DOI: 10.24892/RIJIE/20260303

Обучение роботизированного устройства навигации в виртуальной среде Unity с применением ML-Agents*

Исхаков Р.Т., Зыков Е.Ю.
Казанский (Приволжский) федеральный университет
г. Казань, Российская Федерация
irt10let@mail.ru, Evgeniy.Zykov@kpfu.ru

Аннотация. **В статье рассматриваются функциональные возможности инструментария Unity ML-Agents и перспективы его применения для обучения роботизированных устройств. Показано, что среда Unity, изначально созданная как игровой движок, в настоящее время является полноценной платформой для проведения исследований в области машинного обучения и искусственного интеллекта.**

Целью работы является анализ инструментария ML-Agents для обучения роботизированного устройства для выполнения базовых задач навигации и движения с последующим переходом к более сложным автономным сценариям. В исследовании рассматриваются особенности обучения с подкреплением, реализованного в ML-Agents. Также сравниваются алгоритмы PPO (Proximal Policy Optimization) и SAC (Soft Actor-Critic) из которых был выбран алгоритм PPO.

Описан принцип обучения агента в среде Unity, рассматриваются особенности работы с векторными и визуальными наблюдениями, настройка гиперпараметров, а также различия между дискретным и непрерывным управлением. Для задач, связанных с перспективным использованием роботизированных манипуляторов, выбран непрерывный способ управления.

Полученные результаты позволяют оценить потенциал ML-Agents как инструмента для построения систем автономного управления роботизированными устройствами и определить его применимость для решения задач выполнения рутинных операций в автономном режиме.

***Ключевые слова*: искусственный интеллект, Unity, ML-Agents, обучение с подкреплением, cirriculuim learning, роботизированное устройство**

Введение

Современные робототехнические системы всё чаще проектируются как автономные агенты, способные ориентироваться в среде, планировать перемещения и выполнять рутинные операции без постоянного участия оператора. Однако обучение таких моделей на физических прототипах связано с высокими затратами, рисками повреждения оборудования и ограниченной воспроизводимостью экспериментов. Поэтому виртуальные симуляторы становятся важным инструментом, позволяющим быстро проверять гипотезы, масштабировать эксперименты и безопасно отрабатывать сценарии навигации и движения до переноса решений на реальное устройство.

В данной работе рассматривается использование платформы Unity и пакета Unity ML-Agents как виртуальной среды для обучения роботизированного устройства базовым навыкам перемещения и навигации методами обучения с подкреплением. Особое внимание уделяется циклу взаимодействия системе награждений, выбору типа наблюдений (векторные и сенсорные данные), а также роли учебного плана (Curriculum Learning) и рандомизации, позволяющих переходить от простых задач к более сложным конфигурациям препятствий. Для первичной оценки применимости инструментария анализируются поддерживаемые алгоритмы (в частности PPO и SAC) и обосновывается выбор PPO как более стабильного решения для задач начального уровня.

Связанные работы

Работа с обучением нейронной сети роботизированного устройства в совокупности связана с другими работами направленную в том или ином случае на автономное управление роботизированным устройством с помощью различных систем управления, от кнопочного до использования VR-гарнитур.

В данных статьях [1-3] рассматривается создание интуитивно понятного способа удаленного управления и обеспечения качественной визуальной обратной связи для оператора, работающего в экстремальных условиях. Результаты работы будут использованы в качестве набора данных для обучения нейронной сети, чтобы обучить роботизированное устройство выполнять рутинные задачи, обеспечив при этом автономную работу роботизированного устройства [4].

Использование виртуальных моделей, а также виртуальных сред для обучения нейронных сетей, позволяет ускорить процесс обучения и проводить без риска для основной физической модели роботизированного устройства эксперименты. В качестве виртуального роботизированного устройства использовалась модель, используемая в ранней работе [5]. На данный момент были достигнуты определенные результаты касаемо телеуправления и различных способов управления роботизированным устройством. Теперь же стоит задача наделить роботизированное устройство базовым автопилотом. С помощью достигнутых результатов прошлых работ требуется подключить, настроить и грамотно обучить нейронную сеть, выстроив при этом оптимальный процесс обучения роботизированного устройства.

