

© 2024 г. П.А. ТОЧИЛИН, канд. физ.-мат. наук (tochilin@cs.msu.ru)
(Московский государственный университет им. М.В. Ломоносова;
Институт проблем управления им. В.А. Трапезникова РАН, Москва),
М.В. ПАРШИКОВ (miron232734@gmail.com)
(Московский государственный университет им. М.В. Ломоносова)

ОБ ИСПОЛЬЗОВАНИИ МЕТОДОВ ЭЛЛИПСОИДАЛЬНОГО ОЦЕНИВАНИЯ В АЛГОРИТМЕ ПОИСКА СУБОПТИМАЛЬНЫХ ПУТЕЙ RRT*¹

Статья посвящена разработке алгоритма приближенного решения задачи быстрогодействия для системы обыкновенных дифференциальных уравнений при условии огибания неподвижных препятствий и при выполнении заданных поточечных ограничений на возможные значения управляющих параметров. Основная идея состоит в использовании модификации алгоритма поиска субоптимальных путей при помощи быстрорастущих случайных деревьев (RRT*). Наиболее сложная часть этого алгоритма состоит в поиске оптимальных траекторий для задач перевода системы из одной фиксированной позиции в другую, близкую к ней, без учета фазовых ограничений. Эту подзадачу предлагается решать при помощи методов эллипсоидального исчисления. Такой подход позволяет достаточно эффективно искать субоптимальные траектории как для линейных систем с большой размерностью фазового пространства, так и для систем с нелинейной динамикой. Последовательно разобраны алгоритмы как для линейного, так и для нелинейного случая. Приведены соответствующие примеры вычислений.

Ключевые слова: системы управления, задача быстрогодействия, планирование движения, эллипсоидальное оценивание.

DOI: 10.31857/S0005231024020041, **EDN:** UHSVVV

1. Введение

В данной работе рассматривается задача перевода управляемого объекта из начальной позиции в заданное целевое множество за наименьшее возможное время, с огибанием препятствий. Такого рода задачи возникают при управлении автономным (беспилотным) движением на плоскости или в пространстве, при разработке алгоритмов управления роботами или манипуляторами. Как правило, точное решение такой задачи является очень трудоем-

¹ Работа выполнена при финансовой поддержке Российского научного фонда (проект № 22-11-00042).

ким с точки зрения вычислений из-за большой размерности вектора фазовых переменных, нелинейной динамики и сложной структуры препятствий. Поэтому в последние десятилетия активно развиваются разнообразные приближенные методы [1, 2], позволяющие строить субоптимальные траектории за малое время (в идеале — в режиме реального времени, что необходимо во многих приложениях).

Один из классов упомянутых методов основывается на использовании «быстрорастущих» случайных графов или, в частном случае, деревьев. Такого рода алгоритмы в англоязычной литературы получили общее обозначение PRM — Probabilistic Roadmap (в случае графов с возможными циклами) и RRT — Rapidly Exploring Random Tree (в случае деревьев). Основная идея здесь состоит в построении последовательности случайных точек фазового пространства и соединении их в граф. Количество таких точек N предполагается очень большим. Среди них содержится начальная позиция. Ребра искомого графа должны быть построены в соответствии с требованием обхода препятствий. На текущий момент накоплен достаточно большой опыт построения таких графов с различными свойствами. Как правило, эти методы различаются эвристиками, используемыми при случайном добавлении каждой новой вершины в граф, а также при соединении новой вершины с ранее добавленными при помощи ребер. Так, в [3] рассмотрены различные модификации методов случайных графов и случайных деревьев. В частности, проанализировано свойство полноты, под которой понимается тот факт, что вероятность попадания некоторого пути на графе из начальной точки в целевое множество стремится к 1 при $N \rightarrow \infty$.

Следующим шагом является добавление в алгоритмы построения случайных графов критерия качества путей (например, длины пути или времени движения по нему). Соответствующие семейства алгоритмов обозначают через PRM* и RRT*. В [3] для определенных разновидностей метода RRT* доказано свойство асимптотической оптимальности: если существует оптимальная траектория, то вероятность того, что значение функционала качества для построенного субоптимального пути на графе стремится к своему минимальному значению при $N \rightarrow \infty$, равна 1. Таким образом, если задача перевода объекта из начальной позиции в целевое множество разрешима, то метод RRT* даст ее субоптимальное решение, сколь угодно близкое к оптимальному, если число N достаточно велико.

В [3] рассмотрены лишь случаи простейших движений по ломаным в фазовом пространстве, где не учитываются ограничения, связанные с движением по траекториям конкретных дифференциальных уравнений. В многочисленных более поздних работах были предприняты попытки учитывать такого рода ограничения. Получившиеся методы обычно носят название Kinodynamic RRT* (далее будем использовать обозначение KRRT*). Ключевой подзадачей здесь является проблема построения оптимальной (или хотя бы субоптимальной) траектории дифференциального уравнения для перевода объекта из точки в точку на малом отрезке времени (локально).

От метода решения этой подзадачи существенно зависит качество полученных в итоге субоптимальных траекторий уже на большом отрезке времени (глобально).

Один из возможных подходов к решению указанной подзадачи оптимального управления на малом отрезке времени состоит в использовании принципа максимума Л.С. Понтрягина в задаче с фазовыми ограничениями [4, 5].

В [6] предпринята попытка построения при помощи модификации метода RRT* траекторий, являющихся решением системы линейных дифференциальных уравнений. Части таких траекторий получаются в результате решения вспомогательных линейно-квадратичных подзадач оптимального управления с функционалом специального, упрощенного вида. Серьезным недостатком данного подхода является невозможность учитывать поточечные ограничения на управляющие параметры, а также корректно обобщить предложенный подход на случай нелинейной динамики.

В [7] предложена модификация метода RRT*, в которой части искомого пути составлены из «простейших» траекторий исследуемой нелинейной системы дифференциальных уравнений. При этом наиболее сложная часть алгоритма, состоящая в построении оптимальных траекторий при поточечных ограничениях на управления, рассмотрена лишь для трех конкретных примеров.

В [8] предложен способ построения субоптимальных траекторий методом RRT* при помощи перебора по конечному набору постоянных управлений или за счет случайного выбора таких управлений. Предложенный подход может быть эффективен лишь в частных случаях, при малых размерностях вектора управляющих параметров.

