

УДК 004.07

## Разработка системы автономного движения и картографии для робота ROIN-RTS-100

Н.С. Пугачев✉

Воронежский государственный университет, Воронеж, Россия

*Задача управления роботом на пересеченной местности в условиях большой дистанции между оператором и роботом, как правило, осложняется помехами управляющего сигнала и проблемами связи. В целом не исключены ситуации полной потери связи. В таком случае поведение робота может быть непредсказуемо и привести к аварии и поломке. Для исключения подобных сценариев желательно иметь функционал автономного принятия решений роботом на основе состояния окружающей среды, которые робот может получить из установленных на борту датчиков и камер. Данная работа приводит пример разработки такого функционала, цель которого обеспечить возможность для робота, в случае потери связи с оператором, продолжить движение в обратном направлении в автономном режиме, с целью либо вернуться в точку, в которой сигнал был еще доступен, либо же вернуться в точку начала движения.*

*Ключевые слова:* компьютерное зрение, автономное вождение, SLAM, Gazebo, ROS 2, камеры глубины, RTAB-Map.

## Development of an Autonomous Driving and Mapping System for the ROIN-RTS-100 Robot

N.S. Pugachev✉

Voronezh State University, Voronezh, Russia

*The task of controlling a robot in rough terrain in condition of a large distance between the operator and the robot is usually complicated by interference of the control signal and communication problems. Overall, situations of complete loss of communication are not excluded. In this case, the robot's behavior may be unpredictable and lead to an accident and breakdown. To avoid such scenarios it is desirable to have the functionality of autonomous decision-making by the robot based on the state of the environment, which the robot can perceive from sensors and cameras installed on board. This work provides an example of the development of such functionality, the purpose of which is in case of loss of communication with the operator to provide the ability for the robot to continue moving in the opposite direction in autonomous mode, with the purpose of either returning to the point where the signal was still available, or returning to the starting point of the movement.*

*Keywords:* computer vision, autonomous driving, SLAM, Gazebo, ROS 2, depth camera, RTAB-Map.

### Введение

Современные автономные роботизированные системы сталкиваются с необходимостью эффективной навигации в сложных, динамически изменяющихся средах – от промышленных цехов до обширных и неизвестных пространств пересеченной местности. Несмотря на прогресс в алгоритмах SLAM [1] (Simultaneous Localization and Mapping) и планирования движения, разработка и отладка комплексов систем навигации все же выполняется индивидуально для каждого робота и задач, которые ему необходимо выполнять. Это связано с особенностями и возможностями

конструкции, вычислительными мощностями и оборудованием, установленном на конкретной машине. Ключевые проблемы, которые влияют на работу этих систем, такие как устойчивость к шумам сенсоров, долгосрочная согласованность карт и адаптация к присутствию в сцене нестационарных объектов, остаются актуальными. Кроме того, учитывая затраты, разработку и тестирование такой системы целесообразно проводить именно в симуляции.

Целью данной работы является разработка масштабируемого программного комплекса позволяющего симулировать физическую и визуальную среду, в которой действует робот, обеспечить механизм картографии и локализации с возможностью навигации по составленной карте, а также разработать автоматизированный алгоритм тестирования успешности навигации в рамках работы данного комплекса.

### **Методы исследования**

Необходимо отметить, что цель данной работы состоит не только в обеспечении надежного инструмента картографии в ранее неизвестном окружении, со всеми необходимыми инструментами визуализации и контроля, но и в построении масштабируемой системы, способной работать с разным оборудованием, и при необходимости включить в себя дополнительное программное обеспечение, например, для обработки шумных данных, получаемых во время работы в реальных условиях. Эти требования являются обоснованием выбора системы ROS2 [2]. ROS2 – операционная система с открытым исходным кодом, используемая в робототехнике. Эта операционная система позволяет осуществлять запуск разных компонентов программного обеспечения робота в качестве независимых модулей и налаживать коммуникацию между этими модулями. В результате такая система является распределенной, что положительно сказывается на ее отказоустойчивости и масштабируемости.

Выбор среды симуляции Gazebo [3] обусловлен совместимостью с ROS2 и наличием открытого исходного кода. Таким образом, комбинация систем ROS2 и Gazebo позволяет работать с цифровым двойником робота и проводить дальнейшую отладку системы навигации.

В ходе разработки программного комплекса был проведен анализ существующих реализаций SLAM алгоритмов: ORB-SLAM2 [4], RGBDSLAMv2 [5] и RTAB-Map [6], и др. В результате этого анализа, с учетом требований к масштабируемости системы и эффективности вычислений, была выбрана реализация SLAM из фреймворка RTAB-Map.

