

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное
образовательное учреждение высшего образования
«Национальный исследовательский университет ИТМО»
(Университет ИТМО)

Факультет систем управления и робототехники

**ОТЧЕТ
О ПРОХОЖДЕНИИ УЧЕБНОЙ ПРАКТИКИ**

ОСНОВЫ СИМУЛЯЦИИ И ОБУЧЕНИЯ ЧЕТВЕРОНОГО РОБОТА В
СРЕДЕ ISAAC SIM

Студент:

Группа № R3237

В.А. Кутузов

Руководитель практики:

Профессор практики, к.т.н.

И.И. Борисов

Санкт-Петербург
2025

СОДЕРЖАНИЕ

Содержание	2
Введение	4
1 Ознакомление с симулятором Isaac Sim	5
1.1 Архитектура и основные возможности Isaac Sim	5
1.2 Установка и настройка симуляционной среды	5
1.3 Изучение обучающих материалов и базовые эксперименты в Isaac Sim	7
1.3.1 Работа с обучающими материалами Isaac Sim	7
1.3.2 Создание и настройка виртуальной среды	7
1.3.3 Работа с роботами и управление суставами	8
1.4 Введение в обучение с подкреплением (RL) для роботов	8
2 Обучение квадроного робота ходьбе	10
2.1 Постановка задачи обучения ходьбе	10
2.2 Использование фреймворка Isaac Lab для обучения	11
2.2.1 Представление состояния и действий в Isaac Lab	11
2.2.2 Запуск и настройка процесса обучения	11
2.2.3 Преимущества использования готового фреймворка	12
2.3 Анализ результатов обучения с использованием TensorBoard ...	12
2.3.1 Основные отслеживаемые метрики	13
2.4 Результаты	13
3 Обучение квадроного робота сложным движениям	14
3.1 Определение сложных движений и их параметризация	14
3.1.1 Типы сложных движений	14
3.1.2 Параметризация сложных движений	14
3.2 Попытка ретаргетинга движений из датасета LaFAN	15
3.2.1 Работа с датасетом LaFAN	15
3.2.2 Извлечение танцевальных движений	16
3.2.3 Ретаргетинг движений на квадроного робота	16
3.2.4 Проблемы и ограничения ретаргетинга	16

Заключение	18
Список литературы	19
Список использованных источников.....	20

ВВЕДЕНИЕ

В современной робототехнике использование симуляторов для обучения роботов становится все более важным этапом перед реализацией алгоритмов на реальных устройствах. Это позволяет сэкономить ресурсы, избежать поломок дорогостоящего оборудования и роботов, значительно ускорить процесс разработки. Особенно актуален данный подход при работе с алгоритмами обучения с подкреплением (Reinforcement Learning, RL), которые требуют большого количества итераций для достижения желаемого результата.

В рамках данной практики исследуются возможности симулятора Isaac Sim от NVIDIA для обучения квадропедного робота типа Anymal или подобных.

Цель практики: освоить методы обучения квадропедного робота ходьбе и сложным движениям с использованием алгоритмов обучения с подкреплением в симуляторе Isaac Sim.

Задачи практики:

1. Ознакомиться с симулятором Isaac Sim, изучить его архитектуру и основные возможности.
2. Изучить литературу по применению алгоритмов обучения с подкреплением для управления квадропедными роботами.
3. Реализовать и протестировать базовые алгоритмы обучения робота ходьбе.
4. Попробовать реализовать сложные движения робота (танцевальные элементы).

[Ссылка на удаленный репозиторий](#)

1 ОЗНАКОМЛЕНИЕ С СИМУЛЯТОРОМ ISAAC SIM

1.1 Архитектура и основные возможности Isaac Sim

Isaac Sim представляет собой симуляционную платформу для роботов, разработанную компанией NVIDIA на базе платформы Omniverse. Эта среда предназначена для разработки, тестирования и обучения роботов в виртуальных средах с точным моделированием физики.

