

СИДОРЕНКО А.В., СОЛОДУХО Н.А.

АЛГОРИТМ ОГИБАНИЯ ПРЕПЯТСТВИЙ ПРИ ПЕРЕМЕЩЕНИИ МОБИЛЬНОГО РОБОТА С ИСПОЛЬЗОВАНИЕМ Q-ОБУЧЕНИЯ И ТЕХНОЛОГИИ БЛОКЧЕЙН

Белорусский государственный университет
г. Минск, Республика Беларусь

Аннотация. Предложен алгоритм моделирования движения робота с огибанием препятствий при использовании метода машинного Q-обучения. Применение Q-обучения позволяет сохранять полученные при моделировании вознаграждения, выполняя оптимальные действия в каждом конкретном состоянии, при этом в Q-таблице содержится информация о состоянии и действии робота. Сохранение Q-таблицы в блокчейне с использованием технологии IPFS (InterPlanetary File System) позволяет надежно и децентрализованно хранить данные о состояниях и действиях робота. Контент адресации в IPFS отделяет данные от их местоположения и извлекает файлы из нескольких источников в пириновом режиме. Для предложенного алгоритма проведен вычислительный эксперимент с применением среды моделирования движения робота. В пакете визуализации Gazebo 11 показано, что с использованием нового алгоритма препятствия огибаются быстрее (на 59,8 %), чем при использовании предыдущей версии алгоритма.

Ключевые слова: робот, блокчейн, IPFS, Q-обучение, перемещение, препятствия, ROS 2, Gazebo 11

Введение

Интенсивное развитие информационных методов искусственного интеллекта, технологии блокчейн, а также использование для обеспечения связи с наземными объектами структур InterPlanetary File System (IPFS) выдвигают новые задачи в области моделирования и разработки алгоритмов огибания препятствий при перемещении роботизированных систем.

В представленных нами работах [1, 2] были предложены алгоритмы огибания препятствий при перемещении робота, в которых решение задачи проводилось следующим образом:

- 1) когда робот встречает препятствие, он начинает огибать его, двигаясь вдоль препятствия, постоянно держась одной и той же стороной к препятствию (алгоритм I, он же bug-алгоритм);
- 2) использование модифицированного алгоритма (далее – алгоритм II), сочетающего алгоритм I и алгоритм vffh, что позволило не только находить оптимальные маршруты, но и эффективно справляться с длинными и сложными препятствиями, встречающимися на его пути.

Данная статья является продолжением работы, представленной в статье [2], и определяет параметры системы управления роботом, позволяющим реализовать возможности использования машинного обучения, технологии блокчейн и сохранения информационных данных в IPFS (далее – алгоритм III).

Вначале остановимся на особенностях машинного Q-обучения и системы IPFS.

Алгоритм Q-обучения

Используя данный алгоритм и полученное от среды вознаграждение, агент (далее – робот) создаёт функцию полезности Q . Это позволяет ему случайным образом выбирать стратегию поведения, учитывая при этом опыт предыдущих взаимодействий со средой.

Алгоритм Q-обучения анализирует пару "состояние-действие". Это позволяет вычислить ценность действия в конкретном состоянии. В любом процессе принятия решений с использованием Q-обучения определяется оптимальное значение Q .

Алгоритм Q-обучения позволяет агенту получать вознаграждение, выполняя наиболее оптимальные действия в каждом конкретном состоянии. Исходя из таблицы вознаграждений, алгоритм позволяет выбрать следующее действие на основе его полезности и дает возможность агенту обновлять Q -значение. Q -значения инициализируются случайными значениями и затем обновляются согласно выражению:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r_t + 1 + \gamma \max_a Q] \quad (1)$$

где a_t и s_t – действие и состояние агента в момент времени t , α и γ – скорость обучения и дисконтирующий множитель, параметры которых находятся в диапазоне $[0, 1]$, r – значение вознаграждения [3].

Таким образом, создается новая таблица, называемая Q -таблицей, которая содержит информацию о состояниях и действиях агента. При значительном увеличении числа состояний и действий хранение информации в Q -таблице может стать проблематичным и привести к сбоям.

