

## РАЗРАБОТКА НАВИГАЦИОННОЙ СИСТЕМЫ РОБОТА НА ПОВЕРХНОСТИ ЛУНЫ С ИСПОЛЬЗОВАНИЕМ АЛГОРИТМОВ ОДНОВРЕМЕННОЙ ЛОКАЛИЗАЦИИ И КАРТОГРАФИИ (SLAM)

Спирина Н.С., Спиринов В.В., Тарачков М.В., Шарамет А.А.

DOI: [10.31618/nas.2413-5291.2019.3.50.123](https://doi.org/10.31618/nas.2413-5291.2019.3.50.123)

### Аннотация

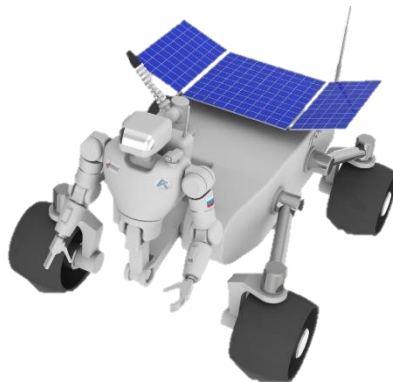
В данной статье рассматривается возможность применения алгоритмов одновременной локализации и картографии (SLAM) для навигации робототехнических систем на поверхности Луны. Автор на прототипе колесного робота показывает, какое техническое и программное обеспечение используется для реализации системы навигации с применением алгоритмов SLAM. Приводится пример использования программной платформы Robot Operating System (ROS) для управления роботом. В заключении автор указывает на возможные улучшения представленной системы.

**Ключевые слова:** робототехника, манипулятор, 3D-модель, Луна, навигация, SLAM, ROS, программное обеспечение

### Введение

Освоение Луны является важным шагом для дальнейшего проникновения человека в космическое пространство. Данная задача сложна тем, что в настоящее время проблематично длительное время поддерживать нахождение на Луне космонавтов для проведения исследований.

[1] Поэтому научное сообщество прибегает к использованию робототехнических систем. Вариантом такой системы может быть робот на колесной платформе с закрепленной на ней торсовой частью, обладающей двумя манипуляторами [2].



*Рис. 1. Луноход с торсовой частью антропоморфного робота для исследований каменистых, скальных пород и с возможностью ремонта (себя и другой лунной техники).*

Актуальной является проблема управления таким роботом, в частности, решение задачи навигации. Супервизорное управление осуществляется с задержкой порядка трёх секунд и чувствительно к качеству связи с Землей. Поэтому возможным вариантом является автономное управление. На Земле навигация робота была бы решена при помощи данных со спутников о его местоположении. Обладая развитой системой базовых станций можно получить точность позиционирования до 2 см в режиме кинематики реального времени (РТК). Но в условиях нахождения на поверхности Луны данный подход неприменим (по крайней мере, до создания в околослунной области специальной навигационной системы).

Поэтому одним из возможных решений может быть использование алгоритмов одновременной локализации и картографии (SLAM). [3] С их помощью робот, получая данные от датчиков (лазерный сканер, камеры, одометры), способен построить карту окружающего пространства и определить свое местоположение.

В данной статье описан вариант колесной платформы, которая по данным от лазерного сканера и одометров осуществляет автономную навигацию.

### Программная платформа ROS

В настоящее время для управления роботами широко используется программная платформа Robot Operating System (ROS). [4] Это свободное программное обеспечение, которое работает под управлением ОС Linux, позволяет осуществлять взаимодействие разнородных программ, которые могут быть написаны на разных языках, посредством механизма обмена сообщениями. ROS имеет большое и развивающееся сообщество и готовые пакеты (отдельные программы, модули), которые можно использовать за основу при разработке собственных. В том числе, ROS содержит большое количество пакетов, в которых реализован алгоритм SLAM.

Робот в ROS описывается при помощи формата Unified Robot Description Format (URDF). [5] Данный формат вводит особенности при создании трехмерной модели робота.

### Создание 3D-модели колесного робота

Трехмерная модель колесного робота была создана в программе Solidworks. Важно выделить отдельными элементами колеса робота и лидар. Модели элементов следует создавать двух видов. Одни должны быть максимально простыми. Это связано с тем, что при планировании движений

робота происходит расчет возможных соударений элементов робота с объектами окружающей среды. Чем модель проще, тем быстрее происходит расчет. Второй вариант моделей элементов может быть более изящным, поскольку используется только для визуализации и в расчетах соударений не участвует.

