Международный университет информационных технологий

УДК: 004.896 На правах рукописи

МОМЫНҚУЛОВ ЗЕЙНЕЛЬ ЗЕЙНУЛЛАҰЛЫ

Разработка моделей глубокого обучения с подкреплением для управления роботизированным манипулятором в промышленных приложениях

8D06105 – Наука о данных

Диссертация на соискание степени доктора философии (PhD)

Отечественный научный консультант:
PhD, профессор-исследователь,
Омаров Батырхан Султанович

Зарубежный научный консультант:
PhD, PhD professor Azizah Suliman,
университет Asia Metropolitan University,
Subang Jaya, Selangor, Malaysia,

Республика Казахстан
Алматы, 2026

СОДЕРЖАНИЕ

ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

ПЛК	-	Программируемый логический контроллер
DDPG	-	Deep Deterministic Policy Gradient (Глубокий детерминированный градиент политики)
DRL	-	Deep Reinforcement Learning (Глубокое обучение с подкреплением)
HRC	-	Human-Robot Collaboration (Коллаборативное взаимодействие человека и робота)
IL	-	Imitation Learning (Имитационное обучение)
IoT	-	Internet of Things (Интернет вещей)
LfD	-	Learning from Demonstration (Обучение на демонстрациях)
LLM	-	Large Language Model (Большая языковая модель)
MPC	-	Model Predictive Control (Предиктивное управление на основе модели)
OMPL	-	Open Motion Planning Library (Библиотека планирования движения)
PAI	-	Physical Artificial Intelligence (Физический искусственный интеллект)
PID	-	Proportional-Integral-Derivative Controller (Пропорционально-интегрально-дифференциальный регулятор)
ROS	-	Robot Operating System (Операционная система для роботов)
SAC	-	Soft Actor-Critic (Мягкий актор-критик)
TD3	-	Twin Delayed Deep Deterministic Policy Gradient (Двойной отложенный детерминированный градиент политики)
DDM	-	Data-Driven Model (Модель, основанная на данных)

ВВЕДЕНИЕ

Актуальность

Современные промышленные предприятия характеризуются высокой степенью автоматизации, однако во многих технологических процессах по-прежнему сохраняется значительная доля ручного труда, особенно на участках с повышенной опасностью. К таким зонам относятся операции, связанные с высокими температурами, токсичными веществами, повышенным уровнем шума, вибраций, а также риском механических повреждений. Работа в подобных условиях представляет угрозу для здоровья и жизни человека. А как результат увеличивает вероятность производственного травматизма и требует дополнительных затрат на обеспечение безопасности. В связи с этим одной из ключевых задач современной промышленности является минимизация участия человека в опасных и тяжелых производственных процессах. Эффективным решением данной проблемы является внедрение роботизированных манипуляторов, способных выполнять широкий спектр операций без непосредственного участия человека. Использование роботов позволяет существенно снизить уровень производственных рисков, повысить стабильность выполнения операций и обеспечить непрерывность технологического процесса. Помимо повышения безопасности, внедрение роботизированных систем имеет значительную экономическую выгоду. Автоматизация процессов позволяет сократить затраты на оплату труда, снизить издержки, связанные с производственными травмами и простоем оборудования. В добавок к этому повысить производительность за счет круглосуточной работы и высокой повторяемости действий. Роботизированные манипуляторы обеспечивают более точное и стабильное выполнение задач, что особенно важно в высокоточных производственных процессах, таких как сварка и маркировка разных объектов.

Однако традиционные подходы к управлению роботами, основанные на жестко заданных алгоритмах и заранее запрограммированных траекториях, обладают ограниченной гибкостью. В условиях реального производства часто возникают неопределенности: изменение положения объектов, необходимость адаптации к новым задачам. В таких ситуациях классические методы управления требуют сложной перенастройки и не обеспечивают достаточной адаптивности.

В последние годы перспективным направлением является использование методов глубокого обучения с подкреплением, позволяющих роботам самостоятельно обучаться эффективным стратегиям управления на основе взаимодействия со средой. Такие методы позволяют учитывать сложные динамические процессы и адаптироваться к изменяющимся условиям, что особенно важно для работы в реальных промышленных

сценариях. Дополнительным преимуществом является возможность интеграции с методами компьютерного зрения, что позволяет роботам ориентироваться в пространстве и выполнять задачи без точной предварительной настройки.

Таким образом, разработка интеллектуальных систем управления роботизированными манипуляторами, способных эффективно работать в опасных и динамически изменяющихся условиях, является актуальной задачей. Решение данной задачи позволит не только повысить уровень безопасности на производстве, но и обеспечить значительный экономический эффект за счет повышения эффективности, надежности и автономности технологических процессов.

Основной **целью** диссертационного исследования является разработка интеллектуальной системы управления роботизированным манипулятором на основе методов глубокого обучения с подкреплением, обеспечивающей адаптивное и высокоточное выполнение промышленных задач в условиях заданной среды.

Для достижения поставленной цели необходимо было решить следующие **задачи:**

1) провести анализ современных методов глубокого обучения с подкреплением, применяемых для управления роботизированными манипуляторами в промышленных приложениях;

2) разработать виртуальную среду моделирования обеспечивающую воспроизведение типовых промышленных задач (сварка и маркировка);

3) сформировать пространство состояний и действий для роботизированного манипулятора, включая параметры суставов, положения объектов и целевые состояния;

4) разработать функцию вознаграждения, учитывающую точность выполнения задачи, эффективность траектории, избегание столкновений и устойчивость управления;

5) реализовать и обучить модели глубокого обучения с подкреплением для управления манипулятором;

6) провести экспериментальное исследование обученных моделей в симуляционной среде с оценкой качества управления по метрикам точности, времени выполнения и устойчивости;

7) выполнить оптимизацию архитектуры модели и параметров обучения для повышения эффективности и обобщающей способности системы;

8) провести валидацию разработанного подхода на реальных или приближенных к реальным условиям, включая интеграцию с промышленным роботизированным оборудованием или цифровым двойником;

Объектом исследования является роботизированные манипуляторы, применяемые в промышленных задачах автоматизации (перемещение, захват и обработка объектов).

