

УДК 004.421.4

## ЛОКАЛИЗАЦИЯ РОБОТА С ПОМОЩЬЮ ЛИДАРА

*Е. Б. Лаврентьев, В. Э. Кирпа*

Донской государственный технический университет (г. Ростов-на-Дону, Российская Федерация)

Описывается способ локализации мобильного робота в недетерминированном 2D-пространстве, основанной на использовании метода одновременной локализации и построения карты (SLAM).

**Ключевые слова:** распознавание, лидар, робот, алгоритм, обход препятствий, построение карты.

UDC 004.421.4

## ROBOT LOCALIZATION WITH LIDAR

*E. B. Lavrentyev, V. E. Kirpa*

Don State Technical University (Rostov-on-Don, Russian Federation)

The article describes a method for localization of a mobile robot in a non-deterministic 2D space based on the method of simultaneous localization and mapping (SLAM).

**Keywords:** recognition, lidar, robot, algorithm, obstacle avoidance, mapping.

**Введение.** Решение задачи локализации мобильного робота в недетерминированном 2D-пространстве основано на использовании метода одновременной локализации и построения карты (SLAM). Данная SLAM-система была создана в рамках подготовки к Всероссийской робототехнической олимпиаде. В качестве бортового компьютера была выбрана платформа *myRio 1900*, поскольку это было одним из условий соревнований.

**Постановка задачи.** Для создания SLAM-системы были использованы среда моделирования и разработки *Labview 2017* и лидар *Hokuyo URG-04LX-UG01*. Указанный лидар позволяет производить измерения со средней погрешностью около 3% относительно измеряемого диапазона. Также данный лидар имеет зону обнаружения  $240^\circ$  с угловым разрешением  $0,36^\circ$  и производит приблизительно 683 измерения за одно сканирование. Он выполняет одно сканирование каждые 100 мс с диапазоном сканирования от 0,02 до 4 м. На расстоянии 4 м диаметр луча составляет около 40 мм (рис. 1).

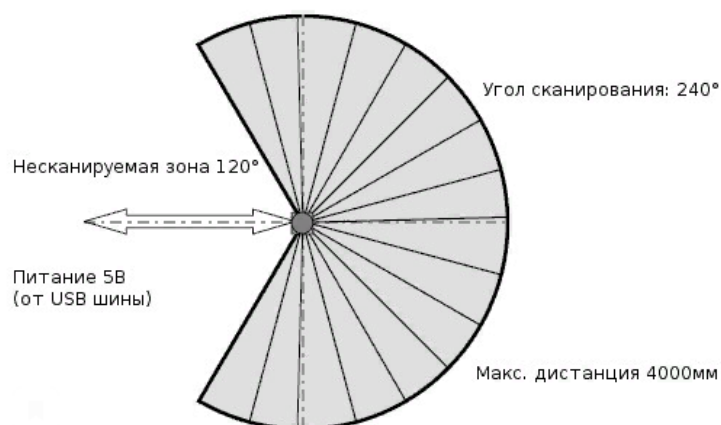


Рис. 1. Параметры лидара

**Реализация метода.** Как было сказано, для разработки SLAM была использована среда моделирования и разработки *Labview 2017*. Данная среда была выбрана потому, что в ней уже при-

сутствуют необходимые драйверы и библиотеки для получения массива точек. На рис. 2 приведен фрагмент структуры программы для извлечения данных с лидара. Эти данные представляют собой одномерный массив, состоящий из значений расстояний и соответствующих углов. Таким образом, данные передаются в полярной системе координат.

Первым шагом после передачи данных с помощью специального математического блока полученные данные переводятся из полярной системы в декартову систему координат.

