

Оригинальная статья / Original article

<https://doi.org/10.21869/2223-1560-2020-24-4-107-125>

Сглаживание кривизны траектории движения наземного робота в трехмерном пространстве

К.С. Захаров¹✉, А.И. Савельев¹

¹ Санкт-Петербургский институт информатики и автоматизации Российской академии наук (СПИИРАН)
14-я линия В.О. 39, г. Санкт-Петербург 199178, Российская Федерация

✉ e-mail: kon7666007@yandex.ru

Резюме

Цель исследования. Разработка алгоритма сглаживания траектории движения наземного робота по пересеченной местности, представленной в виде графа в трехмерном пространстве.

Методы. В данной статье представлен алгоритм CSA (Curve Smoothing and Averaging) для сглаживания на плоскости Оху локальных кривых, составляющих глобальную кривую, представленную в виде пути на связном графе в трехмерном пространстве. Представленный алгоритм основан на разработанном ранее подходе LRLHD-A*, в котором используется информация о вершинах графа, их соседях и соединяющих их ребрах для выбора области, через которую будет пролегать сглаженная кривая. Во избежание на выходе алгоритма ломаной кривой был разработан метод усреднения кривой, идея которого заключается в смещении серединных точек локальных кривых, вдоль ребер, на которых они находятся.

Результаты. Было произведено экспериментальное сравнение кривизны траекторий, получаемых с помощью алгоритма сглаживания кривой с усреднением кривой (CSA) и без него (CS). Тестирование метода проводилось на трехмерной карте местности, представленной в виде графа с 100082 вершинами. Для проведения экспериментов использовалась выборка из 10 пар случайных вершин, между которыми строился путь с помощью алгоритма LRLHD-A*. Результаты экспериментов показали, что усреднение кривой после сглаживания снижает ее кривизну от 24 до 42%.

Заключение. Траектории, сглаженные с помощью разработанного алгоритма CSA, имеют более плавные изгибы кривой на поворотах, по сравнению с алгоритмом, взятым за основу. Это позволяет добиться более плавного движения роботов и как следствие снижения расхода заряда аккумулятора робота.

Ключевые слова: планирование пути; автономная навигация; сглаживание траектории; трехмерное пространство; навигационный граф.

Конфликт интересов: Авторы декларируют отсутствие явных и потенциальных конфликтов интересов, связанных с публикацией настоящей статьи.

Финансирование. Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 19-33-90053.

Для цитирования: Захаров К.С., Савельев А.И. Сглаживание кривизны траектории движения наземного робота в трехмерном пространстве // Известия Юго-Западного государственного университета. 2020; 24(4): 107-125. <https://doi.org/10.21869/2223-1560-2020-24-4-107-125>.

Поступила в редакцию 28.09.2020

Подписана в печать 14.10.2020

Опубликована 30.12.2020

Smoothing the Curvature of Trajectory of Ground Robot in 3D Space

Konstantin S. Zakharov ¹ ✉, Anton I. Saveliev ¹

¹ St.Petersburg Institute for Informatics and Automation of the Russian Academy of Sciences (SPIIRAS)
39, 14-th Line V.O., St. Petersburg 199178, Russian Federation

✉ e-mail: kon7666007@yandex.ru

Abstract

Purpose or research. Development of an algorithm for smoothing the trajectory of a ground robot over rough terrain, represented as a graph in three-dimensional space.

Methods. This article presents the CSA (Curve Smoothing and Averaging) algorithm for smoothing local curves in the Oxy plane that make up a global curve, represented as a path on a connected graph in 3D space. The presented algorithm is based on the previously developed LRLHD-A * approach, which uses information about the vertices of the graph, their neighbors and the edges connecting them to select the area through which the smoothed curve will run. In order to avoid a broken curve at the output of the algorithm, a curve averaging method was developed, the idea of which is to shift the midpoints of local curves along the edges on which they are located.

Results. An experimental comparison was made of the curvature of the trajectories obtained using the curve smoothing algorithm with curve averaging (CSA) and without it (CS). The method was carried out on a three-dimensional map of the area, presented in the form of a graph with 100082 vertices. For the experiments, a sample of 10 pairs of random vertices was used, between which a path was built using the LRLHD-A * algorithm. The results of the experiments have shown that averaging the curve after smoothing reduces its curvature from 24 to 42%.

Conclusion. Trajectories smoothed using the developed CSA algorithm have smoother curve bends at turns, compared to the algorithm taken as a basis. This allows the robots to move more smoothly and, as a consequence, reduce the consumption of the robot's battery.

Keywords: path planning; self-navigation; smoothing the trajectory; three-dimensional space; navigation graph.

Conflict of interest. The authors declare the absence of obvious and potential conflicts of interest related to the publication of this article.

Funding: The research was carried out with the financial support of the Russian Foundation for Basic Research within the framework of scientific project No. 19-33-90053.

For citation: Zakharov K. S., Saveliev A. I. Smoothing the Curvature of Trajectory of Ground Robot in 3D Space. *Izvestiya Yugo-Zapadnogo gosudarstvennogo universiteta = Proceedings of the Southwest State University*. 2020; 24(4): 107-125 (In Russ.). <https://doi.org/10.21869/2223-1560-2020-24-4-107-125>.

Received 28.09.2020

Accepted 14.10.2020

Published 30.12.2020

Введение

В настоящее время наблюдается повышенный интерес к решению задач, связанных с автономной навигацией робототехнических комплексов (РТК). Это обусловлено высоким спросом на введение в действие РТК в сельскохо-

зяйственных целях, в целях исследования труднодоступных территорий, картографирования местности и т.д. В зависимости от особенностей местности, по которой будет передвигаться РТК, разрабатывается алгоритм планирования траектории его движения, в кото-

ром траектории представляются в виде множества точек, которые надо посетить РТК для достижения цели, либо в виде кривых, описываемых уравнениями. Существующие алгоритмы планирования траектории не всегда учитывают необходимость сглаживания траектории. Однако эта особенность позволяет сократить время, затрачиваемое роботом на совершение поворотов. В данной работе разработан алгоритм сглаживания траектории движения РТК на плоскости Оху, представленной в виде пути на связном графе в трехмерном пространстве.

Существует множество методов сглаживания кривой в двумерном пространстве. Например, с помощью метода кривых Дубинса [1-4], суть которого заключается в замене частей отрезков, прилегающих к углу ломаной, дугой окружности. Кривые Дубинса представляют собой простую, но в то же время эффективную, технику для сглаживания пути в режиме реального времени, поскольку их построение не требует больших вычислительных затрат. Кривые Дубинса также могут быть скомбинированы с клотоидами для удовлетворения различных ограничений. Также существуют методы сглаживания кривой с помощью кривых Безье [5-11], которые используют контрольные точки для определения формы кривой. Одними из популярных являются алгоритмы на основе обобщения кривых Безье – В-сплайнов [12-16]. В-сплайны

являются эффективным средством для сглаживания пути, так как желаемая траектория может быть сгенерирована для разных значений кривизны и конфигураций препятствий. Методы на основе неоднородных рациональных В-сплайнов (NURBS) [17-21] являются очень гибким инструментом для построения сглаженных траекторий за счёт того, что они манипулируют как контрольными точками, так и весами точек. Основными недостатками использования NURBS-кривых является большее количество используемой памяти в сравнении с аналогичными алгоритмами. Также неправильная инициализация весов точек может привести к плохой параметризации. Несмотря на это, использование NURBS-кривых является эффективной техникой сглаживания траектории благодаря четким геометрическим интерпретациям, уменьшения количества времени работы алгоритма по сравнению с аналогами и стабильностью при нахождении траектории. Существуют алгоритмы сглаживания на основе квадратичной интерполяции [22, 23], которые строят непрерывную кривую, используя квадратные многочлены и функции принадлежности. У алгоритмов на основе квадратичной интерполяции есть два основных недостатка: высокая вычислительная сложность и феномен Рунге, который является классической иллюстрацией несходимости полиномиальной интерполяции. Алгоритмы сглаживания пути

в двумерном пространстве не могут быть использованы для сглаживания пути в трехмерном пространстве, так как они не могут учитывать рельеф местности. Это является серьезным недостатком, так как траектории, построенные с помощью подобных алгоритмов, могут пролегать в областях с большими перепадами высот, что может сильно затруднять движение робота по данной траектории.

