

УДК 004.85

DOI: 10.18413/2518-1092-2024-9-3-0-7

**Тихонов М.К.
Непомнящий Д.О.
Хайдукова В.Н.****МЕТОД ОБУЧЕНИЯ ИНТЕЛЛЕКТУАЛЬНОГО АГЕНТА
С ПОМОЩЬЮ СЕТЕЙ DOUBLE DQN, ПУТЕВЫХ
ТОЧЕК И ФУНКЦИИ ВОЗНАГРАЖДЕНИЯ**Институт космических и информационных технологий СФУ,
ул. Академика Киренского, 26Б, Красноярск, Красноярский край, 660074, Россия*e-mail: samualgame@gmail.com***Аннотация**

Рассмотрены задачи повышения эффективности управления автономных транспортных средств. Выделена проблема снижения требуемых вычислительных ресурсов для интеллектуального модуля управления автомобилем. Предложен алгоритм обучения нейронной сети для архитектуры Double DQN с модифицированной функцией вознаграждения. Основой предлагаемого решения является использование сегментации полосы движения, функции вознаграждения и использования дополнительных путевых точек при обучении. Разработана программная модель и выполнено моделирование процесса обучения. Полученные результаты сравнительного анализа с известными решениями показывают стабильное повышение длительности эпизода, и эффективное обучение в реалистичной городской симуляции. Исследование указывает на возможность уменьшения необходимости в высокой вычислительной мощности, что даст возможность использовать центральные процессоры (CPU) для основных функций беспилотных автомобилей вместо графических процессоров (GPU).

Ключевые слова: беспилотный автомобиль; полоса движения; интеллектуальное управление; агент; нейронная сеть; глубокое Q-обучение; симуляция

Для цитирования: Тихонов М.К., Непомнящий Д.О., Хайдукова В.Н. Метод обучения интеллектуального агента с помощью сетей Double DQN, путевых точек и функции вознаграждения // Научный результат. Информационные технологии. – Т.9, №3, 2024. – С. 63-72. DOI: 10.18413/2518-1092-2024-9-3-0-7

**Tikhonov M.K.
Nepomnyashchiy D.O.
Khaidukova V.N.****A METHOD FOR TRAINING AN INTELLIGENT
AGENT USING DOUBLE DQN NETWORKS,
WAYPOINTS AND REWARD FUNCTION**Institute of Space and Information Technologies SFU,
26B Academician Kirenskiy st., Krasnoyarsk, 660074, Russia*e-mail: samualgame@gmail.com***Abstract**

The problems of increasing the control efficiency of autonomous vehicles are considered. The problem of reducing the required computational resources for the intelligent vehicle control module is highlighted. A neural network training algorithm for Double DQN architecture with modified reward functions is proposed. The basis of the proposed solution is the use of lane segmentation, reward function and the use of additional waypoints in training. A software model has been developed and simulation of the learning process has been performed. The results obtained from a comparative analysis with known solutions show a stable increase in episode duration, and effective training in a realistic urban simulation. The study points to the possibility of reducing the need for high computing power, which will enable the use of central processing units (CPUs) for basic functions of unmanned vehicles instead of graphics processing units (GPUs).

Keywords: self-driving car; lane following; intelligent control; agent; neural network; deep Q-learning; simulation

For citation: Tikhonov M.K., Nepomnyashchiy D.O., Khaidukova V.N. A method for training an intelligent agent using Double DQN networks, waypoints and reward function // Research result. Information technologies. – Т. 9, №3, 2024. – Р. 63-72. DOI: 10.18413/2518-1092-2024-9-3-0-7

ВВЕДЕНИЕ

Развитие технологий автономного управления транспортными средствами является одним из важнейших направлений современных исследований в области искусственного интеллекта и робототехники. Несмотря на значительный прогресс в этой области одной из ключевых проблем является разработка эффективных алгоритмов управления беспилотными автомобилями, для вождения в реальных дорожных условиях.

Эти задачи решаются, в том числе, путем снижения времени обучения, что при функционировании в режиме реального времени, в реальных дорожных условиях является критичным. То есть интеллектуальный агент, как средство управления автомобилем, должен обладать восприятием окружающей среды и реакцией на аварийные ситуации не только в соответствии с правилами дорожного движения, но и федеральным стандартам безопасности [1].

