ISSN 0005-1179 (print), ISSN 1608-3032 (online), Automation and Remote Control, 2024, Vol. 85, No. 2, pp. 144–159.

On the Use of Ellipsoidal Estimation Techniques in the RRT* Suboptimal Pathfinding Algorithm

P. A. Tochilin[*,**,a] **and M. V. Parshikov**[*,b]

[*]*Moscow State University, Moscow, Russia*
[**]*Trapeznikov Institute of Control Sciences, Russian Academy of Sciences, Moscow, Russia*
e-mail: [a]*tochilin@cs.msu.ru,* [b]*miron232734@gmail.com*

Received February 28, 2023
Revised October 30, 2023
Accepted December 21, 2023

Abstract—The article is devoted to the development of an algorithm for the approximate solution of the time-optimal control problem for a system of ordinary differential equations, under the condition of avoiding collisions with stationary obstacles and subject to the specified pointwise constraints on the possible values of the control parameters. The main idea is to use a modification of the algorithm for finding suboptimal paths using rapidly growing random trees (RRT*). The most difficult part of this algorithm is to find the optimal trajectories for the problems of transferring the system from one fixed position to another, close to it, without taking into account state constraints. This subproblem is proposed to be solved using the methods of ellipsoidal calculus. This approach makes it possible to efficiently search suboptimal trajectories both for linear systems with large state space dimension and for systems with nonlinear dynamics. Algorithms for the linear and non-linear cases are sequentially analyzed in the paper, and the corresponding examples of calculations are presented.

Keywords: control systems, time-optimal problem, motion planning, ellipsoidal estimation

DOI: 10.31857/S0005117924020042

1. INTRODUCTION

This paper considers the problem of transferring a controlled object from the initial position to the given target set in the shortest possible time avoiding obstacles. Such type of problems arise when controlling the autonomous (unmanned) motion on a plane or in space, when developing algorithms for controlling robots or manipulators. As a rule, the exact solution of such a problem is very hard in terms of computation due to the large dimension of the state variables vector, nonlinear dynamics and complex structure of obstacles. Therefore, in recent decades, various approximate methods [1, 2] have been actively developed, making it possible to construct suboptimal trajectories in a short time (ideally, in a real time, which is necessary in many applications).

One of the classes of the mentioned methods is based on the use of “fast-growing” random graphs or, in a particular case, trees. Such algorithms are generally referred to as PRM — Probabilistic Roadmap (in the case of graphs with possible cycles) and RRT — Rapidly Exploring Random Tree (in the case of trees). The main idea here is to construct a sequence of random points in the state space and connect them into a graph. The number of such points N is assumed to be very large. The initial position is contained among them. The edges of the desired graph must be constructed in accordance with requirement of avoiding obstacles. Quite a lot of experience has been accumulated in constructing such graphs with various properties currently. As a rule, these methods differ in the heuristics used when randomly adding each new vertex to the graph, as well as when connecting a new vertex with previously added ones using edges. So, in [3] various modifications of random graphs and random trees methods are considered. In particular, the completeness property is analyzed, which means the fact that the probability of existence of a certain path on the graph getting from the starting point to the target set tends to 1 as the number of vertices N tends to infinity.

The next step is to add a criterion for the paths quality (for example, the length of the path or the time of movement along it) to the algorithms for constructing random graphs. The corresponding families of algorithms are denoted by PRM* and RRT*. In [3] for certain varieties of the RRT* method the property of asymptotic optimality is proven: if there is an optimal trajectory, then the probability that the quality functional value for the constructed suboptimal path on the graph tends to its minimum value at $N \to \infty$ is equal to 1. Thus, if the problem of transferring an object from the initial position to the target set is solvable, then the RRT* method will give its suboptimal solution, arbitrarily close to the optimal one, if the number N is big enough.

In [3] only the cases of the simplest motions along broken lines in state space are considered, where the constraints associated with the motion along the trajectories of specific differential equations are not taken into account. Numerous later works have attempted to take this kind of limitation into account. The resulting methods are usually called Kinodynamic RRT* (hereinafter we will use the notation KRRT*). The key subproblem here is the problem of constructing an optimal (or at least a suboptimal) trajectory of the differential equation for transferring an object from point to point in a short period of time (locally). The quality of the resulting suboptimal trajectories over a long period of time (globally) significantly depends on the method for solving this subproblem.

One of the possible approaches to solving this subproblem of optimal control over a short period of time is to use the Pontryagin's maximum principle for the problem with state constraints [4, 5]. By using a modification of the RRT* method, in [6] an attempt was made to construct trajectories that are solutions to a system of linear differential equations. Parts of such trajectories are obtained as a result of solving auxiliary linear-quadratic optimal control subproblems with the functional of a special, simplified form. A serious disadvantage of this approach is the impossibility to take into account pointwise constraints on the control parameters, as well as to correctly generalize the proposed approach to the case of nonlinear dynamics.

In [7] the modification of the RRT* method is proposed, in which parts of the desired path are composed of the "primitive" trajectories of the considered nonlinear differential equations system. Wherein the most complex part of the algorithm, which consists of constructing optimal trajectories under pointwise constraints on the controls, is considered only for three specific examples.

By using the RRT* method, [8] proposes the method for constructing suboptimal trajectories by using enumeration over a finite set of constant controls or through a random selection of such controls. The proposed approach can be effective only in special cases, with small dimensions of the control parameters vector.

Numerous works are also devoted to speeding up the operation of RRT* by changing the algorithms of adding new vertices to graphs. For example, in [9] the approach is proposed related to the construction of reachability sets estimates for the considered differential equations in small neighborhoods of the random graph vertices. These estimates can be used to cut off new vertices that will obviously be redundant, i.e. won't allow the random graph to advance significantly towards the target set. Unfortunately, the authors of the work did not propose a specific general method for constructing such estimates, only the simplest examples were considered. Wherein the problem of control synthesis on the graph edges, as in the above mentioned work [8], was solved in a primitive way, by enumerating a finite set of control parameters.

Thus, one could argue that the development of the RRT* algorithm modifications taking into account both nonlinear dynamics and the pointwise constraints on the controls, is a relevant and unsolved problem. This paper proposes one possible approach to solving this problem for a wide class of systems. For this purpose, the RRT* method has been modified with the additional use of ellipsoidal estimation algorithms [10, 11], which make it possible to effectively solve local subproblems of transferring a system from point to point. The solution to such a subproblem includes: 1) constructing a set of internal ellipsoidal estimates for the solvability set of the considered control system; 2) calculation the feedback control that solves the subproblem and obtained by “aiming” at the ellipsoidal estimates; 3) obtaining the desired open-loop control from the previously constructed strategy.

2. THE CONTROL PROBLEM FOR A LINEAR SYSTEM

In the space $\mathbb{R}^n$, $n \geqslant 2$, consider some compact set Ω. Let's consider the motion of a controlled object described by a system of linear differential equations

$$\dot{x} = Ax + Bu + f, \quad x \in \Omega, \quad t \in [0, +\infty). \tag{1}$$

Let us assume that the initial state $x(0) = x_0 \in \Omega$ is fixed. Matrices $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, as well as the vector $f \in \mathbb{R}^n$ are assumed to be independent of t. The system contains control parameters $u \in \mathbb{R}^m$, the admissible values of which are subject to pointwise constraints: $u \in \mathcal{P} = \mathcal{E}(p, P)$. Here $\mathcal{E}(p, P)$ is an ellipsoid with center $p \in \mathbb{R}^m$ and configuration matrix $P \in \mathbb{R}^{m\times m}$, $P = P^T \geqslant 0$:

$$\mathcal{E}(p, P) = \left\{ y \in \mathbb{R}^m : \ \langle y, l\rangle \leqslant \langle l, p\rangle + \sqrt{\langle l, Pl\rangle}, \ \forall l \in \mathbb{R}^m \right\}.$$

If $P > 0$, then the equivalent definition can be used:

$$\mathcal{E}(p, P) = \left\{ y \in \mathbb{R}^m : \ \Big\langle (y - p, P^{-1}(y - p)) \Big\rangle \leqslant 1 \right\}.$$

Let $\mathcal{U}$ denote the class *of admissible open-loop controls*, which contains all possible piecewise continuous functions $u = u(t) \in \mathcal{P}$, $t \geqslant 0$. Let $x(t, 0, x_0)|_{u(\cdot)}$ denote the trajectory of (1) constructed using some admissible control $u(\cdot)$, released at the initial time $t_0 = 0$ from the starting position x_0. We will further consider only parts of the trajectories $x(t, 0, x_0)|_{u(\cdot)}$ contained in the Ω region.