Помимо сглаживания кривой в двумерном пространстве также существуют методы сглаживания кривых в трехмерном пространстве. В работе [24] представлен алгоритм сглаживания пути на графе в трехмерном пространстве. Данный подход позволяет находить сглаженные траектории в трехмерном пространстве с различными значениями параметров кривизны выходной траектории и максимального расстояния до исходной кривой. Главным недостатком данного алгоритма являются ситуации появления ломаных линий в сглаженной траектории, что может значительно увеличить время движения робота до цели. Это связано с тем, что данный алгоритм учитывает кривизну локальной кривой, состоящей всего из трех точек. В статье [25] представлен алгоритм сглаживания трехмерной траектории движения робота по поверхности Дарбу [26], который последовательно оптимизирует траекторию в горизонтальной, фронтальной и профильной проекции с заранее заданными ограничениями по линейной скорости, угловой скорости и

ширине робота. Представленный подход включает в себя два уровня: планировщик неоптимальной геодезической кривизны и планировщик скорости, который использует набор кривых из предыдущего уровня для аналитического вычисления скорости на основе параметров безопасности робота и его кинематических ограничений, а также для создания графика измерения ускорения. Данный подход полезен для улучшения процесса строительства автомагистралей, он позволяет оценить их жизненный цикл за счет поиска оптимальных по времени путей. Однако данный подход не подходит для сглаживания траектории движения робота по неровной местности, так как трехмерная модель местности не может быть представлена в виде поверхности Дарбу.

В данной работе рассмотрена проблема сглаживания траектории движения в трехмерном пространстве. Для ее решения предложен алгоритм сглаживания траектории движения на графе в трехмерном пространстве, основанный на подходе, представленном в [24]. Особенностью разработанного алгоритма является возможность сглаживания кривых, представленных в виде пути на графе в трехмерном пространстве, что позволяет находить гладкую траекторию движения наземного робота по неровной местности. Также представленный алгоритм устраняет проблему, связанную с появлением резких изгибов на поворотах в плоскости Oxy .

Материалы и методы

А. Сглаживание траектории

В данной работе решается задача сглаживания траектории движения в трехмерном пространстве для различных моделей местности, представленных в виде связного графа. Исходными данными для этой задачи являются: связный граф, отображающий особенности местности, путь, представленный в виде упорядоченного множества вершин на связном графе, в котором каждая вершина соединена со следующей в последовательности вершиной (кроме последней), а также граничные значения параметров, определяющих максимальное расстояние, на которое могут быть смещены точки исходной кривой, и максимальную кривизну конечной кривой. Пусть $M \in \mathbb{R}^3$ – трехмерная сетка. Сетка состоит из вершин V с соответствующими позициями $x_i \in \mathbb{R}^3$ и ребер $E = \{(i, j) | i, j \in V\}$. Для решения задачи сглаживания траектории движения был разработан алгоритм итеративного сглаживания траектории, представленной в виде пути на связном графе в трехмерном пространстве, с усреднением конечной кривой. На рис. 1 представлена блок-схема алгоритма CSA.

Основой алгоритма является итеративное сглаживание кривой после её инициализации. Под инициализацией здесь понимается нахождение безколлизионной траектории движения с помощью алгоритмов планирования пути. После инициализации исходной траектории ее вершины добавляются в упо-

рядоченное множество P . Далее идёт вычисление кривизны текущей траектории P , и если она ниже определенного значения, то алгоритм заканчивает работу. В противном случае начинается процесс сглаживания траектории. Кривизна траектории определяется как наибольшая кривизна локальной кривой, состоящей из 3 точек, среди всех локальных кривых, входящих в глобальную кривую. Далее создаётся пустое множество P_n , в которое добавляется первая вершина из множества P , то есть начальную вершину исходного пути. После этого необходимо создать копию множества, содержащего текущий путь P_c , чтобы манипулировать новой траекторией, не влияя на старую. Далее из множества P_c извлекаются три первые вершины и добавляются в пустое множество C_i для составления локальной кривой, вершины которой будут смещаться. После создания локальной кривой C_i начинается процесс ее сглаживания. Если точка кривой находится на ребре графа, то она смещается в том или ином направлении вдоль ребра. В том случае, когда точка кривой лежит непосредственно на вершине графа, необходимо использовать особый подход. Особенностью данного алгоритма является то, что точки выходной кривой не проходят через вершины графа. Данный подход предполагает два варианта шага сглаживания для точки p_i : p_i находится на ребре, p_i находится на вершине графа.