Далее возникает необходимость выбора алгоритма построения маршрута и контроллера (алгоритма, обеспечивающего движение по этому маршруту). Стоит отметить, что, в связи с тем, что данные алгоритмы обеспечивают автономное движение, в них должны быть учтены особенности конструкции конкретной робота. Так как робот ROIN-RTS-100 стоит на гусеничных шасси, в ходе сравнения функционалов алгоритмов была выбрана комбинация алгоритма построения маршрутов Theta Planner\* [7] и контроллера MPPI [8].

### **Агрегация данных с помощью RTAB-Map**

В ходе работы был проведен анализ нескольких инструментов картографии. Самым подходящим был выбран фреймворк RTAB-Map. RTAB-Map (Real-Time Appearance-Based Mapping) – это фреймворк для одновременной локализации и картографии (SLAM), ориентированный на работу в трехмерных средах с

использованием данных, получаемых с различных сенсоров, таких как RGB-D камеры, лидары и стереопары. Его ключевая особенность заключается в комбинации методов appearance-based mapping и loop closure [9]. Отличительными чертами этого алгоритма являются:

1. Возможность агрегации данных, полученных из конфигурируемого набора датчиков (Лидары, камеры глубины, инерционные датчики).
2. Механизм Working/Long-Term Memory (критично для долгосрочной навигации в больших пространствах):
  - рабочая память хранит актуальные ключевые кадры для быстрого анализа;
  - долговременная память архивирует старые данные, сохраняя информацию для обнаружения петель (loop closure) даже в крупных средах;
  - автоматическое удаление избыточных данных, предотвращающее перегрузку вычислительных ресурсов.
3. Интеграция с ROS2.
4. Возможность вычисления визуальной одометрии (особенно важно для работы с гусеничной техникой).
5. Высокая, по сравнению с другими алгоритмами, эффективность вычислений.

Алгоритм строит карту на основе визуальных особенностей среды, используя дескрипторы SIFT [10], SURF [11] или ORB [12] для анализа сходства между кадрами. При обнаружении совпадений в текстурах или структуре (loop closure) система корректирует накопленные ошибки одометрии, оптимизируя граф поз через GTSAM [13]. Возможность такой коррекции особенно важна в долгосрочных сценариях работы программного комплекса, в которых робот возвращается в ранее изученные зоны: RTAB-Map сравнивает текущие данные с историей, хранящейся в «долговременной памяти», и переносит релевантные ключевые кадры в «рабочую память» для ускорения loop closure. Appearance-based подход обеспечивает не только геометрическую, но и семантическую информацию, создавая текстурированные 3D-карты, которые отражают визуальные характеристики объектов (цвет, текстуру), что полезно для задач навигации в средах, не имеющих четкой структуры (пересеченной местности). Благодаря балансу между точностью и производительностью, а также поддержке многомодальных данных, RTAB-Map эффективен в сложных динамических средах, где чисто геометрические методы или алгоритмы, ориентированные на один тип сенсора, могут давать сбои. На рисунке 1 представлена визуализация результата агрегирования кадров, собранных с камеры глубины во время движения робота, в трехмерный скан.

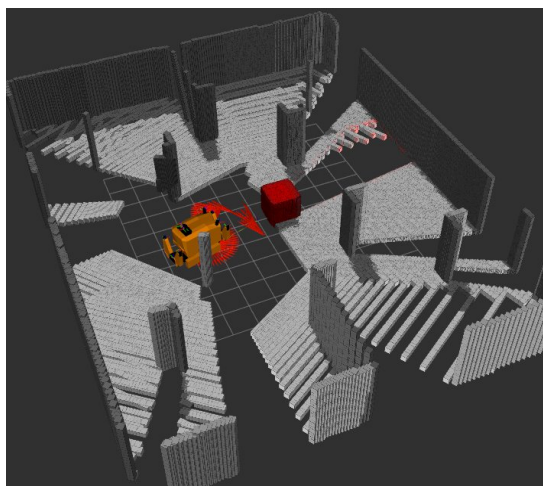


Рисунок 1. Пример сканирования окружения RTAB-Map в процессе движения

### Локальная и глобальная карты стоимости

Для построения маршрута, по полученным на предыдущем шаге данным о среде, необходимо преобразовать эту карту в карту стоимости [14]. Карта стоимости представляет собой изображение, на котором каждому пикселю присваивается определенная цена прохождения робота по местности, которой этот пиксель соответствует. Для препятствий значение стоимости максимально. Карты затем используются для планирования оптимальной траектории маршрута и движения по ней. Следует отметить, что для удобства восприятия карта может быть представлена в многослойном виде. В данной работе используется три слоя:

1. Статичный слой – содержит в себе все известные препятствия, полученные из скана пространства RTAB-Map (1).