* Публикация выполнена за счет средств субсидии, выделенной Казанскому федеральному университету для выполнения проекта № FZSM-2026-0019 «Телеуправление удаленными устройствами с использованием виртуальной реальности». Статья публикуется по рекомендации программного комитета Всероссийской научно-практической конференции Индустрия 4.0", https://smartindustrycon.ru

Возможности и перспективы использования Unity

Движок Unity разрабатывался во многом для создания игр. Но с каждым годом с добавлением некоторых дополнений в виде пакетов и библиотек функциональность Unity выросла до такого уровня, что возможности движка стали намного шире, что позволило стать Unity не только игровым движком, но и испытательным полигоном для различных исследований в различных областях. Яркими примерами могут быть примеры работ, направленных на разработку системы автономного вождения с помощью искусственного интеллекта [6], а также применение виртуальных сред для обучения нейронной сети на основе глубокого обучения с подкреплением для упрощения управления протезом [7].

Кроме этого, примеры использования инструмента ML-Agents можно найти не только в разработке видеоигр, но и при обучении роботизированных устройств [8]. В данной работе обращалось внимание на разрозенности информации касаемо применении ML-Agents для решения тех или иных задач, авторы предлагают классифицировать конкретные задачи с лучшими методами их решения.

Существуют много различных подходов к интеллектуальным методам управления роботизированным устройством. При этом, чаще комбинируют несколько подходов в целях достижения наиболее высоких результатов обучения – обучение с учителем и обучение с подкреплением. Обучение с учителем происходит на основе маркированных данных или конкретных действий оператора если мы говорим про обучение роботизированных устройств. Именно обучение с подкреплением играет ключевую роль в создании автономных роботов, и описывается структура предстоящей статьи, которая рассмотрит алгоритмы RL [9].

Инструментарий ML-Agents использует подход обучение с подкреплением для обучения нейронных сетей, что остается наиболее ключевым, ссылаясь на опыт разработки автономных роботов [9]. Однако он поддерживает также и обучение на основе демонстраций, для более сложных сценариев обучения. Также ML-Agents поддерживает такие алгоритмы как PPO и SAC:

- алгоритм PPO (Proximal Policy Optimization) в ML-Agents стоит по умолчанию, так как считается довольно стабильным и оптимальным алгоритмом для быстрого обучения выполнения простых задач. Алгоритм PPO обучается на данных, которые собирает прямо сейчас, это чуть менее эффективно в плане расхода данных, однако такой алгоритм проще настраивать;
- алгоритм SAC (Soft Actor-Critic) в отличие от PPO использует буфер памяти для повторного обучения на старом опыте. Благодаря буферу обмена нейронной сети может требоваться в 5-10 раз меньше шагов для решения той же или похожей задачи. Однако такой алгоритм тратит больше времени на математические обновления. Есть проблемы со стабильностью из-за высокой чувствительности к гиперпараметрам, из-за чего в начале обучения процесс обучения может быть нестабильным.

Для текущей апробации метода и оценки пригодности инструмента ML-Agents достаточно применения алгоритма PPO, обучая робота простым задачам на достижение целевого объекта, минуя препятствия в виде баррикад и неровностей.

Принцип работы ML-Agents Unity

Любой процесс обучения в ML-Agents на основе подхода обучения с подкреплением, где анализирует окружающую среду, реагирует на неё различными действиями, в результате которых начисляется либо награда, либо штраф. Оценка корреляции всех событий и собранных данных как раз и поддерживает процесс обучения. Таким образом, схема обучения в ML-Agents можно представить в виде замкнутого цикла обучения (рис. 1).

