СИСТЕМНЫЙ АНАЛИЗ, УПРАВЛЕНИЕ И ОБРАБОТКА ИНФОРМАЦИИ, СТАТИСТИКА

УДК 004.8+629.113 Научная статья
DOI: 10.35330/1991-6639-2025-27-3-39-54
EDN: DHTLAK

Разработка системы управления курсом беспилотного автомобиля на основе обучения с подкреплением

А. Е. Ушаков[1], М. М. Стебулянин[1], М. А. Шереужев✉[1], Ф. В. Девяткин[1,2]

[1]Московский государственный технологический университет «СТАНКИН»
127055, Россия, Москва, Вадковский пер., 1
[2]Московский государственный технический университет имени Н. Э. Баумана
105005, Россия, Москва, 2-я Бауманская улица, 5

Аннотация. Рост развития автономного транспорта связан с повышением безопасности на дорогах, снижением столкновений и повышением эффективности логистических операций. На безопасность также влияет такой фактор, как усложнение дорожных условий и задач, связанных с навигацией и управлением автомобиля, и поэтому традиционные алгоритмы управления оказываются недостаточно качественными и эффективными. **Цель исследования** – разработка интеллектуальной системы, которая позволяет автономному транспортному средству самостоятельно управлять курсом движения автономного агента (модель автомобиля), который обучается навигации и следованию по заданному курсу с помощью обучения с подкреплением на основе взаимодействия с имитационной средой методом актер-критик. **Материалы и методы**. В данной работе для реализации и обучения модели с подкреплением использовалась библиотека Stable-Baselines3 (SB3), построенная на фреймворке PyTorch. В качестве среды обучения использовался симулятор DonkayCar. Для повышения скорости и эффективности обучения был применен алгоритм шумоподавляющего автокодера для выделения зоны интереса. **Результаты**. В рамках исследования была проведена серия сравнительных тестов, направленных на оценку влияния различных параметров эффективности обучения модели – ограничение скорости, ограничение угла поворота колес, ширины допустимого отклонения, непрерывности движения, коэффициента дисконтирования, частоты отрисовки кадров. **Выводы**. Результаты исследования позволяют сделать выводы о потенциале использования обучения с подкреплением в сфере автономного транспорта, включая необходимость дообучения модели на реальных данных, перспективы масштабирования на транспортные средства различного класса, ограничения, связанные с вычислительными ресурсами и необходимостью безопасной верификации поведения.

Ключевые слова: обучение с подкреплением, беспилотный автомобиль, Q-learning, DQN (Deep Q-Network), актер-критик, имитационное моделирование, интеллектуальная система, симуляционная среда, устойчивость обучения

Поступила 19.05.2025, одобрена после рецензирования 28.05.2025, принята к публикации 02.06.2025

Для цитирования. Ушаков А. Е., Стебулянин М. М., Шереужев М. А., Девяткин Ф. В. Разработка системы управления курсом беспилотного автомобиля на основе обучения с подкреплением // Известия Кабардино-Балкарского научного центра РАН. 2025. Т. 27. № 3. С. 39–54. DOI: 10.35330/1991-6639-2025-27-3-39-54

MSC: 68T05 Original article

Development of an unmanned vehicle course control system based on reinforcement learning

A.E. Ushakov[1], M.M. Stebulyanin[1], M.A. Shereuzhev✉[1], F.V. Devyatkin[1, 2]

[1]Moscow State University of Technology "STANKIN"
127055, Russia, Moscow, 1 Vadkovsky lane
[2]Bauman Moscow State Technical University
105005, Russia, Moscow, 5, 2-nd Baumanskaya street

Abstract. At present, there is a growing development of autonomous transportation, driven by the need to improve road safety, reduce collisions, and enhance the efficiency of logistics operations. This trend is also influenced by increasing complexity in road conditions and challenges related to vehicle navigation and control, which make traditional control algorithms insufficient in terms of quality and effectiveness. **Aim**. The objective of this research is to develop an intelligent system that enables an autonomous vehicle to independently control its course. The autonomous agent (a vehicle model) learns to navigate and follow a predefined trajectory using reinforcement learning through interaction with a simulation environment, based on the Actor-Critic method. **Materials and Methods**. In this work, the Stable-Baselines 3 (SB3) library built on the PyTorch framework was used to implement and train the reinforcement learning model. The DonkeyCar simulator served as the training environment. To improve the speed and efficiency of training, a denoising autoencoder algorithm was applied to extract the region of interest (ROI). **Results**. A series of comparative experiments was conducted to evaluate the impact of various parameters on training efficiency – such as speed limits, steering angle constraints, allowable deviation width from the lane center, movement continuity, discount factor, and frame rendering rate. **Conclusion**. The results of the study demonstrate the potential of reinforcement learning in the field of autonomous transport, while also highlighting the need for further training on real-world data, the prospects for scaling the approach to different classes of vehicles, and limitations related to computational resources and the need for safe behavior verification.

Keywords: reinforcement learning, unmanned vehicle, Q-learning, DQN (Deep Q-Network), actor-critic, simulation modeling, intelligent system, simulation environment, training stability

Submitted 19.05.2025, approved after reviewing 28.05.2025, accepted for publication 02.06.2025

For citation. Ushakov A.E., Stebulyanin M.M., Shereuzhev M.A., Devyatkin F.V. Development of an unmanned vehicle course control system based on reinforcement learning. *News of the Kabardino-Balkarian Scientific Center of RAS.* 2025. Vol. 27. No. 3. Pp. 39–54. DOI: 10.35330/1991-6639-2025-27-3-39-54

Введение

В последние годы наблюдается стремительное развитие технологий автономного вождения. Интеллектуальные транспортные системы становятся неотъемлемой частью концепции «умного города» и рассматриваются как ключевой элемент в повышении безопасности дорожного движения, снижении аварийности, а также оптимизации логистических и эксплуатационных процессов [1]. Одной из важнейших задач в этой области является разработка системы управления курсом движения беспилотного автомобиля, способной адаптироваться к изменяющимся условиям дорожной среды[1]. В условиях растущей сложности задач традиционные алгоритмы управления становятся недостаточно эффективными, что

[1]Когда беспилотные автомобили появятся на дорогах [Электронный ресурс]. Режим доступа: https://bespilot.com/chastye-voprosy/kogda-ba-poyavyatsya-na-dorogakh (дата обращения: 19.03.2025).

актуализирует использование методов искусственного интеллекта, в частности обучения с подкреплением (Reinforcement Learning, RL) [2–5].

Методы обучения с подкреплением в последние годы активно исследуются в научной среде и находят применение в широком спектре задач – от игр и робототехники до систем управления автономными транспортными средствами. Тем не менее практическая реализация RL в задачах управления курсом движения автомобиля все еще сопряжена с рядом трудностей [6], включая необходимость настройки среды, корректного определения функций награды, обеспечения стабильности и скорости обучения [7]. В отечественной и зарубежной литературе накоплен теоретический материал по алгоритмам Q-learning, DQN, Actor-Critic, однако реальных внедрений в условиях, приближенных к транспортным, пока немного, что подчеркивает научную и прикладную значимость исследования [8–9].

В будущем разработанная система может быть использована в составе более сложных автономных транспортных платформ. Также возможно ее применение в учебных и исследовательских целях для демонстрации и тестирования поведения RL-агентов в задачах управления мобильными объектами [1].