Согласно исследованиям RAND [2] для достижения требуемого уровня точности, соизмеримого с водителем-человеком, интеллектуальная система управления беспилотным автомобилем должна пройти обучение в эквиваленте пройденного расстояния в 11 миллиардов тестовых миль [3].

Достижение этих параметров требует значительных временных затрат не только на этапах предварительного обучения, но и в процессе функционирования при условии самообучения сети [4].

Кроме того, для повышения эффективности принятия решений и обеспечения работы в режиме реального времени требуется обеспечить снижение вычислительных ресурсов, поскольку использование сверхмощных специализированных вычислителей на борту автомобиля является экономически и технически нецелесообразным [5].

1. СВЯЗАННЫЕ РАБОТЫ

Известен ряд подходов для решения обозначенных задач. Например, в работах «Rodrigues P., Vieira – IEEE, 2020» [6] и «Chishti S. O. A. et al. – IEEE, 2018». [7] были предложены подходы по созданию интеллектуальных агентов для управления электротранспортом на основе сверточных нейронных сетей и сетей глубокого Q-обучения, которые показали многообещающие результаты при моделировании. Однако предложенные агенты требовали значительных временных затрат при обучении. Такой эффект, по-видимому, обусловлен накоплением нерелевантных данных, а также высокой размерностью пространства состояний, что значительно снижало эффективность обучения, это подтверждается данными статьями «Chhikara P. et al. Practice and Experience. – 2022» [8] и «Malekloo A. et al. //Structural Health Monitoring. – 2022» [9].

Особого внимания заслуживает архитектура глубоких Q-сетей, рассматриваемая в работе «Peng B. et al. //Automotive Innovation. – 2021» [10], которая сочетает в себе алгоритмы Dueling DQN и Double DQN [11]. Данный подход показывает, что является эффективным инструментом для реализации автономного управления и не зависит от большого количества размеченных данных. Однако эффективность такого алгоритма тестировалась только на гоночных треках, а результаты отработки в сложных городских симуляциях не известны.

При этом в работах «Kiran B. R. et al. –IEEE, 2021» [12] и «Lin Q. et al. – IEEE, 2020» [13] показано, что использование путевых точек в качестве вспомогательных целей при обучении модели вождения сокращает время сходимости модели и повышает безопасность за счет дополнительных навигационных подсказок.

Таким образом, можно предположить, что использованием комбинированного подхода на основе путевых точек и сетей Double DQN может обеспечить более низкие технические требования к аппаратуре вычислителя и повысить скорость обучения агента.

2. ПРЕДЛАГАЕМОЕ РЕШЕНИЕ

Предлагаемый подход заключается в следующем:

1. Используется нейронная сеть Double DQN и оптимизатор Adam для реализации автономного вождения.

2. PyTorch используется в качестве фреймворка для реализации глубокого обучения с подкреплением, позволяющего тренировать нейронные сети без необходимости в большом объеме размеченных данных.

3. Обновление весов происходит методом градиентного спуска.

4. Используются путевые точки при обучении в качестве вспомогательной модели для «поощрения» автомобиля при удерживании в центре полосы движения.

5. Функция вознаграждения при обучении состоит из трех частей: поощрения за наименьшую разность между углом ориентации автомобиля и углом ориентации ближайшей путевой точки на дорожной полосе, расстояние между автомобилем и центром полосы движения и факта столкновения или его отсутствия.

На рисунке 1 представлена общая схема обучения интеллектуального агента для системы управления автомобилем.

Где s – набор состояний (продольная скорость, поперечная скорость, обороты двигателя и скорости вращения всех четырех колес), s' – новое состояние, a – набор действий, r – функция вознаграждения, θ – угол ориентации автомобиля и углом ориентации ближайшей путевой точки на дорожной полосе, Q – значение действия состояния, J – функция потерь, ∇ – градиент.

Для реализации автономного вождения используется нейронная сеть Double DQN, которая обучается с помощью алгоритма глубокого обучения с подкреплением, это позволяет ей обучаться в процессе взаимодействия с моделируемой средой. При таком подходе не требуется значительное количество размеченных данных.