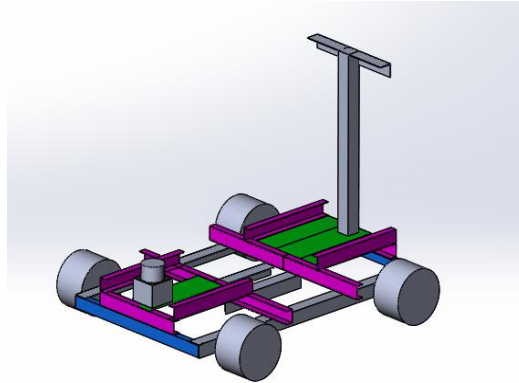


Рис 2. Трехмерная модель робота в программе Solidworks.

Для САПР Solidworks существует плагин Solidworks2URDF Exporter, который осуществляет конвертацию трехмерной модели в формат URDF. [6] В результате чего подготавливается пакет ROS, содержащий описание робота.

Визуализация робота в программе RViz

RViz – это программа для визуализации, которая входит в программную платформу ROS. RViz позволяет визуализировать модель робота, карту местности, облака точек, отображать вспомогательные объекты и осуществлять управление моделью робота.

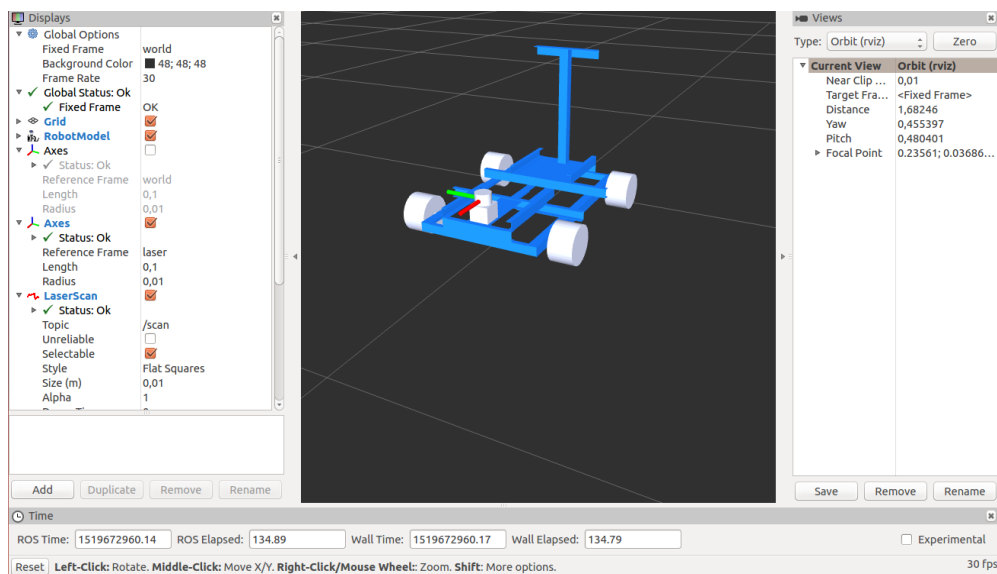


Рис 3. Визуализация модели робота в программе RViz.

### Сборка робота

По трехмерной модели был собран реальный робот. Конструкция робота была сделана из элементов конструктора, потому что это позволяет

осуществлять быстрое прототипирование. Были использованы всенаправленные колеса, или колеса Илона, которые позволяют роботу осуществлять перемещение в любом направлении.



*Рис 4. Всенаправленные колеса.*

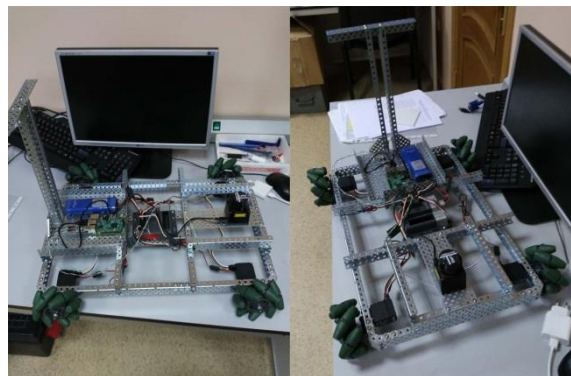
В качестве приводов использовались коллекторные электродвигатели с редуктором и встроенным инкрементным энкодером для определения скорости вращения. Реальная

конструкция транспортной платформы на Луне может значительно отличаться от приведенной, но это связано решением самой навигационной задачи.