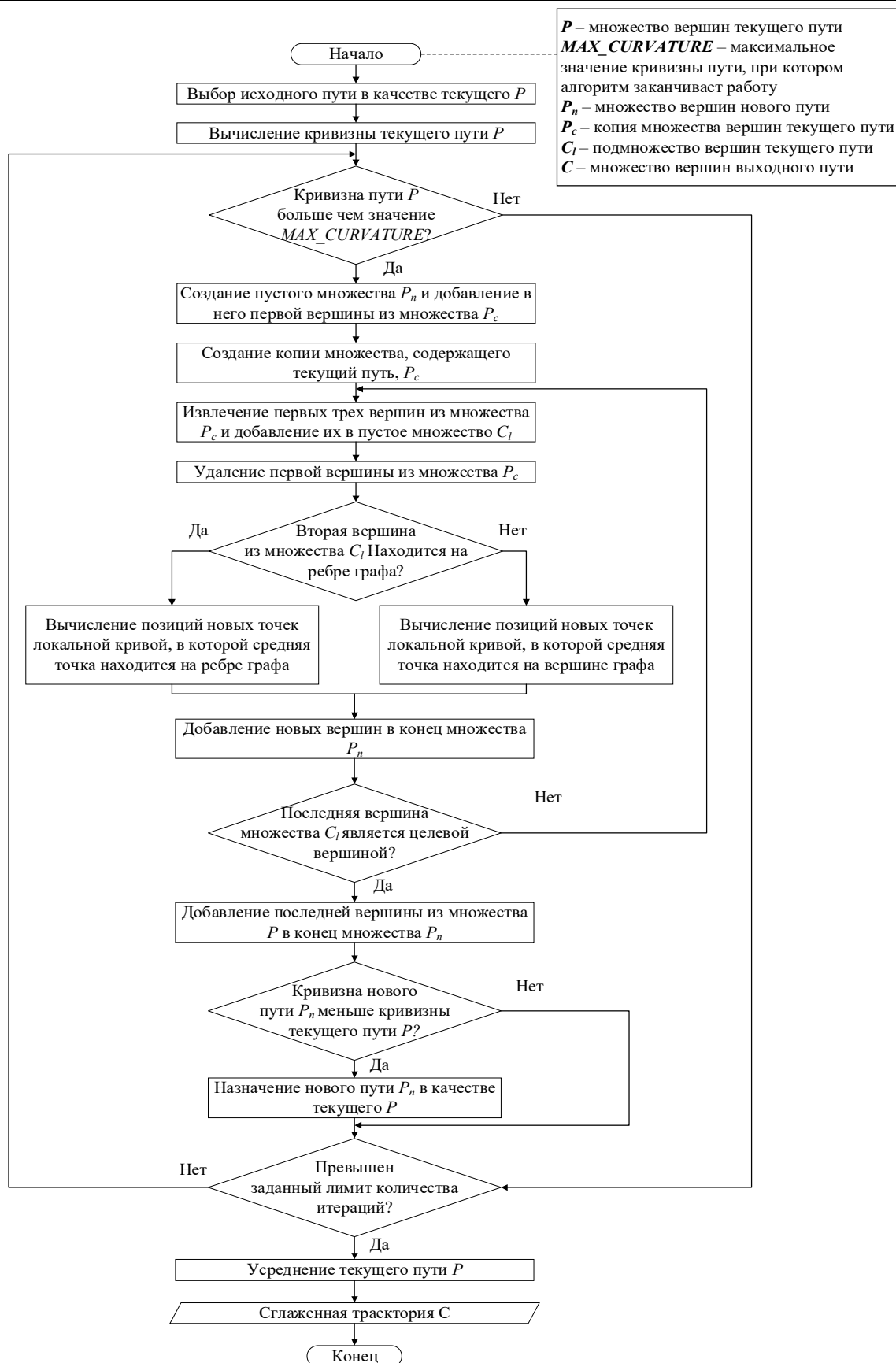


Рис. 1. Алгоритм сглаживания траектории движения CSA

Fig. 1. Algorithm for smoothing the trajectory of CSA

В первом варианте p_i может быть смещена в двух направлениях, а именно в сторону одной из точек, образующих ребро. Во втором варианте p_i заменяется на несколько точек, лежащих на ребрах, инцидентных вершине p_i . Одна итерация сглаживания кривой включает в себя смещение каждой точки данной кривой. Алгоритм сглаживания кривой совершает шаг сглаживания для каждой последовательной тройки точек исходной кривой и заканчивает работу, когда заданный критерий кривизны траектории будет достигнут или по истечению определенного количества итераций.

Вычисление позиции точки, находящейся на ребре

В случае, если точка кривой находится на ребре, то для нее вычисляется новая позиция, которая обеспечит наименьшую кривизну локальной кривой. В алгоритме на рис. 1 данная процедура выполняется в блоке «Вычисление позиций новых точек локальной кривой, в которой средняя точка находится на ребре графа». Пусть p_i – точка, лежащая на ребре e , $p_- = p_{i-1}$ и $p_+ = p_{i+1}$ – её соседние точки, а p_{e1} и p_{e2} – точки, образующие ребро e . Тогда координата x новой точки p^* находится путем объединения уравнений прямых (p_-, p_+) и (p_{e1}, p_{e2}) в двумерной плоскости в систему и её решения:

$$\begin{cases} y = \frac{k_a * x + k_{1a}}{k_{2a}} \\ y = \frac{k_b * x + k_{1b}}{k_{2b}}, \end{cases} \quad (1)$$

где параметры выражены следующими уравнениями:

$$\begin{aligned} k_a &= p_-(y) - p_+(y), \\ k_{1a} &= p_-(y) * p_+(x) - p_-(x) * p_+(y), \\ k_{2a} &= p_-(x) - p_+(x), \\ k_b &= p_{e1}(y) - p_{e2}(y), \\ k_{1b} &= p_{e1}(y) * p_{e2}(x) - p_{e1}(x) * p_{e2}(y), \\ k_{2b} &= p_{e1}(x) - p_{e2}(x). \end{aligned}$$

В (1) вычтем из первого уравнения второе и выразим x :

$$x = \frac{k_{2a} * k_{1b} - k_{2b} * k_{1a}}{k_{2b} * k_a - k_{2a} * k_b}. \quad (2)$$

Координаты y и z новой точки p^* находятся путем подстановки координаты x , найденной с помощью формулы (2), в уравнение прямой между точками p_{e1} и p_{e2} :

$$\begin{aligned} y &= \frac{k_{2y} * x + k_y}{k_1}, \\ z &= \frac{k_{2z} * x + k_z}{k_1}, \end{aligned} \quad (3)$$

где параметры выражены следующими уравнениями:

$$\begin{aligned} k_{2y} &= p_{e1}(y) - p_{e2}(y), \\ k_y &= p_{e2}(y) * p_{e1}(x) - p_{e2}(x) * p_{e1}(y), \\ k_z &= p_{e2}(z) * p_{e1}(x) - p_{e2}(x) * p_{e1}(z), \\ k_1 &= p_{e1}(x) - p_{e2}(x). \end{aligned}$$

Далее вычисляется расстояние между p^* и p_{init} – точкой изначальной кривой, к которой применяется шаг сглаживания. Если расстояние оказывается больше определенного порога λ_{dist} ,

тогда координаты p^* вычисляются по формуле

$$p^* = p_{init} + \lambda_{dist} * v(p_{init}, p^*), \quad (4)$$

где $v(p_{init}, p^*)$ – трехмерный вектор от p_{init} к p^* .

Б. Вычисление позиции точки, находящейся на вершине

Второй вариант шага сглаживания применяется в том случае, когда точка p_i локальной кривой $[p_-, p_i, p_+]$ находится на вершине графа, кривая должна быть локально разделена на несколько сегментов, покрывающих ребра $E_i := \{(i, j) \in E \mid j \in V\}$. В алгоритме на рис. 1 данная процедура выполняется в блоке «Вычисление позиций новых точек локальной кривой, в которой средняя точка находится на вершине графа». Мы разделяем смежные с p_i вершины на два упорядоченных множества. Первое множество N_1 перечисляет всех соседей вершины p_i против часовой стрелки, начиная от p_- и заканчивая p_+ . Второе множество N_2 перечисляет всех соседей вершины p_i по часовой стрелке, начиная от p_- и заканчивая p_+ . Пусть p_k^1 и p_k^2 обозначают позиции вершин в N_1 и N_2 соответственно, где конечные точки $p_0^1 = p_0^2 = p_-$ и $p_m^1 = p_n^2 = p_+$.

После сглаживания локальной кривой p_i будет заменена последовательностью точек, лежащих на ребрах: $E_1 = \{(i, j) \mid j \in N_1\}$ или $E_2 = \{(i, j) \mid j \in N_2\}$.

Для выбора между этими двумя множествами вершин мы проводим прямую между p_- и p_+ . Если прямая пересекает в двухмерной плоскости ребра из E_1 , то точки новой локальной кривой будут пролегать через ребра из этого множества. В противном случае, если прямая пересекает в двухмерной плоскости ребра из E_2 , то точки новой локальной кривой будут пролегать через ребра из множества E_2 .