Многочисленные работы посвящены также ускорению работы методов типа RRT* за счет изменений алгоритмов добавления новых вершин в графы. Например, в [9] предложен подход, связанный с построением оценок множеств достижимости для исследуемой системы дифференциальных уравнений в малых окрестностях вершин случайного графа. Эти оценки могут быть использованы для отсекаания новых вершин, которые будут заведомо лишними, т.е. не позволят случайному графу значительно продвигаться в сторону целевого множества. К сожалению, авторами работы так и не был предложен конкретный общий метод построения таких оценок, рассмотрены лишь простейшие примеры. При этом задача синтеза управлений на ребрах графа, как и в упомянутой выше работе [8], решена примитивным образом, за счет перебора конечного набора управляющих параметров.

Таким образом, можно утверждать, что разработка модификаций алгоритма RRT*, учитывающих одновременно нелинейную динамику и поточечные ограничения на управления, является актуальной и нерешенной проблемой. В данной работе предложен один возможный подход к решению этой задачи для широкого класса систем. Для этого модифицирован метод RRT* с дополнительным использованием алгоритмов эллипсоидального оценива-

ния [10, 11], которые позволяют эффективно решать локальные подзадачи о переводе системы из точки в точку. Решение такой подзадачи включает: 1) построение совокупности внутренних эллипсоидальных оценок множества разрешимости рассматриваемой управляемой системы; 2) вычисление позиционного управления, решающего подзадачу и полученного за счет «прицеливания» на эллипсоидальные оценки; 3) получение искомого программного управления из построенной позиционной стратегии.

2. Задача управления для линейной системы

В пространстве $\mathbb{R}^n$, $n \geq 2$, рассмотрим некоторое компактное множество Ω . Рассмотрим движение управляемого объекта, описываемое системой линейных дифференциальных уравнений

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

Предположим, что начальное состояние $x(0) = x_0 \in \Omega$ фиксировано. Матрицы $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$, а также вектор $f \in \mathbb{R}^n$ предполагаются независимыми от t . 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 $P \in \mathbb{R}^{m \times m}$, $P = P^T \geq 0$:

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

Если $P > 0$, то можно использовать эквивалентное определение

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

Через $\mathcal{U}$ обозначим класс *допустимых программных управлений*, который содержит все возможные кусочно-непрерывные функции $u = u(t) \in \mathcal{P}$, $t \geq 0$. Через $x(t, 0, x_0)|_{u(\cdot)}$ обозначим траекторию (1), порожденную некоторым допустимым управлением $u(\cdot)$, выпущенную в начальный момент времени $t_0 = 0$ из начальной позиции x_0 . Будем далее рассматривать только части траекторий $x(t, 0, x_0)|_{u(\cdot)}$, которые содержатся в области Ω .

Также в области Ω выделим некоторый набор множеств $\mathcal{M}_i$, $i = 1, \dots, M$, – препятствий, которые необходимо учитывать при движении объекта. Предположим, что каждое из этих множеств задается совокупностью неравенств

$$(2) \quad \mathcal{M}_i = \{x \in \Omega : \varphi_{i,j}(x) \leq 0, \quad j = 1, \dots, s_i\},$$

где непрерывные функции $\varphi_{i,j}(x)$, а также натуральные числа s_i заданы. Предполагается, что положение препятствий $\mathcal{M}_i$ не меняется со временем, т.е. функции $\varphi_{i,j}$ не зависят от t . Введем следующее обозначение:

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

Предположим, что $x_0 \in \mathcal{X}_{free}$. Зафиксируем также некоторое компактное множество $\mathcal{X}_{goal} \subset \mathcal{X}_{free}$, причем $x_0 \notin \mathcal{X}_{goal}$ и $\mu(\mathcal{X}_{goal}) > 0$, где $\mu(\mathcal{X}_{goal})$ – лебегова мера множества $\mathcal{X}_{free}$.

Теперь можно сформулировать **основные задачи управления**, решаемые в данной работе:

1) Необходимо найти такое $u(\cdot) \in \mathcal{U}$, для которого найдется $t_1 > 0$ и такая траектория $x(t, 0, x_0)|_{u(\cdot)}$ дифференциального уравнения (1), для которой

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

2) Среди управлений и соответствующих траекторий, найденных в предыдущем пункте, необходимо найти такие, для которых величина t_1 будет наименьшей.

Заметим, что множество $\mathcal{X}_{free}$ замкнуто, а значит, в данной постановке задач допускается движение вдоль границ препятствий.

3. Алгоритм построения случайного дерева

Описанные выше задачи управления будем решать приближенно, при помощи модификации алгоритма KRRТ*. Ниже приведена его общая схема. Основной целью алгоритма является построение ориентированного дерева $\Gamma = (V, E)$, где V – множество вершин, E – множество дуг. Каждой вершине $v \in V$ будет сопоставлено минимальное найденное время ее достижения $C(v)$ из начальной точки x_0 , являющейся корнем дерева. Для каждой вершины $v \neq x_0$ через $parent(v)$ обозначим ту вершину, для которой $(parent(v), v) \in E$.

Алгоритм является итерационным, где i – номер итерации, а фиксированное число N – максимальное количество итераций. На каждом шаге алгоритма должны быть построены вспомогательные эллипсоидальные оценки множества разрешимости $\mathcal{E}(w(t), W_k(t))$, $k = 1, \dots, K$. Количество оценок K считается фиксированным. Каждая оценка зависит от вектора параметров $l_k \in \mathbb{R}^n$ и определена при $t \in [T, 0]$, где число $T < 0$ фиксировано. Выбор параметра $T < 0$ здесь обусловлен стационарностью системы (1) и удобством фиксации нулевого конечного момента времени. Для итогового управления и траектории далее будет сделана замена переменной, и в результате эти функции будут определены уже при $t \geq 0$. Формулы для вычисления оценок будут приведены ниже. В приведенном далее алгоритме используются два вспомогательных множества вершин графа $\mathcal{V}_0$ и $\mathcal{V}_1$.

