

УДК 517.977, 519.6, 004.02

О ПОСТРОЕНИИ СУБОПТИМАЛЬНЫХ ТРАЕКТОРИЙ ДЛЯ ЛИНЕЙНОЙ УПРАВЛЯЕМОЙ СИСТЕМЫ ПРИ ФАЗОВЫХ ОГРАНИЧЕНИЯХ ПО ЧАСТИ ПЕРЕМЕННЫХ¹

П. А. Точилин, М. В. Паршиков

В статье рассматривается задача управления линейной системой дифференциальных уравнений при фазовых ограничениях и поточечных ограничениях на управляющие параметры. Такие задачи часто встречаются в приложениях из робототехники, в проблемах управления автономным движением на плоскости или в пространстве. Точное математическое решение в подобных задачах получить, как правило, не удастся, а традиционные численные методы могут быть неэффективными из-за медленной скорости работы. В последние годы широкое распространение получили методы приближенного решения задач управления с фазовыми ограничениями при помощи случайных графов. Они показали высокую эффективность в случае с тривиальной динамикой управляемого объекта, когда возможно движение без инерции, по ломаным линиям. При этом задачи с так называемыми кинодинамическими ограничениями (когда динамика описывается нетривиальными дифференциальными уравнениями) до недавнего времени оставались нерешенными. Существенного продвижения в этой области удалось достичь за счет совмещения идей построения случайных графов и методов эллипсоидального оценивания, разработанных ранее академиком А. Б. Куржанским и его учениками. Данная статья продолжает исследования в этой области. Авторами предложена новая модификация ранее разработанных методов, которая позволит увеличить их эффективность, сделать подходящими для решения конкретных прикладных задач. Улучшение характеристик метода удалось достичь за счет выделения переменных, отвечающих за фазовые ограничения, и отдельной обработки таких переменных и оставшейся части вектора состояния.

Ключевые слова: оптимальное управление, фазовые ограничения, случайный граф, эллипсоидальное исчисление, эллипсоидальный синтез, множество достижимости.

P. A. Tochilin, M. V. Parshikov. On the construction of suboptimal trajectories for a linear control system with state constraints on part of variables.

The article considers the problem of controlling a linear system of differential equations under state constraints and pointwise restrictions on control parameters. Such problems are often found in applications from robotics, in the field of controlling autonomous movement on a plane or in space. As a rule, it is not possible to obtain an exact mathematical solution to such problems, and traditional numerical methods may be ineffective due to the slow speed of work. In recent years, methods for approximate solution of control problems with state constraints using random graphs have become widespread. They have shown high efficiency in the case of trivial dynamics of a controlled object, when it is possible to move without inertia along broken lines. At the same time, problems with so-called kinodynamic constraints (when dynamics are described by nontrivial differential equations) have remained unsolved until recently. Significant progress in this area has been achieved by combining the ideas of constructing random graphs and ellipsoidal estimation techniques developed earlier by academician A. B. Kurzhansky and his students. This article continues the research in this area. The authors propose a new modification of the previously developed methods, which will increase its effectiveness and make it suitable for solving specific applied problems. The improvement of the method's characteristics was achieved by allocating the variables responsible for state constraints and separately processing such variables and the remaining part of the state vector.

Keywords: optimal control, state constraints, random graph, ellipsoidal calculus, ellipsoidal synthesis, reachability set.

MSC: 93B03, 93B40, 93B52, 49M30, 49N05

DOI: 10.21538/0134-4889-2025-31-2-fon-03

Введение

Данная работа посвящена приближенному решению задач оптимального управления линейными системами обыкновенных дифференциальных уравнений при наличии фазовых огра-

¹Работа выполнена при финансовой поддержке Минобрнауки РФ в рамках реализации программы Московского центра фундаментальной и прикладной математики по соглашению № 075-15-2022-284.

ничений и жестких поточечных ограничений на управления. Необходимо перевести траекторию системы из заданной начальной позиции в целевое множество, минимизировав при этом некоторый функционал качества (например, время движения).

В простейшем случае, когда допускается движение управляемого объекта по прямой с мгновенным разворотом на месте, для решения указанной выше задачи хорошо себя зарекомендовали алгоритмы, связанные с построением в фазовом пространстве случайных деревьев (алгоритмы RRT, RRT* и другие) или графов более общего вида (например, методы PRM, PRM*) (см. [1–4]). В частности, алгоритм RRT* (*Rapidly Exploring Random Trees*) состоит в последовательном (по шагам) построении случайного дерева, корнем которого является заданная начальная точка. На каждом шаге случайным образом добавляется в фазовое пространство новая точка (метку), соединив ее оптимальным образом с одной из ранее добавленных вершин дерева. Если построенное при этом ребро графа не пересекается с препятствиями (т. е. выполнены фазовые ограничения), то новая вершина и соответствующее ребро добавляются к дереву. Кроме того, каждой вершине сопоставляется минимальное время ее достижения из корневой вершины, которое может пересчитываться (обновляться) на каждом следующем шаге работы алгоритма. Особенностью алгоритма RRT* (в отличие от более простого RRT) является специальная процедура “пересвязывания” дерева, которая позволяет локально менять его структуру: заменять одни ребра на другие с целью минимизации функционала качества (времени достижения вершины). Алгоритм может завершить свою работу после попадания очередной новой вершины дерева в целевое множество.

В случае нетривиальных дифференциальных уравнений, задающих закон движения, необходимо учитывать так называемые “кинодинамические” ограничения [5–8]. В соответствующих алгоритмах (например, KRRT*, использующем случайные деревья) ключевой является подзадача о переводе управляемого объекта из точки в точку на малом отрезке времени. Такую задачу нужно решать многократно и достаточно быстро, чтобы получившийся в результате метод был эффективным. Так, в статье [5] предложен подход к решению задачи быстродействия при помощи случайных деревьев (алгоритм RRT*) или графов более общей структуры (алгоритм FMT*) для нелинейных систем дифференциальных уравнений с возможными поточечными ограничениями на управляющие параметры, а также с фазовыми ограничениями. Для решения подзадачи об оптимальном переводе системы из одной заданной точки в другую используется численный метод, состоящий в дискретизации задачи оптимального управления и сведении поиска ее приближенного решения к последовательности вспомогательных задач квадратичного программирования.

Разработке модификации алгоритма KRRT* для линейных и нелинейных управляемых систем достаточно общего вида с жесткими ограничениями на управления была посвящена работа [9]. Настоящая статья в основном использует идеи указанной работы и является ее логическим продолжением.

Во многих приложениях встречаются задачи управления, в которых фазовые ограничения наложены не на весь вектор фазовых переменных, а лишь на его часть. Например, в задаче управления колесным роботом в помещении с препятствиями имеет смысл рассматривать только ограничения на положение робота в каждый промежуточный момент времени, но не на значения вектора скорости или угла поворота его колес. При решении такого рода задач, конечно, можно использовать разные модификации методов случайных графов и деревьев в полном фазовом пространстве, но соответствующие алгоритмы имеют избыточную сложность, связанную с тем, что на каждой итерации их работы при создании очередной метки для искомого пути производится попытка подобрать значение тех фазовых переменных, которые не влияют на качество итогового движения. Более эффективный подход предполагает выделение таких “дополнительных” фазовых переменных в процессе решения локальной подзадачи об оптимальном переводе траектории системы. Соответственно, вместо задачи о переводе системы “из точки в точку” необходимо рассматривать задачу о ее переводе “из точки во множество”, где в целевом множестве зафиксирована лишь часть компонент фазового вектора, а остальные

компоненты могут быть произвольными. Идея ускорения работы алгоритмов, использующих случайные деревья, за счет уменьшения размерности пространства меток (вершин деревьев) не нова (см., например, [10]). Однако, на данный момент рассмотрены лишь наиболее простые постановки таких задач, когда либо динамика системы является очень простой, либо в задаче управления отсутствуют поточечные ограничения на управляющие параметры. Целью настоящего исследования является разработка более общего метода приближенного решения задач управления линейными управляемыми системами с жесткими ограничениями на управления для разных видов функционалов качества. Для этого предлагается использовать некоторые методы эллипсоидального исчисления [11; 12]. В данной статье представлены общие схемы таких алгоритмов, разобраны некоторые наиболее сложные детали их реализации. Работа алгоритмов продемонстрирована на конкретных примерах.

1. Задачи управления

В пространстве $\mathbb{R}^n$ рассмотрим некоторое компактное множество Ω . Пусть движение управляемого объекта описывается системой обыкновенных дифференциальных уравнений

$$\dot{z} = Az + Bu + f, \quad z \in \Omega, \quad t \in \mathbb{R}_+ = [0, +\infty), \quad (1.1)$$

где $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$, $f \in \mathbb{R}^n$ — заданные матрицы. Предположим, что начальное состояние $z(0) = z_0 \in \Omega$ фиксировано. Система содержит управляющие параметры $u \in \mathbb{R}^m$, на допустимые значения которых наложены жесткие поточечные ограничения: $u \in \mathcal{P} = \mathcal{E}(p, P)$. Здесь $\mathcal{E}(p, P)$ — эллипсоид с центром $p \in \mathbb{R}^m$ и матрицей конфигурации $P \in \mathbb{R}^{m \times m}$, $P = P^T \geq 0$:

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

Если матрица P не вырождена, то справедливо альтернативное определение эллипсоида:

$$\mathcal{E}(p, P) = \{y \in \mathbb{R}^m: \langle y - p, P^{-1}(y - p) \rangle \leq 1\}. \quad (1.2)$$

Через $\mathcal{U}$ обозначим класс *допустимых программных управлений*, который содержит все возможные измеримые функции $u = u(t)$ такие, что для почти всех $t \in \mathbb{R}_+$ $u(t) \in \mathcal{P}$. Через $z(t, 0, z_0)|_{u(\cdot)}$ обозначим траекторию системы (1.1), порожденную некоторым допустимым управлением $u(\cdot)$, выпущенную в начальный момент времени $t_0 = 0$ из начальной позиции z_0 . Будем далее рассматривать только части траекторий $z(t, 0, z_0)|_{u(\cdot)}$, которые содержатся во множестве Ω .