После выбора множества ребер N_c , через которые будет пролегать новая локальная кривая, мы вычисляем позиции новых точек, начиная от p_0^c до p_+ . Для каждой новой точки p_k^c при $k = 1, \dots, m-1$ вычисляется её позиция, где m – мощность множества N_c . Координата x точки p_k^c находится как пересечение прямых (p_-, p_+) и (p_i, n_k^c) в двухмерной плоскости с помощью формул (1) и (2), а координаты y и z находятся с помощью подстановки найденной координаты x в уравнение прямой (p_i, n_k^c) в формуле, где n_k^c – k -ая смежная с p_i из множества N_c . Если расстояние между новой вершиной p_k^c и вершиной изначальной кривой $p_i = p_{init}$ больше λ_{dist} , то позиция вершины p_k^c вычисляется по формуле (4), где v – трехмерный вектор от p_{init} к n_p^c .

После нахождения позиций новых вершин локальных кривых они заносятся в конец множества. Если последняя

вершина множества C_i является целевой вершиной пути, то в конец множества P_n добавляется целевая вершина. После этого осуществляется оценка кривизны пути из множества P_n , и если значение кривизны пути из множества P_n меньше, чем значение кривизны пути из множества P , то новый путь P_n назначается в качестве текущего пути P . Если превышен заданный лимит количества итераций или кривизна текущего пути P меньше, чем значение $MAX_CURVATURE$, то алгоритм заканчивает основной этап сглаживания траектории движения. В противном случае начинается новая итерация сглаживания пути с блока «Создание пустого множества P_n и добавление в него первой вершины из множества P » с рис. 1.

В. Усреднение кривой

После завершения основного этапа сглаживания кривой необходимо снизить кривизну траектории на критических участках, чтобы сделать её более плавной в местах поворота. Главная проблема алгоритма сглаживания, взятого за основу [24], в том, что он учитывает кривизну локальной кривой, состоящей всего из трех точек, поэтому в некоторых местах траектория движения может быть ломаной. Для устранения данной проблемы был разработан метод усреднения кривой. На рис. 1 данная процедура идёт в конце блок-схемы, в блоке «Усреднение текущего пути P ».

Усреднение кривой заключается в смещении срединных точек локальных кривых, состоящих из 6 точек, вдоль ребер, на которых они находятся. Изменение позиции точки кривой p_i будет влиять на кривизну локальной кривой $[p_{i-2}, p_{i-1}, p_i, p_{i+1}, p_{i+2}]$, поэтому, чтобы избежать «изломов» глобальной кривой мы будем последовательно извлекать из глобальной кривой локальную кривую $c = [p_{i-5}, p_{i-4}, p_{i-3}, p_{i-2}, p_{i-1}, p_i]$, где $i = 6, \dots, n$, и будем изменять позиции точек p_3 и p_4 . Вычислим альтернативные позиции точек с помощью формулы (4):

$$\begin{aligned} p_{i-3}^1 &= p_{i-3} + shift_{dist} * v(p_{i-3}, p_{i-3}^{e1}), \\ p_{i-3}^2 &= p_{i-3} + shift_{dist} / 2 * v(p_{i-3}, p_{i-3}^{e1}), \\ p_{i-3}^3 &= p_{i-3} + shift_{dist} * v(p_{i-3}, p_{i-3}^{e2}), \\ p_{i-3}^4 &= p_{i-3} + shift_{dist} / 2 * v(p_{i-3}, p_{i-3}^{e2}), \\ p_{i-2}^1 &= p_{i-2} + shift_{dist} * v(p_{i-2}, p_{i-2}^{e1}), \\ p_{i-2}^2 &= p_{i-2} + shift_{dist} / 2 * v(p_{i-2}, p_{i-2}^{e1}), \\ p_{i-2}^3 &= p_{i-2} + shift_{dist} * v(p_{i-2}, p_{i-2}^{e2}), \\ p_{i-2}^4 &= p_{i-2} + shift_{dist} / 2 * v(p_{i-2}, p_{i-2}^{e2}), \end{aligned}$$

где $shift_{dist}$ – расстояние, соразмерное с длиной ребер, на которое будут смещены точки кривой, а p_{i-3}^{e1} , p_{i-3}^{e2} , p_{i-2}^{e1} , p_{i-2}^{e2} – точки, соединённые ребрами, на которых находятся точки p_{i-3} и p_{i-2} соответственно. Тогда $P_{i-3} = [p_{i-3}, p_{i-3}^1, p_{i-3}^2, p_{i-3}^3, p_{i-3}^4]$ и $P_{i-2} = [p_{i-2}, p_{i-2}^1, p_{i-2}^2, p_{i-2}^3, p_{i-2}^4]$ – множества альтернативных позиций точек p_{i-3} и p_{i-2} соответственно. После нахождения

ния альтернативных позиций точек p_{i-3} и p_{i-2} на место этих точек в локальную кривую c подставляется каждая возможная пара вершин из множеств P_{i-3} и P_{i-2} . Позиции точек p_{i-3} и p_{i-2} изменяются на те, которые обеспечивают наименьшее значение кривизны локальной кривой c . Благодаря попарному изменению позиций серединных точек локальной кривой метод усреднения кривой находит оптимальные пози-

ции точек, находящихся на изгибах

кривой, обеспечивая тем самым более плавное движение робота.

Результаты и их обсуждение

Для апробации разработанного решения был выбран симулятор Gazebo, в котором была смоделирована трехмерная карта и РТК, передвигающийся по ней. На рис. 2 представлен изначальный путь РТК, построенный алгоритмом LRLHD-A* [27] (белые точки), и полученная на выходе сглаженная кривая (зелеными точками).

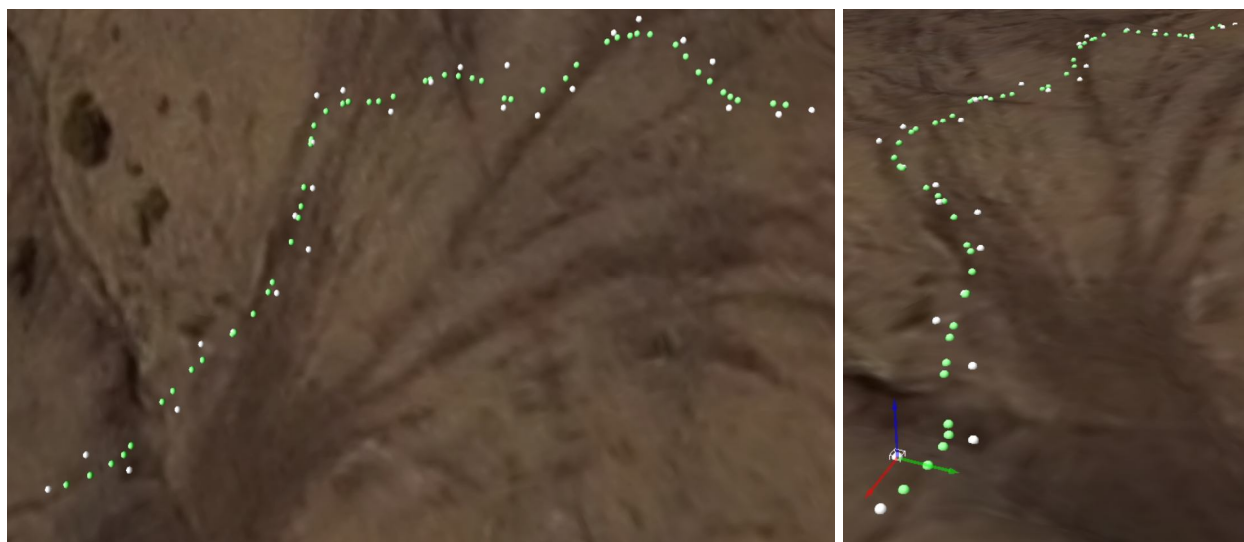


Рис. 2. Визуализация построенного пути и сглаженной кривой в Gazebo

Fig. 2. Visualization of the constructed path and smoothed curve in Gazebo

Как видно из рис. 2, сглаженная кривая не имеет сильных изгибов по сравнению с исходной траекторией, а также точки выходной кривой находятся недалеко от соответствующих им точек исходного пути.