Предметом исследования является методы и алгоритмы интеллектуального управления роботизированными манипуляторами, обеспечивающие адаптивное планирование траекторий, точное позиционирование и эффективное выполнение задач в динамических условиях промышленной среды на основе глубокого обучения с подкреплением, оптимизационных подходов.

Методологическую основу работы составляет разработка и исследование гибридной системы управления роботизированным манипулятором, объединяющей методы обучения с подкреплением и предиктивного управления. В рамках исследования выполнен анализ современных подходов к управлению роботизированными системами, включая классические методы планирования траекторий, алгоритмы оптимального управления и методы глубокого обучения с подкреплением. Для моделирования динамики манипулятора и построения траекторий использованы кинематические и геометрические модели движения в трехмерном пространстве с учетом ограничений по скорости, ускорению и рабочей области робота. Для адаптивного управления и принятия решений использованы алгоритмы глубокого обучения с подкреплением DDPG и TD3. А также иерархические и многоуровневые схемы управления. Экспериментальная часть исследования реализована в симуляционной среде, предназначенной для воспроизведения сценариев роботизированной манипуляции. Для проверки эффективности разработанных подходов использованы задачи дуговой сварки и промышленной маркировки, требующие высокой точности позиционирования и плавности движения. Оценка эффективности разработанных моделей выполнена на основе экспериментальных исследований с использованием метрик средней абсолютной ошибки, максимального отклонения траектории, показателей ускорения, стабильности движения и успешности выполнения задачи. Сравнительный анализ результатов позволил определить влияние параметров MPC и алгоритмов обучения с подкреплением на качество управления роботизированным манипулятором в промышленных приложениях.

Научные положения, выносимые на защиту:

1) Разработанная симуляционная среда для обучения роботизированного манипулятора, обеспечивающая моделирование промышленных сценариев (захват, перемещение и сортировка объектов) с учётом кинематических и динамических ограничений, приближенных к реальным условиям эксплуатации.

2) Архитектура модели глубокого обучения с подкреплением, учитывающая неопределённость положения объектов, шум сенсорных данных и вариативность динамики системы, что обеспечивает устойчивость управления и высокую обобщающую способность в изменяющихся условиях.

3) Разработанная модель управления роботизированным манипулятором на основе обучения с подкреплением, позволяющая эффективно решать совокупность задач позиционирования, захвата и перемещения объектов в непрерывном пространстве состояний и действий.

4) Предложенный подход к интеграции алгоритмов обучения с подкреплением в систему управления коллаборативным роботом, обеспечивающий адаптивное выполнение различных промышленных задач и возможность практического применения разработанных методов.

Основные результаты исследования

1) Выполнен комплексный анализ современных методов управления, включая классические подходы теории управления, методы Model Predictive Control и алгоритмы глубокого обучения с подкреплением. Проведённый анализ позволил выявить ограничения традиционных методов, связанные с недостаточной адаптивностью к динамически изменяющимся условиям промышленной среды, а также обосновать необходимость разработки интеллектуальных систем управления, сочетающих прогнозирующее управление и обучение с подкреплением.

2) Разработана гибридная архитектура управления роботизированным манипулятором, объединяющая методы Model Predictive Control и Deep Reinforcement Learning, что обеспечивает одновременно высокую точность траекторного движения, устойчивость управления и способность адаптации к неопределённым условиям промышленной среды.

3) Создана симуляционная среда для моделирования задач роботизированной манипуляции, включающая кинематическую модель манипулятора, систему генерации объектов взаимодействия. Разработанная среда позволяет проводить обучение и тестирование алгоритмов управления в условиях, приближенных к реальным промышленным сценариям.

4) Реализованы и исследованы алгоритмы глубокого обучения с подкреплением DDPG, TD3 и иерархический TD3 для управления роботизированным манипулятором в задачах построения траектории и взаимодействия с объектами. Проведено сравнительное исследование алгоритмов по критериям устойчивости обучения, точности позиционирования, скорости сходимости и способности избегания препятствий.

5) Разработан метод оптимизации параметров для построения трёхмерных траекторий движения манипулятора. Выполнен поиск оптимальных параметров предиктивного управления с использованием

Парето-фронта и многокритериальной оптимизации, что позволило определить конфигурации, обеспечивающие минимальную среднюю ошибку, снижение ускорений и повышение плавности траектории.

6) Проведено сравнительное исследование методов генерации траекторий, включая MPC и альтернативные подходы на основе сплайнов и классических методов планирования движения. Установлено, что применение MPC обеспечивает более высокую точность следования траектории и устойчивость в условиях ограничений динамики манипулятора.

7) Разработана функция награды для задач обучения с подкреплением, учитывающая расстояние до целевой точки. Предложенная функция позволила повысить эффективность обучения агента и обеспечить формирование устойчивых стратегий поведения в сложной среде.

8) Проведены вычислительные эксперименты в симуляционной среде для задач роботизированной сборки, маркировки и дуговой сварки. Полученные результаты подтвердили способность разработанной системы обеспечивать точное позиционирование, плавное движение и адаптацию к изменению конфигурации среды.

9) Выполнен анализ результатов применения гибридной системы RL и MPC в промышленных сценариях. Установлено, что комбинированное использование предиктивного управления и обучения с подкреплением позволяет снизить ошибку позиционирования, повысить стабильность движения и уменьшить вероятность столкновений по сравнению с использованием отдельных методов управления.

10) Подтверждена практическая применимость разработанного подхода для задач промышленной робототехники, включая операции дуговой сварки и маркировки поверхностей. Полученные результаты демонстрируют перспективность интеграции методов глубокого обучения с подкреплением и предиктивного управления для создания интеллектуальных роботизированных систем нового поколения.

Научная новизна работы исследования заключается в разработке, специализированной симуляционной среды для обучения роботизированного манипулятора, включающей широкий спектр промышленных сценариев, таких как захват, перемещение и сортировка объектов в условиях, приближенных к реальным производственным процессам. В рамках исследования предложена архитектура модели управления, учитывающая дополнительные факторы внешней среды, что позволило повысить устойчивость алгоритмов управления к изменяющимся условиям. Также разработана модель глубокого обучения с подкреплением для управления роботизированным манипулятором, обеспечивающая одновременное решение задач позиционирования, захвата и перемещения объектов с учётом кинематических и динамических ограничений роботизированной системы.

