ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ
INTELLIGENT SYSTEMS

УДК 004.032.26, 004.942, 519.876.5
https://doi.org/10.37661/1816-0301-2024-21-3-48-62

Оригинальная статья
Original Article

Разработка метода подражательного обучения для нейросетевой системы управления движением мобильного робота на примере задачи поиска выхода из лабиринта

Т. Ю. Ким✉, Г. А. Прокопович

Объединенный институт проблем информатики
Национальной академии наук Беларуси,
ул. Сурганова, 6, Минск, 220012, Беларусь
✉*E-mail: tatyana_kim92@mail.ru*

Аннотация
Цели. Поставлена цель разработать новый метод обучения системы управления мобильным роботом поиску выхода из лабиринта на основе обучения с подкреплением и алгоритма правой руки.
Методы. В работе применен метод компьютерного моделирования в среде MATLAB/Simulink.
Результаты. Предложен новый метод обучения системы управления мобильным роботом, способный реализовывать алгоритм правой руки для поиска выхода из лабиринта. Данный метод основан на работе двух агентов, взаимодействующих между собой: первый непосредственно реализует поисковый алгоритм и ищет выход из лабиринта, а второй, следуя за ним, с помощью метода подражательного обучения пытается научиться находить выход из лабиринта. Агент-эксперт, реализуя дискретный алгоритм движения по лабиринту, совершает точные дискретные шаги и движется почти независимо от второго агента. Единственным ограничением является скорость его движения, которая прямо пропорционально зависит от расстояния между агентами. Второй агент, агент-ученик, методом проб и ошибок старается сократить расстояние до первого. Для реализации процесса обучения использовался метод обучения с подкреплением в режиме подражания, для которого была разработана соответствующая функция вознаграждения, позволяющая удерживать центр масс робота в центре коридора и при необходимости поворачивать, следуя за агентом-экспертом. Агенты передвигаются по виртуальному полигону, состоящему из разветвленных коридоров, достаточно широких для реализации различных маневров движений.
Заключение. Было доказано, что благодаря предложенному методу подражательного обучения агент-ученик способен не только перенимать от агента-эксперта требуемые паттерны поведения (искать в ранее неизвестном лабиринте выход по алгоритму правой руки), но и самостоятельно приобретать новые (изменять скорость на повороте, обходить небольшие коридоры-тупики), которые положительным образом влияют на выполнение поставленной задачи.

Ключевые слова: мобильный робот, агент, обучение с подкреплением, алгоритм правой руки, лабиринт, подражательное обучение

Благодарности. Работа была выполнена при поддержке гранта БРФФИ Ф22КИТГ-002 и задания Т31 ГПНИ «Цифровые и космические технологии, безопасность человека, общества и государства» (2021–2025).

Для цитирования. Ким, Т. Ю. Разработка метода подражательного обучения для нейросетевой системы управления движением мобильного робота на примере задачи поиска выхода из лабиринта / Т. Ю. Ким, Г. А. Прокопович // Информатика. – 2024. − Т. 21, № 3. – С. 48–62.
https://doi.org/10.37661/1816-0301-2024-21-3-48-62

Конфликт интересов. Авторы заявляют об отсутствии конфликта интересов.

Поступила в редакцию | Received 18.07.2024
Подписана в печать | Accepted 15.08.2024
Опубликована | Published 30.09.2024

Development of an imitation learning method for a neural network system of mobile robot's movement on example of the maze solving

Tatyana Yu. Kim✉, Ryhor A. Prakapovich

The United Institute of Informatics Problems
of the National Academy of Sciences of Belarus,
st. Surganova, 6, Minsk, 220012, Belarus
✉*E-mail: tatyana_kim92@mail.ru*

Abstract

Objectives. To develop a new method for training a mobile robot control system to use a maze solver algorithm based on reinforcement learning and the right-hand algorithm.

Methods. The work uses the method of computer modeling in the MATLAB/Simulink environment.

Results. A new method for training a mobile robot control system capable of implementing the right-hand algorithm for finding an exit from a maze is proposed. The proposed method is based on the work of two agents interacting with each other: the first directly implements the search algorithm and searches for an exit from the maze, and the second, following it, tries to learn using the imitation learning method. The expert agent, implementing a discrete algorithm for moving through the maze, makes precise discrete steps and moves almost independently of the second agent. The only limitation is its speed, which is directly proportional to the distance between the agents. The second agent, the student agent, tries to reduce the distance to the first agent by trial and error. The learning process was implemented using the reinforcement learning method, which was used in the imitation mode and for which a corresponding reward function was developed, allowing the robot's center of mass to be kept in the center of the corridor and, if necessary, to turn, following the expert agent. The agents move along a virtual polygon consisting of branched corridors wide enough to implement various movement maneuvers.

Conclusion. It was proven that, thanks to the proposed method of imitative learning, the student agent is able not only to adopt the required behavior patterns from the expert agent – to search for an exit in a previously unknown labyrinth using the right-hand algorithm, but also to independently acquire new ones (changing speed on a turn, bypassing small dead-end corridors), which positively influence the performance of the assigned task.

Keywords: mobile robot, agent, reinforcement learning, right-hand algorithm, maze, imitative learning

Acknowledgments. The work was supported by the BRFFR grant F22KITG-002 and the task T31 of the State Program for Scientific Research "Digital and Space Technologies, Security of Man, Society and the State" (2021–2025).

For citation. Kim T. Yu., Prakapovich R. A. *Development of an imitation learning method for a neural network system of mobile robot's movement on example of the maze solving*. Informatika [*Informatics*], 2024, vol. 21, no. 3, pp. 48−62 (In Russ.). https://doi.org/10.37661/1816-0301-2024-21-3-48-62

Conflict of interest. The authors declare of no conflict of interest.

Введение. Роботы все больше интегрируются в различные сферы жизни общества, включая здравоохранение, военную сферу, реагирование на стихийные бедствия, мониторинг окружающей среды и бытовые задачи. Способность к автономной навигации увеличивает их полезность при выполнении сложных задач в непредсказуемых условиях. Автономная навигация осуществляется, когда робот перемещается в среде без какого-либо вмешательства со стороны внешнего контроллера (например, человека). Она является одной из ключевых тем исследований в области мобильной робототехники [1]. Благодаря развитию искусственного интеллекта и компьютерного зрения были достигнуты огромные успехи в автономной навигации мобильных роботов [2]. Однако по-прежнему остается сложной задача обеспечения автономной навигацией мобильных роботов в реальном мире.

За последние несколько лет популярность глубокого обучения с подкреплением резко возросла. Она началась с двух историй успеха, когда впечатляющие результаты были достигнуты при сочетании обучения с подкреплением с глубокими нейронными сетями.