2. Слой препятствий  $M_{(x,y)}^{obs}$  – содержит препятствия, найденные из облака точек на камеры (1) глубины и обновляется в реальном времени.

3. Слой инфляции  $M_{(x,y)}^{inf}$  – слой, добавляющий буферную зону вокруг препятствий, для исключения ситуации движения робота в опасной близости к ним (2) [15].

Ниже приведены формулы расчета слоев:

$$M_{(x,y)}^{obs} = \begin{cases} 254, & (x^w, y^w) - \text{содержит препятствие,} \\ 0, & (x^w, y^w) - \text{свободно,} \end{cases} \quad (1)$$

$$M_{(x,y)}^{inf} = \exp(-c \cdot (dist(x^w, y^w) - R)) \cdot 254, \quad (2)$$

где  $c$  – экспоненциальный коэффициент затухания по радиусу инфляции,  $R$  – заданный в метрах радиус вписанной окружности проекции робота в горизонтальной плоскости,  $(x^w, y^w)$  местоположение робота, соответствующее индексам  $(x, y)$  на карте,  $dist$  – Евклидово расстояние от конкретного местоположения робота  $(x^w, y^w)$  до ближайшего препятствия.

Предположим, мы имеем робота с радиусом 1 метр, карту разрешением  $30 \times 30$  пикселей, где площадь одного пикселя составляет  $0,25 \text{ см}^2$ . Тогда при установке параметра  $c = 1,5$  можно получить функцию инфляции (рис. 2), слой инфляции и карту стоимости, показанные на рисунке 3.