Also in the Ω region we select a certain range of sets $\mathcal{M}_i$, $i = 1, \ldots, M$, – obstacles that must be taken into account when an object moves. Let's assume that each of these sets is given by a set of inequalities

$$\mathcal{M}_i = \{x \in \Omega : \ \varphi_{i,j}(x) \leqslant 0, \ j = 1, \ldots, s_i\}, \tag{2}$$

where continuous functions $\varphi_{i,j}(x)$, as well as natural numbers s_i are given. It is assumed that the positions of obstacles $\mathcal{M}_i$ do not change with time, i.e. the functions $\varphi_{i,j}$ do not depend on t. Let's introduce the following notation:

$$\mathcal{X}_{free} = \overline{\Omega \setminus \left(\bigcup_{i=1}^{M} \mathcal{M}_i \right)}.$$

Suppose that $x_0 \in \mathcal{X}_{free}$. Let's also fix some compact set $\mathcal{X}_{goal} \subset \mathcal{X}_{free}$, with $x_0 \notin \mathcal{X}_{goal}$ and $\mu(\mathcal{X}_{goal}) > 0$, where $\mu(\mathcal{X}_{goal})$ is the Lebesgue measure of the set $\mathcal{X}_{free}$.

Now we can formulate **the main control problems** solved in this work:

1) It is necessary to find such $u(\cdot) \in \mathcal{U}$ for which there exists $t_1 > 0$ and such a trajectory $x(t, 0, x_0)|_{u(\cdot)}$ of differential equation (1), for which

$$x(t_1) \in \mathcal{X}_{goal}, \ x(\tau) \in \mathcal{X}_{free}, \ \forall \tau \in [0, t_1].$$

2) Among the controls and corresponding trajectories found in the previous case, it is necessary to find those for which the value t_1 will be the smallest.

Note that the set $\mathcal{X}_{free}$ is closed, which means that movement along the boundaries of obstacles is allowed.

3. THE ALGORITHM FOR CONSTRUCTING A RANDOM TREE

Let's solve the control problems described above approximately using a modification of the KRRT* algorithm. Below its general scheme is presented. The main goal of the algorithm is to construct an oriented tree $\Gamma = (V, E)$, where V is the set of vertices, E is the set of arcs. Each vertex $v \in V$ will be associated with the minimum found time of its reaching $C(v)$ from the initial point x_0, which is the root of the tree. For each vertex $v \neq x_0$ let denote $parent(v)$ as the vertex for which $(parent(v), v) \in E$.

The algorithm is iterative, where i is the iteration number, and a fixed number N is the maximum number of iterations. At each step of the algorithm, auxiliary ellipsoidal estimates of the solvability set $\mathcal{E}(w(t), W_k(t))$, $k = 1, \ldots, K$ must be constructed. The number of evaluations K is considered fixed. Each estimate depends on the parameter vector $l_k \in \mathbb{R}^n$ and is defined at $t \in [T, 0]$, where the number $T < 0$ is fixed. The choice of the parameter $T < 0$ here is due to the stationarity of the system (1) and the convenience of fixing the zero final time instant. For the resulting control and trajectory, a change of variable will then be made, and as a result, these functions will be defined already for $t \geqslant 0$. Formulas for calculating ellipsoidal estimates will be given below. The following algorithm uses two auxiliary sets of graph vertices $\mathcal{V}_0$ and $\mathcal{V}_1$.

Algorithm 1 Ellipsoidal KRRT*

1: $V := \{x_0\}$, $C(x_0) := 0$, $E := \emptyset$
2: form a set of vectors $l_k,\ l_k \in \mathbb{R}^n,\ \|l_k\| = 1,\ k = 1, \ldots, K$
3: **for** $i = 1, \ldots, N$ **do**
4: generate a random point $x^{(i)} \in \mathcal{X}_{free}$
5: construct ellipsoidal estimates $\mathcal{E}(w(t), W_k(t))$, $k = 1, \ldots, K$, $t \in [T, 0]$
6: form a set of vertices $\mathcal{V}_0 = V \cap \big(\cup_k \cup_t \mathcal{E}(w(t), W_k(t)) \cap \mathbb{B}_{r_i}(x^{(i)})\big)$
7: $\mathcal{V}_1 := \emptyset$
8: **for** $v_j \in \mathcal{V}_0$ **do**
9: $t_{\min,j} := \max_k \max\{t \in [T, 0] :\ v_j \in \mathcal{E}(w(t), W_k(t))\}$
10: $k^* :=$ the corresponding ellipsoidal estimate number (the maximizer)
11: determine the optimal control function $u^*(j, k^*, t)$, $t \in [t_{\min,j}, 0]$
12: build the trajectory $x^*(j, k^*, t)$, $t \in [t_{\min,j}, 0]$, $x^*(j, k^*, t_{\min,j}) = v_j$
13: **if** $x^*(j, k^*, t) \in \mathcal{X}_{free}$, $\forall t \in [t_{\min,j}, 0]$ **then**
14: $\mathcal{V}_1 := \mathcal{V}_1 \cup \{v_j\}$
15: **if** $\mathcal{V}_1 \neq \emptyset$ **then**
16: $j^* := \arg\min\{C(v_j) - t_{\min,j} :\ v_j \in \mathcal{V}_1\}$
17: $V := V \cup \{x^{(i)}\}$, $E := E \cup \{(v_{j^*}, x^{(i)})\}$
18: $C(x^{(i)}) := C(v_{j^*}) - t_{\min,j^*}$
19: **for** $v_j \in \mathcal{V}_1$ **do**
20: **if** $\Delta_j = C(x^{(i)}) - t_{\min,j} - C(v_j) < 0$ **then**
21: for each vertex v' of the subtree with the root v_j $C(v') := C(v') + \Delta_j$
22: $E := E \setminus \{(parent(v_j), v_j)\}$
23: $E := E \cup \{(x^{(i)}, v_j)\}$

In the line 4, the coordinates of each new point $x^{(i)}$ are random, corresponding to a uniform distribution on the set Ω. In the line 6 of the algorithm, vertices v_j are selected according to the following condition:

$$v_j \in \left(\bigcup_{k=1}^{K} \bigcup_{t \in [T,0]} \mathcal{E}(w(t), W_k(t)) \right) \cap \mathbb{B}_{r_i}(x^{(i)}). \tag{3}$$

Here $r_i > 0$ is a parameter that limits the enumeration of graph vertices and is responsible for reducing the final complexity of calculations. It will be discussed in more details below.

When constructing the control and trajectory in the lines 11–12, the algorithm of "aiming" at a specific ellipsoidal tube is used, which will be described below. Because of this fact the final functions $x^*(\cdot)$ and $u^*(\cdot)$ depend on the number of the ellipsoidal estimate used. Function $x^*(j, k^*, t)$, $t \in [t_{\min,j}, 0]$, is a solution to the Cauchy problem for the system (1), after substituting the control $u^*(j, k^*, t)$, with the boundary condition $x^*(j, k^*, t_{\min,j}) = v_j$.

The instructions of the main algorithm in lines 19–23 correspond to the well-known (see [3]) rule from the description of the RRT* algorithm of "rewiring" the tree vertices. Thanks to these actions, the original RRT* algorithm achieves the property of asymptotic optimality.

An additional condition for exiting the outer loop can be added to the above algorithm when the target set is first reached, i.e. if at the iteration with number i the condition $x^{(i)} \in \mathcal{X}_{goal}$ is satisfied. This condition is necessary when solving the first of the two main control problems described above.

Remark 1. The condition $\mu(\mathcal{X}_{goal}) > 0$ is essential for the proposed algorithm. In the case when the target set is a single point ($\mathcal{X}_{goal} = \{x_{goal}\}$), the algorithm can be modified by adding instructions for constructing ellipsoidal estimates for the solvability set released from the target point, and for subsequent actions, similar to the lines 6–18, by replacing $x^{(i)}$ with x_{goal}.

