

УДК: 519.6

Метод поиска касательных в задаче быстрогодействия для колесного мобильного робота

В. Н. Белотелов^{1,a}, А. Н. Дарьина^{1,2,b}

¹Федеральный исследовательский центр «Информатика и управление» РАН,
Россия, 119333, г. Москва, ул. Вавилова, д. 44, стр. 2

²Московский государственный университет, факультет ВМиК,
Россия, 119992, г. Москва, Ленинские горы, ГСП-2

E-mail: ^a vbelotelov@gmail.com, ^b anna.daryina@mail.ru

Получено 11.04.2025, после доработки — 30.05.2025.

Принято к публикации 30.05.2025.

Поиск оптимальной траектории движения является нетривиальной задачей, на решение которой направлено большое число исследований. Большинство этих исследований посвящено решению задачи в общем виде вне зависимости от модели движения объекта. В такой постановке поиск оптимальной траектории возможен только численными методами. Вместе с тем в некоторых случаях возможно нахождение оптимальной траектории в аналитическом виде. В данной статье рассмотрена задача быстрогодействия с фазовыми ограничениями для колесного мобильного робота, движущегося по горизонтальной плоскости. Математическая модель робота является кинематической. Фазовые ограничения соответствуют препятствиям на плоскости, заданным в виде непересекающихся кругов, которые требуется избегать при движении. Независимыми управляющими воздействиями являются скорости колес, которые ограничены по абсолютной величине. Такая постановка часто применяется в тех случаях, когда динамические переходные процессы несущественны, например при управлении медленно движущимися гусеничными или колесными устройствами, в которых приоритет отдается мощности двигателей, а не их скорости. В статье показывается, что оптимальная траектория движения из начальной точки в конечную в выбранной кинематической постановке представляет собой последовательность отрезков общих касательных к парам кругов и дуг окружностей этих кругов. Геометрически кратчайший путь между начальной и конечной точками также состоит из отрезков касательных и дуг окружностей, поэтому оптимальное по быстродействию движение соответствует одному из локальных минимумов при поиске кратчайшего пути. Предложен аналитический метод поиска оптимальной траектории движения, основанный на построении графа возможных траекторий, где ребрами являются прямолинейные отрезки и дуги, а вершинами — точки их соединений, и поиска кратчайшего (быстрейшего) пути на графе с помощью метода Дейкстры. Представлено обоснование метода. Приведены результаты численных экспериментов по нахождению оптимальной траектории.

Ключевые слова: задача быстрогодействия, фазовые ограничения, кратчайший путь, колесный робот, кинематическая модель, дифференциальный привод

Работа выполнена при финансовой поддержке Министерства науки и высшего образования РФ, грант № 075-15-2024-544.

UDC: 519.6

Tangent search method in time optimal problem for a wheeled mobile robot

V. N. Belotelov^{1,a}, A. N. Daryina^{1,2,b}

¹Federal research center “Computer science and control”,
44/2 Vavilova st., Moscow, 119333, Russia

²Moscow State University, Faculty of Computational Mathematics and Cybernetics,
Leninskiye Gory, GSP-2, Moscow, 119992, Russia

E-mail: ^a vbelotelov@gmail.com, ^b anna.daryina@mail.ru

Received 11.04.2025, after completion — 30.05.2025.

Accepted for publication 30.05.2025.

Searching optimal trajectory of motion is a complex problem that is investigated in many research studies. Most of the studies investigate methods that are applicable to such a problem in general, regardless of the model of the object. With such general approach, only numerical solution can be found. However, in some cases it is possible to find an optimal trajectory in a closed form. Current article considers a time optimal problem with state limitations for a wheeled mobile differential robot that moves on a horizontal plane. The mathematical model of motion is kinematic. The state constraints correspond to the obstacles on the plane defined as circles that need to be avoided during motion. The independent control inputs are the wheel speeds that are limited in absolute value. Such model is commonly used in problems where the transients are considered insignificant, for example, when controlling tracked or wheeled devices that move slowly, prioritizing traction power over speed. In the article it is shown that the optimal trajectory from the starting point to the finishing point in such kinematic approach is a sequence of straight segments of tangents to the obstacles and arcs of the circles that limit the obstacles. The geometrically shortest path between the start and the finish is also a sequence of straight lines and arcs, therefore the time-optimal trajectory corresponds to one of the local minima when searching for the shortest path. The article proposes a method of search for the time-optimal trajectory based on building a graph of possible trajectories, where the edges are the possible segments of the trajectory, and the vertices are the connections between them. The optimal path is sought using Dijkstra’s algorithm. The theoretical foundation of the method is given, and the results of computer investigation of the algorithm are provided.

Keywords: time-optimal problem, state constraints, shortest path, wheeled robot, kinematic model, differential drive

Citation: *Computer Research and Modeling*, 2025, vol. 17, no. 3, pp. 401–421 (Russian).

This research was funded by the Ministry of Science and Higher Education of the Russian Federation, project No. 075-15-2024-544.

1. Введение

Задачи быстродействия с фазовыми ограничениями часто встречаются в области управления, автоматизации и теории управления. Они возникают в различных областях, таких как робототехника, авиация, автомобилестроение и др. В таких задачах необходимо оптимизировать производительность системы при соблюдении определенных фазовых ограничений. Фазовые ограничения — это ограничения на состояние системы, например максимальные и минимальные значения угловых координат, скорости, ускорения и т. д. Эти ограничения могут быть связаны с физическими свойствами системы или требованиями безопасности. Задача быстродействия для колесного мобильного робота может включать в себя множество аспектов, таких как планирование пути, управление движением, избегание препятствий и оптимизация траектории.

Задача оптимального быстродействия для движения робота из одной заданной точки в другую при наличии фазовых ограничений является фундаментальной проблемой в робототехнике. Она заключается в нахождении траектории, которая минимизирует время перемещения робота между двумя точками, учитывая ограничения на его динамику (скорость, ускорение, рывок) и избегая столкновений с препятствиями. При наличии нескольких препятствий эта задача становится задачей глобальной оптимизации с многими локальными минимумами.

Известные методы и алгоритмы для решения этой задачи можно разделить на аналитические (основанные на принципе максимума Понтрягина [Понтрягин и др., 1962; Понтрягин, 1989]), численные (методы оптимизации [Nocedal, Wright, 2006], в том числе методы, нацеленные изначально на решение задач оптимального управления [Грачев, Евтушенко, 1979; Федоренко, 1978], генетические и эволюционные алгоритмы [Дивеев, Константинов, 2018], метод сетевого оператора [Дивеев, Мендес Флорес, 2021], метод на основе оптимизации роя частиц (PSO) [Дарьина, Прокопьев, 2020]), вероятностные (марковские процессы принятия решений [Puterman, 1994]), методы машинного обучения (обучение с подкреплением [Sutton, Barto, 2018]), методы на основе дискретизации конфигурационного пространства (Rapidly-exploring Random Tree, RRT [Lavalle, 1998; Karaman, Frazzoli, 2011]), PRM (Probabilistic Roadmap Method [Kavraki et al., 1996]) и другие.

Методы оптимизации формулируют задачу планирования траектории как задачу оптимизации и используют численные методы для ее решения. Преимущества методов оптимизации в том, что они могут находить оптимальные траектории с учетом сложных динамических ограничений, в том числе когда область допустимых траекторий робота не является выпуклой. Методы численной оптимизации, широко развивающиеся в настоящее время, как показывают исследования, далеко не все пригодны для решения задачи оптимального управления [Дивеев, Константинов, 2018]; для их реализации зачастую требуется хорошее начальное приближение, а сами методы могут быть вычислительно дорогостоящими и не применимыми на борту робота во время его работы.

Среди методов, показывающих хорошие результаты на фазовых пространствах, которые дискретизированы сеткой, одним из самых популярных алгоритмов для поиска пути является метод A^* (A-star). Он использует эвристическую функцию для оценки стоимости пути и позволяет находить оптимальный маршрут с учетом препятствий. A^* сочетает в себе свойства алгоритма Дейкстры и жадного поиска. Метод D (Dynamic A) является расширением метода A^* . Он способен эффективно перепланировать траекторию при изменении окружения (например, при появлении новых препятствий). Однако, как и A^* , требует дискретизации пространства. Алгоритм Дейкстры находит кратчайший путь от начальной точки до всех остальных на пространствах, заданных в виде графов, что делает его полезным для поиска оптимального маршрута. Однако он может быть менее эффективным по сравнению с A^* , поскольку не использует эвристику.

