S., "Numerical method for solving time-optimal differential games with life line," *Matematicheskaya teoriaya igr i ee prilozheniya (Mathematical Game Theory and Its Applications)*, Vol. 10, No. 3, 2019, pp. 48–75. (in Russian).