

ВЕРОЯТНОСТНЫЙ МЕТОД ПЛАНИРОВАНИЯ ТРАЕКТОРИИ ДЛЯ АВТОНОМНОЙ ПЛАТФОРМЫ

С.М. Караф

karaf.sau@gmail.com
SPIN-код: 5531-9309

МГТУ им. Н.Э. Баумана, Москва, Российская Федерация

Аннотация

Разработан стенд для тестирования и отладки алгоритмов распознавания и поиска пути для автономных транспортных платформ. Местоположение автономной платформы и точек, обозначающих траекторию (контрольных точек), определены с помощью алгоритма распознавания цветов. Рассмотрен вероятностный метод для построения пути автономной платформы через заданную траекторию. Для нахождения угла поворота передних колес транспортной платформы после определения траектории был использован метод Стэнли. Команды управления записывались в файл формата JavaScript Object Notation и передавались на одноплатный компьютер Raspberry Pi с помощью Node.js сервера.

Ключевые слова

Планирование пути, беспилотный автомобиль, вероятностный метод, теорема Байеса, распознавание цветов, устройство автономной платформы, Стэнли метод

Поступила в редакцию 26.07.2021
© МГТУ им. Н.Э. Баумана, 2021

Введение. В настоящее время робототехнические решения находят все более широкое применение во всех сферах деятельности человека. Разрабатываются агроботы для использования в сельском хозяйстве, роботы-информаторы для интерактивного взаимодействия с людьми, роботы-курьеры для доставки посылок. Перспективным направлением в робототехнике является разработка беспилотных автомобилей, которые помогут решить широкий спектр задач.

При разработке беспилотного автомобиля особое внимание уделяется быстрому и надежному планировщику траектории движения. Существуют множество алгоритмов нахождения пути, различающиеся как скоростью работы, так и количеством требуемой памяти.

Выбор алгоритма планирования зависит от набора данных, приходящих с датчиков. Поскольку нейросетевые технологии широко используются при разработке беспилотных автомобилей, целесообразно использовать алгоритмы, основанные на методах математической статистики и теории вероятностей.

Рассмотрим структуру разрабатываемой системы (рис. 1). На вход системы подается изображение с видеокамеры. Необходимо, чтобы в ее поле зрения входила автономная платформа и искомые точки траектории. После получения изображения запускается алгоритм распознавания цветов. Выходными данными

алгоритма служит массив точек, содержащий их координаты (x, y) и цвет. Далее этот массив подается блоку планировщика пути, который строит триангуляцию Делоне [1]. Далее с помощью триангуляции строится дерево всевозможных путей, корнем которого является координата автономной платформы. Рассчитав апостериорные вероятности для каждого пути, можно найти наиболее вероятный путь. С помощью найденной траектории и положения машины генерируются команды управления методом Стенли [2, 3]. После выполнения всех вышеперечисленных шагов команды могут быть отправлены на автономную платформу с помощью сервера Node-JS в формате JavaScript Object Notation (json).

Команда управления имеет два поля: угол поворота передних колес и разрешение на движение. Для управления автономной платформой используются фреймворк ROS2 [4] и библиотека pigpio [5].