Генетические алгоритмы, требующие дискретизации времени, медленно сходятся при большом числе узлов разбиения времени в силу того, что целевая функция сильно изрезана,

и начальные значения оказывают существенно большее влияние на качество, чем значения ближе к концу. Метод символьной регрессии, приближенно находящий управляющую функцию в аналитическом виде, менее подвержен влиянию на свою точность числа узлов дискретизации времени, — один из наиболее успешных, но, как и генетические и прочие методы численной оптимизации, является вычислительно трудным.

При решении задач оптимального управления численными методами оптимальное решение находится неточно. Чтобы оценить качество работы численных методов, полезно знать абсолютное значение минимума, которое может принимать функционал качества.

Настоящая работа является развитием идеи, предложенной в [Bereznev, 2021]. Здесь показано, что в кинематической постановке, которая широко используется при исследовании движения колесных роботов, задача поиска оптимальной траектории сводится к конечномерной, и можно применять метод поиска кратчайшего пути на графе. В данной работе приводится метод построения графа, в котором узлами являются точки пространства. Эти точки можно найти аналитически, а для графа применимы известные методы дискретного поиска, например Дейкстры.

Данная статья организована следующим образом. В § 2 приводятся описание модели робота и постановка задачи. В § 3 подробно изложен метод поиска касательных, который включает такие этапы как нахождение общих касательных, исключение заблокированных участков, вычисление максимальной скорости на участках пути, построение графа переходов. В этом же разделе приводится алгоритм для решения задачи быстродействия для колесного робота методом поиска касательных, оценивается вычислительная сложность алгоритма. В § 4 представлены результаты вычислительных экспериментов с предложенным методом. Статья завершается коротким заключением в § 5.

2. Описание модели робота и постановка задачи

Кинематическая модель колесного робота

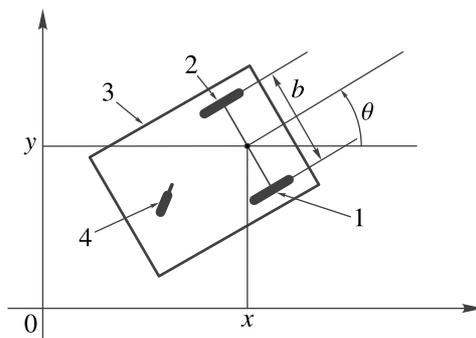


Рис. 1. Схема двухколесного робота: 1, 2 — колеса робота, 3 — корпус, 4 — пассивное (рояльное) колесо-подпорка. x и y — декартовы координаты центра колесной пары, θ — курсовой угол робота

В настоящей работе рассматривается движение по горизонтальной плоскости робота с дифференциальным приводом, т. е. с двумя ведущими колесами, которые управляются независимо друг от друга. Схема робота представлена на рис. 1, где 1, 2 — правое и левое ведущие колеса робота, 3 — корпус робота, 4 — пассивное рояльное колесо, выполняющее функцию дополнительной опоры для того, чтобы робот имел три точки опоры. Координаты x , y — декартовы координаты центра колесной пары робота, θ — курсовой угол робота; переменные x , y , θ образуют фазовый вектор $\mathbf{x} = (x, y, \theta)^T \in \mathbb{R}^3$. Кинематическая схема, представленная на рис. 1,

описывается одним геометрическим параметром — расстоянием между ведущими колесами робота b . Значение этого параметра влияет на соотношение усилий, которые требуется приложить к роботу для его продольного движения и для его поворота. Математическая модель, описывающая движение такого робота, имеет следующий вид:

$$\begin{aligned} \dot{x} &= \frac{u_1 + u_2}{2} \cos(\theta), \\ \dot{y} &= \frac{u_1 + u_2}{2} \sin(\theta), \\ \dot{\theta} &= \frac{u_1 - u_2}{b}. \end{aligned} \quad (1)$$

В качестве управляющих воздействий u_1 и u_2 выбраны скорости соответственно правого и левого колес. Управляющие воздействия образуют управляющий вектор $\mathbf{u} = (u_1, u_2)^T \in \mathbf{U} \subseteq \mathbb{R}^2$, принадлежащий компактной области двумерного пространства, которая определяет ограничения на управляющие воздействия:

$$|u_i| \leq u_{\max}, \quad i = 1, 2. \quad (2)$$

Описанная модель часто применяется в задачах, связанных с движением колесных роботов, когда не требуется учитывать переходные процессы, связанные с динамикой робота. В кинематической постановке корпус робота и его колеса считаются невесомыми, принимается допущение о том, что можно напрямую управлять скоростями вращения колес (т.е. линейными скоростями их центров). Геометрическими размерами робота пренебрегаем, считая робота математической точкой.

ЗАМЕЧАНИЕ. В тех случаях, когда необходимо учитывать реальные размеры робота, можно для простоты считать, что робот сам ограничен в горизонтальных размерах окружностью с некоторым постоянным радиусом r . Все рассуждения, приведенные в данной статье, остаются в силе, если увеличить радиус препятствий на величину r , а робота по-прежнему считать точкой. В случае некруговых препятствий можно расширить границы этих препятствий также на величину r . Такое расширение является частным случаем сложения Минковского для геометрических фигур, представляющих препятствия, и круга, представляющего робота.

Постановка задачи быстродействия

Рассмотрим классическую задачу быстродействия. Пусть математическая модель движения робота задана как (1) с ограничениями на управляющие воздействия (2).

В начальный момент времени $t^0 = 0$ заданы значения фазовых координат робота:

$$x(0) = x_0, \quad y(0) = y_0, \quad \theta(0) = \theta_0. \quad (3)$$

Задано терминальное условие:

$$x(t^f) = x^f, \quad y(t^f) = y^f, \quad \theta(t^f) = \theta^f, \quad (4)$$

где t^f — неизвестное заранее время достижения терминального состояния.

Движение робота по плоскости с препятствиями в общем виде можно исследовать только численными методами. Будем считать, что все препятствия в нашей задаче являются круговыми и непересекающимися. Это определяет следующие фазовые ограничения:

$$\|(x(t), y(t)) - \mathbf{C}^k\| \leq r^k, \quad k = 1, \dots, K, \quad (5)$$

где K — количество препятствий. Ограничения (5), определяющие область недопустимых значений фазовых переменных, означают, что траектория робота ни в какой момент времени не должна пересекать круги с радиусами r^k и с центрами $\mathbf{C}^k = (x_c^k, y_c^k)^T$.

Требуется достичь терминального состояния за минимально возможное время с учетом фазовых ограничений (5), т. е. критерий качества имеет вид

$$J = \int_0^{t^f} 1 dt = t^f \rightarrow \min_{\mathbf{u} \in \mathbf{U}}. \quad (6)$$

Требуется найти функцию $\mathbf{u} = \mathbf{u}(t) \in \mathbf{U}$, которая при ее подстановке в систему (1) обеспечивает переход этой системы из начального состояния (3) в терминальное (4) за минимальное время t^f с учетом фазовых ограничений (5).

Ограничения (5) означают, что область допустимых траекторий системы не является выпуклой, что сильно затрудняет поиск оптимального решения аналитическими методами или методами математического программирования.

3. Метод поиска касательных

В описанной постановке возможно аналитически найти оптимальную по быстрдействию траекторию, которая, как будет показано ниже, совпадает в данном случае с кратчайшим в геометрическом смысле путем из начальной точки в терминальную с учетом ориентации робота в начальном положении и заданной ориентацией этого робота в терминальном положении, причем при оптимальном движении робот будет совершать повороты на месте только в начальной и в терминальной точках. Для поиска такой оптимальной траектории будем использовать метод, описанный далее. Для реализации метода необходимо автоматизировать приведенные ниже вычисления. Чтобы не загромождать формулы, будем использовать одинаковые обозначения для точек и их радиус-векторов. Например, A может обозначать также и радиус-вектор точки A .

Нахождение отрезков касательных

Для нахождения касательных к окружности, проходящих через заданную точку, воспользуемся следующим построением (рис. 2). Считаем, что точка P лежит вне круга с радиусом r с центром в точке C . Тогда существует две касательных, проходящих через точку P и касающихся окружности в точках A и B , симметричных относительно прямой, соединяющей центр круга C и точку P . Поскольку треугольник PCA прямоугольный, можно определить косинус и синус угла φ при вершине C как

$$\begin{aligned} \cos \varphi &= \frac{r}{d}, \quad d = \|PC\|, \\ \sin \varphi &= \sqrt{1 - \cos^2 \varphi}. \end{aligned} \quad (7)$$

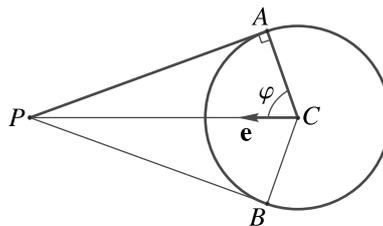


Рис. 2. Нахождение касательных к окружности, проходящих через заданную точку. Точки A и B находятся как вершины прямоугольных треугольников PAC и PBC

Составим матрицу поворота:

$$\mathbf{M} = \begin{bmatrix} \cos \varphi & -\sin \varphi \\ \sin \varphi & \cos \varphi \end{bmatrix}.$$

Тогда точки A и B можно найти как

$$A = C + r\mathbf{M}^T \vec{e}, \quad B = C + r\mathbf{M}\vec{e},$$

где вектор $\vec{e} = \frac{P-C}{d}$.