Во-первых, сообществом DeepMind разработан агент обучения с подкреплением, который способен играть одновременно в несколько видеоигр Atari 2600 на человеческом уровне [3]. В основе агента лежит метод, известный как глубокая Q-сеть, которая использует многослойную нейронную сеть в качестве аппроксиматора функции Q-обучения и решает проблему нестабильности. Во-вторых, сообществом DeepMind разработана программа AlphaGo [4], победившая чемпиона мира Ли Седоля в настольной игре Go. AlphaGo демонстрирует эффективное сочетание контролируемого обучения и обучения с подкреплением для освоения стратегической игры Go. Глубокое обучение с подкреплением в робототехнике по-прежнему остается сложной задачей, однако в последние годы оно применяется в таких областях, как роботизированная манипуляция [5], локомоция [6] и автономное управление автомобилем [7–9]. Трудности создания беспилотных систем управления заключаются не столько в проблемах распознавания различных объектов на дороге [10], сколько в описании сложных зрительных сцен и выборе соответствующей последовательности действий. Примером может служить шагающий робот, перемещающийся в сложной среде и адаптирующийся к изменениям условий с большой автономностью и эффективностью [11]. Как правило, при этом процесс обучения заключается в описании проблемы в виде оптимизационной задачи, для которой требуется найти минимум или максимум функции вознаграждения. Для большинства практических задач можно формализовать и реализовать процесс оптимизации на основе одной целевой функции, но есть задачи, где функция вознаграждения не является гладкой функцией, которую можно представить последовательностью условных (логических) операторов. Таким образом, в процессе усложнения задачи ставится цель усложнить вознаграждение для формирования лучшей политики. Также в процессе обучения агент привыкает решать однотипные задачи. В данных случаях традиционное обучение с подкреплением не может осуществляться с помощью обычных алгоритмов[1].

Из-за большого числа возможных вариантов точный выбор действий может стать сложной задачей. На высоких размерностях и длинных последовательностях действий такой подход уже не справляется. В результате при выполнении сложных задач в структурированных средах конечное решение принимает оператор или водитель с помощью сложных программных систем.

Чтобы правильно передвигаться, агент должен понимать окружающую среду и действовать согласно данным, соответствующим реальному миру. Для решения подобных задач существуют методы с применением подражательного обучения (Imitation Learning) [12]. Реализация данных методов заключается в том, что происходит взаимодействие только эксперта и награды, а ученик взаимодействует лишь с экспертом [13, с. 208]. В этих методах есть свои недостатки – это трудности в реализации наблюдения за движением эксперта, который, в свою очередь, контролирует двигательный аппарат ученика [14]. При этом ученик не видит всей картины и конечной цели.

[1]Part 2: Kinds of RL Algorithms [Electronic resource]. – Mode of access: https://spinningup.openai.com/en/latest/spinningup/rl_intro2.html. – Date of access: 12.06.2024.

На основе подражательного обучения авторами предложен новый метод реализации системы управления мобильным роботом. Традиционное обучение с подкреплением управляет учеником, а алгоритм правой руки – экспертом. Обучение является сложным, так как задача может быть каждый раз разной и иметь различную навигацию, в том числе включать различные образцы. Необходимо сделать так, чтобы ученик не просто копировал действие эксперта в процессе обучения, а, скорее, пытался достичь цели эксперта, выполняя новые действия, подобные таким, которые выполняет эксперт.

В настоящей работе описывается попытка создания программного комплекса для решения поставленных задач. Для упрощения и первого приближения задача автономного управления может быть формализована как задача поиска выхода из лабиринта. В качестве системы правил движения в условной местности – лабиринте – было предложено воспользоваться алгоритмом выхода из лабиринта по правилу правой руки.

1. Исследование методов классического обучения с подкреплением. Метод обучения с подкреплением основан на реализации процесса максимизации некоторого сигнала вознаграждения при переборе различных вариантов поведения исследуемых систем-агентов. Агент учится выполнять те действия, которые могут принести ему наибольшую награду. В наиболее интересных и важных случаях действия агента могут влиять не только на локальные вознаграждения, получаемые немедленно, но и на возникшую ситуацию в целом. Формирование долгосрочной награды – достаточно сложный процесс, так как правильно сформированная награда принесет лучший результат, сократит время обучения и процесс обучения пройдет качественнее [15].

Принцип обучения с подкреплением показан на рис. 1, где s_t – состояние агента в момент времени t; a_t – действие, совершаемое агентом в момент времени t в среде. Следующее состояние s_{t+1} среды достигает действия a_t, при этом среда генерирует новую обратную связь r_{t+1} в следующем состоянии s_{t+1}. Действие a_{t+1} агент выполняет с помощью s_{t+1} и r_{t+1}, повторяя этот процесс до тех пор, пока не достигнет конца итерации [16].

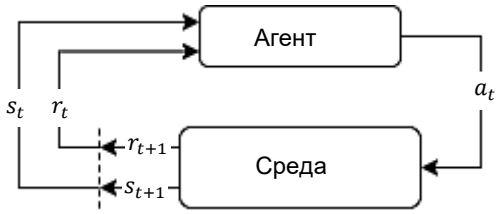


Рис. 1. Принцип обучения с подкреплением

Fig. 1. Reinforcement learning principle

Одним из классических алгоритмов обучения с подкреплением является алгоритм Q-learning, который обладает высокой надежностью и способностью адаптироваться к неопределенной среде [17], но в то же время имеет такие недостатки, как длительное время обучения, низкая эффективность исследования и медленная скорость сходимости [18].

1.1. Применение классического алгоритма Q-learning. Q-learning – это безмодельный алгоритм обучения с подкреплением, позволяющий узнавать ценность действий в определенном состоянии методом проб и ошибок. Оптимальное действие аппроксимируется с помощью алгоритма Q-learning, который постоянно обновляет функцию значения состояния-действия $Q(s_t, a_t)$ во время итерации [19]. Q-значение алгоритма Q-learning обновляется по формуле

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[R_{t+1} + \gamma \cdot \max_a Q(s_{t+1}, a) - Q(s_t, a_t)], \quad (1)$$

где $\alpha \epsilon [0,1]$ – параметр скорости обучения, R_t – вознаграждение состояния s в момент времени t, $\gamma \epsilon [0,1]$ – коэффициент дисконтирования.

В результате создается новая таблица, называемая Q-таблицей, в которой хранится вся информация о состоянии и действии агента.

В работах [20, 21] описано обучение с подкреплением, где использовался алгоритм Q-learning, идея которого состоит в том, чтобы применять табличный метод обучения, представленный по координатам.

Для того чтобы агент выполнял поставленную задачу, пространство его действий задается следующим образом:

$$\text{Action} = [\text{up}, \text{down}, \text{left}, \text{right}].$$

При этом угол направленности агента имеет вид

$$\text{Rot}_\text{A} = [90^\text{o}, -90^\text{o}, 0^\text{o}, 180^\text{o}].$$

В эксперименте, поставленном авторами, сетка (лабиринт) состоит из 27×27 ячеек. Координаты положения агента рассчитываются по формуле (1).