Основные компоненты и возможности Isaac Sim:

- **Физический движок PhysX 5.0** — обеспечивает реалистичное моделирование физических взаимодействий, включая контактные силы, трение и динамику тел.
- **Интеграция с USD (Universal Scene Description)** — формат представления сцен, позволяющий описывать сложные трехмерные сцены.
- **Поддержка RTX** — использование технологии рейтрейсинга для создания фотореалистичной визуализации.
- **Интеграция с ROS/ROS2** — обеспечивает возможность использования пакетов Robot Operating System.
- **Библиотеки для обучения с подкреплением** — встроенная поддержка Isaac Lab и интеграция с популярными фреймворками RL.
- **Симуляция датчиков** — моделирование различных типов сенсоров, включая LiDAR, камеры, IMU и другие.

1.2 Установка и настройка симуляционной среды

В процессе подготовки к работе с Isaac Sim была выявлена недостаточная вычислительная мощность имеющегося ноутбука для работы с симулятором. Isaac Sim требует высокопроизводительного графического ускорителя и значительных вычислительных ресурсов, особенно при использовании алгоритмов обучения с подкреплением, которые предполагают параллельную симуляцию множества агентов.

Для решения этой проблемы был выбран подход с использованием контейнеризации на удаленном вычислительном кластере:

1. Доступ к вычислительному кластеру с GPU-ускорителями NVIDIA был предоставлен знакомым.
2. На кластере был развернут Docker-контейнер со специализированным образом для работы с Isaac Sim.
3. В контейнер были установлены необходимые компоненты:
 - Isaac Sim 4.5.0 — основная симуляционная платформа
 - Isaac Lab — набор инструментов для обучения с подкреплением
 - Visual Studio Code — для удаленной разработки и отладки кода
 - Python с необходимыми библиотеками (PyTorch, NumPy, SciPy, и др.)
 - ROS/ROS2 интеграция для работы с робототехническими пакетами

Для взаимодействия с удаленным контейнером был настроен SSH-туннель с перенаправлением портов для доступа к графическому интерфейсу Isaac Sim через веб-браузер. Такой подход обеспечил полноценную работоспособность среды разработки даже при подключении с относительно слабого ноутбука.

```
File Edit Format View Help
FROM skabrits/web-ubuntu:0.4.1

USER 0

RUN apt update
RUN wget -O vs_code.deb https://go.microsoft.com/fwlink/?LinkID=760868
RUN apt install ./vs_code.deb -y

USER 1000
```

Рисунок 1 — Пример: Dockerfile с установкой VS Code

Такой подход позволил эффективно использовать вычислительные ресурсы кластера для ресурсоемких задач обучения, в то время как локальная машина использовалась в основном для подготовки данных и анализа результатов.

На настройку контейнера и изучения необходимой литературы потребовалось немало времени, однако это то, что позволило выполнять практику в данном симуляторе, поэтому затраченные усилия оправданы.

1.3 Изучение обучающих материалов и базовые эксперименты в Isaac Sim

Важной частью освоения Isaac Sim стало изучение обучающих материалов и проведение базовых экспериментов для знакомства с функциональностью симулятора. NVIDIA предоставляет приличную коллекцию туториалов, демонстрирующих различные возможности платформы.

1.3.1 Работа с обучающими материалами Isaac Sim

Особенно полезными оказались примеры взаимодействия с API симулятора через Python-скрипты, которые позволяют программно управлять параметрами симуляции и автоматизировать эксперименты.

1.3.2 Создание и настройка виртуальной среды

После изучения базовых туториалов были проведены эксперименты по созданию собственных виртуальных сред:

1. **Создание примитивов** — добавление в сцену базовых геометрических форм (кубы, цилиндры) с различными физическими свойствами.
2. **Настройка материалов и текстур** — применение различных визуальных свойств к объектам.
3. **Формирование ландшафта** — создание поверхностей различного типа для тестирования передвижения роботов.
4. **Настройка физических параметров** — эксперименты с гравитацией.