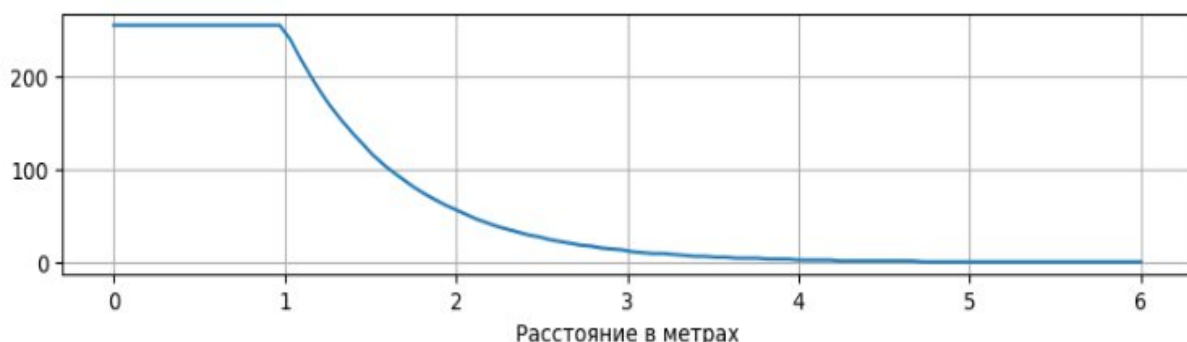


Рисунок 2. График функции инфляции

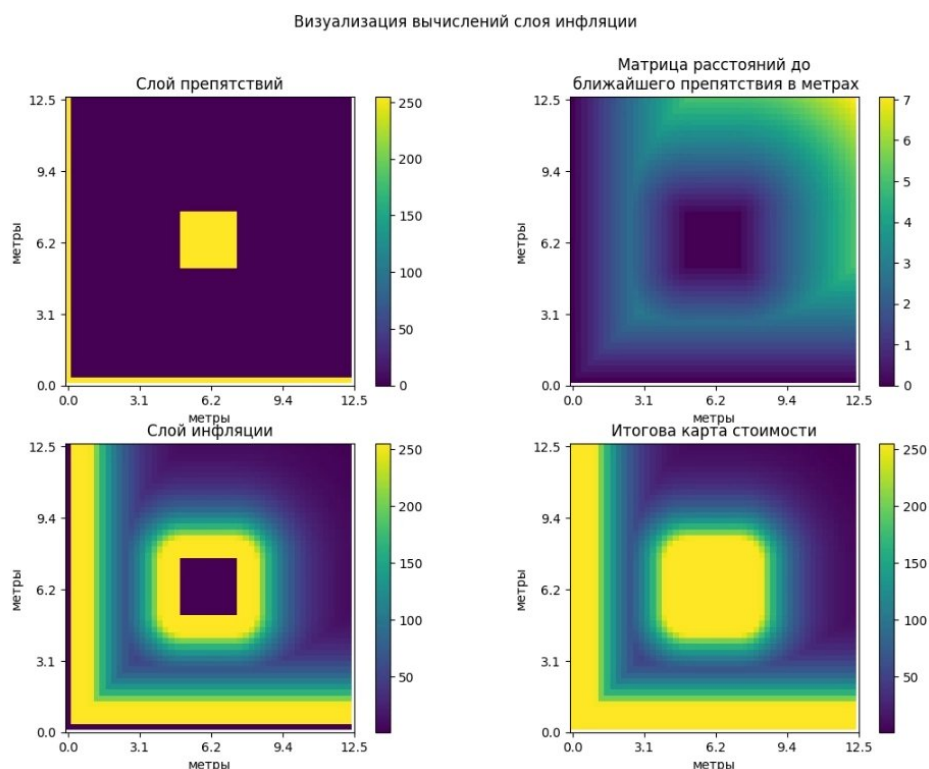


Рисунок 3. Визуализация расчета карты стоимости

Как правило, в системах навигации используются две карты: локальная и глобальная. Глобальная карта (статичный слой) используется для построения оптимального маршрута от начальной точки до цели в рамках заранее полученной карты (например, созданной с помощью SLAM алгоритма). Эта карта (рис. 4) отражает структуру окружающей среды, включая статические препятствия (стены, зоны с ограниченным доступом).

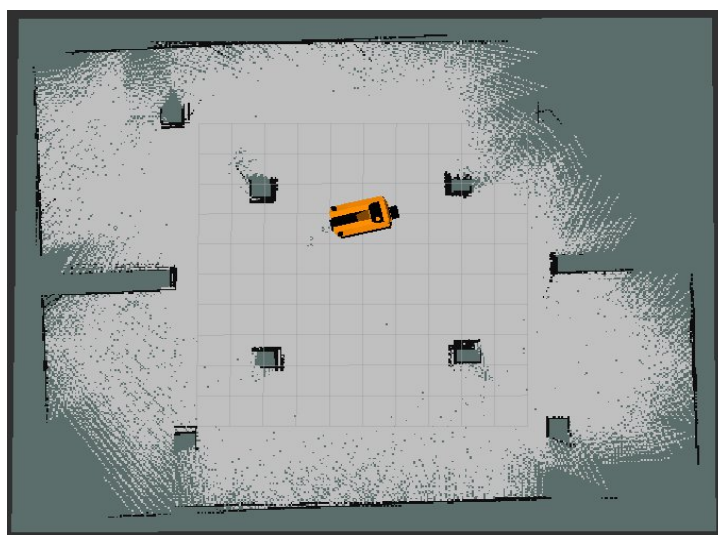


Рисунок 4. Визуализация глобальной карты (статичного слоя)

Локальная карта, показанная на рисунке 5, также представляет собой карту стоимости, состоящую из слоя препятствий и слоя инфляции. Исходя из названия

понятно, что она существует только в некоторой области вокруг робота. Локальная карта необходима для интеграции информации, получаемой с датчиков в режиме реального времени, в то время как в глобальной карте хранится информация, полученная на предыдущих шагах работы системы. Визуализацию локальной и глобальной карт можно увидеть на рисунке 5. На рисунке чем темнее участок, тем выше стоимость локальной карты.

