

# Планирование движения автономного робота в лабиринте с препятствиями

Э. И. Залилов<sup>1</sup>, А. С. Долгий<sup>2</sup>, А. В. Шокуров<sup>3 4</sup>

В рамках данной работы решаем задачу планирования движения автономного робота, применяя и сравнивая классический подход решения этой задачи и подход с применением машинного обучения. Была поставлена задача “проехать из точки А в точку В” на различных экспериментальных картах, представляющих собой лабиринт с препятствиями. Мы изучали поведение двухколесного робота с разными методами построения маршрутов и разными наборами сенсоров. Предполагается, что подход с применением машинного обучения более легкий для разработки и нуждается в меньшем количестве сенсоров, что существенно снижает стоимость такого робота.

**Ключевые слова:** планирование движения; робототехника; примитивы движения; обучение с подкреплением

## 1. Классический беспилотный подход

Главная возможность автономного робота - это умение добраться из текущей точки в любую другую выбранную точку на карте, преодолевая возникающие препятствия. Классический подход планирования движения автономного робота предполагает наличие нескольких этапов: локализация, распознавание, планирование маршрута и управление.

---

<sup>1</sup> *Залилов Эмиль Ильгизович* — аспирант каф. теоретической информатики мех.-мат. ф-та МГУ, e-mail: zalilov\_emil@mail.ru

Emil Ilgizovich Zalilov — graduate student, Lomonosov Moscow State University, Faculty of Mechanics and Mathematics, Chair of Theoretical Informatics.

<sup>2</sup> *Долгий Алексей Сергеевич* — аспирант каф. теоретической информатики мех.-мат. ф-та МГУ, e-mail: dolgy62@mail.ru

Dolgiy Aleksey Sergeevich — graduate student, Lomonosov Moscow State University, Faculty of Mechanics and Mathematics, Chair of Theoretical Informatics.

<sup>3</sup> *Шокуров Антон Вячеславович* — к.ф.-м.н., научный сотрудник лаборатории вычислительных методов механико-математического факультета МГУ им. М.В.Ломоносова, e-mail: shokurov.anton.v@yandex.ru

Shokurov Anton Vyacheslavovich — Ph.D., research associate of Computational Methods Laboratory of Mechanics and Mathematics Faculty of Lomonosov Moscow State University

<sup>4</sup> Исследование выполнено при поддержке Междисциплинарной научно-образовательной школы Московского университета «Мозг, когнитивные системы, искусственный интеллект»

This research has been supported by the Interdisciplinary Scientific and Educational School of Moscow University «Brain, Cognitive Systems, Artificial Intelligence»

На этапе локализации роботу необходимо понимать, где он находится в данный момент времени. В зависимости от поставленной задачи и имеющихся сенсоров для локализации могут быть применены различные методы, такие как GNSS, Одометрия, EKF, ICP, часть из которых описано в [3].

На этапе распознавания роботу необходимо распознать значимые окружающие его объекты. Для того, чтобы робот мог успешно обойти все препятствия, ему необходимо уметь отличить возникающие препятствия и классифицировать их. В результате должно быть получено виртуальное представление внешней среды.

Робот должен спланировать маршрут. При планировании маршрута так же существует несколько разных подходов каждый из которых может быть использован в зависимости от поставленной задачи и имеющихся ограничений. Для построения наиболее оптимального пути, при этом реагируя на все возникающие препятствия, используется двухуровневая архитектура планирования пути. Глобальный планировщик работает в паре с локальным планировщиком, дополняя друг друга. Глобальный планировщик занимается долгосрочным планированием пути, используя доступную информацию о местности и не учитывая неожиданные препятствия. Алгоритм построения пути [6]:

- Дискретизируем пространство и выбираем примитивы движения (рис. 1)
- Строим граф (рис. 1)
- Ищем наиболее короткие пути: Алгоритм Дейкстры,  $A^*$  (используется чаще остальных, более подробно в [6, 2]), Rapidly exploring random tree(RRT)

Робот должен как то автономно управляться. На этом этапе выполняется заданная планировщиком траектория.

Для решения поставленной в этой работе задачи наш робот должен уметь построить карту местности. Для этого необходимо проехать по разным маршрутам местности, используя один из доступных алгоритмов построения карт. Опять таки методов достаточно много, они отличаются точностью и применяемыми сенсорами. Есть варианты без визуальной одометрии, использующие только лидары. Однако, они будут уступать в точности алгоритмам визуальной одометрии, которая позволяет оценить перемещение робота, его текущую позицию и ускорение на основе данных видеопотока с камеры.