*Рис 5. Электродвигатель с встроенным энкодером.*

Общий вид колесного робота приведен на рисунке 6.



*Рис 6. Колесный робот.*

Также на раме робота был закреплен лазерный сканер HOKUYO URG-04LX-UG01. Данный

дальномер имеет 1 луч, точность 3 см., дальность от 0.3 до 5 м, угол сканирования 240 градусов.



Рис 7. Лазерный дальномер Нокиуо.

### Электроника

Электрическая схема колесного робота создавалась исходя из необходимости управления четырьмя приводами всенаправленных колес. Она состоит из драйверов электродвигателей на основе мостовой схемы, которая обеспечивает вращение колес в обе стороны, а также регулировку скорости вращения при помощи ШИМ. Для измерения скорости вращения колес были использованы инкрементные энкодеры, расположенные на каждом колесе. Управления драйверами электродвигателей и чтение сигналов от энкодеров осуществляет микроконтроллер STM32F103, который находится внутри контроллера VEX. Для

осуществления высокоуровневого управления колесным роботом используется микрокомпьютер Raspberry Pi 3. Он располагается на роботе и взаимодействует с микроконтроллером посредством преобразователя USB-UART. Поскольку помимо автономного управления предполагалось управление колесным роботом в супервизорном режиме, а также для визуализации текущего положения робота и состояния окружающей его среды в схему был включен компьютер оператора, который связан с Raspberry Pi 3 через роутер. Полная электрическая схема приведена на рис. 8.

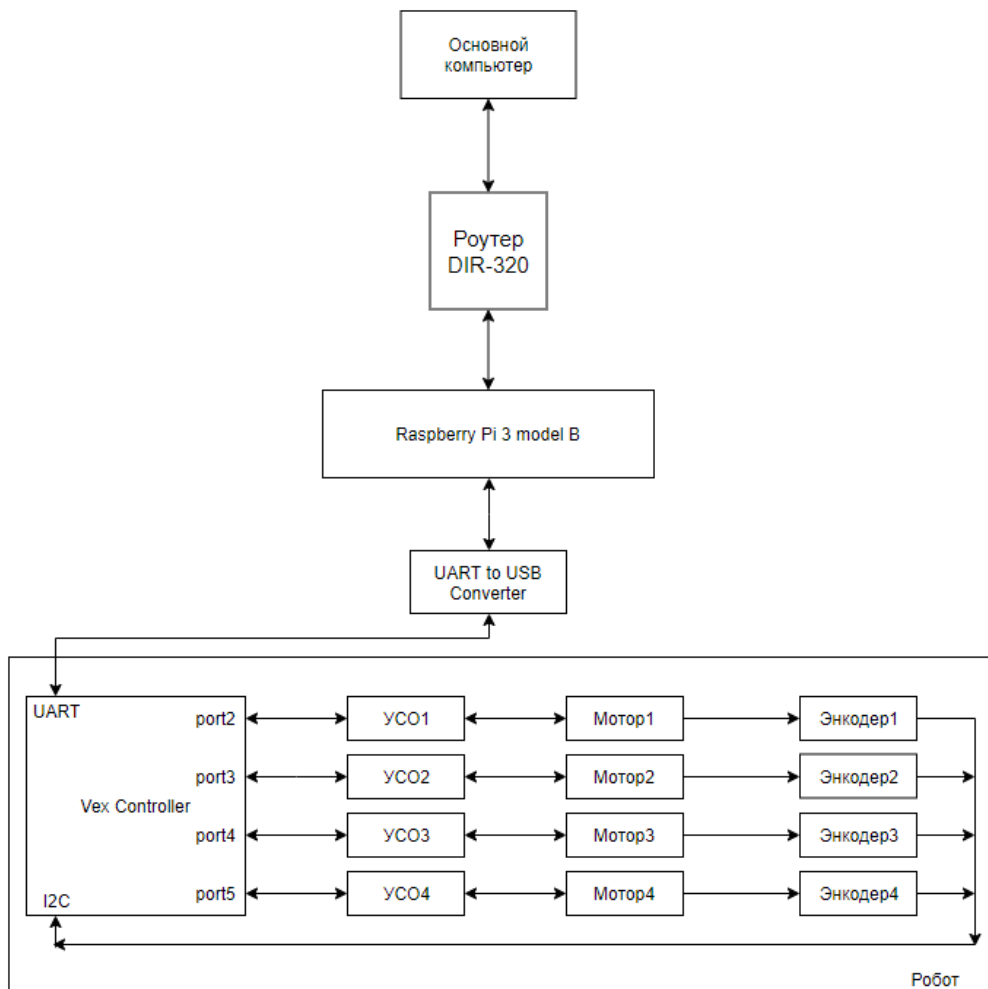


Рис 8. Электрическая схема робота.