Проблема сглаживания траектории движения наземного робота в трехмерном пространстве на данный момент недостаточно хорошо изучена [24, 25],

поэтому для оценки разработанного алгоритма сравним его с исходным алгоритмом и алгоритмом из работы [24]. Были проведены эксперименты по сравнению кривизны траекторий, получаемых с помощью алгоритма сглаживания кривой с усреднением кривой (CSA) и без него (CS). Для вычисления кривизны глобальной траектории в данной работе используется следующий

метод: каждые три последовательные точки глобальной траектории формируют локальную кривую $c = [p_i, p_{i+1}, p_{i+2}]$; кривизна каждой локальной кривой вычисляется как арккосинус скалярного произведения двухмерных направленных векторов $\overrightarrow{p_i p_{i+1}}$ и $\overrightarrow{p_{i+1} p_{i+2}}$, поделенного на произведение модулей этих векторов:

$$\alpha_i = \arccos \left(\frac{\overrightarrow{p_i p_{i+1}} * \overrightarrow{p_{i+1} p_{i+2}}}{|\overrightarrow{p_i p_{i+1}}| * |\overrightarrow{p_{i+1} p_{i+2}}|} \right).$$

Кривизна глобальной траектории C равна наибольшему значению угла между векторами одной из локальных кривых, входящих в нее:

$$C = \max_{i=1 \dots n-2} \alpha_i,$$

где n – количество точек глобальной траектории.

Тестирование проводилось на трехмерной карте местности, представлен-

ной 100082 вершинами и 198580 треугольниками. Для проведения экспериментов использовалась выборка из 10 пар случайных вершин, между которыми строился путь с помощью алгоритма LRLHD-A*. На рис. 3 представлена зависимость кривизны траектории от ее длины для исходного пути и сглаженных с помощью методов CS и CSA кривых.

Из рис. 3 видно, что сглаживание кривой без усреднения (CS) снижает кривизну изначальной кривой более чем в 2 раза. Также из рис. 3 видно, что применение операции усреднения к сглаженной траектории (CSA) снижает её кривизну на 24-42 %. Это обусловлено тем, что кривая, сглаженная по методу CS, имеет сильные изгибы, что снижает общую кривизну такой кривой. На рис. 4 представлен участок карты, на котором визуализированы траектории, сглаженные с помощью алгоритма CS (белыми точками) и алгоритма CSA (зелеными точками).

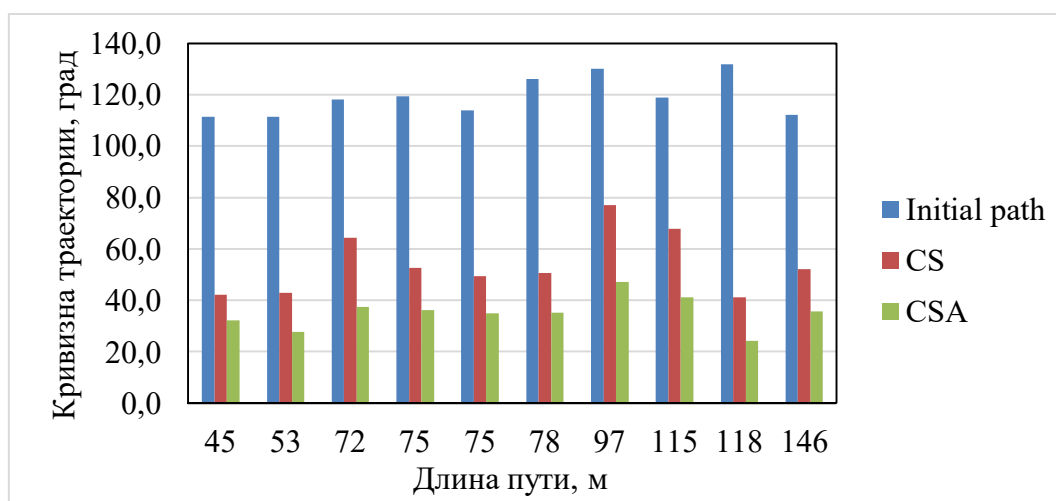


Рис. 3. Зависимость кривизны траектории от ее длины

Fig. 3. Dependence of the trajectory curvature on its length

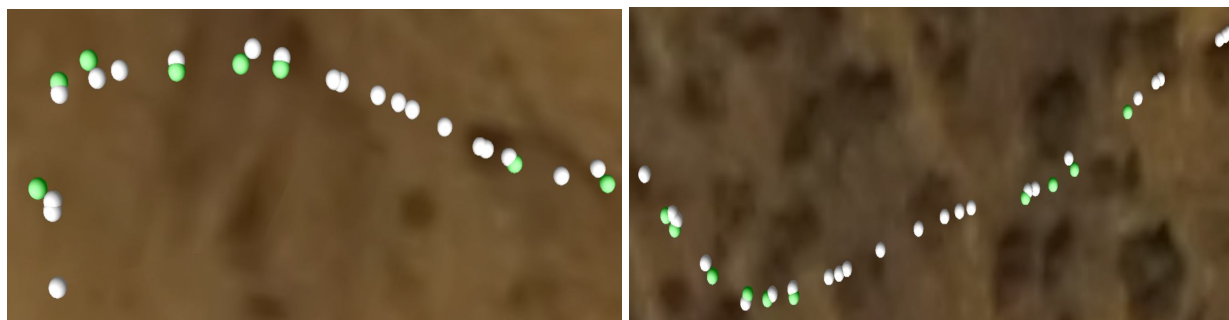


Рис. 4. Траектории, сглаженные с помощью алгоритмов CS и CSA

Fig. 4. Trajectories smoothed using the CS and CSA algorithms

Из рис. 4 видно, что траектория, сглаженная с помощью алгоритма CS, имеет резкие повороты, где вершины кривой образуют почти прямой угол, в то время как траектория, сглаженная с помощью алгоритма CSA, имеет более плавные изгибы на поворотах. Применение операции усреднения к кривой перемещает ее точки в проблемных участках таким образом, чтобы кривизна локальных кривых, находящихся в этих участках, была минимальной.