Цель исследования – разработка интеллектуальной системы управления курсом движения автономного транспортного средства на основе методов обучения с подкреплением. В условиях стремительного развития технологий автономного вождения и повышения требований к адаптивности и надежности систем управления особенно актуальными становятся подходы, позволяющие транспортному средству обучаться самостоятельному принятию решений в сложных и изменяющихся условиях дорожной среды.

В рамках исследования предполагается реализация системы, в которой агент (модель автомобиля) обучается навигации и следованию по заданной траектории, взаимодействуя с имитационной средой. Особенность подхода заключается в применении алгоритма актер-критик, позволяющего эффективно комбинировать достоинства методов обучения с политикой и ценностной функцией. Обучение осуществляется в условиях ограниченного пространства действий, ограничений по управлению (скорость, угол поворота) и с использованием специально разработанной функции вознаграждения, стимулирующей безопасное и эффективное поведение агента.

Методы

Традиционно обучение с подкреплением считалось сложной вычислительной задачей, тогда как нейронные сети относятся к задачам, работающим с большими объемами данных. Но развитие глубоких нейронных сетей пошло на пользу обучению с подкреплением. Сейчас нейронные сети начали широко использоваться в моделях обучения с подкреплением, что дало начало глубокому обучению с подкреплением [10].

Несмотря на недавние достижения, ландшафт обучения с подкреплением остается не очень удобным для разработчиков. Интерфейсы для обучения моделей постепенно становятся проще, но эта тенденция пока не проникла в сферу обучения с подкреплением. Еще одна сложность – значительные требования к вычислительным ресурсам и ко времени сходимости модели (до момента, когда обучение можно считать завершенным). Обучение модели может занимать дни, а то и недели [10].

Ключевой характеристикой обучения с подкреплением является то, что в процессе обучения система взаимодействует с окружающей средой. Таким образом, процесс является замкнутым, как представлено на рис. 1. Объектом обучения с подкреплением является так называемый агент, который принимает решения на основе совершения различных действий. Все, что находится вне агента, обозначается как среда [11]. Так, обучение с подкреп-

лением представляет собой процесс, в котором агент взаимодействует с окружающей средой, совершая различные действия, ощущая состояния и получая вознаграждения. Цель агента – научиться действовать таким образом, чтобы максимизировать общее накопленное в течение всего процесса вознаграждение [12].

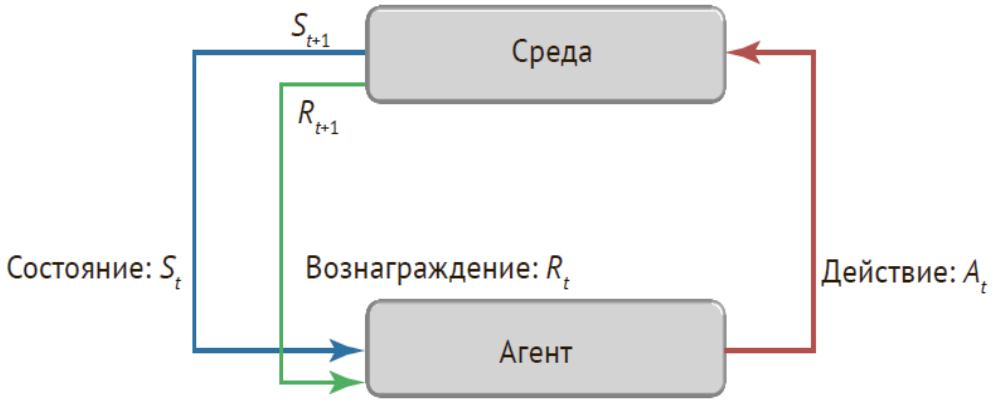


***Рис. 1**. Структура процесса обучения с подкреплением: агент взаимодействует с окружающей средой (совершает действие, ощущает состояние и получает вознаграждение) [6]*

***Fig. 1**. The structure of the reinforcement learning process: the agent interacts with the environment (performs an action, senses a state and receives a reward) [6]*

В симуляторе агент взаимодействует с окружающей средой, выполняя случайные действия, которые затем оцениваются с помощью функции вознаграждения. Каждому действию в определенном состоянии присваивается числовая оценка: чем она выше, тем более оптимальным считается выбранное действие. Информация о состоянии, действии, вознаграждении сохраняется в буфере воспроизведения и используется для обучения. Суть подхода заключается в том, чтобы обучить автопилот на основе накопленных оценок, начиная с активной фазы исследования среды, которая постепенно снижается по мере приобретения агентом опыта [10].

В симуляторе входные изображения обрабатываются с частотой от 20 до 120 кадров (в зависимости от настройки), и на каждом шаге автомобиль переходит в новое состояние. Сначала модель выбирает случайные действия, но со временем доля использования накопленного опыта увеличивается, а исследовательских – уменьшается. Этот процесс может следовать линейной или экспоненциальной стратегии с экспоненциальным уменьшением доли исследовательских действий в практике обучения с подкреплением (рис. 2).

Также остается проблема сбалансированного обучения системы управления. Глубокие нейронные сети требуют больших объемов данных, и 500 изображений могут быть недостаточны для полноценного обучения [13]. При ограниченном наборе данных сеть может переобучиться, запоминая обучающие данные, и показывает высокие результаты на них, но плохо обобщает новые данные. Это приводит к низкой точности на контрольном наборе, как показано на примере синусоидального распределения данных[2]. Простая модель не может точно аппроксимировать распределение, тогда как мощная (переобученная) модель может запомнить обучающие данные и показать действительно хорошие результаты (рис. 2)[3].

[2]Что такое TensorFlow? [Электронный ресурс]. Режим доступа: https://ru.realiventblog.com/ 27-what-is-tensorflow-the-machine-learning-library-explained (дата обращения: 26.03.2025).

[3]Pylot [Электронный ресурс]. Режим доступа: https://pylot.readthedocs.io/en/latest/index.html (дата обращения: 20.03.2025).