Algorithm 1 Ellipsoidal KRRT*

```
1:  $V := \{x_0\}$ ,  $C(x_0) := 0$ ,  $E := \emptyset$ 
2: формируется совокупность векторов  $l_k$ ,  $l_k \in \mathbb{R}^n$ ,  $\|l_k\| = 1$ ,  $k = 1, \dots, K$ 
3: for  $i = 1, \dots, N$  do
4:   генерируется случайная точка  $x^{(i)} \in \mathcal{X}_{free}$ 
5:   строятся эллипсоидальные оценки  $\mathcal{E}(w(t), W_k(t))$ ,  $k = 1, \dots, K$ ,  $t \in [T, 0]$ 
6:   формируется множество вершин  $\mathcal{V}_0 = V \cap (\cup_k \cup_t \mathcal{E}(w(t), W_k(t)) \cap \mathbb{B}_{r_i}(x^{(i)}))$ 
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:     $k^* :=$  соответствующий номер эллипсоидальной оценки (максимизатор)
11:    определяется оптимальное управление  $u^*(j, k^*, t)$ ,  $t \in [t_{\min,j}, 0]$ 
12:    строится траектория  $x^*(j, k^*, t)$ ,  $t \in [t_{\min,j}, 0]$ ,  $x^*(j, k^*, t_{\min,j}) = v_j$ 
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:          для каждой вершины  $v'$  поддерева с корнем  $v_j$   $C(v') := C(v') + \Delta_j$ 
22:           $E := E \setminus \{\text{parent}(v_j), v_j\}$ 
23:           $E := E \cup \{(x^{(i)}, v_j)\}$ 
```

В строке 4 координаты каждой новой точки $x^{(i)}$ являются случайными, соответствующими равномерному распределению на множестве Ω . В строке 6 алгоритма производится отбор вершин v_j , удовлетворяющих следующему условию:

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

Здесь $r_i > 0$ – параметр, ограничивающий перебор вершин графа и отвечающий за уменьшение итоговой сложности вычислений. Подробнее о нем будет сказано далее.

При построении управления и траектории в строках 11–12 используется алгоритм «прицеливания» на конкретную эллипсоидальную трубку, который будет описан ниже. Именно из-за этого итоговые функции $x^*(\cdot)$ и $u^*(\cdot)$ зависят от номера используемой эллипсоидальной оценки. Функция $x^*(j, k^*, t)$, $t \in [t_{\min,j}, 0]$, является решением задачи Коши для системы (1), после подстановки управления $u^*(j, k^*, t)$, с краевым условием $x^*(j, k^*, t_{\min,j}) = v_j$.

Инструкции основного алгоритма в строках 19–23 соответствуют известному (см. [3]) из описания алгоритма RRT* правилу «пересвязывания» вершин дерева. Благодаря этим действиям в алгоритме RRT* достигается свойство асимптотической оптимальности.

В указанный выше алгоритм может быть добавлено дополнительное условие выхода из внешнего цикла при первом достижении целевого множества, т.е. если на очередной итерации с номером i выполнено условие $x^{(i)} \in \mathcal{X}_{goal}$. Это условие необходимо при решении первой из двух описанных выше основных задач управления.

Замечание 1. Условие $\mu(\mathcal{X}_{goal}) > 0$ существенно для предлагаемого алгоритма. В случае, когда целевое множество является одноточечным ($\mathcal{X}_{goal} = \{x_{goal}\}$), алгоритм может быть модифицирован за счет добавления инструкций для построения эллипсоидальных оценок множества разрешимости, выпущенных из целевой точки, и для последующих действий, аналогичных строкам 6–18 с заменой $x^{(i)}$ на x_{goal} .

Для простоты изложения в приведенной выше схеме алгоритма для ребер графа не указана сопоставленная с ними дополнительная информация. Однако для решения основных задач управления понадобится дополнительно запоминать для каждого ребра программное управление и соответствующую часть траектории, которые были подсчитаны в строках 11–12 алгоритма для соответствующих значений индексов i, j . Более того, в процедуре пересвязывания графа необходимо делать дополнительную замену переменной t для того, чтобы проходить ранее построенную траекторию в обратном направлении. При этом существенно, что исходная система (1) является стационарной.

После построения дерева Γ можно решить сформулированные выше основные задачи управления:

- Если $\exists v_j \in V, v_j \in \mathcal{X}_{goal}$, то задача 1 разрешима.
- Если выполнено условие из предыдущего пункта, то оценка времени бысродействия может быть найдена следующим образом:

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

Пусть v^* – минимизатор в (4).

- Оптимальное управление $u^*(t)$ теперь может быть составлено из частей, сопоставленных отдельным ребрам графа. При этом нужно двигаться из вершины v^* к корню дерева, по дугам

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

учитывая обратный ход времени при объединении частей функции $u^*(t)$.

4. Эллипсоидальные оценки и закон управления

Основное отличие алгоритма, применяемого в данной работе, от известных вариантов алгоритма KRRT* состоит в использовании внутренних оценок

множеств разрешимости [12] системы (1), выпущенных из каждой очередной случайно подобранной точки $x^* \in \mathcal{X}_{free}$ (в схеме алгоритма это точки $x^{(i)}$).

Для фиксированного множества $\mathcal{X}_1 \subset \Omega$, для некоторых $t \leq t_1$ множество разрешимости $\mathcal{W}(t, t_1, \mathcal{X}_1)$ системы (1) состоит из всех возможных точек $x_0 \in \Omega$, для каждой из которых существует управление $u(\cdot) \in \mathcal{U}$, для которого $x(t_1, t, x_0)|_{u(\cdot)} \in \mathcal{X}_1$. В данной работе представляют интерес множества разрешимости из одноточечного целевого множества с нулевым конечным моментом времени: $\mathcal{W}(t, 0, x^*)$, $x^* \in \mathbb{R}^n$. Задача построения подобных множеств даже для линейной стационарной системы является нетривиальной. Одним из наиболее эффективных подходов является использование эллипсоидальных оценок множеств разрешимости: внешних или внутренних, в зависимости от конкретной решаемой задачи. Здесь основной целью является поиск управления, а для его построения удобно использовать внутренние оценки.

При заданном $x^* \in \mathbb{R}^n$, используя результаты из [10], определим семейство эллипсоидальных оценок $\mathcal{E}(w(t), W(t))$, $t \leq 0$, задаваемых следующими дифференциальными уравнениями:

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

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

Здесь $\mathbb{O}_{n \times n}$ – нулевая матрица размера $n \times n$, $\mathbf{P} = BPB^T$, а $S(t)$ – ортогональная матрица, непрерывно зависящая от t , для которой

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

Здесь $\mathbb{I}_{n \times n}$ – единичная матрица размера $n \times n$. Выбор параметров $S(t)$ и $\lambda(t)$ обусловлен требованием касания эллипсоидальной оценки и множества разрешимости в направлении, заданном вектором $l(t)$.