В процессе обучения устанавливаются путевые точки, побуждающие модель автомобиля придерживаться центра полосы движения. Это реализуется с помощью специализированной функции вознаграждения, которая предусматривает поощрения за правильное вождение и штрафы за отклонения. Многократно взаимодействуя с окружающей средой, выполняя действия и получая вознаграждения, нейронная сеть оптимизирует свои параметры для выбора оптимальных действий по удержанию автомобиля в центре полосы движения.

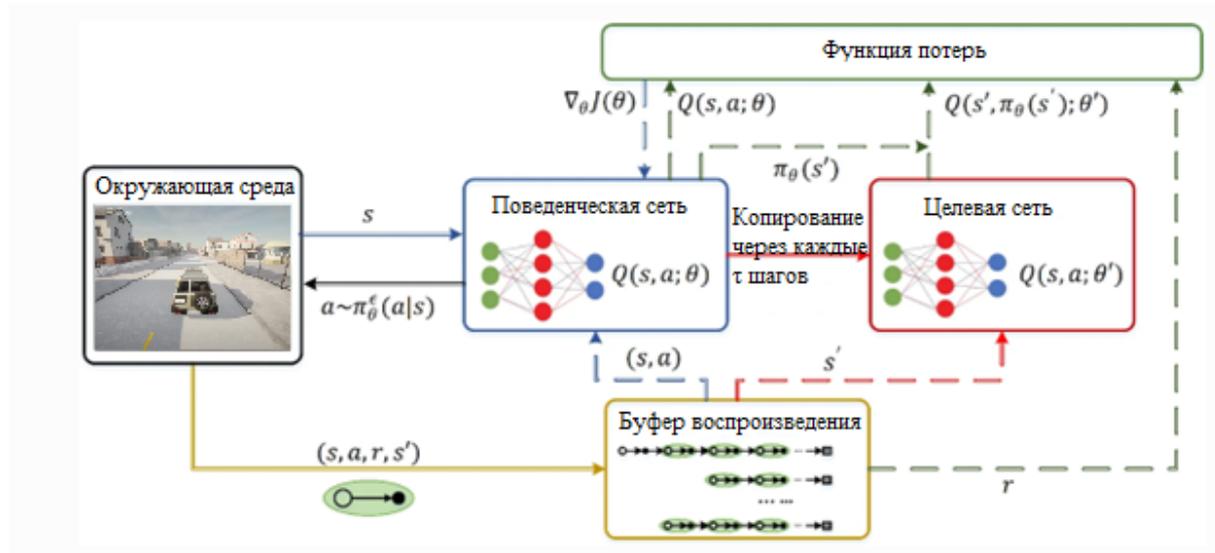


Рис. 1. Схема системы предварительного и адаптивного обучения интеллектуального агента управления

Fig. 1. Scheme of the system of preliminary and adaptive training of an intelligent control agent

При расчете функции вознаграждения предложено учитывать три ключевых фактора – это разность между углом ориентации автомобиля и углом ориентации ближайшей путевой точки на дорожной полосе, обозначаемая θ , расстояние между автомобилем и центром полосы движения, обозначаемое d и факт столкновения или его отсутствие, обозначаемое как c . При этом полная функция вознаграждения определится как:

$$r = \lambda_1 * \cos * \theta - \lambda_2 * d - \lambda_3 * c \quad (1)$$

Где $\lambda_1, \lambda_2, \lambda_3$ – неотрицательные весовые коэффициенты.

Отметим, что угол θ вычисляется следующим образом:

$$\theta = (V - W) * \pi / 180 \quad (2)$$

Где: V – угол ориентации автомобиля, W – угол ориентации ближайшей путевой точки на дорожной полосе.

Меньший угол означает, что автомобиль строго следует по центру полосы движения в верном направлении.

Окружающая среда для модели представляет собой как показания датчиков, так и комплект изображений, которые требуют подготовки для обработки нейронной сетью. Предлагается реализация пошагового процесса предварительной обработки изображений, который включает преобразование в градации серого, уменьшение размера, корректировку диапазона и, в конечном итоге, нормализацию для обеспечения оптимальной работы нейронной сети (Рисунок 2).