1.3.3 Работа с роботами и управление суставами

Значительная часть экспериментов была посвящена добавлению и управлению робототехническими системами:

- **Импорт моделей роботов** — добавление в сцену готовых моделей из библиотеки Isaac Sim, включая различные квадроопедные платформы (ANYmal от ANYbotics, Spot от Boston Dynamics).
- **Изучение структуры модели** — анализ иерархии суставов, звеньев и систем координат робота через Stage Tree и Property Window.
- **Базовое управление суставами** — эксперименты с позиционным управлением отдельными суставами через графический интерфейс и программно.

Особый интерес представляли эксперименты по программному управлению суставами квадроопедного робота. Был разработан простой python-скрипт для последовательного движения суставами левой передней ноги:

1.4 Введение в обучение с подкреплением (RL) для роботов

Обучение с подкреплением представляет собой парадигму машинного обучения, в которой агент (робот) обучается путем взаимодействия с окружающей средой. Основные компоненты RL-системы:

- **Агент/Agent** — сущность, принимающая решения и выполняющая действия (в нашем случае, виртуальный квадроопедный робот).
- **Окружение/Environment** — среда, с которой взаимодействует агент (виртуальный мир в Isaac Sim).
- **Состояния/States** — информация о текущем положении и параметрах агента и окружения.
- **Действия/Actions** — возможные команды, которые агент может выполнить.
- **Награды/Rewards** — сигналы, указывающие на успешность или неудачность действий агента.
- **Политика/Policy** — стратегия выбора действий агентом в зависимости от состояния.

Процесс обучения заключается в нахождении оптимальной политики, делающей суммарную награду максимальной. В рамках данной практики был рассмотрен следующий алгоритм:

- **Proximal Policy Optimization (PPO)** — алгоритм, основанный на оптимизации стохастического градиента с ограничениями на размер шага обновления политики.

В Isaac Sim имеется поддержка нескольких фреймворков для реализации RL-алгоритмов, например:

- **Isaac Lab** — это унифицированный модульный фреймворк для обучения роботов, который призван упростить общие рабочие процессы в исследованиях робототехники (такие как обучение с подкреплением, обучение на демонстрациях и планирование движений).

2 ОБУЧЕНИЕ КВАДРОПЕДНОГО РОБОТА ХОДЬБЕ

2.1 Постановка задачи обучения ходьбе

Обучение квадропедного робота ходьбе представляет собой фундаментальную задачу в области робототехники. Цель состоит в том, чтобы научить робота стабильно и эффективно передвигаться по различным поверхностям. Формально задача может быть определена следующим образом:

Входные данные:

- Состояние робота: положение и ориентация корпуса, углы суставов, угловые скорости.
- Состояние окружения: характеристики поверхности, препятствия (если есть).
- Целевое направление и скорость движения.

Выходные данные:

- Управляющие сигналы для приводов суставов робота (углы или крутящие моменты).

Для успешного решения задачи необходимо определить следующие компоненты:

1. **Представление состояния** — выбор информативных признаков, описывающих текущее состояние робота и среды.
2. **Пространство действий** — определение возможных управляющих сигналов для приводов робота.
3. **Функция награды** — метрика, оценивающая успешность выполнения задачи ходьбы.
4. **Параметры обучения** — гиперпараметры алгоритма RL, определяющие процесс обучения.

2.2 Использование фреймворка Isaac Lab для обучения

Для реализации обучения квадропедного робота ходьбе был использован встроенный в Isaac Lab фреймворк для обучения с подкреплением, расположенный в директории `scripts/reinforcement_learning/rsl_rl/`. Данный фреймворк предоставляет реализацию алгоритма PPO.

2.2.1 Представление состояния и действий в Isaac Lab

Фреймворк Isaac Lab уже включает оптимизированное представление состояния для квадропедного робота, которое содержит:

- Положение центра масс робота (3 координаты).
- Ориентация корпуса робота (кватернион, 4 значения).
- Линейная скорость центра масс (3 компоненты).
- Угловая скорость корпуса (3 компоненты).
- Углы суставов (12 значений для 4 ног по 3 сустава).
- Угловые скорости суставов (12 значений).
- Информация о контакте ног с поверхностью (4 бинарных значения).
- Целевое направление и скорость движения (3 значения).