Выводы

Проблема сглаживания траектории движения наземного робота в трехмерном пространстве еще недостаточно изучена. Решение этой проблемы позволит более эффективно планировать пути движения наземных роботов, эксплуатируемых в исследовательских, сельскохозяйственных, картографических и других целях. В данной работе

был предложен алгоритм сглаживания траектории, представленной в виде пути на графе в трехмерном пространстве, с усреднением конечной кривой. Разработанный алгоритм CSA и алгоритм CS был апробированы в симуляторе Gazebo. В результате экспериментов установлено, что метод CSA строит кривые, кривизна которых на 24-42 % меньше, чем у кривых, находимых с помощью метода CS. Траектории, сглаженные с помощью алгоритма CSA, имеют более плавные изгибы кривой на поворотах, по сравнению с алгоритмом, взятым за основу. Благодаря этому повышается эффективность роботов, движущихся по данным траекториям. В дальнейшем разработанный алгоритм будет применяться в задачах планирования маршрутов на с/х территориях [28-31] и в модульной робототехнике [32, 33].

Список литературы

1. Dubins L.E. On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents // American Journal of mathematics. 1957. № 79(3). P. 497-516. <https://doi.org/10.2307/2372560>

2. Pérez J., Godoy J., Villagrà J., Onieva E. Trajectory generator for autonomous vehicles in urban environments // *2013 IEEE International Conference on Robotics and Automation*. IEEE, 2013. P. 409-414. <https://doi.org/10.1109/ICRA.2013.6630608>
3. Gerlach A.R., Kingston D., Walker B.K. UAV navigation using predictive vector field control // *2014 American Control Conference*. IEEE, 2014. P. 4907-4912. <https://doi.org/10.1109/ACC.2014.6859082>
4. Lin Y., Saripalli S. Path planning using 3D dubins curve for unmanned aerial vehicles // *2014 international conference on unmanned aircraft systems (ICUAS)*. IEEE, 2014. P. 296-304. <https://doi.org/10.1109/ICUAS.2014.6842268>
5. Choi J., Curry R., Elkaim G. Path planning based on bézier curve for autonomous ground vehicles // *Advances in Electrical and Electronics Engineering-IAENG Special Edition of the World Congress on Engineering and Computer Science 2008*. IEEE, 2008. P. 158-166. <https://doi.org/10.1109/WCECS.2008.27>
6. Rastelli J.P., Lattarulo R., Nashashibi F. Dynamic trajectory generation using continuous-curvature algorithms for door to door assistance vehicles // *2014 IEEE Intelligent Vehicles Symposium Proceedings*. IEEE, 2014. P. 510-515. <https://doi.org/10.1109/IVS.2014.6856526>
7. Walton D.J., Meek D.S., Ali J.M. Planar G2 transition curves composed of cubic Bézier spiral segments // *Journal of Computational and Applied Mathematics*. 2003. № 157(2). P. 453-476. [https://doi.org/10.1016/S0377-0427\(03\)00435-7](https://doi.org/10.1016/S0377-0427(03)00435-7)
8. Montes N., Mora M.C., Tornero J. Trajectory generation based on rational bezier curves as clothoids // *2007 IEEE Intelligent Vehicles Symposium*. IEEE, 2007. P. 505-510. <https://doi.org/10.1109/IVS.2007.4290165>
9. Real-time clothoid approximation by Rational Bezier curves / N. Montés, A. Herraéz, L. Armesto, J. Tornero // *2008 IEEE International Conference on Robotics and Automation*. IEEE, 2008. P. 2246-2251. <https://doi.org/10.1109/ROBOT.2008.4543548>
10. Bezier curve based path planning for autonomous vehicle in urban environment / L. Han, H. Yashiro, H.T.N. Nejad, Q.H. Do, S. Mita // *2010 IEEE Intelligent Vehicles Symposium*. IEEE, 2010. P. 1036-1042. <https://doi.org/10.1109/IVS.2010.5548085>
11. Continuous curvature planning with obstacle avoidance capabilities in urban scenarios / D. González, J. Perez, R. Lattarulo, V. Milanés, F. Nashashibi // *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2014. P. 1430-1435. <https://doi.org/10.1109/ITSC.2014.6957887>
12. Elbanhawi M., Simic M., Jazar R.N. Continuous path smoothing for carlike robots using B-spline curves // *Journal of Intelligent & Robotic Systems*. 2015. № 80(1). P. 23-56. <https://doi.org/10.1007/s10846-014-0172-0>
13. Elbanhawi M., Simic M., Jazar R. Randomized bidirectional B-Spline parameterization motion planning // *IEEE Transactions on intelligent transportation systems*. 2015. № 17(2). P. 406-419. <https://doi.org/10.1109/TITS.2015.2477355>

14. Komoriya K., Tanie K. Trajectory design and control of a wheel-type mobile robot using B-spline curve // *Proceedings. IEEE/RSJ International Workshop on Intelligent Robots and Systems'. (IROS'89)'The Autonomous Mobile Robots and Its Applications.* IEEE, 1989. P. 398-405. <https://doi.org/10.1109/IROS.1989.637937>
15. Berglund T., Brodnik A., Jonsson H., Staffanson M., Soderkvist I. Planning smooth and obstacle-avoiding B-spline paths for autonomous mining vehicles // *IEEE Transactions on Automation Science and Engineering.* 2009. № 7(1). P. 167-172. <https://doi.org/10.1109/TASE.2009.2015886>
16. Yang K., Sukkarieh S. An analytical continuous-curvature path-smoothing algorithm // *IEEE Transactions on Robotics.* 2010. № 26(3). P. 561-568. <https://doi.org/10.1109/TRO.2010.2042990>
17. Herrmann P., Gerngroß M., Endisch C. NURBS based trajectory generation for an industrial five axis needle winding robot // *2018 4th International Conference on Control, Automation and Robotics (ICCAR).* IEEE, 2018. P. 31-36. <https://doi.org/10.1109/ICCAR.2018.8384640>
18. Ravari A.N., Taghirad H.D. NURBS-based representation of urban environments for mobile robots // *2016 4th International Conference on Robotics and Mechatronics (ICROM).* IEEE, 2016. P. 20-25. <https://doi.org/10.1109/ICRoM.2016.7886782>
19. Shi X., Fang H., Guo L. Multi-objective optimal trajectory planning of manipulators based on quintic NURBS // *2016 IEEE International Conference on Mechatronics and Automation.* IEEE, 2016. P. 759-765. <https://doi.org/10.1109/ICMA.2016.7558658>
20. NURBs trajectory generation and following by an autonomous mobile robot navigating in 3D environment / H. Belaidi, A. Hentout, B. Bouzouia, H. Bentarzi, A. Belaidi // *The 4th Annual IEEE International Conference on Cyber Technology in Automation, Control and Intelligent.* IEEE, 2014. P. 168-173. <https://doi.org/10.1109/CYBER.2014.6917455>
21. Guo H., Meng Y., Jin Y. Swarm robot pattern formation using a morphogenetic multi-cellular based self-organizing algorithm // *2011 IEEE International Conference on Robotics and Automation.* IEEE, 2011. P. 3205-3210. <https://doi.org/10.1109/ICRA.2011.5979821>
22. Huh U.Y., Chang S.R. AG 2 continuous path-smoothing algorithm using modified quadratic polynomial interpolation // *International Journal of Advanced Robotic Systems.* – 2014. № 11(2). P. 25. <https://doi.org/10.5772/57340>
23. Chang S.R., Huh U.Y. A collision-free G 2 continuous path-smoothing algorithm using quadratic polynomial interpolation // *International Journal of Advanced Robotic Systems.* 2014. № 11(12). P. 194. <https://doi.org/10.5772/59463>
24. Lawonn K., Gasteiger R., Rössl C., Preim B. Adaptive and robust curve smoothing on surface meshes // *Computers & graphics.* 2014. № 40(22-35). <https://doi.org/10.1016/j.cag.2014.01.004>