Аналогичным образом найдем общие касательные к двум окружностям с радиусами r_1, r_2 и центрами в точках C_1, C_2 . Считаем, что эти окружности не пересекаются (рис. 3). Из построений, показанных на рис. 3, видно, что косинусы и синусы углов φ_1 и φ_2 можно найти как

$$\begin{aligned} \cos \varphi_1 &= \frac{r_1 - r_2}{d}, & d &= \|C_1 C_2\|, \\ \cos \varphi_2 &= \frac{r_1 + r_2}{d}, \\ \sin \varphi_i &= \sqrt{1 - \cos^2 \varphi_i}, & i &= 1, 2. \end{aligned} \tag{8}$$

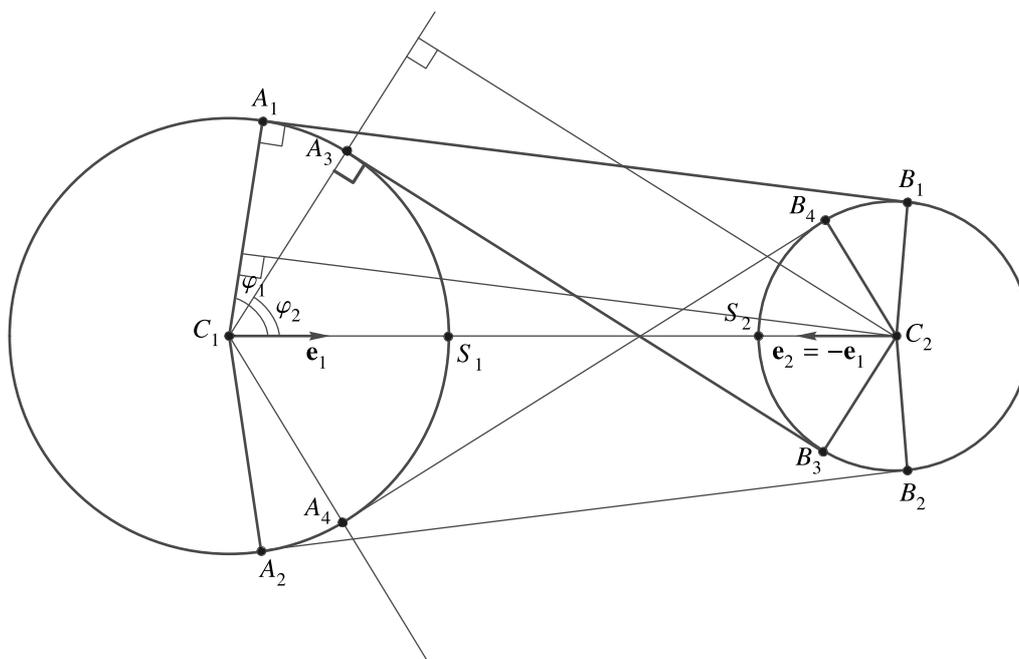


Рис. 3. Общие касательные к двум окружностям. Точки A_i и B_i находятся путем определения углов φ_1 и φ_2

Составим матрицы поворотов:

$$\mathbf{M}_i = \begin{bmatrix} \cos \varphi_i & -\sin \varphi_i \\ \sin \varphi_i & \cos \varphi_i \end{bmatrix}$$

и найдем точки A_j, B_j как

$$\begin{aligned} A_1 &= C_1 + r_1 \mathbf{M}_1 \vec{e}_1, & B_1 &= C_2 + r_2 \mathbf{M}_1 \vec{e}_1, \\ A_2 &= C_1 + r_1 \mathbf{M}_1^T \vec{e}_1, & B_2 &= C_2 + r_2 \mathbf{M}_1^T \vec{e}_1, \\ A_3 &= C_1 + r_1 \mathbf{M}_2 \vec{e}_1, & B_3 &= C_2 - r_2 \mathbf{M}_2 \vec{e}_1, \\ A_4 &= C_1 + r_1 \mathbf{M}_2^T \vec{e}_1, & B_4 &= C_2 - r_2 \mathbf{M}_2^T \vec{e}_1, \end{aligned}$$

где вектор $\vec{e}_1 = \frac{C_2 - C_1}{d}$.

Исключение заблокированных отрезков

При построении пути исключаются такие отрезки прямых, которые пересекаются с какими-либо препятствиями. Для определения пересечения отрезка прямой P_1P_2 с кругом радиусом r и с центром в точке C будем использовать следующий критерий (рис. 4).

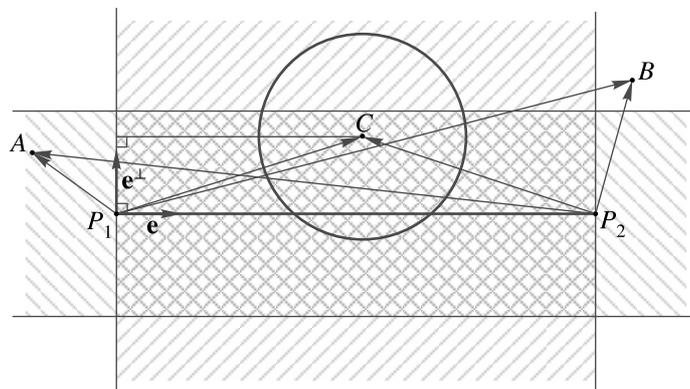


Рис. 4. Определение пересечения отрезка с кругом. При попадании центра круга C в заштрихованную область, являющуюся пересечением заштрихованных горизонтальной и вертикальной полос, отрезок P_1P_2 считается заблокированным кругом

Найдем единичный вектор \vec{e} вдоль отрезка P_1P_2 и ортогональный ему вектор \vec{e}^\perp как

$$\vec{e} = (e_1, e_2)^T = \frac{\overrightarrow{P_2 - P_1}}{\|P_2 - P_1\|}, \quad \vec{e}^\perp = (-e_2, e_1)^T.$$

Условием того, что расстояние от центра круга до прямой, соединяющей точки P_1 и P_2 , меньше или равно радиусу круга r , является

$$d = \|(\overrightarrow{C - P_1}) \cdot \vec{e}^\perp\| \leq r; \quad (9)$$

это условие означает, что точка C лежит внутри горизонтальной полосы шириной $2r$ (рис. 4). При этом проекция центра круга C на отрезок P_1P_2 должна лежать на отрезке P_1P_2 , что эквивалентно условию

$$\left((\overrightarrow{C - P_1}) \cdot \vec{e} \right) \cdot \left((\overrightarrow{C - P_2}) \cdot \vec{e} \right) \leq 0, \quad (10)$$

что означает, что точка C лежит внутри вертикальной полосы, ширина которой равна длине отрезка P_1P_2 (рис. 4). Так, для точки A оба угла — между \vec{e} и P_1A и между \vec{e} и P_2A — являются тупыми, поэтому оба скалярных произведения в (10) отрицательны. Для точки B углы между \vec{e} и P_1B и между \vec{e} и P_2B острые, поэтому оба скалярных произведения положительны. Для точки C угол между \vec{e} и P_1C острый, а между \vec{e} и P_2C — тупой, поэтому одно скалярное произведение положительно, другое — отрицательно.

Равенство нулю допускается как в (9), так и в (10). В первом случае оно означает касание круга и прямой P_1P_2 , во втором — то, что центр круга C лежит на прямой, перпендикулярной P_1P_2 и проходящей через конец отрезка.