IPFS (Межпланетная файловая система)

Межпланетная файловая система (InterPlanetary File System, IPFS), представляет собой децентрализованную сеть для хранения и обмена данными с использованием технологии блокчейн. IPFS работает по принципу пиринговой сети, позволяя пользователям обмениваться и сохранять данные без центрального сервера. Когда файл загружается в IPFS, он разбивается на малые части, каждая из которых получает свой уникальный идентификатор (так называемый CID, полученный на основе хеша). Эти части хранятся по всей сети, и, когда кто-то хочет получить доступ к этому файлу, он собирается снова из этих частей с использованием хешей. Блокчейн добавляет неизменяемый журнал транзакций в IPFS. Любое изменение данных фиксируется в блокчейне, что обеспечивает прозрачность и проверяемость.

В отличие от HTTP, который загружает файлы с одного сервера за раз, контент-адресация в IPFS отделяет данные от их местоположения. IPFS извлекает файлы из нескольких источников одновременно в пиринговом режиме, что позволяет решать следующие задачи.

1. *Делать цифровой отпечаток данных.* Идентификаторы контента (CID) дают каждому элементу контента, данным или файлу постоянный хеш-адрес, который можно закрепить (pin), чтобы сохранить. Для закрепления можно использовать как сайт <https://ipfs.tech/>, так и другие специально предназначенные для этого сайты, например: <https://pinata.cloud/>, <https://www.infura.io/>, <https://www.4everland.org/>, <https://fleek.co/>.

2. *Свободно хранить и распространять данные.* Пиринговая сеть IPFS объединяет тысячи узлов. Таким образом, можно найти все узлы с необходимыми данными и помочь другим обнаружить контент, который есть у вас.

3. *Оптимизировать доставку данных.* IPFS кеширует и хранит данные на локальных серверах, экономя пропускную способность благодаря более эффективному распределению и доставке. Если одни и те же данные расположены в разных местах сети, и какому-то узлу необходимо получить доступ к этим данным, в качестве источника информации будет использован тот узел, который расположен ближе к данному узлу.

Проведение вычислительного эксперимента

Программно реализованные алгоритмы обучения, примененные к разработанной нами модели управления системой мобильных роботов, позволили провести вычислительный эксперимент с использованием разработанной компьютерной программы. При работе с симуляцией движения робота исполь-

зовался визуализатор Gazebo 11 и ROS 2 Humble Hawksbill. Для управления роботом использовались команды командной оболочки UNIX bash в ОС Linux Ubuntu 22.04 Jammy Jellyfish. Исходный код представлен в ветке “with-blockchain-json” [4]. Для работы алгоритма необходимо запустить скрипт `bash lines.sh`. Для задания конечной точки движения необходимо задать координаты X (переменная “targetX”) и Y (переменная “targetY”) в строках 3-16 файла `lines.sh`.

При проведении вычислительного эксперимента в качестве среды использовалась поверхность 20 на 20 клеток с препятствием в виде длинной стены (стартовая точка в координатах (-3;5), конечная точка в координатах (3;3)). В эксперименте использовались следующие параметры, необходимые для корректной работы Q -обучения: максимальная награда, при которой прекращается обучение, равнялась 100; максимальное количество шагов в эпизоде, по достижению которых эпизод обучения начинался заново, – 50. В координатах цели награда r за любое движение для достижения цели составляла 60. Мгновенная награда за достижение цели r составляла 100. Мгновенная награда r без достижения цели составляла -10. Значение α равнялось 0,1. Значение γ равнялось 0,9.

Структурная схема предложенного алгоритма представлена на рисунке 1.

Образец Q -таблицы в формате json в используемой программе приведен на рисунке 2.

Симуляционное пространство в пакете визуализации Gazebo 11 приведено на рисунке 3.