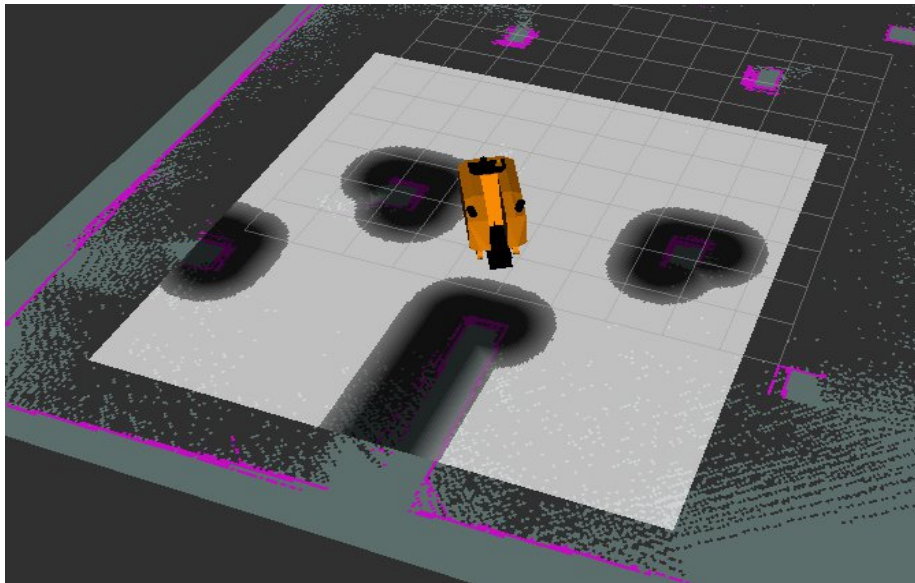


Рисунок 5. Визуализация локальной карты в абсолютных значениях, относительно глобальной

Наличие локальной карты позволяет производить краткосрочное планирование, которое используется для управления движением на коротких дистанциях (например, в пределах 3–5 метров вокруг робота). Цель такого планирования состоит в постоянном уточнении первоначальной траектории, основанной на глобальной карте. А необходимость использования для этого локальной карты состоит не только в том, что за время фиксации локации на глобальной карте и повторному проходу робота по ней могли появиться дополнительные препятствия, но и в том, что во время движения робота неизбежно накапливается погрешность в определении его позиции, поэтому построенный изначально на основе глобальной карты маршрут по определению не может оптимальным.

Следует отметить, что в процессе движения, как уже было сказано выше, глобальная карта сохраняется, равно как и данные об одометрии, что позволяет в случае утраты связи с роботом загрузить сохраненную в процессе движения карту, а затем использовать последнюю точку нахождения робота для локализации на ней. После чего появляется возможность проложить маршрут к месту начала пути.

### **Построение маршрутов и автономное управление**

При наличии данных глобальной карты, известной позиции робота на ней и наличии точки, в которую необходимо попасть, появляется возможность проложить маршрут из точки положения робота в точку назначения. Для этого используется планировщик Theta-Star Planner. Theta-STAR Planner – это алгоритм поиска пути, расширяющий возможности A-STAR [15], позволяет строить более плавные и естественные траектории без строгого следования узлам сетки. Theta Planner\* разбивает пространство на регулярную сетку (например, квадратные или шестиугольные ячейки),

где каждая ячейка соответствует узлу. Узлы могут быть проходимыми (без препятствий) или непроходимыми. Размер ячейки определяет детализацию карты – мелкая сетка повышает точность, но увеличивает вычислительные затраты. Затем строится маршрут наименьшей стоимости. Существуют два параметра функции оценки итогового маршрута: критика длины и критика стоимости. Ключевая особенность Theta Planner\* – проверка прямой видимости (line-of-sight) между узлами, что сокращает изломы пути. Стоит обратить внимание, что сокращение изломов пути, один из главных критериев для роботов на гусеничных шасси. Это связано с особенностями конструкции и тем что в реальности такому роботу проще изначально повернуться к цели, а затем продолжить движение по прямой, так как каждый новый поворот связан с дополнительным риском проскальзывания гусениц, что ведет к возможным сбоям в расчете одометрии и возможным проблемам с локализацией.

После построения маршрута минимальной стоимости, необходимо проследовать по нему до целевой точки. За следование робота по маршруту отвечает контроллер – специальный алгоритм, который обеспечивает управление. Стоит отметить, что именно контроллер получает на вход локальную карту стоимости, что позволяет ему уточнять и оптимизировать изначально построенный маршрут.

В качестве контроллера в этой работе используется алгоритм MPPI. Ключевые особенности алгоритма:

- генерирует множество «виртуальных» траекторий (сэмплов) на основе текущего состояния системы;
- для каждой траектории рассчитывается стоимость (cost) с учётом цели, препятствий и других ограничений;
- не требуется аналитический расчёт производных, в отличие от классического MPC.

Для корректной работы контроллера необходимо рассчитать параметры генерации сэмплов, а также определить расчет их стоимости. В качестве параметров генерации выступают следующие:

- 1) model\_dt = 0,1 – длина каждого временного шага в секундах;
- 2) time\_steps = 56 – количество временных шагов (точек) в траекториях-кандидатах;
- 3) batch\_size = 2000 – количество случайно выбранных траекторий-кандидатов из текущей оптимальной последовательности управления в данной итерации.

