

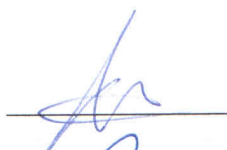
**МИНОБРНАУКИ РОССИИ**  
**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**  
**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**  
**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**  
**Кафедра МО ЭВМ**

**ОТЧЕТ**  
**по научно-исследовательской деятельности в первом семестре**

Аспирант

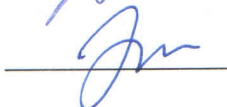
направление 09.06.01

направленность 05.13.11



Филатов Антон Юрьевич

Руководитель д.т.н., проф



Экало Александр Владимирович

Санкт-Петербург

2016

## **Цель работы**

Изучить различия в существующих алгоритмах SLAM, а также сравнить некоторые популярные решения с целью выяснить, какие из них наиболее подходят для мобильных роботов, работающих в помещениях.

## **Введение**

Задача одновременного построения карты и определения на ней собственного местоположения (англ. SLAM — Simultaneous Localization And Mapping) — одна из ключевых задач в мобильной робототехнике. Любые роботы, вынужденные двигаться автономно, в том или ином виде решают эту задачу. Задача SLAM ставится перед марсоходами, подводными лодками, даже роботами-пылесосами. В научных целях используются мобильные роботы, оснащённые различными датчиками.

## **Описание проделанной работы**

Самым простым роботом, который может решать задачу SLAM, является робот, оснащённый лазерным дальномером. Это самый простой датчик. Его работа заключается в том, чтобы с некоторой частотой (обычно она измеряется в десятках Герц) вычислять расстояние до всех окружающих робота объектов. Чтобы не проводить предобработку этих данных, необходимо лишь на полученном скане (который снимается на 360 градусов) обозначить позицию, с которой проводилось наблюдение. Таких данных достаточно, чтобы в самом грубом приближении решить задачу SLAM.

Кроме лазерного скана можно использовать видеокамеры, даже трёхмерные. Такой алгоритм будет выполняться дольше и будет требовать больших ресурсов, но он будет заведомо точнее. Лазерный дальномер и видео камеру обычно называют сенсорами робота. Помимо сенсоров обычно используется датчик одометрии. Он вычисляет перемещение робота.

Сложность в использовании датчика одометрии в том, что на нём накапливается погрешность, которую невозможно устранить аппаратно. Дело в том, что он вычисляет количество оборотов колёс или чего-то подобного, но не проверяет, не произошло ли некоторого проскальзывания колёс или является ли сцепление робота с поверхностью хорошим. Поэтому данные одометрии являются точными лишь непродолжительный отрезок времени. Далее их необходимо корректировать.

Вернёмся к алгоритмам решения задачи SLAM. Работу такого алгоритма можно описать так:

- 1) На каждом шаге алгоритма робот получает новые данные сенсора и одометрии
- 2) Полученные данные необходимо сопоставить с уже имеющейся картой местности (на первом шаге карты нет, поэтому весь полученный скан считается первой версией карты)

3) Найдя наиболее подходящее соответствие между новым сканом и построенной картой необходимо скорректировать одометрию, чтобы получить положение робота на текущем шаге

4) Используя данные скана необходимо обновить или дополнить текущее представление о карте

Наиболее ёмким и сложным шагом описанного алгоритма является процесс определения соответствия между сканом и картой. За этот процесс отвечает узел, называемый скан матчером (scan matcher). В литературе описывается несколько возможных подходов к реализации этого процесса. Наиболее простой алгоритм скан матчинга состоит в следующем.

Допустим, пришедший скан — это результат работы лазерного дальномера. Тогда скан — это совокупность точек, обозначающих пересечение лазера с препятствием, или, проще говоря, положение препятствий на карте. Если довериться одометрии, то можно будет «наложить» такой скан на карту, используя данные одометрии, как априорную оценку положения робота. Но поскольку одометрия может иметь погрешность, то такое наложение может быть неточным. Это можно увидеть невооружённым взглядом: линия стены на скане может не совпадать с реальной стеной на карте, или какое-то препятствие на скане может быть сдвинуто относительно его положения.