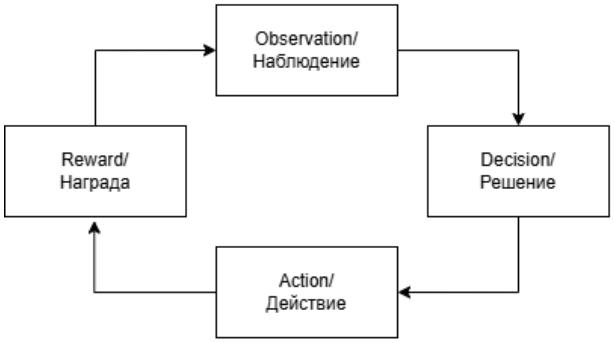

Рис. 1. Упрощенный цикл обучения в ML-Agents

На этапе «Наблюдение» агент собирает данные об окружающей среде. Эти данные могут быть как векторными, числовыми, так и визуальными в виде сенсоров, что расширяет применение инструмента в будущем. В рамках текущей задачи на сцене не будут присутствовать движущие объекты, что позволяет обойтись стандартным инструментарием ML-Agents в виде сенсора на основе лучей (Ray perception sensor).

На этапе «Решение», нейронная сеть агента анализирует полученные данные из окружающей среды, а затем выбирает наиболее на её взгляд подходящее действие из набора заранее возможных. На этом этапе нейронная сеть сама пытается подобрать эту комбинацию, на ее выбор мы можем повлиять только с помощью гипер-параметров, которые задаются в специальном конфигурационном файле. В гипер-параметрах можно задать размер нейронной сети, количество шагов, коэффициент энтропии и др.

На этапе «Действие» ML-Agents поддерживает как работу с дискретными (Discrete Action) так и с непрерывными (Continuous Action) действиями. В дискретном способе управления агентом, нейронная сеть из заданного количества ветвей и количества значений в этих ветвях подбирает оптимальную комбинацию. В таком случае число возможных вариантов ограничено и обучение будет протекать быстро. В непрерывном способе управлении агента, нейронная сеть подбирает комбинацию дробных значений из диапазона от -1 до +1. Поскольку значений бесконечное число процесс обучения будет протекать медленно, но при этом позволяет достичь более естественного и точного поведения агента что позволит обеспечить плавную работу с манипуляторами. Учитывая, что в дальнейшем планируется обучение реального роботизированного устройства с манипуляторами, было принято решение остановиться на непрерывном способ управления объектом.

Во время обучения, система вознаграждения и штрафов составляется вручную, как и способ управления агенто. Требуется указать, в каких случаях агент получит награду, а в

каких получит штраф. В нашем примере, мы указали штраф при врезании в стену и награду при достижении цели.

ML-Agents, также поддерживает эвристическое поведение, в качестве ручного управления и записи демонстраций, однако для выполнения простых задач на движение и навигации алгоритм PPO должен справится и без демонстраций.

Реализация обучения

В рамках работы была поставлена задача обучения агента на выполнение простой задачи по движению и навигации, а именно – достигнуть цели в виде небольшой сферы, минуя различные препятствия. На сцене (рис. 2) будут расположены: роботизированное устройство, цель к которой он должен будет дойти (желтая сфера), статичные препятствия (стены-ограничители, красные кубы и различные неровностей на поверхности). В целом задачи на планирования пути и навигацию, как правило требовательны к ресурсам и стратегий для обучения [10]. Но в нашем эксперименте предполагается решение такого типа задач исключительно с использованием инструмента ML-Agents.

Для обучения агента выполнять простую задачу по достижению до целевого объекта минуя препятствия потребуется подготовить обучающую среду.