Дополнительно интегрирована система управления коллаборативным роботом, ориентированная на выполнение различных промышленных задач. Включая операции сборки, маркировки и дуговой сварки. Предложенный подход обеспечивает повышение точности управления, адаптивности и устойчивости роботизированной системы по сравнению с классическими методами управления и традиционными алгоритмами машинного обучения.

Практическая значимость диссертационной работы является обученная и оптимизированная модель управления роботизированным манипулятором на основе глубокого обучения с подкреплением, способная выполнять задачи захвата, перемещения и сортировки объектов с высокой точностью и устойчивостью. Разработанная симуляционная среда позволяет эффективно обучать модели в условиях, приближенных к реальным, что существенно снижает затраты на экспериментальную отладку и минимизирует риски при внедрении. Дополнительно реализована интеграция разработанных алгоритмов в систему управления коллаборативным роботом, что обеспечивает возможность их применения для широкого спектра промышленных операций. Практическая реализация результатов исследования демонстрирует, что предложенные методы позволяют повысить уровень автоматизации производственных процессов, снизить зависимость от ручного труда и увеличить точность выполнения технологических операций. Таким образом, разработанные методы и программные инструменты обладают высокой степенью готовности к внедрению и представляют значительный интерес для промышленного применения.

Теоретическая значимость диссертационной работы заключается в развитии и адаптации методов глубокого обучения с подкреплением для задач управления роботизированными манипуляторами в условиях промышленной эксплуатации. В отличие от большинства существующих подходов, ориентированных либо на классические методы управления, либо на изолированное применение алгоритмов обучения с подкреплением, в данном исследовании расширена научно-методологическая база за счёт интеграции непрерывных методов обучения с подкреплением с учётом кинематических и динамических ограничений робототехнических систем. Существенным вкладом является формализация задачи управления манипулятором как многокритериальной оптимизационной задачи, учитывающей параметры как: точность позиционирования, плавность траектории и устойчивость к внешним возмущениям. Тем самым формируется теоретическая основа для построения интеллектуальных систем управления, способных функционировать в сложных и неопределённых промышленных средах.

Достоверность полученных результатов и обоснованность научных положений диссертации подтверждаются корректным применением методов теории управления, математического моделирования, кинематики

роботизированных систем, а также алгоритмов глубокого обучения с подкреплением и предиктивного управления. Особую значимость имеет использование кинематически обоснованных моделей движения манипулятора с учетом ограничений динамики, ускорений и взаимодействия с объектами среды, что обеспечивает адекватность разработанных алгоритмов реальным промышленным условиям. Эмпирической основой исследования послужила специализированная симуляционная среда, включающая задачи захвата, перемещения объектов. В среде учитывались геометрия рабочего пространства, препятствия, шум сенсорных данных и неопределенность положения объектов, что позволило обеспечить воспроизводимость вычислительных экспериментов и провести комплексное тестирование алгоритмов управления. Оценка эффективности разработанных моделей проводилась с использованием общепринятых метрик качества, включая среднюю ошибку, плавность траектории, ускорение и устойчивость движения. Для поиска оптимальных параметров MPC применялись методы многокритериальной оптимизации и анализ Парето-фронта. Достоверность результатов также подтверждается сравнительным анализом алгоритмов DDPG, TD3 и иерархического TD3, а также сравнением MPC с альтернативными методами построения траекторий. Экспериментально подтверждено преимущество предложенного гибридного подхода RL и MPC по точности, устойчивости и адаптивности управления в промышленных задачах робототехники.

Апробация диссертационной работы

Основные результаты работы были представлены и докладывались на следующих научных мероприятиях:

1) Momynkulov, Z., Tursynova, A., Olzhayev, O., Ikramov, A., Ibrayev, S., & Omarov, B. (2025). Three-Dimensional trajectory planning for robotic manipulators using model predictive control and point cloud optimization. Computer Modeling in Engineering & Sciences, 145(1), 891–918. https://doi.org/10.32604/cmes.2025.068615;

2) S. Ibrayev, B. Omarov, A. Ibrayeva, and Z. Momynkulov, “DeepSurNET-NSGA II: Deep Surrogate Model-Assisted Multi-Objective Evolutionary Algorithm for enhancing leg linkage in walking robots,” Computers, Materials & Continua/Computers, Materials & Continua (Print), vol. 81, no. 1, pp. 229–249, Jan. 2024, doi: 10.32604/cmc.2024.053075;

3) Zeinel Momynkulov, Azhar Tursynova, Olzhas Olzhayev, Akhanseri Ikramov, Sayat Ibrayev, Amandyk Tuleshov, Batyrkhan Omarov. Pareto-Optimized Model Predictive Control for Real-Time 3D Trajectory Planning of Collaborative Robots. In Proceedings of PAMDAS 2025 - International conference on Physical Asset Management and Data Science, 17-18 Jul. 2025, Coimbra, Portugal;

4) Zeinel Momynkulov, Azhar Tursynova, Olzhas Olzhayev, Akhanseri Ikramov, Sayat Ibrayev, Amandyk Tuleshov, Batyrkhan Omarov. Trajectory Optimization for Collaborative Robots via the Deep Deterministic Policy Gradient Algorithm. In Proceedings of PAMDAS 2025 - International conference on Physical Asset Management and Data Science, 17-18 Jul. 2025, Coimbra, Portugal;

5) Z. Momynkulov & B. Omarov (2025). DDPG for Trajectory Generation. 10th International Conference on Digital Technologies in Education, Science and Industry (DTESI), 19-20 Nov. 2025, Almaty, Kazakhstan;

6) Z.Z. Momynkulov, O. M. Olzhayev, A. T. Tursynova, A. K. Tuleshov, & S. M. Ibrayev. (2025). CONSTRUCTION AND GENERATION OF OPTIMAL TRAJECTORIES USING THE DDPG REINFORCEMENT LEARNING ALGORITHM. Science and Technology of Kazakhstan, №4, 2025.

Связь данной работы с другими научно-исследовательскими работами.