For simplicity of presentation, in the above algorithm's scheme additional information associated with the edges of the graph is not indicated. However, to solve the main control problems, it will be necessary to additionally remember for each edge the open-loop control and the corresponding part of the trajectory, which were calculated in lines 11–12 of the algorithm for the corresponding values of the indices i, j. Moreover, in the graph rewiring procedure it is necessary to make an additional replacement of the t variable in order to go through the previously constructed trajectory in the opposite direction. In this case, it is important that the original system (1) is stationary.

After constructing the Γ tree, the main control problems formulated above can be solved:

— If $\exists v_j \in V$, $v_j \in \mathcal{X}_{goal}$, then the problem 1) is solvable.

— If the condition from the previous paragraph is satisfied, then the performance time estimate can be found as follows:

$$t_1^* = \min\{C(v_j) :\ v_j \in V,\ v_j \in \mathcal{X}_{goal}\}. \tag{4}$$

Let v^* be the minimizer in (4).

— The optimal control $u^*(t)$ can now be composed of parts associated with separate graph edges. In this case, it is needed to move from the vertex v^* to the root of the tree, along the arcs

$$(parent(v^*), v^*),\ (parent(parent(v^*)), parent(v^*)),\ \ldots,\ (x_0, \ldots),$$

taking into account the reverse progression of time when combining parts of the function $u^*(t)$.

4. THE ELLIPSOIDAL ESTIMATES AND THE CONTROL LAW

The main difference between the algorithm used in this work and the known versions of the KRRT* algorithm is the use of internal estimates for *solvability sets* [12] of the system (1), released from a randomly selected point $x^* \in \mathcal{X}_{free}$ (in the algorithm's scheme such points are denoted as $x^{(i)}$).

For a fixed set $\mathcal{X}_1 \subset \Omega$, for some $t \leqslant t_1$ the solvability set $\mathcal{W}(t, t_1, \mathcal{X}_1)$ of the system (1) consists of all possible points $x_0 \in \Omega$, for each of which there exists a control function $u(\cdot) \in \mathcal{U}$ for which $x(t_1, t, x_0)|_{u(\cdot)} \in \mathcal{X}_1$. In this work, the solvability sets from the one-point target set with zero finite time are of interest: $\mathcal{W}(t, 0, x^*)$, $x^* \in \mathbb{R}^n$. The task of constructing such sets even for a linear stationary system is nontrivial. One of the most effective approaches is the use of ellipsoidal estimates for solvability sets: external or internal, depending on the specific problem being solved. Here the main goal is to find the control function, and for that purpose it is convenient to use internal estimates of the solvability sets.

For a given $x^* \in \mathbb{R}^n$, using the results from [10], we define a family of ellipsoidal estimates $\mathcal{E}(w(t), W(t))$, $t \leqslant 0$, defined by the following differential equations:

$$\begin{cases} \dot{w}(t) = Aw + Bp + f \\ w(0) = x^*, \end{cases} \tag{5}$$

$$\begin{cases} \dot{W}(t) = AW(t) + W(t)A^T - W^{1/2}(t)S(t)\mathbf{P}^{1/2} - \mathbf{P}^{1/2}S^T(t)W^{1/2}(t) \\ W(0) = \mathbb{O}_{n\times n}. \end{cases} \tag{6}$$

Here $\mathbb{O}_{n\times n}$ is the zero matrix of size $n \times n$, $\mathbf{P} = BPB^T$, and $S(t)$ is an orthogonal matrix, continuously depending on t, for which

$$S(t)\mathbf{P}^{1/2}l(t) = \lambda(t)\mathbf{P}^{1/2}l(0),\ S(0) = \mathbb{I}_{n\times n},\quad \lambda(t) = \sqrt{\frac{\langle l(t), \mathbf{P}l(t)\rangle}{\langle l(0), \mathbf{P}l(0)\rangle}}. \tag{7}$$

Here $\mathbb{I}_{n\times n}$ is the identity matrix of size $n \times n$. The choice of parameters $S(t)$ and $\lambda(t)$ is caused by the requirement that the ellipsoidal estimate touches the solvability set in the direction specified by the vector $l(t)$.

Each ellipsoidal estimate depends on some curve $l(t)$, which is the solution to the following auxiliary Cauchy problem:

$$\begin{cases} \dot{l} = -A^T l \\ l(0) = l^* \end{cases} \tag{8}$$

for an arbitrary vector $l^* \in \mathbb{R}^n$: $\|l^*\| = 1$. In order to emphasize such dependence, we will use the notation $\mathcal{E}(w(t), W(t, l^*))$. By searching through various values of l^*, different estimates can be obtained (in the main algorithm, such vectors $l^* = l_k$ for constructing different estimates are formed in the line 2; wherein, $W_k(t) = W(t, l_k)$). We will further assume that the vectors l^* are chosen in such a way that $\langle l^*, \mathbf{P}l^*\rangle \neq 0$, which is necessary for the correctness of the formulas (7). The following statement is true (see [10]):

Theorem 1. *For any* $t \leqslant 0$

$$\mathcal{W}(t, 0, x^*) = \bigcup \left\{\mathcal{E}(w(t), W(t, l^*)) :\ l^* \in \mathbb{R}^n,\ \|l^*\| = 1\right\}.$$

In the line 6 of the main algorithm it is necessary to select those vertices of the graph that satisfy the condition

$$v_j \in \bigcup_{k=1,\ldots,K} \bigcup_{t\in[T,0]} \mathcal{E}(w(t), W_k(t)).$$

Using the definition of an ellipsoid, this condition can be rewritten as an auxiliary optimization problem:

$$\min_{k=1,\ldots,K} \min_{t\in[T,0]} \max \left\{\langle v_j - w(t), l\rangle - \sqrt{\langle l, W_k(t)l\rangle} :\ l \in \mathbb{R}^n,\ \|l\| = 1\right\} \leqslant 0.$$

Now we can construct the *feedback control* that transfers the system's trajectory (1) from the position (t, x), $t < 0$, to the position $(0, x^*)$. It can be found by "aiming" [12] at one of the constructed ellipsoids $\mathcal{E}(w(t), W(t))$:

$$u^*(t,x) = \begin{cases} p - \dfrac{PB^T l^0(t)}{\sqrt{\langle l^0(t), \mathbf{P} l^0(t)\rangle}}, & B^T l^0(t) \neq 0 \\ \mathcal{E}(p, P) & \text{otherwise,} \end{cases} \tag{9}$$

where $l^0(t)$ is the maximizer in the half-distance expression

$$h_+(t,x) = \max\left\{ \left\langle l, e^{-tA}(x - w(t))\right\rangle - \sqrt{\left\langle l, e^{-tA} W(t) e^{-tA^T} l\right\rangle} \,\Big|\, \|l\| \leqslant 1 \right\}.$$

For $x \notin \operatorname{int} \mathcal{E}(w(t), W(t))$ the vector $l^0(t)$ can be found in the same way as

$$l^0(t) = 2\lambda(W(t) + \lambda E)^{-1}(x - w(t)),\ E = e^{A^T t} e^{At},$$

where λ is the only non-negative root of the equation

$$\left\langle (W(t) + \lambda E)^{-1}(x - w(t)), W(t)(W(t) + \lambda E)^{-1}(x - w(t))\right\rangle = 1.$$

If $x \in \operatorname{int} \mathcal{E}(w(t), W(t))$, then $l^0(t) = 0$.

Note that in the main algorithm (lines 9–11) if the next point $x^{(i)}$ does not coincide with any of the previously constructed graph vertices v_j (the probability of this event is equal to 1), then the value $t_{\min,j} < 0$ for each corresponding value j should be chosen such that $v_j \in \partial\mathcal{E}(w(t_{\min,j}), W_{k^*}(t_{\min,j}))$. I.e. at the initial time instant the point lies on the boundary of the ellipsoid. The used method of extreme aiming at an ellipsoid has the property (see [12] for more details) that the trajectory of the system, starting at the boundary of the ellipsoid, will remain on it until the final moment of time. Thus, the case when the point x lies on the boundary of the ellipsoid $\mathcal{E}(w(t), W(t))$ for any $t \leqslant 0$ is especially important in this work. For that case

$$x = w(t) + \frac{W(t)s}{\sqrt{\langle s, W(t)s\rangle}}, \quad s = W^{-1}(t)(x - w(t))\|W^{-1}(t)(x - w(t))\|^{-1}.$$

The corresponding value of the vector $l^0(t) = e^{tA^T} s$ can be found without solving the above mentioned auxiliary optimization problem or equation for λ.