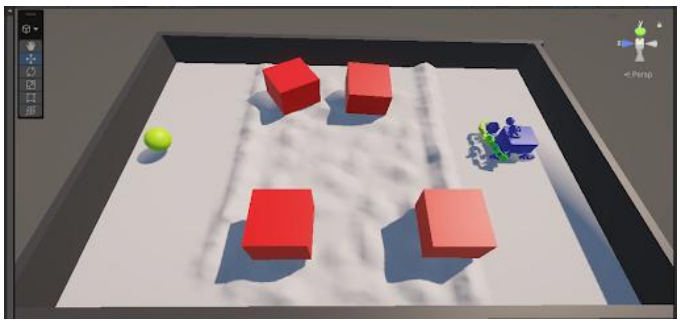

Рис. 2. Пример конечной сцены при обучении агента.

Согласно похожим исследованиям [11, 12] обучать агента сразу в относительно сложной среде можно, но крайне неэффективно. Большинство профильных компаний решали задачу изначально в упрощенной среде, а затем постепенно наращивали сложность. Такой способ обучения называется обучение по учебной программе или Curriculum Learning. По такой методике обучения, целевая задача разбивается на ряд постепенно усложняющихся этапов. В нашем случае учебный план можно разделить на несколько уроков:

1) Обучение агента на ровной поверхности без препятствий (рис. 3);

2) Обучение агента на неровной поверхности без препятствий;

3) Обучение агента на неровной поверхности с одним статичным препятствием;

4) Интеграция слабого влияния случайности на изначальное расположение цели, баррикады и агента;

5) Интеграция дополнительного препятствия;

6) Использование 4 препятствий с абсолютно случайным расположением объектов на сцене.

Такая система динамически изменяемых сцен позволит уменьшить ситуации, когда система «заучивает» один и тот же маршрут, вместо поиска оптимального решения.

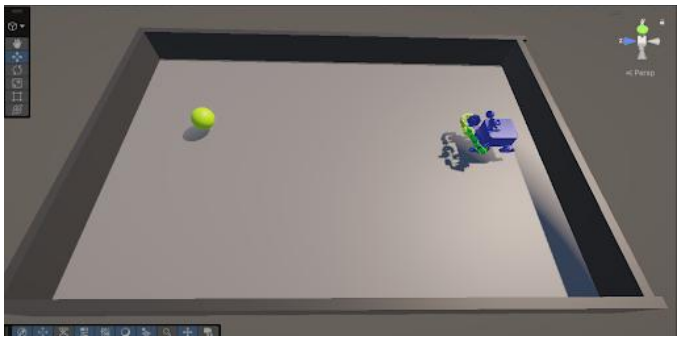

Рис. 3. Пример начальной сцены при обучении агента

Помимо обучающей сцены на которой будет протекать процесс обучения, требуется настроить агента для выполнения задачи базового движения и обхода препятствий. Для простой реализации нам потребуется подать агенту данные в качестве наблюдаемых, указать набор действий и построить систему наград. Для наглядности простой сценарий обучения можно визуализировать в виде блок-схемы (рис. 4).

Рис. 4. Простой сценарий обучения агента

В качестве наблюдаемых данных использоваться вектор координат расположения агента и цели в пространстве. Так агент будет знать, где находится сам и где находится его цель. Он будет двигаться и искать корреляции в своих изменяющихся позициях. В данной настройки агента мы не использовали сенсоры ML Agents.

В качестве действий агента мы реализовали простую систему управления движений агента вперед, назад, влево, вправо через компонент физики Unity «Rigidbody», чтобы максимально приблизить виртуальные условия к реальным.

Система начислений наград и штрафов, как и описывалось ранее достаточно тривиальна: касание препятствия приводит к штрафу системы и перезапуску эпизода обучения, а достижение цели приводит к награждению и перезапуску эпизода обучения, для закрепления навыка.

Для начала обучения агента в виртуальной среде достаточно проделанных настроек выше. Такие простые настройки являются далеко не универсальными для задач с базовым движением и навигации, однако даже с такими настройками можно выделить некоторые яркие проблемы обучения с подкреплением. Для наглядности результата мы взяли данные совокупной награды с течением кол-ва шагов агента. Шаг агента – это количество повторений полного цикла обучения, а поскольку цикл замкнут, в процессе обучения полных повторений цикла может быть достаточно большим.