В векторе фазовых переменных z выделим первые $n_1 \leq n$ компонент, $n_1 \in \mathbb{N}$. Пусть также $n_2 = n - n_1$. Первые n_1 компонент вектора z будем всюду далее обозначать буквой x , а остальные n_2 компонент — буквой y : $z = (x; y)$. Здесь и далее точкой с запятой будет обозначено объединение двух векторов-столбцов в один вектор большей длины.

Рассмотрим некоторый набор множеств $\mathcal{M}_i \subset \mathbb{R}^{n_1}$, $i = 1, \dots, M$, — препятствий, которые необходимо учитывать при движении объекта. Заметим, что в отличие от задачи, рассмотренной в работе [9], здесь мы будем предполагать, что фазовые ограничения касаются лишь части вектора z , а именно компоненты x . Это позволит далее построить более эффективный алгоритм решения задачи управления.

Пусть $\text{Pr}_1 \in \mathbb{R}^{n_1 \times n}$ — матрица, выделяющая первые n_1 координат n -мерного вектора ($\text{Pr}_1 z = x$), и

$$\mathcal{M}_i = \{x \in \text{Pr}_1 \Omega: \varphi_{i,j}(x) \leq 0, \quad j = 1, \dots, s_i\}, \quad (1.3)$$

где непрерывные функции $\varphi_{i,j}(x)$, а также натуральные числа s_i заданы. Положения препятствий $\mathcal{M}_i$ не изменяются со временем, т. е. функции $\varphi_{i,j}$ не зависят от t . Далее будем также использовать обозначение Pr_2 для матрицы из $\mathbb{R}^{n_2 \times n}$, позволяющей выделить последние n_2 компонент фазового вектора: $\text{Pr}_2 z = y$.

Введем следующее обозначение для области безопасного (свободного от препятствий) движения в пространстве $\mathbb{R}^{n_1}$:

$$\mathcal{X}_{free} = (\text{Pr}_1 \Omega) \setminus \overline{\left(\bigcup_{i=1}^M \mathcal{M}_i \right)}.$$

Предположим, что $\text{Pr}_1 z_0 \in \mathcal{X}_{free}$. Зафиксируем также некоторое компактное множество $\mathcal{Z}_{goal} \subset \Omega$ такое, что $\text{Pr}_1 \mathcal{Z}_{goal} \subset \mathcal{X}_{free}$, $z_0 \notin \mathcal{Z}_{goal}$.

Теперь можно сформулировать *основные задачи управления*, решению которых посвящена данная работа.

1. Необходимо найти $u(\cdot) \in \mathcal{U}$, для которого найдется $t_1 > 0$ и такая траектория $z(t) = z(t, 0, z_0)|_{u(\cdot)}$, $t \in [0, t_1]$, дифференциального уравнения (1.1), для которой

$$z(t_1) \in \mathcal{Z}_{goal}, \quad \text{Pr}_1 z(\tau) \in \mathcal{X}_{free}, \quad z(\tau) \in \Omega \quad \forall \tau \in [0, t_1].$$

Здесь t_1 — момент первого попадания траектории в целевое множество $\mathcal{Z}_{goal}$.

2. Среди управлений и соответствующих траекторий, найденных в предыдущем пункте, необходимо выделить такие, для которых величина $t_1 \geq 0$ будет наименьшей, т. е. нужно вычислить верхнюю оценку времени быстрогодействия.
3. Среди управлений и соответствующих траекторий, найденных в п. 1, необходимо выделить такие, для которых будет наименьшим значение функционала

$$J(u(\cdot)) = \int_0^{t_1} (a^T z(\tau, 0, z_0)|_{u(\cdot)} + b^T u(\tau) + c) d\tau, \quad (1.4)$$

где векторы $a \in \mathbb{R}^n$, $b \in \mathbb{R}^m$, а также скаляр c фиксированы.

Легко увидеть, что задача быстрогодействия из п. 2 является частным случаем задачи 3 при $a = 0$, $b = 0$, $c = 1$. Однако, мы специально выделили эту задачу в отдельный случай из-за особенностей ее решения, о которых речь пойдет ниже.

2. Задача быстрогодействия для линейной системы

Сформулированные выше задачи управления будем решать приближенно при помощи модификации алгоритма KRRT*, а точнее ранее разработанного [9] метода KRRT* с использованием методов эллипсоидального исчисления. Базовый алгоритм будем далее обозначать как EKRRRT*.

Ниже приведена общая схема нового алгоритма для решения задач 1 и 2. Основной целью здесь является построение ориентированного дерева $\Gamma = (V, E)$, где V — множество вершин, E — множество дуг. Вершинами являются векторы из множества $\text{Pr}_1 \Omega \subset \mathbb{R}^{n_1}$, причем точка $v_0 = \text{Pr}_1 z_0$ — корень дерева. Каждой вершине $v \in V$ сопоставим минимальное (из найденных) время ее достижения $C(v)$ из начальной точки v_0 . Также каждой вершине сопоставим вектор $y = y(v) \in \mathbb{R}^{n_2}$, содержащий дополнительные компоненты позиции траектории системы (1.1), т. е. $z = (v; y(v))$. Для корня дерева положим $C(v_0) = 0$, $y(v_0) = \text{Pr}_2 z_0$. Для каждой вершины $v \neq v_0$ через $\text{parent}(v)$ обозначим ту вершину Γ , для которой $(\text{parent}(v), v) \in E$ — дуга графа, соединяющая вершины $\text{parent}(v)$ и v .

Алгоритм является итерационным, i — номер итерации, а фиксированное натуральное число N — максимальное количество итераций. На каждом шаге работы алгоритма необходимо решить вспомогательную подзадачу о переводе траектории системы (1.1) из одного множества в другое за наименьшее время. Для решения этой подзадачи необходимо построить вспомогательные множества разрешимости [12, с. 61–62] или их внутренние эллипсоидальные оценки $\mathcal{E}(w(t), W_k(t))$, $k = 1, \dots, K$. Далее требуемое управление можно найти за счет “прицеливания” на трубку разрешимости или на соответствующие эллипсоидальные оценки. Последние

должны быть построены в пространстве полной размерности $\mathbb{R}^n$. Количество таких оценок K считается фиксированным. Каждая оценка зависит от вектора параметров $l_k \in \mathbb{R}^n$ и определена при $t \in [T_1, 0]$, где число $T_1 < 0$ задано (это параметр метода). Формулы для вычисления оценок будут приведены ниже. Заметим, что поскольку система (1.1) является стационарной, то при построении эллипсоидальных оценок удобно фиксировать конечный момент времени 0, и далее рассчитывать эти оценки при отрицательных t . Однако затем, при “склеивании” итоговой траектории (соответствующей функции управления) из частей, сопоставленных ребрам дерева, нужно сделать замену для параметра t , после которой время станет неотрицательным.

На описанном ниже этапе локального пересвязывания дерева (аналогично идее алгоритма RRT*) вместо оценок множеств разрешимости $\mathcal{E}(w(t), W_k(t))$ необходимо использовать внутренние оценки множеств достижимости [12, с. 58], которые будем обозначать как $\mathcal{E}(\tilde{w}(t), \tilde{W}_k(t))$. Не ограничивая общности, будем считать, что их количество K — то же, что и для оценок множеств разрешимости. Однако, множества достижимости будут построены уже не в попятном, а в прямом времени, на некотором заданном отрезке $[0, T_2]$, где параметр $T_2 > 0$ считается фиксированным.

В приведенном далее алгоритме используются вспомогательные множества вершин графа $\mathcal{V}_0$ и $\mathcal{V}_1$, которые содержат все такие вершины, которые лежат в малой окрестности очередной случайной точки и одновременно в построенных эллипсоидальных оценках. Алгоритмы формирования этих множеств отличаются только тем, что в первом случае будут задействованы оценки множеств разрешимости, а во втором — достижимости. Использование указанных окрестностей (аналогично алгоритму RRT* [3]) позволяет избежать экспоненциального роста сложности вычислений с увеличением количества итераций алгоритма. Вершины из $\mathcal{V}_0$ ($\mathcal{V}_1$) далее должны быть проверены на возможность построения траекторий, решающих вспомогательные подзадачи “из точки во множество”, без нарушения фазовых ограничений.

Основным отличием приведенного ниже алгоритма от ранее рассмотренного в [9] является то, что случайные точки — новые вершины дерева — и должны быть построены не во множестве Ω , а в $\text{Pr}_1 \Omega$, являющемся частью пространства меньшей размерности n_1 .

Каждая итерация основного цикла работы алгоритма (см. схему алгоритма 1) состоит из следующих действий:

- 1) создание очередной случайной точки $x^{(i)}$ (кандидата на роль новой вершины графа) (строка 4);
- 2) вычисление ансамбля эллипсоидальных трубок — внутренних оценок множеств разрешимости (строка 5), выпущенных из целевого множества, в точках которого только первые n_1 компонент фиксированы и совпадают с ранее найденной случайной точкой;
- 3) поиск оптимальной траектории, соединяющей одну из ранее построенных вершин графа с новой вершиной (строки 7–14), за счет “прицеливания” на построенные эллипсоидальные оценки;
- 4) “пересвязывание” ранее построенной части графа для уменьшения оценки времени быстрого действия (см. схему алгоритма 2).

Основная идея процесса пересвязывания дерева аналогична той, что используется в алгоритме RRT* [7], которая отличает его от более простого алгоритма RRT [8]. Эту часть алгоритма можно убрать для повышения скорости его работы; при этом, однако, может значительно ухудшиться итоговая оценка времени быстрого действия (аналогично ситуации с алгоритмами RRT и RRT*, см. [3]).

В строке 4 алгоритма 1 координаты каждой новой точки $x^{(i)}$ являются случайными, соответствующими равномерному распределению на множестве $\mathcal{X}_{free}$. Для повышения эффективности работы алгоритма в конкретных примерах можно использовать и другие распределения. Использование альтернативных вероятностных распределений и алгоритмов вычисления точек $x^{(i)}$ может, например, основываться на априорной информации о положении и форме препятствий, а также о направлении движения от начальной точки к целевому множеству (т. е. можно применять различные эвристики).

