

УДК 004.896

Sim2real перенос с обучением с подкреплением для решения робототехнических задач

Гамарник Александр Александрович

Волосатова Тамара Михайловна (научный руководитель)

МГТУ им. Н.Э. Баумана, Москва, Россия

Рассмотрено применение подхода sim2real с использованием обучения с подкреплением (Reinforcement Learning, RL) для планирования траектории движения роботоманипулятора. Цифровой двойник робота манипулятора разрабатывается с использованием модели URDF и файлов геометрии STL, а затем моделируется во фреймворке PyBullet для обнаружения столкновений. Агент, обучаемый в симуляции, оптимизирует движения манипулятора методом проб и ошибок. Выученная политика переносится на физическую роботизированную руку, оснащенную двигателями Dynamixel. Результаты экспериментов показывают высокую долю успешных запусков, достигающую 100 % в симуляции, с сопоставимой производительностью при внедрении на реальное устройство. Исследование подчеркивает потенциал подхода sim2real вместе с RL, а также выявляет ограничения, такие как вычислительная сложность и проблема разрыва между симуляцией и реальностью. Данная работа представляет эффективный подход для разработки и отладки роботизированных систем в симуляции перед внедрением в реальные устройства.

Ключевые слова: обучение с подкреплением, робот-манипулятор, sim2real, цифровой двойник, метод проб и ошибок, PyBullet

Введение. Растущая автоматизация робототехники в различных отраслях демонстрирует важность эффективных и надежных методов обучения роботов решению сложных задач. Обучение с подкреплением (Reinforcement Learning, RL) стало перспективным подходом, позволяющим роботам обучаться и адаптироваться в динамических средах. Данный подход предполагает первоначальное обучение роботов в симуляциях методом проб и ошибок с последующим переносом выработанной стратегии на реального робота, что эффективнее обучения в реальном мире из-за соображений безопасности и высоких затрат [1]. В данном исследовании эти проблемы решаются за счет использования подхода sim2real [2], при котором агент обучается в виртуальной среде с помощью RL, а затем выученная стратегия переносится на реального робота. Такой подход не только снижает риски и затраты, но и ускоряет разработку автономных роботизированных систем.

Несмотря на значительный прогресс в области RL и робототехники, в современных исследованиях сохраняется ряд ограничений. Многие работы сосредоточены исключительно на обучении в симуляциях [3–4], не затрагивая проблему переноса стратегий на реальных роботов. Это приводит к низкой производительности из-за различий между виртуальной и реальной средами

— явления, известного как «разрыв реальности» (reality gap) [5]. Кроме того, недостаточно комплексных исследований, сравнивающих эффективность различных алгоритмов RL для решения робототехнических задач в сочетании с переносом sim2real [4]. Эти нерешенные вопросы подчеркивают необходимость исследований в данной области.

Цель исследования — продемонстрировать практическую эффективность подхода, при котором задача достижения целевого объекта манипулятором решается в симуляции, а выученная стратегия переносится на реальный манипулятор без риска повреждения оборудования.

Методология. Обучение с подкреплением (Reinforcement Learning, RL) — это парадигма машинного обучения, в которой агент учится выполнять задачу методом проб и ошибок, взаимодействуя со средой [3, 6–7]. В контексте роботизированного манипулирования RL позволяет манипулятору находить оптимальные стратегии, максимизируя кумулятивную награду. Агент получает состояние среды в качестве входных данных, такое как положение и ориентация конечного эффектора манипулятора, расположение объектов в пространстве, и выбирает действия, изменяющие позицию хвата. Со временем агент улучшает свою стратегию для достижения целевого результата.

Робот-манипулятор, используемый в данном исследовании, оборудован 6-ю сервоприводами Dynamixel, которые программируются для управления соединениями: 2х XL430-W250-T, 4х XL330-M288-T (рис. 1). Для отправки управляющих команд используется Python-API. Конструктивные элементы манипулятора разработаны в CAD-программах, преобразованы в формат STL и изготовлены с помощью 3D-печати.

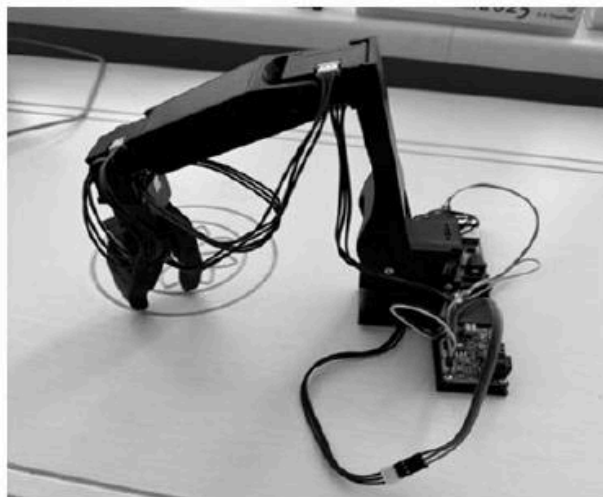


Рис. 1. Сборка робота-манипулятора

Цифровой двойник [8] робота манипулятора был сконструирован путем объединения STL-файлов геометрии, добавлением физических характеристик, и установкой звеньев и соединений в нужных позициях (рис. 2). После этого

модель сохраняется в формате Unified Robot Description Format (URDF) [9], который определяет кинематические и динамические свойства манипулятора.