Диссертационное исследование выполнено по программе грантового финансирования МНВО РК: «Разработка роботов, научно-техническое и программное обеспечение гибкой роботизации и промышленной автоматизации (RPA) автопромышленных предприятий Казахстана на основе искусственного интеллекта» (2024-2026 гг., № BR24992947). Исследование соответствует стратегическим приоритетам развития Республики Казахстан и способствует реализации Концепции развития искусственного интеллекта на 2024–2029 годы в части развития отечественных интеллектуальных технологий, робототехнических систем и методов машинного обучения для промышленной автоматизации. Разрабатываемые модели глубокого обучения с подкреплением и гибридные подходы управления роботизированными манипуляторами направлены на повышение уровня автономности, точности и безопасности промышленных процессов. Предложенные методы интеграции управления способствуют развитию национального научно-технического потенциала в области искусственного интеллекта, интеллектуальной робототехники и цифровой трансформации промышленности Казахстана.

Основное содержание диссертации

В данной диссертационной работе исследуется задача разработки интеллектуальной системы управления роботизированными манипуляторами на основе гибридного подхода, объединяющего методы обучения с подкреплением и предиктивного управления.

В первом разделе обосновывается актуальность автоматизации управления роботами в условиях перехода к Индустрии 4.0 и 5.0, формулируются цель и задачи исследования, а также определяется научная новизна и практическая значимость предложенного подхода.

Во втором разделе проводится комплексный обзор литературы, охватывающий развитие систем управления роботами, макроэкономические

аспекты роботизации и современные направления, такие как воплощенный интеллект и фундаментальные модели. Уделяется внимание интеграции больших языковых моделей в робототехнику, а также применению глубокого обучения с подкреплением для задач управления и оптимизации. Рассматриваются методы планирования траекторий, навигации и пространственного восприятия, включая подходы на основе MPC и DRL. Дополнительно анализируются задачи роботизированной сборки, и манипуляции, а также вопросы коллаборативной робототехники и обеспечения безопасности. Также упоминается об анализе экономической выгоде применение автономных систем в промышленных отраслях.

В третьем разделе подробно описывается методология исследования. Представляется концепция гибридного управления, в которой высокоуровневое планирование осуществляется с помощью методов обучения с подкреплением, а низкоуровневое управление реализуется через Model Predictive Control. Рассматривается разделение уровней управления и математические основы MPC, включая формализацию целевой функции и ограничений. Детально анализируются алгоритмы DDPG и TD3, а также их расширения, такие как иерархическое обучение и многоуровневые архитектуры. Формулируется задача построения траектории в трехмерном пространстве, задаются граничные условия, метрики оценки и методы оптимизации параметров, включая анализ Парето-фронта.

В четвертом разделе представлены результаты экспериментальных исследований. Проводится анализ эффективности MPC при различных критериях оптимизации, включая минимизацию ошибки и ускорения. Рассматриваются результаты обучения с подкреплением и их поведение в симуляционной среде. Выполняется комплексный анализ полученных данных, включая сравнение различных подходов и оценку устойчивости системы. Также демонстрируется применение разработанных методов в прикладных задачах, таких как дуговая сварка и маркировка, с учетом реальных технических характеристик роботов.

В пятом разделе подводятся итоги работы. Подтверждается достижение поставленной цели, формулируются основные научные и практические результаты, а также делаются выводы о перспективности применения гибридных методов RL и MPC в задачах промышленной робототехники. Отмечается потенциал дальнейшего развития системы в направлении повышения автономности, расширения области применения и интеграции с реальными роботизированными платформами.

1 АНАЛИЗ СОВРЕМЕННЫХ МЕТОДОВ УПРАВЛЕНИЯ РОБОТИЗИРОВАННЫМИ СИСТЕМАМИ

В данной главе представлен анализ современного состояния исследований в области разработки управления для роботизированными манипуляторами в промышленных приложениях. Главной целью обзора является систематизация существующих подходов к построению интеллектуальных систем управления. Способность обеспечивать высокоточное выполнение манипуляционных задач. А также выявление ключевых научно-технических ограничений, определяющих актуальность и направление настоящего диссертационного исследования.

Логика изложения материала выстроена от фундаментальных основ теории управления и обучения с подкреплением к анализу современных алгоритмических и архитектурных решений. В начале главы рассматривается значимость автоматизации манипуляционных операций в контексте перехода к Индустрия 4.0 и Индустрия 5.0, где роботизированные системы становятся ключевым элементом гибких производственных линий. Подчеркивается роль интеллектуальных манипуляторов в задачах сборки, транспортировки, сварки и прецизионной обработки, а также анализируются требования к таким системам: точность позиционирования, устойчивость к неопределенностям среды, способность к обучению и адаптации в реальном времени.

Далее осуществляется ретроспективный анализ эволюции методов управления роботами: от классических подходов, основанных на аналитической кинематике и динамике, включая методы на основе Denavit-Hartenberg parameters и Product of Exponentials, к современным стратегиям, использующим обучение с подкреплением. Рассматривается переход от дискретных алгоритмов, таких как Q-learning, к непрерывным методам управления, включая Deep Deterministic Policy Gradient, Twin Delayed DDPG и Soft Actor Critic, которые обеспечивают эффективную работу в высокоразмерных непрерывных пространствах состояний и действий.

В рамках обзора также анализируются существующие работы, базы и критерии, используемые для оценки эффективности алгоритмов управления роботами. Отмечаются ограничения текущих подходов. Включая высокую вычислительную сложность, нестабильность обучения, чувствительность к выбору функции награды и трудности интерпретации поведения агента.

В обзор включены только рецензируемые научные публикации. Статьи журналов и материалы ведущих конференций по робототехнике и управлению, а также обзорные статьи (surveys, reviews) дополнительно к этому систематические обзоры. Проверка источников выполнена в пользу издательских площадок и архивов, связанных с IEEE, Springer и Elsevier и не только, с обязательной фиксацией DOI [4].