Действия в модели представлены в виде целевых углов для 12 суставов робота. Фреймворк использует позиционное управление, которое обеспечивает более стабильное обучение на начальных этапах по сравнению с управлением крутящими моментами.

2.2.2 Запуск и настройка процесса обучения

Процесс обучения запускался с помощью скрипта `train.py` из директории `scripts/reinforcement_learning/rsl_rl/`. Данный скрипт автоматизирует весь процесс обучения. Например:

```
python scripts/reinforcement_learning/rsl_rl/train.py \
    --task=Isaac-Velocity-Rough-Anymal-C-v0 \
    --num_envs=256 \
    --num_iterations=5000 \
```

--headless

В процессе экспериментов варьировались следующие параметры:

- **Количество итераций обучения** — для определения оптимального времени обучения.
- **Параметры робота** — характеристики, жесткость суставов.
- **Весовые коэффициенты функции награды** — для достижения лучшей скорости, стабильности.

При этом гиперпараметры алгоритма PPO оставались по большей части неизменными, так как они уже были хорошо оптимизированы разработчиками Isaac Lab.

2.2.3 Преимущества использования готового фреймворка

Использование встроенного в Isaac Lab фреймворка для обучения с подкреплением предоставило ряд существенных преимуществ:

- **Оптимизированная производительность** — фреймворк использует параллельную симуляцию нескольких (до 4096) экземпляров робота для эффективного сбора данных.
- **Готовые инструменты мониторинга** — автоматическое логирование метрик обучения для дальнейшего анализа в TensorBoard.
- **Проверенные реализации алгоритмов** — использование тщательно протестированной и оптимизированной реализации PPO.
- **Простота экспериментирования** — быстрое изменение параметров через конфигурационные файлы без необходимости модификации кода.

2.3 Анализ результатов обучения с использованием TensorBoard

Для мониторинга и анализа процесса обучения использовался инструмент TensorBoard, который позволяет визуализировать метрики и сравнивать результаты разных запусков. Например:

```
ubuntu@tester-web-ubuntu-7854c599db-rlvrх:  
~/Documents/IsaacLab-main$  
./isaaclab.sh -p -m tensorboard.main --logdir  
logs/rsl_rl/anymal_c_rough
```

2.3.1 Основные отслеживаемые метрики

В процессе обучения с помощью TensorBoard отслеживались следующие метрики:

- Средняя награда за эпизод
- Компоненты награды
- Средняя скорость движения
- Стабильность ориентации
- Потери функции ценности и политики
- Энтропия политики

2.4 Результаты

Результаты, полученные при использовании фреймворка Isaac Lab, показали, что после примерно 2000 итераций алгоритм достигает стабильной политики ходьбы.

3 ОБУЧЕНИЕ КВАДРОПЕДНОГО РОБОТА СЛОЖНЫМ ДВИЖЕНИЯМ

3.1 Определение сложных движений и их параметризация

После обучения робота ходьбе следующим этапом стало обучение более сложным движениям, таким как танцевальные элементы или акробатические трюки. Для этого необходимо четко определить целевые движения и способы их параметризации.

3.1.1 Типы сложных движений

Для четырехногого робота можно выделить следующие категории сложных движений:

1. **Позиционные трюки** — специфические статические позы, такие как стойка на двух ногах, приседание, подъем передней части корпуса и т.д.
2. **Динамические трюки** — движения, требующие балансировки и координированной работы всех суставов, например, прыжки, повороты в воздухе, перекаты.
3. **Танцевальные последовательности** — серии синхронизированных движений, имитирующие танцевальные.
4. **Имитационные движения** — подражание движениям животных или других роботов.

В рамках данной практики было решено сосредоточиться на создании танцевальной последовательности.