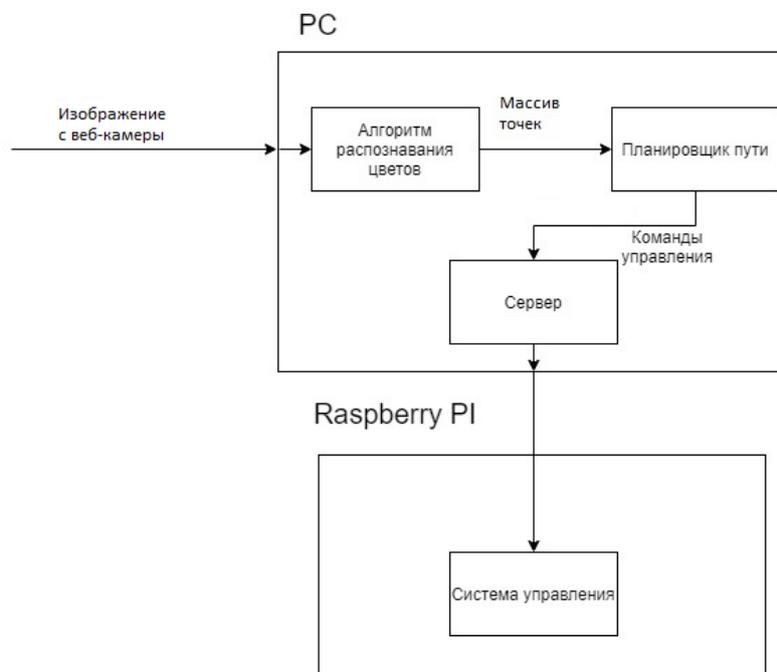


Рис. 1. Структура устройства автономной платформы

Алгоритм распознавание цветов. Если представить искомый цвет и цвет пикселя в виде точек в декартовой системе координат, то для нахождения пикселей нужного цвета можно использовать геометрическое расстояние между концами векторов

$$D = |\vec{C} - \vec{C}_n| = \sqrt{(R - R_n)^2 + (G - G_n)^2 + (B - B_n)^2}, \quad (1)$$

где вектор $\vec{C} = (R, G, B)$ характеризует искомый цвет, вектор $\vec{C}_n = (R_n, G_n, B_n)$ — цвет пикселя. Данное значение сравнивается с пороговым. Если расстояние между искомым цветом и цветом пикселя меньше чем пороговое расстояние, то либо со-

здается новый кластер и данный пиксель добавляется в нее, либо пиксель добавляется к ближайшему кластеру. Размеры кластеров также ограничены максимальной занимаемой площадью. Результаты работы алгоритма показаны на рис. 2 и 3.

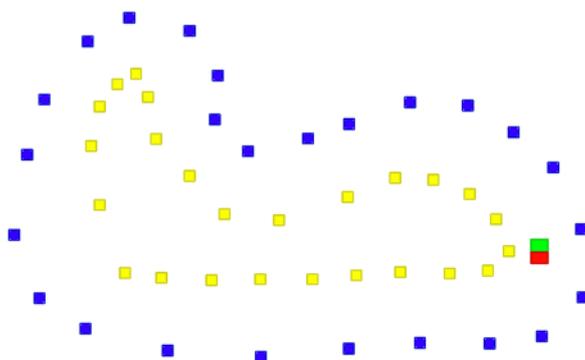


Рис. 2. Распознавание синего, красного, зеленого и желтого цветов в идеальных условиях

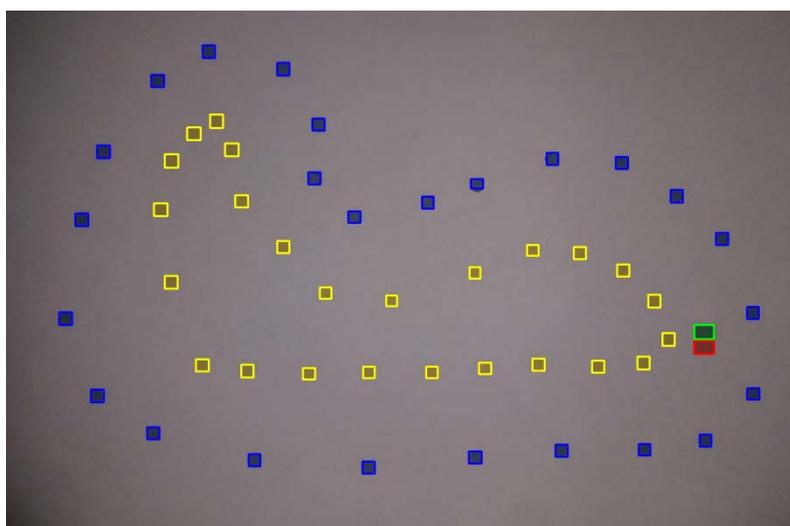


Рис. 3. Распознавание синего, красного, зеленого и желтого цветов в условиях плохой освещенности

Основным преимуществом такого подхода является простота реализации и отладки, но данный алгоритм очень чувствителен к изменениям освещения и требует калибровки для более точного распознавания цвета.

Алгоритм определения траектории. Рассматриваемый алгоритм [6, 7] предусматривает три шага:

- 1) дискретизацию пространства поиска при помощи триангуляции Делоне;
- 2) создание дерева всевозможных путей с помощью алгоритма триангуляции;
- 3) выбор наиболее вероятного пути, используя теорему Байеса.

Дискретизация пространства поиска. Дискретизация пространства поиска осуществляется с помощью библиотеки Fade_2d [8]. Алгоритм на входе прини-

мает контрольные точки трассы, а на выходе возвращает массив треугольников триангуляции (рис. 4).