Algorithm 1. EKRRT* с фазовыми ограничениями по части переменных

```

1:  $V := \{x_0\}$ ,  $C(x_0) := 0$ 
2: сформируем совокупность векторов  $l_1, \dots, l_K$ ,  $l_k \in \mathbb{R}^n$ ,  $\|l_k\| = 1$ 
3: for  $i = \overline{1, N}$  do
4:   определим случайную точку  $x^{(i)} \in \mathcal{X}_{free}$ 
5:   построим эллипсоидальные оценки  $\mathcal{E}(w(t), W_k(t))$ ,  $k = \overline{1, K}$ ,  $t \in [T_1, 0]$ , выпущенные из
      целевого множества  $x^{(i)} \times \text{Pr}_2 \Omega$ 
6:   сформируем множество  $\mathcal{V}_0$  вершин  $v_j$ ,  $(v_j, y(v_j)) \in \bigcup_{k,t} \mathcal{E}(w(t), W_k(t)) \cap \tilde{\mathbb{B}}_{r_i}(x^{(i)})$ 
7:    $j^* := -1$ 
8:   for  $v_j \in \mathcal{V}_0$  do
9:      $t_{\min,j} := \max_k \max\{t \in [T_1, 0] : (v_j; y(v_j)) \in \mathcal{E}(w(t), W_k(t))\}$ 
10:     $k^* :=$  соответствующий номер эллипсоидальной оценки (максимизатор)
11:    if  $(j^* < 0) \vee ((C(v_j) - t_{\min,j}) < (C(v_{j^*}) - t_{\min,j^*}))$  then
12:      определим оптимальное управление  $u^*(j, k^*, t)$ ,  $t \in [t_{\min,j}, 0]$ 
13:      построим траекторию  $z^*(j, k^*, t)$ ,  $t \in [t_{\min,j}, 0]$ ,  $z^*(j, k^*, t_{\min,j}) = (v_j; y(v_j))$ 
14:      if  $\text{Pr}_1 z^*(j, k^*, t) \in \mathcal{X}_{free}$ ,  $z^*(j, k^*, t) \in \Omega$ ,  $\forall t \in [t_{\min,j}, 0]$  then  $j^* := j$ 
15:   if  $j^* \geq 0$  then
16:      $V := V \cup \{x^{(i)}\}$ ,  $E := E \cup \{(v_{j^*}, x^{(i)})\}$ 
17:      $C(x^{(i)}) := C(v_{j^*}) - t_{\min,j^*}$ 
18:      $y(x^{(i)}) := \text{Pr}_2 z^*(j^*, k^*(j^*), 0)$ 
19:     используем алгоритм 2 для пересвязывания части графа в окрестности вершины  $x^{(i)}$ 

```

В строке 6 алгоритма 1 производится отбор вершин v_j , удовлетворяющих следующему условию:

$$v_j \in \left(\bigcup_{t \in [T, 0]} \bigcup_{k=1}^K \mathcal{E}(w_k(t), W_k(t)) \right) \cap \tilde{\mathbb{B}}_{r_i}(x^{(i)}). \quad (2.1)$$

Аналогичный отбор производится и в алгоритме 2 (строка 2). Здесь множество $\tilde{\mathbb{B}}_{r_i}(x^{(i)})$ является цилиндром в пространстве $\mathbb{R}^n$:

$$\tilde{\mathbb{B}}_{r_i}(x^{(i)}) = \mathbb{B}_{r_i}(x^{(i)}) \times \mathbb{R}^{n_2},$$

где $\mathbb{B}_{r_i}(x^{(i)}) \subset \mathbb{R}^{n_1}$ — замкнутый шар радиуса r_i с центром в точке $x^{(i)}$. Параметр $r_i > 0$ ограничивает перебор вершин графа и отвечает за уменьшение итоговой сложности вычислений. Определим его аналогично [3]:

$$r_i = \min \left\{ \gamma \left(\frac{\ln(\kappa)}{\kappa} \right)^{1/n_1}, \eta \right\}, \quad \gamma > (2(1 + 1/n_1))^{1/n_1} \left(\frac{\mu(\mathcal{X}_{free})}{\zeta_{n_1}} \right)^{1/n_1}, \text{ если } \kappa > 1,$$

и $r_i = \eta$ при $\kappa = 1$. Здесь ζ_{n_1} — объем единичного шара в $\mathbb{R}^{n_1}$, $\mu(\mathcal{X}_{free})$ — лебегова мера множества $\mathcal{X}_{free}$ (также в пространстве $\mathbb{R}^{n_1}$), $\kappa = |V|$ — общее количество ранее добавленных (до i -й итерации) вершин графа. Постоянная $\eta > 0$ может быть взята произвольным образом; эта величина отвечает за уменьшение сложности вычислений на первых итерациях работы алгоритма (при малых κ).

При построении управления и траектории в строках 12, 13 алгоритма 1 (и аналогично в строке 15 алгоритма 2) используется метод “прицеливания” на конкретную эллипсоидальную трубку (разрешимости или достижимости), который будет описан ниже. Функция $z^*(j, k^*, t)$, $t \in [t_{\min,j}, 0]$ ($t \in [0, \tilde{t}_{\min,j}]$ — в случае алгоритма 2), является решением задачи Коши для системы дифференциальных уравнений (1.1) после подстановки управления $u^*(j, k^*, t)$ с краевым условием $z^*(j, k^*, t_{\min,j}) = (v_j; y(v_j))$ ($z^*(j, \tilde{k}^*, 0) = (x^{(i)}; y(x^{(i)}))$).

Algorithm 2. Пересвязывание части графа в окрестности вершины $(x^{(i)}, y(x^{(i)}))$

```

1: Построим эллипсоидальные оценки  $\mathcal{E}(\tilde{w}(t), \tilde{W}_k(t))$ ,  $k = \overline{1, K}$ ,  $t \in [0, T_2]$ , выпущенные из
   начальной позиции  $(x^{(i)}; y(x^{(i)}))$ 
2: сформируем множество  $\mathcal{V}_1$  вершин  $v_j$ , для которых  $v_j \neq v_{j^*}$ ,  $v_j \neq x_0$ ,  $(v_j; y(v_j)) \in$ 
    $\bigcup_{k,t} \mathcal{E}(\tilde{w}(t), \tilde{W}_k(t)) \cap \tilde{\mathbb{B}}_{r_i}(x^{(i)})$ 
3: for  $v_j \in \mathcal{V}_1$  do
4:   if  $v_j$  — концевая вершина дерева  $\Gamma$  then
5:      $\tilde{t}_{\min,j} := \min_k \min\{t \in [0, T_2]: v_j \in \text{Pr}_1 \mathcal{E}(\tilde{w}(t), \tilde{W}_k(t))\}$ 
6:   else
7:      $\tilde{t}_{\min,j} := \min_k \min\{t \in [0, T_2]: (v_j; y(v_j)) \in \mathcal{E}(\tilde{w}(t), \tilde{W}_k(t))\}$ 
8:   if  $\tilde{t}_{\min,j} < \infty$  then
9:      $\tilde{k}^* :=$  соответствующий минимизатор
10:    if  $v_j$  — концевая вершина дерева  $\Gamma$  then
11:       $y^* —$  любой вектор, для которого  $(v_j; y^*) \in \mathcal{E}(\tilde{w}(\tilde{t}_{\min,j}), \tilde{W}_{\tilde{k}^*}(\tilde{t}_{\min,j}))$ 
12:    else
13:       $y^* = y(v_j)$ 
14:    if  $\Delta_j = C(x^{(i)}) + \tilde{t}_{\min,j} - C(v_j) < 0$  then
15:      определим оптимальную пару  $u^*(j, \tilde{k}^*, t)$ ,  $z^*(j, \tilde{k}^*, t)$ ,  $t \in [0, \tilde{t}_{\min,j}]$ ,  $z^*(j, \tilde{k}^*, 0) =$ 
 $(x^{(i)}; y(x^{(i)}))$ ,  $z^*(j, \tilde{k}^*, \tilde{t}_{\min,j}) = (v_j; y^*)$ 
16:      if  $\text{Pr}_1 z^*(j, \tilde{k}^*, t) \in \mathcal{X}_{free}$ ,  $z^*(j, \tilde{k}^*, t) \in \Omega$ ,  $\forall t \in [0, \tilde{t}_{\min,j}]$  then
17:        для каждой вершины  $v'$  поддерева с корнем  $v_j$   $C(v') := C(v') + \Delta_j$ 
18:        if  $v_j$  — концевая вершина дерева  $\Gamma$  then  $y(v_j) := y^*$ 
19:         $E := E \setminus \{(parent(v_j), v_j)\}$ ,  $E := E \cup \{(x^{(i)}, v_j)\}$ 

```

В схему алгоритма 1 может быть добавлено дополнительное условие выхода из внешнего цикла при первом достижении целевого множества, т. е. если на очередной итерации с номером i (после успешного добавления новой вершины в граф) выполнено условие

$$(x^{(i)}; y(x^{(i)})) \in \mathcal{Z}_{goal}.$$

Это условие необходимо при решении первой из трех описанных выше основных задач управления. Однако, при решении задачи быстродействия имеет смысл не останавливать алгоритм при обнаружении первого допустимого пути до целевого множества, а продолжить вычисления с целью минимизации времени перехода.

Для простоты изложения в приведенной выше схеме алгоритма для ребер графа не была указана сопоставленная с ними дополнительная информация. Однако, для решения основных задач управления понадобится дополнительно запоминать для каждого ребра программное управление, которое было подсчитано в строке 12 алгоритма 1 и строке 15 алгоритма 2.

После построения дерева Γ можно легко решить сформулированные выше первые две задачи управления:

- Если $\exists v_j \in V$, $(v_j; y(v_j)) \in \mathcal{Z}_{goal}$, то задача 1 разрешима и решена.
- Если выполнено условие из предыдущего пункта, то верхняя оценка времени быстродействия в задаче 2 может быть найдена следующим образом:

$$t_1^* = \min\{C(v_j): v_j \in V, (v_j; y(v_j)) \in \mathcal{Z}_{goal}\}. \quad (2.2)$$

Пусть также v^* — произвольный минимизатор в (2.2).