Предложенный алгоритм для огибания препятствий роботизированной системой состоит в следующем. В работе использовано Q -обучения с вариантами движения робота (действиями): 1) движение по алгоритму VFH* на расстояние предела видимости до начала движения; 2) движение по bug-алгоритму до тех пор, пока впереди не встретится препятствие. Состояние робота задавалось целыми значениями координатной сетки. Q -таблица представляет собой json-файл, в котором прописана награда за выбранное действие в выбранной ячейке координатной сетки. Начальные значения Q -таблицы задавались следующим образом: движение по алгоритму vfh: 1.1; движение по алгоритму bug левой стороной: 1.00001; движение по алгоритму bug правой стороной: случайное число от 0 до 1. Данные значения выбраны для ускорения обучения. Q -таблица хранилась в виде json-файла и после обновления автоматически сохранялась в блокчейн с использованием IPFS [5]. В качестве сайта для закрепления данных по технологии IPFS мы использовали <https://www.4everland.org/>. Блок-схема предложенного алгоритма представлена на рисунке 1. Образец элемента Q -таблицы представлен на рисунке 2.

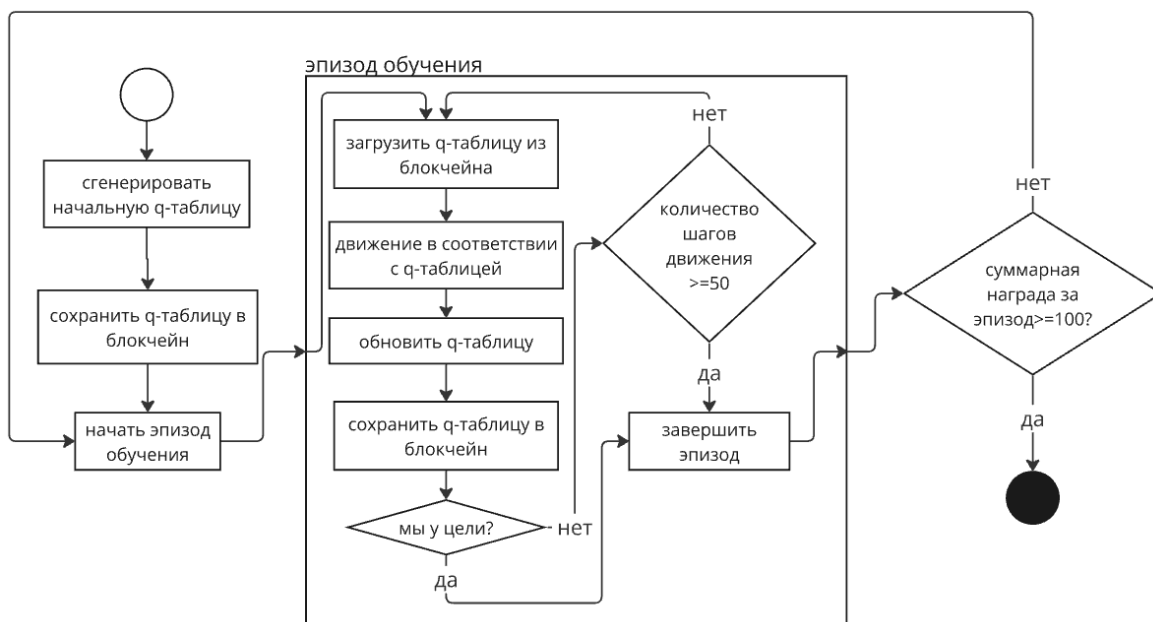


Рисунок 1. Структурная схема предложенного алгоритма

JSON	Необработанные данные	Заголовки
Сохранить	Копировать	Свернуть все
Развернуть в		
▼ [-10.0, -9.0]:		
▼ [-10.0, -9.0]:		
vfh:	1.1	
bug_left:	1.00001	
bug_right:	0.09728686372827333	
► [-9.0, -8.0]:	{...}	
► [-8.0, -7.0]:	{...}	
► [-7.0, -6.0]:	{...}	
► [-6.0, -5.0]:	{...}	
► [-5.0, -4.0]:	{...}	
► [-4.0, -3.0]:	{...}	
► [-3.0, -2.0]:	{...}	
► [-2.0, -1.0]:	{...}	
► [-1.0, 0.0]:	{...}	
► [0.0, 1.0]:	{...}	
► [1.0, 2.0]:	{...}	
► [2.0, 3.0]:	{...}	
► [3.0, 4.0]:	{...}	
► [4.0, 5.0]:	{...}	
► [5.0, 6.0]:	{...}	
► [6.0, 7.0]:	{...}	
► [7.0, 8.0]:	{...}	
► [8.0, 9.0]:	{...}	
► [9.0, 10.0]:	{...}	
▼ [-9.0, -8.0]:		
▼ [-10.0, -9.0]:		
vfh:	1.1	
bug_left:	1.00001	
bug_right:	0.22310654314829692	