Классический алгоритм использует моделирование на основе сеток (таблиц), поэтому размер его пространства состояний может определяться размером таблицы. Один из недостатков Q-learning заключается в том, что матрица вознаграждений и политики имеет ту же размерность, что и лабиринт, и оценивается согласно карте, на которой обучался агент (рис. 2, *a*).

На рис. 2, *b* показан результат обученного агента на экспериментальной карте 27×27. Видно, что агент достиг конечной точки, так как сформированная политика и матрица вознаграждения соответствуют первоначальной карте, на которой обучался робот. Если изменить размер или цель в лабиринте, то агенту необходимо переучиваться с нуля на новом лабиринте.

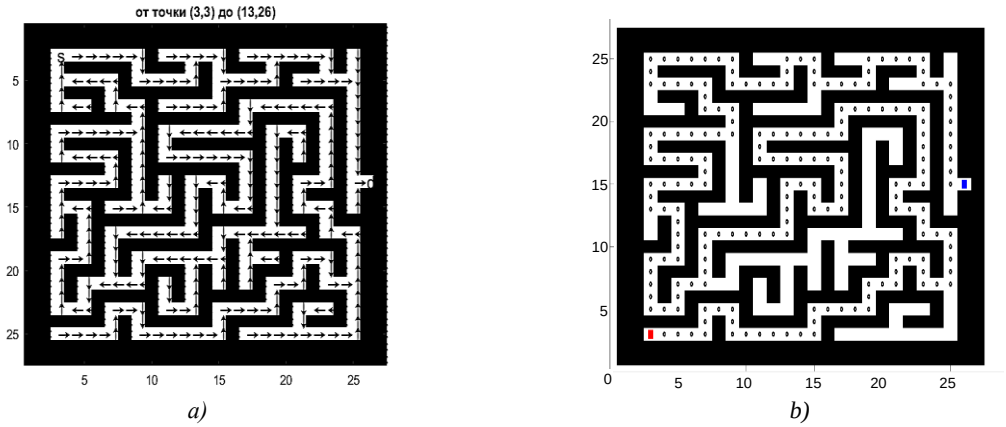


Рис. 2. Реализация алгоритма Q-learning: *a)* иллюстрация функции политики с помощью диаграммы политики; *b)* результат моделирования алгоритма

Fig. 2. Implementation of the Q-learning algorithm: a) illustrating a policy function using a policy diagram; b) algorithm simulation result

Черным цветом обозначены стены, красным – стартовая позиция, синим – выход из лабиринта
The walls are indicated in black, the starting position in red, and the exit from the maze in blue

Для того чтобы преодолеть возникшую проблему, представим состояния более общим способом в виде пикселов на карте и данных с лидара. Состояния, выраженные в виде функции, используют аппроксимацию функции, которая позволит обобщить множество различных лабиринтов, применяемых в обучении. Авторам неизвестны работы, в которых доказана возможность прохождения обученным роботом по неизвестной карте.

1.2. Алгоритм Актор-Критик обучения с подкреплением. Более сложным и вариабельным методом реализации обучения с подкреплением является подход «глубокое обучение с подкреплением», использующий современный алгоритм Актор-Критик, который позволяет работать в непрерывных средах [22, 23]. Алгоритм содержит две нейронные сети (рис. 3): Актор и Критик. Актор решает, какое действие следует предпринять, а Критик сообщает Актору, насколько хорошим было действие и как его следует откорректировать.

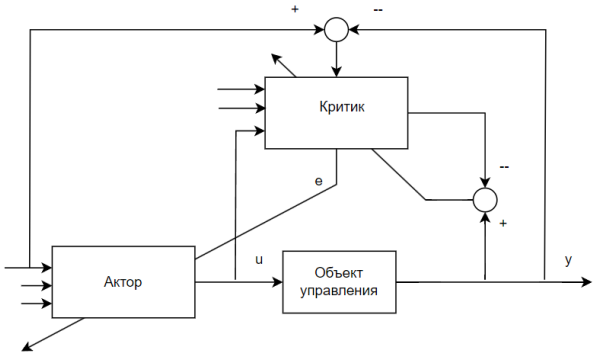


Рис. 3. Архитектура алгоритма Актор-Критик
Fig. 3. Architecture of the Actor-Critic Algorithm

Обучение объекта управления основано на стратегическом градиентном подходе. Критик оценивает действие, произведенное Актором, вычисляя функцию вознаграждения. Поэтому разработка функции вознаграждения для решения задач – возможно, самое важное в глубоком обучении с подкреплением. Награда должна помогать агенту совершать действия, которые максимизируют долгосрочную награду, а чем больше рекомендаций дать, тем быстрее и правильнее агент научится. Дополнительным критерием является получение награды агентом после выполнения определенного действия. Цель агента – получить награду, которая будет соответствовать поставленной задаче.

Для того чтобы агент передвигался по лабиринту, сначала необходимо создать подходящие условия, а затем обучить агента. Для этого следует выполнить следующие шаги:

Наблюдение. Данные для наблюдения представлены 27-элементным вектором, в котором 25 элементов содержат информацию с лидара и два элемента – сигналы обратной связи по линейной и угловой скоростям.

Критерии остановки. Моделирование начинается заново, если минимальное расстояние, полученное по показаниям лидара ($\mathrm{lid}_{\min}$), меньше радиуса ученика (2) или сумма каждого четвертого элемента вектора (3) равна нулю:

$$\mathrm{isdone} = \begin{cases} \mathrm{lid}_{\min} \le \mathrm{Rob}_{\mathrm{rad}}, & (2) \\ \sum_{i=1}^{5} v_{4i}(t) = 0, & (3) \end{cases}$$

где $v_{4i}(t)$ – каждый четвертый элемент 20-значного вектора (v) в момент времени (t).

Вознаграждение. На данном этапе формируется положительное вознаграждение, называемое наградой. Лидар определяет ближайшее препятствие впереди на пути передвижения, а получение награды поощряет прямолинейное движение по следующей формуле:

$$\mathrm{lid}_{\min} = \frac{1}{5} \cdot \sum_{k=-2}^{2} \mathrm{lid}_{i+k} - \mathrm{Rob}_{\mathrm{rad}}, \quad (4)$$

где lid_{i+k} – значение пяти центральных лучей лидара с (k), изменяющимся от –2 до 2.

Учитывая минимальный показатель лидара ($\mathrm{lid}_{\min}$), агент не допускает столкновения с препятствием. Награда вычисляется по формуле

$$\mathrm{reward} = 0{,}5 \cdot \mathrm{lid}_{\min} + 0{,}8 \cdot v_{\mathrm{rew}} + k \cdot \mathrm{finish}, \quad (5)$$

где v_{rew} – средняя линейная скорость последних пяти шагов моделирования; k =10 – коэффициент, который подбирался экспериментально для того, чтобы агент получил наивысшую

награду за выход из лабиринта и наказание за столкновение с внешней стеной; finish – переменная, которая принимает значение 1, если ученик достиг выхода из лабиринта, и 0 – в противном случае.