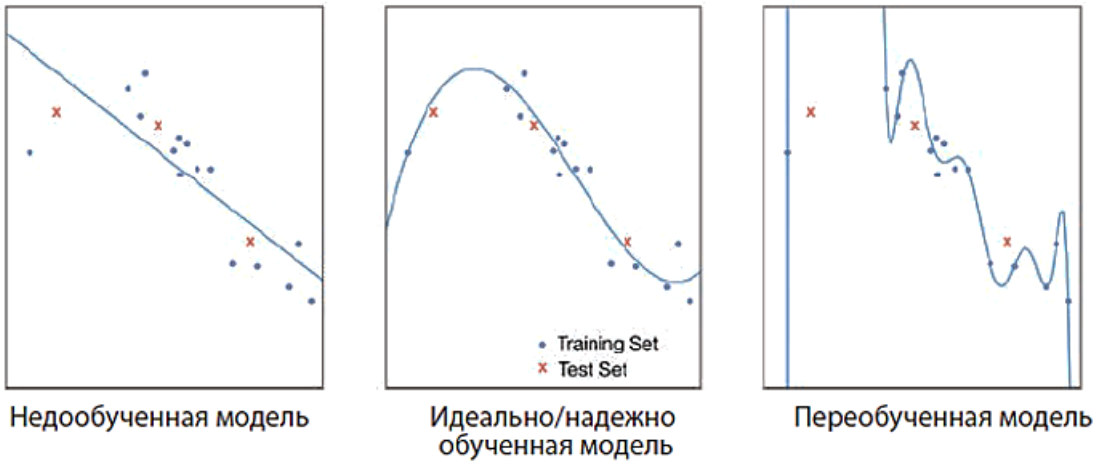


***Рис. 2**. Недообучение, переобучение и идеальное обучение на точках с распределением, близким к синусоидальному [12]*

***Fig. 2**. Underfitting, overfitting, and ideal learning on points with a distribution close to sinusoidal [12]*

В данной работе для реализации и обучения агентов с подкреплением использовалась библиотека Stable-Baselines3 (SB3), разработанная в Центре робототехники и мехатроники Немецкого аэрокосмического центра (DLR-RM)[4]. Stable-Baselines3 представляет собой современную реализацию популярных алгоритмов глубокого обучения с подкреплением, построенную на фреймворке PyTorch, и ориентирована на воспроизводимость, модульность и удобство интеграции в исследовательские и прикладные проекты [14].

При обучении в качестве симулятора взято программное обеспечение симулятора движения Donkey, в котором можно извлекать и воздействовать на такие параметры, представленные в табл. 1. Визуальное объяснение некоторых параметров функции вознаграждения показано на рис. 3 [11].

***Таблица 1**.* Параметры, которые можно извлекать и воздействовать на них [11]

***Table 1**.* Parameters that can be extracted and influenced [11]

Описание параметра	**Параметр**
Расстояние от осевой линии трека в метрах	"distance_from_center": float
Признак, что автомобиль находится слева от осевой линии трека	"is_left_of_center": Boolean
Отклонение автомобиля от основного направления в градусах	"heading": float
Доля трека в процентах, которую преодолел автомобиль	"progress": float
Количество пройденных шагов	"steps": int
Скорость автомобиля в м/с	"speed": float
Угол поворота колес автомобиля в градусах	"steering_angle": float
Ширина трека	"track_width": float
Список координат (x, y) путевых точек на осевой линии трека	"waypoints": [(float, float),]
Индексы двух ближайших путевых точек	"closest_waypoints": [int, int]

[4]Документация Stable-Baselines3 (SB3) [Электронный ресурс]. Режим доступа: https://stable-baselines3.readthedocs.io/en/master/index.html (дата обращения: 19.03.2025).

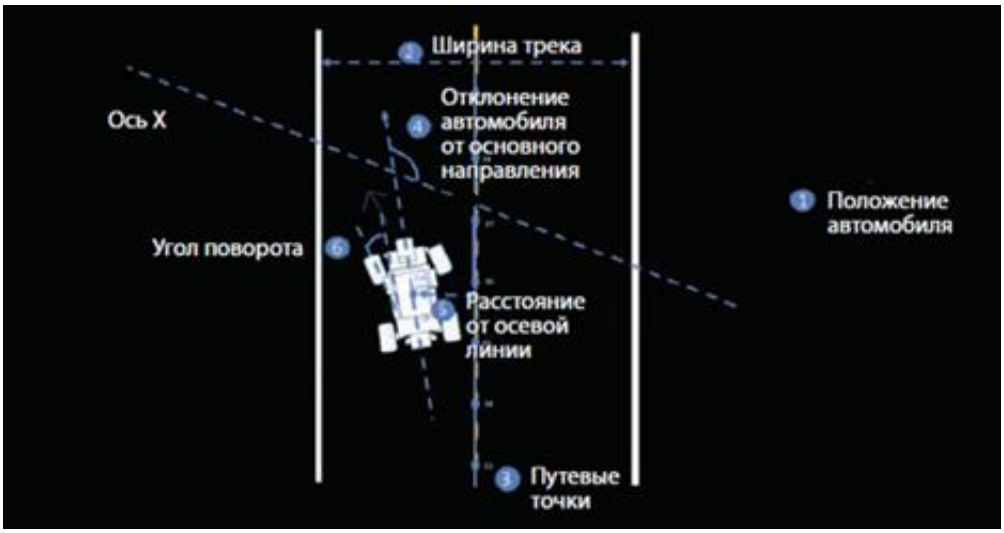


Рис. 3. *Визуальное объяснение некоторых параметров, которые можно извлекать и воздействовать на них [12]*

Fig. 3. *Visual explanation of some of the parameters that can be extracted and influenced [12]*

Логика, заложенная в данной функции вознаграждения, заключается в стремлении направить автомобиль по максимально безопасной траектории вдоль трассы. Вес вознаграждения очень важен – при правильной настройке параметров вероятность того, что автомобиль, движущийся по осевой линии, съедет с трека, значительно снижается. С этой целью трасса условно разделена на три зоны, каждая из которых соответствует определенному уровню поощрения – узкая центральная зона, вторая (нецентральная) зона и краевая зона (рис. 4). Назначение веса каждого параметра указано в табл. 2.

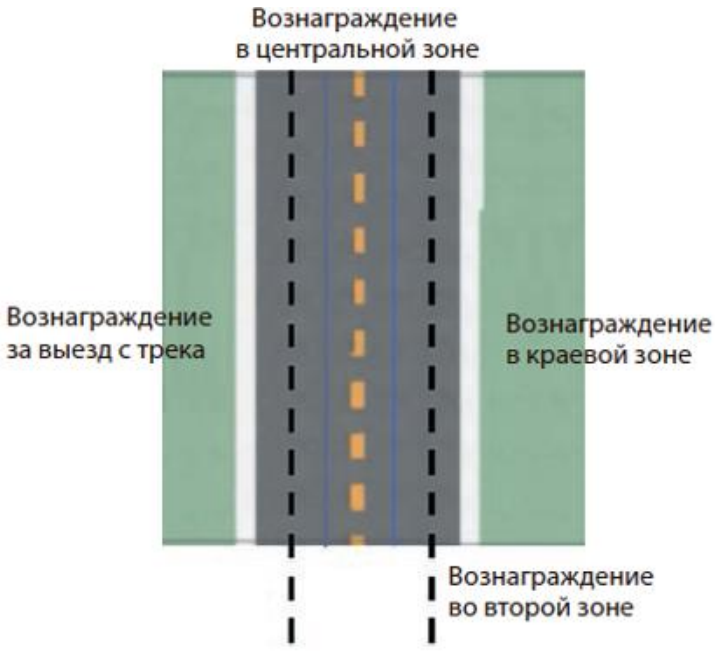


Рис. 4. *Определение зон вознаграждения [2]*

Fig. 4. *Definition of reward zones [2]*

Таблица 2. Назначение веса каждого параметра

Table 2. Assignment of weight to each parameter

Параметр	Вес
Нахождение в узкой центральной зоне	2
Нахождение во второй (нецентральной) зоне	0,5
Нахождение в краевой зоне	0,2

Если уменьшить вознаграждение за нахождение в центральной зоне или увеличить вознаграждение за нахождение во второй зоне, мы фактически побудим машину использовать большую часть поверхности трека [11].