1.1 Развитие систем управления роботами

Эволюция управления роботами в промышленности прошла путь от классических контуров обратной связи и регуляторов PID к более современным методам. Методам, сочетающим модельные подходы как оптимизацию, вероятностные методы планирования, а также обучение (imitation learning, reinforcement learning и глубокие модели). На ранних этапах ключевыми факторами внедрения были устойчивость, простота настройки и возможность гарантировать качество при повторяемых операциях. В результате PID и каскадные сервоконтуры на уровне приводов стали базовой промышленной практикой, а дальнейшие достижения строились поверх этой основы (модели кинематики и динамики, а также архитектуры управления промышленными роботами) [1]. Ключевой перелом в 1980-х связан с формализацией управляемого взаимодействия с внешней средой: гибридное управление положением и силой и импедансное управление (стратегия управления взаимодействием робота с окружающей средой, контролирующая динамическую зависимость между позицией и силой) сделали возможными стабильные контактные операции (сборка, шлифование, полировка) позиционирования. Именно здесь управление стало тесно связано с требованиями к безопасности, надежности и контролю качества. По сей день остается центральным для промышленности и в 2020-х, особенно в коллаборативной робототехнике и человеко роботном взаимодействии [2]. С 1990-х усилилась роль «управления с восприятием» сокращенно визуальное управление или visual servoing закрепил парадигму «перцепция в контуре управления», где датчики и обработка сигналов становятся частью управления. Параллельно развивались методы планирования движения: sampling based (kinodynamic planning) и trajectory optimization (CHOMP или TrajOpt), что позволило переносить часть сложности из низкоуровневого управления в уровень планирования траекторий и ограничений [3]. В 2010-е и 2020-е промышленный фокус расширился от «одного робота» к целым системам: распределенное и кооперативное управление. Другими словами, флоты мобильных роботов, много роботов в одной ячейке, кооперативная манипуляция. А также интеграция через стандартизованные программные платформы. ROS и особенно ROS 2 стали важными слоями межкомпонентной интеграции и ускорения разработки, а ROS Industrial выступил мостом к промышленным задачам. Однако промышленная эксплуатация требует детерминизма, безопасности и поддержки стандартов, что стимулировало развитие безопасного Model Predictive Control (MPC) и data driven MPC с гарантиями. И вырос спрос на систематические обзоры требований безопасности коллаборативных систем (ISO 15066) [4].

Для методов обучения характерна асимметрия между исследовательской демонстрацией и промышленным внедрением. Обзоры по RL и deep RL фиксируют рост реальных успешных кейсов, но также подчеркивают ограничения. Ограничение как данные, безопасность, перенос из симуляции, гарантии. В промышленности методы Imitation Learning и data driven модели активно комбинируются с оптимизацией и MPC, чтобы сохранить предсказуемость и обеспечить ограничения, а deep learning часто используется как компонент восприятия, моделирования или вспомогательной адаптации, а не как единственный способ управления [5].

Хронологическая рамка охватывает ранние теоретические основы. А именно года 1940-е и 1950-е, включая PID и кинематическую нотацию. Далее ориентируется на момент, которые непосредственно повлияли на робототехнику и промышленное применение. А именно такие понятия как контактное управление, модельное управление динамикой, восприятие в контуре, планирование, распределенные системы, learning based методы, а также программные платформы и безопасность коллаборативной робототехники [1]. В таблице они используются как индикатор научного влияния, но не как критерий промышленной готовности. Промышленное внедрение также определяется сертификацией, требованиям безопасности, стоимостью интеграции и совместимостью с существующей инфраструктурой [6].

Ниже приведена таблица 1 ключевых этапов методов на промышленные задачи указанного периода времени. Формулировки в ней соответствуют «этапам» в литературе и отражают переходы от классического управления к современным системам. А именно как контактное управление, модельным методам, планированию, распределенным системам и методам основанными на обучении (learning based) подходам. Указания на публикации и DOI приведены в следующей таблице «Ключевые публикации» и в тексте обзора [1].

Таблица 1 – Методы, применяемые в промышленных задачах в указанном периоде

Период	Методы в литературе и практике	Промышленные задачи
1940-е - 1970-е	Классическая обратная связь, PID, формирование инженерных практик настройки; развитие формализма моделей	Стабилизация приводов, простые позиционные операции, регуляторные контуры в мехатронике

Продолжение таблицы 1

1980-е	Контактное управление (hybrid position, force, impedance), оптимальные по времени траектории, operational space; усиление модели динамики	Сборка с контактом, шлифование, полировка, обработка с силовым взаимодействием, ускорение циклов
1990-е	Робастные и адаптивные методы, visual servoing; усиление роли датчиков и реального времени	Компенсация неопределенностей, управление по зрению и датчикам в контуре, повышение устойчивости к вариациям
2000-е	Sampling-based планирование в высоких размерностях; multi-robot теория (task allocation)	Навигация и координация, ранние флоты, сложные ячейки и логистика
2010-е	Trajectory optimization и программные библиотеки (OMPL); рост LfD и RL; развитие middleware	Интеграция perception-planning-control, ускорение разработки, прототипирование промышленных кейсов
2020-е - наст. время	Safety, HRC и стандарты; MPC и data-driven MPC с ограничениями; ROS 2; обзоры deep RL реальных успехов	Коллаборативные ячейки, безопасная совместная работа, микро сборка и гибкие линии, автономная логистика

Классическое управление и PID в роботах обычно реализуются в каскадных контурах уровня привода и уровня суставов, обеспечивая стабилизацию и хорошую повторяемость. В промышленности эта архитектура ценна тем, что позволяет инженеру прогнозировать поведение. В дополнение к этому калибровать контуры и обеспечивать стабильное качество цикла. Исторические работы по конфигурации и анализу архитектуры управления промышленными роботами показывают формирование структуры управления и сегментации функций между аппаратной частью. Частями как сервоконтуром и высокоуровневым планированием [7]. Модельное управление (model based control) расширило возможности. Кинематическая нотация и формализация моделей создали основу для вычисления прямой и обратной кинематики, динамики и компенсации нелинейностей. На этой базе возникли подходы и управление с учетом динамики, где задача формулируется в рабочем пространстве. Силы и движения могут согласованно задаваться в единой структуре. В промышленности модельные методы особенно важны, когда нужно повысить точность при высокой скорости или учитывать ограничения приводов и обеспечивать устойчивое поведение при

изменении нагрузки [8]. Оптимальное управление и оптимальное по времени (time-optimal) задачи в робототехнике сыграли ключевую роль. Особенно в ускорении циклов и в инженерной постановке «минимального времени при ограничениях приводов». В промышленности эти идеи проявляются как оптимизация профилей скорости по траектории, сокращение такта и баланс между скоростью и качеством. Классические решения time-optimal вдоль заданного пути стали важным звеном между планированием траектории и контуром управления. Особенно в задачах подбора и обработке поверхностей, где траектория известна, но скорость по ней нужно подбирать под ограничения [9]. Контактное управление и силовое взаимодействие являются центральными для многих промышленных операций вне «свободного пространства».