Let's summarize the properties of the constructed feedback control in the following statement

Theorem 2. *Let for some values j, k there is a vertex $v_j \in \mathcal{E}(w(t_{\min,j}), W_k(t_{\min,j}))$. Consider the Cauchy problem for the differential inclusion $\dot{x} \in Ax + Bu^*(t,x) + f$, $x(t_{\min,j}) = v_j$, $t \in [t_{\min,j}, 0]$. This problem has solutions, and for any such solution $x(t) = x(t, t_{\min,j}, v_j)|_{u^*(\cdot)}$ the condition $x(0) = x^*$ is satisfied.*

In the main algorithm, for each new random point $x^* = x^{(i)}$, it is necessary to construct internal ellipsoidal estimates for solvability sets and, using them, determine those previously constructed vertices of the graph Γ that can be connected to the new vertex (line 6 of the algorithm). Let ellipsoidal estimates $\mathcal{E}(w(t), W_k(t))$, $k = 1, \ldots, K$, $t \in [T, 0]$ be constructed for the target point x^*. At this iteration of the algorithm, only those vertices $v_j \in V$ should be processed for which the condition (3) is satisfied. This formula uses an auxiliary parameter (see [7])

$$r_i = \min\left\{\gamma\left(\frac{\ln(\kappa)}{\kappa}\right)^{1/n}, \eta\right\}, \quad \gamma > (2(1 + 1/n))^{1/n}\left(\frac{\mu(\mathcal{X}_{free})}{\zeta_n}\right)^{1/n}, \tag{10}$$

where ζ_n is the volume of the unit sphere in $\mathbb{R}^n$, $\mu(\mathcal{X}_{free})$ is the Lebesgue measure of the set $\mathcal{X}_{free}$, $\kappa = |V|$ – the total number of previously added (before the ith iteration) graph vertices. A constant $\eta > 0$ can be taken arbitrary; this value is responsible for reducing the complexity of calculations on the first steps of the algorithm (for small κ). If we additionally assume that the following condition is satisfied

$$\exists \delta > 0: \quad \mathbb{B}_\delta(x^*) \subseteq \bigcup_{t\in[T,0]} \bigcup_{k=1}^{K} \mathcal{E}(w(t), W_k(t)),$$

then, according to [3], with the above mentioned choice of r_i, the presented algorithm has the property of asymptotic optimality.

For each vertex v_j selected according to (3), the absolute minimum time value $t_{\min,j}$ of transition from v_j to point $x^{(i)}$ can be determined, as well as the number of the corresponding ellipsoidal estimate k^*: $v_j \in \mathcal{E}(w(t_{\min,j}), W_{k^*}(t_{\min,j}))$. Further it is necessary to define the control function. The formula (9) cannot be used directly in the main algorithm, since we need open-loop control (line 11 of the algorithm), instead of the feedback strategy. The transition from the closed-loop control to the open-loop one can always be accomplished by solving the auxiliary Cauchy problem

$$\begin{cases} \dot{x}(t) \in Ax(t) + Bu^*(t,x) + f \\ x(t_{\min,j}) = v_j, \end{cases} \qquad u^*(t,x) = \begin{cases} p - \dfrac{PB^T l^0(t)}{\sqrt{\langle l^0(t), \mathbf{P} l^0(t)\rangle}}\,, & B^T l^0(t) \neq 0 \\ \mathcal{E}(p,P) & \text{otherwise,} \end{cases} \tag{11}$$

and subsequent selection of a single-valued selector from the set-valued mapping $u^*(t, x^*(t))$. Here $x^*(t)$ is the solution to the problem (11), $l^0(t)$ depends on x.

For each constructed trajectory $x^*(t)$, $t \in [t_{\min,j}, 0]$, the following condition must then be checked (line 13 of the algorithm): $x^*(t) \in \mathcal{X}_{free}$, $\forall t \in [t_{\min,j}, 0]$. According to (2) it is enough to check the inequality

$$\min_{t\in[t_{\min,j},0]} \min_{i=1,\ldots,M} \max_{r=1,\ldots,s_i} \varphi_{i,r}(x(t)) \geqslant 0.$$

If this condition is not satisfied, then the trajectory will be rejected by the algorithm. Otherwise it can be used to construct an edge connecting the new vertex $x^* = x^{(i)}$ of the graph Γ with the old vertex v_j. If several suitable vertices v_j are found for different j, then the one (with the number j^*) for which the value $t_{\min,j}$ is maximum will be selected.

Remark 2. According to (6), the ellipsoids' configuration matrices $W(t)$ do not depend on the specific point x^* from which it is necessary to construct solvability sets at each iteration of the algorithm. Thus, matrices $W(t)$, $t \in [T, 0]$, for different values of l^* can be calculated in advance, before the start of calculations in the main loop of the algorithm. Similarly, from (5) it is clear that for the centers of ellipsoids the following relation is valid:

$$w(t) = \tilde{w}(t) + w^*(t), \quad \tilde{w}(t) = \int_0^t e^{A(t-\tau)}(Bp + f)d\tau, \; w^*(t) = e^{At}x^*,$$

which means that the function $\tilde{w}(t)$ can be calculated in advance. Inside the main cycle of the algorithm only the functions $w^*(t)$ need to calculated and summed up with $\tilde{w}(t)$. This makes calculations easier.

Remark 3. When implementing the Ellipsoidal KRRT* algorithm for a specific system (1), it is important to select the most appropriate value for the parameter $T < 0$. If it turns out that the absolute value of T is too small, then this will lead to the creation of an artificial limitation in the formation of new random tree branches, and therefore the finally found trajectory may be far from optimal. If the absolute value of T is too large, then this can lead to additional, unnecessary calculations of ellipsoidal estimates, which will negatively affect the algorithm's speed. To select the most appropriate value of the parameter T, a series of experiments can be carried out followed by analysis of the realized values $t_{\min,j}$.

5. EXAMPLE (LINEAR DYNAMICS)

Let's consider an example of a suboptimal trajectory constructing, and also compare the obtained results with the ones obtained with the help of the algorithm discussed in [6]. Let $x \in \mathbb{R}^2$ and the system's dynamics (1) is given by the following parameters:

$$A = \begin{pmatrix} 8 & 2 \\ 2 & 8 \end{pmatrix}, \quad B = \begin{pmatrix} 4 & 1 \\ 1 & 4 \end{pmatrix}, \quad f = \begin{pmatrix} 1 \\ 1 \end{pmatrix}.$$

The following constraints are imposed on the control parameters: $u \in \mathcal{P} = \mathcal{E}(p, P)$, where

$$p = \begin{pmatrix} 2 \\ 2 \end{pmatrix}, P = \begin{pmatrix} 16 & 4 \\ 4 & 16 \end{pmatrix}.$$

Note that these constraints are taken into account in both algorithms. In (10) we put $\eta = 4$. Let's define the auxiliary matrix $R = \mathbb{I}_{2\times2}$, necessary for calculating the functional in the algorithm from [6]. We fix $T = -2$. This choice is due to the values of $t_{\min,j}$ obtained by the Ellipsoidal KRRT* for different values of T in a series of experiments:

	Test number									
	1		2		3		4		5	
T	t_{avg}	$t_{\min}$	t_{avg}	$t_{\min}$	t_{avg}	$t_{\min}$	t_{avg}	$t_{\min}$	t_{avg}	$t_{\min}$
−0.1	−0.05	−0.1	−0.06	−0.1	−0.05	−0.1	−0.05	−0.1	−0.05	−0.1
−0.5	−0.11	−0.46	−0.13	−0.49	−0.13	−0.47	−0.12	−0.49	−0.13	−0.46
−1	−0.11	−0.46	−0.14	−0.65	−0.15	−0.81	−0.14	−0.95	−0.13	−0.71
−2	−0.11	−0.46	−0.14	−0.65	−0.15	−1.17	−0.15	−0.97	−0.14	−0.71
−5	−0.14	−0.46	−0.16	−0.66	−0.22	−1.18	−0.18	−1.01	−0.16	−0.71

Here $t_{avg} = \sum_{j=1}^{|V|} t_{\min,j}/|V|$, $t_{\min} = \min\limits_{j} t_{\min,j}$. It can be seen that the values $t_{\min} < -2$ hasn't been found in the experiments, and therefore there is no point in taking the value $T < -2$. The value -2 is taken with a small margin relative to the smallest of $t_{\min}$.