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

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

для произвольного вектора $l^* \in \mathbb{R}^n$: $\|l^*\| = 1$. Для того, чтобы подчеркнуть такую зависимость, будем использовать обозначение $\mathcal{E}(w(t), W(t, l^*))$. Перебором различных значений l^* могут быть получены разные оценки (в основном алгоритме такие векторы $l^* = l_k$ для построения разных оценок формируются в строке 2; при этом $W_k(t) = W(t, l_k)$). Будем далее считать, что векторы l^* выбираются таким образом, что $\langle l^*, \mathbf{P}l^* \rangle \neq 0$, что необходимо для корректности формул (7). Справедливо следующее утверждение, в общем случае обоснованное в [10]:

Теорема 1. Для любого $t \leq 0$

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

В строке 6 основного алгоритма необходимо отбирать те вершины графа, которые удовлетворяют условию

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

Используя определение эллипсоида, это условие можно переписать в виде вспомогательной оптимизационной задачи:

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

Теперь можно построить *позиционное управление*, переводящее траекторию системы (1) из позиции (t, x) , $t < 0$, в позицию $(0, x^*)$. Оно может быть найдено за счет «прицеливания» [12] на один из построенных эллипсоидов $\mathcal{E}(w(t), W(t))$:

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

где $l^0(t)$ – максимизатор в выражении для полурасстояния

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

При $x \notin \text{int } \mathcal{E}(w(t), W(t))$ вектор $l^0(t)$ может быть найден так же, как

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

где λ – единственный неотрицательный корень уравнения

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

Если же $x \in \text{int } \mathcal{E}(w(t), W(t))$, то $l^0(t) = 0$.

Заметим, что в основном алгоритме (строки 9–11) если очередная точка $x^{(i)}$ не совпадает ни с одной из ранее построенных вершин графа v_j (вероятность этого события равна 1), то значение $t_{\min, j} < 0$ для каждого соответствующего значения j выбирается таким образом, что $v_j \in \partial \mathcal{E}(w(t_{\min, j}), W_{k^*}(t_{\min, j}))$. То есть в начальный момент времени точка лежит на границе эллипсоида. Используемый метод экстремального прицеливания на эллипсоид обладает тем свойством (см. подробнее [12]), что траектория замкнутой системы, начинающаяся на границе эллипсоида, останется на ней

до конечного момента времени. Таким образом, особенно важным в данной работе является тот случай, когда точка x лежит на границе эллипсоида $\mathcal{E}(w(t), W(t))$ при любых $t \leq 0$. При этом

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

Можно найти соответствующее значение вектора $l^0(t) = e^{tA^T} s$, не решая указанную выше вспомогательную задачу оптимизации или уравнение относительно λ .

Подытожим свойства построенного позиционного управления в следующем утверждении

Теорема 2. Пусть для некоторых значений j, k имеется вершина $v_j \in \mathcal{E}(w(t_{\min,j}), W_k(t_{\min,j}))$. Рассмотрим задачу Коши для дифференциального включения $\dot{x} \in Ax + Bu^(t, x) + f$, $x(t_{\min,j}) = v_j$, $t \in [t_{\min,j}, 0]$. Эта задача имеет решения, причем для любого такого решения $x(t) = x(t, t_{\min,j}, v_j)|_{u^*(\cdot)}$ выполнено условие $x(0) = x^*$.*

В основном алгоритме для каждой новой случайной точки $x^* = x^{(i)}$ необходимо построить внутренние эллипсоидальные оценки множеств разрешимости и при помощи их определить те ранее построенные вершины графа Γ , которые можно соединить с новой вершиной (строка 6 алгоритма). Пусть для x^* построены эллипсоидальные оценки $\mathcal{E}(w(t), W_k(t))$, $k = 1, \dots, K$, $t \in [T, 0]$. На данной итерации работы алгоритма должны быть обработаны только те вершины $v_j \in V$, для которых выполнено условие (3). В этой формуле используется вспомогательный параметр (см. [7])

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

где ζ_n – объем единичной сферы в $\mathbb{R}^n$, $\mu(\mathcal{X}_{free})$ – лебегова мера множества $\mathcal{X}_{free}$, $\kappa = |V|$ – 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 шагах работы алгоритма (при малых κ). Если дополнительно предположить, что выполнено условие

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

то согласно [3] при указанном выше выборе r_i представленный алгоритм обладает свойством асимптотической оптимальности.

Для каждой отобранной согласно (3) вершины v_j можно определить наименьшее по абсолютной величине время $t_{\min,j}$ перехода из v_j в точку $x^{(i)}$, а также соответствующий номер эллипсоидальной оценки k^* : $v_j \in$

$\in \mathcal{E}(w(t_{\min,j}), W_{k^*}(t_{\min,j}))$. Далее необходимо построить управление. Формулу (9) нельзя непосредственно использовать в основном алгоритме, так как там речь идет о программном управлении (строка 11 алгоритма), а не о позиционном. Переход от позиционного управления к программному всегда можно выполнить за счет решения вспомогательной задачи Коши

$$(11) \quad \begin{cases} \dot{x}(t) \in Ax(t) + Bu^*(t, x) + f \\ x(t_{\min,j}) = v_j, \end{cases}$$

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

и последующего выделения однозначного селектора из многозначного отображения $u^*(t, x^*(t))$. Здесь $x^*(t)$ – решение задачи (11).

Для каждой построенной траектории $x^*(t)$, $t \in [t_{\min,j}, 0]$ далее должна быть произведена проверка выполнения следующего условия (строка 13 алгоритма): $x^*(t) \in \mathcal{X}_{free}$, $\forall t \in [t_{\min,j}, 0]$. Согласно (2) достаточно проверить выполнение неравенства

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

Если это условие не выполнено, то траектория отбраковывается алгоритмом. В противном случае она может быть использована для построения ребра, соединяющего новую вершину $x^* = x^{(i)}$ 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.

Замечание 2. Согласно (6) матрицы конфигурации эллипсоидов $W(t)$ не зависят от конкретной точки x^* , из которой необходимо строить множества разрешимости на каждой итерации работы алгоритма. Таким образом, матрицы $W(t)$, $t \in [T, 0]$, для разных l^* могут быть подсчитаны заранее, до начала вычислений в основном цикле алгоритма. Аналогично, из (5) видно, что для центров эллипсоидов справедливо соотношение

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

а значит, функция $\tilde{w}(t)$ может быть подсчитана заранее. Внутри основного цикла работы алгоритма нужно лишь рассчитывать функции $w^*(t)$ и суммировать их с $\tilde{w}(t)$. Это позволяет упростить вычисления.

Замечание 3. При реализации алгоритма Ellipsoidal KRRT* для конкретной системы (1) важно подобрать наиболее подходящее значение параметра $T < 0$. Если окажется, что абсолютная величина T слишком мала, то

это приведет к созданию искусственного ограничения при формировании новых ветвей случайного дерева, а потому найденная в итоге траектория может быть далеко не оптимальной. Если же абсолютная величина T слишком велика, то это может привести к дополнительным, лишним вычислениям эллипсоидальных оценок, что отрицательно скажется на скорости работы алгоритма. Для выбора наиболее адекватного значения параметра T можно поставить серию экспериментов и проанализировать статистику по реализовавшимся в них значениям величин $t_{\min,j}$.