Если мы хотим подготовить быстро и качественно модель, то нам нужно сократить пространство обзора камеры агента, чтобы на обработку изображений с камеры тратилось меньше вычислительных ресурсов. При взгляде на кадр с камеры его удобно разбить на три горизонтальных сегмента (рис. 6 а). Нижний сегмент можно исключить, т.к. там мало информации, которая нам требуется, такой как дорога и разметка, верхний сегмент тоже довольно однороден – в нем доминируют серо-белые оттенки неба и облаков. Средний же сегмент кардинально отличается: он заполнен многочисленными деталями, такими как дорога, разделительная полоса, ограждения и т. д.

Предложим автопилоту сосредоточиться исключительно на нужной части кадра для упрощения обучения. Для этого на изображении выделим область интереса ROI (Region Of Interest) [14]. Поскольку строгих требований к форме ROI не существует, ее выбор определяется конкретной задачей. В нашем случае камера жестко зафиксирована, а наклон дороги остается постоянным, поэтому зону ROI можно представить в виде простого прямоугольника, охватывающего проезжую часть и ее границы [15]. Выделение будет выполняется при помощи шумоподавляющего автокодера – принципиальная схема его работы показана на рис. 5.

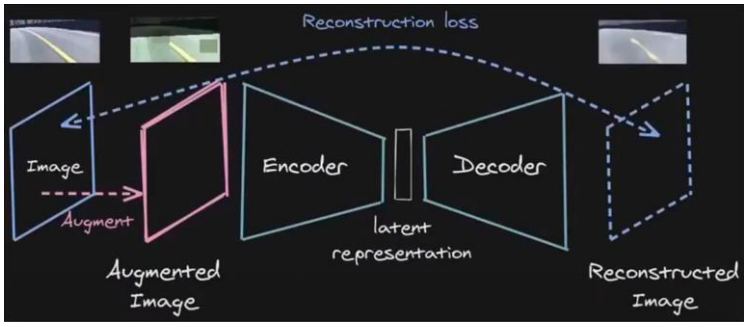


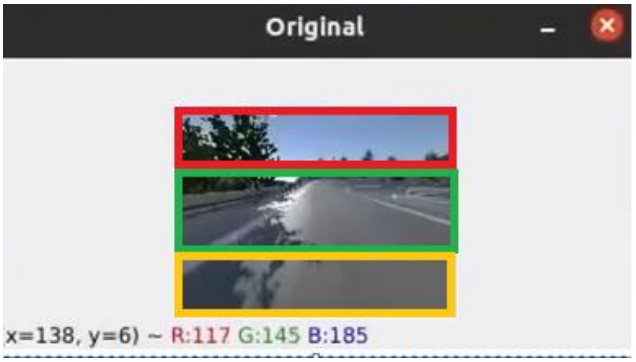
Рис. 5. *Схема работы шумоподавляющего автокодера*[5]

Fig. 5. *Operation scheme for the noise-suppressing autoencoder*

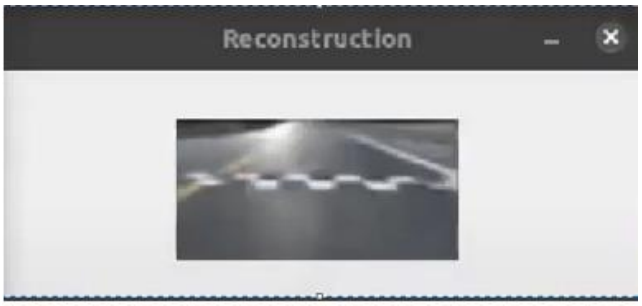
Суть его работы заключается в том, что есть изображение в качестве входных данных, оно кодируется в небольшом скрытом пространстве латентного вектора-представления этого изображения, и, используя этот вектор, оно пытается реконструировать изображение. В конечном итоге автокодер перебрасывает изображение после обработки на вход обратно. В результате после обработки можно увидеть изображение, как на рис. 6 б. Теперь модель будет принимать на вход изображение только непосредственно дороги. Другие изменения

[5]Araffin A. aae-train-donkeycar: GitHub repository. URL: https://github.com/araffin/aae-train-donkeycar (дата обращения: 19.05.2025).

в верхней и нижней зоне не будут входить, что позволит уменьшить размер изображения вдвое и существенно упростит обучение нейронной сети.



а) Выделение областей интереса



б) Работа шумоподавляющего автокодера

***Рис. 6**. Определение области интереса (разработано авторами)*

***Fig. 6**. Area of interest definition (developed by the authors)*

Путем тестов было выявлено что наиболее сбалансированный и эффективный набор сети (net_arch) [16] будет 256 x 256 точек. В таблице 3 представлен список всех требований, которые мы только что определили.

Таблица 3. Список компонентов и требований

Table 3. List of components and requirements

Компонент	**Требования**
Операционная система	Linux
Симулятор	DonkeyCar Simulator
Язык и среда программирования	Python. Visual Studio Code
CPU/GPU	Процессор от 8 ядер, частота 4.2 Гц Видеоускоритель с большим количеством видеопамяти – от 16 Гб
Другие инструменты + зависимости для Python	TensorFlow+Keras Шумоподавляющий автокодер Библиотека Stable-Baselines3 (SB3)
Размер сети	256 x 256
Обратные вызовы	Функция ReduceLROnPlateau

После определения требуемых параметров для обучения модели можно приступать непосредственно к процессу обучения.

Проведение эксперимента

Поначалу агент находится на старте (рис. 7 а), через какое-то время начинается выполнение действия.

После выполнения действия, выбранного случайно или на основе прежнего опыта, автомобиль переходит в новое состояние (рис. 7 б). С помощью функции вознаграждения вычисляется оценка, и результат присваивается выбранному действию.

Этот процесс повторяется для каждого состояния, пока не будет достигнуто конечное состояние, то есть когда автомобиль выкатится за пределы трека (рис 7 в) или завершит круг, после чего автомобиль будет установлен на старт, и заезд повторится. Шаг – это переход из одного состояния в другое, и на каждом шаге записываются данные (состояние, действие, вознаграждение, новое состояние). Коллекция шагов от начального до конечного состояния называется эпизодом.

Также на рис. 7 б показано, как агент проходит эпизод. Функция вознаграждения с весом 2 побуждает агента придерживаться осевой линии – это самый короткий и быстрый путь от старта до финиша.

а) агент находится на старте

б) агент переходит в новое состояние

в) агент выкатился за пределы трека

***Рис. 7**. Процесс обучения модели (разработан авторами)*

***Fig. 7**. Model training process (developed by the authors)*

Обсуждение результатов

Если агенту не ограничить (конкретизировать) условия обучения, то на общее обучение будет затрачено гораздо больше времени. Поэтому для агента задается ограничение угла поворота колес относительно основной прямой траектории движения. На графике (рис. 8) приведены три результата опыта при трех различных параметрах угла отклонения агента от траектории. Максимально «широкое» ограничение изменения угла отклонения от траектории агента (-0.8, 0.8). Усредненное значение ограничения изменения угла отклонения от траектории агента (-0.6, 0.6). Минимально «узкое» ограничение изменения угла отклонения от траектории агента (-0.5, 0.5). Так, при «расширенном» ограничении изменения угла отклонения от траектории агента в пределах (-0.8, 0.8) автомобиль затратит больше времени на прохождение того же пути, что он пройдет при более «суженном» ограничении изменения угла отклонения от траектории агента в пределах (-0.5, 0.5). Другими словами, если ограничение изменения угла отклонения от траектории агента будет задано в более «узких» предельных значениях, то времени на процесс обучения будет потрачено меньше, а пройденный путь будет больше.