Среда для симуляции была реализована с использованием физического движка PyBullet, который поддерживает динамику твердых тел, обнаружение столкновений и визуализацию в реальном времени [4]. Эти характеристики делают PyBullet подходящей платформой для тренировки и оценки алгоритмов RL в виртуальной среде (рис. 3).

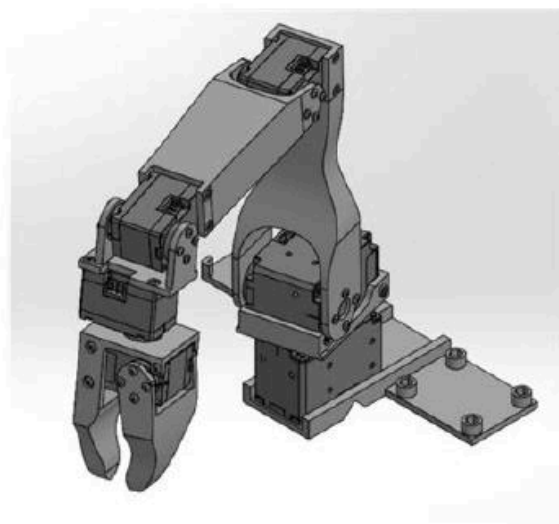


Рис. 2. Цифровой двойник робота-манипулятора

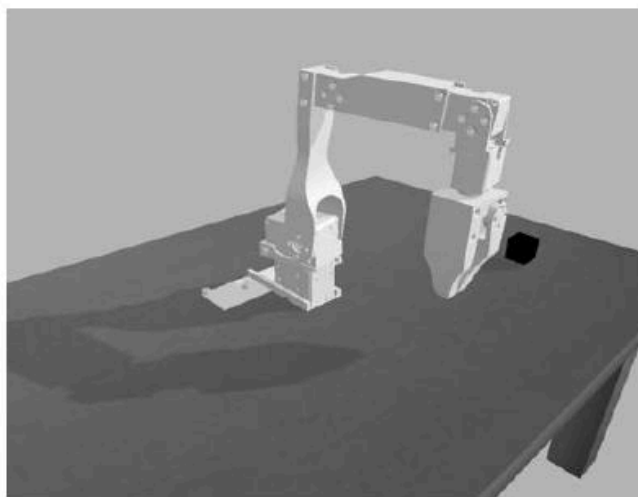


Рис. 3. Робот-манипулятор и объект в симуляции PyBullet

Основной задачей роботизированного манипулятора является достижение целевого объекта хватом. Для управления процессом обучения была определена функция вознаграждения, которая вычисляется в виде отрицательного расстояния между хватом и целевым объектом. Производительность алгоритмов RL оценивалась с использованием таких метрик, как процент успешных выполнений и кумулятивная награда за время обучения. Эти показатели дают представление об эффективности алгоритмов как в симуляции, так и при работе в реальных условиях.

Процент успешных выполнений S определяет, как часто агент успешно выполнял задачу при фиксированном числе запусков эпизода

$$S = \frac{N_{success}}{N_{total}}, \quad (1)$$

где $N_{success}$ — число эпизодов, в которых агент успешно выполнил задачу; N_{total} — общее число эпизодов.

Кумулятивная награда — это сумма всех наград, полученных агентом на каждом шаге времени во время эпизода. Она показывает, как агент максимизирует долгосрочную награду в течение эпизода, что является основной целью в RL. Кумулятивная награда G_t на шаге t определяется формулой

$$G_t = \sum_{k=0}^T \gamma^k r_{t+k}, \quad (2)$$

где r_{t+k} — награда, полученная на шаге $t + k$; γ — дисконтирующий фактор, значение в интервале $[0, 1]$; T — общее число шагов в эпизоде.

Результаты. Эксперимент был проведен в симуляции, после чего на реальной роботизированной системе для оценки эффективности переноса *sim2real*. PyBullet обеспечил реалистичную среду для обучения (рис. 3). Для обучения манипулятора в симуляции использовался алгоритм обучения с подкреплением (RL) с последующим переносом стратегии на реального робота. Реальный робот воспроизводил действия манипулятора из симуляции при выполнении задачи.