Отметим, что на самом деле отрезок и круг не пересекаются в том и только в том случае, если центр круга лежит за пределами области, заштрихованной на рис. 5, которая состоит из прямоугольной полосы шириной $2r$ и двух полуокругов с радиусом r и с центрами в концах отрезка. Эта область порождает дополнительную проверку расстояния от центра круга до начала

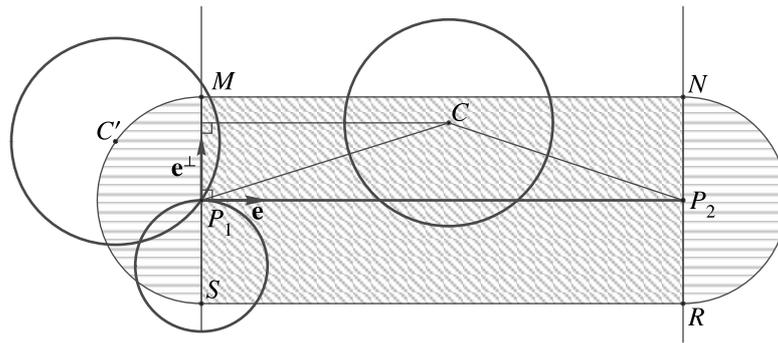


Рис. 5. Условие заблокированности отрезка кругом. При попадании центра круга в заштрихованную область круг пересекается с отрезком P_1P_2 и отрезок является заблокированным

и конца отрезка P_1P_2 . Однако в нашей задаче P_1P_2 — это отрезок касательной к одному из кругов, обозначающих препятствия, поэтому, если центр круга C находится внутри одного из полу кругов (включая его внешнюю границу), значит, круговые препятствия пересекаются (рис. 5), что по условию не допускается.

Таким образом, проверка условий (9) и (10) в нашей задаче достаточна.

Вычисление максимальной скорости на участках пути

В соответствии с моделью, на прямых участках пути максимальная скорость движения $v_{\max} = u_{\max}$. Время движения по соответствующему отрезку составляет

$$t_{\min} = \frac{l}{u_{\max}}. \tag{11}$$

На дугах окружностей максимальная скорость определяется следующим образом. Пусть робот движется по окружности против часовой стрелки. Тогда максимальное значение управляющего воздействия u_1 равно u_{\max} . При этом угловая скорость робота равна угловой скорости движения его центра по соответствующей окружности, при этом

$$\begin{aligned} v &= \frac{u_1 + u_2}{2}, \\ \omega &= \frac{u_1 - u_2}{b}, \\ v &= \omega r. \end{aligned} \tag{12}$$

Отсюда следует, что при $u_1 = u_{\max}$

$$u_2 = u_{\max} \frac{2r - b}{2r + b},$$

и максимальная скорость при движении по дуге составляет

$$v_{\max} = u_{\max} \frac{2r}{2r + b}.$$

Соответственно, время движения по дуге окружности с радиусом r с углом φ составит

$$t_{\min} = \frac{l}{v_{\max}} = \frac{r\varphi}{v_{\max}} = \frac{r\varphi(2r + b)}{2ru_{\max}} = \frac{\varphi(2r + b)}{2u_{\max}}. \tag{13}$$

Время, которое робот тратит для поворота на месте на угол φ , можно определить из последнего соотношения в (12). Максимальная возможная угловая скорость поворота равна

$$\omega_{\max} = \frac{2u_{\max}}{b},$$

и, таким образом, минимальное время такого поворота составит

$$t_{\min} = \frac{\varphi b}{2u_{\max}}, \quad (14)$$

что, если сравнить с (13), соответствует движению по окружности с нулевым радиусом.

Эти соотношения используются при расчете времени движения по ребрам графа.

Обратим внимание, что для рассматриваемой кинематической модели время движения как по прямым участкам, так и по дугам окружностей прямо пропорционально длине пути и обратно пропорционально максимальному значению, ограничивающему управляющие воздействия в системе. Это означает, что, за исключением поворотов на месте, самый быстрый путь из начальной точки в конечную, в случае если точки лежат на границах препятствий, будет совпадать с одним из локальных минимумов в задаче поиска кратчайшего пути в геометрическом смысле. Выбор такого локального минимума зависит от геометрического параметра b . Докажем следующее предложение.

Предложение 1. *Для кинематической модели движения двухколесного робота оптимальным по быстродействию путем будет локально кратчайший в геометрическом смысле путь с добавлением поворотов в начальной и конечной точках.*

Доказательство. Рассмотрим две окружности 1 и 2 с центрами в точках C_1 и C_2 и радиусами r_1 и r_2 соответственно. Рассмотрим путь из точки D , принадлежащей окружности 1, до точки B , принадлежащей окружности 2. Пусть для определенности точки C_1 и C_2 лежат на горизонтальной прямой. Этот путь проходит от точки D по дуге первой окружности до точки A , в которой общая касательная к двум окружностям AB касается первой из них, а затем — по отрезку AB до касания со второй окружностью. Пусть для определенности точки C_1 и C_2 лежат на горизонтальной прямой, расстояние между центрами окружностей $\|C_1C_2\|$ равно d , а угол между касательными к окружности 1 в точке D и в точке A равен ψ (рис. 6).

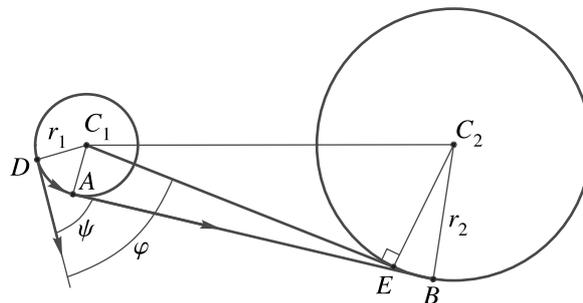


Рис. 6. К доказательству предложения 1. При стремлении радиуса r_1 к нулю время движения стремится к времени, затрачиваемому на поворот и на прямолинейное движение к точке касания

Такой путь является кратчайшим с геометрической точки зрения. Как было отмечено выше, этот путь, проходящий по дуге окружности и отрезку прямой, также является и быстрее, поскольку среди локальных минимумов по длине пути время, затрачиваемое на этот путь, пропорционально длине дуги, проходимой по окружности 1. Время движения по такому пути равно

сумме времен движения по дуге t_a и по отрезку t_l и составляет, в соответствии с (13) и (11),

$$t = t_a + t_l = \frac{\psi(2r_1 + b)}{2u_{\max}} + \frac{\sqrt{d^2 - (r_2 - r_1)^2}}{u_{\max}}. \tag{15}$$

Величина $\sqrt{d^2 - (r_2 - r_1)^2}$ является длиной отрезка AB ; для получения этого выражения обратимся к рис. 3. Угол дуги DA равен ψ .

В выражении (15) перейдем к пределу при $r_1 \rightarrow 0$. Точки A и D при этом будут стремиться к точке C_1 , точка B — к точке E , где C_1E — касательная к окружности 2, а угол ψ — к углу φ (рис. 6) между начальным направлением и направлением BE . Первое слагаемое при этом обратится в

$$t_a \rightarrow \frac{\varphi b}{2u_{\max}}, \tag{16}$$

а второе — в

$$t_l \rightarrow \frac{\sqrt{l^2 - r_2^2}}{u_{\max}}. \tag{17}$$

При этом, поскольку суммарное время $t = t_a + t_l$ является минимальным при любом радиусе r_1 , в пределе при $r_1 = 0$ оно также останется минимальным для всех возможных движений. □

Время (16) является временем поворота на месте в точке C_1 на угол φ , а время (17) — временем движения по касательной из точки C_1 до точки E , принадлежащей окружности 2. Таким образом, мы доказали, что оптимальным по быстродействию движением из точки с произвольной ориентацией корпуса робота до точки касания на окружности является поворот на месте до направления движения по касательной к окружности, а затем прямолинейное движение по этой касательной.

Аналогичным образом можно доказать оптимальность составного движения, которое включает в себя повороты на месте и прямолинейный участок, для движения из одной точки в другую без препятствий между ними, перейдя в выражении (15) к пределу при $r_1, r_2 \rightarrow 0$.

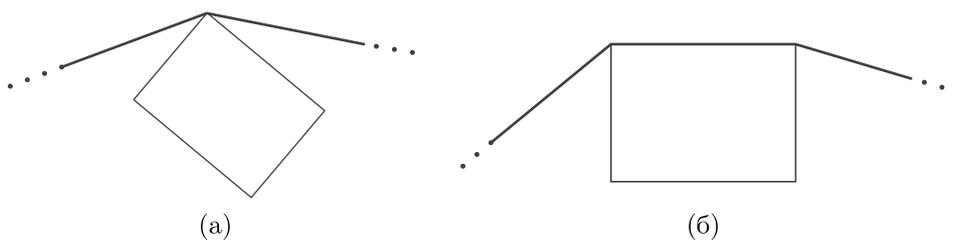


Рис. 7. Кратчайший путь, обходящий полигональное препятствие. Время движения складывается из времени движения по левому участку, по правому участку и времени поворота

В случае если допускаются полигональные препятствия на пути робота, при этом геометрически кратчайший путь проходит через угол полигона (рис. 7, а) или по грани полигона (рис. 7, б), верно следующее

Следствие 1. В случае полигональных препятствий кратчайший путь также является одним из локальных минимумов в задаче поиска кратчайшего пути.