▼ [-9.0, -8.0]:		
vfh:	1.1	
bug_left:	1.00001	
bug_right:	0.7321155557545258	
▼ [-8.0, -7.0]:		
vfh:	1.1	
bug_left:	1.00001	
bug_right:	0.3424112180046296	
▼ [-7.0, -6.0]:		
vfh:	1.1	
bug_left:	1.00001	
bug_right:	0.34414906006681645	
▼ [-6.0, -5.0]:		
vfh:	1.1	

Рисунок 2. Образец Q-таблицы в формате json в используемой программе

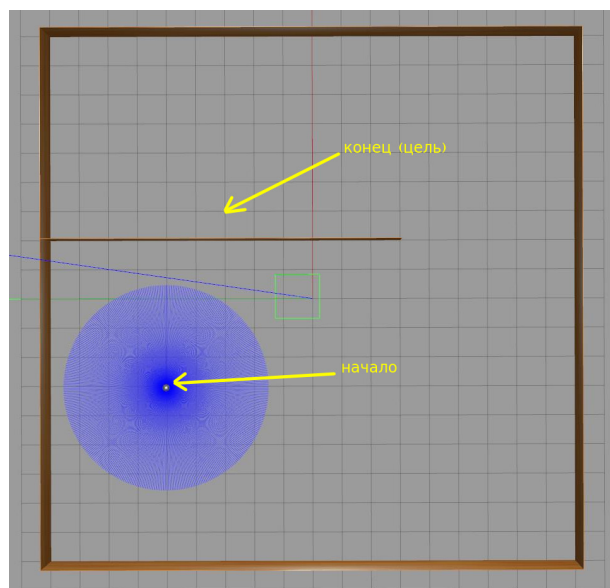


Рисунок 3. Симуляционное пространство в пакете визуализации Gazebo 11

Мы предварительно обучаем нашу систему до тех пор, пока награда за эпизод не станет равной или выше 100. Как показал эксперимент, после этого робот уверенно находит цель, оптимально огибая препятствия.

Анализ результатов вычислительного эксперимента

Результаты проведенного вычислительного эксперимента позволили представить кривую изменения суммарной награды за эпизод в течение обучения (рисунок 4) в процессе вычислительного эксперимента. Необходимо отметить, что несмотря на все прилагаемые усилия по программированию робота, иногда он все же сталкивался с препятствием, т.к. окружение Gazebo 11 моделирует реальное окружение, а в реальном окружении есть задержка прослушивания информации, получаемой от сенсоров, – роботу нужно время, чтобы получить и обработать информацию, среагировав на изменения в окружении в процессе движения. В случае соударения робота с препятствием мы просто перезапускаем окружение и начинаем эпизод обучения заново. Этим обуславливаются некоторые внезапные резкие снижения суммарной награды – робот просто столкнулся с препятствием. После обучения, для оценки времени достижения цели с огибанием препятствий мы запускаем очередной эпизод обучения и регистрируем время, которое понадобилось роботу для достижения цели.

Временные параметры движения роботизированной системы при применении различных разработанных нами алгоритмов (алгоритм I из работы [1], алгоритм II из работы [2], алгоритм III – предлагаемый алгоритм) показаны на рисунке 5.