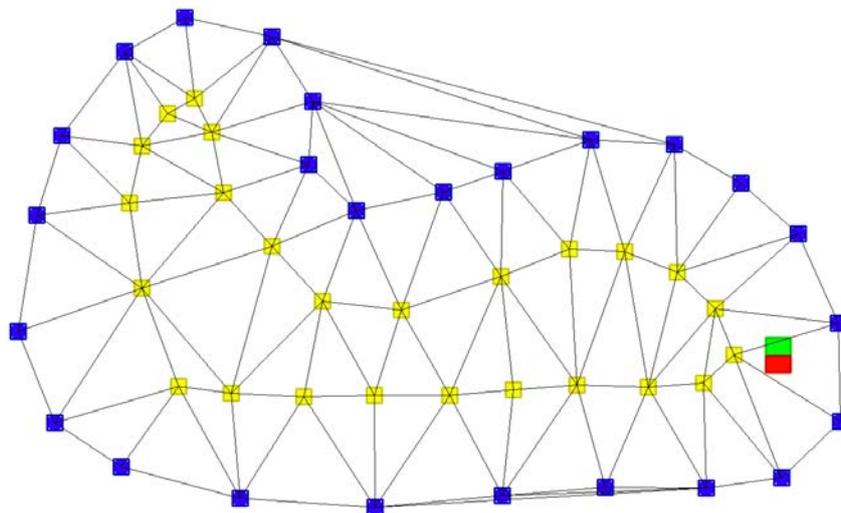


Рис. 4. Выполнение дискретизации пространства поиска в идеальных условиях

Построение дерева всевозможных путей. При помощи метода поиска в ширину происходит создание дерева всевозможных путей. Поиск в ширину работает путем последовательного просмотра отдельных уровней графа, начиная с узла-источника u — местоположение автономной платформы.

Рассмотрим все ребра (u, v) , выходящие из узла u , где узлы v — середины сторон треугольника. Каждый узел v добавляется в очередь. После того как будут проверены все ребра, выходящие из узла u , из очереди извлекается следующий узел и процесс повторяется до тех пор, пока очередь не окажется пустой. Дерево установлено таким образом, чтобы каждый узел имел ссылку на свои дочерние узлы.

Нахождение наиболее вероятной траектории. Для нахождения наиболее вероятной траектории применяют априорную, условную и апостериорную вероятности. В данной работе маргинальной вероятностью можно принять равной единице, поскольку она не влияет на окончательный результат.

Априорная вероятность — вероятность того, что данная конфигурация контрольных точек представляет правильный путь. Правильность пути определяется характеристиками F_j . Формула для нахождения априорной вероятности приведена ниже:

$$P_l = e^{-W_{prior}C}; \quad (2)$$

$$C = \sum_{i=1}^6 a_i (F_i - S_i)^2, \quad (3)$$

где W_{prior} — весовой коэффициент; C — функция стоимости; a_j — масштабирующий коэффициент; S_j — целевые значения. К характеристикам F_j относятся:

наибольшее изменение угла (F_1); среднее квадратичное отклонение (СКО) расстояния между левыми и правыми точками (F_2 и F_3 соответственно); СКО ширины трассы (F_4); количество узлов, из которых состоит путь (F_5); длина траектории (F_6).

Условная вероятность — произведение вероятности правильного распознавания цветов. Информацию о вероятностях правильного распознавания цвета обычно можно получить с помощью систем машинного зрения, основанных на нейронных сетях. Условную вероятность можно найти по формуле

$$P_{pr} = \prod_{i=1}^n P(C_i = c), \quad (4)$$

где $P(C_i = c)$ — вероятность, что точка C_i имеет цвет c .

Отметим, что в данной работе условная вероятность была принята равной единице, поскольку используемый алгоритм поиска цветов не возвращает оценку правильности распознавания. Для наиболее эффективной работы алгоритма стоит использовать метод, основанный на нейронных сетях.

Апостериорную вероятность можно найти как произведение условной и априорной вероятности:

$$P_{post} = P_l P_{pr} = e^{-W_{prior} C} \prod_{i=1}^n P(C_i = c). \quad (5)$$

Вероятность рассчитывают для каждой найденной траектории и выбирают путь с самой большой вероятностью (рис. 5).