Для доказательства достаточно разбить путь на участки, проходящие между углами полигональных препятствий, через которые проходит кратчайший путь. В силу доказанного выше

утверждения каждый такой участок будет оптимальным с точки зрения быстродействия при движении робота от начала участка к его концу, при этом в конечной точке производится поворот на такой угол, при котором новое направление движения робота совпадает с продолжением движения по следующему участку. Этот поворот производится вокруг вершины полигонального препятствия.

Следствие 2. В случае препятствий, заданных в виде произвольных гладких выпуклых фигур, кратчайший путь также является одним из локальных минимумов в задаче поиска кратчайшего пути.

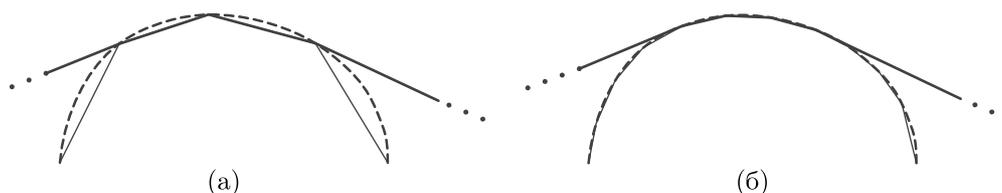


Рис. 8. Кратчайший путь, обходящий гладкое выпуклое препятствие. Время его обхода складывается из времени движения по прямолинейному отрезку, длина которого равна длине участка границы препятствия, и времени поворота из начальной ориентации к конечной

Для доказательства этого следствия рассмотрим полигональное приближение произвольного выпуклого препятствия. Оптимальная траектория будет проходить по углам и ребрам такой фигуры (рис. 8, а). При увеличении числа углов полигона, приближающего гладкое препятствие, такая траектория остается оптимальной. Следовательно, в предельном переходе при количестве углов приближающего полигона, стремящемся к бесконечности, оптимальная траектория будет стремиться к геометрически кратчайшей траектории, обходящей гладкое препятствие (рис. 8, б).

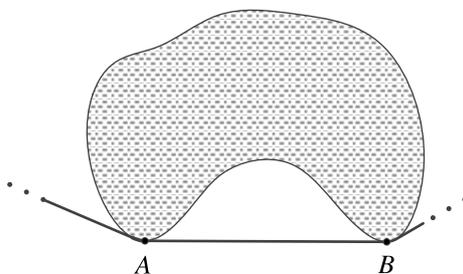


Рис. 9. Кратчайший путь, обходящий невыпуклое препятствие. Время обхода считается так же, как если бы это препятствие было заменено на его выпуклую оболочку

Что касается невыпуклых препятствий, рассмотрим, например, локально кратчайший геометрический путь, обходящий невыпуклую часть препятствия на рис. 9. Точки A и B — крайние точки границы препятствия, между которыми путь идет по прямой. В отсутствие препятствия этот участок, как было отмечено, является оптимальным с точки зрения быстродействия. Наложение любых дополнительных фазовых ограничений не улучшает оптимальных траекторий, поэтому при наличии препятствия участок AB остается оптимальным. Следовательно, и в случае невыпуклых препятствий кратчайший путь является быстреешим, а все невыпуклые препятствия можно в некоторых случаях для поиска оптимальной траектории заменить на их выпуклые оболочки. Такая замена, однако, невозможна, если существуют препятствия, которые не пересекаются друг с другом, при этом их выпуклые оболочки пересекаются. Типичным примером является лабиринт.

Таким образом, доказано, что для кинематической модели движения робота при наличии произвольных препятствий оптимальным по быстродействию является один из локальных минимумов в задаче поиска кратчайшего пути, при этом в точках перегиба пути оптимальным движением является поворот на месте вокруг точки перегиба. Выбор локального минимума зависит от геометрического параметра b робота, при этом быстрейший путь может не совпадать с геометрически кратчайшим, однако принцип поиска такого оптимального пути сохраняется. Следовательно, мы можем определить быстрейший путь методом перебора возможных путей на графе.

Построение графа переходов

Оптимальный путь, соединяющий две заданные точки и обходящий непересекающиеся круговые препятствия, состоит из отрезков прямых и дуг окружностей, при этом отрезки и дуги чередуются. В начале и в конце траектории производятся повороты на месте.

Для поиска оптимального пути составим граф Γ переходов между точками, лежащими на возможных оптимальных траекториях. Этими точками являются начало пути и конец пути, а также точки, лежащие на окружностях, которые ограничивают препятствия.

Все переходы следует добавлять с проверкой, является ли данный переход свободным или заблокированным каким-либо препятствием.

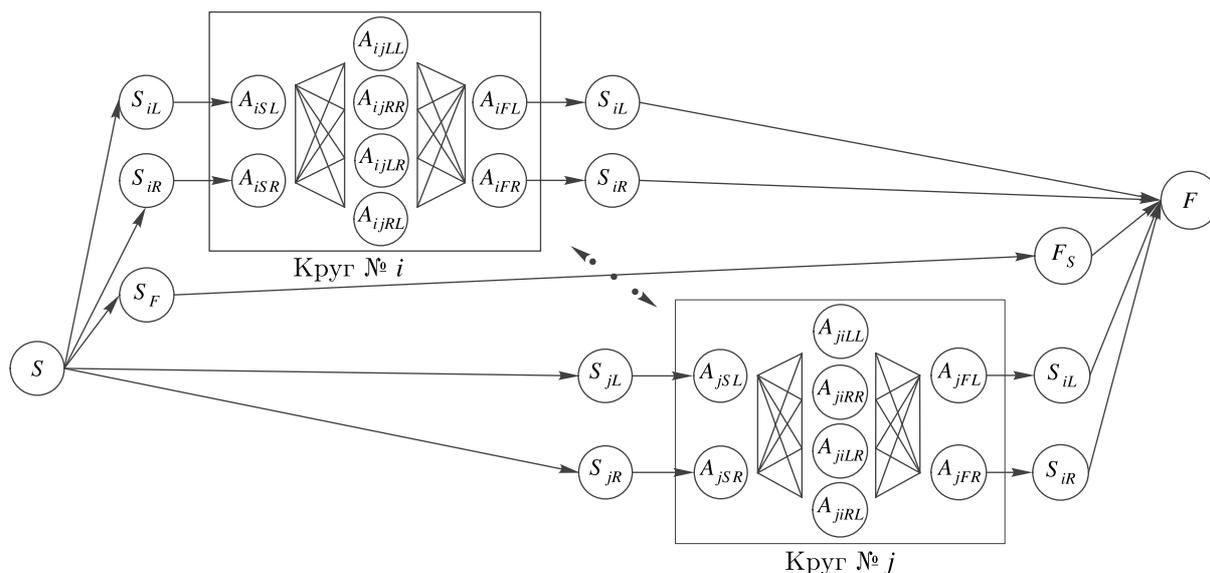


Рис. 10. Граф переходов между начальной и конечной точками и окружностями. Показаны узлы графов, схематично отмечены соединения вершин, относящиеся к одной и той же окружности, а также соединения между окружностями

Переход между точкой старта и точкой финиша с учетом поворота робота до необходимого направления из изначально заданного, а также конечного поворота в точке финиша до заданной терминальной ориентации добавим как путь из трех вершин. Обозначим точку старта через S ; точку финиша — через F ; вершину, соответствующую точке старта после поворота до направления движения к финишу, — через S_F ; вершину, соответствующую концу этой прямой до поворота к терминальному направлению, — через F_S (рис. 10).

Ребра графа, соответствующие поворотам на месте, имеют нулевую длину, однако время, требующееся для их прохождения (вес соответствующего ребра), определяется соотношением (14). При этом самым быстрым будет такой поворот в том направлении из двух возможных —

по часовой стрелке или против часовой стрелки, — для которого угол между начальной и конечной ориентацией меньше.