- Оптимальное управление $u^*(t)$ теперь может быть составлено из частей, сопоставленных отдельным дугам дерева в процессе его построения. При этом нужно двигаться из

вершины v^* к корню дерева, т. е. по дугам

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

учитывая обратный ход времени при объединении частей функции $u^*(t)$. Зная управление $u^*(t)$, $t \in [0, t_1^*]$, и начальную позицию z_0 , далее можно легко построить соответствующую траекторию системы (1.1).

3. Эллипсоидальные оценки для линейной системы

Наиболее сложная часть описанного выше алгоритма состоит в построении внутренних эллипсоидальных оценок *множеств разрешимости* или *достижимости* системы (1.1), выпущенных из окрестности каждой очередной случайно подобранной точки $x^{(i)} \in \mathcal{X}_{free}$. Рассмотрим сначала задачу построения внутренних эллипсоидальных оценок $\mathcal{E}(w(t), W_k(t))$ (см. строку 5 алгоритма 1) для множества разрешимости.

Для фиксированного множества $\mathcal{Z}_1 \subset \Omega$, а также для некоторых вещественных $t \leq t_1$ множество разрешимости $\mathcal{W}(t, t_1, \mathcal{Z}_1)$ системы (1.1) состоит из всех возможных точек $z_0 \in \mathbb{R}^n$, для каждой из которых существует некоторое управление $u(\cdot) \in \mathcal{U}$, для которого $z(t_1, t, z_0)|_{u(\cdot)} \in \mathcal{Z}_1$. Задача точного построения таких множеств является нетривиальной. В случае линейной динамики одним из наиболее эффективных подходов для их приближенного построения является использование эллипсоидальных оценок: внешних или внутренних, в зависимости от конкретной решаемой задачи. В данной работе основной целью является поиск управления, гарантирующего определенное значение функционала качества. Для его вычисления удобно использовать *внутренние эллипсоидальные оценки множеств разрешимости*. Каждое нужное нам множество разрешимости в каждый фиксированный момент времени t будет оценено при помощи K параметризованных эллипсоидальных оценок с общим центром $w(t)$, но разными матрицами конфигурации $W_k(t)$.

Для построения оценок $\mathcal{E}(w(t), W_k(t))$ множеств разрешимости $\mathcal{W}(t, 0, x^{(i)} \times \text{Pr}_2 \Omega)$, $t \in [T_1, 0]$, нужно сначала построить внутреннюю эллипсоидальную оценку целевого множества $x^{(i)} \times \text{Pr}_2 \Omega$. Пусть $\mathcal{E}(q, Q)$ — некоторый эллипсоид в пространстве $\mathbb{R}^{n_2}$, являющийся внутренней оценкой $\text{Pr}_2 \Omega$. Тогда множество $x^{(i)} \times \text{Pr}_2 \Omega$ можно оценить изнутри эллипсоидом $\mathcal{E}(z^*, Z^*)$, где $z^* = (x^{(i)}; q)$ и $Z^* = \text{diag}(\mathbb{O}_{n_1 \times n_1}, Q)$. Здесь $\mathbb{O}_{n_1 \times n_1}$ — нулевая матрица размера $n_1 \times n_1$. Пусть $\mathbf{P} = B P B^T$. Используя [12, с. 128–130], определим семейство эллипсоидальных оценок $\mathcal{E}(w(t), W(t))$, $t \in [T_1, 0]$, задаваемых следующими дифференциальными уравнениями:

$$\begin{cases} \dot{w}(t) = Aw + Bp + f, \\ w(0) = z^*, \end{cases} \quad (3.1)$$

$$\begin{cases} \dot{W}(t) = AW(t) + W(t)A^T - W^{1/2}(t)S(t)\mathbf{P}^{1/2} - \mathbf{P}^{1/2}S^T(t)W^{1/2}(t), \\ W(0) = Z^*. \end{cases} \quad (3.2)$$

Каждая эллипсоидальная оценка зависит от некоторой кривой $l(t)$, являющейся решением следующей вспомогательной задачи Коши:

$$\begin{cases} \dot{l} = -A^T l, & t \in [T_1, 0], \\ l(0) = l^* \end{cases}$$

для произвольного вектора $l^* \in \mathbb{R}^n$: $\|l^*\| = 1$. Будем считать, что в представленном выше алгоритме используется конечный набор таких векторов $l^* \in \{l_1, \dots, l_K\}$, а для того, чтобы подчеркнуть зависимость матриц конфигураций эллипсоидов от этих векторов, будем использовать обозначение $W_k(t) = W(t, l_k)$ для матрицы, которая построена при использовании вектора $l^* = l_k$.

В уравнении (3.2) $S(t)$, $t \in [T_1, 0]$, — ортогональные матрицы, непрерывно зависящие от t . Для них должны быть выполнены следующие соотношения:

$$(Z^*)^{1/2}l^* = \lambda_0 l_0, \quad S(t)\mathbf{P}^{1/2}l(t) = \lambda(t)l_0, \quad \lambda(t) = \sqrt{\frac{\langle l(t), \mathbf{P}l(t) \rangle}{\langle l_0, l_0 \rangle}} \quad \forall t \leq 0.$$

Здесь l_0 — некоторый фиксированный ненулевой вектор. Если $(Z^*)^{1/2}l^* \neq 0$, то в качестве l_0 можно взять $(Z^*)^{1/2}l^*$ и положить $\lambda_0 = 1$. При $(Z^*)^{1/2}l^* = 0$ положим $\lambda_0 = 0$. При этом если для некоторого $t^* \in [T_1, 0]$ $\mathbf{P}^{1/2}l(t^*) \neq 0$, то можно взять $l_0 = \mathbf{P}^{1/2}l(t^*)$, $S(t^*) = I$, $\lambda(t^*) = 1$. Будем далее считать, что для каждой используемой в описанном выше алгоритме оценки ненулевой вектор l_0 удалось определить.

Выбор параметров $S(t)$ и $\lambda(t)$ обусловлен требованием касания эллипсоидальной оценки и множества разрешимости в направлении, заданном вектором $l(t)$ (т. е. согласно терминологии из [12, с. 129–130] искомые оценки должны быть “тугими”).

Справедливо следующее утверждение (см. теорему 3.10.1 из [12]):

Теорема 1. Для любого $t \leq 0$

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

Заметим, что матрицы конфигурации эллипсоидов $W_k(t)$ не зависят от $x^{(i)}$. То есть в основном алгоритме эти матрицы достаточно единожды подсчитать для разных значений $t \in [T_1, 0]$, $k = \overline{1, K}$, и далее использовать одни и те же матрицы на разных итерациях (для разных значений i). Кроме того, для центров эллипсоидальных оценок справедливы формулы

$$w(t) = e^{At}(x^{(i)}, q) + \nu(t), \quad \nu(t) = \int_0^t e^{A(t-\tau)} d\tau (Bp + f).$$

Здесь функцию $\nu(t)$ и матричную экспоненту e^{At} можно подсчитать (для разных $t \in [T_1, 0]$) один раз и далее многократно использовать для построения разных оценок на разных шагах работы основного алгоритма. Указанные замечания позволяют существенно уменьшить сложность вычислений.

Теперь можно построить *позиционное управление*, переводящее траекторию системы (1.1) из позиции (t, z) , $t < 0$, в целевое множество $x^{(i)} \times \text{Pr}_2 \Omega$ в момент времени $t = 0$. Оно может быть найдено за счет “прицеливания” [12, с. 133–137] на один из построенных эллипсоидов $\mathcal{E}(w(t), W_{k^*}(t))$:

$$u^*(t, z) = \begin{cases} p - \frac{PB^T l^0(t, z)}{\sqrt{\langle l^0(t, z), \mathbf{P}l^0(t, z) \rangle}}, & B^T l^0(t, z) \neq 0, \\ \mathcal{E}(p, P) & \text{иначе,} \end{cases} \quad (3.3)$$

где $l^0(t, z) = e^{-tA^T} l^*(t, z)$, $l^*(t, z)$ — максимизатор в выражении для расстояния от точки z до эллипсоидальной оценки множества разрешимости

$$h_+(t, z) = \max \left\{ \langle l, e^{-tA}(z - w(t)) \rangle - \sqrt{\langle l, e^{-tA}W_{k^*}(t)e^{-tA^T}l \rangle} \mid \|l\| \leq 1 \right\}.$$

В [9] было указано, как можно в некоторых важных частных случаях вычислять $l^0(t, z)$, не решая указанную выше задачу оптимизации для функции $h_+(t, z)$.

Теорема 2. Пусть $z(t)$, $t \in [T_1, 0]$, — любая из траекторий системы (1.1) с подставленным в нее позиционным управлением (3.3) при $z(T_1) \in \mathcal{E}(w(T_1), W_{k^*}(T_1))$. Тогда $z(0) \in \mathcal{E}(z^*, Z^*)$.

Доказательство. Заметим, что многозначное отображение (3.3) полунепрерывно сверху по (t, z) и принимает выпуклые, компактные значения. Следовательно, выполнены условия теоремы [13, теорема 1, с. 60–61] о существовании решения задачи Коши для полученного в результате подстановки в (1.1) управления (3.3) дифференциального включения. При этом в качестве начального условия можно взять любой вектор $z(T_1) \in \mathcal{E}(w(T_1), W_{k^*}(T_1))$, причем $h_+(T_1, z(T_1)) = 0$.