Исходя из произведения параметров 1 и 2 видно, что горизонт планирования контроллера составляет 5,6 секунд, визуализация генерации сэмплов показана на рисунке 6.

Определить расчет стоимости траекторий удобно как сумму функций (критик), каждая из которых оценивает траекторию в определенном ключе. Полная конфигурация контроллера представлена в приложении, однако стоит обратить внимание на основные функции-критики:

1. Obstacle critic – штраф, накладываемый на пути, неминуемо ведущие к коллизиям.
2. Cost critic – штраф, определяемый значениями локальной карты стоимости, по которым движется робот.
3. Prefer Forward Critic – штраф за движение в обратном направлении.
4. Path Align Critic – штраф в случае отклонения от изначального маршрута.

На рисунке 6 изображена визуализация сгенерированного набора траекторий MPPI контроллера. Каждая траектория состоит из набора точек равного количеству временных шагов, оттенки при визуализации точек траектории варьируются в

зависимости от расстояния от центра робота до горизонта планирования, равного произведению количеству временных шагов на их длину по времени. Кроме того, на рисунке показаны значения локальной карты стоимости. Светло синим цветом показаны участки локальной карты, в которых вероятность столкновения с препятствием гарантирована. Значение стоимости в этих участках максимально. Далее красным цветом показана инфляция стоимости.

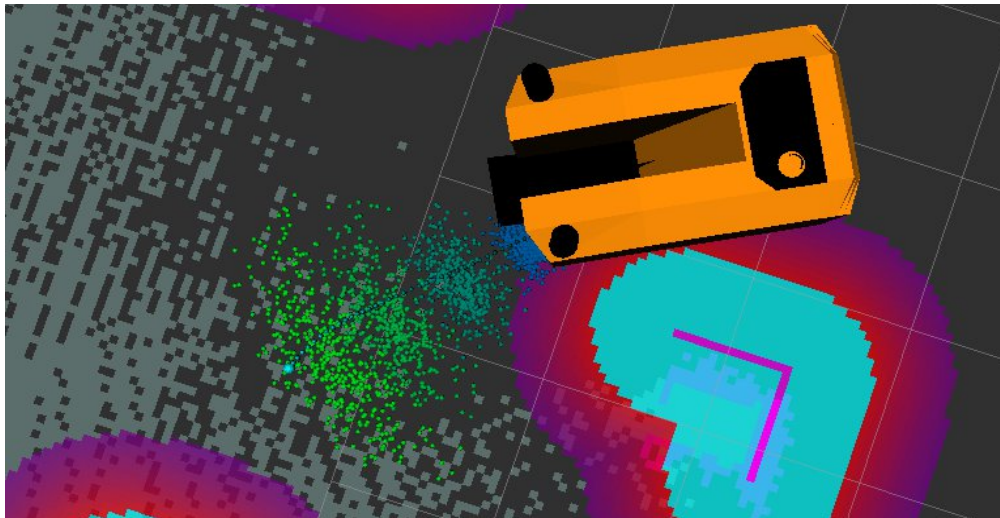


Рисунок 6. Визуализация сгенерированного набора траекторий

Известно, что основная проблема в управлении роботами на гусеничных шасси состоит в трудности расчета одометрии. Трудность заключается в том, что гусеницы могут проскальзывать, и это проскальзывание не может быть предсказано. Проскальзывание вызвано как самой конструкцией гусеничных шасси, так и грунтом, по которому они передвигаются. Стоит отметить, что модель движения гусениц является частным случаем дифференциального привода. Отличие состоит в учете коэффициента скольжения по поверхности. Поскольку в данном решении задачи навигации используется визуальная одометрия фреймворка RTAB-Map, без учета одометрии, рассчитываемой на основе положения гусениц, скольжением можно пренебречь и использовать модель дифференциального привода.

### Тестирование системы

На основе перечисленных выше методов была разработана система автономного движения для робота ROIN-RTS-100. В связи с ранее описанным сценарием использования, система предполагает два режима работы:

1. Построение карты местности используя ручное управление.
2. Автономное движение к указанным точкам на основе глобальной карты, построенной на предыдущем шаге, и локальной карты, обновляемой в режиме реального времени.