Отрицательное вознаграждение, называемое штрафом (наказанием) агента за каждое действие, побуждает его делать как можно меньше лишних шагов и сокращать время обучения. Для этого было решено использовать угловую скорость (w), чтобы уменьшить ее влияние на линейную (v). Функция штрафа вычисляется по формуле

$$\text{penalty} = 0{,}2 \cdot w + v_{\text{penalty}} + 9 \cdot \text{isdone}, \tag{6}$$

где w – угловая скорость, isdone – условия критериев остановки (2), (3), а v_{penalty} определяется по условию

$$v_{\text{penalty}} \begin{cases} -1, \text{если } v = 0, \\ 0, \text{если } v > 0. \end{cases} \tag{7}$$

Совместив награду (5) и штраф (6), получим функцию вознаграждения для агента:

$$\text{reward}_{\text{Agent}} = \text{reward} - \text{penalty}. \tag{8}$$

Обучение. Шаг моделирования составляет 0,1 с. Моделирование завершается, если закончилось время, отведенное для этого, или агент достиг своей цели. На рис. 4 видно, что обученный агент обходит препятствия и поворачивает, но его траектория еще не соответствует предполагаемой (выделена зеленым).

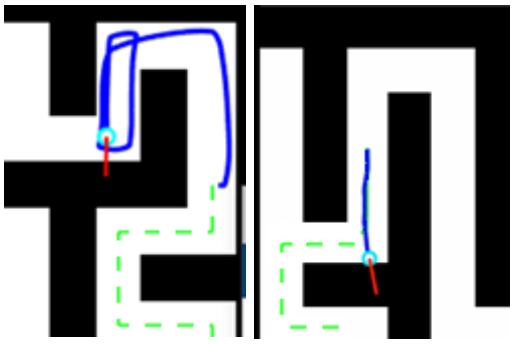


Рис. 4. Синий путь – траектория движения агента, который передвигается по коридору, зеленый путь – предполагаемая для него траектория

Fig. 4. The blue path is the trajectory of the agent moving along the corridor, the green path is the expected trajectory

Таким образом, при применении алгоритма Q-learning идет поиск оптимального пути, что хорошо для решения определенных задач. Однако сформированная таблица актуальна только для карты, которая была на этапе обучения. Обученный агент находил выход из лабиринта, указав на карте любую стартовую точку. Тот же агент может не справиться, если разместить его на новой карте, так как она не соответствует его политике обучения. Кроме того, агент действует дискретно, т. е. передвигается по клеткам, и действия его ограничены. При применении алгоритма Актор-Критик агент действует непрерывно. Данные, полученные от наблюдения, недостаточны для поставленной цели, так как функция вознаграждения сложная, нелинейная и разрывная и требует последовательных логических решений после получения определенных данных от сенсора. Поэтому авторы предлагают свой метод реализации поставленной задачи.

2. Комбинированный метод системы управления с применением обучения с подкреплением и алгоритма правой руки. Так как широко используемые методы обучения с подкреплением не позволили реализовать обучение нейросетевой системы управления автономным роботом в задаче поиска выхода из лабиринта, авторами был предложен кардинально новый подход. Ключевой идеей предлагаемого метода является введение в систему управления дополнительного агента, выполняющего роль эксперта, который действует по правилам так, как бы хотелось авторам, чтобы действовал искомый агент-ученик. В этом случае агент представляет ученика. Тогда функция вознаграждения будет определяться физическими величинами, которыми можно описать систему взаимодействия эксперта и ученика.

2.1. План эксперимента. Предположим, что есть два агента, условно называемые экспертом и учеником, которых поместили в неизвестную среду (лабиринт). Предлагается использовать навигационную схему эксперта, который руководствуется дискретным алгоритмом правой руки. Цель ученика – найти выход из лабиринта, следуя за экспертом и используя алгоритм обучения с подкреплением. Суть эксперимента заключается в том, чтобы обучить ученика следовать за экспертом, причем таким образом, чтобы положение ученика в процессе движения по лабиринту было максимально близким к центру коридора. При этом робот может двигаться только вперед, совершая при необходимости повороты и даже развороты. В результате это помогает ученику держаться центра коридора и гарантированно найти выход.

На рис. 5 показана схема предлагаемого метода совместного использования алгоритма обучения с подкреплением и алгоритма правой руки. Дискретный агент функционирует на основе алгоритма правой руки и выступает в качестве ориентира для обучения второго, аналогового, агента, нейросетевая система управления которого способна экстраполировать входные сенсорные данные на управляющие сигналы своего привода. Оба алгоритма взаимодействуют со средой, от которой они получают сенсорные данные о местоположении и углах направления их движений. Однако аналоговый агент получает больше сенсорных данных, так как у него есть собственный лидар, также он получает сенсорные данные о расстоянии до первого агента. В свою очередь, дискретный агент получает от второго агента управляющий сигнал о расстоянии между ними, который может замедлять его собственную скорость движения.

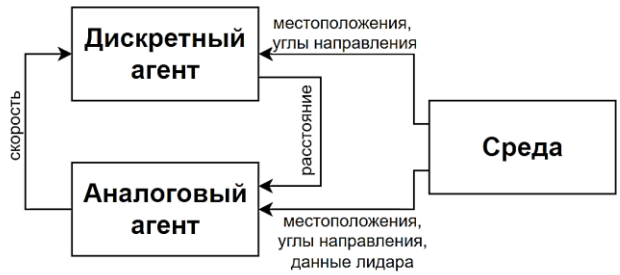


Рис. 5. Схема предложенного метода применения обучения с подкреплением совместно с алгоритмом правой руки

Fig. 5. Scheme of the proposed method of applying reinforcement learning together with the right-hand algorithm

Рассмотрим функциональную схему на рис. 6, где *осн* – оценочное состояние награды. Красной пунктирной линией обозначено состояние, от которого зависит скорость передвижения эксперта по лабиринту. Предложенная комбинация глубокого обучения с подкреплением и алгоритма правой руки работает следующим образом. Блок Среда включает ученика и эксперта. Блок Ученик, который является объектом управления, принимает множество действий (v, w) из блоков Актора и Критика. Сеть Критика корректирует (направляет) обучение Актора путем оценивания значения награды (*осн*) за действие-состояние. Актор выдает окончательную стратегию обучения в каждой итерации.