Figures 1 and 2 below demonstrate the results of the Ellipsoidal KRRT* and the KRRT* algorithms, respectively, which were obtained as follows: the KRRT* algorithm (from [6]) used as random points the same sequence of points $Samples = \{x^{(i)}\}_{i=1}^{N}$, $N = 2728$, which had been required by the Ellipsoidal KRRT* algorithm to construct the tree $\Gamma_1 = (V_1, E_1)\colon |V_1| = 500$.

Note that the KRRT* algorithm in this case built the tree $\Gamma_2 = (V_2, E_2)\colon\ |V_2| = 469$. The decrease in the number of vertices is due to the fact that some of the trial edges were rejected due to the control constraints violations or due to the state constraints. Also note that in Fig. 1 the suboptimal trajectory passes through the shaded area, which the KRRT* algorithm could not find.

For the given target set

$$X_{goal} = \{x \in \mathbb{R}^2\colon 9.5 < x_1 < 11.5,\ 10.5 < x_2 < 12.5\}$$

the following values of the functionals were obtained:

$$t_1^* = \min\{C(v):\ v \in V_1,\ v \in X_{goal}\} = 0.26, \quad t_2^* = \min\{C(v):\ v \in V_2,\ v \in X_{goal}\} = 0.47.$$

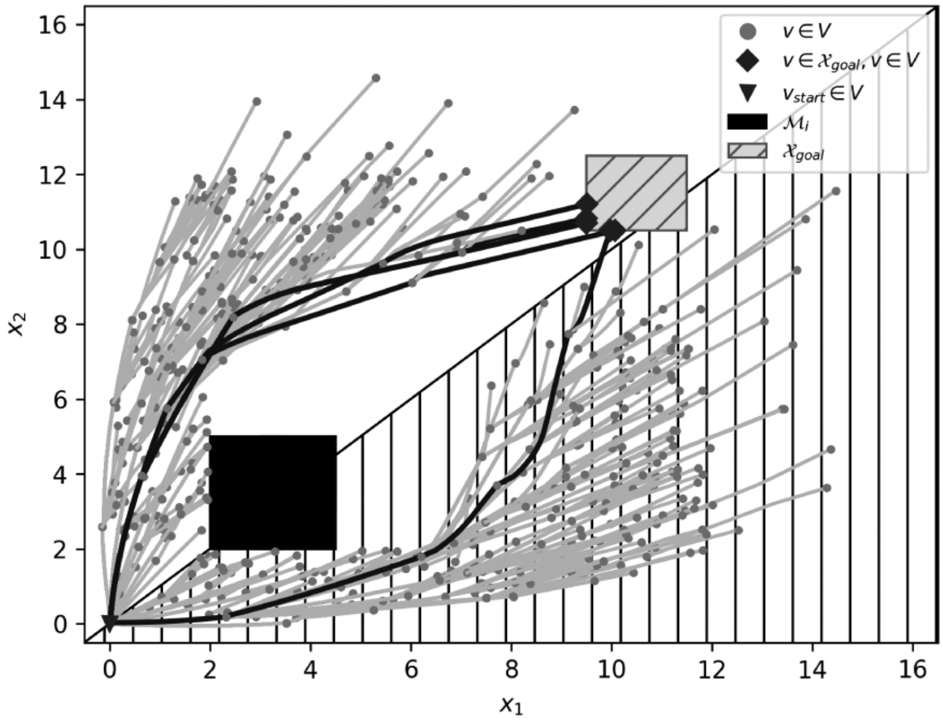

Fig. 1. Ellipsoidal KRRT*.

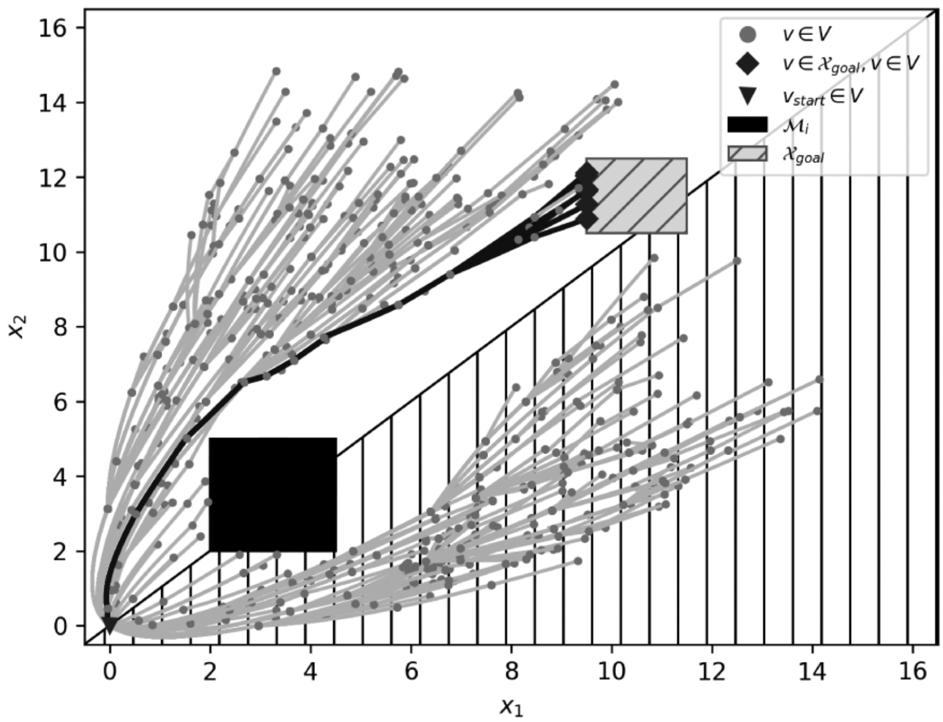

Fig. 2. KRRT*.

6. NONLINEAR DYNAMICS

Let's apply now the method described above for constructing a suboptimal trajectory of a linear system to solve similar problems in the case of nonlinear dynamics. Consider the following system of ordinary differential equations:

$$\dot{x} = f(x) + g(x)u, \ x \in \Omega, \ t \in [0, +\infty). \tag{12}$$

As before, we assume that the initial state $x(0) = x_0 \in \Omega$ is fixed, $u \in \mathcal{P} = \mathcal{E}(p, P) \subset \mathbb{R}^m$. The functions $f(x)$ and $g(x)$ are assumed to be twice continuously differentiable for $x \in \Omega$.

The Ellipsoidal KRRT* algorithm, which solves the control problems formulated above, remains similar in form. We only note the differences associated with nonlinear dynamics.