На первом шаге происходит преобразование изображений в градации серого, при котором удаляется не нужная для расчетов цветовая информация, но при этом сохраняются требуемые детали. Это позволяет упростить представление изображения для снижения вычислительной сложности. Принцип преобразования заключается в усреднении красного, зеленого и синего каналов изображения.

Далее для нормализации и формирования постоянного размера входных данных для нейронной сети происходит уменьшение изображений до размера 128x128 пикселей. Это позволяет не только обеспечить фиксированный размер данных, но и снизить требования к вычислительным мощностям.

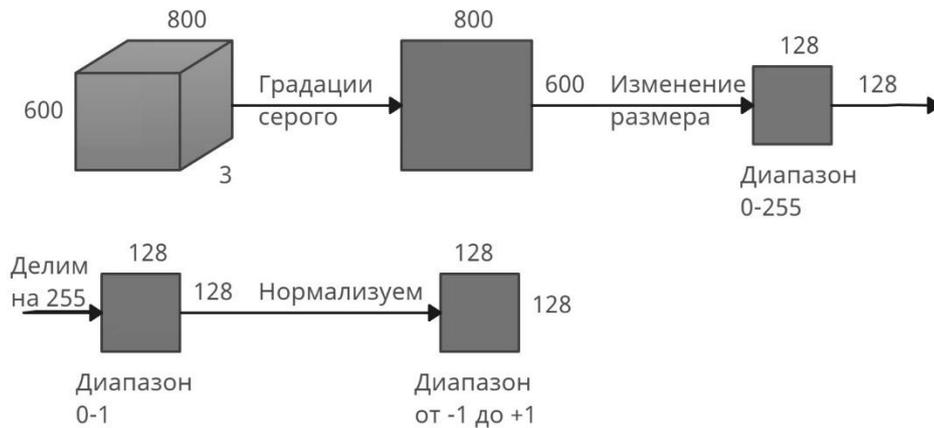


Рис. 2. Обработка изображения
Fig. 2. Image processing

Следующее преобразование заключается в приведении значений пикселей к бинарному диапазону от 0 до 1. Механизм приведения заключается в делении каждого значения пикселя на 255.

Для дальнейшего повышения производительности и устойчивости сети предлагается выполнять операцию нормализации данных изображения. Нормализация приводит значения пикселей к среднему значению 0 и стандартному отклонению 0,5. Этот шаг обеспечивает сбалансированный диапазон входных данных, сосредоточенный вокруг нуля, и обеспечивает предотвращение проблемы насыщения или перегрузки в процессе обучения. Таким образом, обеспечивается стандартное представление входных данных для нейронной сети и приведение их к диапазону от -1 до +1.

Входной слой нейронной сети состоит из двух частей: изображения с камеры размером 128x128 пикселей в оттенках серого и 7-мерного вектора, описывающего состояние автомобиля.

В свою очередь сверточная часть состоит из 3 слоев:

- 1-й сверточный слой: 32 фильтра размером 8x8 с шагом 4;
- 2-й сверточный слой: 64 фильтра размера 4x4 с шагом 2;
- 3-й сверточный слой: 64 фильтра размера 3x3 с шагом 1.

Выход сверточных слоев преобразуется в вектор размера 15488 и конкатенируется с 7-мерным вектором скорости, в результате чего получается вектор размера 15495.

Далее следуют полносвязные слои, построенные по архитектуре «dueling network» включающие [14]:

- Поток создания ценности: 3 полносвязных слоя с 128, 32 и 1 нейроном соответственно.
- Поток преимущества: 3 полносвязных слоя с 128, 32 и N нейронами, где N – количество возможных действий (для данного случая 13).

Определены следующие гиперпараметры:

- Размер партии (batch size) – 32;
- Коэффициент дисконтирования (discount factor) – $\gamma = 0.9$;
- Скорость обучения (learning rate) – $\alpha = 0.0005$;
- Размер буфера воспроизведения (replay buffer size) – 10,000;
- Вероятность случайного действия ϵ -greedy – $\epsilon = 0.1$.