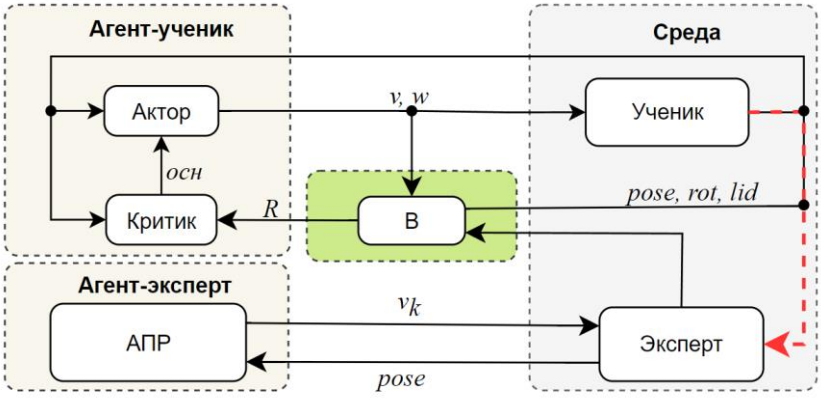


Рис. 6. Схема применения системы Актор-Критик глубокого обучения с подкреплением в блоке Агент-ученик совместно с алгоритмом правой руки (АПР) в блоке Агент-эксперт

Fig. 6. The application diagram shows the Actor-Critic system of deep reinforcement learning in the Agent-student block together with the right-hand algorithm (RHA) in the Agent-expert block

2.2. Реализация предложенного метода. Основополагающим этапом является разработка функции вознаграждения для ученика. Полученная награда должна помогать агенту-ученику обучаться. Чем больше ограничений (рекомендаций) будет иметь функция вознаграждения, тем быстрее и правильнее он научится. Дополнительным критерием является то, что агент получает награду, когда выполнит действие.

Для реализации предложенного метода необходимо выполнить следующие действия:

Наблюдение. В процессе обучения было определено, что влияние сигналов обратной связи по линейной и угловой скоростям недостаточно велико в сравнении с влиянием сигналов лидара. Для повышения влияния сигналов скорости на результат обучения в наблюдение были добавлены следующие величины: $\sin(v), \cos(v), \sin(w), \cos(w), v'', w''$. Всего шесть элементов, где некоторая нелинейная функция от линейной и угловой скоростей дополняет массив входных данных лидара и позволяет избежать повышения входных данных лидара. В качестве нелинейной функции выбраны функции синуса и косинуса. Наблюдаемые данные представлены в виде 33-элементного вектора, где 25 элементов содержат информацию с лидара, два элемента – сигналы обратной связи по линейной и угловой скоростям и шесть элементов – нелинейные функции синуса и косинуса, укорения (v'') и углового ускорения (w'').

Критерии остановки. В ходе многочисленных экспериментов были определены критерии остановки. Остановка происходит в следующих случаях:

– сумма каждого четвертого элемента вектора линейной скорости 20-элементного вектора равна нулю (9);

– минимальное расстояние до препятствия по показаниям лидара меньше, чем радиус объекта управления ученика (физическая составляющая) (10);

– ученик нашел выход из лабиринта (11);

– евклидово расстояние между учеником и экспертом (C) больше, чем $C_{\max}$, где $C_{\max}$ = = 0,3 м (12):

$$\text{isdone} = \begin{cases} \sum_{i=1}^{5} v_{4i}(t) = 0, & (9) \\ \text{lid}_{\min} \le \text{Rob}_{\text{rad}}, & (10) \\ \begin{cases} x_{\text{real}} = x \\ y_{\text{real}} = y, \end{cases} & (11) \\ C > C_{\max}. & (12) \end{cases}$$

Вознаграждение. Для гарантии, что агент-ученик пройдет весь путь, введем переменную dist – расстояние между начальной и текущей позициями ученика. Чем дальше ученик прошел за экспертом, тем больше он получит награду. Далее, для того чтобы ученик притормаживал при повороте либо в центре перекрестка и следовал алгоритму правой руки, в формуле (4) заменим переменную $\mathrm{Rob}_{\mathrm{rad}}$ на переменную width (ширина коридора). В результате получим равенство

$$\mathrm{lid}_{\mathrm{rot}} = \frac{1}{5} \cdot \sum_{k=-2}^{2} \mathrm{lid}_{i+k} - \frac{\mathrm{width}}{2}. \tag{13}$$

Как будет показано далее на рис. 7, *a* и *b*, ученик передвигается по коридору, но не способен преодолеть тупик, поэтому было решено ввести переменную impasse, которая зависит от линейной и угловой скоростей:

$$\mathrm{impasse} = \begin{cases} 0{,}5, & \text{если } \sum_{i=t-1}^{t} v_i = 0 \text{ и } \sum_{j=t-5}^{t} w_j \neq 0, \\ -0{,}5, & \text{если } \sum_{i=t-1}^{t} v_i = 0 \text{ и } \sum_{j=t-5}^{t} w_j = 0, \end{cases} \tag{14}$$

где v_i – значение линейной скорости, w_j – значение угловой скорости, t – текущий момент времени.

В результате получим награду по формуле

$$\mathrm{reward} = \mathrm{dist} + \mathrm{lid}_{\mathrm{rot}} + v + 0{,}0015 \cdot w^2 + 0{,}3 \cdot \mathrm{impasse} + 10 \cdot \mathrm{finish}. \tag{15}$$

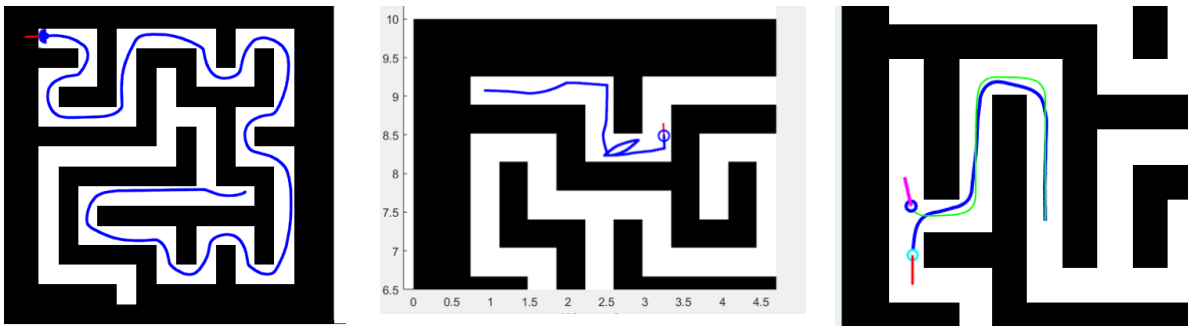
Далее разработаем штраф для обучения. Чтобы побудить ученика смотреть в том же направлении, что и эксперт (рис. 7, *c*), согласно алгоритму правой руки, вычислим разницу между углами поворотов агентов и получим rot_{α}. Для сокращения евклидова расстояния между агентами воспользуемся переменной C (12). В итоге получим следующую формулу штрафа:

$$\mathrm{penalty} = 0{,}8 \cdot \mathrm{rot}_{\alpha} + C + 1{,}5 \cdot \mathrm{isdone}. \tag{16}$$

Совместив награду (15) и штраф (16), запишем функцию вознаграждения для агента-ученика:

$$\mathrm{reward}_{\mathrm{agent}} = \mathrm{reward} - \mathrm{penalty}. \tag{17}$$

3. Результаты. На рис. 7 показаны промежуточные результаты, благодаря которым усовершенствована функция вознаграждения.