Процесс обучения включает обучение манипулятора для решения поставленной задачи методом проб и ошибок. Со временем агент оптимизирует свою стратегию для максимизации кумулятивной награды, достигая эффективного и надежного выполнения задачи. В ходе обучения каждый новый эпизод начинается со случайного позиционирования целевого объекта.

Результаты процесса обучения представлены на рисунках 4 и 5, где показаны процент успешных запусков, рассчитанный по формуле (1) и кумулятивная награды в процессе обучения, полученная из формулы (2). Доля успешных запусков возрастала по мере оптимизации стратегии алгоритмом. Рост кумулятивного вознаграждения за эпизод свидетельствует об успешном выполнении задачи.

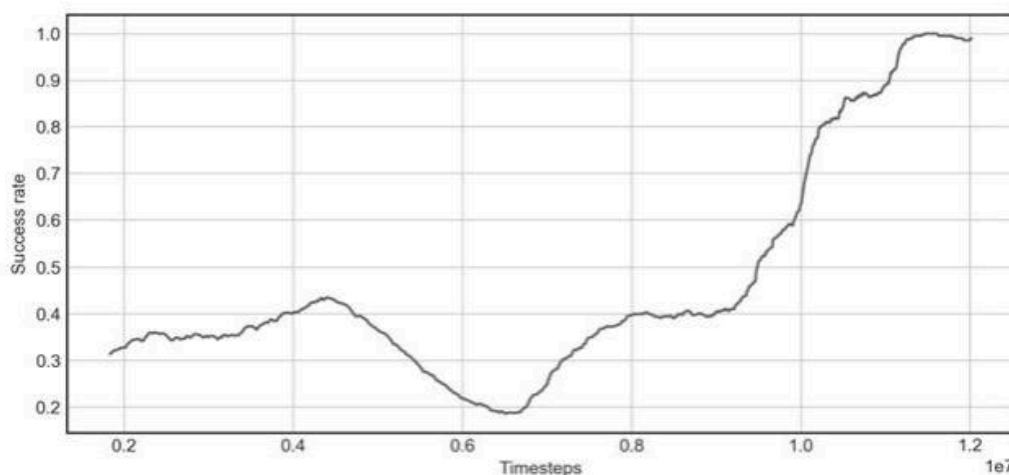


Рис. 4. График зависимости доли успешных запусков и от общего числа временных шагов в процессе обучения

На этапе переноса *sim2real* освоенная стратегия была развернута на физическом манипуляторе. Реальный робот воспроизвел поведение, отработанное в симуляции, достигнув сопоставимой эффективности и продемонстрировав стабильную работу в реальных условиях.

Обсуждение полученных результатов. Результаты эксперимента демонстрируют успешное применение метода *sim2real* в сочетании с RL для решения задачи планирования пути манипулятора. Агент в симуляции достиг 100 %-ного успеха в достижении целевого объекта, что подтверждается соответствующими метриками в процессе обучения. Перенос стратегии на физи-

ческий манипулятор дополнительно доказывает эффективность предложенного метода. Эти результаты подчеркивают потенциал подхода `sim2real` для преодоления разрыва между симуляцией и реальностью, обеспечивая эффективное и безопасное обучение роботизированных систем.

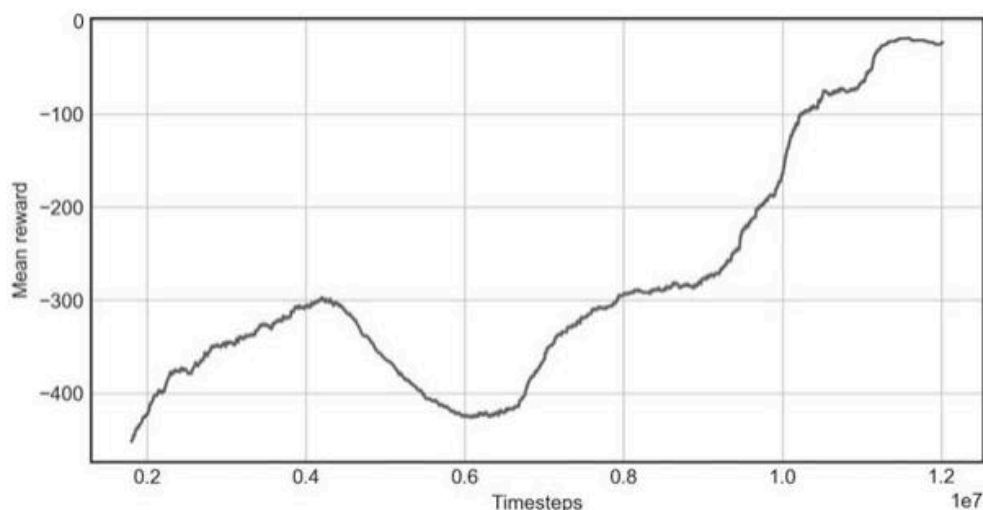


Рис. 5. График зависимости кумулятивной награды от общего числа временных шагов в процессе обучения