Функция вознаграждения из формулы 1 используется для обновления Q-функции в процессе обучения. В частности, после того как агент выполняет действие a в состоянии s и получает вознаграждение r , Q-функция обновляется с использованием следующего правила обновления: (3):

$$Q(s, a) \leftarrow Q(s, a) + \alpha * (r + \gamma * \max Q(s', a') - Q(s, a)) \quad (3)$$

где:

α – скорость обучения (learning rate),

γ – коэффициент дисконтирования (discount factor),

s' – новое состояние после выполнения действия a ,

$\max Q(s', a')$ – максимальное значение Q -функции для нового состояния s' по всем возможным действиям a' .

Таким образом, процесс обучения с подкреплением, в котором используются формулы 1 и 3, позволяет агенту находить стратегию поведения, которая максимизирует суммарное ожидаемое вознаграждение в долгосрочной перспективе. Это достигается путем постоянного взаимодействия с окружающей средой и обновления Q -функции в соответствии с получаемыми вознаграждениями и наблюдаемыми изменениями в среде.

3. ПОЛУЧЕННЫЕ РЕЗУЛЬТАТЫ

Для создания среды моделирования интеллектуального агента управления автомобилем использовался симулятор Carla [15]. В симуляторе имеется возможность построения траектории в виде путевых точек, расположенных по центру полосы движения и на ключевых участках дороги, таких как перекрестки или повороты. Среди множества известных, выбор данного симулятора обусловлен следующим: открытый исходный код для реализации собственного функционала, детализированные 3D-модели городов, пешеходов, транспортных средств и дорожных знаков, для создания моделей с реалистичными сценариями вождения, наличие встроенных датчиков – камер, лидаров, радаров, для генерации реалистичных сенсорных данных в процессе обучения нейронных сетей.

При функционировании модели, изображения с камеры синхронно обрабатываются вместе с вектором текущей скорости автомобиля и формируют многомерный набор состояний текущего этапа. Этот набор поступает на вход нейронной сети, которая обучается в автономном режиме методом глубокого обучения с подкреплением. Нейронная сеть выдает управляющие команды, обеспечивающие движение автомобиля вдоль опорной траектории [16]. При неудаче, т.е. отклонении от траектории или столкновении, сценарий начинается заново для накопления статистики и дальнейшего обучения нейросети.

3.1 Симуляция функции вознаграждения

На рисунке 3а показана симуляция в CARLA при расчете разности между углом ориентации автомобиля и углом ориентации ближайшей путевой точки на дорожной полосе – θ .



Рис. 3. Симуляция при расчете функции вознаграждения а) Расчет разности между углом ориентации автомобиля и углом ориентации ближайшей путевой точки на дорожной полосе, б) Расчет положения в полосе движения

Fig. 3. Simulation during the calculation of the reward function а) Calculation of the difference between the vehicle orientation angle and the orientation angle of the nearest waypoint on the roadway, б) Calculation of the position in the traffic lane

При моделировании движения автомобиля этот угол обеспечивался минимальным, поскольку при меньшем угле снижается риск потери управления или съезда с дороги.

Моделирование показало, что благодаря близкому расположению к центру полосы движения, автомобиль сохраняет более стабильную траекторию движения, что сводит к минимуму вероятность заноса или скольжения. Отметим, что при реальном вождении это особенно важно в сложных условиях движения, таких как резкие повороты, мокрые или обледенелые поверхности, неровности рельефа.

Кроме того, моделирование показало, что минимизация угла уменьшает боковые силы, действующие на автомобиль. При большом угле θ возникает большая разница между направлением движения и касательной к дороге, что приводит к увеличению боковых сил. При реальном вождении эти силы могут негативно повлиять на устойчивость, тяговое сопротивление и топливную экономичность автомобиля. Так же при минимизации θ уменьшаются боковые силы, что приводит к более плавному и контролируемому движению автомобиля.

Еще одним важным свойством обеспечения минимального угла является улучшение «отзывчивости» и маневренности автомобиля, что при реальном вождении облегчает прохождение сложных участков дороги или объезд препятствий. Кроме того, это позволяет более точно регулировать рулевое управление и ускорение, что сокращает время реакции на изменение дорожных условий или возникновение нештатных ситуаций.

На рисунке 3б показаны результаты симуляции движения по путевым точкам, которые задают приоритетные участки дороги и центр полосы движения.