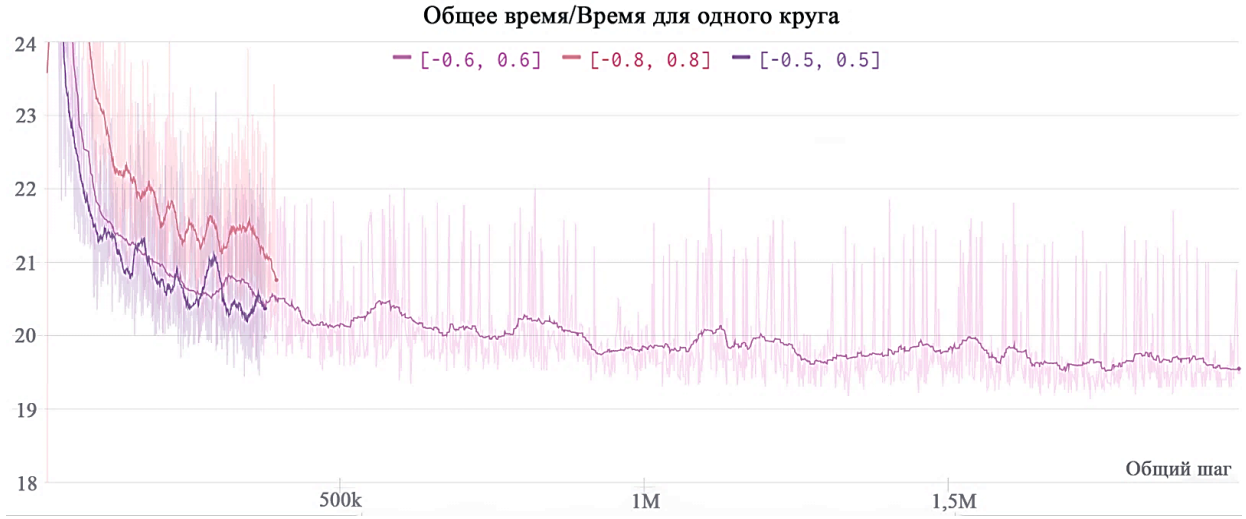


***Рис. 8**. График сравнения различных максимальных значений отклонения колес при разных ограничениях угла поворота*

***Fig. 8**. Comparison chart for maximum wheel deflection values at steering angle limits*

Ограничение использования скорости также либо ускоряет, либо усложняет процесс обучения агента. Как видно из графика (рис. 9), положительная единица, соответствующая ускорению автомобиля, прямо пропорциональна отрицательной единице торможения. Для исследования взяты разные отношения ограничений ускорения-торможения.

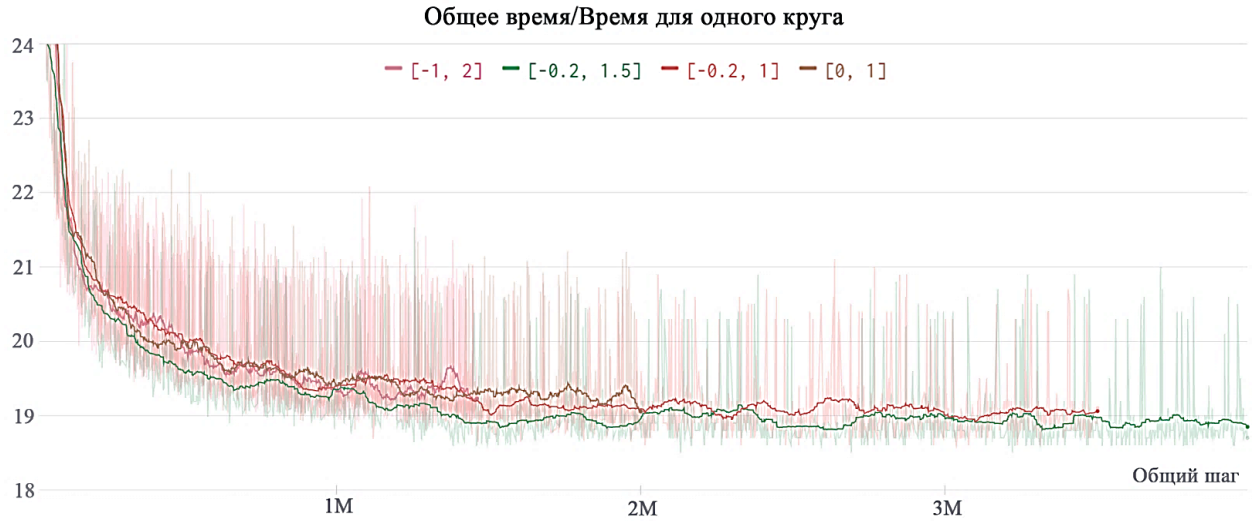


Рис. 9. *График сравнения различных пределов использования скорости*

Fig. 9. *Comparison chart for speed limits*

Агент едет по дороге с конкретно заданной шириной. Если не ограничивать ширину пространства его действия, которое он может использовать для построения маршрута и маневрирования, то он будет просто за счет значений других показателей пытаться решить задачу (рис 10). Т.е. вместо того, чтобы замедлиться для вхождения в поворот на узкой полосе дороги, агент будет постоянно ускоряться для того, чтобы пройти трассу за более короткий промежуток времени. Но задавать максимально узкую полосу для построения маршрутных маневров тоже нельзя. Агент должен научиться проходить трассу максимально коротким, но правильным маршрутом – пройти трассу за наименьшее время и без столкновений. Т.е. агент должен научиться двигаться не только по центру полосы/участка движения, но и сокращать общий путь движения для более быстрого прохождения круга.