В зависимости от сложности местности, по которой придется передвигаться роботу, возможно использование разных наборов senso-

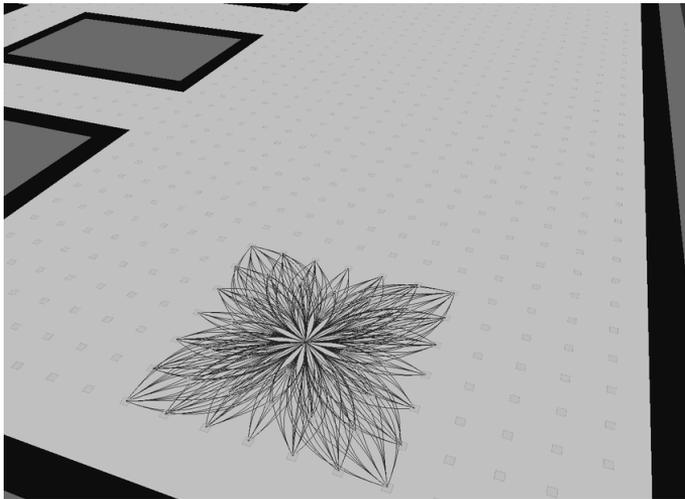


Рис. 1. Дискретизированное пространство и построенный на нем граф примитивов движения

ров(лидары, камеры, радары). Однако, каждый из наборов будет требовать адаптации и поддержки каждого из модулей планирования движения, что сильно усложняет разработку такой системы.

## 2. Использование обучения с подкреплением

Для обучения с подкреплением (Reinforcement Learning)[4, 7] необходимо построить модель, содержащую агента (в нашем случае - робота), среду (лабиринт, см. Рис. 2), а также множества действий доступных агенту и множество состояний среды (см. Рис 3.).

Кроме того необходимо знать функции перехода из одного состояния в другое при конкретном действии, но в нашем случае этим занимается сам симулятор. Робот на каждом этапе выбирает и совершает действие из набора доступных действий, получает за это награду (возможно отрицательную), и среда переходит в новое состояние.

На первом этапе необходимо определиться с вариантами структуры модели обучения с подкреплением, в нашем случае, в качестве действий можно, например, взять два действия: движение робота вперед и вращение. А также давать награды за движение, а штрафовать за столкновение со стенами. Однако на практике, этот вариант себя показал плохо, поскольку почти всегда робот очень быстро учится просто крутиться на месте. В качестве “состояний среды” - используются дискретизированные данные с лидара. Обучение практиковалось по разным алгоритмам:

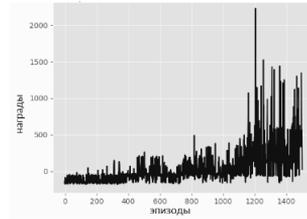
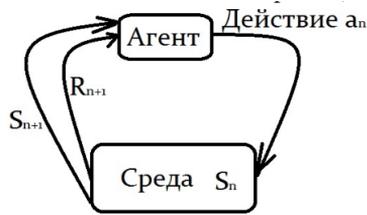
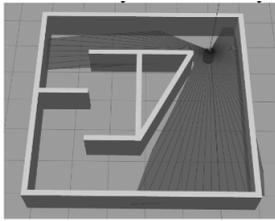


Рис. 2. На левом рисунке изображение робота в лабиринте. Линиями изображен лидар.

Рис 3. На правом рисунке традиционная схема обучения с подкреплением.

Рис 4. График возрастания наград по эпизодам. Высокие отметки с наградой порядка 1200 - соответствуют нескольким полным кругам по лабиринту без столкновений и завершению эпизода по таймауту.

Q-learning, SARSA, и DQN [7, 1, 5], а также на картах разного уровня сложности. Обучался робот в симуляторе Gazebo с использованием ROS.

Под своим состоянием робот понимает набор из  $N$  чисел, которые получаются путем выбора из лидара  $N$  равномерно расставленных лучей и округление их длин. Роботу предстоит основываясь на своем состоянии выбрать одно из 3 действий за каждое из которых своя награда: линейное движение прямо (6 очков), движение вокруг своей оси с небольшой линейной скоростью влево/вправо (1 очко). Такой подход позволил вынудить робота больше двигаться вперед, преодолевать зигзаг, и проходить лабиринт, а не кружиться на месте.