Суммарная награда за эпизод обучения

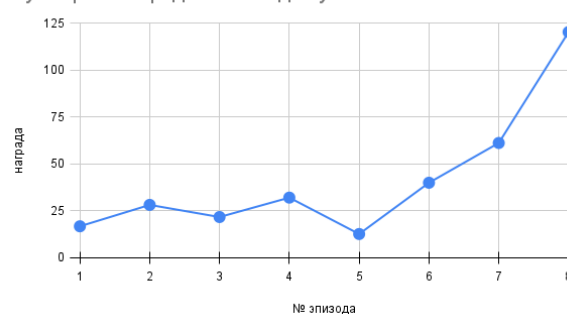


Рисунок 4. Суммарное значение вознаграждения относительно номера эпизода

При проведении эксперимента проводилось 10 измерений, погрешность рассчитывалась с использованием коэффициентов Стьюдента для доверительной вероятности $p = 0,95$. Время достижения цели с использованием алгоритма II составило 54,3 % относительно времени достижения цели алгоритма I, а алгоритма III – 40,2 % относительно времени достижения цели алгоритма I.

Время огибания препятствий

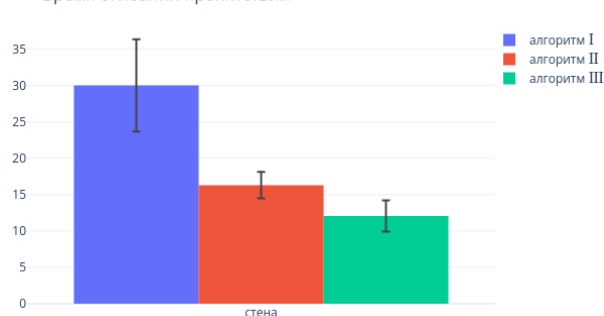


Рисунок 5. Временные параметры огибания препятствия роботом при использовании различных алгоритмов (в минутах)

Примеры движения робота по различным алгоритмам представлены в видео на платформе YouTube [6]. Как видно, предложенный алгоритм обладает наилучшими временными характеристиками.

Заключение

В данной работе мы предложили алгоритм огибания препятствий роботизированной системой с использованием машинного Q -обучения. Проведен вычислительный эксперимент по огибанию препятствия в виде длинной стены (рисунок 3). Оценено время движения до цели с огибанием препятствий (рисунок 5) в пространстве визуализации Gazebo 11 (рисунок 3) с использованием всех разработанных нами алгоритмов огибания препятствий. Выявлено, что с использованием предложенного

алгоритма III движение до цели без столкновения с препятствием происходит быстрее, чем при использовании алгоритмов, предложенных в работах [1] и [2]; время достижения цели для алгоритмов II и III составляет 54,3 % и 40,2 % относительно времени при использовании алгоритма I. Для каждого применяемого алгоритма записано видео с примером движения робота, которое затем было размещено на сайте YouTube.

В результате проведенной работы установлено, что верификация предложенного нами алгоритма продемонстрировала его преимущества перед ранее предложенными алгоритмами [1] и [2] в плане более высокой скорости достижения препятствия, а также его большую универсальность – данный алгоритм потенциально можно использовать для абсолютно любых типов препятствий, если правильно настроить необходимую плотность Q -таблицы для координатной сетки.

ЛИТЕРАТУРА

1. Сидоренко, А. В. Огибание препятствий при перемещении мобильного робота / А. В. Сидоренко, Н. А. Солодухо // Системный анализ и прикладная информатика. – 2023. – № 1. – С. 4-9. – DOI: 10.21122/2309-4923-2023-1-4-9
2. Сидоренко, А. В. Алгоритм огибания препятствий при перемещении мобильного робота / А. В. Сидоренко, Н. А. Солодухо // Системный анализ и прикладная информатика. – 2024. – № 4. – С. 29-33. – DOI: 10.21122/2309-4923-2024-4-29-33
3. Altuntas, N. Reinforcement learning-based mobile robot navigation / Nihal Altuntaş, Erkan Imal, Nahit Emanet, Ceyda Nur Ozturk // Turkish Journal of Electrical Engineering & Computer Sciences. – 2016. – Vol. 24, № 3. – P. 1747-1767. – DOI: org/10.3906/elk-1311-129
4. GitHub : [site]. – [San Francisco, California], 2008. – URL: <https://github.com/MikitaSaladukha/ROSproject/tree/with-blockchain-json> (date of access: 20.01.2025).
5. IPFS : [site]. – URL: <https://ipfs.tech/> (date of access: 20.01.2025).
6. YouTube : [site]. – URL: https://www.youtube.com/playlist?list=PLU4iF0_W9xOlE8brTv5WYbpmQgZQ0OHk (date of access: 20.01.2025).