Рассмотрим произвольную точку $z \notin \text{int } \mathcal{E}(w(t), W_{k^*}(t))$ для некоторого $t \in (T_1, 0)$. Тогда максимизатор $l^*(t, z)$ определен единственным образом и управление $u^*(t, z)$ определено однозначно. Преобразуем выражение для полной производной от функции $h_+(t, z)$ вдоль траектории замкнутой системы, используя (3.1), (3.2)²:

$$\begin{aligned} \frac{dh_+(t, z)}{dt} &= \langle l^*, e^{-tA}(Az + Bu^* + f) \rangle + \langle l^*, -Ae^{-tA}(z - w) \rangle - \langle l^*, e^{-tA}\dot{w} \rangle \\ &\quad - \frac{\langle l^*, -Ae^{-tA}W_{k^*}e^{-tA^T}l^* \rangle + \langle l^*, e^{-tA}\dot{W}_{k^*}e^{-tA^T}l^* \rangle + \langle l^*, -e^{-tA}W_{k^*}A^Te^{-tA^T}l^* \rangle}{2\sqrt{\langle l^*, e^{-tA}W_{k^*}e^{-tA^T}l^* \rangle}} \\ &= -\langle l^*, e^{-tA}Bp \rangle + \frac{\langle l^*, e^{-tA}(W_{k^*}^{1/2}SP^{1/2} + P^{1/2}S^TW_{k^*}^{1/2})e^{-tA^T}l^* \rangle}{2\sqrt{\langle l^*, e^{-tA}W_{k^*}e^{-tA^T}l^* \rangle}} + \langle l^*, e^{-tA}Bu^* \rangle \\ &= -\langle l^*, e^{-tA}Bp \rangle + \frac{\langle e^{-tA^T}l^*, W_{k^*}^{1/2}SP^{1/2}e^{-tA^T}l^* \rangle}{\sqrt{\langle l^*, e^{-tA}W_{k^*}e^{-tA^T}l^* \rangle}} + \langle l^*, e^{-tA}Bu^* \rangle. \end{aligned}$$

Воспользуемся неравенством Коши — Буняковского

$$\langle e^{-tA^T}l^*, Pe^{-tA^T}l^* \rangle^{1/2} \geq \langle l^*, e^{-tA}We^{-tA^T}l^* \rangle^{-1/2} \langle e^{-tA^T}l^*, W^{1/2}SP^{1/2}e^{-tA^T}l^* \rangle,$$

а также формулой (3.3). В результате получим следующую оценку:

$$\begin{aligned} \frac{dh_+(t, z)}{dt} &\leq -\langle l^*, e^{-tA}Bp \rangle + \langle e^{-tA^T}l^*, Pe^{-tA^T}l^* \rangle^{1/2} + \langle l^*, e^{-tA}Bu^* \rangle \\ &= -\langle B^Tl^0, p \rangle + \langle l^0, Pl^0 \rangle^{1/2} + \langle B^Tl^0, u^* \rangle = 0. \end{aligned}$$

Поскольку $h_+(t, z) \geq 0$ для любых t, z и $h_+(t, z) = 0$ для любого $z \in \mathcal{E}(w(t), W_{k^*}(t))$, то из полученной выше оценки следует, что $h_+(0, z(0)) = 0$, т. е. $z(0) \in \mathcal{E}(z^*, Z^*)$. $\square$

Для каждой отобранной согласно (2.1) вершины v_j можно определить наименьшее время $t_{\min, j}$ перехода из v_j в целевое множество $x^{(i)} \times \text{Pr}_2 \Omega$, а также соответствующий номер эллипсоидальной оценки k^* : $v_j \in \mathcal{E}(w(t_{\min, j}), W_{k^*}(t_{\min, j}))$. Далее необходимо построить управление. Заметим, что формулу (3.3) нельзя непосредственно использовать в основном алгоритме, так как там речь идет о программном управлении, а не о позиционном. Однако переход от позиционного управления к программному всегда можно выполнить за счет решения вспомогательной задачи Коши для дифференциального включения

$$\begin{cases} \dot{z}(t) \in Az(t) + Bu^*(t, z) + f, \\ z(t_{\min, j}) = (v_j, y(v_j)), \end{cases} \quad (3.4)$$

где управление $u^*(\cdot)$ определено согласно (3.3). Затем необходимо выделить однозначный измеримый селектор из многозначного отображения $u^*(t, z^*(t))$, где $z^*(t)$ — какое-то из решений задачи (3.4).

²Здесь для краткости не указаны аргументы t, z используемых функций.

Для каждой построенной траектории $z^*(t)$, $t \in [t_{\min,j}, 0]$, далее должна быть выполнена проверка выполнения следующего условия (строка 14 алгоритма 1): $\Pr_1 z^*(t) \in \mathcal{X}_{free} \forall t \in [t_{\min,j}, 0]$. Согласно (1.3) проверка данного условия может быть сведена к решению задачи оптимизации

$$\min_{t \in [t_{\min,j}, 0]} \min_{i=1, M} \max_{r=1, s_i} \varphi_{i,r}(\Pr_1 z^*(t)) \geq 0.$$

Если указанные условия не выполнены, то траектория отбраковывается алгоритмом. Иначе она может быть использована для построения ребра, соединяющего новую вершину $x^{(i)}$ графа Γ со старой вершиной v_j . Если найдено несколько подходящих вершин v_j при разных j , то выбирается та из них (с номером j^*), для которой величина $t_{\min,j}$ максимальна. Заметим, что множество неотбракованных алгоритмом вершин графа может быть пустым — в этом случае и новая вершина $x^{(i)}$ в граф добавлена не будет, а алгоритм перейдет к следующему шагу и новой случайной точке $x^{(i+1)}$.

В алгоритме 2 используются внутренние эллипсоидальные оценки $\mathcal{E}(\tilde{w}(t), \tilde{W}_k(t))$ для множества достижимости $\mathcal{X}(t, 0, z_0)$, $z_0 = (x^{(i)}; y(x^{(i)}))$, $t \in [0, T_2]$. Формулы, позволяющие построить такие оценки, аналогичны выписанным выше формулам для случая со множествами разрешимости (см. [11, с. 218–220] и [12, с. 126–127]):

$$\begin{cases} \dot{\tilde{w}}(t) = A\tilde{w} + Bp + f, \\ \tilde{w}(0) = z_0, \end{cases}$$

$$\begin{cases} \dot{\tilde{W}}_k(t) = A\tilde{W}_k(t) + \tilde{W}_k(t)A^T + \tilde{W}_k^{1/2}(t)S(t)\mathbf{P}^{1/2} + \mathbf{P}^{1/2}S^T(t)\tilde{W}_k^{1/2}(t), \\ \tilde{W}_k(0) = \mathbb{O}_{n \times n}, \end{cases}$$

где $\mathbf{P} = BPB^T$, $\mathbb{O}_{n \times n}$ — нулевая матрица размера $n \times n$.

Каждая эллипсоидальная оценка зависит от некоторой кривой $l(t)$, являющейся решением следующей вспомогательной задачи Коши:

$$\begin{cases} \dot{l} = -A^T l, & t \in [0, T_2], \\ l(0) = l^*, \end{cases}$$

для произвольного вектора $l^* \in \mathbb{R}^n$: $\|l^*\| = 1$. Путем перебора различных значений l^* могут быть получены разные оценки (в основном алгоритме такие векторы $l^* = l_k$; при этом $\tilde{W}_k(t) = \tilde{W}(t, l_k)$). Будем далее считать, векторы l^* выбираются таким образом, что $\langle l^*, \mathbf{P}l^* \rangle \neq 0$.

Ортогональные матрицы $S(t)$ непрерывно зависят от t и удовлетворяют следующим равенствам:

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

Здесь $\mathbb{I}_{n \times n}$ — единичная матрица размера $n \times n$.

В алгоритме 2 (строка 15) необходимо найти управление $u^*(t)$, которое переводит начальную позицию $z_0 = z(0)$ в заданную точку $z_1 \in \mathcal{E}(\tilde{w}(t_1), \tilde{W}_k(t_1))$ для фиксированных k , $t_1 > 0$. Для этого сначала определим соответствующее позиционное управление:

$$u^*(t, z) = \begin{cases} p + \frac{PB^T l^0(t, z)}{\sqrt{\langle l^0(t, z), \mathbf{P}l^0(t, z) \rangle}}, & B^T l^0(t, z) \neq 0, \\ \mathcal{E}(p, P) & \text{иначе,} \end{cases} \quad (3.5)$$

где $l^0(t, z) = e^{-tA^T} l^*(t, z)$, $l^*(t, z)$ — максимизатор в выражении для расстояния от точки z до эллипсоидальной оценки множества достижимости

$$\tilde{h}_+(t, z) = \max \left\{ \langle l, e^{-tA}(z - \tilde{w}(t)) \rangle - \sqrt{\langle l, e^{-tA}\tilde{W}_k(t)e^{-tA^T}l \rangle} \mid \|l\| \leq 1 \right\}.$$

Теорема 3. Пусть $z(t)$, $t \in [0, t_1]$, — любая из траекторий системы (1.1) с подставленным в нее позиционным управлением (3.5) при $z(t_1) \in \mathcal{E}(\tilde{w}(t_1), \tilde{W}_k(t_1))$. Тогда $z(0) = z_0$.

Доказательство данного утверждения производится по аналогии с теоремой 2. Единственное отличие состоит в том, что полная производная функции $\tilde{h}_+(t, z)$ вне трубки достижимости будет неотрицательной, а не неположительной, как в случае с оценками множеств разрешимости.

Для получения программного управления из позиционного рассмотрим задачу Коши для дифференциального включения:

$$\begin{cases} \dot{z}(t) \in Az(t) + Bu^*(t, z) + f, & t \in [0, t_1], \\ z(t_1) = z_1, \end{cases} \quad (3.6)$$

где позиционное управление $u^*(\cdot)$ определено согласно (3.5). Далее необходимо выделить однозначный измеримый селектор $u^*(t)$ из многозначного отображения $u^*(t, z^*(t))$, где $z^*(t)$ — какое-то из решений задачи (3.6).

В строке 11 алгоритма 2 необходимо найти произвольную точку эллипсоида с фиксированной первой частью (первыми n_1 компонентами). Для решения этой подзадачи воспользуемся следующим утверждением.

Лемма 1. Пусть заданы некоторые векторы и матрицы вида

$$v \in \mathbb{R}^{n_1}, \quad w = \begin{pmatrix} w_1 \\ w_2 \end{pmatrix}, \quad W = \begin{pmatrix} W_{11} & W_{12} \\ W_{12}^T & W_{22} \end{pmatrix} > 0, \quad W^{-1} = \begin{pmatrix} V_{11} & V_{12} \\ V_{12}^T & V_{22} \end{pmatrix},$$

$w_1 \in \mathbb{R}^{n_1}$, $w_2 \in \mathbb{R}^{n_2}$, $W_{11} \in \mathbb{R}^{n_1 \times n_1}$. В пространстве $\mathbb{R}^{n_2}$ рассмотрим следующий эллипсоид:

$$E = \mathcal{E}\left(w_2 - V_{22}^{-1}V_{12}^T(v - w_1), (1 + \langle v - w_1, (V_{12}V_{22}^{-1}V_{12}^T - V_{11})(v - w_1) \rangle) V_{22}^{-1}\right).$$

Тогда для любого вектора $y \in E$ будет выполнено соотношение $(v; y) \in \mathcal{E}(w, W)$.