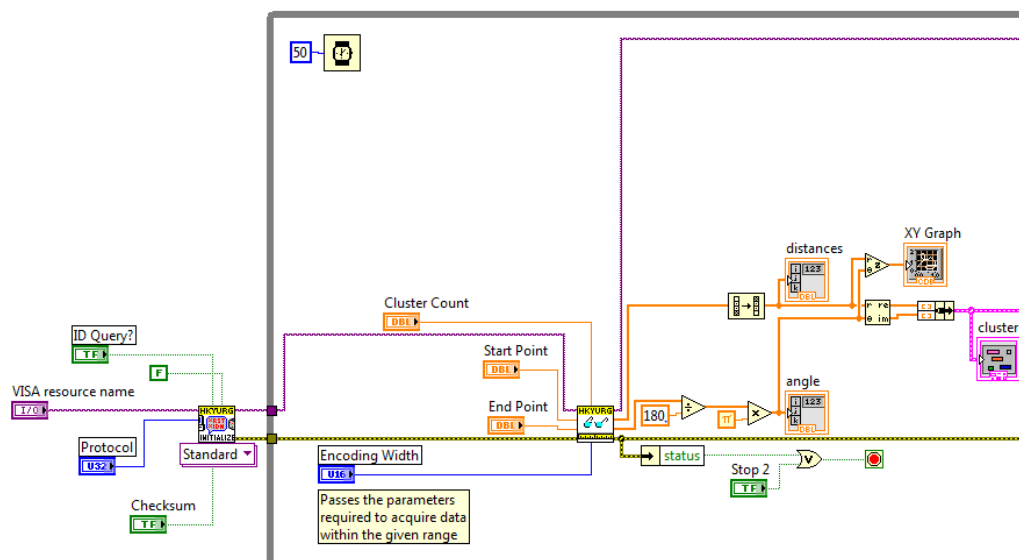


Рис. 2. Извлечение данных с лидара

Следующим шагом программы является реализация алгоритма обхода препятствий. Метод обхода препятствий заключался в том, что, если робот обнаруживал препятствие перед собой, то производилась проверка на наличие препятствия слева от него. Если препятствие слева присутствовало, робот отклонялся вправо (рис. 3).

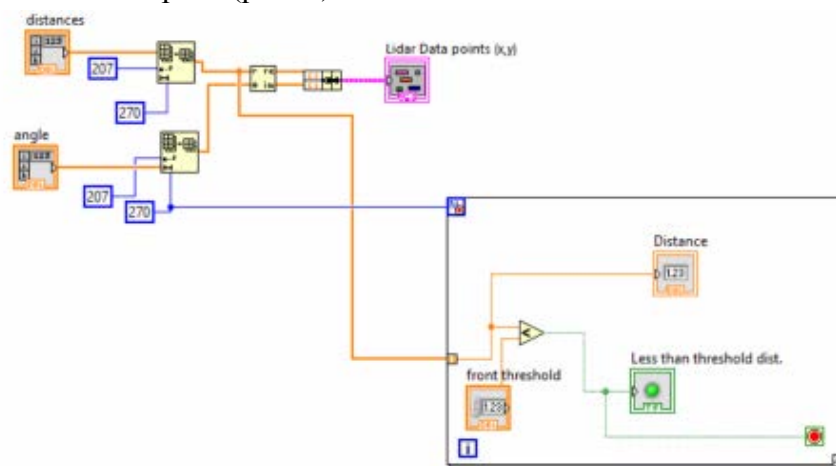


Рис. 3. Алгоритм объезда препятствий

Далее был реализован алгоритм движения вдоль стены. Он использовался для облегчения навигации робота по определенным участкам поля. Поскольку расположение препятствий на трассе не было определено, следование вдоль стены также позволило роботу избегать их, оставаясь в непосредственной близости от стен.

Основной алгоритм движения робота заключался в том, чтобы, двигаясь параллельно стене, оставаться на некотором расстоянии от нее. Стратегия алгоритма следования вдоль стены следу-

ющая: если расстояние до стены больше, чем расстояние  $d_{upper}$ , робот будет двигаться к стене, а если расстояние до стены будет меньше  $d_{lower}$ , робот будет отходить от стены (рис. 4–5).