3.1.2 Параметризация сложных движений

Для описания сложных движений необходима соответствующая параметризация. Были рассмотрены несколько подходов:

1. **Явное задание ключевых поз** — определение набора опорных конфигураций робота.

2. **Использование периодических функций** — представление движений как комбинации синусоид различной частоты и амплитуды.
3. **Обучение с подкреплением со специальной функцией награды** — формулировка цели движения через функцию награды без явного задания траекторий.
4. **Ретаргетинг движений человека** — перенос движений из захвата движений человека на кинематическую структуру робота.

3.2 Попытка ретаргетинга движений из датасета LaFAN

Одной из наиболее интересных исследовательских задач в рамках данной работы была попытка переноса человеческих танцевальных движений на квадроцепного робота. Для этого был выбран датасет LaFAN (Ubisoft LaForge Animation Dataset), содержащий разнообразные захваты движений человека, включая танцевальные последовательности.

3.2.1 Работа с датасетом LaFAN

LaFAN — это набор данных захвата движений человека, разработанный Ubisoft LaForge для исследований в области анимации персонажей и машинного обучения. Датасет содержит различные типы движений, включая ходьбу, бег, танцы и акробатические элементы.

Для работы с датасетом были использованы предоставляемые инструменты оценки и извлечения данных. В частности, был изучен файл `evaluate.py`, который демонстрирует базовый пайплайн работы с данными LaFAN:

1. Извлечение статистических данных из тренировочного набора
2. Загрузка тестового набора данных
3. Оценка результатов интерполяции движений

Для проверки корректности работы с датасетом был также использован скрипт `evaluate_test.py`.

3.2.2 Извлечение танцевальных движений

Для извлечения танцевальных движений из файлов формата BVH был создан скрипт `extract_bvh.py`. Этот скрипт выполняет следующие операции:

1. Загрузка BVH-файла с танцевальной последовательностью
2. Извлечение информации о суставах и кадрах анимации
3. Преобразование данных в формат JSON с временными метками
4. Сохранение обработанных данных в файл `human_dance_poses.json`

3.2.3 Ретаргетинг движений на квадроножного робота

Для переноса человеческих танцевальных движений на кинематическую структуру квадроножного робота был разработан скрипт `retarget.py`. Основная идея заключалась в сопоставлении ключевых суставов человека с соответствующими суставами робота.

Скрипт выполняет следующие операции:

1. Загрузка URDF-модели квадроножного робота в симуляторе PyBullet
2. Загрузка данных о движениях человека из файла `human_dance_poses.json`
3. Сопоставление суставов человека и робота
4. Генерация соответствующих движений робота
5. Сохранение результатов в файл `robot_dance_poses.json`

3.2.4 Проблемы и ограничения ретаргетинга

Несмотря на значительные усилия, полноценно реализовать ретаргетинг танцевальных движений с человека на квадроножного робота не удалось по ряду причин:

1. **Фундаментальные различия в кинематике** — человеческий скелет и скелет четвероножного робота имеют принципиально различную структуру, что затрудняет прямое сопоставление суставов.

2. **Различия в рабочих диапазонах** — диапазоны движения суставов робота значительно отличаются от человеческих, что требует масштабирования и адаптации углов поворота, решения сложных задач обратной кинематики.
3. **Отсутствие прямых аналогов для верхней части тела** — движения рук, плеч и головы человека, которые составляют важную часть выразительности танца, не имеют прямых аналогов в кинематической структуре квадропедного робота.

ЗАКЛЮЧЕНИЕ

В ходе прохождения практики был успешно выполнен ряд поставленных задач (за исключением качественной реализации танца робота), что позволило получить ценный опыт в области программирования и обучения робототехнических систем. Основные результаты работы можно сформулировать следующим образом:

1. Приобретены навыки работы с Docker контейнерами и контейнеризацией в целом.
2. Освоена среда Isaac Sim, изучены основные принципы работы с симулятором для разработки и тестирования алгоритмов управления роботами.
3. Разработаны базовые программные компоненты на языке Python для управления отдельными суставами квадроцепного робота.
4. Успешно реализовано обучение квадроцепного робота ходьбе с использованием фреймворка Isaac Lab и алгоритма PPO. Проведены эксперименты с различными параметрами функции награды и конфигурациями модели.
5. Проведен подробный анализ результатов, позволивший выявить зависимость качества обучения от количества итераций и параметров функции награды.