Цифровой двойник в сочетании с физическим движком PyBullet позволяет агенту RL освоить стратегию, которая затем переносится на реальное роботизированное устройство. Этот подход согласуется с предыдущими исследованиями, подчеркивающими важность точной симуляции для эффективного переноса `sim2real` [3–4]. Кроме того, определение функции награды играет важную роль, поскольку это напрямую влияет на кумулятивную награду и долю успешных эпизодов.

Несмотря на положительные результаты, необходимо обсудить несколько ограничений. Во-первых, обучение алгоритмов RL требует значительных вычислительных ресурсов и времени, особенно для задач с пространством состояний и действий больших размерностей. Во-вторых, сложность среды симуляции, несмотря на ее преимущества для обучения, может не полностью отражать все аспекты реальной динамики, что приводит к проблеме, известной как «разрыв с реальностью». Наконец, в текущем исследовании рассматривалась только одна задача, и необходимы дальнейшие работы для оценки возможности обобщения подхода на более сложные задачи и среды.

Будущие исследования должны быть направлены на поиск способов сокращения времени обучения. Кроме того, использование доменной рандомизации в процессе обучения может повысить устойчивость переноса `sim2real` за счет ознакомления агента RL с более широким диапазоном симулированных условий. Несмотря на эти сложности, сочетание методов `sim2real` и RL открывает значительные перспективы для развития роботизированной автоматизации и предлагает масштабируемое и экономически эффективное решение для обучения роботов выполнению сложных задач.

Литература

- [1] Wang S., Zhang J., Wang P., Law J., Calinescu R., Mihaylova L. A deep learning-enhanced digital twin framework for improving safety and reliability in human–robot collaborative manufacturing. *Robotics and Computer-Integrated Manufacturing*, 2024, vol. 85. <https://doi.org/10.1016/j.rcim.2023.102608>
- [2] Zhao W., Queralta J. P., Westerlund T. Sim-to-real transfer in deep reinforcement learning for robotics: a survey. *IEEE symposium series on computational intelligence (SSCI)*, 2020, pp. 737–744. Arxiv:2009.13303v2.
- [3] Liu Z.X., Wang Q., Yang B. [Retracted] Reinforcement Learning-Based Path Planning Algorithm for Mobile Robots. *Wireless Communications and Mobile Computing*, 2022, vol. 2022, no. 1, art. no. 1859020.
- [4] Gallouédec Q. Open-source goal-conditioned environments for robotic learning. Arxiv preprint arXiv:2106.13687, 2021.
- [5] Collins J., Howard D., Leitner J. Quantifying the reality gap in robotic manipulation tasks. *International Conference on Robotics and Automation (ICRA)*, 2019, pp. 6706–6712. Arxiv:1811.01484v2.
- [6] Luo J. Deep reinforcement learning for robotic assembly of mixed deformable and rigid objects. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2018, pp. 2062–2069.
- [7] Sangiovanni B. Deep reinforcement learning for collision avoidance of robotic manipulators. *European Control Conference (ECC)*, 2018, pp. 2063–2068. <https://doi.org/10.23919/ECC.2018.8550363>
- [8] Grieves M. Digital twin: manufacturing excellence through virtual factory replication. White paper, 2014, vol. 1, no. 2014, pp. 1–7.
- [9] Tola D., Corke P. Understanding URDF: A dataset and analysis. *IEEE Robotics and Automation Letters*, 2024. <https://doi.org/10.48550/arXiv.2308.00514>

Sim2real transfer with reinforcement learning to solve robotic arm tasks

Gamarnik Alexander Alexandrovich

Volosatova Tamara Mikhailovna (*scientific advisor*)

BMSTU, Moscow, Russia

This study explores an application of sim2real with reinforcement learning (RL) for path planning in a robotic arm. A digital twin of the robotic arm is developed with a URDF model and STL geometry files and simulated in the PyBullet framework. The RL algorithm trained in a simulation learns to optimize the arm's movements through trial-and-error. The learned policy is transferred to a physical robotic arm. Experimental results show 100 % success rate in the simulation with comparable performance during deployment into the physical arm. The study highlights the potential of sim2real RL for robotic automation. It also identifies limitations such as computational complexity and a reality gap problem. This work presents an effective solution to train robotic systems in changing environments.

Keywords: reinforcement learning, robotic arm, sim2real, digital twin, trial-and-error method, PyBullet