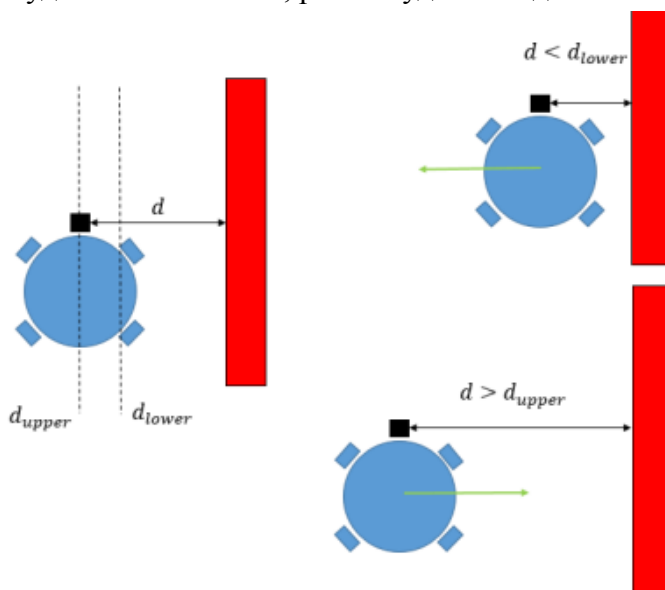


Рис. 4. Стратегия алгоритма следования робота вдоль стены

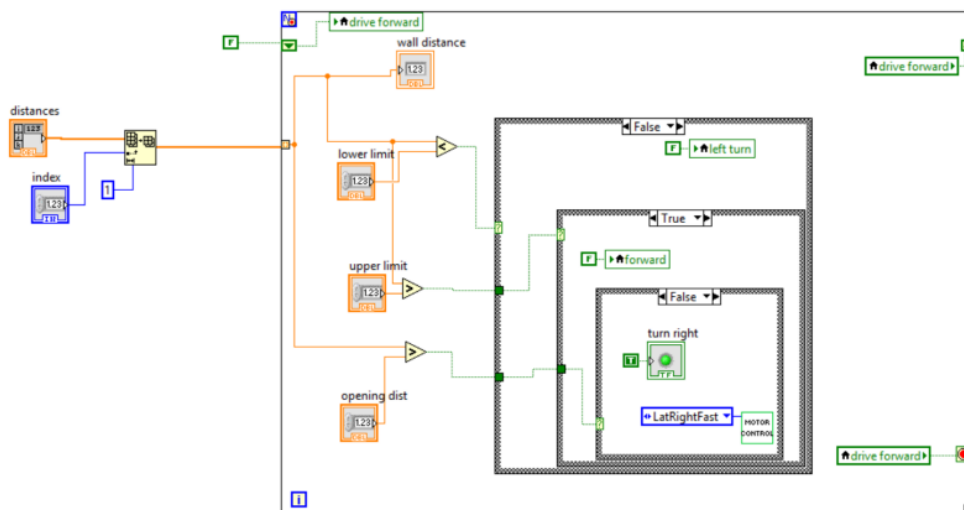


Рис. 5. Пример программной реализации алгоритма следования вдоль стены

Для использования алгоритма локализации возникла необходимость создания из полученного с лидара массива точек прямые линии. Для этого очень хорошо подходят методы «объединение» и «разделения» [1]. Так как робот передвигается по лабиринту с плоскими прямолинейными стенками, этот метод хорошо подходит для извлечения информации об отдельных статических препятствиях. В алгоритме использованы две основные функции: «Разделение» и «Объединение». Функция «Разделение» использует в своей работе рекурсивное разделение данных с лидара.

Базовое уравнение положения лидара (и соответственно, робота) определяется с использованием первого и последнего элементов массива данных с лидара. Затем вычисляется расстояние по нормали от каждой точки в массиве до базовой линии. Если расстояние превышает пороговое значение, данные разделяются на два отдельных массива по индексу точки с наибольшим расстоянием от линии. Затем этот процесс происходит рекурсивно, то есть самое дальнее расстояние от базовой линии меньше порогового значения.