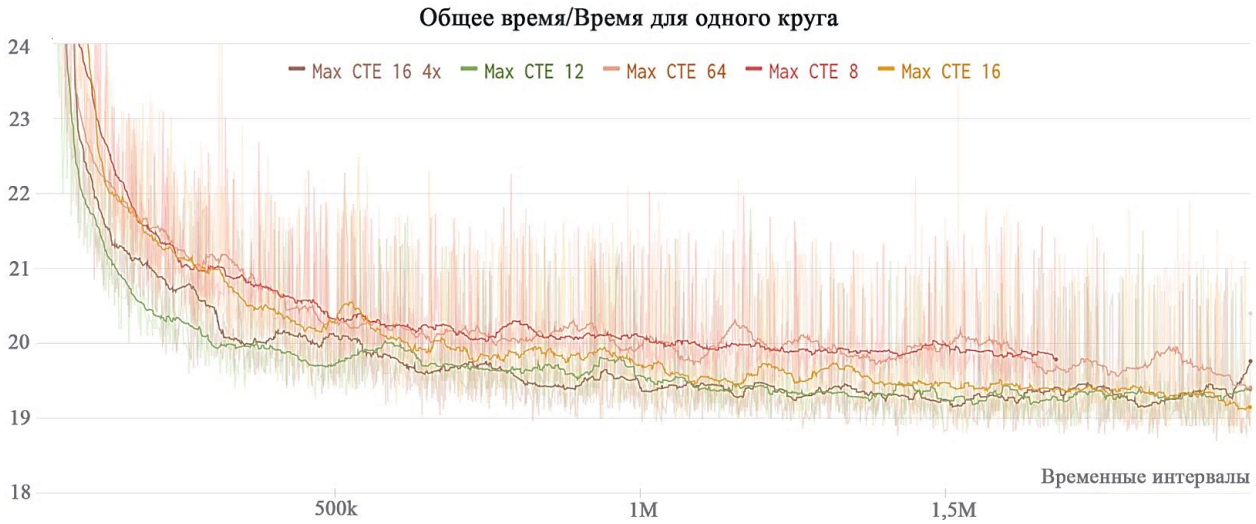


Рис. 10. *График сравнения различной ширины допустимого отклонения*

Fig. 10. *Comparison chart for widths tolerance*

На время прохождения трассы также влияет и непрерывность скорости движения агента (рис. 11). Если у агента стоит задача преодолеть круг за наиболее короткий путь и наиболее короткое время, то ему нужно также научиться минимизировать количество остановок во время движения для экономии этого самого времени. Чем меньше агенту приходиться прерывать свое движение, тем быстрее будет преодолена трасса.

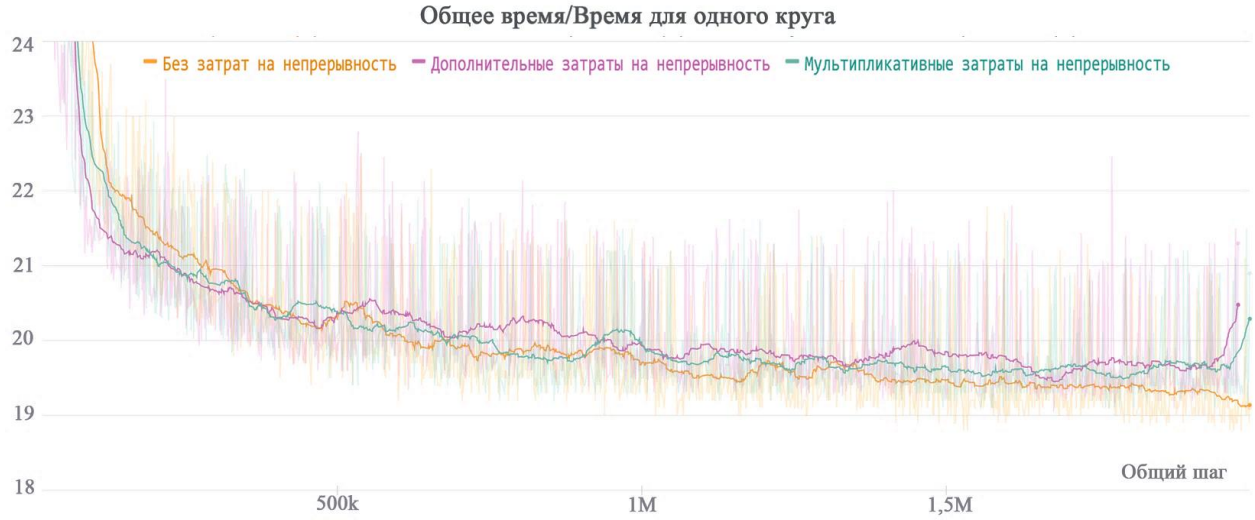


Рис. 11. *График сравнения непрерывного движения и остановок*

Fig. 11. *Comparison chart for continuous movement and stops*

Это значение, при котором агент получает вознаграждение за правильно пройденный круг трассы (рис. 12). Если значение gamma будет составлять 0.98, то агент будет часто получать сиюминутные вознаграждения за правильно пройденный участок трассы на допустимой скорости, но не научиться сокращать и оптимизировать свой путь. Если значение gamma будет составлять 0.995, то агент будет искать идеальные условия и траектории движения для быстрого преодоления трассы, но это увеличит время обучения и количество попыток для идеального преодоления трассы.

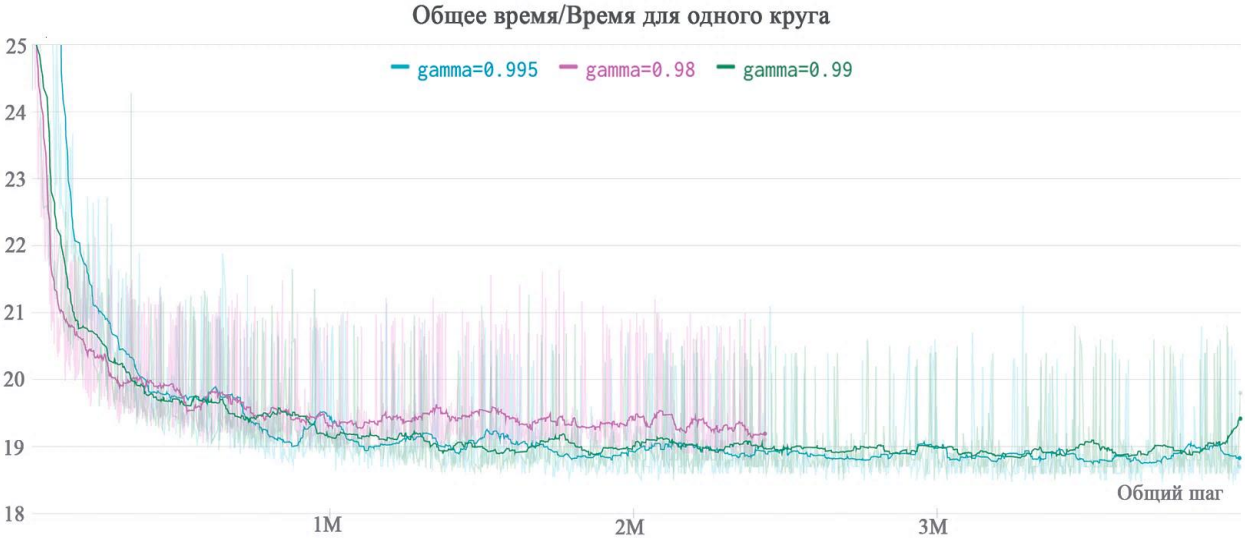


Рис. 12. *График сравнения с различными коэффициентами дисконтирования (gamma)*

Fig. 12. *Comparison chart with different discount factors (gamma)*

Частота отрисовки кадров в секунду также важна для эффективного обучения. Чем стабильней и выше частота кадров в секунду, тем больше шагов проходит агент без совершения ошибок (рис. 13).