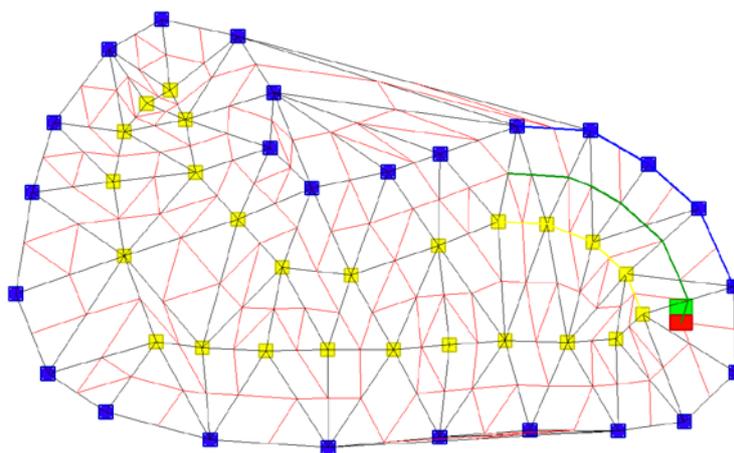


Рис. 5. Поиск пути в идеальных условиях

Контроллер Стенли. Контроллер стремится уменьшить расстояние от траектории до передней оси платформы e и погрешность в направлении ψ , изменяя угол поворота передних колес δ (рис. 6).

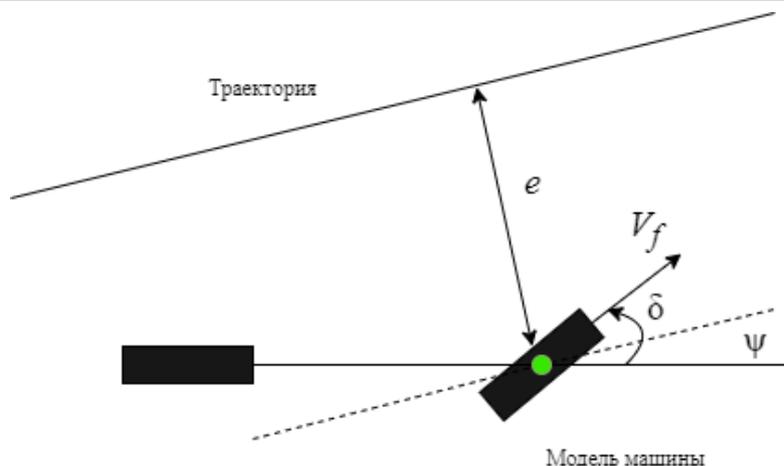


Рис. 6. Схематическое изображение транспортной платформы

Для ликвидации погрешности направления необходимо выполнение условия

$$\delta(t) = \psi(t). \quad (6)$$

Для уменьшения расстояния e необходимо повернуть колеса на угол

$$\delta(t) = \arctg\left(\frac{ke(t)}{k_s + V_f(t)}\right), \quad (7)$$

где k — коэффициент усиления контроллера; $V_f(t)$ — скорость машины (в данной работе принята постоянной); k_s — сглаживающая постоянная (в данной работе принята постоянной).

Таким образом,

$$\delta(t) = \psi(t) + \arctg\left(\frac{ke(t)}{k_s + v_f(t)}\right). \quad (8)$$

Базовые понятия ROS. Приведем некоторые базовые понятия, которые понадобятся далее [9].

«Рабочей» единицей программного обеспечения, написанной на ROS2, является нода. Можно провести аналогию с одной исполняемой программой. Рекомендуется использовать одну ноду для решения определенной задачи.

Для передачи данных между нодами применяют специальный формат. Эти данные называют сообщением, а их описание — типом сообщения.

Одним из видов обмена сообщениями в ROS2 является топик. Нода издателя (publisher) начинает публикацию сообщений в топик, а ноды подписчиков (subscriber) получают данные из этого топика.

Реализация системы. Рассмотрим работу автономной платформы. Для приема команд управления и изменения угла поворота машины в ней применяется

одноплатный компьютер Raspberry Pi. Для этого используется фреймворк ROS2. Архитектура программы на Raspberry Pi имеет вид, показанный на рис. 7.

Нода для приема json файла написана на языке Python и является нодой-издателем. Она с частотой 10 Гц запрашивает у сервера на персональном компьютере команды управления. При получении команд она отправляет сообщение в топик. Данное сообщение имеет два поля: угол поворота и разрешение на езду. После приема сообщения нода управления должна выставить необходимый угол поворота и начать езду по найденной траектории.