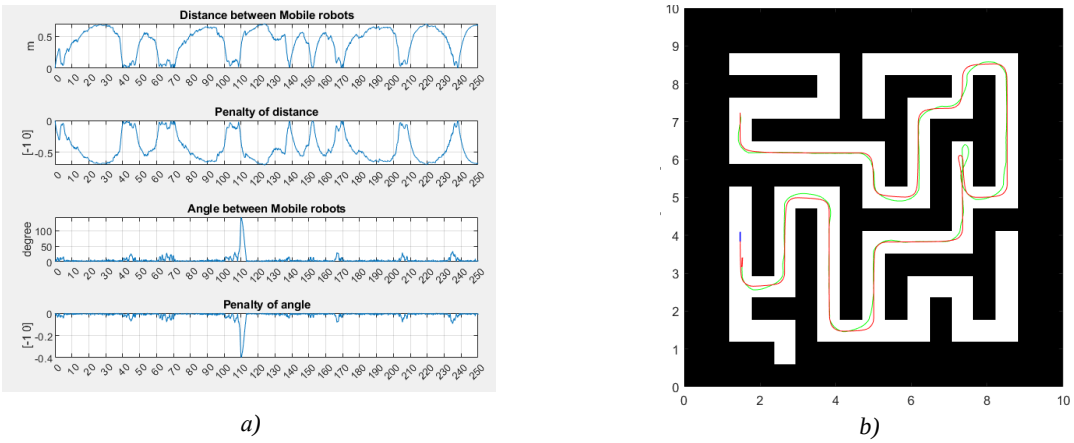


a) *b)* *c)*

Рис. 7. Промежуточный результат обучения: *a)* результат обучения без блока обхода тупика (14); *b)* влияние угловой скорости ученика; *c)* результат обучения до добавления штрафа на угловую скорость (16) и решение одностороннего направления между агентами

Fig. 7. Intermediate learning result: a) Result of training without a deadlock bypass block (14); b) the influence of the angular velocity of the student; c) the result of training before adding the penalty to the angular velocity (16) and the decision of one-way direction between agents

На рис. 8 изображен один из моментов обучения, когда ученик следует за экспертом. На визуализации хорошо прослеживается, как разница между направлениями агентов и расстоянием между ними влияет на общую награду.



a) *b)*

Рис. 8. Итоговый результат передвижения двух агентов после обучения: *a)* статистика передвижения агентов, где штрафная функция зависит от дистанции между роботами (тот же принцип действует на угол направленности между роботами); *b)* визуализация агентов, где красный и зеленый пути означают передвижения агента-эксперта и агента-ученика соответственно

Fig. 8. The final result of the two agents movement after training: a) statistics of agent movement, where the penalty function depends on the distance between robots, the same principle applies to the angle of orientation between robots; b) visualization of agents, where the red path and the green path represent the movement of the expert agent and the student agent respectively

Результаты экспериментов показывают, что спустя 2500 эпизодов ученик, следуя за экспертом, смог найти выход из лабиринта.

По завершении обучения агент-ученик принимает на себя управление, устраняя действия контроля агента-эксперта. Таким образом, остается только обученный ученик. Особенность обученного ученика заключается в том, что на участке, где эксперт следует правилу правой руки и потом направляется в тупик, ученик, предугадывая это, сокращает путь и направляется сразу к финишу. Это говорит о решении неоднотипной задачи для обучаемого агента-ученика. На рис. 9 места, где агент-ученик обходит данные участки, выделены красным.

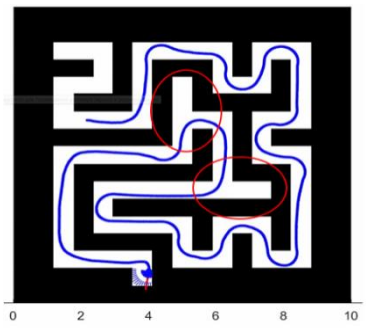


Рис. 9. Результат действия ученика после обучения без контроля эксперта

Fig. 9. The result of the student's action after training without the control of an expert

С помощью предложенной комбинации алгоритма обучения с подкреплением и алгоритма правой руки авторы вычислили, что ученик прошел лабиринт и нашел выход за 256 с.

4. Верификация обученного агента в различных средах. Обученный агент-ученик передвигается в незнакомой среде, где больше вариантов выбора повернуть направо или налево, количество стен и тупиковых ситуаций, вариаций ширины коридоров и стен.

Последовательно усложняя функцию вознаграждения, удалось решить задачу поиска пути в лабиринте. Для верификации обученного агента-ученика было решено проверить его поведение в незнакомой среде (рис. 10).

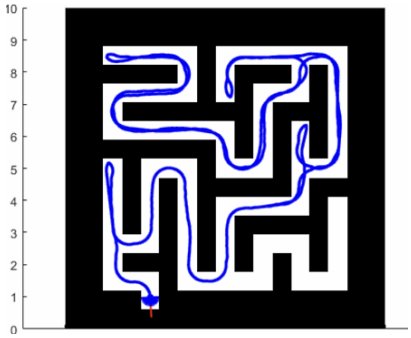


Рис. 10. Результаты моделирования поведения ученика в незнакомой среде

Fig. 10. Results of modeling student behavior in an unfamiliar environment

Результаты обучения показали, что совместное использование алгоритма обучения с подкреплением и алгоритма правой руки, а также разработанной функции вознаграждения побудило агента-ученика держаться центра коридора, совершать повороты и развороты по лабиринту, а также избегать столкновений.

Заключение. Предложенный авторами метод подражания предполагает совместное использование дискретного управления для эксперта и аналогового – для ученика. Несмотря на то что алгоритмы управления эксперта могут быть достаточно сложными и даже неизвестными ученику, путем анализа действий эксперта предложенный метод может обучить ученика сложному поведению. В результате подражательного обучения аналоговый агент приобретает новые паттерны поведения [24], что позволяет ему ориентироваться в неизвестной среде по правилу правой руки. Разработанный метод имеет большой потенциал для использования обучения с подкреплением в тех областях, где его раньше было сложно реализовать. В результате обученный ученик обобщает заданные действия и выводит собственные правила благодаря разработанной универсальной функции вознаграждения. Также преимущество предложенного метода состоит в том, что ученик передвигается с динамической скоростью, снижая ее при поворотах и достигая максимальной скорости на длинных дистанциях [25].