Переходы между началом пути и окружностями определим как прямые, соединяющие точку старта и точки, в которых касательные к соответствующему кругу и проходящие через точку старта касаются этого круга. Обозначим соответствующие точки на окружностях через A_{iSL} и A_{iSR} и зададим одноименные вершины графа. В этих обозначениях i задает номер окружности, индекс S обозначает, что прямая соединяется с точкой старта, индексы L и R — что прямые подходят к окружности соответственно слева и справа, если смотреть со стороны окружности на точку старта. Аналогичным образом определим прямые, соединяющие окружности и точку финиша, обозначив их как A_{iFL} и A_{iFR} . При этом необходимо учитывать, что поворот робота до его движения по соответствующей прямой, а также время, затрачиваемое на этот поворот, зависят от направления движения по этой прямой, поэтому необходимо добавить вершины графа Γ , соответствующие поворотам из начальной ориентации робота до соответствующего направления (рис. 10). Эти вершины обозначим через S_{iL} и S_{iR} для точки старта и через F_{iL} и F_{iR} для точки финиша. Будем обозначать левую и правую стороны для точки старта и точки финиша, смотря на точку со стороны окружности.

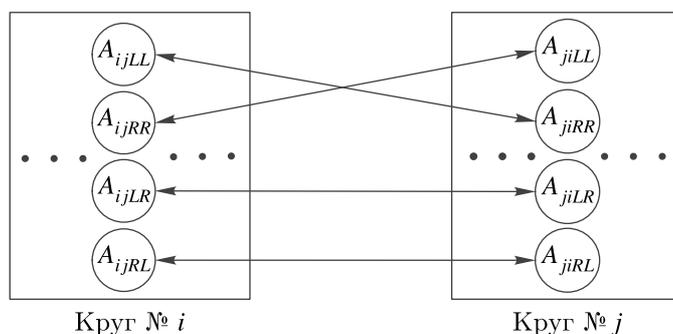


Рис. 11. Участки графа переходов между двумя окружностями

Переходы между точками двух различных окружностей определяются общими касательными к этим окружностям. Без исключения заблокированных отрезков каждая пара окружностей добавляет четыре вершины, соответствующие точкам касания на одной из этих окружностей, и четыре — на другой. Точки касания (и вершины графа Γ) будем обозначать как A_{ijLL} , A_{ijRR} , A_{ijLR} , A_{ijRL} , где индексы i , j обозначают порядковые номера окружностей в списке, а L и R — стороны (левую и правую), с которой касательные подходят к одной и к другой окружностям. Эти точки соединены четырьмя отрезками касательных, которые задают ребра графа Γ (рис. 11).

После определения всех точек касания, принадлежащих окружностям, для каждой из этих окружностей зададим переходы между всеми такими точками. Точки на каждой окружности можно разделить на два типа (рис. 12). В точках первого типа касательная подходит слева, если смотреть из центра окружности (точки с номерами 1 и 4 на рис. 12). В точках второго типа (2 и 3) касательная подходит справа. При этом если траектория робота приходит на окружность в точке типа 1, то она должна выйти с окружности в точке типа 2, и наоборот. Траектории, содержащие переход из одной точки в другую точку такого же типа, заведомо не являются оптимальными, поскольку содержат точки разворота (рис. 13, путь 3). Оптимальная траектория не должна содержать разворотов, поэтому движение по окружности из точки типа 1 в точку типа 2 должно производиться по часовой стрелке, а из точки типа 2 в точку типа 1 — против часовой стрелки (рис. 13, путь 2). Однако в некоторых случаях, а именно, когда угол дуги, которую необходимо пройти по окружности, больше π , неоптимальная траектория с разворотом оказывается короче (и быстрее); см. путь 1 на рис. 13. Это означает, что если от точки входа на окружность до точки

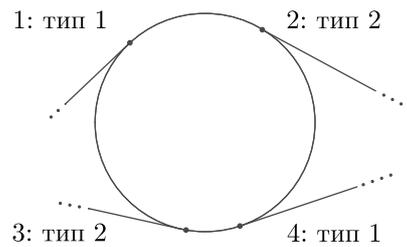


Рис. 12. Два типа точек касания на окружности: тип 1 — касательная подходит «слева», тип 2 — касательная подходит «справа»

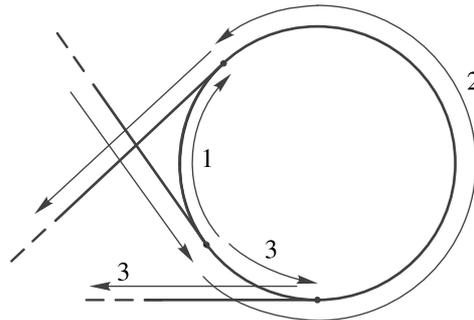


Рис. 13. Возможные участки траектории, заведомо не являющиеся оптимальными. Траектория подходит из левого верхнего угла и уходит в левом нижнем углу. Путь 1 через точки A и C и путь 3 через точки A и B содержат развороты, путь 2 длиннее, чем путь 1

выхода с нее необходимо пройти угол, больший π , то такая траектория заведомо не является оптимальной, и при составлении графа переходов такие дуги можно не включать в граф.

Отметим, что в принятых обозначениях точками типа 1 являются все точки, принадлежащие окружностям, которые имеют в третьей позиции индекса литеру R , а точками типа 2 — те, которые в этом индексе имеют литеру L .

Итак, опишем общий вид алгоритма для построения графа переходов.

Алгоритм

Шаг 1. Проверить, заблокирован ли отрезок SF .

Если отрезок SF не заблокирован:

- добавить точки S_F и F_S ;
- добавить переход типа «поворот» от S до S_F ;
- добавить переход типа «прямая» от S_F до F_S ;
- добавить переход типа «поворот» от F_S до F ;
- конец.

Иначе перейти к шагу 2.

Шаг 2. Для каждой окружности № i из списка:

- найти касательные к окружности № i , проходящие через точку S ;
- для каждого из отрезков SA_{iSR} , SA_{iSL} найденных левой и правой касательных проверить, заблокирован ли этот отрезок.

Если отрезок не заблокирован:

- добавить к графу вершину S_{iR} или S_{iL} для левой или правой касательной соответственно;
- добавить соответствующую вершину A_{iSR} или A_{iSL} ;
- добавить переход типа «поворот» от S до S_{iR} или S_{iL} ;
- добавить переход типа «прямая» от S_{iR} до A_{iSR} или от S_{iL} до A_{iSL} соответственно.

Шаг 3. Для каждой окружности № i из списка:

- найти касательные к окружности № i , проходящие через точку F ;
- для каждого из отрезков $A_{iSR}F$, $A_{iSL}F$ найденных левой и правой касательных проверить, заблокирован ли этот отрезок.

Если отрезок не заблокирован:

- добавить к графу вершину F_{iR} или F_{iL} для левой или правой касательной соответственно;
- добавить соответствующую вершину A_{iFR} или A_{iFL} ;
- добавить переход типа «поворот» от F_{iR} или F_{iL} до F ;
- добавить переход типа «прямая» от A_{iFR} до F_{iR} или от A_{iFL} до F_{iL} соответственно.

Шаг 4. Для каждой окружности № i из списка:

для каждой окружности № j из списка, $j > i$:

- найти все общие касательные между окружностями № i и № j ;
- Для каждой найденной касательной проверить, заблокирован ли отрезок этой касательной между точками касания с окружностями.

Если отрезок не заблокирован:

- * добавить в граф вершину, соответствующую точке касания на окружности № i : A_{ijLL} , A_{ijRR} , A_{ijLR} или A_{ijRL} ;
- * добавить в граф вершину, соответствующую точке касания на окружности № j : A_{jiRR} , A_{jiLL} , A_{jiLR} , A_{jiRL} ;
- * Добавить переход типа «прямая» между добавленными точками.

Шаг 5. Для каждой окружности № i из списка:

- найти в списке вершин графа все вершины, соответствующие точкам на данной окружности (т. е. у которых первый индекс равен i);
- разделить найденные точки на два типа — тип 1 и тип 2; к первому типу отнести все точки со значением индекса R (RR , RL), ко второму — со значением индекса L (LR , LL);
- для каждой точки A из множества точек первого типа:

для каждой точки B из множества точек второго типа проверить угол при движении по дуге AB по часовой стрелке; если этот угол меньше π :

- * добавить переход типа «дуга» из точки A в точку B с движением по часовой стрелке;
- * добавить переход типа «дуга» из точки B в точку A с движением против часовой стрелки.

■ Конец.

В алгоритме в первую очередь проверяется возможность перейти из точки старта непосредственно в точку финиша. Если этот участок пути не заблокирован препятствиями, значит, это и будет кратчайший путь, и дальнейших построений не требуется. В противном случае для определения оптимальной траектории необходимо построить весь граф.

Нахождение оптимальной траектории

Для нахождения оптимальной траектории с помощью построенного графа Γ будем использовать алгоритм Дейкстры поиска кратчайших путей на графе из заданной начальной вершины в его модификации, когда требуется определить кратчайший путь до другой заданной вершины.