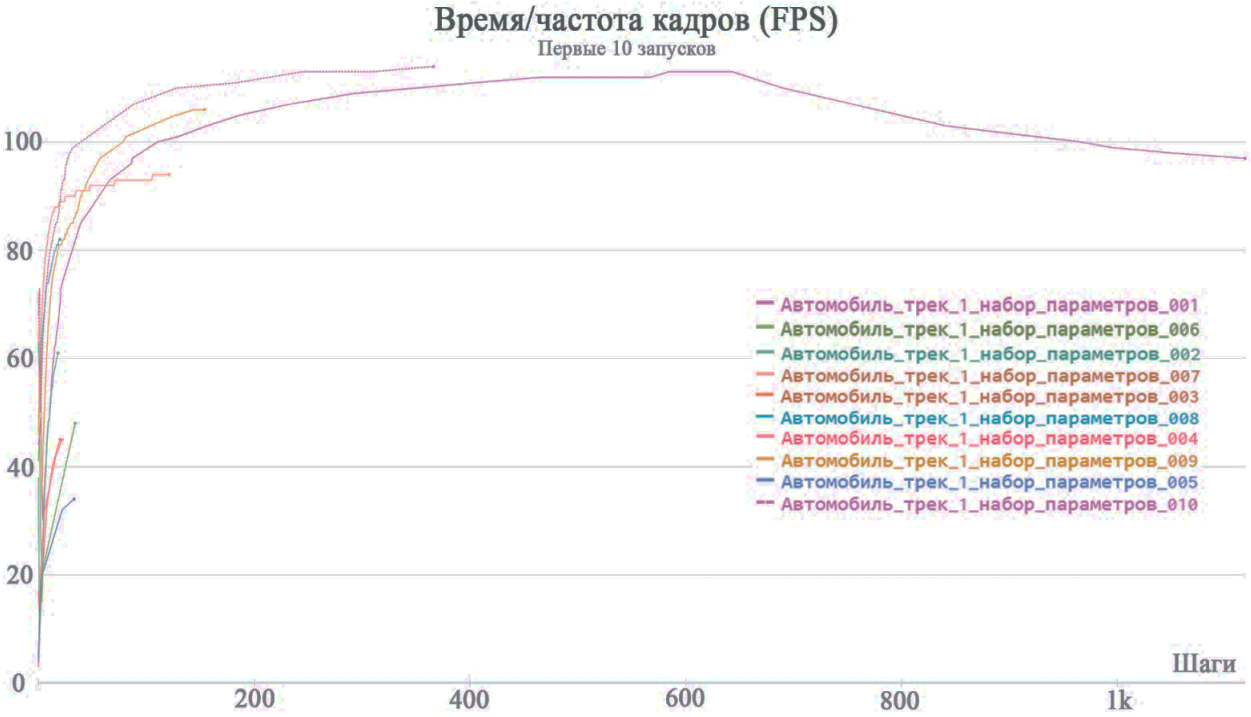


Рис. 13. *График зависимости пройденных шагов от количества кадров в секунду (FPS/s)*

Fig. 13. *Dependence graph between the number of steps traveled and the number of frames per second (FPS/s)*

Выводы

Результаты исследования позволяют сделать выводы о потенциале использования обучения с подкреплением в сфере автономного транспорта, включая необходимость дообучения модели на реальных данных, перспективы масштабирования на транспортные средства различного класса, ограничения, связанные с вычислительными ресурсами и необходимостью безопасной верификации поведения. Как показали эксперименты, модели с более высокой максимальной скоростью движения обучаются дольше. Можно выявить общую зависимость времени адаптации агента к ограничениям – чем больше ограничений, которые воздействуют на автомобиль, тем лучше модель адаптируется к среде. Метод актер-критик может быть использован в обучении, но не так быстро, как бы того хотелось. На обучение одной только этой модели ушло 125 часов.

СПИСОК ЛИТЕРАТУРЫ / REFERENCES

1. *Сыркин И. С., Дубинкин Д. М., Юнусов И. Ф., Ушаков А. Е.* Системы управления автономного карьерного самосвала // Россия молодая: сб. материалов XIV Всероссийской науч.-практ. конф. с междунар. участием, Кемерово, 19–21 апр. 2022 г. Кемерово: Кузбасский гос. техн. ун-т им. Т. Ф. Горбачева, 2022. С. 420071–420078. EDN: CXHGOK

Syrkin I.S., Dubinkin D.M., Yunusov I.F., Ushakov A.E. Control systems of autonomous mining dump trucks. *Young Russia: Proceedings of the XIV All-Russian Scientific and Practical Conference with International Participation*, Kemerovo, April 19–21, 2022. Kemerovo: T.F. Gorbachev Kuzbass State Technical University, 2022. Pp. 420071–420078. EDN: CXHGOK. (In Russian)

2. Toromanoff M., Wirbel E., Moutarde F. End-to-end model-free reinforcement learning for urban driving using implicit affordances. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020. С. 7151–7160. DOI: 10.1109/CVPR42600.2020.00718

3. Sauer A., Savinov N., Geiger A. Conditional affordance learning for driving in urban environments. *Proceedings of the Conference on Robot Learning (CoRL).* 2018. DOI: 10.48550/arXiv.1806.06498

4. *Шереужев М. А., У Го, Серебренный В. В.* Модификация алгоритма глубокого обучения для распределения функций и задач между робототехническим комплексом и человеком в условиях неопределенности и переменности окружающей среды // Известия Кабардино-Балкарского научного центра РАН. 2024. Т. 26. № 6. С. 208–218. DOI: 10.35330/1991-6639-2024-26-6-208-218.

Shereuzhev M.A., U Go, Serebrenny V.V. Modification of a deep learning algorithm for the distribution of functions and tasks between a robotic system and a human under conditions of uncertainty and environmental variability. *News of the Kabardino-Balkarian Scientific Center of RAS*. 2024. Vol. 26. No. 6. P. 208–218. DOI: 10.35330/1991-6639-2024-26-6-208-218. (In Russian)

5. Tampuu A., Semikin M., Muhammad N. et al. Survey of end-to-end driving: Architectures and training methods: arXiv preprint arXiv:2003.06404. 2020.

6. *Лютикова Л. А.* Применение метода машинного обучения для анализа неполных данных // Известия Кабардино-Балкарского научного центра РАН. 2024. Т. 26. № 6. С. 139–145. DOI: 10.35330/1991-6639-2024-26-6-139-145.

Lyutikova L.A. Application of a machine learning method for the analysis of incomplete data. *News of the Kabardino-Balkarian Scientific Center of RAS.* 2024. Vol. 26. No. 6. Pp. 139–145. DOI: 10.35330/1991-6639-2024-26-6-139-145. (In Russian)

7. *Шереужев М. А., Арабаджиев Д. И., Семянников И. В.* Моделирование алгоритма предотвращения столкновений в робототехнических коллаборативных системах // Известия Кабардино-Балкарского научного центра РАН. 2024. Т. 26. № 6. С. 67–81. DOI: 10.35330/1991-6639-2024-26-6-67-81.

Shereuzhev M.A., Arabadzhiev D.I., Semyannikov I.V. Modeling of a collision avoidance algorithm in collaborative robotic systems. *News of the Kabardino-Balkarian Scientific Center of RAS*. 2024. Vol. 26. No. 6. Pp. 67–81. DOI: 10.35330/1991-6639-2024-26-6-67-81. (In Russian)

8. He K., Zhang X., Ren S., Sun J. Deep residual learning for image recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).* Las Vegas (NV), 2016. Pp. 770–778. DOI: 10.1109/CVPR.2016.90

9. *Петренко В. И.* Классификация задач мультиагентного обучения с подкреплением // Известия Кабардино-Балкарского научного центра РАН. 2021. Т. 3. № 101. С. 32–44. DOI: 10.35330/1991-6639-2021-3-101-32-44.