Гибридное управление положением и силой формализует разделение степеней свободы на управляемые по положению и по силе, что критично для сборки и операций с контактом. Импедансное управление или метод управления роботами, который настраивает динамическое взаимодействие между манипулятором и окружающей средой включая в себя регуляцию жесткости, демпфирование и инерцию (impedance control), в том числе специально ориентированное на промышленных роботов, предложило формальную настройку «механического поведения». Робот как часть податливой системы, что снижает риск повреждения детали и повышает устойчивость процесса при вариациях [2]. Развитие physical human-robot interaction и коллаборативной робототехники усилило требования к слоям безопасности. Мониторинг расстояний, ограничению энергии и взаимодействию подвергнлись к сильному контролю. pHRI систематизировал взаимодействие «совместного пространства» человека и робота. Современные промышленные кейсы демонстрируют архитектуры, где безопасность достигается как комбинация сенсорики, мониторинга и управляющих ограничений.

Параллельно формируются обзорные работы, которые интерпретируют требования ISO 15066 как инженерные механизмы обеспечения безопасности (анализ опасностей, риска и ограничения). Что как результат, напрямую влияет на практику внедрения коллаборативных возможностей [10]. Адаптивное управление исторически нацелено на компенсацию неопределенности параметров и вариаций, что полезно в реальных производственных условиях (износ, изменение массы, трение, температурные эффекты). В робототехнике значимую роль сыграли обучающие и адаптивные методы управления, которые формально доказывают сходимость и устойчивость при неопределенности. А также обзоры, разъясняющие структуру адаптивного обратного уравнения динамики и управления для роботов. В промышленности адаптация часто внедряется осторожно: как

ограниченная подстройка параметров, как надстройка над моделью, либо в виде learning based компенсации, чтобы не потерять предсказуемость [11]. Робастное управление в промышленной робототехнике рассматривается как путь к устойчивости при ограниченных моделях и внешних возмущениях. Промышленный смысл здесь в том, чтобы гарантировать приемлемое качество при вариациях условий без постоянной перенастройки. В литературе по робастному управлению роботами регулярно подчеркивается задача подавления неопределенностей и сохранения устойчивости. Что становится особенно важным на фоне усложнения механизмов и требований к скорости [12]. Управление с восприятием в контуре, прежде всего visual servoing, формирует класс задач, где ошибка определяется в изображении. А управление строится с учетом геометрии формирования изображения и чувствительности измерений. Для промышленности это критично в операциях, где точность позиционирования определяется не только энкодерами (датчиками обратной связи). В частности, положением детали, допусками, а также изменениями сцены (сборка, вставка, контроль качества, манипуляции с неопределенными объектами и т.п.) [13].

Планирование траекторий и движение в высоких размерностях стали самостоятельным «слоем» между задачей и управлением. Вероятностные дорожные карты представляет класс sampling-based методов, которые позволяют строить граф достижимых конфигураций, a kinodynamic planning и Rapidly-exploring Random Tree (RRT) методы заносят в планирование динамические ограничения. В индустрии это проявляется как рост роли оффлайн-локального и онлайн планирования, а также как интеграция конвейер планирования движения в программные платформы. В 2010-х trajectory optimization методы (CHOMP, затем TrajOpt) усилили практику «оптимизировать траекторию под ограничения и столкновения» и стали частью инструментальных цепочек планирования для манипуляторов [14].

Распределенное и кооперативное управление в промышленности проявляется в двух ключевых сценариях. Первый это комплексы мобильных роботов в логистике и складских системах, где необходимо совмещать навигацию, управление и безопасность. Второй это кооперативные системы (несколько манипуляторов, мобильные манипуляторы, совместный перенос объектов), где управление связано с оптимизацией, согласованием ограничений и коммуникацией. Обзор по кооперативной манипуляции систематизирует моделирование и контроль таких систем, а прикладные обзоры по роботизированным складским системам фиксируют архитектуры и алгоритмы, применяемые в реальных развертываниях [15].

Learning based методы в управлении роботами исторически включали итеративные способы (iterative learning control) для повторяемых циклов (сценарий, когда операция повторяется тысячи раз), затем learning from

demonstration (LfD) и imitation learning для снижения стоимости программирования и переноса навыков. а потом reinforcement learning и deep RL для получения политик на сложных распределениях состояния. Обзорные статьи по LfD и IL в робототехнике подчеркивают практические мотивы. Традиционное программирование требует высокой экспертизы и времени, а демонстрация и имитация снижают барьер разработки, но создают задачи обобщения и безопасности. Обзоры RL и deep RL фиксируют рост успешных случаев использовании, однако промышленное внедрение обычно требует дополнительных слоев: ограничений, мониторинга, безопасного обучения и интеграции с планированием [16].

MPC и его вариации в промышленной робототехнике становятся одним из наиболее практичных «мостов» между теорией управления и эксплуатационными требованиями. Предиктивные модели, естественно, учитывают ограничения по состоянию и управлению. И может работать как траекторный регулятор. И может быть дополнен data-driven моделями. Современные работы демонстрируют модельные и data-driven MPC для манипуляторов, а также направление safe data-driven MPC, где явно ставится вопрос гарантий безопасности при сложной динамике. В промышленной перспективе MPC часто используется как высокоуровневый слой или как «реактивное планирование», оставляя низкоуровневые контуры стабильными [17]. Наконец, промышленное внедрение определяется инженерной экосистемой и стандартами. По мере роста сложности систем становится критичным программный слой. Включая в себя возможности как масштабируемость и реальное применение в разных задачах. Промышленные кейсы показывают применение ROS 2 для микросборки и автоматизации. К дополнению к вышесказанному развитие ROS Industrial как механизма переноса исследовательских инструментов в индустриальные применения [4]. В приведенной таблице 2, представлен список статей по времени с вкладом и применением в промышленности.