Моделирование показало, что траектория движения строго соответствует центру полосы, что демонстрирует эффективность предложенного к использованию метода «путевых точек» для обучения модели и автономного вождения. Таким образом, это позволяет автомобилю оставаться в «своей» полосе в соответствии с дорожной разметкой.

В таблице приведены результаты сравнительного анализа полученных решений с рядом известных подходов при обучении агента.

Таблица

Результаты сравнения предложенного решения с известными

Table

Results of comparison of the proposed solution with known solutions

Используемые технологии	Кол-во эпизодов	Максимальная продолжительность эпизода, с	Средняя продолжительность эпизода, с	Окружение	Оборудование
Известное решение на основе технологии DQN, свёрточной нейронной сети и стекового фрейма [17]	2000	131	55	CARLA Simulator - Реалистичная 3d симуляция города	GPU: NVIDIA GeForce RTX 3080
Предлагаемое решение на основе технологии Double DQN, сверточных нейронных сетей, путевых точек и функции вознаграждения	5000	240	80	CARLA Simulator - Реалистичная 3d симуляция города	CPU: Xeon E5-2696 v4

Как видно из таблицы, предложенный подход выигрывает как по максимальной продолжительности эпизода, так и по средней. В сравниваемой работе с увеличением эпизодов

сокращалось время его выполнения. Самые лучшие результаты были достигнуты в первых 225 эпизодах. Автор сравниваемой работы предполагает, что это связано с слишком "жестким" детерминированным обновлением целевой сети и возможной неоптимальностью выбора алгоритма DQN для данной задачи. При моделировании в работе автора отмечалось, что с увеличением времени обучения повышалась продолжительность эпизода. В отличие от сравниваемой работы на начальном этапе продолжительность эпизода составляла не более 10 секунд, но со временем постепенно увеличивалась и на момент 5000 эпизода, рост продолжительности эпизода не остановился.

ЗАКЛЮЧЕНИЕ

Анализ предметной области и известных подходов к решению проблем обучения и подготовки к вождению интеллектуальных систем управления автономных транспортных средств показал актуальность поиска решений, которые могли бы эффективно выполнять обучения агента без использования более вычислительной аппаратуры, которая требует дополнительных денежных затрат. Отмечено, что известные подходы, в том числе и методы глубокого обучения, такие как Dueling DQN, позволяют эффективно решать задачу обучения автономному вождению не только в упрощенных условиях, например на закрытых гоночных треках. Но и в реалистичных 3D городских симуляциях

Предложен подход к обучению интеллектуального агента с использованием изменённой функции вознаграждения из работы [17], метода Double DQN, сверточных нейронных сетей, путевых точек и функции вознаграждения, предложенная комбинация методов позволяющий эффективно реализовать обучение в городском симуляторе, даже на CPU.

Результаты моделирования так же позволили отметить, что обучение в реалистичной городской среде происходит значительно сложнее, чем в некоторых из связанных работ, в которых обучение происходило на гоночных треках.

На основании полученных данных можно сделать вывод, что предложенное решение, демонстрирует лучшие результаты в симуляторе CARLA по сравнению с решением, основанным на технологии DQN с сверточной нейронной сетью и стековым фреймом.

Ключевые моменты, подтверждающие преимущество предложенного решения:

1. Увеличение максимальной продолжительности эпизода: со 131 секунды до 240 секунд, что может свидетельствовать о более стабильном и длительном взаимодействии с окружающей средой в рамках одного эпизода.
2. Увеличение средней продолжительности эпизода: с 55 секунд до 80 секунд, что говорит о повышении общей эффективности обучения агента.
3. Продолжительность эпизодов увеличивается со временем, в отличие от сравниваемой работы, где наилучшие результаты наблюдались в начале обучения, а дальнейшее увеличение количества эпизодов приводило к сокращению времени продолжительности эпизода. Это может свидетельствовать о более эффективном обучении и адаптации агента к сложным условиям моделирования в предлагаемом решении.
4. Увеличение длительности эпизодов не прекращается после 5000 эпизодов, что может свидетельствовать о возможности дальнейшего улучшения производительности при увеличении времени обучения.