REFERENCES

1. Sidorenko, A.V. Bending obstacles when moving a mobile robot / A.V. Sidorenko, N.A. Saladukha // System analysis and applied information science. – 2023. – № 1. – P. 4-9. – DOI: 10.21122/2309-4923-2023-1-4-9
2. Sidorenko A.V. Obstacle avoidance algorithm in mobile robot motion / A.V. Sidorenko, N.A. Saladukha // System analysis and applied information science. 2024. – № 4. – P. 29-33. – DOI: 10.21122/2309-4923-2024-4-29-33
3. Altuntas, N. Reinforcement learning-based mobile robot navigation / Nihal Altuntaş, Erkan Imal, Nahit Emanet, Ceyda Nur Ozturk // Turkish Journal of Electrical Engineering & Computer Sciences. – 2016. – Vol. 24, № 3. – P. 1747-1767. – DOI: org/10.3906/elk-1311-129.
4. GitHub : [site]. – [San Francisco, California], 2008. – URL: <https://github.com/MikitaSaladukha/ROSproject/tree/with-blockchain-json> (date of access: 20.01.2025).
5. IPFS : [site]. – URL: <https://ipfs.tech/> (date of access: 20.01.2025).
6. YouTube : [site]. – URL: https://www.youtube.com/playlist?list=PLU4iF0_W9xOlE8brTv5WYbpmQgZQ0OHk (date of access: 20.01.2025).

SIDORENKO A.V., SALADUKHA M.A.

ALGORITHM FOR OBSTACLE AVOIDANCE IN MOBILE ROBOT NAVIGATION USING Q-LEARNING AND BLOCKCHAIN TECHNOLOGY

Belarusian State University
Minsk, Republic of Belarus

Abstract. A robot movement modeling algorithm with obstacle avoidance using the Q-learning machine learning method is proposed. Q-learning allows for preserving the rewards obtained during modeling by performing optimal actions in each specific state. The Q-table contains information about the state and actions of the robot. Storing the Q-table in the blockchain using IPFS (InterPlanetary File System) technology ensures reliable and decentralized storage of data about the robot's states and actions. Content addressing in IPFS separates the data from its location and retrieves files from multiple sources in a peer-to-peer mode. A computational experiment for the proposed algorithm was conducted using a robot movement simulation environment. In the Gazebo 11 visualization package, it was shown that using the new algorithm, obstacles are avoided faster (by 59.8 %) compared to the previous version of the algorithm.

Keywords: robot, blockchain, IPFS, Q-learning, movement, obstacles, ROS 2, Gazebo 11



Сидоренко Алевтина Васильевна, Белорусский государственный университет, Минск, Республика Беларусь. Доктор технических наук, профессор факультета радиофизики и компьютерных технологий.

A. V. Sidorenko, Belarusian State University, Minsk, Republic of Belarus, D.Sc., Professor at the Department of radiophysics and computer technologies faculty.

E-mail: sidorenkoa@yandex.by



Солодухо Никита Александрович, Белорусский государственный университет, Минск, Республика Беларусь. Магистр физико-математических наук, старший преподаватель факультета радиофизики и компьютерных технологий.

Mikita A. Saladukha, Belarusian State University, Minsk, Republic of Belarus. Master of Science. Senior Lecturer at the Department of radiophysics and computer technologies faculty.