На основе второго компонента, был разработан алгоритм тестирования системы. В качестве входа алгоритм тестирования принимает 3D модель окружения, построенные по этому окружению с помощью RTAB-Map трехмерный скан и глобальную карту стоимости, а также набор локаций, каждая из которых может являться точкой начала движения или выступать как его цель. Далее происходит случайный выбор пар (начало-конец). Для каждой пары запускается сессия работы системы навигации, результат которой бинарный, указывает на то, достигнута цель или

нет. По результатам тестирования в двух окружениях и определенной в процессе тестирования оптимальной конфигурации системы навигации, а именно радиусов и инфляции локальной карты стоимости и коэффициентов весов функций-критик для МРРІ контроллера, был получен результат 100% достижения целей.

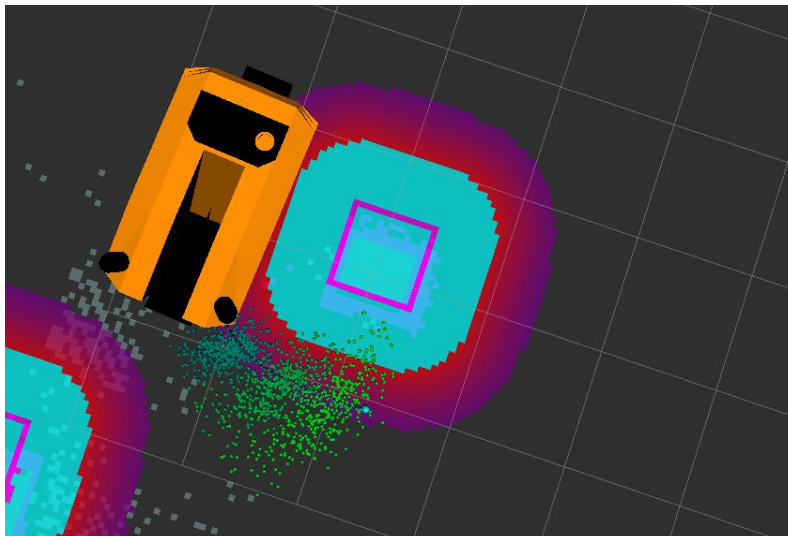


Рисунок 7. Визуализация набора траекторий и траектории самой низкой стоимости при объезде препятствия

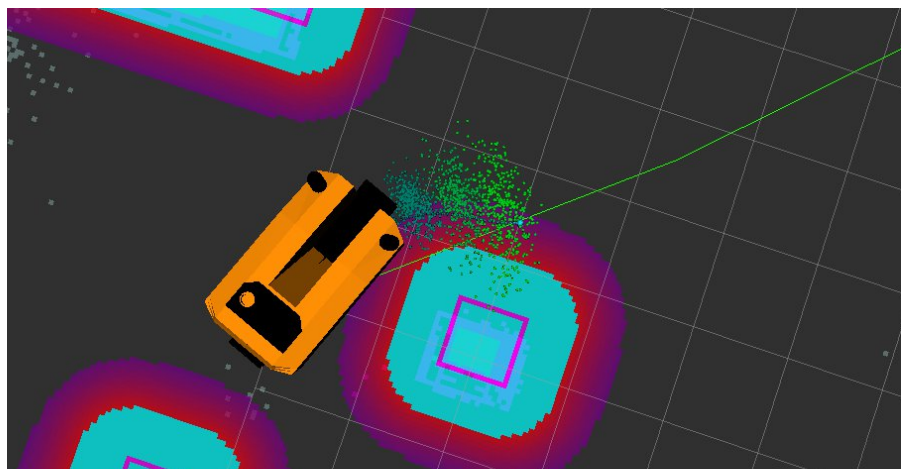


Рисунок 8. Пример корректировки изначального пути (зеленая линия) для снижения общей стоимости маршрута (робот объезжает зону с высокой стоимостью)

RTAB-Map проводит обновление глобальной карты с частотой 1 hz. Согласно заявленной конфигурации робота, максимальная скорость движения робота 1,67 м/с. В итоговой конфигурации МРРІ контроллер способен работать с частотой 10 hz. Получение необходимой для работы контроллера локальной карты не является проблемой, т.к. она вычисляется в режиме реального времени в соответствии с FPS камеры глубины, который выставляется в соответствии с частотой работы контроллера. Исходя из приведенных выше замеров видно, что соблюден сценарий использования глобальной и локальной карт.

### Заключение

В ходе работы было разработан и оценен программный комплекс, интегрирующий RTAB-Map, алгоритм МРПИ и симулятор Gazebo. Данный инструмент позволяет посредством симуляции осуществлять управление роботом, визуализировать показания датчиков, проводить картографию, сохранять и визуализировать полученную карту. Для оптимизации параметров системы был разработан алгоритм тестирования, позволивший определить оптимальные параметры системы. Была проведена оценка производительности системы. В итоговой конфигурации система позволяет выполнять загрузку составленной ранее карты для осуществления навигации к вводимым пользователем координатам в автономном режиме.