Вклад авторов. *Т. Ю. Ким* – разработка нового метода на базе глубокого обучения с подкреплением и алгоритма правой руки, программная реализация функции вознаграждения, верификация полученных результатов, проведение экспериментальных исследований с помощью классических алгоритмов обучения с подкреплением Q-learning и Актор-Критик. *Г. А. Прокопович* – постановка проблемы, разработка концепции статьи, обоснование актуальности работы, развитие ключевых целей и задач с помощью обучения с подкреплением, критический анализ работы.

Список использованных источников

1. Towards continuous control for mobile robot navigation: A reinforcement learning and slam based approach / K. A. A. Mustafa [et al.] // Intern. Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences. – 2019. – Vol. 42. – P. 857–863. https://doi.org/10.5194/isprs-archives-XLII-2-W13-857-2019

2. Truong, X. T. Toward socially aware robot navigation in dynamic and crowded environments: A proactive social motion model / X. T. Truong, T. D. Ngo // IEEE Transactions on Automation Science and Engineering. – 2017. – Vol. 14, no. 4. – P. 1743–1760. https://doi.org/10.1109/TASE.2017.2731371

3. Playing Atari with Deep Reinforcement Learning [Electronic resource] / V. Mhin [et al.]. – 2013. – Mode of access: https://doi.org/10.48550/arXiv.1312.5602. – Date of access: 20.06.2024.

4. Mastering the game of Go with deep neural networks and tree search / D. Silver [et al.] // Nature. – 2016. – Vol. 529, no. 7587. – P. 484–489.

5. Learning dexterous in-hand manipulation / M. Andrychowicz [et al.] // The Intern. J. of Robotics Research. – 2020. – Vol. 39, no. 1. – P. 3–20. https://doi.org/10.1177/0278364919887447

6. Emergence of Locomotion Behaviours in Rich Environments [Electronic resource] / N. Heess [et al.]. – 2017. – Mode of access: https://doi.org/10.48550/arXiv.1707.02286. – Date of access: 20.06.2024.

7. Autonomous vehicle perception: The technology of today and tomorrow / J. V. Brummelen [et al.] // Transportation Research Part C: Emerging Technologies. – 2018. – No. 86. – P. 384–406. https://doi.org/10.1016/j.trc.2018.02.012

8. Huang, W. Learning to drive via Apprenticeship Learning and Deep Reinforcement Learning [Electronic resource] / W. Huang, F. Braghin, Z. Wang. – 2020. – P. 1–7. – Mode of access: https://doi.org/10.48550/arXiv.2001.03864. – Date of access: 20.06.2024.

9. Robust AI driving strategy for autonomous vehicles / S. Nageshrao [et al.] // AI-enabled Technologies for Autonomous and Connected Vehicles. – Springer, 2022. – P. 161–212.

10. Sensor and sensor fusion technology in autonomous vehicles: A review / D. J. Yeong [et al.] // Sensors. – 2021. – Vol. 21, iss. 6. – P. 2140. https://doi.org/10.3390/s21062140

11. Kweon, J. Deep reinforcement learning for guidewire navigation in coronary artery phantom / J. Kweon, K. Kim, Ch. Lee // IEEE Access. – 2021. – Vol. 9. – P. 166409–166422. https://doi.org/10.1109/ACCESS.2021.3135277

12. An Algorithmic Perspective on Imitation Learning / T. Osa [et al.]. – Boston : Now publishers Inc., 2018. – 188 p.

13. Лонца, А. Алгоритмы обучения с подкреплением на Python / А. Лонца ; пер. с англ. А. А. Слинкина. – М. : ДМК Пресс, 2020. – 285 с.

14. Chella, A. Imitation learning and anchoring through conceptual spaces / A. Chella // Applied Artificial Intelligence. – 2007. – No. 21. – P. 343–359.

15. Kim, T. Automatic tuning of the motion control system of a mobile robot along a trajectory based on the reinforcement learning method / T. Kim, R. Prakapovich // Communications in Computer and Information Science. – Springer, Cham, 2022. – Vol. 1562. – P. 234–244. https://doi.org/10.1007/978-3-030-98883-8_17

16. Sutton, R. S. Reinforcement Learning: An Introduction / R. S. Sutton, A. G. Barto. – 2nd ed. – London, England : The MIT Press, 2014. – 352 p.

17. Watkins, C. Q-learning / C. Watkins, P. Dayan // Machine Learning. – 1992. – Vol. 8, iss. 3–4. – P. 279–292.

18. Duan, J. M. Prior knowledge based Q-learning path planning algorithm / J. M. Duan, Q. L. Chen // Electronics Optics & Control. – 2019. – Vol. 26, iss. 9. – P. 29–33.

19. Sutton, R. S. Reinforcement Learning: An Introduction / R. S. Sutton, A. G. Barto. – 2nd ed. – London, England : The MIT Press, 2014. – 338 p.

20. Rossi, F. Horizontal and vertical scaling of container-based applications using reinforcement learning / F. Rossi, M. Nardelli, V. Cardellini // 2019 IEEE 12th Intern. Conf. on Cloud Computing (CLOUD), Milan, Italy, 8–13 July 2019. – Milan, 2019. – P. 329–338. https://doi.org/10.1109/CLOUD.2019.00061

21. PAC model-free reinforcement learning / A. L. Strehl [et al.] // ICML'06: Proc. of the 23th Intern. Conf. on Machine Learning, Pittsburgh, Pennsylvania, USA, 25–29 June 2006. – Pittsburgh, 2006. – P. 881–888. https://doi.org/10.1145/1143844.114395

22. Ravichandiran, S. Deep Reinforcement Learning with Python / S. Ravichandiran. – 2nd ed. – Packt Publishing, 2020. – 760 p.

23. Yu, Ch. Supervised-actor-critic reinforcement learning for intelligent mechanical ventilation and sedative dosing in intensive care units / Ch. Yu, G. Ren // BMC Medical Informatics and Decision Making. – 2020. – No. 20 (S3). – P. 1–8. https://doi.org/10.1186/s12911-020-1120-5

24. Imitation learning: progress, taxonomies and challenges [Electronic resource] / B. Zheng [et al.] // IEEE Transactions on Neural Networks and Learning Systems. – 2022. – P. 1–22. – Mode of access: https://arxiv.org/abs/2106.12177. – Date of access: 20.06.2024.

25. Ким, Т. Ю. Форсированное управление движением мобильного робота / Т. Ю. Ким, Г. А. Прокопович, А. А. Лобатый // Информатика. – 2022. − Т. 19, № 3. – С. 86–100. https://doi.org/10.37661/1816-0301-2022-19-3-86-100

References

1. Mustafa K. A. A., Botteghi N., Sirmacek B., Poel M., Stramigioli S. Towards continuous control for mobile robot navigation: A reinforcement learning and slam based approach. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2019, vol. 42, pp. 857–863. https://doi.org/10.5194/isprs-archives-XLII-2-W13-857-2019

2. Truong, X. T., Ngo T. D. Toward socially aware robot navigation in dynamic and crowded environments: A proactive social motion model. *IEEE Transactions on Automation Science and Engineering*, 2017, vol. 14, no. 4, pp. 1743–1760. https://doi.org/10.1109/TASE.2017.2731371

3. Mhin V., Kavukcuoglu K., Silver D., Graves A., Antonoglou I., …, Riedmiller M. *Playing Atari with Deep Reinforcement Learning*, 2013. Available at: https://doi.org/10.48550/arXiv.1312.5602 (accessed 20.06.2024).