To construct estimates for solvability sets of a nonlinear system in the neighborhood of each new vertex $x^{(i)}$ of the graph Γ, we use the linearization of the equations (12). Let $f_s(x)$ – be the sth component of the vector $f(x)$, $g_s(x)$ – be the sth row of the matrix $g(x)$, $s = 1, \ldots, n$, p_r – be the rth component of vector p, $r = 1, \ldots, m$. Then

$$\begin{aligned}\dot{x}_s = f_s(x) + g_s(x)u = f_s(x^{(i)}) + \left\langle \frac{\partial f_s}{\partial x}(x^{(i)}), x - x^{(i)} \right\rangle + g_s(x^{(i)})u \\ + \sum_{r=1}^{m} \left\langle \frac{\partial g_{sr}}{\partial x}(x^{(i)}), (x - x^{(i)}) \right\rangle p_r + \frac{1}{2} \sum_{r=1}^{m} \left\langle x - x^{(i)}, Q^{g,(sr)}(\zeta_{sr})(x - x^{(i)}) \right\rangle p_r \\ + \sum_{r=1}^{m} \left\langle \frac{\partial g_{sr}}{\partial x}(\eta_{sr}), (x - x^{(i)}) \right\rangle (u_r - p_r) + \frac{1}{2} \left\langle x - x^{(i)}, Q^{f,(s)}(\xi_s)(x - x^{(i)}) \right\rangle,\end{aligned}$$

where $Q^{g,(sr)}(\zeta_{sr})$ is a matrix of second derivatives of the function $g_{sr}(x)$, calculated at some point $\zeta_{sr} \in \mathbb{B}_{r_i}(x^{(i)})$, $Q^{f,(s)}(\xi_s)$ is a matrix of second derivatives of the function $f_s(x)$, calculated at point $\xi_s \in \mathbb{B}_{r_i}(x^{(i)})$. The radius of the ball $\mathbb{B}_{r_i}(x^{(i)})$ is still determined from the formulas (10). Also some points $\eta_{sr} \in \mathbb{B}_{r_i}(x^{(i)})$ are used here. Note that the ball $\mathbb{B}_{r_i}(x^{(i)})$ (see (3)) is used here as the admissible range of the variable x variation. This allows to estimate the linearization error:

$$\begin{aligned}\Bigg| \frac{1}{2} \sum_{r=1}^{m} \left\langle x - x^{(i)}, Q^{g,(sr)}(\zeta_{sr})(x - x^{(i)}) \right\rangle p_r + \sum_{r=1}^{m} \left\langle \frac{\partial g_{sr}}{\partial x}(\eta_{sr}), (x - x^{(i)}) \right\rangle (u_r - p_r) \\ + \frac{1}{2} \left\langle x - x^{(i)}, Q^{f,(s)}(\xi_s)(x - x^{(i)}) \right\rangle \Bigg| \leqslant R_s,\end{aligned}$$

$$\begin{aligned}R_s = R_s(x^{(i)}) = \frac{r_i^2}{2} \sum_{r=1}^{m} \max \left\{ \lambda_{\max}(Q^{g,(sr)}(\zeta)) : \ \zeta \in \mathbb{B}_{r_i}(x^{(i)}) \right\} |p_r| \\ + \frac{r_i^2}{2} \max \left\{ \lambda_{\max}(Q^{f,(s)}(\xi)) : \xi \in \mathbb{B}_{r_i}(x^{(i)}) \right\} \\ + r_i \sum_{r=1}^{m} \max \left\{ \left\| \frac{\partial g_{sr}}{\partial x}(\eta) \right\| : \eta \in \mathbb{B}_{r_i}(x^{(i)}) \right\} \max \left\{ |u_r - p_r| : u \in \mathcal{P} \right\}.\end{aligned} \tag{13}$$

Here $\lambda_{\max}(Q)$ is the eigenvalue of maximum modulus of the symmetric matrix Q. As a result, the linearized system can be written in the following form:

$$\dot{x} = A(x^{(i)})x + B(x^{(i)})u + h(x^{(i)}) + v, \ x \in \mathbb{B}_{r_i}(x^{(i)}), \tag{14}$$

where

$$A(x^{(i)}) = \begin{pmatrix} \left(\dfrac{\partial f_1}{\partial x}(x^{(i)}) + \sum\limits_{r=1}^{m} \dfrac{\partial g_{1r}}{\partial x}(x^{(i)})p_r\right)^T \\ \cdots \\ \left(\dfrac{\partial f_n}{\partial x}(x^{(i)}) + \sum\limits_{r=1}^{m} \dfrac{\partial g_{nr}}{\partial x}(x^{(i)})p_r\right)^T \end{pmatrix}, \quad B(x^{(i)}) = \begin{pmatrix} g_1(x^{(i)}) \\ \cdots \\ g_n(x^{(i)}) \end{pmatrix},$$

$$h(x^{(i)}) = \begin{pmatrix} f_1(x^{(i)}) - \left\langle \dfrac{\partial f_1}{\partial x}(x^{(i)}), x^{(i)} \right\rangle - \sum\limits_{r=1}^{m} \left\langle \dfrac{\partial g_{1r}}{\partial x}(x^{(i)}), x^{(i)} \right\rangle p_r \\ \cdots \\ f_n(x^{(i)}) - \left\langle \dfrac{\partial f_n}{\partial x}(x^{(i)}), x^{(i)} \right\rangle - \sum\limits_{r=1}^{m} \left\langle \dfrac{\partial g_{nr}}{\partial x}(x^{(i)}), x^{(i)} \right\rangle p_r \end{pmatrix}.$$

We will further interpret the linearization error v as the uncertainty $v(t)$, on the possible values of which the pointwise constraints are imposed:

$$v(t) \in \mathbb{B}_{\rho(x^{(i)})}(0), \quad \rho(x^{(i)}) = \sqrt{R_1^2(x^{(i)}) + \ldots + R_n^2(x^{(i)})}. \tag{15}$$

For any control $u(\cdot) \in \mathcal{U}$ and any measurable bounded function $v(t)$ satisfying the constraint from (15) denote $x(t, 0, x_0)|_{u(\cdot),v(\cdot)}$ to be the corresponding trajectory of system (14) released from the initial position x_0 at the 0 moment of time. Note that any trajectory $x(t, 0, x_0)|_{u(\cdot)}$ of the system (12) is also a trajectory of (14) for some admissible noise $v(\cdot)$, unless $x(\tau, 0, x_0)|_{u(\cdot)} \in \mathbb{B}_{r_i}(x^{(i)})$, $\forall \tau \in [t, 0]$. The converse statement is not true.

Remark 4. Since during the execution of the Ellipsoidal KRRT* algorithm it is necessary to repeatedly linearize the system (12) in the neighborhood of each successive point $x^{(i)}$, then to speed up the calculations it makes sense to estimate the linearization error more roughly, namely in the formula (13) replace the maximum taken over $\mathbb{B}_{r_i}(x^{(i)})$ with the maximum taken over the set Ω. Then it will be enough to count the restrictions only once.

Due to the existence of the linearization errors, it is necessary to modify the formulas for calculating ellipsoidal estimates for solvability sets, that were used above for the case of linear dynamics. To do this, the results from [11] can be used. Namely, now the ellipsoids $\mathcal{E}(w(t), W(t))$, $t \leqslant 0$, are given by the following equations:

$$\begin{cases} \dot{w}(t) = A(x^{(i)})w + B(x^{(i)})p + h(x^{(i)}) \\ w(0) = x^{(i)}, \end{cases} \tag{16}$$

$$\begin{cases} \dot{W} = AW + WA^T - W^{1/2}S\mathbf{P}^{1/2} - \mathbf{P}^{1/2}S^TW^{1/2} + \pi(t)W + \dfrac{\rho^2(x^{(i)})}{\pi(t)}\mathbb{I}_{n\times n} \\ W(0) = \varepsilon_i^2\mathbb{I}_{n\times n}. \end{cases} \tag{17}$$

Here $W = W(t)$, $A = A(x^{(i)})$, $B = B(x^{(i)})$, $\mathbf{P} = BPB^T$, $S = S(t)$ is an orthogonal matrix for which

$$S\mathbf{P}^{1/2}l(t) = \lambda(t)\mathbf{P}^{1/2}l(0), \quad S(0) = \mathbb{I}_{n\times n}, \quad \lambda(t) = \sqrt{\frac{\langle l(t), \mathbf{P}l(t)\rangle}{\langle l(0), \mathbf{P}l(0)\rangle}}, \quad \pi(t) = \frac{\rho(x^{(i)})\|l(t)\|}{\sqrt{\langle l(t), W(t)l(t)\rangle}}.$$

As in the case of linear dynamics, each ellipsoidal estimate depends on some curve $l(t)$, which is the solution to the Cauchy problem (8) for an arbitrary $l^* \in \mathbb{R}^n$: $\|l^*\| = 1$. The functions $S(t)$ and $\pi(t)$ depend continuously on t.

Note that due to the uncertainty $v(t)$, the problem of transferring the system's trajectory (14) exactly to the point $x^{(i)}$ can no longer be talked about. Consider a certain small neighborhood of this point of radius $\varepsilon_i > 0$. Now the system's trajectory should be transferred into this neighborhood despite the uncertainty $v(\cdot)$. Of course, the inequality $\varepsilon_i < r_i$ must be valid.