5. Пример (линейная динамика)

Рассмотрим пример построения субоптимальной траектории в задаче быстрого перевода траектории из начальной позиции в целевое множество, а также сравним полученные результаты с алгоритмом, рассмотренным в [6].

Пусть $x \in \mathbb{R}^2$ и динамика системы (1) задается следующими параметрами:

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

На управляющие параметры наложены ограничения: $u \in \mathcal{P} = \mathcal{E}(p, P)$, где

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

Отметим, что эти ограничения учитываются в обоих алгоритмах. В (10) положим $\eta = 4$. Зададим вспомогательную матрицу $R = \mathbb{I}_{2 \times 2}$, необходимую для расчета функционала в алгоритме из [6]. Зафиксируем $T = -2$. Такой выбор обусловлен значениями $t_{\min,j}$, полученными Ellipsoidal KRRT* при разных T в серии экспериментов:

	Номер теста									
	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

Здесь $t_{avg} = \sum_{j=1}^{|V|} t_{\min,j}/|V|$, $t_{\min} = \min_j t_{\min,j}$. Видно, что значения $t_{\min} < -2$

в экспериментах не встречаются, а потому брать величину $T < -2$ нет смысла. Значение -2 взято с небольшим запасом относительно наименьшего из $t_{\min}$.

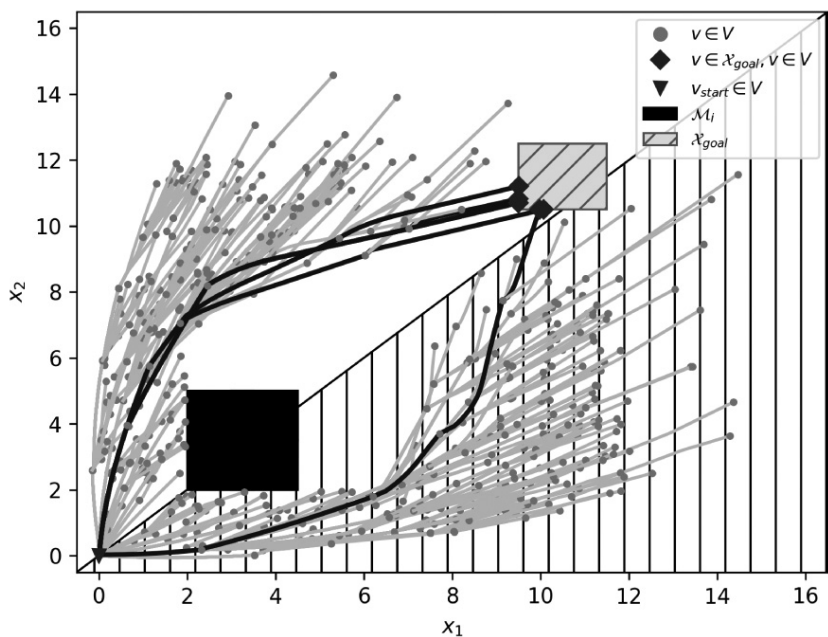


Рис. 1. Ellipsoidal KRRT*.

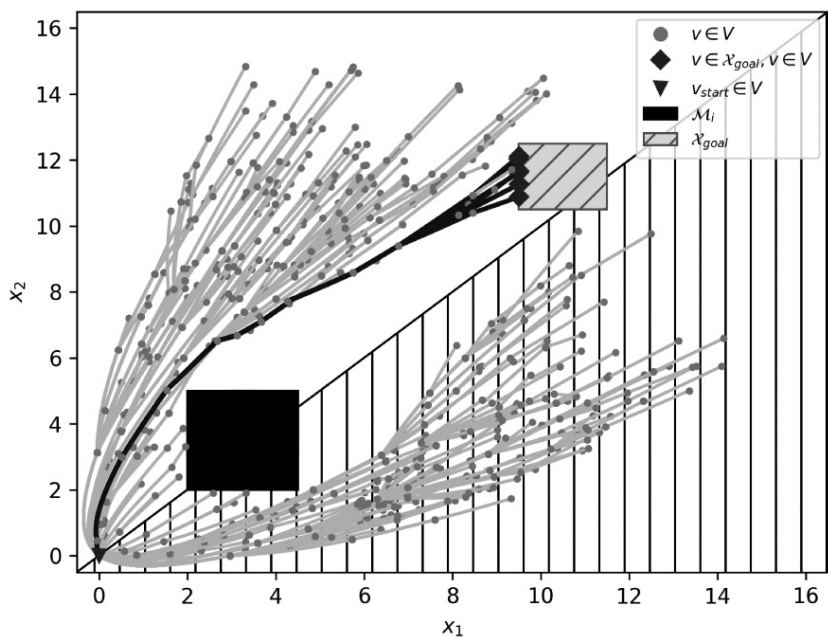


Рис. 2. KRRT*.

На рис. 1 и рис. 2 представлены результаты работы алгоритмов Ellipsoidal KRRT* и KRRT* соответственно, которые были получены следующим образом: алгоритм KRRT* (из [6]) в качестве случайных точек использо-

вал ту же последовательность точек $Samples = \{x^{(i)}\}_{i=1}^N$, $N = 2728$, которая потребовалась алгоритму Ellipsoidal KRRT* для построения дерева $\Gamma_1 = (V_1, E_1)$: $|V_1| = 500$.

Заметим, что алгоритм KRRT* в этом случае построил дерево $\Gamma_2 = (V_2, E_2)$: $|V_2| = 469$. Уменьшение количества вершин связано с тем, что часть пробных ребер была отбракована из-за нарушений ограничений на управления или из-за фазовых ограничений. Также отметим, что на рис. 1 через заштрихованную область проходит субоптимальная траектория, которую не смог найти алгоритм KRRT*.

При заданном целевом множестве

$$X_{goal} = \{x \in \mathbb{R}^2 : 9,5 < x_1 < 11,5, 10,5 < x_2 < 12,5\}$$

получились следующие значения функционалов:

$$\begin{aligned} t_1^* &= \min\{C(v) : v \in V_1, v \in X_{goal}\} = 0,26, \\ t_2^* &= \min\{C(v) : v \in V_2, v \in X_{goal}\} = 0,47. \end{aligned}$$

6. Нелинейная динамика

Применим теперь описанный выше метод построения субоптимальной траектории линейной системы для решения аналогичных задач в случае нелинейной динамики. Рассмотрим следующую систему обыкновенных дифференциальных уравнений:

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

Как и ранее, полагаем, что начальное состояние $x(0) = x_0 \in \Omega$ фиксировано, $u \in \mathcal{P} = \mathcal{E}(p, P) \subset \mathbb{R}^m$. Функции $f(x)$ и $g(x)$ предполагаются дважды непрерывно дифференцируемыми при $x \in \Omega$.