На графике (рис. 5) по оси X отображается количество шагов, по оси Y изображена награда в диапазоне от -1 до 1. Для удобного чтения графика, мы взяли начальные, конечные и промежуточные результаты обучения:

- левый график представляет перед собой результаты обучения первого поколения, в котором нету препятствий и неровностей, в этом эпизоде агент учится просто доходить до цели;
- средний график является результатом 4 поколения тренировок, когда мы начинаем задавать случайные координаты в узком диапазоне;
- правый график показывает результаты последнего (6) поколения, когда сложность среды находится в своем пике.

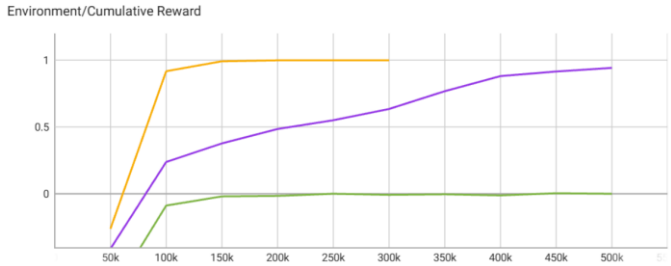

Рис. 5. График качества обучения на разных задачах.

Как можно заметить по правому графику, совокупная награда на сложной задаче стремится к нулю, что означает, что агент не справляется с выполнением задачи по достижению цели в виде сферы. Было выделено две проблемы:

1) Агент долгое время не мог дойти до цели из-за чего кружился на одном месте, стараясь не получить штраф за касание стены или баррикады;

2) При переходе на этап рандомизации позиции, агент учился обходить препятствие, с одной стороны, вне зависимости от положения цели. Качество обучения падало, так как такое поведение закреплялось в поведении агента.

Успех такого агента во многом зависел от случайности, сможет ли нейронная сеть подобрать оптимальные значения или нет. Такая реализация обучения слишком неэффективна и нуждается в ряде улучшений, позволяющих агенту действительно выучить навык, а не заучивать одно действие, которое может работать в одном эпизоде, но не работает в других.

Оптимизация алгоритма обучения

Для повышения эффективности обучения, а также преодоления нарастающей сложности в процессе обучения, была улучшена и оптимизирована настройка агента (рис. 6).

В новом подходе, в качестве подаваемых данных для наблюдения агенту использовалась информация о направлении к цели относительно робота, расстояние до цели. В отличии от прошлого подхода, новый даёт больше ценной информации агенту. Помимо векторных и числовых значений мы использовали Ray Perception Sensor, чтобы агент старался держать цель в поле зрения и двигался к ней обходя препятствия.