Питание электрических элементов осуществлялось от расположенных на роботе аккумуляторов.

Низкоуровневые драйвера

Разработка программного обеспечения для контроллера VEX осуществлялась на языке C. Была реализована схема управления с обратной связью по скорости на основании данных от установленном на каждом колесе инкрементном энкодере.

По собственному протоколу контроллер VEX передает данные о скорости вращения колес на микрокомпьютер Raspberry Pi 3. Также через него осуществляется задание уставки скорости вращения для каждого из колес.

Ответная часть на микрокомпьютере Raspberry Pi 3 была реализована с использованием пакета ROS Control.[7] Данный пакет осуществляет

взаимодействие с виртуальным последовательным портом и принимает данные от более высокоуровневых программ, каковой является программа управления движением робота.

Использование алгоритма семейства SLAM

Для тестирования на собранном колесном роботе был выбран алгоритм построения карты GMapping, потому что он является наиболее простым в реализации и показывает удовлетворительные результаты. Данный алгоритм получал в качестве входных данных облако точек с лазерного сканера и показания одометров. Алгоритм Adaptive Monte Carlo Localization (AMCL) использовался для определения местоположения робота на карте и определения пути к заданной точке.

Управление роботом осуществлял пакет move\_base [8].

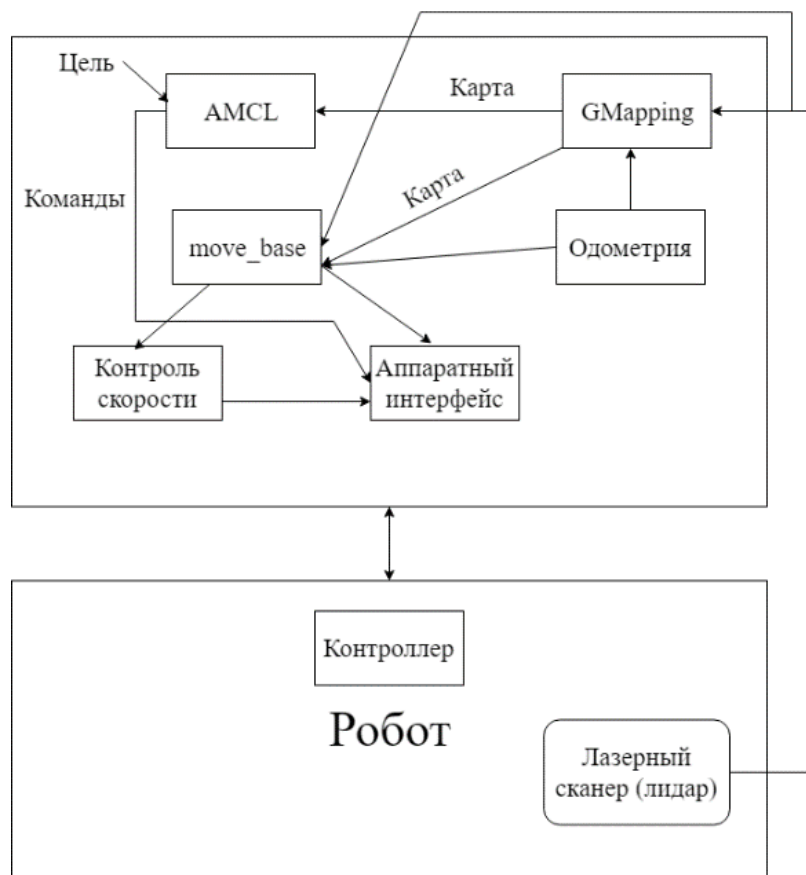


Рис 9. Схема взаимодействия оборудования и программного обеспечения.

Результаты

Были проведены испытания колесного робота, в ходе которых он должен был переместиться из точки А в точку В, ориентируясь на показания

сенсоров. Оператор мог наблюдать процесс построения карты окружающего пространства на компьютере супервизорного управления в программе RViz.