Petrenko V.I. Classification of multi-agent reinforcement learning tasks. *News of the Kabardino-Balkarian Scientific Center of RAS*. 2021. Vol. 3. No. 101. Pp. 32–44. DOI: 10.35330/1991-6639-2021-3-101-32-44. (In Russian)

10. *Коул А., Ганджу С., Казам М.* Искусственный интеллект и компьютерное зрение: реальные проекты на Python, Keras и TensorFlow. Санкт-Петербург: Питер, 2019. 356 с. ISBN: 978-1-492-04305-0.

Cole A., Gandju S., Kazam M. *Iskusstvennyy intellekt i komp'yuternoye zreniye: real'nyye proyekty na Python, Keras i TensorFlow* [Artificial intelligence and computer vision: Real projects using Python, Keras, and TensorFlow]. St. Petersburg: Piter, 2019. 356 p. ISBN: 978-1-492-04305-0. (In Russian)

11. *Ушаков А. Е., Стебулянин М. М.* Исследование параметров обучения модели для системы управления курсом движения // Интернаука: электронный научный журнал. 2025. № 1-3(365). С. 53–57. EDN: OXPGLQ

Ushakov A.E., Stebulyanin M.M. Study of model training parameters for a course control system. *Internauka*. 2025. No. 1-3(365). Pp. 53–57. EDN: OXPGLQ. (In Russian)

12. *Ушаков А. Е.* Использование симулятора для исследования технологий автономного движения // Российская наука в современном мире: сборник статей LXVII международной научно-практической конференции, Москва, 15 января 2025 г. М.: Актуальность. РФ, 2025. С. 155–158. EDN: JFUWYO

Ushakov A.E. Using a simulator to study autonomous driving technologies. *Russian Science in the Modern World: Proceedings of the LXVII International Scientific and Practical Conference.* Moscow, January 15, 2025. Moscow: Aktualnost. RF, 2025. Pp. 155–158. EDN: JFUWYO. (In Russian)

13. Sutton R.S., Barto A.G. Reinforcement Learning: An Introduction. Cambridge, MA: MIT Press, 2018. 552 c.

14. Liang X., Wang T., Yang L., Xing E. CIRL: Controllable imitative reinforcement learning for vision-based self-driving. *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018. DOI: 10.48550/arXiv.1807.03776

15. *Шереужев М. А., Шереужев М. А., Кишев А. Ю.* Вопросы выбора системы технического зрения сельскохозяйственных робототехнических комплексов для контроля сорной растительности // Известия Кабардино-Балкарского научного центра РАН. 2022. № 4(108). С. 84–95. DOI: 10.35330/1991-6639-2022-4-108-84-95

Shereuzhev M.A., Shereuzhev M.A., Kishev A.Yu. Issues of selecting a machine vision system for agricultural robotic complexes for weed control. *News of the Kabardino-Balkarian Scientific Center of RAS.* 2022. No. 4(108). Pp. 84–95. DOI: 10.35330/1991-6639-2022-4-108-84-95. (In Russian)

16. Chen D., Zhou B., Koltun V., Krähenbühl P. Learning by Cheating: arXiv preprint arXiv:1912.12294. 2019

Конфликт интересов: Авторы заявляют об отсутствии конфликта интересов.

Conflict of interest: The authors declare no conflicts of interests.

Вклад авторов:

Ушаков А. Е. – постановка эксперимента, написание выводов статьи,
Стебулянин М. М. – разработка концепта статьи,
Шереужев М. А. – постановка проблемы,
Девяткин Ф. Д. – подготовка списка литературы, оформление рисунков и таблиц.

Contribution of the authors:

Ushakov A.E. – experimental setup, writing the paper's conclusions,
Stebulyanin M.M. – article conceptualization,
Shereuzhev M.A. – problem statement,
Devyatkin F.D. – preparation of the reference list and the design of figures and tables.

Финансирование. Работа выполнена при финансовой поддержке Министерства науки и высшего образования Российской Федерации в рамках темы госзадания 2024–2026 (проект FSFS-2024-0012).

Funding. This work was financially supported by the Ministry of Science and Higher Education of the Russian Federation within the framework of the state task 2024–2026 (project FSFS-2024-0012).

Информация об авторах

Ушаков Александр Евгеньевич, аспирант, инженер-исследователь кафедры «Роботехника и мехатроника», Московский государственный технологический университет «СТАНКИН»;
127055, Россия, Москва, Вадковский пер., 1;
ushakov_ae@internet.ru, ORCID: https://orcid.org/0009-0006-1467-5043, SPIN-код: 5174-7378

Стебулянин Михаил Михайлович, д-р техн. наук, профессор, заведующий кафедрой «Роботехника и мехатроника», Московский государственный технологический университет «СТАНКИН»;
127055, Россия, Москва, Вадковский пер., 1;
mmsteb@rambler.ru, ORCID: https://orcid.org/0009-0007-3443-0593, SPIN-код: 4389-1120

Шереужев Мадин Артурович, кан. тех. наук, доцент кафедры «Роботехника и мехатроника», Московский государственный технологический университет «СТАНКИН»;
127055, Россия, Москва, Вадковский пер., 1;
shereuzhev@gmail.com, ORCID: https://orcid.org/0000-0003-2352-992X, SPIN-код: 1734-9056

Девяткин Федор Владимирович, аспирант кафедры СМ7 «Робототехнические системы и мехатроника», Московский государственный технический университет имени Н. Э. Баумана;
105005, Россия, Москва, 2-я Бауманская улица, 5;
инженер, Московский государственный технологический университет «СТАНКИН»;
127055, Россия, Москва, Вадковский пер., 1;
feodor-dev@ya.ru, ORCID: https://orcid.org/0009-0000-2639-9521, SPIN-код: 7738-5724

Information about the authors

Alexander E. Ushakov, Postgraduate student, Research Engineer, Department of Robotics and Mechatronics, Moscow State University of Technology “STANKIN”;
127055, Russia, Moscow, 1 Vadkovsky lane;
ushakov_ae@internet.ru, ORCID: https://orcid.org/0009-0006-1467-5043, SPIN-code: 5174-7378

Mikhail M. Stebulyanin, Doctor of Technical Sciences, Professor, Head of the Department of Robotics and Mechatronics, Moscow State University of Technology “STANKIN”;
127055, Russia, Moscow, 1 Vadkovsky lane;
mmsteb@rambler.ru, ORCID: https://orcid.org/0009-0007-3443-0593, SPIN-code: 4389-1120

Madin A. Shereuzhev, Candidate of Engineering Sciences, Associate Professor at the Department of Robotics and Mechatronics, Moscow State University of Technology “STANKIN”;
127055, Russia, Moscow, 1 Vadkovsky lane;
shereuzhev@gmail.com, ORCID: https://orcid.org/0000-0003-2352-992X, SPIN-code: 1734-9056

Fedor V. Devyatkin, Postgraduate student at the Department of ME7 “Robotic Systems and Mechatronics”, The Bauman Moscow State Technical University;
105005, Russia, Moscow, 5, 2-nd Baumanskaya street;
Engineer, Moscow State University of Technology “STANKIN”;
127055, Russia, Moscow, 1 Vadkovsky lane;
feodor-dev@ya.ru, ORCID: https://orcid.org/0009-0000-2639-9521, SPIN-code: 7738-5724