Алгоритм Ellipsoidal KRRT*, решающий сформулированные выше задачи управления, остается схожим по форме. Отметим лишь отличия, связанные с нелинейной динамикой.

Для построения оценок множеств разрешимости нелинейной системы в окрестности каждой очередной новой вершины $x^{(i)}$ графа Γ воспользуемся линеаризацией уравнений (12). Пусть $f_s(x)$ – s -я компонента вектора $f(x)$, $g_s(x)$ – s -я строка матрицы $g(x)$, $s = 1, \dots, n$, p_r – r -я компонента вектора p , $r = 1, \dots, m$. Тогда

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

где $Q^{g,(sr)}(\zeta_{sr})$ – матрица из вторых производных функции $g_{sr}(x)$, подсчитанная в некоторой точке $\zeta_{sr} \in \mathbb{B}_{r_i}(x^{(i)})$, $Q^{f,(s)}(\xi_s)$ – матрица из вторых производных функции $f_s(x)$, подсчитанная в точке $\xi_s \in \mathbb{B}_{r_i}(x^{(i)})$. Радиус шара $\mathbb{B}_{r_i}(x^{(i)})$ по-прежнему определяется из формул (10). Также здесь использованы некоторые точки $\eta_{sr} \in \mathbb{B}_{r_i}(x^{(i)})$. Заметим, что шар $\mathbb{B}_{r_i}(x^{(i)})$ (см. (3)) здесь используется в качестве допустимой области изменения переменной x . Это позволяет оценить погрешность линеаризации:

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

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

Здесь $\lambda_{\max}(Q)$ – максимальное по модулю собственное значение симметричной матрицы Q . В итоге линеаризованную систему можно записать в следующем виде:

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

где

$$\begin{aligned} A(x^{(i)}) = & \begin{pmatrix} \left(\frac{\partial f_1}{\partial x}(x^{(i)}) + \sum_{r=1}^m \frac{\partial g_{1r}}{\partial x}(x^{(i)})p_r \right)^T \\ \dots \\ \left(\frac{\partial f_n}{\partial x}(x^{(i)}) + \sum_{r=1}^m \frac{\partial g_{nr}}{\partial x}(x^{(i)})p_r \right)^T \end{pmatrix}, \quad B(x^{(i)}) = \begin{pmatrix} g_1(x^{(i)}) \\ \dots \\ g_n(x^{(i)}) \end{pmatrix}, \\ h(x^{(i)}) = & \begin{pmatrix} f_1(x^{(i)}) - \left\langle \frac{\partial f_1}{\partial x}(x^{(i)}), x^{(i)} \right\rangle - \sum_{r=1}^m \left\langle \frac{\partial g_{1r}}{\partial x}(x^{(i)}), x^{(i)} \right\rangle p_r \\ \dots \\ f_n(x^{(i)}) - \left\langle \frac{\partial f_n}{\partial x}(x^{(i)}), x^{(i)} \right\rangle - \sum_{r=1}^m \left\langle \frac{\partial g_{nr}}{\partial x}(x^{(i)}), x^{(i)} \right\rangle p_r \end{pmatrix}. \end{aligned}$$

Погрешность линеаризации v будем далее интерпретировать как неопределенность $v(t)$, на возможные значения которой наложены поточечные ограничения:

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

Для любого управления $u(\cdot) \in \mathcal{U}$ и любой измеримой ограниченной функции $v(t)$, удовлетворяющей указанному выше ограничению, через $x(t, 0, x_0)|_{u(\cdot), v(\cdot)}$ обозначим соответствующую траекторию системы (14), выпущенную из начальной позиции x_0 в момент времени 0. Заметим, что любая траектория $x(t, 0, x_0)|_{u(\cdot)}$ системы (12) является также траекторией (14) при некоторой допустимой помехе $v(\cdot)$, если только $x(\tau, 0, x_0)|_{u(\cdot)} \in \mathbb{B}_{r_i}(x^{(i)})$, $\forall \tau \in [t, 0]$. Обратное утверждение неверно.

Замечание 4. Поскольку в ходе выполнения алгоритма Ellipsoidal KRRT* необходимо многократно производить линеаризацию системы (12) в окрестности каждой очередной точки $x^{(i)}$, то для ускорения вычислений имеет смысл оценить погрешность линеаризации более грубо, а именно в формуле (13) заменить максимумы по $\mathbb{B}_{r_i}(x^{(i)})$ на максимумы по множеству Ω . Тогда достаточно будет подсчитать их единожды.

Из-за появления в уравнениях погрешности линеаризации необходимо модифицировать формулы для вычисления эллипсоидальных оценок множеств разрешимости, использованные выше для случая линейной динамики. Для этого можно воспользоваться результатами из [11]. А именно, теперь эллипсоиды $\mathcal{E}(w(t), W(t))$, $t \leq 0$, задаются следующими уравнениями:

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

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

Здесь $W = W(t)$, $A = A(x^{(i)})$, $B = B(x^{(i)})$, $P = BPB^T$, $S = S(t)$ – ортогональная матрица, для которой

$$SP^{1/2}l(t) = \lambda(t)P^{1/2}l(0), \quad S(0) = \mathbb{I}_{n \times n},$$

$$\lambda(t) = \sqrt{\frac{\langle l(t), Pl(t) \rangle}{\langle l(0), Pl(0) \rangle}}, \quad \pi(t) = \frac{\rho(x^{(i)})\|l(t)\|}{\sqrt{\langle l(t), W(t)l(t) \rangle}}.$$

Как и в случае с линейной динамикой, каждая эллипсоидальная оценка зависит от некоторой кривой $l(t)$, являющейся решением задачи Коши (8) для произвольного $l^* \in \mathbb{R}^n$: $\|l^*\| = 1$. Функции $S(t)$ и $\pi(t)$ непрерывно зависят от t .

Заметим, что из-за неопределенности $v(t)$ теперь уже нельзя говорить о задаче перевода траектории системы (14) точно в точку $x^{(i)}$. Можно рассмотреть лишь некоторую малую окрестность этой точки радиуса $\varepsilon_i > 0$, в

которую и нужно перевести траекторию системы, несмотря на неопределенность $v(\cdot)$. Конечно, должно быть выполнено неравенство $\varepsilon_i < r_i$.