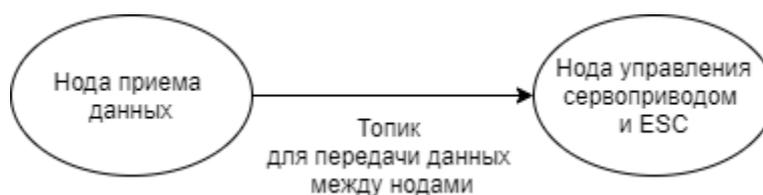


Рис. 7. Архитектура программы на Raspberry Pi

Для проверки общей работоспособности системы были проведены испытания на двух трассах (рис. 8). Трасса, показанная на рис. 8, а, представляет собой полуокружность с внутренним радиусом примерно 0,25 м и внешним радиусом примерно 1 м; на рис. 8, б — кривая траектория длиной примерно 3 м и шириной 1 м. В обоих случаях был найден путь (фиолетовая линия) и границы траектории (желтая и синяя линии), что говорит об общей работоспособности системы.

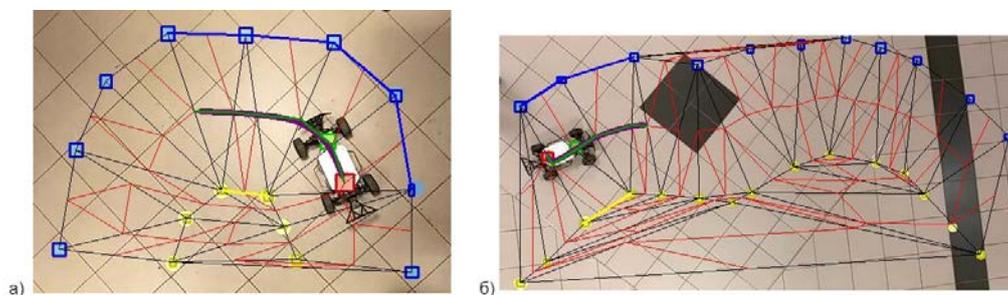


Рис. 8. Результаты выполнения программы.

Заключение. В ходе проделанной работы были получены следующие результаты:

- реализован алгоритм для поиска цветов, позволяющий находить точки заданной траектории при помощи геометрического расстояния между искомым цветом и цветом пикселя;
- разработана программа, реализующая вероятностный метод поиска пути автономной транспортной платформы;
- для управления углом поворота передними колёсами автономной транспортной платформы выбран метод Стэнли;
- разработан и собран стенд, с помощью которого проведена проверка правильности работы алгоритма поиска пути.

Литература

- [1] Rebay S. Efficient unstructured mesh generation by means of Delaunay triangulation and Bowyer-Watson algorithm. *J. Comput. Phys.*, 1993, vol. 106, no. 1, pp. 125–138. DOI: <https://doi.org/10.1006/jcph.1993.1097>
- [2] Thrun S., Montemerlo M., Dahlkamp H. et al. Stanley: the robot that won the DARPA grand challenge. *J. Field Robot.*, 2006, vol. 23, no. 9, pp. 661–692. DOI: <https://doi.org/10.1002/rob.20147>
- [3] Hoffmann G.M., Tomlin C.J. Autonomous automobile trajectory tracking for off-road driving: controller design, experimental validation and racing. *American Control Conf.*, 2007, pp. 2296–2301. DOI: <https://doi.org/10.1109/ACC.2007.4282788>
- [4] ROS 2 documentation. *docs.ros.org: веб-сайт*. URL: <https://docs.ros.org/en/foxy/> (дата обращения: 15.05.2021).
- [5] Pignio C interface. *abyz.me.uk: веб-сайт*. URL: <http://abyz.me.uk/rpi/pigpio/cif.html> (дата обращения: 15.05.2021).
- [6] Kabzan J., de la Iglesia Valls M., Reijgwart V. et al. AMZ driverless: the full autonomous racing system. URL: <http://arxiv.org/abs/1905.05150> (дата обращения: 15.05.2021).
- [7] Andresen L., Brandemuehl A., Honger A. et al. Mapping and planning for autonomous racing. *IEEE/RSJ IROS*, 2021, pp. 4743–4749. DOI: <https://doi.org/10.1109/IROS45743.2020.9341702>
- [8] Fade2D documentation. *geom.at: веб-сайт*. URL: <https://www.geom.at/fade2d/html/index.html> (дата обращения: 15.05.2021).
- [9] Слабуха Н. Введение в Robot Operating System. URL: <http://docs.voltbro.ru/starting-ros/> (дата обращения: 15.05.2021).