Одной из трудностей, возникших при реализации функции разбиения, был расчет базового уравнения. Расстояние от точки  $(x_P, y_P)$  до прямой

$$Ax + By + C = 0$$

задано уравнением:

$$d = \frac{|Am + Bm + C|}{\sqrt{A^2 + B^2}}, \quad (1)$$

где прямая  $Ax + By + C = 0$ , имеет наклон  $-A/B$ . Наклон  $-A/B$  может быть легко найден путем вычисления наклона отрезка между первым и последним элементами данных с лидара. Тем не менее, появились трудности при определении коэффициента  $C$ . Его можно определить как

$$C = -Ax - By.$$

Однако, когда это было реализовано, программа зависла и плата myRIO отключилась.

В результате было предложено использовать другое уравнение, которое позволило рассчитать расстояние по нормали от точки  $(x_P, y_P)$  до линии, определяемой двумя точками  $(x_1, y_1)$  и  $(x_2, y_2)$ , что и показано в следующем уравнении:

$$d = \frac{|(y_2 - y_1)x_0 - (x_2 - x_1)y_0 + x_2y_1 - y_2x_1|}{\sqrt{(y_2 - y_1)^2 + (x_2 - x_1)^2}} \quad (2)$$

Это уравнение эквивалентно уравнению (1), однако отпадает необходимость расчета  $C$ , поскольку уравнение выражается в терминах известных точек, которые будут извлечены из лидара. Фрагмент алгоритма для реализации данной функции изображен на рис. 6.

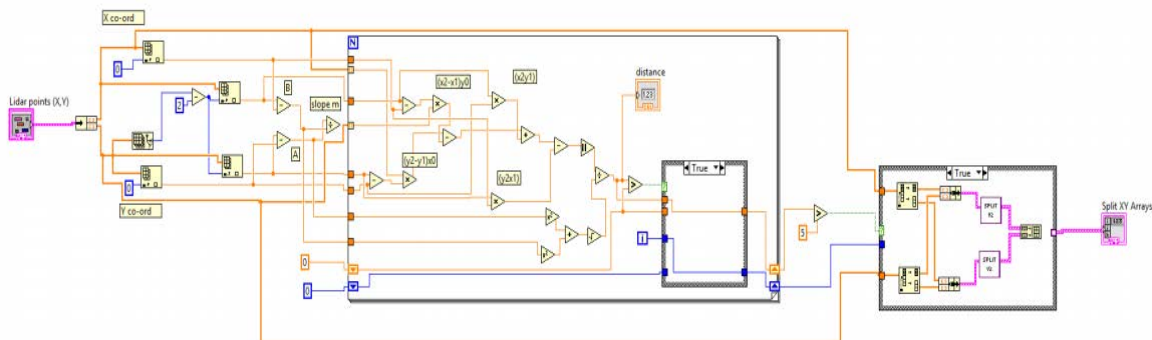


Рис. 6. Фрагмент алгоритма работы функции «Разделение»

Часть результирующих «линий» алгоритма «Объединение» проходит через набор разделенных массивов (где каждый массив представляет собой «линию», определенную серией точек) и определяет, какие массивы можно объединить. Это достигается путем нахождения базового уравнения между первыми двумя строками. Расстояние по нормали от точки пересечения двух линий до базовой линии сравнивается с пороговым значением. Если расстояние меньше порогового значения, две линии объединяются, иначе они остаются как две отдельные линии. Пример алгоритма работы функции «Объединение» показан на рис. 7.