Некоторые (или даже все) из эллипсоидальных оценок, вычисляемых по формулам (16), (17), могут в какой-то момент времени $t = \tilde{t} < 0$ вырождаться. То есть может оказаться, что $W(\tilde{t}) \not\geq 0$. В этом случае процесс вычисления такой оценки следует прекратить, а в основном алгоритме нужно учесть, что эта оценка построена лишь при $t \in [\tilde{t}, 0]$, а не на всем отрезке $[T, 0]$.

Соответствующее позиционное управление, переводящее траектории системы (14) из позиции (t, x) , $t < 0$, в ε_i -окрестность точки $x^{(i)}$ в момент времени $t = 0$, может быть найдено за счет «прицеливания» на один из построенных эллипсоидов $\mathcal{E}(w(t), W(t))$ аналогично (9). Основные свойства такого позиционного управления указаны в следующем утверждении

Теорема 3. Пусть для некоторых значений индексов j, k имеется вершина $v_j \in \mathcal{E}(w(t_{\min,j}), W_k(t_{\min,j}))$, причем $W_k(t) > 0$, $\forall t \in [t_{\min,j}, 0]$. Рассмотрим задачу Коши для дифференциального включения $\dot{x} \in A(x^{(i)})x + B(x^{(i)})u^(t, x) + h(x^{(i)}) + v(t)$, $x(t_{\min,j}) = v_j$, $t \in [t_{\min,j}, 0]$. Для любой допустимой помехи $v(\cdot)$ (т.е. любой измеримой функции, удовлетворяющей ограничениям (15)), существует решение указанной задачи Коши $x(t) = x(t, t_{\min,j}, v_j)|_{u^*(\cdot), v(\cdot)}$, причем для него выполнено условие $x(0) \in \mathbb{B}_{\varepsilon_i}(x^{(i)})$. В частности, среди построенных таким образом траекторий содержится и функция $x^*(t)$, являющаяся решением задачи Коши для дифференциального включения $\dot{x} \in f(x) + g(x)u^*(t, x)$, $x(t_{\min,j}) = v_j$, $t \in [t_{\min,j}, 0]$.*

Поскольку в случае нелинейной динамики линеаризованная система (14) адекватно приближает нелинейную систему (12) лишь при $x \in \mathbb{B}_{r_i}(x^{(i)})$, то в алгоритме следует изменить (по сравнению со случаем с линейной динамикой) условия для отсеивания лишних траекторий (строка 13 основного алгоритма):

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

Для получения оптимальной пары $(x(t), u(t))$, для которой $x(t_{\min,j}) = v_j$, $x(0) \in \mathbb{B}_{\varepsilon_i}(x^{(i)})$, необходимо модифицировать соотношения (11), использованные для случая линейной динамики:

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

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

То есть построенное позиционное управление должно быть поставлено в исходную нелинейную систему, а не в ее линеаризованный аналог, для последующего получения программного управления.

Заметим, что траектория $x(t)$, стартуя с границы эллипсоида $\mathcal{E}(w(t_{\min,j}), W_{k^*}(t_{\min,j}))$, при $t > t_{\min,j}$ может попасть внутрь соответствующего эллипсоида из-за «неоптимальности» неопределенности $v(\cdot)$. Более того, может оказаться, что итоговая траектория попадет в точку $x^{(i)}$ при некотором $t^* < 0$. В этом случае следует завершить вычисление траектории $x(t)$, а в качестве оценки времени быстрогодействия для ребра дерева использовать величину $(t^* - t_{\min,j^*})$, а не $-t_{\min,j^*}$. Как уже было сказано выше, возможна и другая ситуация, когда траектория системы в момент времени $t = 0$ попадет в точку $\tilde{x} \in \mathbb{B}_{\varepsilon_i}(x^{(i)})$, $\tilde{x} \neq x^{(i)}$. Это также следует учесть в основном алгоритме: во множество вершин графа V должна быть добавлена не точка $x^{(i)}$, а $\tilde{x}$ (см. строки 17–23 основного алгоритма). Здесь, однако, возникает сложность с той частью базового алгоритма RRT*, в которой производится «пересвязывание» ранее построенной части графа (см. строки 19–23). Поскольку теперь найденное управление не гарантирует попадание траектории в точку $x^{(i)}$, то либо нужно считать $\varepsilon_i > 0$ достаточно малым и за счет этого отождествлять все точки из ε_i -окрестности $x^{(i)}$, либо нужно в принципе отказаться от «пересвязывания». В последнем случае, однако, качество работы алгоритма может ухудшиться.

7. Пример (нелинейная динамика)

Рассмотрим математическую модель плоского движения автономного аппарата (робота), приводимого в движение двумя роторами. Уравнения движения имеют следующий вид:

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

Здесь (x_1, x_2) – положение центра масс аппарата, φ – угол, задающий его ориентацию на плоскости, m – масса, J – момент инерции, ξ_1 и ξ_2 – коэффициенты вязкого трения, r – радиус аппарата. Через u_1 и u_2 здесь обозначены силы тяги от двух роторов. На управляющие параметры наложены следующие ограничения: $u_i \in [0, u_{\max}]$, $i = 1, 2$, где величина $u_{\max}$ задана.

Приведем систему (19) к следующему виду:

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

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

$$A = \begin{pmatrix} \mathbb{O}_{3 \times 3} & \mathbb{I}_{3 \times 3} \\ \mathbb{O}_{3 \times 3} & -\tilde{A} \end{pmatrix}, \quad g(x) = \begin{pmatrix} \mathbb{O}_{3 \times 2} \\ \tilde{g} \end{pmatrix},$$

$$\tilde{A} = \text{diag} \left(\frac{\xi_1}{m}, \frac{\xi_1}{m}, \frac{\xi_2}{J} \right), \quad \tilde{g} = \begin{pmatrix} \frac{\cos(x_3)}{m} & \frac{\cos(x_3)}{m} \\ \frac{\sin(x_3)}{m} & \frac{\sin(x_3)}{m} \\ \frac{r}{J} & -\frac{r}{J} \end{pmatrix}.$$

Множество допустимых значений управлений оценим при помощи эллипсоида:

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

При линеаризации уравнений (20) в окрестности очередной вершины графа $x^{(i)}$ (см. (14)) получим следующие параметры: $B(x^{(i)}) = g(x^{(i)})$,

$$A(x^{(i)}) = A + \frac{u_{\max}}{m} \begin{pmatrix} \mathbb{O}_{3 \times 6} \\ 0 & 0 & -\sin(x_3^{(i)}) & 0 & 0 & 0 \\ 0 & 0 & \cos(x_3^{(i)}) & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix},$$

$$h(x^{(i)}) = \frac{u_{\max} x_3^{(i)}}{m} \begin{pmatrix} \mathbb{O}_{3 \times 1} \\ \sin(x_3^{(i)}) \\ -\cos(x_3^{(i)}) \\ 0 \end{pmatrix},$$

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

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

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

Препятствия $\mathcal{M}_i$ определим так, что ограничения будут наложены лишь на координаты x_1 и x_2 .