25. Zhang H., Yang S. Smooth path and velocity planning under 3D path constraints for car-like vehicles // *Robotics and Autonomous Systems*. 2018. №107. P. 87-99. <https://doi.org/10.1016/j.robot.2018.05.013>
26. Hameiri E., Shimshoni I. Estimating the principal curvatures and the Darboux frame from real 3-D range data // *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*. 2003. № 33(4). P. 626-637. <https://doi.org/10.1109/TSMCB.2003.814304>
27. Zakharov K., Saveliev A., Sivchenko O. Energy-Efficient Path Planning Algorithm on Three-Dimensional Large-Scale Terrain Maps for Mobile Robots // *International Conference on Interactive Collaborative Robotics*. Springer, Cham, 2020. P. 319-330. <https://doi.org/10.1007/978-3-030-60337-3>
28. Aksamentov E., Astapova M., Usina E. Approach to Obstacle Localization for Robot Navigation in Agricultural Territories // *International Conference on Interactive Collaborative Robotics*. Springer, Cham, 2020. P. 13-20. https://doi.org/10.1007/978-3-030-60337-3_2
29. Robot Navigation System in Stochastic Environment Based on Reinforcement Learning on Lidar Data / D. Dudarenko, A. Kovalev, I. Tolstoy, I. Vatamaniuk // *Proceedings of 14th International Conference on Electromechanics and Robotics "Zavalishin's Readings"*. Springer, Singapore, 2020. P. 537-547. https://doi.org/10.1007/978-981-13-9267-2_44
30. Денисов А.В. Алгоритмы организации беспроводного информационного взаимодействия сенсорных систем и роботизированных устройств // *Известия Волгоградского государственного технического университета*. 2020. № 9(244). С. 30-34. <https://doi.org/10.35211/1990-5297-2020-9-244-30-34>
31. Ронжин А.Л., Тьен Н.К., Винь Н.В. Задачи управления обменом физических ресурсов между сельскохозяйственной техникой разной степени роботизации // *Известия ЮФУ. Технические науки*. 2020. № 1. С. 40-51. <https://doi.org/10.18522/2311-3103-2020-1-39-51>.
32. Ковалев А.Д. Подход к реконфигурации модульной робототехнической системы с использованием полиномиального алгоритма субоптимального поиска // *Известия Волгоградского государственного технического университета*. 2020. № 9(244). С. 48-51. <https://doi.org/10.35211/1990-5297-2020-9-244-48-51>
33. Pavliuk N., Saveliev A., Cherskikh E., Pykhov D. Formation of Modular Structures with Mobile Autonomous Reconfigurable System // *Proceedings of 14th International Conference on Electromechanics and Robotics "Zavalishin's Readings"*. Springer, Singapore, 2019. P. 383-395. https://doi.org/10.1007/978-981-13-9267-2_31

References

1. Dubins L.E. On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents. *American Journal of mathematics*, 1957, no. 79(3), pp. 497-516. <https://doi.org/10.2307/2372560>
2. Pérez J., Godoy J., Villagrà J., Onieva E. Trajectory generator for autonomous vehicles in urban environments. *2013 IEEE International Conference on Robotics and Automation*, IEEE, 2013, pp. 409-414. <https://doi.org/10.1109/ICRA.2013.6630608>
3. Gerlach A.R., Kingston D., Walker B.K. UAV navigation using predictive vector field control. *2014 American Control Conference*, IEEE, 2014, pp. 4907-4912. <https://doi.org/10.1109/ACC.2014.6859082>
4. Lin Y., Saripalli S. Path planning using 3D dubins curve for unmanned aerial vehicles. *2014 international conference on unmanned aircraft systems (ICUAS)*. IEEE, 2014, pp. 296-304. <https://doi.org/10.1109/ICUAS.2014.6842268>
5. Choi J., Curry R., Elkaim G. Path planning based on bézier curve for autonomous ground vehicles. *Advances in Electrical and Electronics Engineering-IAENG Special Edition of the World Congress on Engineering and Computer Science 2008*. IEEE, 2008, pp. 158-166. <https://doi.org/10.1109/WCECS.2008.27>
6. Rastelli J.P., Lattarulo R., Nashashibi F. Dynamic trajectory generation using continuous-curvature algorithms for door to door assistance vehicles. *2014 IEEE Intelligent Vehicles Symposium Proceedings*. IEEE, 2014, pp. 510-515. <https://doi.org/10.1109/IVS.2014.6856526>
7. Walton D.J., Meek D.S., Ali J.M. Planar G2 transition curves composed of cubic Bézier spiral segments. *Journal of Computational and Applied Mathematics*, 2003, no. 157(2), pp. 453-476. [https://doi.org/10.1016/S0377-0427\(03\)00435-7](https://doi.org/10.1016/S0377-0427(03)00435-7)
8. Montes N., Mora M.C., Tornero J. Trajectory generation based on rational bezier curves as clothoids. *2007 IEEE Intelligent Vehicles Symposium*. IEEE, 2007, pp. 505-510. <https://doi.org/10.1109/IVS.2007.4290165>
9. Montés N., Herraiz A., Armesto L., Tornero J. Real-time clothoid approximation by Rational Bezier curves. *2008 IEEE International Conference on Robotics and Automation*. IEEE, 2008, pp. 2246-2251. <https://doi.org/10.1109/ROBOT.2008.4543548>
10. Han L., Yashiro H., Nejad, H.T.N., Do Q.H., Mita S. Bezier curve based path planning for autonomous vehicle in urban environment. *2010 IEEE Intelligent Vehicles Symposium*. IEEE, 2010, pp. 1036-1042. <https://doi.org/10.1109/IVS.2010.5548085>
11. González D., Perez J., Lattarulo R., Milanés V., Nashashibi F. Continuous curvature planning with obstacle avoidance capabilities in urban scenarios. *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2014, pp. 1430-1435. <https://doi.org/10.1109/ITSC.2014.6957887>