### СПИСОК ИСТОЧНИКОВ

1. Stachniss C. Simultaneous Localization and Mapping / C. Stachniss, J.J. Leonard, S. Thrun // Springer Handbook of Robotics. – Cham: Springer, 2016. – P. 1153–1176.
2. Robot Operating System 2: Design, Architecture, and Uses in the Wild / S. Macenski, T. Foote, B. Gerkey [et al.] // Science Robotics. – 2022. – Vol. 7, No. 66. – URL: <https://doi.org/10.1126/scirobotics.abm6074> (дата обращения: 23.06.2025).
3. Gazebo [Электронный ресурс]. – URL: <https://gazebo.org/home> (дата обращения: 23.06.2025).
4. Mur-Artal R. ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras / R. Mur-Artal, J.D. Tardós // IEEE Transactions on Robotics. – 2017. – Vol. 33, Iss. 5. – P. 1255–1262.
5. 3-D Mapping with an RGB-D Camera / F. Endres, J. Hess, J. Sturm [et al.] // IEEE Transactions on Robotics. – 2017. – Vol. 30, Iss. 1. – P. 177–187.
6. Labbé M. RTAB-Map as an Open-Source Lidar and Visual Simultaneous Localization and Mapping Library for Large-Scale and Long-Term Online Operation / M. Labbé, F. Michaud // Journal of Field Robotics. – 2019. – Vol. 36. – P. 416–446.
7. Theta\*: Any-Angle Path Planning on Grids / K. Daniel, A. Nash, S. Koenig, A. Felner // Journal of Artificial Intelligence Research. – 2010. – Vol. 39. – P. 533–579.
8. Aggressive Driving with Model Predictive Path Integral Control / G. Williams, P. Drews, B. Goldfain [et al.] // IEEE International Conference on Robotics and Automation (ICRA). – IEEE, 2016. – P. 1433–1440.
9. Labbé M. Appearance-Based Loop Closure Detection for Online Large-Scale and Long-Term Operation / M. Labbé, F. Michaud // IEEE Transactions on Robotics. – 2013. – Vol. 29, Iss. 3. – P. 734–745.
10. Bay H. SURF: Speeded Up Robust Features / H. Bay, T. Tuytelaars, L.V. Gool // Computer Vision – ECCV 2006: 9<sup>th</sup> European Conference on Computer Vision, Graz, Austria, May 7–13, 2006, Proceedings, Part I. – Berlin, Heidelberg: Springer, 2006. – P. 404–417.
11. Lowe D.G. Distinctive Image Features from Scale-Invariant Keypoints / D.G. Lowe // International Journal of Computer Vision. – 2004. – Vol. 60. – P. 91–110.
12. ORB: An Efficient Alternative to SIFT or SURF / E. Rublee, V. Rabaud, K. Konolige, G. Bradski // International Conference on Computer Vision. – IEEE, 2011. – P. 2564–2571.
13. Dellaert F. Factor Graphs and GTSAM / F. Dellaert, M. Kaess // GTSAM [Электронный ресурс]. – URL: <https://gtsam.org/tutorials/intro.html> (дата обращения: 23.06.2025).

14. Lu D.V. Layered Costmaps for Context-Sensitive Navigation / D.V. Lu, D. Hershberger, W.D. Smart // IEEE/RSJ International Conference on Intelligent Robots and Systems. – IEEE, 2014. – P. 709–715.
15. Inflation Costmap Plugin // Documentation – ROS Wiki [Электронный ресурс]. – URL: [https://wiki.ros.org/costmap\\_2d/hydro/inflation](https://wiki.ros.org/costmap_2d/hydro/inflation) (дата обращения: 23.06.2025).
16. Neto C.A.P. A-Star Path Planning Simulation for UAS Traffic Management (UTM) Application / C.A.P. Neto, G. de Carvalho Bertoli, O. Saotome // arXiv [Электронный ресурс]. – URL: <https://arxiv.org/abs/2107.13103> (дата обращения: 23.06.2025).

#### ИНФОРМАЦИЯ ОБ АВТОРЕ

**Пугачев Николай Сергеевич**, аспирант 2 курса кафедры математических методов исследования операций, Воронежский государственный университет, Воронеж, Россия.  
*e-mail:* [pugachev1997@bk.ru](mailto:pugachev1997@bk.ru)