Также стоит отметить, что предлагаемое решение использует центральный процессор (CPU: Xeon E5-2696 v4), в то время как сравниваемое решение использует графический процессор (GPU: NVIDIA GeForce RTX 3080). Это может свидетельствовать о том, что предлагаемое решение может быть более доступным с точки зрения аппаратного обеспечения, хотя при полном анализе необходимо учитывать и другие факторы, такие как время обучения и энергоэффективность, которые не были описаны автором сравниваемой статьи.

В целом, полученные данные свидетельствуют о том, что предлагаемое решение обладает лучшей производительностью и потенциалом масштабируемости, что делает его

предпочтительным выбором для задач моделирования в симуляторе CARLA. Это открывает перспективы для снижения требований к вычислительным мощностям и, как следствие, упрощению аппаратных систем из состава интеллектуального агента.

Список литературы

1. Final Rule Occupant Protection Amendment Automated Vehicles [Электронный ресурс]. URL: <https://www.nhtsa.gov/document/final-rule-occupant-protection-amendment-automated-vehicles> (дата обращения 10.09.2023).
2. RAND Corporation [Электронный ресурс]. URL: <https://www.rand.org/> (дата обращения 23.09.2023).
3. Malayjerdi M. et al. Virtual simulations environment development for autonomous vehicles interaction // ASME International Mechanical Engineering Congress and Exposition. – American Society of Mechanical Engineers, 2020. – Т. 84492. – С. V02BT02A009.
4. Young R. A. Automated Driving System Safety: Miles for 95% Confidence in “Vision Zero” // SAE International Journal of Advances and Current Practices in Mobility. – 2020. – Т. 2. – №. 2020-01-1205. – С. 3454-3480.
5. Jo K. et al. Development of autonomous car—Part II: A case study on the implementation of an autonomous driving system based on distributed architecture // IEEE Transactions on Industrial Electronics. – 2015. – Т. 62. – №. 8. – С. 5119-5132.
6. Rodrigues P., Vieira S. Optimizing agent training with deep q-learning on a self-driving reinforcement learning environment // 2020 IEEE Symposium Series on Computational Intelligence (SSCI). – IEEE, 2020. – С. 745-752.
7. Chishti S. O. A. et al. Self-driving cars using CNN and Q-learning // 2018 IEEE 21st International Multi-Topic Conference (INMIC). – IEEE, 2018. – С. 1-7.
8. Chhikara P. et al. Data dimensionality reduction techniques for Industry 4.0: Research results, challenges, and future research directions // Software: Practice and Experience. – 2022. – Т. 52. – №. 3. – С. 658-688.
9. Malekloo A. et al. Machine learning and structural health monitoring overview with emerging technology and high-dimensional data source highlights // Structural Health Monitoring. – 2022. – Т. 21. – №. 4. – С. 1906-1955.
10. Peng B. et al. End-to-end autonomous driving through dueling double deep Q-network // Automotive Innovation. – 2021. – Т. 4. – С. 328-337.
11. Riboni A., Candelieri A., Borrotti M. Deep Autonomous Agents Comparison for Self-driving Cars // International Conference on Machine Learning, Optimization, and Data Science. – Cham: Springer International Publishing, 2021. – С. 201-213.
12. Kiran B. R. et al. Deep reinforcement learning for autonomous driving: A survey // IEEE Transactions on Intelligent Transportation Systems. – 2021. – Т. 23. – №. 6. – С. 4909-4926.
13. Lin Q. et al. Reachflow: An online safety assurance framework for waypoint-following of self-driving cars // 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). – IEEE, 2020. – С. 6627-6632.
14. Szoke L. et al. Skills to drive: Successor features for autonomous highway pilot // IEEE Transactions on Intelligent Transportation Systems. – 2022. – Т. 23. – №. 10. – С. 18707-18718.
15. Dosovitskiy A. et al. CARLA: An open urban driving simulator // Conference on robot learning. – PMLR, 2017. – С. 1-16.
16. Barbour E., McFall K. Autonomous vehicle simulation using open source software Carla // Journal of UAB ECTC. – 2019. – Т. 18.
17. GitHub [Электронный ресурс]. – Режим доступа: URL: <https://github.com/rzambrano1/rl-agent-self-driving-carla> (дата обращения 22.02.2024).