В процессе прохождения практики был преодолен ряд технических сложностей, в частности, связанных с разверткой контейнера, настройкой приложений, выполнения задания.

Практический опыт, полученный в ходе работы, имеет для меня высокую ценность для дальнейшего профессионального развития в области робототехники и искусственного интеллекта. За время практики было изучено большое количество информации. Это, частично, повлияло на то, что на практическую работу было намного меньше времени, чем хотелось бы.

В перспективе планируется продолжить освоение симуляторов (Isaac Sim, MuJoCo и др.), изучение статей и нововведений в мире квадроцепной робототехники.

СПИСОК ЛИТЕРАТУРЫ

- [1] Машинное обучение: справочник [Электронный ресурс] // Яндекс Хендбук. — 2025. — Режим доступа: <https://education.yandex.ru/handbook/ml> (дата обращения: 29.06.2025).
- [2] Саттон, Р. С. Обучение с подкреплением / Р. С. Саттон, Э. Г. Барто; пер. с англ. — М.: ДМК Пресс, 2022. — 552 с. — ISBN 978-5-97060-097-6.
- [3] Coumans, E. PyBullet physics engine [Электронный ресурс] // GitHub Repository. — 2021. — Режим доступа: <https://github.com/bulletphysics/bullet3> (дата обращения: 29.06.2025).
- [4] DeepMind x UCL | Introduction to Reinforcement Learning 2015 [Видеозапись] / D. Silver // YouTube, DeepMind. — 2015. — Режим доступа: <https://www.youtube.com/playlist?list=PLqYmG7hTraZDM-OYHWgPebj2MfCFzFObQ> (дата обращения: 29.06.2025).
- [5] Makoviychuk, V. Isaac Gym: High Performance GPU-Based Physics Simulation For Robot Learning / V. Makoviychuk, L. Wawrzyniak, Y. Guo, M. Lu, K. Storey, M. Macklin, D. Hoeller, N. Rudin, A. Allshire, A. Handa, G. State // Advances in Neural Information Processing Systems. — 2021. — Vol. 34. — P. 10378-10390.
- [6] NVIDIA Isaac Lab: A Platform for Developing Robotic Skills with Reinforcement Learning [Электронный ресурс] // — 2024. — Режим доступа: <https://isaac-sim.github.io/IsaacLab/main/index.html> (дата обращения: 29.06.2025).
- [7] NVIDIA Isaac Sim Documentation [Электронный ресурс] // NVIDIA Developer Documentation. — 2024. — Режим доступа: <https://docs.isaacsim.omniverse.nvidia.com/latest/index.html> (дата обращения: 29.06.2025).
- [8] Schulman, J. Proximal Policy Optimization Algorithms / J. Schulman, F. Wolski, P. Dhariwal, A. Radford, O. Klimov // arXiv preprint arXiv:1707.06347. — 2017.

- [9] Silver, D. Teaching [Электронный ресурс] // David Silver's Blog. — 2025. — Режим доступа: <https://davidstarsilver.wordpress.com/teaching/> (дата обращения: 29.06.2025).
- [10] Ubisoft LaForge Animation Dataset (LaFAN1) [Электронный ресурс] // GitHub Repository. — 2020. — Режим доступа: <https://github.com/ubisoft/ubisoft-laforge-animation-dataset> (дата обращения: 29.06.2025).
- [11] Villegas, R. Neural Kinematic Networks for Unsupervised Motion Retargetting / R. Villegas, J. Yang, D. Ceylan, H. Lee // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. — 2018. — P. 8639-8648.