Доказательство. Из определения эллипсоида (1.2) следует, что

$$\langle v - w_1, V_{11}(v - w_1) \rangle \leq 1, \quad V_{22}^{-1} > 0,$$

а значит,

$$(1 + \langle v - w_1, (V_{12}V_{22}^{-1}V_{12}^T - V_{11})(v - w_1) \rangle) \geq 0.$$

Рассмотрим сначала особый случай, когда

$$\langle v - w_1, V_{11}(v - w_1) \rangle = 1 \quad \text{и} \quad \langle v - w_1, V_{12}V_{22}^{-1}V_{12}^T(v - w_1) \rangle = 0.$$

При этом эллипсоид E вырождается в единственную точку — его центр. Если $y = w_2 - V_{22}^{-1}V_{12}^T(v - w_1)$, то

$$\begin{aligned} \langle (v; y) - w, W^{-1}((v; y) - w) \rangle &= \langle v - w_1, V_{11}(v - w_1) \rangle \\ &+ 2 \langle v - w_1, V_{12}(y - w_2) \rangle + \langle y - w_2, V_{22}(y - w_2) \rangle = 1 - \langle v - w_1, V_{12}V_{22}^{-1}V_{12}^T(v - w_1) \rangle = 1. \end{aligned}$$

Используя определение (1.2), получим, что $(v; y) \in \mathcal{E}(w, W)$.

Пусть теперь $\langle v - w_1, V_{11}(v - w_1) \rangle < 1$ или $\langle v - w_1, V_{12}V_{22}^{-1}V_{12}^T(v - w_1) \rangle > 0$. Для любого $y \in E$ справедливо следующее неравенство:

$$\begin{aligned} \langle y - w_2 + V_{22}^{-1}V_{12}^T(v - w_1), V_{22}(y - w_2 + V_{22}^{-1}V_{12}^T(v - w_1)) \rangle \\ \leq 1 + \langle v - w_1, (V_{12}V_{22}^{-1}V_{12}^T - V_{11})(v - w_1) \rangle. \end{aligned}$$

Отсюда после несложных преобразований получаем

$$\begin{aligned} \langle (v; y) - w, W^{-1}((v; y) - w) \rangle &= \langle v - w_1, V_{11}(v - w_1) \rangle \\ &+ 2 \langle v - w_1, V_{12}(y - w_2) \rangle + \langle y - w_2, V_{22}(y - w_2) \rangle \leq 1. \end{aligned}$$

Снова, используя определение (1.2), имеем, что $(v; y) \in \mathcal{E}(w, W)$. $\square$

При использовании леммы 1 в алгоритме 2 следует положить $v = v_j$, $w = \tilde{w}(\tilde{t}_{\min, j})$, $W = \tilde{W}_{k*}(\tilde{t}_{\min, j})$, $y = y^*$. И далее необходимо определить какую-то точку $y = y^*$ из построенного эллипсоида E . Самый простой способ выбора подходящей точки y^* состоит в использовании центра эллипсоида E :

$$y^* = w_2 - V_{22}^{-1} V_{12}^T (v_j - w_1).$$

4. Оптимальное управление линейной системой с линейным функционалом

Описанные в предыдущих разделах методы поиска субоптимального в смысле быстродействия управления могут быть использованы и для решения более общей задачи оптимального управления с интегральным функционалом и линейной подынтегральной функцией (задачи 3). Опишем основные изменения приведенного выше алгоритма, необходимые для решения этой задачи.

Добавим в фазовый вектор системы (1.1) дополнительную переменную z_{n+1} , удовлетворяющую дифференциальному уравнению

$$\dot{z}_{n+1} = a^T z + b^T u + c.$$

Заметим, что переменная z_{n+1} не влияет на фазовые ограничения, а потому может быть добавлена именно ко второй части фазового вектора (а не к первой части из n_1 компонент). Расширенный фазовый вектор $\bar{z} = (x; \bar{y}) \in \mathbb{R}^{n+1}$ будет принимать значения из множества $\bar{\Omega} = \Omega \times \mathbb{R}$.

Далее для получившейся системы в $(n+1)$ -мерном пространстве можно сформулировать аналог описанного выше алгоритма поиска субоптимального управления. Подробно опишем здесь лишь те части нового алгоритма, которые отличают его от ранее разобранных (для задачи быстродействия).

- Для каждой вершины v графа Γ теперь нет смысла отдельно хранить оценку времени быстродействия $C(v)$. Однако, как и ранее, нужно запоминать вторую часть фазового вектора $\bar{y}(v)$, причем последняя компонента этого вектора — это оценка значения функционала качества $J(\cdot)$, соответствующая некоторой уже построенной траектории, которая соединяет начальную позицию z_0 и позицию $(v; y(v))$. Для удобства будем использовать следующие обозначения для выделения этой последней компоненты: $\bar{y}(v) = (y(v); h(v))$, $h(v) \in \mathbb{R}$.
- При вычислении эллипсоидальных оценок (строка 5 алгоритма 1) в пространстве $\mathbb{R}^{n+1}$ нужно построить эллипсоидальную аппроксимацию целевого множества $\mathcal{E}(z^*, Z^*)$, где

$$z^* = (x^{(i)}; q; 0), \quad Z^* = \text{diag}(\mathbb{O}_{n_1 \times n_1}, Q, 0)$$

и, как и ранее, $\mathcal{E}(q, Q)$ — внутренняя оценка множества $\text{Pr}_2 \Omega$.

Для эллипсоидальных оценок множества достижимости (см. строку 1 алгоритма 2) начальное множество является точкой $(x^{(i)}; y(x^{(i)}); 0)$.

- В строке 6 алгоритма 1 последняя, $(n+1)$ -я, компонента не должна влиять на отбор вершин v_j . То есть условие отбора (формирования множества $\mathcal{V}_0$) должно выглядеть теперь так:

$$\exists h \in \mathbb{R} : (v_j; y(v_j); h) \in \bigcup_{k, t} \mathcal{E}(w(t), W_k(t)) \cap \tilde{\mathbb{B}}_{r_i}(x^{(i)}).$$

- В строке 9 алгоритма 1 теперь нужно подсчитывать минимальное значение функционала $J_{\min,j}$, а также соответствующее ему время $t_{\min,j}$ согласно следующим формулам:

$$J_{\min,j} = \min_{t \in [T_1, 0]} \min_{k \in \overline{1, K}} \min \{ -h \in \mathbb{R} : (v_j; y(v_j); h) \in \mathcal{E}(w(t), W_k(t)) \},$$

$t = t_{\min,j}$ — минимизатор в этой формуле. Если матрица эллипсоида $W_k(t)$ не вырождена, то внутренний минимум (при фиксированных t, k) можно вычислить в явном виде, используя доказанную выше лемму 1:

$$h_{\min} = \hat{w} - \frac{p^T(\tilde{x} - \tilde{w}) + \sqrt{(p^T(\tilde{x} - \tilde{w}))^2 - \eta((\tilde{x} - \tilde{w})^T P(\tilde{x} - \tilde{w}) - 1)}}{\eta},$$

$$(W_k(t))^{-1} = \begin{pmatrix} P & p \\ p^T & \eta \end{pmatrix}, \quad P = P^T \in \mathbb{R}^{n \times n}, \quad w(t) = \begin{pmatrix} \tilde{w} \\ \hat{w} \end{pmatrix}, \quad \tilde{x} = \begin{pmatrix} v_j \\ y(v_j) \end{pmatrix}, \quad \hat{w} \in \mathbb{R}.$$

Аналогична ситуация и в случае с оценками множеств достижимости (см. строки 5 и 7 алгоритма 2):

$$\tilde{J}_{\min,j} = \min_{t \in [0, T_2]} \min_{k \in \overline{1, K}} \min \{ h \in \mathbb{R} : \exists y : (v_j; y; h) \in \mathcal{E}(\tilde{w}(t), \tilde{W}_k(t)) \}, \quad \tilde{t}_{\min,j} \text{ — минимизатор,}$$

$$\tilde{J}_{\min,j} = \min_{t \in [0, T_2]} \min_{k \in \overline{1, K}} \min \{ h \in \mathbb{R} : (v_j; y(v_j); h) \in \mathcal{E}(\tilde{w}(t), \tilde{W}_k(t)) \}, \quad \tilde{t}_{\min,j} \text{ — минимизатор.}$$

- Необходимо изменить условия для обработки очередной j -й вершины графа в строке 11 алгоритма 1:

$$(j^* < 0) \vee ((h(v_j) + J_{\min,j}) < (h(v_{j^*}) + J_{\min,j^*})).$$

- Необходимо изменить правила пересчета параметров каждой новой вершины графа (строка 17 алгоритма 1):

$$h(x^{(i)}) := h(v_{j^*}) + J_{\min,j^*}.$$

- В строке 11 алгоритма 2: y^* — любой вектор, для которого

$$(v_j; y^*; \tilde{J}_{\min,j}) \in \mathcal{E}(\tilde{w}(\tilde{t}_{\min,j}), \tilde{W}_{\tilde{k}^*}(\tilde{t}_{\min,j})).$$

В строке 14 неравенство теперь выглядит так:

$$\Delta_j = h(x^{(i)}) + \tilde{J}_{\min,j} - h(v_j) < 0.$$

В строке 17 вместо пересчета величин $C(v_j)$ теперь пересчитываются $h(v_j)$ по следующему правилу: $h(v_j) := h(v_j) + \Delta_j$.

- В строке 18 алгоритма 1, а также строках 13 и 18 алгоритма 2 изменяются только сопоставленные вершинам графа компоненты y из n_2 чисел без учета дополнительных величин h .