Приведем краткое описание алгоритма [Dijkstra, 1959]. Для всех вершин v_i , кроме стартовой, длины пути устанавливаются в $+\infty$ или большое положительное число, заведомо большее любой возможной длины пути. Вершинам присваивается отметка «посещенная» или «непосещенная», причем в начале работы все вершины имеют отметку «непосещенная». Из всех непосещенных вершин выбирается та v_i , для которой длина рассчитанного пути наименьшая, и для каждой соседней с ней вершины v_j рассчитывается длина пути, складывающаяся из рассчитанной длины для данной вершины $l(v_i)$ и веса соответствующего ребра, соединяющего v_i и v_j . Если для v_j рассчитанная таким образом длина меньше, чем сохраненная, то сохраняется новая длина, а «предшествующей» v_j вершиной назначается v_i . Вершина получает отметку «посещенная», когда для нее проверены таким образом все ее соседние непосещенные вершины. Алгоритм прерывается, когда заданная финишная вершина становится посещенной.

Кратчайший путь при этом можно восстановить, пройдя от финишной вершины к стартовой в обратном порядке, используя записанные индексы предшествующих вершин. В качестве длин можно выбрать геометрическую длину соответствующего участка либо время, затрачиваемое на прохождение этого участка. В первом случае получим кратчайшую траекторию, соединяющую начальную и конечную точки, и ее длину; во втором — минимальное время, необходимое для перемещения из начальной точки в конечную.

Вычислительная сложность данного алгоритма для графа Γ , имеющего N вершин, оценивается в $O(N^2)$ операций. Для графа, построенного в нашем случае, при числе окружностей M число вершин графа будет равно (без учета блокирования отрезков, при котором вершины не добавляются в граф) $2 + 2$ (точка старта и точка финиша, а также точки поворота), $+ 2M + 2M$ точек касания прямых, проходящих через точку старта, с соответствующими точками поворота $+ 2M + 2M$ аналогичных точек для финиша, $+ 4M(M - 1)$ общих точек касания попарно для всех касательных к парам окружностей. Таким образом, граф Γ имеет максимум

$$N = 4 + 8M + 4M(M - 1) = O(4M^2)$$

вершин, а вычислительная сложность пропорциональна $O(16M^4)$.

Проверка достижимости заданной конечной точки графа производится параллельно с определением кратчайшего пути. Если в какой-то момент оказывается, что минимальная длина для непосещенной вершины равна $+\infty$, значит, все достижимые из точки старта вершины уже посещены и заданная точка финиша недостижима. На практике это означает, что если при нахождении кратчайшего пути на графе в результате работы алгоритма Дейкстры длина пути до заданной точки равна $+\infty$, значит, заданная точка недостижима. При описанном способе построения графа переходов для непересекающихся препятствий (и если начальная и конечная точки не находятся внутри препятствия) такая ситуация невозможна.

Для поиска кратчайшего пути на графе, помимо алгоритма Дейкстры, можно пользоваться другими методами, например набирающим популярность методом A^* [Hart et al., 1968], который может находить кратчайший путь несколько быстрее, чем метод Дейкстры.

4. Вычислительный эксперимент

Для реализации описанного алгоритма поиска кратчайшего пути была написана программа на языке Python, автоматически считывающая задаваемый файл настроек с заданием начальной и конечной точек, а также окружностей, ограничивающих препятствия. В файле настроек также можно задавать ограничение на управляющие воздействия u_{\max} . Программа автоматически рассчитывает все необходимые касательные, определяет заблокированные отрезки и строит граф переходов, добавляя в него необходимые вершины и ребра.

Все точки в программе задаются тремя значениями. Первые два соответствуют декартовым координатам точки на плоскости. Третье значение определяет угол ориентации робота при его прохождении через данную точку. Так, точка старта в качестве третьего значения будет иметь начальную ориентацию робота, точка финиша — конечную ориентацию.

На выходе программа сохраняет файл, в котором описывается путь в виде последовательности точек, а также производится расшифровка — указывается тип ребра (поворот, дуга окружности, отрезок прямой), необходимые параметры, определяющие соответствующую этому ребру геометрическую фигуру (для поворота — начальная точка, угол и направление поворота; для дуги — координаты центра и радиус окружности, координаты начальной и конечной точек, угловое направление движения; для отрезка прямой — координаты начала и конца отрезка), а также время движения по каждому ребру. Вычисляется и сохраняется также общее (оптимальное) время движения из точки старта в точку финиша.

В программе также можно указать режим поиска — кратчайшего пути или быстрого пути. Как уже было сказано выше, эти пути совпадают, за исключением начального и конечного поворотов, которые при определении длины дают нулевой вклад. В режиме поиска оптимальной длины выдается длина пути без учета ограничений u_{\max} .

При индексации точек в качестве обозначений индексов, соответствующих старту и финишу, литеры S и F повторяются для того, чтобы длины текстовых идентификаторов точек были одинаковыми.

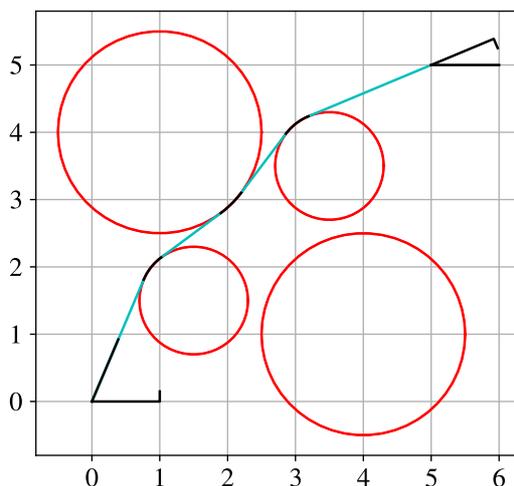


Рис. 14. Оптимальная траектория для четырех препятствий

Пример работы программы приведен на рис. 14. Заданы следующие настройки:

- точка старта: $(x = 0, y = 0, \varphi = 0)$;
- точка финиша: $(x = 5, y = 5, \varphi = 0)$;

- окружность 1: $(x_c = 1,5, y_c = 1,5, r = 0,8)$;
- окружность 2: $(x_c = 3,5, y_c = 3,5, r = 0,8)$;
- окружность 3: $(x_c = 1, y_c = 4, r = 1,5)$;
- окружность 4: $(x_c = 4, y_c = 1, r = 1,5)$;
- ограничение $u_{\max} = 1$;
- расстояние между колесами $b = 1$.

В результате расчетов получены следующие значения для оптимальной траектории.

Оптимальный путь определен как $S \rightarrow S_{0L} \rightarrow A_{0SL} \rightarrow A_{02LR} \rightarrow A_{20LR} \rightarrow A_{21RL} \rightarrow A_{12RL} \rightarrow A_{1FL} \rightarrow F_{1L} \rightarrow F$. Участки движения при этом следующие:

- 1) поворот из точки $(0; 0; 0)$ на угол $1,172$ против часовой стрелки, время движения — $0,586$;
- 2) прямая из точки $(0; 0; 1,172)$ в точку $(0,762; 1,810; 1,172)$, время движения — $1,964$;
- 3) дуга окружности с параметрами $(1,5; 1,5; 0,8)$ из точки $(0,762; 1,810; 1,172)$ в точку $(1,02; 2,14; 0,643)$ по часовой стрелке, время движения — $0,687$;
- 4) прямая из точки $(1,02; 2,14; 0,643)$ в точку $(1,9; 2,8; 0,643)$, время движения — $1,1$;
- 5) дуга окружности с параметрами $(1,0; 4,0; 1,5)$ из точки $(1,9; 2,8; 0,643)$ в точку $(2,2; 3,1; 0,927)$ против часовой стрелки, время движения — $0,567$;
- 6) прямая из точки $(2,2; 3,1; 0,927)$ в точку $(2,86; 3,98; 0,927)$, время движения — $1,1$;
- 7) дуга окружности с параметрами $(3,5; 3,5; 0,8)$ из точки $(2,86; 3,98; 0,927)$ в точку $(3,189; 4,237; 0,398)$ по часовой стрелке, время движения — $0,687$;
- 8) прямая из точки $(3,189; 4,237; 0,398)$ в точку $(5,0; 5,0; 0,398)$, время движения — $1,964$;
- 9) поворот из точки $(5,0; 5,0; 0,398)$ на угол $0,398$ по часовой стрелке, время движения — $0,199$.

Общее время движения — $8,856$ единиц.