Караф Саусар Майк — студент кафедры «Управление в технических системах», МГТУ им. Н.Э. Баумана, Москва, Российская Федерация.

Научный руководитель — Глазков Виталий Владимирович, кандидат технических наук, доцент кафедры «Управление в технических системах», МГТУ им. Н.Э. Баумана, Москва, Российская Федерация.

Ссылку на эту статью просим оформлять следующим образом:

Караф С.М. Вероятностный метод планирования траектории для автономной платформы. *Политехнический молодежный журнал*, 2021, № 09(62). <http://dx.doi.org/10.18698/2541-8009-2021-09-738>

PROBABILISTIC TRAJECTORY PLANNING METHOD FOR AN AUTONOMOUS PLATFORM

S.M. Karaf

karaf.sau@gmail.com

SPIN-code: 5531-9309

Bauman Moscow State Technical University, Moscow, Russian Federation

Abstract

A stand has been developed for testing and debugging recognition and pathfinding algorithms for autonomous transport platforms. The location of the autonomous platform and the points indicating the trajectory (control points) are determined using a color recognition algorithm. A probabilistic method for constructing the path of an autonomous platform through a given trajectory is considered. The Stanley method was used to find the angle of rotation of the front wheels of the transport platform after determining the trajectory. Control commands were written to a JavaScript Object Notation file and transmitted to a Raspberry Pi single-board computer using a Node-js server.

Keywords

Path planning, self-driving car, probabilistic method, Bayes theorem, color recognition, autonomous platform device, Stanley method

Received 26.07.2021

© Bauman Moscow State Technical University, 2021

References

- [1] Rebay S. Efficient unstructured mesh generation by means of Delaunay triangulation and Bowyer-Watson algorithm. *J. Comput. Phys.*, 1993, vol. 106, no. 1, pp. 125–138. DOI: <https://doi.org/10.1006/jcph.1993.1097>
- [2] Thrun S., Montemerlo M., Dahlkamp H. et al. Stanley: the robot that won the DARPA grand challenge. *J. Field Robot.*, 2006, vol. 23, no. 9, pp. 661–692. DOI: <https://doi.org/10.1002/rob.20147>
- [3] Hoffmann G.M., Tomlin C.J. Autonomous automobile trajectory tracking for off-road driving: controller design, experimental validation and racing. *American Control Conf.*, 2007, pp. 2296–2301. DOI: <https://doi.org/10.1109/ACC.2007.4282788>
- [4] ROS 2 documentation. *docs.ros.org: website*. URL: <https://docs.ros.org/en/foxy/> (accessed: 15.05.2021).
- [5] Pigpio C interface. *abyz.me.uk: website*. URL: <http://abyz.me.uk/rpi/pigpio/cif.html> (accessed: 15.05.2021).
- [6] Kabzan J., de la Iglesia Valls M., Reijgwart V. et al. AMZ driverless: the full autonomous racing system. URL: <http://arxiv.org/abs/1905.05150> (accessed: 15.05.2021).
- [7] Andresen L., Brandemuehl A., Honger A. et al. Mapping and planning for autonomous racing. *IEEE/RSJ IROS*, 2021, pp. 4743–4749. DOI: <https://doi.org/10.1109/IROS45743.2020.9341702>
- [8] Fade2D documentation. *geom.at: website*. URL: <https://www.geom.at/fade2d/html/index.html> (accessed: 15.05.2021).
- [9] Slabukha N. Vvedenie v Robot Operating System [Introduction to the Robot Operating System] (in Russ.). URL: <http://docs.voltbro.ru/starting-ros/> (accessed: 15.05.2021).

Karaf S.M. — Student, Department of Automatic Control Systems, Bauman Moscow State Technical University, Moscow, Russian Federation.

Scientific advisor — Glazkov V.V., Cand. Sc. (Eng.), Assoc. Professor, Department of Automatic Control Systems, Bauman Moscow State Technical University, Moscow, Russian Federation.

Please cite this article in English as:

Karaf S.M. Probabilistic trajectory planning method for an autonomous platform. *Politekhnicheskiy molodezhnyy zhurnal* [Politechnical student journal], 2021, no. 09(62). <http://dx.doi.org/10.18698/2541-8009-2021-09-738.html> (in Russ.).