4. Silver D., Huang A., Maddison C. J., Guez A., Sifre L., …, Hassabis D. Mastering the game of Go with deep neural networks and tree search. *Nature*, 2016, vol. 529, no. 7587, pp. 484–489.

5. Andrychowicz M., Baker B., Chociej M., Józefowicz R., McGrew B., …, Zaremba W. Learning dexterous in-hand manipulation. *The International Journal of Robotics Research*, 2020, vol. 39, no. 1, pp. 3–20. https://doi.org/10.1177/0278364919887447

6. Heess N., Dhruva T. B., Sriram S., Lemmon J., Merel J., …, Silver D. *Emergence of Locomotion Behaviours in Rich Environments*, 2017. Available at: https://doi.org/10.48550/arXiv.1707.02286 (accessed 20.06.2024).

7. Brummelen J. V., O'Brien M., Gruyer D., Najjaran H. Autonomous vehicle perception: The technology of today and tomorrow. *Transportation Research Part C: Emerging Technologies*, 2018, no. 86, pp. 384–406. https://doi.org/10.1016/j.trc.2018.02.012

8. Huang W., Braghin F., Wang Z. *Learning to drive via Apprenticeship Learning and Deep Reinforcement Learning*, 2020, pp. 1–7. Available at: https://doi.org/10.48550/arXiv.2001.03864 (accessed 20.06.2024).

9. Nageshrao S., Rahman Y., Ivanovic V., Jankovic M., Tseng E., …, Filev D. Robust AI driving strategy for autonomous vehicles. *AI-enabled Technologies for Autonomous and Connected Vehicles*. Springer, 2022, pp. 161–212.

10. Yeong D. J., Velasco-Hernandez G., Barry J., Walsh J. Sensor and sensor fusion technology in autonomous vehicles: A review. *Sensors*, 2021, vol. 21, iss. 6, p. 2140. https://doi.org/10.3390/s21062140

11. Kweon J., Kim K., Lee Ch. Deep reinforcement learning for guidewire navigation in coronary artery phantom. *IEEE Access*, 2021, vol. 9, pp. 166409–166422. https://doi.org/10.1109/ACCESS.2021.3135277

12. Osa T., Pajarinen J., Neumann G., Bagnell J. A., Abbeel P., Peters J. *An Algorithmic Perspective on Imitation Learning*. Boston, Now publishers Inc., 2018, 188 p.

13. Lonza, A. *Reinforcement Learning Algorithms with Python*. Packt Publishing, 2019, 366 p.

14. Chella, A imitation learning and anchoring through conceptual spaces. *Applied Artificial Intelligence*, 2007, no. 21, pp. 343–359.

15. Kim T., Prakapovich R. Automatic tuning of the motion control system of a mobile robot along a trajectory based on the reinforcement learning method. *Communications in Computer and Information Science*. Springer, Cham, 2022, vol. 1562, pp. 234–244. https://doi.org/10.1007/978-3-030-98883-8_17

16. Sutton R. S., Barto A. G. *Reinforcement Learning: An Introduction*, 2nd edition. London, England, The MIT Press, 2014, 352 p.

17. Watkins C., Dayan P. Q-learning. *Machine Learning*, 1992, vol. 8, iss. 3–4, pp. 279–292.

18. Duan J. M., Chen Q. L. Prior knowledge based Q-learning path planning algorithm. *Electronics Optics & Control*, 2019, vol. 26, iss. 9, pp. 29–33.

19. Sutton R. S., Barto A. G. *Reinforcement Learning: An Introduction*, 2nd edition. London, England, The MIT Press, 2014, 338 p.

20. Rossi F., Nardelli M., Cardellini V. Horizontal and vertical scaling of container-based applications using reinforcement learning. *2019 IEEE 12th International Conference on Cloud Computing (CLOUD), Milan, Italy, 8–13 July 2019*. Milan, 2019, pp. 329–338. https://doi.org/10.1109/CLOUD.2019.00061

21. Strehl A. L., Li L., Wiewiora E., Langford J., Littman M. L. PAC model-free reinforcement learning. *ICML'06: Proceeding of the 23th International Conference on Machine Learning. Pittsburgh, Pennsylvania, USA, 25–29 June 2006*. Pittsburgh, 2006, pp. 881–888. https://doi.org/10.1145/1143844.114395

22. Ravichandiran S. *Deep Reinforcement Learning with Python*, 2nd edition. Packt Publishing, 2020, 760 p.

23. Yu Ch., Ren G. Supervised-actor-critic reinforcement learning for intelligent mechanical ventilation and sedative dosing in intensive care units. *BMC Medical Informatics and Decision Making*, 2020, no. 20 (S3), pp. 1–8. https://doi.org/10.1186/s12911-020-1120-5

24. Zheng B., Verma S., Zhou J., Tsang I., Chen F. Imitation learning: progress, taxonomies and challenges. *IEEE Transactions on Neural Networks and Learning Systems*, 2022, pp. 1–22. Available at: https://arxiv.org/abs/2106.12177 (accessed 20.06.2024).

25. Kim T. Yu., Prakapovich R. A. Lobatiy A. A. *Forced motion control of a mobile robot*. Informatika [*Informatics*], 2022, vol. 19, no. 3, pp. 86−100 (In Russ.). https://doi.org/10.37661/1816-0301-2022-19-3-86-100

Информация об авторах

Ким Татьяна Юрьевна, младший научный сотрудник, лаборатория робототехнических систем № 116, Объединенный институт проблем информатики Национальной академии наук Беларуси.
E-mail: tatyana_kim92@mail.ru
http://orcid.org/0000-0002-4126-6572

Прокопович Григорий Александрович, кандидат технических наук, доцент, Объединенный институт проблем информатики Национальной академии наук Беларуси.
E-mail: rprakapovich@robotics.by
http://orcid.org/0000-0002-3412-9174

Information about the authors

Tatyana Yu. Kim, Junior Researcher, Laboratory of Robotic Systems No. 116, The United Institute of Informatics Problems of the National Academy of Sciences of Belarus, Minsk, Belarus.
E-mail: tatyana_kim92@mail.ru
http://orcid.org/0000-0002-4126-6572

Ryhor A. Prakapovich, Ph. D. (Eng.), Assoc. Prof., The United Institute of Informatics Problems of the National Academy of Sciences of Belarus, Minsk, Belarus.
E-mail: rprakapovich@robotics.by
http://orcid.org/0000-0002-3412-9174