Some (or even all) of the ellipsoidal estimates calculated using the formulas (16), (17) may degenerate at some point of time $t = \tilde{t} < 0$. I.e. it may turn out that $W(\tilde{t}) \not> 0$. In this case the process of calculating such an estimate should be stopped, and it should be taken into account in the main algorithm that this estimate is valid only for $t \in [\tilde{t}, 0]$ and not on the entire interval $[T, 0]$.

The corresponding feedback control that transfers trajectories of the system (14) from the state (t, x), $t < 0$, to the ε_i-neighborhood of the point $x^{(i)}$ at the time instant $t = 0$, can be found by "aiming" at one of the constructed ellipsoids $\mathcal{E}(w(t), W(t))$ similarly to (9). The main properties of such a feedback control are indicated in the following statement

Theorem 3. *Let for some indices j, k there exist a vertex $v_j \in \mathcal{E}(w(t_{\min,j}), W_k(t_{\min,j}))$, and $W_k(t) > 0$, $\forall t \in [t_{\min,j}, 0]$. Consider the Cauchy problem for the differential inclusion $\dot{x} \in A(x^{(i)})x + B(x^{(i)})u^*(t, x) + h(x^{(i)}) + v(t)$, $x(t_{\min,j}) = v_j$, $t \in [t_{\min,j}, 0]$. For any admissible noise $v(\cdot)$ (i.e., any measurable function satisfying constraints* (15)*), there exists a solution to the specified Cauchy problem $x(t) = x(t, t_{\min,j}, v_j)|_{u^*(\cdot),v(\cdot)}$ such that $x(0) \in \mathbb{B}_{\varepsilon_i}(x^{(i)})$. In particular, among the trajectories constructed in this way there is also the function $x^*(t)$, which is the solution to the Cauchy problem for the differential inclusion $\dot{x} \in f(x) + g(x)u^*(t, x)$, $x(t_{\min,j}) = v_j$, $t \in [t_{\min,j}, 0]$.*

As the linearized system (14) appropriately approximates the nonlinear system (12) only for $x \in \mathbb{B}_{r_i}(x^{(i)})$, then it is necessary to change (compared to the case with linear dynamics) the conditions for filtering out appropriate trajectories (line 13 of the main algorithm):

$$x^*(j, k^*, t) \in \mathcal{X}_{free} \cap \mathbb{B}_{r_i}(x^{(i)}),\ \forall t \in [t_{\min,j}, 0].$$

To obtain the optimal pair $(x(t), u(t))$ for which $x(t_{\min,j}) = v_j$, $x(0) \in \mathbb{B}_{\varepsilon_i}(x^{(i)})$, it is necessary to modify the relations (11) used for the case of linear dynamics:

$$\begin{cases} \dot{x}(t) \in f(x(t)) + g(x(t))u^*(t, x) \\ u^*(t, x) = \begin{cases} p - \dfrac{PB^T l^0(t)}{\sqrt{\langle l^0(t), \mathbf{P} l^0(t)\rangle}}, & B^T l^0(t) \neq 0 \\ \mathcal{E}(p, P) & \text{otherwise}, \end{cases} \qquad l^0(t) = e^{tA^T} s(t), \\ x(t_{\min,j}) = v_j, \end{cases} \tag{18}$$

$$s(t) = \begin{cases} \operatorname{argmax}\{\langle x(t) - w(t), s\rangle - \sqrt{\langle s, W_{j^*}(t) s\rangle} : \ \|s\| \leqslant 1\}, & x(t) \notin \operatorname{int} \mathcal{E}(w(t), W_{j^*}(t)) \\ 0 & \text{otherwise.} \end{cases}$$

The constructed feedback control should be substituted in the original nonlinear system, rather than to its linearized analogue, in order to subsequently obtain the open-loop control.

Note that the trajectory $x(t)$, starting from the boundary of the ellipsoid $\mathcal{E}(w(t_{\min,j}), W_{k^*}(t_{\min,j}))$, for $t > t_{\min,j}$ can get inside the corresponding ellipsoid due to the "non-optimality" of uncertainty $v(\cdot)$. Moreover, it may turn out that the final trajectory will end up at point $x^{(i)}$ (or it's small neighborhood) for some $t^* < 0$. In this case calculation of the trajectory $x(t)$ should be stopped, and the value $(t^* - t_{\min,j^*})$ should be used as a minimal time estimation instead of $-t_{\min,j^*}$. As mentioned above, another situation is also possible, when the trajectory of the system at time $t = 0$ hits the point $\tilde{x} \in \mathbb{B}_{\varepsilon_i}(x^{(i)})$, $\tilde{x} \neq x^{(i)}$. This should also be taken into account in the main algorithm: the point $\tilde{x}$ should be added to the set V of the graph's vertices instead of $x^{(i)}$ (see lines 17–23 of the main algorithm). Here, however, arises a difficulty with that part of the basic RRT* algorithm in which the previously constructed part of the graph should be "rewired" (see lines 19–23). Now the constructed control function does not guarantee that the trajectory will hit the point $x^{(i)}$. Then either it is necessary to consider $\varepsilon_i > 0$ sufficiently small and thereby identify all points from the ε_i-neighborhood of $x^{(i)}$, or to completely abandon "rewiring". In the latter case, however, the quality of the algorithm may worsen.

7. EXAMPLE (NONLINEAR DYNAMICS)

Let's consider a mathematical model of the planar motion of an autonomous vehicle (robot) driven by two rotors. The equations of motion are as follows:

$$\begin{cases} m\ddot{x}_1 = -\xi_1\dot{x}_1 + (u_1+u_2)\cos(\varphi) \\ m\ddot{x}_2 = -\xi_1\dot{x}_2 + (u_1+u_2)\sin(\varphi) \\ J\ddot{\varphi} \;\; = -\xi_2\dot{\varphi} + (u_1-u_2)r. \end{cases} \tag{19}$$

Here (x_1, x_2) is the position of the vehicle's center of mass, φ – the angle that specifies its orientation on the plane, m – the mass, J – the moment of inertia, ξ_1 and ξ_2 – the coefficients of viscous friction, r – the radius of the device. u_1 and u_2 here denote the forces from two rotors. The following constraints are imposed on the control parameters: $u_i \in [0, u_{\max}]$, $i = 1, 2$, where the value of $u_{\max}$ is given.

Let's reduce the system (19) to the following form:

$$\dot{x} = Ax + g(x)u. \tag{20}$$

Here $x \in \mathbb{R}^6$, $x_3 = \varphi$, $x_4 = \dot{x}_1$, $x_5 = \dot{x}_2$, $x_6 = \dot{x}_3$, $u \in \mathbb{R}^2$,

$$A = \begin{pmatrix} \mathbb{O}_{3\times3} & \mathbb{I}_{3\times3} \\ \mathbb{O}_{3\times3} & -\tilde{A} \end{pmatrix},\; g(x) = \begin{pmatrix} \mathbb{O}_{3\times2} \\ \tilde{g} \end{pmatrix},\; \tilde{A} = \operatorname{diag}\left(\frac{\xi_1}{m}, \frac{\xi_1}{m}, \frac{\xi_2}{J}\right),\; \tilde{g} = \begin{pmatrix} \dfrac{\cos(x_3)}{m} & \dfrac{\cos(x_3)}{m} \\ \dfrac{\sin(x_3)}{m} & \dfrac{\sin(x_3)}{m} \\ \dfrac{r}{J} & -\dfrac{r}{J} \end{pmatrix}.$$

Let's estimate the set of admissible control values using an ellipsoid:

$$u \in \mathcal{E}(p, P), \quad p = \left(\frac{u_{\max}}{2}, \frac{u_{\max}}{2}\right),\; P = \operatorname{diag}\left(\frac{u_{\max}^2}{4}, \frac{u_{\max}^2}{4}\right).$$