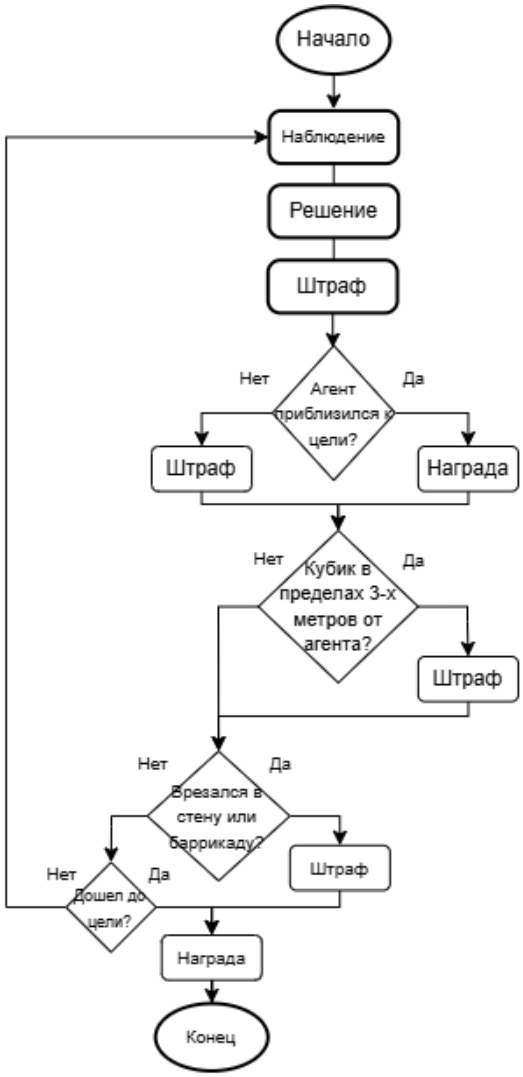

Рис. 6. Улучшенный сценарий обучения агента.

Стандартные гипер-параметры, влияющие на работу нейронной сети, считаются оптимальными для различных тестовых запусков, однако для повышения эффективности мы изменили некоторые параметры. Для сложных задач по типу обхода препятствий больший объем опыта для обучения, в теории может сделать обучение более стабильным, так как нейросеть учится на большом и разнообразном наборе данных, а не на случайных удачных или неудачных моментах. Также мы увеличили максимальное число шагов обучения агента и повысили энтропию, чтобы заставить агента больше исследовать среду для обучения.

Систему наград стала более гибкой, в отличии от простой награды при достижении цели и штрафа при врезании в стену или баррикаду. В новой итерации было решено награждать агента при любом приближение к цели. Чтобы агент искал наиболее быстрый путь был добавлен штраф за каждый шаг, чтобы агент старался найти наиболее быстрый и короткий путь к цели. Также для того, чтобы агент

не врезался в баррикады в диапазоне 3 метров агенту начислялся штраф в зависимости от степени приближения агента к баррикаде.

Для анализа результатов обучения после проведения ряда улучшений, мы также взяли первые, промежуточные и последние результаты эпизодов учебного плана обучения агента (рис. 7).

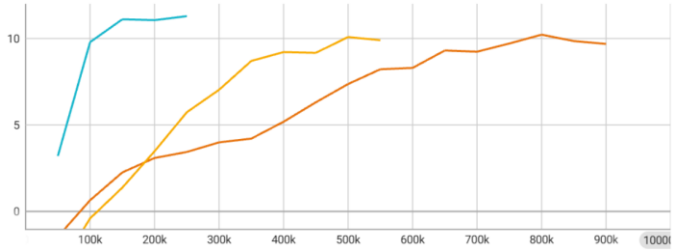

Рис. 7. Результаты качества обучения после оптимизации

Поскольку мы внесли изменения в систему начисления наград и штрафов, совокупная награда вышла за рамки диапазона от -1 до +1. Для более точных сравнений с прошлыми результатами обучения до применимых улучшений, требуется нормализация данных. Для этого потребуется вычислить максимальную награду за одно успешное прохождение. В нашем случае для нашей системы наград и штрафов подсчет максимальной награды можно свести к простой линейной формуле (1):

$$MaxReward = Reward_{Goal} + (Reward_{Step} - Penalty_{Step}) \times Steps_{Min}, \quad (1)$$

где $Reward_{Goal}$ – награда за достижение цели; $Reward_{Step}$ – награда при приближении к цели; $Penalty_{Step}$ – штраф за общее кол-во шагов; $Steps_{Min}$ – минимальное число шагов, чтобы дойти до цели.