5. Примеры

5.1. Пример для задачи быстродействия

Для демонстрации работы приведенного выше алгоритма, а также для сравнения его с версией алгоритма, ранее описанной в [9] (т.е. без возможности учитывать фазовые ограничения по части переменных), рассмотрим следующий простой пример. Пусть $n_1 = n_2 = 2$, и система (1.1) выглядит следующим образом:

$$\begin{cases} \dot{x}_1 = y_1, \\ \dot{x}_2 = y_2, \\ \dot{y}_1 = u_1, \\ \dot{y}_2 = u_2. \end{cases} \quad z = (x_1, x_2, y_1, y_2)^T, \quad z(0) = \mathbb{O}_{4 \times 1}, \quad u_1^2 + u_2^2 \leq 10. \quad (5.1)$$

Кроме того, пусть $\Omega = [-5, 50] \times [-5, 30] \times [-6, 7] \times [-6, 6]$, $\mathcal{Z}_{goal} = [38, 42] \times [2, 6] \times [-6, 7] \times [-6, 6]$. Фазовые ограничения по первым n_1 переменным в виде лабиринта изображены на рис. 1 и 2

В табл. 1 приведены результаты сравнения двух алгоритмов, которые были получены следующим образом: для каждого $k = 1, 2, 3, 4$, (k — это номер серии экспериментов) и $j = 1, \dots, 10$, (j — это номер эксперимента в рамках конкретной серии) предлагаемый алгоритм строил дерево $\Gamma_{2j}(k) = (V_{2j}(k), E_{2j}(k))$: $|V_{2j}(k)| = 500 * k$, используя в качестве случайных точек $x^{(i)} = \text{Pr}_1 z^{(i)}$: $\text{Samples}_j = \{z^{(i)}\}_{i=1}^{N_j(k)}$ — последовательность, которая потребовалась алгоритму EKRRT* из [9] для построения дерева $\Gamma_{1j}(k) = (V_{1j}(k), E_{1j}(k))$: $|V_{1j}(k)| = 500 * k$.

Здесь для каждого алгоритма приведены: $t^*(k) = \frac{1}{10} \sum_{j=1}^{10} t_j^*(k)$, $t_j^*(k)$ — найденное

Т а б л и ц а 1

Усредненные по запускам результаты работы двух версий алгоритма EKRRT*

$ V $	500			1000			1500			2000		
	t^*	N	T	t^*	N	T	t^*	N	T	t^*	N	T
Γ_1	41	23240	699.55	38.01	40523	1727.13	37.23	58030	3001.13	36.58	76278	4453.97
Γ_2	21.85	689	299.72	20.98	1190	693.44	20.83	1691	1119.23	20.83	2192	1567.58

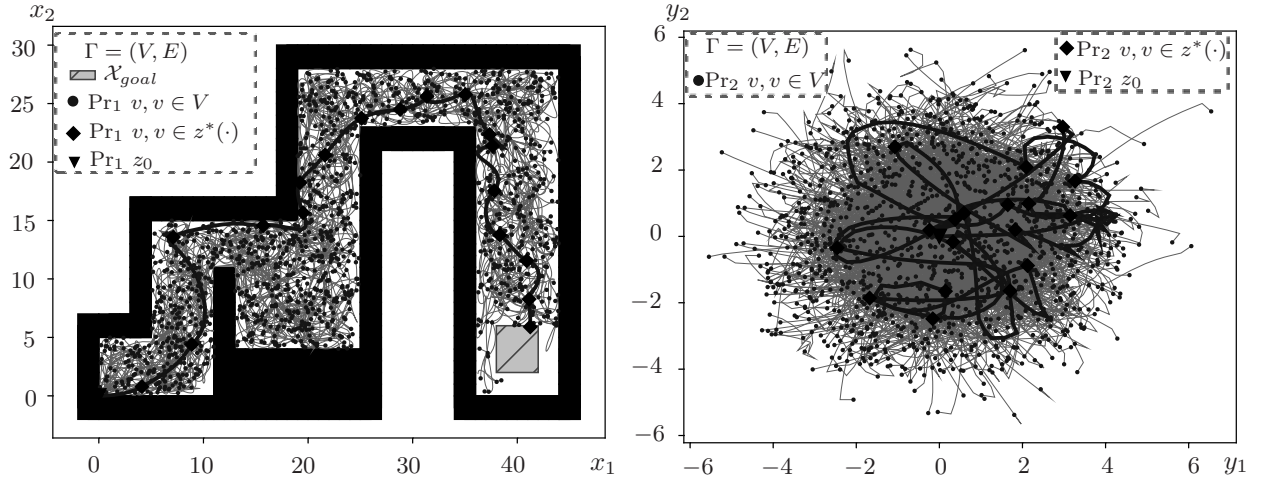


Рис. 1. Результаты работы EKRRT*.

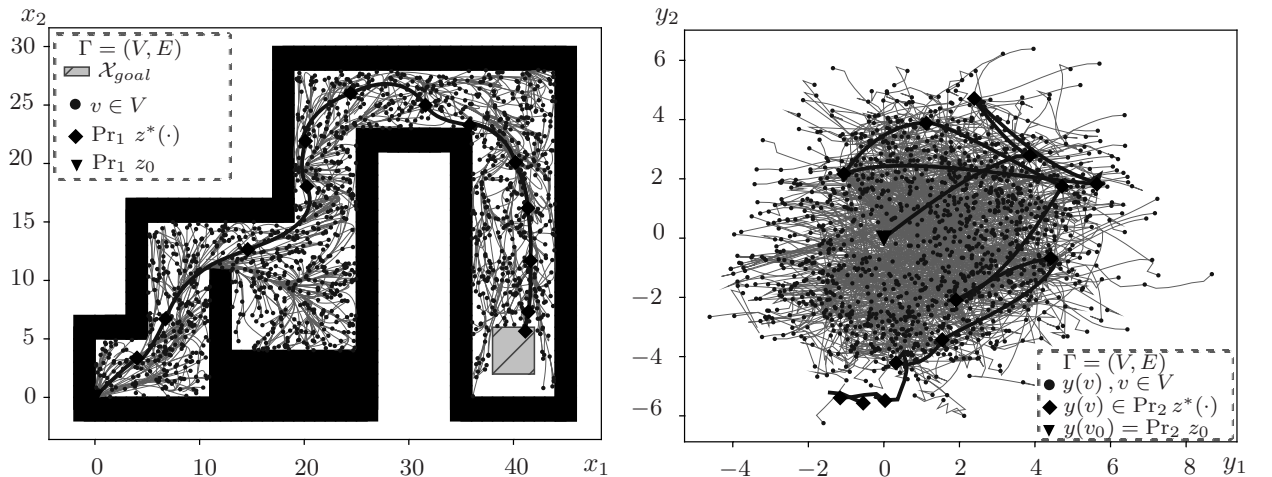


Рис. 2. Результаты работы EKRRT* с фазовыми ограничениями по части переменных.

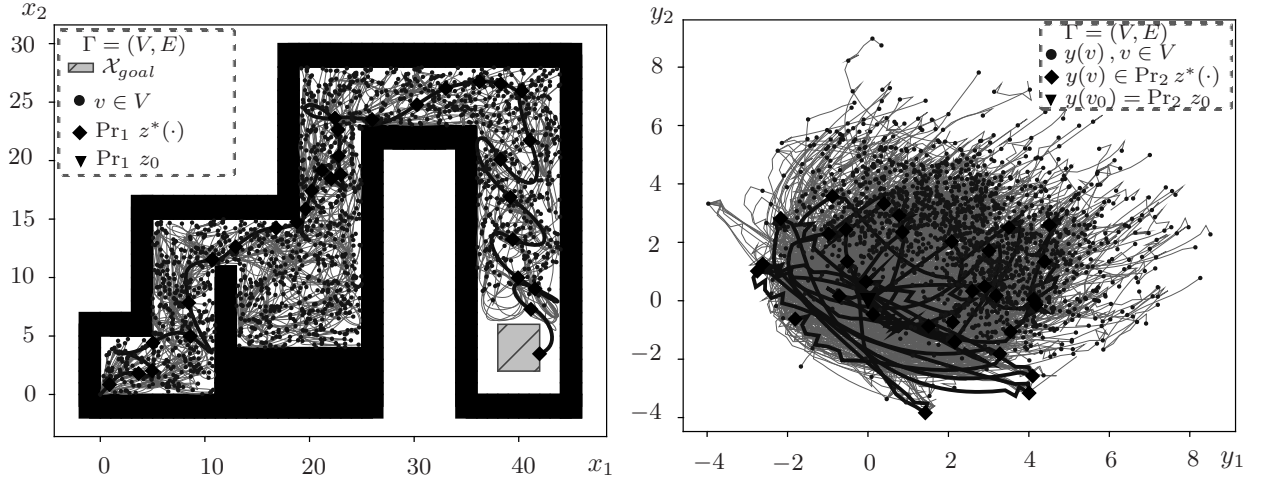


Рис. 3. Результаты работы EKRRT* с фазовыми ограничениями по части переменных (задача с интегральным функционалом).

время быстрогодействия; $N(k) = \frac{1}{10} \sum_{j=1}^{10} N_j(k)$, $N_j(k)$ — количество случайных точек, потребовавшихся для построения соответствующего дерева (частично точки отбраковывались алгоритмами и не образовывали вершин деревьев); $T(k) = \frac{1}{10} \sum_{j=1}^{10} T_j(k)$, $T_j(k)$ — время работы алгоритма в секундах.

Конечно, время вычислений зависит от конкретной программной реализации, но проведенные вычисления позволяют говорить о том, что в данном примере второй алгоритм (разобранный в данной работе) работает в несколько раз быстрее, чем первый (из [9]). Все вычисления для примеров в данной работе были реализованы на языке программирования Python3; для вычислений использовался чип Apple M1 с тактовой частотой 3.2 ГГц и 8 Гб оперативной памяти.

На рис. 1 и 2 представлены примеры работы алгоритмов для $k = 3$, $i = 5$. Толстой черной кривой выделена найденная субоптимальная траектория.

5.2. Пример для задачи с линейным функционалом

Используем систему (5.1) и зададим в (1.4) $a = \mathbb{O}_{4 \times 1}$, $b = (-10, -10)^T$, $c = 1$. Также используем набор из $l_1, \dots, l_K$, $l_k \in \mathbb{R}^5$, $K = 10$, векторов.