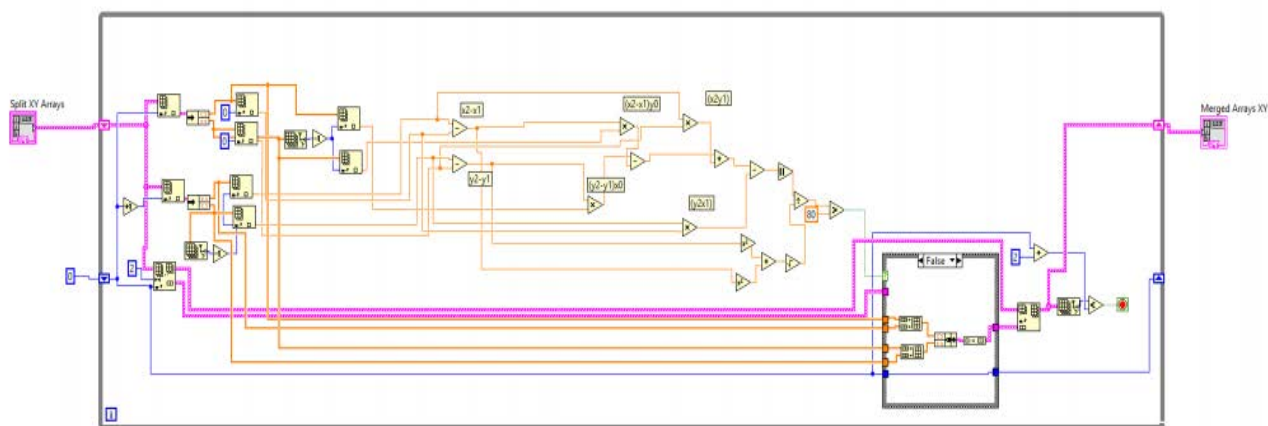


Рис. 7. Алгоритм функции «Объединение»

На рис. 8.1 приведен график исходного сканирования лидара, а на рис. 8.2 — результат работы функций «Разделение» и «Объединение».

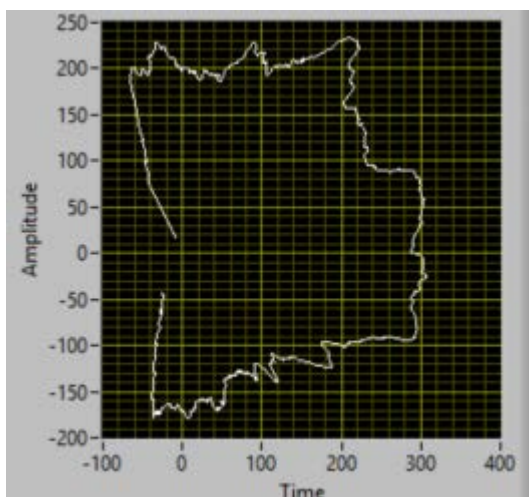


Рис. 8.1

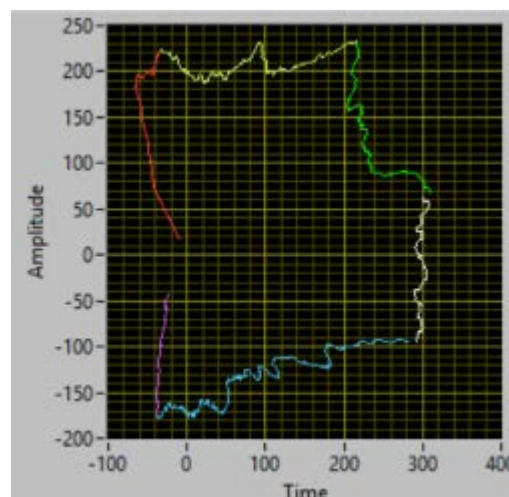


Рис. 8.2

**Заключение.** Таким образом, данная SLAM-система показала хорошие результаты в недетерминированной среде. Робот успешно преодолевал статические препятствия и достаточно точно определял свои координаты, не используя при этом одометрию. Однако данная SLAM-система не подходит для среды с динамическими препятствиями. В настоящее время ведутся разработки в этом направлении.

### Библиографический список

1. Описание методов преобразования [Электронный ресурс] / ArcGIS Desktop. — Режим доступа: <https://desktop.arcgis.com/ru/arcmap/latest/manage-data/las-dataset/lidar-point-classification.htm> (дата обращения: 22.04.2019).

*Об авторах:*

**Лаврентьев Евгений Борисович**, старший преподаватель Донского государственного технического университета (344000, РФ, г. Ростов-на-Дону, пл. Гагарина, 1), [leb210570@gmail.com](mailto:leb210570@gmail.com)

**Кирпа Виталий Эдуардович**, студент Донского государственного технического университета (344000, РФ, г. Ростов-на-Дону, пл. Гагарина, 1), [kirpavitali@yandex.ru](mailto:kirpavitali@yandex.ru)