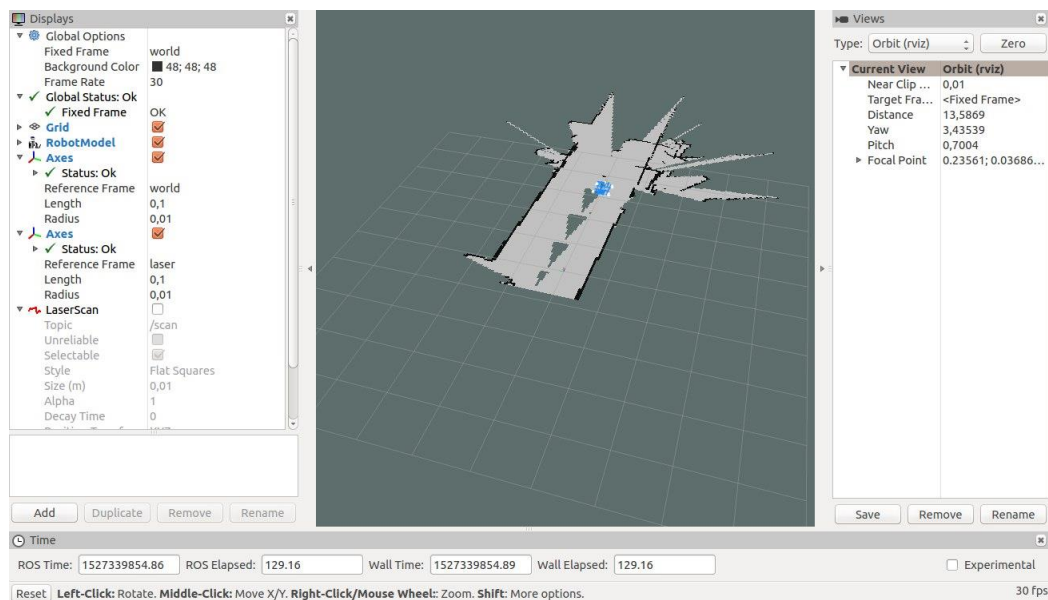


Рис 10. Карта местности в программе RViz.

### Выводы

Алгоритмы семейства SLAM могут быть использованы в системе навигации робототехнических систем на поверхности Луны, где отсутствуют спутниковые системы навигации, а управление с Земли осложняется задержкой сигнала порядка 3 секунд. Применение алгоритмов SLAM в системе навигации робота позволит задавать с Земли только точку назначения.

Программная платформа ROS упрощает реализацию алгоритмов семейства SLAM и управления роботом.

Для получения более плотного облака точек могут быть использованы многолучевые лидары, например, Velodyne VLP32c, Ouster OS 1-64.

Для получения цветов облака точек может быть использована стереосистема камер или омникамера, например, Ladybug 5+.

### Литература

1. А. В. Сидоров, Д. В. Щеголькова, «Освоение Луны: политика, коммерческий интерес или научные исследования»
2. O. Saprykin. About actual robots for scientific research of Moon and creation of a circumlunar infrastructure. / The 8th CSA-IAA Conference on Advanced Space Technology. Shanghai, China, 2019.

3. Котов К.Ю., Мальцев А.С., Соболев М.А. «Метод решения задачи одновременной локализации и построения карты при движении автономного мобильного робота»

4. WHY ROS? // ROS.ORG. [Электрон. ресурс]. Режим доступа: <http://www.ros.org/is-ros-for-me/> (дата обращения 15.08.2019)

5. Joseph L. Mastering ROS for robotics programming. Packt Publishing Ltd., p. 67

6. SolidWorks to URDF Exporter // WIKI.ROS.ORG. [Электрон. ресурс]. Режим доступа: [http://wiki.ros.org/sw\\_urdf\\_exporter](http://wiki.ros.org/sw_urdf_exporter) (дата обращения 16.08.2019)

7. Тарачков М.В., Ширкин А.Е., Перминов И.К. Разработка элементов системы управления антропоморфным роботом AP-601. Издательство БФУ им. И. Канта, 2018. С. 368

8. Mahtani A., Sanchez L., Fernandez E., Martinez A. Effective Robotics Programming with ROS. Packt Publishing Ltd., 2016, p. 104

9. Перминов И.К. Разработка модели, имитирующей работу видеокамер и лидара для системы распознавания образов. Сборник статей XV Международной научно-практической конференции. В 2 ч. Ч. 1. С 11.