На рис. 3 продемонстрированы результаты работы описанного в данной работе алгоритма EKRRT* с фазовыми ограничениями по части переменных для этой задачи при $|V| = 2000$.

Было получено значение функционала $J^* = -39.21$, для чего алгоритму потребовалось сгенерировать 2275 случайных точек и 1897 с.

Заключение

Рассмотренные в работе примеры и проведенные вычислительные эксперименты говорят о принципиальной возможности использования предложенных алгоритмов для решения достаточно широкого класса задач управления линейными системами с фазовыми ограничениями. Эти методы могут быть также применены для систем с нелинейной динамикой за счет локальной линеаризации, как это было сделано в [9], но с учетом новых идей, изложенных в данной работе. Остается открытым вопрос о вероятностной полноте и асимптотической оптимальности [3] разработанных алгоритмов. Исследованию необходимых и достаточных условий для выполнения указанных свойств будут посвящены дальнейшие работы авторов.

СПИСОК ЛИТЕРАТУРЫ

1. Казаков К.А., Семенов В.А. Обзор современных методов планирования движения // Труды ИСП РАН. 2016. Т. 28, № 4. С. 241–294. [https://doi.org/10.15514/ISPRAS-2016-28\(4\)-14](https://doi.org/10.15514/ISPRAS-2016-28(4)-14)
2. Paden B., Cap M., Yong S.Z., Yershov D., Frazzoli E. A Survey of Motion Planning and Control Techniques for Self-Driving Urban Vehicles // IEEE Trans. Intell. Veh. 2016. Vol. 1. no. 1. P. 33–55. <https://doi.org/10.1109/TIV.2016.2578706>
3. Karaman S., Frazzoli E. Sampling-based algorithms for optimal motion planning // The Inter. J. Robot. Res. 2011. Vol. 30. no. 7. P. 846–894. <https://doi.org/10.1177/0278364911406761>
4. Shkolnik A., Walter M., Tedrake R. Reachability-guided sampling for planning under differential constraints // Proc. 2009 IEEE Int. Conf. on Robotics and Automat. (ICRA). 2009. P. 2859–2865. <https://doi.org/10.1109/ROBOT.2009.5152874>
5. Xie C., van der Berg J., Patil S., Abbeel P. Toward asymptotical optimal motion planning for kinodynamic systems using a two-point boundary value problem solver // Proc. 2015 IEEE Int. Conf. on Robotics and Automat. (ICRA). 2015. P. 4187–4194. <https://doi.org/https://doi.org/10.1109/ICRA.2015.7139776>.
6. Webb D.J., van der Berg J. Kinodynamic RRT*: Asymptotically optimal motion planning for robots with linear dynamics // Proc. 2013 IEEE Conf. on Robotics and Automat. (ICRA). 2013. P. 5054–5061. <https://doi.org/10.1109/ICRA.2013.6631299>
7. Karaman S., Frazzoli E. Optimal kinodynamic motion planning using incremental sampling-based methods // Proc. of the 49th IEEE Conference on Decision and Control. 2010. P. 7681–7687. <https://doi.org/10.1109/CDC.2010.5717430>
8. LaValle S.M., Kuffner J.J. Randomized kinodynamic planning // Int. J. Robotics Res. 2001. Vol. 20, no. 5. P. 378–400. <https://doi.org/10.1177/02783640122067453>
9. Точилин П.А., Паршиков М.В. Об использовании методов эллипсоидального оценивания в алгоритме поиска субоптимальных путей RRT* // Автоматика и телемеханика. 2024 № 2. С. 60–80. <https://doi.org/10.31857/S0005231024020041>.
10. Zheng D., Tsiotras P. Accelerating kinodynamic RRT* through dimensionality reduction // Proc. 2021 IEEE/RSJ Int. Conf. on Intell. Robots and Systems (IROS). 2021. P. 3674–3680. <https://doi.org/10.1109/IROS51168.2021.9636754>
11. Kurzhanski A.B., Varaiya P. On ellipsoidal techniques for reachability analysis. Part II: internal approximations, box-valued constraints // Optimiz. Methods and Software. 2002. Vol. 17. P. 207–237. <https://doi.org/10.1080/1055678021000012435>
12. Kurzhanski A.B., Varaiya P. Dynamics and control of trajectory tubes. Theory and computation. Birkhäuser, 2014. 445 p. <https://doi.org/10.1007/978-3-319-10277-1>
13. Филиппов А.Ф. Дифференциальные уравнения с разрывной правой частью. М.: Наука, 1985. 224 с.

Поступила 29.01.2025

После доработки 26.02.2025

Принята к публикации 3.03.2025

Опубликована онлайн 28.03.2025

Точилин Павел Александрович
канд. физ.-мат. наук, доцент
факультет вычислительной математики и кибернетики
МГУ имени М. В. Ломоносова, г. Москва
e-mail: tochilin@cs.msu.ru

Паршиков Мирон Вячеславович
аспирант
факультет вычислительной математики и кибернетики
МГУ имени М. В. Ломоносова, г. Москва
e-mail: miron232734@gmail.com

REFERENCES

1. Kazakov K.A., Semenov V.A. An overview of modern methods for motion planning. *Proc. Inst. System Progr. RAS (Proc. of ISP RAS)*, 2016, vol. 28, no. 4, pp. 241–294 (in Russian). [https://doi.org/10.15514/ISPRAS-2016-28\(4\)-14](https://doi.org/10.15514/ISPRAS-2016-28(4)-14)
2. Paden B., Cap M., Yong S.Z., Yershov D., Frazzoli E. A survey of motion planning and control techniques for self-driving urban vehicles. *IEEE Trans. Intell. Veh.*, 2016, vol. 1, no. 1, pp. 33–55. <https://doi.org/10.1109/TIV.2016.2578706>
3. Karaman S., Frazzoli E. Sampling-based algorithms for optimal motion planning. *The Inter. J. Robot. Res.*, 2011, vol. 30, no. 7, pp. 846–894. <https://doi.org/10.1177/0278364911406761>
4. Shkolnik A., Walter M., Tedrake R. Reachability-guided sampling for planning under differential constraints. In: *Proc. 2009 IEEE Int. Conf. on Robotics and Automat. (ICRA)*, 2009, pp. 2859–2865. <https://doi.org/doi:10.1109/ROBOT.2009.5152874>
5. Xie C., van der Berg J., Patil S., Abbeel P. Toward asymptotical optimal motion planning for kinodynamic systems using a two-point boundary value problem solver. In: *Proc. 2015 IEEE Int. Conf. on Robotics and Automat. (ICRA)*, 2015, pp. 4187–4194. <https://doi.org/https://doi.org/10.1109/ICRA.2015.7139776>
6. Webb D.J., van der Berg J. Kinodynamic RRT*: Asymptotically optimal motion planning for robots with linear dynamics. In: *Proc. 2013 IEEE Int. Conf. on Robotics and Automat., (ICRA)*, 2013, pp. 5054–5061. <https://doi.org/10.1109/ICRA.2013.6631299>
7. Karaman S., Frazzoli E. Optimal kinodynamic motion planning using incremental sampling-based methods. In: *Proc. 49th IEEE Conf. on Decision and Control (CDC)*, 2010, pp. 7681–7687. <https://doi.org/10.1109/CDC.2010.5717430>
8. LaValle S.M., Kuffner J.J. Randomized kinodynamic planning. *Int. J. Robotics Res.*, 2001, vol. 20, no. 5, pp. 378–400. <https://doi.org/10.1177/02783640122067453>
9. Tochilin P.A., Parshikov M.V. On the use of ellipsoidal estimation techniques in the RRT* suboptimal pathfinding algorithm. *Autom. Rem. Contr.*, 2024, vol. 85, no. 2, pp. 144–159. <https://doi.org/10.31857/S0005117924020042>
10. Zheng D., Tsiotras P. Accelerating kinodynamic RRT* through dimensionality reduction. In: *Proc. 2021 IEEE/RSJ Int. Conf. on Intell. Robots and Systems (IROS)*, 2021, pp. 3674–3680. <https://doi.org/10.1109/IROS51168.2021.9636754>
11. Kurzhanski A.B., Varaiya P. On ellipsoidal techniques for reachability analysis. Part II: internal approximations, box-valued constraints. *Optimiz. Methods and Software*, 2002, vol. 17, pp. 207–237. <https://doi.org/10.1080/1055678021000012435>
12. Kurzhanski A., Varaiya P. *Dynamics and control of trajectory tubes: theory and computation*. Boston, Birkhäuser, 2014, 445 p. <https://doi.org/10.1007/978-3-319-10277-1>
13. Filippov A.F. *Differential equations with discontinuous right-hand sides*. Springer Dordrecht, 1988, 304 p. <https://doi.org/10.1007/978-94-015-7793-9>. Original Russian text published in Filippov A.F. *Differentsial'nyye uravneniya s razryvnoy pravoy chast'yu*, Moscow, Nauka Publ., 1985, 225 p.

Received January 29, 2025

Revised February 26, 2025

Accepted March 3, 2025

Published online March 28, 2025

Funding Agency: The paper was published with the financial support of the Ministry of Education and Science of the Russian Federation as part of the program of the Moscow Center for Fundamental and Applied Mathematics under the agreement no. 075-15-2022-284.

Pavel Aleksandrovich Tochilin, Cand. Sci. (Phis.-Math), Faculty of Computational Mathematics and Cybernetics of the Lomonosov Moscow State University, Moscow, 119991 Russia, e-mail: tochilin@cs.msu.ru.

Miron Vyacheslavovich Parshikov, PhD student, Faculty of Computational Mathematics and Cybernetics of the Lomonosov Moscow State University, Moscow, 119991 Russia, e-mail: miron232734@gmail.com.

Cite this article as: P. A. Tochilin, M.V. Parshikov. On the construction of suboptimal trajectories for a linear control system with state constraints on part of variables. *Trudy Instituta Matematiki i Mekhaniki UrO RAN*, 2025, <https://doi.org/10.21538/0134-4889-2025-31-2-fon-03>