Linearizing the equations (20) in the neighborhood of the graph vertex $x^{(i)}$ (see (14)), we obtain the following parameters: $B(x^{(i)}) = g(x^{(i)})$,

$$A(x^{(i)}) = A + \frac{u_{\max}}{m}\begin{pmatrix} & & \mathbb{O}_{3\times6} & & & \\ 0 & 0 & -\sin(x_3^{(i)}) & 0 & 0 & 0 \\ 0 & 0 & \cos(x_3^{(i)}) & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix},\; h(x^{(i)}) = \frac{u_{\max}x_3^{(i)}}{m}\begin{pmatrix} \mathbb{O}_{3\times1} \\ \sin(x_3^{(i)}) \\ -\cos(x_3^{(i)}) \\ 0 \end{pmatrix},$$

$$R_1 = R_2 = R_3 = R_6 = 0,$$

$$R_4 = \frac{r_i^2 u_{\max}}{2m}\max\left\{|\cos(\zeta)|:\; |\zeta - x_3^{(i)}| \leqslant r_i\right\} + \frac{r_i u_{\max}}{m}\max\left\{|\sin(\zeta)|:\; |\zeta - x_3^{(i)}| \leqslant r_i\right\},$$

$$R_5 = \frac{r_i^2 u_{\max}}{2m}\max\left\{|\sin(\zeta)|:\; |\zeta - x_3^{(i)}| \leqslant r_i\right\} + \frac{r_i u_{\max}}{m}\max\left\{|\cos(\zeta)|:\; |\zeta - x_3^{(i)}| \leqslant r_i\right\}.$$

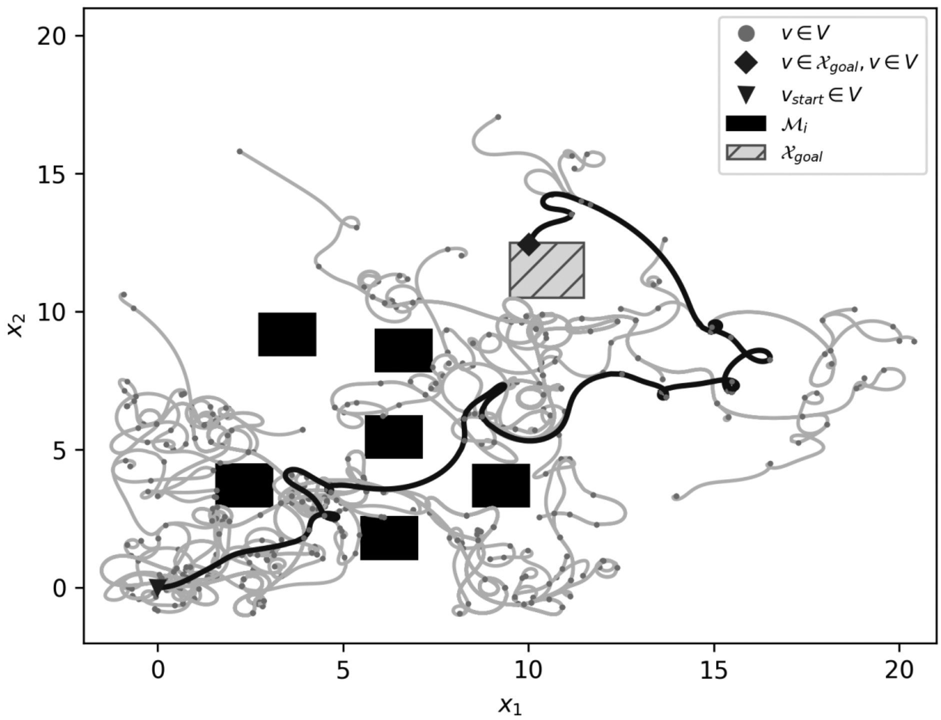

Fig. 3. Projection of the tree $\Gamma = (V, E)$ onto the plane (x_1, x_2).

We define obstacles $\mathcal{M}_i$ so that constraints will be imposed only on the coordinates x_1 and x_2. Below are the results of the Ellipsoidal KRRT* algorithm in this case. Figure 3 shows the projections of the resulting graph Γ edges, as well as obstacles onto the plane of variables (x_1, x_2) at the algorithm's iteration when $|V| = 369$.

The parameters that were used in this numerical simulation: $T = -15$, $m = 5$ kg, $J = 0.05\ \frac{\text{kg}\cdot\text{m}}{\text{rad}}$, $r = 0.24$ m, $\xi_1 = 8.5\ \frac{\text{kg}}{\text{s}}$, $\xi_2 = 0.08\ \frac{\text{kg}\cdot\text{m}^2}{\text{s}\cdot\text{rad}}$, $u_{\max} = 1.5$ N, $x_1(0) = \dot{x}_1(0) = x_2(0) = \dot{x}_2(0) = \varphi(0) = \dot{\varphi}(0) = 0$.

For the given target set

$$X_{goal} = \{x \in \mathbb{R}^2 \colon 9.5 < x_1 < 11.5,\ 10.5 < x_2 < 12.5\}$$

the value of the functional in this example is as follows:

$$t^* = \min\{C(v) :\ v \in V,\ v \in X_{goal}\} = 134.42.$$

Unlike the linear version of the algorithm proposed in this paper, it is not possible to compare its operation in the case of nonlinear dynamics with the existing KRRT* type algorithms due to the fact that the latter either don't allow systems with nonlinear dynamics at all or can only be used to calculate specific, specially selected examples.

8. CONCLUSION

The modification of the RRT* algorithm considered in this article makes it possible to construct suboptimal trajectories for objects with both linear and nonlinear dynamics. The software implementation of the proposed method can be used to solve various applied problems. In some situations the presented algorithm can be improved (be made more efficient) by the use of various heuristics. Further work of the authors will be devoted to the study of these issues.

FUNDING

The work was supported by the Russian Scientific Foundation, project no. 22-11-00042.

REFERENCES

1. Kazakov, K.A. and Semenov, V.A., An Overview of Modern Methods for Motion Planning, *Proc. of the Institute for System Programming of RAS*, 2016, vol. 28, no. 4, pp. 241–294.
2. Paden, B., Cap, M., Yong, S.Z., Yershov, D., and Frazzoli, E., A Survey of Motion Planning and Control Techniques for Self-Driving Urban Vehicles, *IEEE Transactions on Intelligent Vehicles*, 2016, vol. 1, no. 1, pp. 33–55.
3. Karaman, S. and Frazzoli, E., Sampling-Based Algorithms for Optimal Motion Planning, *Int. J. Robot. Res.*, 2011, vol. 30, no. 7, pp. 846–894.
4. Arutyunov, A.V., Magaril-Il'yaev, G.G., and Tikhomirov, V.M., *Printsip maksimuma Pontryagina* (Pontryagin's Maximum Principle), 2006, Moscow: Faktorial.
5. Dubovitskii, A.Ya. and Milyutin, A.A., Extremum Problems in the Presence of Restrictions, *USSR Computational Mathematics and Mathematical Physics*, 1965, vol. 5, no. 3, pp. 1–80.
6. Webb, D.J. and van der Berg, J., Kinodynamic RRT*: Asymptotically Optimal Motion Planning for Robots with Linear Dynamics, *Proc. of the IEEE Conf. on Robotics and Automation*, 2013, pp. 5054–5061.
7. Karaman, S. and Frazzoli, E., Optimal Kinodynamic Motion Planning Using Incremental Sampling-Based Methods, *Proc. of the 49th IEEE Conference on Decision and Control*, 2010, pp. 7681–7687.
8. LaValle, S.M. and Kuffner, J.J., Randomized Kinodynamic Planning, *Int. J. Robot. Res.*, 2001, vol. 20, no. 5, pp. 378–400.
9. Shkolnik, A., Walter, M., and Tedrake, R., Reachability-Guided Sampling for Planning under Differential Constraints, *Proc. of the IEEE Conf. on Robotics and Automation*, 2009, pp. 2859–2865.
10. Kurzhanski, A.B. and Varaiya, P., On Ellipsoidal Techniques for Reachability Analysis. Part II: Internal Approximations, Box-Valued Constraints, *Optimization Methods and Software*, 2002, vol. 17, pp. 207–237.
11. Kurzhanski, A.B. and Varaiya, P., Reachability Analysis for Uncertain Systems – the Ellipsoidal Technique, *Dynam. Contin. Discrete Impuls. Syst. Ser. B*, 2002, vol. 9, no. 3, pp. 347–367.
12. Kurzhanski, A.B. and Varaiya, P., Dynamics and Control of Trajectory Tubes. Theory and Computation. Birkhäuser, 2014.

This paper was recommended for publication by A.A. Galyaev, a member of the Editorial Board