При подсчете выяснилось, что в самом благоприятном случае максимальная награда будет равна 14,2. Используя данное значение можно найти нормализованную совокупную награду и сравнить с предыдущими результатами:

1) Синий график (левый, рис. 7) представляет результаты обучения первого поколения. Эффективность на момент конца процесса обучения в рамках данного эпизода после нормализации равна: 11,2871 / 14,2 ≈ 0,79%. Сравнивая с предыдущим графиком, можно предположить, что результат получился хуже, однако стоит учитывать, что агент получал штраф за общее число шагов в рамках одного прохождения, что повлияло на значение совокупной награды. В целом положительная тенденция обоих графиков схожа и выражена ярко;

2) Результатом 4-ого поколения (оранжевый) обучения и имеет в отличии от других, высокое число шагов обучения. Именно в этом поколении происходила рандомизация положения агента, цели и баррикад, поэтому агенту требуется больше шагов для обучения. После нормализации конечное значение совокупной награды равна: 9,6821 / 14,2 ≈ 0,68%. Сухое выравнивание значений может привести к мысли, что результат получился хуже, однако из-за особенностей системы наград и штрафов значение в 68% является довольно хорошим результатом, так как агент учится именно обходить препятствие, а не заучивает один наиболее эффективный маршрут;

3) Результат последнего поколения (желтый) после нормализации: 9,8982 / 14,2 ≈ 0,7%. Это ключевое сравнение с прошлыми результатами, так как в этом случае агент смог научиться обходить препятствия и доходить до цели вне зависимости от собственного положений и положения других объектов в виде цели и баррикад, а также в условиях неровности, получив при этом высокую награду за меньшее число шагов, чем на предыдущих поколениях.

После улучшений и оптимизации процесса обучения, агент стал справляться с поставленными задачами, мог чаще достигать цели, минуя препятствия на неровной поверхности. Полученные результаты могут быть улучшены на будущих итерациях. Например, можно провести более детальную настройку гипер-параметров или же поменять поддерживаемый ML-Agents алгоритм обучения.

Заключение

После проведения ряда попыток обучить агента выполнять простые задачи на движение и навигацию, были выявлены различные общими проблемы подхода обучения с подкреплением. Были найдены пути оптимизации, путем улучшения сценария обучения, а также более детальной настройки агента с гипер-параметрами для нейронной сети. На практике показно, что качество обученной модели нейронной сети напрямую зависит от вида и качества подаваемых данных для наблюдения, детальной настройки конфигурационного файла с гипер-параметрами и сложной системой начисления наград и штрафов. Анализ проведенных результатов до улучшений и после улучшений способов обучения агента, показал, что ML Agents является довольно гибким инструментом в плане настройки агента под нужды конкретных задач и в целом подходит для дальнейших целей по обучению в виртуальном пространстве различных роботизированных устройств.

Литература

1. Gabdullina D.R. Using OpenCV for augmented reality applications / D.R. Gabdullina, E.Yu. Zykov, V.V. Kugurakova // Journal of Advances in Artificial Life Robotics. – 2023. – Т.3, Вып. 3. – С. 179-184.

2. Кугуракова В.В. Удаленное управление роботизированным устройством с использованием технологий виртуальной реальности / В.В. Кугуракова, М.Р. Хафизов, С.А. Кадыров, Е.Ю. Зыков // Программные продукты и системы. – 2022. – Т. 35, № 3. – С. 348-361.

3. Кугуракова В.В. VR-телеуправление «многорукими» устройствами: проблемы, гипотезы, постановка задачи / В.В. Кугуракова, И.Д. Сергунин, Е.Ю. Зыков, О.Д. Сергунин, А.В. Уланов, Д.Р. Габдуллина, А.Ш. Гилемянов // Электронные библиотеки. – 2023. – Т. 25, № 5. – С. 441-471.

4. Кугуракова В.В. Точечное телеуправление для сельскохозяйственной техники, выполняющей задачи на основе искусственного интеллекта / В.В. Кугуракова, Р.Т. Исхаков, М.М. Низамутдинов, А.Т. Исхаков, И.Д. Сергунин, А.В. Уланов // Kazan Digital Week 2025. – 2025. – Т.1. – С. 1964-1971.

5. Gabdullin R.R. Creation of a synthetic dataset for training precise movements of robots for in various industries / R.R. Gabdullin, V.V. Kugurakova, R.T. Iskhakov // BIO Web of Conferences. – 2024. – Т.145, №03026. – С. 1-9.