Ниже приведены результаты работы алгоритма Ellipsoidal KRRT* в этом случае. На рис. 3 изображены проекции ребер построенного в итоге графа Γ , а также препятствий на плоскость переменных (x_1, x_2) на момент, когда $|V| = 369$.

Параметры, использованные при численном моделировании: $T = -15$, $m = 5 \text{ кг}$, $J = 0,05 \frac{\text{кг} \cdot \text{м}}{\text{рад}}$, $r = 0,24 \text{ м}$, $\xi_1 = 8,5 \frac{\text{кг}}{\text{с}}$, $\xi_2 = 0,08 \frac{\text{кг} \cdot \text{м}^2}{\text{с} \cdot \text{рад}}$, $u_{\max} = 1,5 \text{ Н}$, $x_1(0) = \dot{x}_1(0) = x_2(0) = \dot{x}_2(0) = \varphi(0) = \dot{\varphi}(0) = 0$.

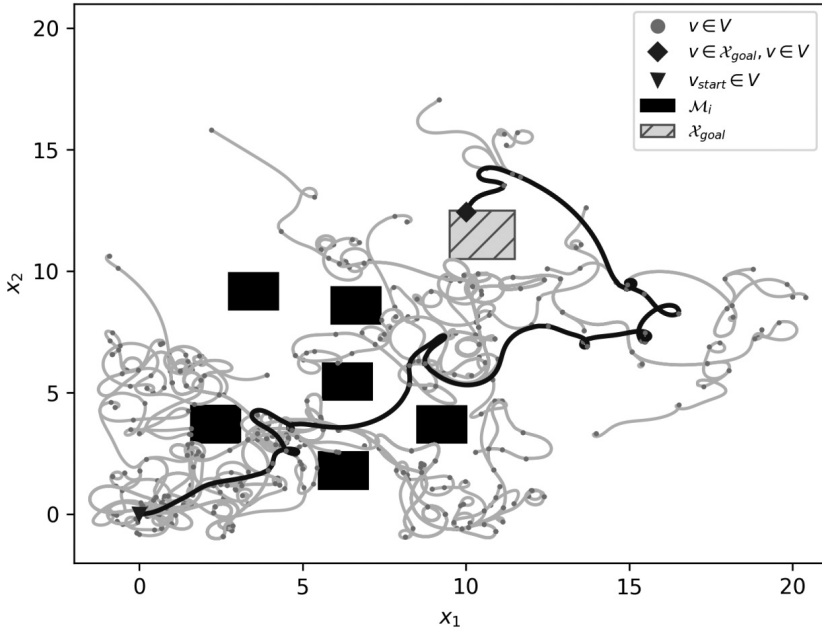


Рис. 3. Проекция дерева $\Gamma = (V, E)$ на плоскость (x_1, x_2) .

При заданном целевом множестве

$$X_{goal} = \{x \in \mathbb{R}^2 : 9,5 < x_1 < 11,5, 10,5 < x_2 < 12,5\}$$

значение функционала в этом примере следующее:

$$t^* = \min\{C(v) : v \in V, v \in X_{goal}\} = 134,42.$$

В отличие от линейного варианта предложенного в данной работе алгоритма сравнить его работу в случае нелинейной динамики с существующими алгоритмами типа KRRT* не представляется возможным из-за того, что последние либо вовсе не рассчитаны на системы с нелинейной динамикой, либо могут быть использованы только для расчета конкретных, специально подобранных примеров.

8. Заключение

Рассмотренная в работе модификация алгоритма RRT* позволяет строить субоптимальные траектории для объектов как с линейной, так и с нелинейной динамикой. Программная реализация предлагаемого метода может быть использована для решения различных прикладных задач. В отдельных ситуациях представленный алгоритм можно улучшить (сделать более эффективным) за счет использования различных эвристик. Исследованию этих вопросов будет посвящена дальнейшая работа авторов.

СПИСОК ЛИТЕРАТУРЫ

1. *Казаков К.А., Семенов В.А.* Обзор современных методов планирования движения // Тр. ИСП РАН. 2016. Т. 28. № 4. С. 241–294.
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 Transactions on Intelligent Vehicles. 2016. V. 1. No. 1. P. 33–55.
3. *Karaman S., Frazzoli E.* Sampling-based algorithms for optimal motion planning // Int. J. Robot. Res. 2011. V. 30. No. 7. P. 846–894.
4. *Арутюнов А.В., Магарил-Ильяев Г.Г., Тихомиров В.М.* Принцип максимума Понтрягина. М.: Факториал, 2006.
5. *Дубовицкий А.Я., Милютин А.А.* Задачи на экстремум при наличии ограничений // ЖВМ и МФ. 1965. Т. 5. № 3. С. 395–453.
6. *Webb D.J., 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. P. 5054–5061.
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.
8. *LaValle S.M., Kuffner J.J.* Randomized kinodynamic planning // Int. J. Robot. Res. 2001. V. 20. No. 5. P. 378–400.
9. *Shkolnik A., Walter M., Tedrake R.* Reachability-guided sampling for planning under differential constraints // Proc. of the IEEE Conf. on Robotics and Automation. 2009. P. 2859–2865.
10. *Kurzhanski A.B., Varaiya P.* On ellipsoidal techniques for reachability analysis. Part II: internal approximations, box-valued constraints // Optimization Methods and Software. 2002. V. 17. P. 207–237.
11. *Kurzhanski A.B., Varaiya P.* Reachability analysis for uncertain systems — the ellipsoidal technique // Dynam. Contin. Discrete Impuls. Syst. Ser. B. 2002. V. 9. No. 3. P. 347–367.
12. *Kurzhanski A.B., Varaiya P.* Dynamics and control of trajectory tubes. Theory and computation. Birkhäuser, 2014.

Статья представлена к публикации членом редколлегии А.А. Галеевым.

Поступила в редакцию 28.02.2023

После доработки 30.10.2023

Принята к публикации 21.12.2023