Таблица 2 – Список статей по времени с вкладом и применением в промышленности

Авторы	Год	Метод или тема	Вклад	Применение в промышленности
J. G. Ziegler и соавт.	1942	PID, настройка	Практические правила настройки PID как инженерная процедура	Сервоконтуры приводов, базовая регуляция в мехатронике
J. Denavit и соавт.	1955	Кинематика	Нотация DH как опора для модельных методов	Модели манипуляторов, программирование траекторий, калибровка

Продолжение таблицы 2

J. Y. S. Luh	1983	Архитектура промышленных роботов	Обзор «анатомии» промышленного робота и его систем управления	Инженерная структура контроллеров и интерфейсов
M. H. Raibert и соавт.	1981	Hybrid position/force	Формализация одновременного управления положением и силой	Сборка, вставка, работа с контактами и допусками
N. Hogan	1984	Impedance control	Импедансное управление, ориентированное на промышленных роботов	Полировка, шлифование, сборка с контактом, снижение дефектов
J. E. Bobrow и соавт.	1985	Оптимальное управление	Time-optimal движение по заданной траектории при ограничениях приводов	Сокращение такта, оптимизация скорости по траектории
O. Khatib	1987	Model-based, operational space	Единая формулировка движения и сил в operational space	Высокоточное управление в рабочем пространстве, силовые задачи
R. Ortega и соавт.	1989	Adaptive control	Обзор адаптивного управления rigid robots и inverse dynamics схем	Компенсация параметрических вариаций, нагрузок и трения
M. W. Spong	1992	Robust control	Робастное управление роботами-манипуляторами в условиях неопределенности	Повышение стабильности качества при вариативной среде
S. Hutchinson и соавт.	1996	Visual servoing	Учебный обзор visual servoing, таксономия подходов	Сборка по зрению, коррекция траектории по датчикам
S. Chiaverini и соавт.	1999	Interaction control	Обзор схем управления взаимодействием с экспериментальным сравнением	Контактные операции и критерии внедрения схем управления
L. E. Kavraki и соавт.	1996	PRM planning	Probabilistic roadmaps как базовый sampling-based метод	Offline planning, планирование в сложной геометрии
S. M. LaValle и соавт.	2001	Kinodynamic planning (RRT family)	Рандомизированное kinodynamic planning в высоких размерностях	Планирование движения с динамическими ограничениями
B. P. Gerkey и соавт.	2004	Multi-robot task allocation	Таксономия Multi-Robot Task Allocation задач и формальный анализ	Диспетчеризация флотов и кооперация роботов на производстве
R. Olfati-Saber и соавт.	2007	Consensus, distributed control	Теоретическая рамка консенсуса и кооперации в сетевых системах	Координация флотов, распределенное принятие решений
A. De Santis и соавт.	2008	pHRI обзор	Карта pHRI проблем, надежности и безопасности	Коллаборативные ячейки, совместные операции, требования безопасности
B. D. Argall и соавт.	2009	Learning from Demonstration	Обзор LfD и постановка задач обобщения и интерфейсов	Ускорение программирования, перенос навыков в манипуляции
T. Osa и соавт.	2018	Imitation learning	Алгоритмическая перспектива IL и классификация подходов	Программирование навыков, сборка, контактные операции
J. Kober и соавт.	2013	Reinforcement learning	Обзор RL в робототехнике, постановка задач и методов	Обучаемые стратегии, но с учетом ограничений безопасности
C. Tang и соавт.	2025	Deep RL survey	Обзор реальных успешных применений deep RL в роботах	Идентификация факторов успешного внедрения и ограничений

Продолжение таблицы 2

L. Jin и соавт.	2018	Neural network control	Обзор НС подходов к управлению манипуляторами	Компенсация неопределенности, модель-free компоненты
A. I. Károly и соавт.	2020	Deep learning survey	Обзор структур моделей и стратегий обучения DL в робототехнике	DL как компонент perception и моделирования
N. Ratliff и соавт.	2009	CHOMP	Trajectory optimization с градиентными методами	Манипуляция, генерация траекторий с качественными ограничениями
J. Schulman и соавт.	2014	TrajOpt (sequential convex)	Trajectory optimization через последовательную выпуклую оптимизацию	Быстрое планирование с проверкой коллизий, интеграция в пайплайны
I. A. Șucan и соавт.	2012	OMPL	Библиотека sampling-based планирования и инфраструктура	Индустриальные пайплайны планирования, переносимость
A. Carron и соавт.	2019	Data-driven MPC	MPC с data-driven моделью ошибки и обратной динамикой	Точное слежение у податливых и легких манипуляторов
M. Krämer и соавт.	2020	MPC for cobot	Online trajectory optimization на основе MPC для коллаборативных манипуляторов	HRC сценарии, ограничения и предсказуемость
I. Mitsioni и соавт.	2023	Safe data-driven MPC	Безопасный data-driven MPC для сложной динамики	Safety-first подходы в автономных системах, релевантно индустрии
P. Chemweno и соавт.	2020	Safety, ISO 15066 review	Обзор требований безопасности коллаборативных систем и design safeguards	Проектирование HRC ячеек, риск-анализ и инженерные меры
E. Mariotti и соавт.	2019	Admittance control in pHRI	Admittance control с F/T сенсором на промышленном роботе для pHRI	Hand-guiding, безопасное взаимодействие, промышленная интеграция
E. Magrini и соавт.	2020	HRC в открытых ячейках	Архитектура безопасности и взаимодействия без классических клеток	Поверхностная обработка с участием человека, мониторинг дистанции
Z. Feng и соавт.	2020	Multi-robot manipulation survey	Обзор моделирования, управления и оптимизации кооперативной манипуляции	Кооперативные манипуляторы, мобильная манипуляция, логистика
J. Alonso-Mora и соавт.	2017	Multi-robot optimization	Constrained optimization для формаций и переноса объектов	Кооперация роботов в динамической среде, логистика
Í. R. da Costa Barros и соавт.	2021	Warehouse fleets	Обзор RMFS (роботы в складах) и архитектур	Склады, диспетчеризация, безопасность и эффективность
S. Macenski и соавт.	2022	ROS 2	Обзор архитектуры ROS 2 и кейсы применения	Production-ориентированная интеграция и переносимость
Min Ling Chan и соавт.	2017	ROS-Industrial	Развитие ROS-Industrial как расширение промышленных применений	Инструменты интеграции, стандартизация разработки
Niklas Terei и соавт.	2024	ROS 2 in industry	Кейс ROS2 управления роботизированной микро сборкой	Микро-сборка, интеграция,ограничения и преимущества

Таблица-3 ориентирована на промышленный выбор методов, где важны: гарантии безопасности, предсказуемость, стоимость внедрения, требования к данным и вычислениям, а также совместимость с инфраструктурой (контроллеры, сенсоры, middleware).