Для решения такого конфликта и нужен скан матчер. Его задача «подвигать» пришедший скан так, чтобы наибольшее количество препятствий на скане и на карте совпали. Тогда можно будет обновить знание о собственном положении, а также скорректировать карту, если она была недостроена или содержала неточности.

Есть несколько подходов к реализации процесса скан матчинга. Самый простой в реализации — это стохастический. Он строит предположения о положении робота в некоторой области вокруг априорной оценки положения. Если он находит положение, в котором скан накладывается на карту «лучше», чем было до того, то такое положение робота считается более точным, и процесс повторяется ещё раз до тех пор, пока не будет найдена оптимальная

позиция или не будет превышен некоторый порог попыток поиска. Такой скан матчер получил название скан матчером Монте-Карло.

Другой подход, который является более простым в плане идеи, но гораздо более требовательным к вычислительным ресурсам — это полный перебор. В некоторой области вокруг априорной оценки позиции строятся гипотезы о положении робота и совершается полный перебор этих гипотез. Для каждой гипотезы вычисляется оценка соответствия скана карте. Затем находится гипотеза с максимальной оценкой. Недостатком такого подхода является неопределённость в выборе шага дискретизации, а также большая стоимость временных и вычислительных ресурсов, что критично для мобильных роботов, работающих в режиме реального времени.

Ещё один подход основан на применении преобразования Хафа к скану и карте. На скане и карте выделяются прямые линии и строится матрица ковариации между сканом и картой. Затем начинается «варщение» и перемещение скана с целью максимизировать ковариацию. Такой метод даёт очень точную оценку перемещения, но очень чувствителен к вращению априорной оценки положения робота, поскольку частота дискретизации угла вращения является заранее неизвестной величиной.

Было проведено сравнение трёх описанных выше алгоритмов скан матчинга из которого можно сделать вывод, что наиболее подходящим алгоритмом для мобильных роботов является стохастический метод, поскольку можно ограничить время его работы, а также поскольку он не требует большого количества вычислительных ресурсов. Такой подход работает в условиях с низкой фрактальностью карты, то есть когда при случайном перемещении скана мы не рискуем попасть на область, повторяющую пришедший скан, но лежащую в отдалении от него.

## **Заключение**

В ходе научно-исследовательской работы была изучена задача SLAM, а также было проведено сравнение нескольких популярных подходов к реализации процесса скан матчинга — основного компонента различия SLAM-ов. В результате было получено, что наиболее подходящим для мобильных роботов алгоритмом скан матчера является алгоритм Монте-Карло. Алгоритм, реализующий такой скан матчер называется `tinySLAM`, его работа была протестирована на записанных последовательностях входных данных, на которых было подтверждено, что этот алгоритм выполняется точно.

## Список литературы

1. Past, present, and future of simultaneous localization and mapping: Toward the robust perception age / C. Cadena, L. Carlone, H. Carrillo, et al. // IEEE Transactions on Robotics. - 2016. - Vol. 32. - P. 1309–1332.
2. A comparison of slam algorithms based on a graph of relations / W. Burgard, C. Stachniss, G. Grisetti, et al. // 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems. - 2009. - Vol 41. - P. 2089–2095.
3. B. Gerkey, Ros slam gmapping [Electronic resource] / B. Gerkey // <http://wiki.ros.org/slam-gmapping>. [Accessed 15-Jan-2018].
4. Google, 2d cartographer backpack deutsches museum / Google // <https://github.com/googlecartographer>
5. The mit stata center dataset. / M. Fallon, H. Johannsson, M. Kaess, J. J. Leonard. // The International Journal of Robotics Research. - 2013. - Vol 32. - P. 1695–1699.