References

1. Final Rule Occupant Protection Amendment Automated Vehicles [Electronic resource]. URL: <https://www.nhtsa.gov/document/final-rule-occupant-protection-amendment-automated-vehicles> (circulation date 10.09.2023).
2. RAND Corporation [Electronic resource]. URL: <https://www.rand.org/> (circulation date 23.09.2023).
3. Malayjerdi M. et al. Virtual simulations environment development for autonomous vehicles interaction // ASME International Mechanical Engineering Congress and Exposition. – American Society of Mechanical Engineers, 2020. – Т. 84492. – pp. V02BT02A009.

4. Young R. A. Automated Driving System Safety: Miles for 95% Confidence in “Vision Zero” // SAE International Journal of Advances and Current Practices in Mobility. – 2020. – Т. 2. – №. 2020-01-1205. – pp. 3454-3480.
5. Jo K. et al. Development of autonomous car—Part II: A case study on the implementation of an autonomous driving system based on distributed architecture // IEEE Transactions on Industrial Electronics. – 2015. – Т. 62. – №. 8. – pp. 5119-5132.
6. Rodrigues P., Vieira S. Optimizing agent training with deep q-learning on a self-driving reinforcement learning environment // 2020 IEEE Symposium Series on Computational Intelligence (SSCI). – IEEE, 2020. – pp. 745-752.
7. Chishti S. O. A. et al. Self-driving cars using CNN and Q-learning // 2018 IEEE 21st International Multi-Topic Conference (INMIC). – IEEE, 2018. – pp. 1-7.
8. Chhikara P. et al. Data dimensionality reduction techniques for Industry 4.0: Research results, challenges, and future research directions // Software: Practice and Experience. – 2022. – Т. 52. – №. 3. – pp. 658-688.
9. Malekloo A. et al. Machine learning and structural health monitoring overview with emerging technology and high-dimensional data source highlights // Structural Health Monitoring. – 2022. – Т. 21. – №. 4. – pp. 1906-1955.
10. Peng B. et al. End-to-end autonomous driving through dueling double deep Q-network // Automotive Innovation. – 2021. – Т. 4. – pp. 328-337.
11. Riboni A., Candelieri A., Borrotti M. Deep Autonomous Agents Comparison for Self-driving Cars // International Conference on Machine Learning, Optimization, and Data Science. – Cham: Springer International Publishing, 2021. – pp. 201-213.
12. Kiran B. R. et al. Deep reinforcement learning for autonomous driving: A survey // IEEE Transactions on Intelligent Transportation Systems. – 2021. – Т. 23. – №. 6. – pp. 4909-4926.
13. Lin Q. et al. Reachflow: An online safety assurance framework for waypoint-following of self-driving cars // 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). – IEEE, 2020. – pp. 6627-6632.
14. Szoke L. et al. Skills to drive: Successor features for autonomous highway pilot // IEEE Transactions on Intelligent Transportation Systems. – 2022. – Т. 23. – №. 10. – pp. 18707-18718.
15. Dosovitskiy A. et al. CARLA: An open urban driving simulator // Conference on robot learning. – PMLR, 2017. – pp. 1-16.
16. Barbour E., McFall K. Autonomous vehicle simulation using open source software Carla // Journal of UAB ECTC. – 2019. – pp. 18.
17. GitHub [Electronic resource]. – Mode of access: URL: <https://github.com/rzambrano1/rl-agent-self-driving-carla> (circulation date 22.02.2024).

Тихонов Максим Константинович, аспирант, Сибирский федеральный университет, институт космических и информационных технологий

Непомнящий Дмитрий Олегович, магистрант, Сибирский федеральный университет, институт космических и информационных технологий

Хайдукова Валерия Николаевна, аспирант, Сибирский федеральный университет, институт космических и информационных технологий

Tikhonov Maksim Konstantinovich, Postgraduate student, Siberian Federal University, Institute of Space and Information Technologies

Nepomnyashchiy Dmitry Olegovich, Master's student, Siberian Federal University, Institute of Space and Information Technologies

Khaidukova Valeria Nikolaevna, Postgraduate student, Siberian Federal University, Institute of Space and Information Technologies