В приведенных методах используется обучение основанное на поиске функции оценки полезности действия через уравнение Беллмана [7].

На Рис. 4 изображены награды по эпизодам. Из графика видно, например, что в первые 200 эпизодов награда почти не становилась положительной, действия робота на тот момент были почти полностью хаотичны, и он очень быстро врезался в стену получая большой штраф (-200 очков).

При обучении использовалась  $\epsilon$ -жадная стратегия. Это значит, что вначале робот действует абсолютно случайным образом, чтобы изучить лабиринт, а далее с каждым эпизодом шанс на случайное действие ( $\epsilon$ ) уменьшается. На 400-600 эпизодах робот уже стал проезжать половину лабиринта, и получать 0-250 очков. А после 1200 стал часто выходить по таймауту, проезжая весь лабиринт.

### 3. Заключение.

Мы изучили лишь несколько вариантов карт, на которых подход с использованием обучения с подкреплением показал себя не хуже классического при более легкой разработке с последующим поддержанием кода и с меньшим набором внешних сенсоров. Однако, этого недостаточно, чтобы утверждать, что мы можем использовать его для широкого класса задач. Необходимо исследовать большее количество карт с разными ограничениями и подобрать наиболее оптимальный подход к решению задачи.

### Список литературы

- [1] Choudhary A., “A Hands-On Introduction to Deep Q-Learning using OpenAI Gym in Python”, 2019.
- [2] Dolgov D., Thrun S., Montemerlo M., Diebel J., “Practical Search Techniques in Path Planning for Autonomous Driving”, *AAAI Workshop - Technical Report*, 2008.
- [3] KumarPanigrahi P., KishoroBisoy S., “Localization strategies for autonomous mobile robots: A review”, *Journal of King Saud University - Computer and Information Sciences*, 2021.
- [4] Sutton R., Barto A., “Reinforcement Learning: An Introduction”, *The MIT Press*, 1998.
- [5] “Reinforcement Learning: Q-Learning vs Sarsa”, <http://www.cse.unsw.edu.au/cs9417ml/RL1/algorithms.html>.
- [6] Ryan De Iaco, Smith S., Czarnecki K., “Learning a Lattice Planner Control Set for Autonomous Vehicles”, 2019, 549–556.
- [7] К.В.Воронцов, “Обучение с подкреплением”, [machinelearning.ru](http://machinelearning.ru).

### **Path planning of an autonomous robot in a maze with obstacles Zalilov E.I., Dolgiy A.S., Shokurov A.V.**

We solve the problem of autonomous robot path planning by applying and comparing classical approach and approach using machine learning algorithms. The task was set to "move from point A to point B" on various experimental maps representing a maze with obstacles. We studied the behavior of a two-wheeled robot with different methods of building routes and different sets of sensors. It is assumed that the machine learning approach is easier to develop and requires fewer sensors, so this significantly reduces the cost of such a robot.

**Keywords:** path planning; robotics; motion primitives; reinforcement learning; rl

## References

- [1] Choudhary A., “A Hands-On Introduction to Deep Q-Learning using OpenAI Gym in Python”, 2019.
- [2] Dolgov D., Thrun S., Montemerlo M., Diebel J., “Practical Search Techniques in Path Planning for Autonomous Driving”, *AAAI Workshop - Technical Report*, 2008.
- [3] KumarPanigrahi P., KishoroBisoy S., “Localization strategies for autonomous mobile robots: A review”, *Journal of King Saud University - Computer and Information Sciences*, 2021.
- [4] Sutton R., Barto A., “Reinforcement Learning: An Introduction”, *The MIT Press*, 1998.
- [5] “Reinforcement Learning: Q-Learning vs Sarsa”, <http://www.cse.unsw.edu.au/cs9417ml/RL1/algorithms.html>.
- [6] Ryan De Iaco, Smith S., Czarnecki K., “Learning a Lattice Planner Control Set for Autonomous Vehicles”, 2019, 549–556.
- [7] K.V.Vorontsov, “Reinforcement training” (In Russian), [machinelearning.ru](http://machinelearning.ru).