12. Elbanhawi M., Simic M., Jazar R.N. Continuous path smoothing for car-like robots using B-spline curves. *Journal of Intelligent & Robotic Systems*, 2015, no. 80(1), pp. 23-56. <https://doi.org/10.1007/s10846-014-0172-0>
13. Elbanhawi M., Simic M., Jazar R. Randomized bidirectional B-Spline parameterization motion planning. *IEEE Transactions on intelligent transportation systems*, 2015, no. 17(2), pp. 406-419. <https://doi.org/10.1109/TITS.2015.2477355>
14. Komoriya K., Tanie K. Trajectory design and control of a wheel-type mobile robot using B-spline curve. *Proceedings. IEEE/RSJ International Workshop on Intelligent Robots and Systems'. (IROS'89)'The Autonomous Mobile Robots and Its Applications* IEEE, 1989, pp. 398-405. <https://doi.org/10.1109/IROS.1989.637937>
15. Berglund T., Brodnik A., Jonsson H., Staffanson M., Soderkvist I. Planning smooth and obstacle-avoiding B-spline paths for autonomous mining vehicles. *IEEE Transactions on Automation Science and Engineering*, 2009, no. 7(1), pp. 167-172. <https://doi.org/10.1109/TASE.2009.2015886>
16. Yang K., Sukkarieh S. An analytical continuous-curvature path-smoothing algorithm. *IEEE Transactions on Robotics*, 2010, no. 26(3), pp. 561-568. <https://doi.org/10.1109/TRO.2010.2042990>
17. Herrmann P., Gerngroß M., Endisch C. NURBS based trajectory generation for an industrial five axis needle winding robot. *2018 4th International Conference on Control, Automation and Robotics (ICCAR)*. IEEE, 2018, pp. 31-36. <https://doi.org/10.1109/ICCAR.2018.8384640>
18. Ravari A.N., Taghirad H.D. NURBS-based representation of urban environments for mobile robots. *2016 4th International Conference on Robotics and Mechatronics (ICROM)*. IEEE, 2016, pp. 20-25. <https://doi.org/10.1109/ICRoM.2016.7886782>
19. Shi X., Fang H., Guo L. Multi-objective optimal trajectory planning of manipulators based on quintic NURBS. *2016 IEEE International Conference on Mechatronics and Automation*. IEEE, 2016, pp. 759-765. <https://doi.org/10.1109/ICMA.2016.7558658>
20. Belaidi H., Hentout A., Bouzouia B., Bentarzi H., Belaidi A. NURBs trajectory generation and following by an autonomous mobile robot navigating in 3D environment. *The 4th Annual IEEE International Conference on Cyber Technology in Automation, Control and Intelligent*. IEEE, 2014, pp. 168-173. <https://doi.org/10.1109/CYBER.2014.6917455>
21. Guo H., Meng Y., Jin Y. Swarm robot pattern formation using a morphogenetic multi-cellular based self-organizing algorithm. *2011 IEEE International Conference on Robotics and Automation*. IEEE, 2011, pp. 3205-3210. <https://doi.org/10.1109/ICRA.2011.5979821>
22. Huh U.Y., Chang S.R. AG 2 continuous path-smoothing algorithm using modified quadratic polynomial interpolation. *International Journal of Advanced Robotic Systems*, 2014, no. 11(2), 25 p. <https://doi.org/10.5772/57340>

23. Chang S.R., Huh U.Y. A collision-free G² continuous path-smoothing algorithm using quadratic polynomial interpolation. *International Journal of Advanced Robotic Systems*, 2014, no. 11(12), 194 p. <https://doi.org/10.5772/59463>
24. Lawonn K., Gasteiger R., Rössl C., Preim B. Adaptive and robust curve smoothing on surface meshes. *Computers & graphics*, 2014, no. 40(22-35). <https://doi.org/10.1016/j.cag.2014.01.004>
25. Zhang H., Yang S. Smooth path and velocity planning under 3D path constraints for car-like vehicles. *Robotics and Autonomous Systems*, 2018, no. 107, pp. 87-99. <https://doi.org/10.1016/j.robot.2018.05.013>
26. Hameiri E., Shimshoni I. Estimating the principal curvatures and the Darboux frame from real 3-D range data. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 2003, no.33(4), pp. 626-637. <https://doi.org/10.1109/TSMCB.2003.814304>
27. Zakharov K., Saveliev A., Sivchenko O. Energy-Efficient Path Planning Algorithm on Three-Dimensional Large-Scale Terrain Maps for Mobile Robots. *International Conference on Interactive Collaborative Robotics*. Springer, Cham, 2020, pp. 319-330. https://doi.org/10.1007/978-3-030-60337-3_31
28. Aksamentov E., Astapova M., Usina E. Approach to Obstacle Localization for Robot Navigation in Agricultural Territories. *International Conference on Interactive Collaborative Robotics*. Springer, Cham, 2020, pp. 13-20. https://doi.org/10.1007/978-3-030-60337-3_2
29. Dudarenko D., Kovalev A., Tolstoy I., Vatamaniuk I. Robot Navigation System in Stochastic Environment Based on Reinforcement Learning on Lidar Data. *Proceedings of 14th International Conference on Electromechanics and Robotics "Zavalishin's Readings"*. Springer, Singapore, 2020, pp. 537-547. https://doi.org/10.1007/978-981-13-9267-2_
30. Denisov A.V. Algoritmy organizatsii besprovodnogo informatsionnogo vzaimodeistviya sensorykh sistem i robotizirovannykh ustroystv [Development of a recommender system for parameter calculation in wireless network of sensor devices]. *Izvestiya Volgogradskogo gosudarstvennogo tekhnicheskogo universiteta = Bulletin of the Volgograd State Technical University*, 2019, no. 7(4), pp. 30-34 (In Russ.). <https://doi.org/10.26102/2310-6018/2019.27.4.025>
31. Ronzhin A.L., Ngo K.T., Nguyen V.V. Zadachi upravleniya obmenom fizicheskikh resursov mezhdu sel'skokhozyaistvennoi tekhnikai raznoi stepeni robotizatsii [Tasks of controlling the exchange of physical resources between agricultural means with varying degrees of robotization]. *Izvestiya YuFU. Tekhnicheskie nauki = Izvestiya SFedU. Engineering Sciences*, 2020, no. 1, pp. 40-51 (In Russ.). <https://doi.org/10.18522/2311-3103-2020-1-39-51>
32. Kovalev A.D. Podkhod k rekonfiguratsii modul'noi robototekhnicheskoi sistemy s ispol'zovaniem polinomial'nogo algoritma suboptimal'nogo poiska [Approach to reconfiguration of a modular robot system with a suboptimal search polynomial algorithm]. *Izvestiya Volgogradskogo gosudarstvennogo tekhnicheskogo universiteta = Bulletin of the Volgograd*

State Technical University. 2020, no. 9(244), pp. 48-51 (In Russ.). <https://doi.org/10.35211/1990-5297-2020-9-244-48-51>

33. Pavliuk N., Saveliev A., Cherskikh E., Pykhov D. Formation of Modular Structures with Mobile Autonomous Reconfigurable System. *Proceedings of 14th International Conference on Electromechanics and Robotics "Zavalishin's Readings"*. Springer, Singapore. 2019, pp. 383-395. https://doi.org/10.1007/978-981-13-9267-2_31

Информация об авторах / Information about the Authors

Захаров Константин Станиславович, младший научный сотрудник лаборатории автономных робототехнических систем, Санкт-Петербургский институт информатики и автоматизации Российской академии наук (СПИИРАН), г. Санкт-Петербург, Российская Федерация, e-mail: kon7666007@yandex.ru

Konstantin S. Zakharov, Junior Researcher of Autonomous Robotic Systems Laboratory, St.Petersburg Institute for Informatics and Automation of the Russian Academy of Sciences (SPIIRAS), St. Petersburg, Russian Federation, e-mail: kon7666007@yandex.ru

Савельев Антон Игоревич, кандидат технических наук, старший научный сотрудник, заведующий лабораторией автономных робототехнических систем, Санкт-Петербургский институт информатики и автоматизации Российской академии наук (СПИИРАН), г. Санкт-Петербург, Российская Федерация, e-mail: saveliev.ais@yandex.ru

Anton I. Saveliev, Cand. of Sci. (Engineering, Senior Researcher, Head of the Autonomous Robotic Systems Laboratory, St.Petersburg Institute for Informatics and Automation of the Russian Academy of Sciences (SPIIRAS), St. Petersburg, Russian Federation, e-mail: saveliev.ais@yandex.ru