6. Mahmoudi R. Reinforcement Learning for Obstacle Avoidance Application in Unity Ml-Agents / R. Mahmoudi, A. Ostreika // Proceedings of the 28th International Conference on Information Society and University Studies, Lithuania, – 2023. – С. 214-221.

7. Done C. Reinforcement Learning-Driven Prosthetic Hand Actuation in a Virtual Environment Using Unity ML-Agents / C. Done, J. Palmer, K. Oakey, A. Gupta, C. Thiros, J. Franklin, M.P. Schoen // Virtual Worlds – 2025. – T. 4(4), №53. – C. 1-21.

8. David H. Navigating the Unity Ml-Agents Ecosystem: A Comprehensive Review and Best-Practice Guide / H. David, S. Surati // 2025 IEEE 17th International Conference on Computational Intelligence and Communication Networks (CICN), India. – 2025. – C. 579-585.

9. Singh B. Reinforcement learning in robotic applications: a comprehensive survey / B. Singh, R. Kumar, V.P. Singh // Artificial intelligence review. – 2022. – T. 55. – C. 945-990.

10. Yang T. Intent-based deep reinforcement learning for multi-agent informative path planning / T. Yang, Y. Cao, G. Sartoretti // 2023 International Symposium on Multi-Robot and Multi-Agent Systems (MRS). – 2023. – C. 71-77.

11. Bengio Y. Curriculum learning / Y. Bengio, J. Louradour, R. Collobert, J. Weston // Proceedings of the 26th annual international conference on machine learning. – 2009. – C. 41-48.

12. Graves A. Automated Curriculum Learning for Neural Networks / A. Graves, M.G. Bellemare, J. Menick, R. Munos, K. Kavukcuoglu // Proceedings of the 34th International Conference on Machine Learning. – 2017. – T. 70. – C. 1311-1320.

DOI: 10.24892/RIJIE/20260303

Training a Robotic Navigation Device in the Unity Virtual Environment Using ML-Agents

Iskhakov R.T., Zykov E.Yu.
Kazan Federal University
Kazan, Russian Federation
irt10let@mail.ru, Evgeniy.Zykov@kpfu.ru

Abstract. **The article discusses the functional capabilities of the Unity ML-Agents toolkit and the prospects for its application in training robotic devices. It shows that the Unity environment, originally created as a game engine, is now a full-fledged platform for research in the field of machine learning and artificial intelligence.**

The goal of this work is to analyze the ML-Agents toolkit for training a robotic device to perform basic navigation and movement tasks, with a subsequent transition to more complex autonomous scenarios. The study examines the features of reinforcement learning implemented in ML-Agents. It also compares the PPO (Proximal Policy Optimization) and SAC (Soft Actor-Critic) algorithms, from which the PPO algorithm was selected.

The principle of agent training in the Unity environment is described, the features of working with vector and visual observations, hyperparameter tuning, and the differences between discrete and continuous control are considered. For tasks related to the prospective use of robotic manipulators, a continuous control method was chosen.

The results obtained allow us to evaluate the potential of ML-Agents as a tool for building autonomous control systems for robotic devices and determine its applicability for solving tasks of performing routine operations in autonomous mode.

Keywords: **artificial intelligence, Unity, ML-Agents, reinforcement learning, curriculum learning, robotic device.**

Библиографическое описание статьи

Исхаков Р.Т. Обучение роботизированного устройства навигации в виртуальной среде Unity с применением ML-Agents / Р.Т. Исхаков, Е.Ю. Зыков // Машиностроение: сетевой электронный научный журнал. – 2026. – Т.13, №3. – С. 16-21. DOI: 10.24892/RIJIE/20260303

Reference to article

Iskhakov R.T., Zykov E.Yu. Training a robotic navigation device in the unity virtual environment using ML-Agents, *Russian Internet Journal of Industrial Engineering*, 2026, vol.13, no.3, pp. 16-21. DOI: 10.24892/RIJIE/20260303