Также в качестве примера приведем построение оптимального пути для большего количества препятствий. Найденный оптимальный путь приведен на рис. 15. Ниже приведены (без расшифровки) файл настроек и файл отчета.

5. Заключение

В работе описан созданный и реализованный в виде законченной программы алгоритм автоматического поиска оптимального пути между двумя заданными точками с набором фазовых ограничений в виде кругов с различными радиусами, которые не пересекаются между собой.

Исследование кинематической модели показывает, что, несмотря на то что в общем виде кратчайший с геометрической точки зрения путь не совпадает с быстрейшим путем при движении из одной заданной точки в другую, тем не менее он совпадает с одним из локальных минимумов в мультимодальной (имеющей несколько локальных минимумов) задаче поиска кратчайшего пути. Это означает, что для поиска быстрейшего движения здесь можно пользоваться теми же методами, что и для поиска кратчайшего пути, а именно поиском оптимального пути

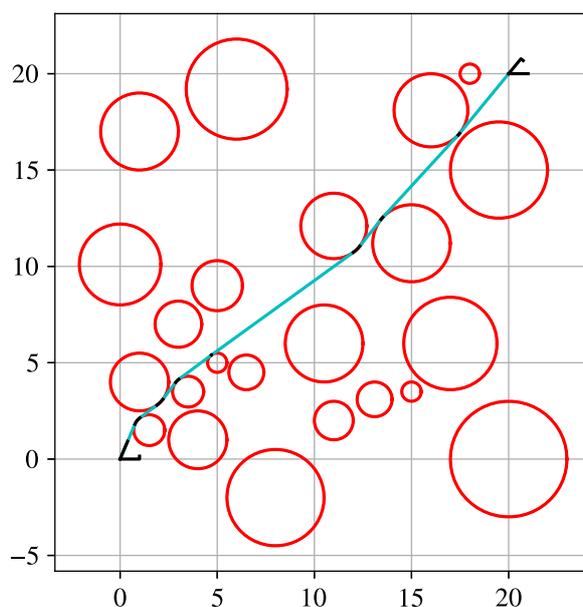


Рис. 15. Оптимальная траектория для 23 препятствий

на графе возможных переходов, с той разницей, что для оценки «длины» этого пути необходимо вычислять время движения по соответствующим отрезкам.

Данное свойство связано со структурой модели и определением управляющих воздействий в ней. Такая модель является достаточно сильно упрощенной, поскольку использует управление скоростями колес и не учитывает динамику переходных процессов, а управляющие воздействия кусочно-постоянны. Тем не менее во многих задачах робототехники такая упрощенная модель оказывается полезной, например при исследовании движения и планировании пути для гусеничных роботов.

Дальнейшее развитие алгоритма позволит учитывать препятствия произвольной формы, в том числе с негладкой границей и невыпуклые. Похожие алгоритмы в настоящее время используются в компьютерных играх, однако там используется поиск кратчайшего пути (в том числе с обходом полигональных препятствий) и не уделяется внимания быстрдействию.

Кроме того, в связи с возросшим интересом к применению методов численной оптимизации к задачам оптимального управления результаты данной работы могут быть интересны с точки зрения оценки качества тех или иных методов оптимизации, поскольку позволяют найти абсолютный минимум времени движения и сравнить его с результатами, получаемыми численно, на такой модельной задаче.

Список литературы (References)

- Дарьина А. Н., Прокопьев И. В. Метод оптимизации параметров контроллера беспилотного транспортного средства на основе оптимизации роя частиц // Надежность и качество сложных систем. Пенза: ПензГУ. — 2020. — № 3 (31). — С. 80–87.
- Daryina A. N., Prokopyev I. V. Metod optimizatsii parametrov kontrollera bespilotnogo transportnogo sredstva na osnove optimizatsii roya chastits [A method of optimization of parameters of controller of an autonomous vehicle based on particle swarm optimization] // Nadezhnost' i kachestvo slozhnyh sistem. Penza: PenzGU. — 2020. — No. 3 (31). — P. 80–87 (in Russian).
- Дивеев А. И., Константинов С. В. Исследование практической сходимости эволюционных алгоритмов оптимального программного управления колесным роботом // Известия РАН: Теория и системы управления. — 2018. — № 4. — С. 80–106.

- Diveev A. I., Konstantinov S. V.* Issledovanie prakticheskoy shodimosti evolyutsionnykh algoritmov optimal'nogo programmnoy upravleniya kolesnym robotom [Research on practical convergence of evolutionary algorithms of control of a wheeled robot] // *Izvestiya RAN: Teoriya i sistemy upravleniya*. — 2018. — No. 4. — P. 80–106 (in Russian).
- Дивеев А. И., Мендес Флорес Н. Х.* Синтез системы пространственной стабилизации мобильного робота на основе обучения методом символьной регрессии // *Вестник Российского университета дружбы народов. Сер. Инженерные исследования*. — 2021. — Т. 22, № 2. — С. 129–138.
- Diveev A. I., Mendez Florez N. J.* Sintez sistemy prostranstvennoy stabilizatsii mobil'nogo robota na osnove obucheniya metodom simvol'noy regressii [Synthesis of spatial stabilization of a mobile robot based on symbolic regression learning method] // *Vestnik RUDN. Ser. Inzhenernye issledovaniya*. — 2021. — Vol. 22, No. 2. — P. 129–138 (in Russian).
- Грачев Н. И., Евтушенко Ю. Г.* Библиотека программ для решения задач оптимального управления // *Журнал вычислительной математики и математической физики*. — 1979. — Т. 19, № 2. — С. 367–387.
- Grachev N. I., Evtushenko Yu. G.* Biblioteka programm dlya resheniya zadach optimal'nogo upravleniya [A collection of programs for solving optimal control problems] // *Computational Mathematics and Mathematical Physics*. — 1979. — Vol. 19, No. 2. — P. 367–387 (in Russian).
- Понтрягин Л. С., Болтянский В. Г., Гамкрелидзе Р. В., Мищенко Е. Ф.* Математическая теория оптимальных процессов. — М.: Наука, 1962. — 393 с.
- Pontryagin L. S., Boltyanskiy V. G., Gamskrelidze R. V., Mischenko E. F.* Matematicheskaya teoriya optimal'nykh processov [Mathematical theory of optimal processes]. — Moscow: Nauka, 1962. — 393 p. (in Russian).
- Понтрягин Л. С.* Принцип максимума в оптимальном управлении. — М.: Наука, 1989. — 64 с.
- Pontryagin L. S.* Printsip maksimuma v optimal'nom upravlenii [Maximum principle in optimal control]. — Moscow: Nauka, 1989. — 64 p. (in Russian).
- Федоренко Р. П.* Приближенное решение задач оптимального управления. — М.: Наука, 1978. — 488 с.
- Fedorenko R. P.* Priblizhennoe resheniye zadach optimal'nogo upravleniya [Approximate solving of optimal control problems]. — Moscow: Nauka, 1978. — 488 p. (in Russian).
- Bereznev V. A.* The principle of dividing feasible trajectories in a robot control problem // *Procedia Computer Science*. — 2021. — P. 456–459.
- Dijkstra E. W.* A note on two problems in connection with graphs // *Numerische Mathematik*. — 1959. — No. 1. — P. 269–271.
- Hart P. E., Nilsson N. J., Raphael B.* A formal basis for the heuristic determination of minimum cost paths // *IEEE Transactions on Systems Science and Cybernetics*. — 1968. — Vol. 4, No. 2. — P. 100–107.
- Karaman S., Frazzoli E.* Sampling-based algorithms for optimal motion planning // *The International Journal of Robotics Research*. — 2011. — Vol. 30, No. 7. — P. 846–894.
- Kavraki L. E., Svestka P., Latombe J. C., Overmars M. H.* Probabilistic roadmaps for path planning in high-dimensional configuration spaces // *IEEE Transactions on Robotics and Automation*. — 1996. — Vol. 12, Vol. 4. — P. 566–580.
- LaValle S. M.* Rapidly-exploring random trees: A new tool for path planning. — Technical Report TR98-11. — Department of Computer Science, Iowa State University, 1998. — 4 p.
- Nocedal J., Wright S. J.* Numerical optimization. — 2nd ed. — Springer, 2006. — 651 p.
- Puterman M. L.* Markov decision processes: discrete stochastic dynamic programming. — John Wiley & Sons. — 1994. — 649 p.
- Stentz A.* The focussed D* algorithm for real-time replanning // *Proceedings of the International Joint Conference on Artificial Intelligence*. — 1995. — P. 1652–1659.
- Sutton R. S., Barto A. G.* Reinforcement learning: an introduction. — MIT Press, 2018. — 352 p.