Таблица 3 – Сравнение методов с учетом разных требований

Класс методов	Требует точной модели	Учет ограничений	Гарантируемость и верифицируемость	Требования к данным	Реальное время	Типичная индустриальная зрелость
PID и классическая обратная связь	нет	ограниченно	высокая (инженерная практика)	низкие	высокая	очень высокая
Model-based (inverse dynamics, operational space)	да	частично	средняя-высокая (при корректной модели)	низкие	высокая	высокая
Adaptive control	частично	частично	средняя (зависит от допущений)	низкие-средние	средняя-высокая	средняя
Robust control	да/ частично	частично	высокая по устойчивости однако сложность дизайна	низкие	высокая	средняя-высокая
Optimal control / time-optimal	да	да	средняя (качество зависит от модели и реализации)	низкие	средняя	средняя
MPC (включая NMPC)	да	да (сильная сторона)	высокая при корректной постановке, растет роль safe вариаций	низкие-средние	средняя	высокая в росте
Data-driven MPC / safe data-driven MPC	частично	да	от средней до высокой, если есть слой safety	средние	средняя	растущая
Impedance / admittance / hybrid force-position	да/частичн о	частично	высокая для контактных задач при корректной настройке	низкие	высокая	высокая
Visual servoing	да (геометрия)	частично	средняя (зависит от восприятия)	средние	средняя	высокая для выбранных сценариев
Motion planning (PRM/RRT/traj ectory optimization)	да /частично	да	средняя (зависит от проверки и модели)	низкие	средняя	высокая как слой над управлением
Distributed / cooperative control	частично	частично	средняя (коммуникация и топология)	средние	средняя	высокая в логистике и кооперации
Imitation learning / LfD	нет (может использовать модель)	частично	средняя, часто требует safety слоя	высокие (демонстрации)	средняя	растущая
RL / deep RL	нет (может использовать модель)	косвенно	низкая-средняя без safety слоя, в обзорах отмечены барьеры	очень высокие (опыт)	средняя	точечно, чаще в R&D
ROS 2 и промышленная интеграция	частично	инфраструктурно	повышает воспроизводимость и интеграцию системы	зависит от задачи	зависит от middlewa re	растущая

Ключевой практический вывод для промышленности состоит в том, что «лучший» метод редко выбирается изолированно. Наиболее устойчивые промышленные решения строятся как стек: стабильные низкоуровневые контуры (часто PID), затем слой модельного управления для ограничений. Затем планирование траекторий, а learning based методы используются для

ускорения программирования (LfD/IL), повышения робастности (data-driven коррекции) или в узких компонентах (например, восприятие и оценка состояния). Такой стек лучше согласуется с требованиями безопасности и эксплуатационной устойчивости, отраженными в обзорах по ISO 15066 и в практических HRC кейсах открытых ячеек.

1.2 Макроэкономическая и технологическая панорама роботизации

На современном этапе развития глобальной экономики промышленная робототехника переживает период фундаментальной трансформации. Данный переход, который в научной литературе классифицируется как сдвиг от парадигмы Индустрии 4.0 к концепции Индустрии 5.0. И характеризуется отходом от использования жестко запрограммированных, узкоспециализированных инструментов в пользу адаптивных, интеллектуальных систем [43]. Этот технологический скачок стимулируется глубокой интеграцией искусственного интеллекта, алгоритмов машинного обучения и архитектур восприятия (камер глубин, лидары). Что коренным образом меняет показатели эффективности, качества и параметры человеко-машинного взаимодействия в производственном секторе [44]. Согласно статистическим данным Международной федерации робототехники (IFR), представленным в аналитическом отчете World Robotics за 2025 год, глобальный спрос на промышленных роботов удвоился за последнее десятилетие [45]. Только в 2024 году было интегрировано 542000 новых роботизированных систем, что позволило преодолеть отметку в полмиллиона установок четвертый год подряд [45]. В результате общее число эксплуатируемых промышленных роботов достигло 4664000 единиц. Продемонстрировав уверенный рост на 9% по сравнению с предшествующим годом [45]. Анализ географического распределения робототехнических комплексов выявляет безоговорочное доминирование стран Азиатско-Тихоокеанского региона, на долю которых в 2024 году пришлось 74% новых внедрений, в то время как доля Европы составила лишь 16%. Северной и Южной Америки только 9%. Особого исследовательского внимания заслуживает внутренний рынок Китайской Народной Республики, который установил исторический рекорд в 295000 внедренных единиц оборудования. Число установок превысило отметку в 2 миллиона роботов, что делает его крупнейшим в мире. Примечательной тенденцией 2024-2025 годов стало то, что впервые китайские производители робототехники заняли 57% собственного внутреннего рынка, агрессивно вытесняя иностранных поставщиков, доля которых за прошедшее десятилетие сократилась. В то же время Япония сохраняет за собой статус второго по величине рынка в мире с

44500 новыми установками в 450500 единиц, активно интегрируя автоматизацию для преодоления вызовов общества [46].

Однако активное использование робототехники в современной промышленности выходит далеко за рамки простого наращивания количественных показателей. Научная литература 2024-2026 годов подчеркивает качественный рост. Производственная среда стала главным испытательным полигоном для концепции физического искусственного интеллекта или Physical AI (PAI) парадигмы. Предполагающей глубокое слияние систем с самообучающимися алгоритмами. В то время как традиционные технологии автоматизации прошлых десятилетий концентрировались на повышении производительности за счет монотонного масштабирования однотипных сборочных операций, современные тенденции делают акцент на достижении автономии, высокой адаптивности и устойчивого развития.

В последние годы научный интерес вокруг активного использования промышленных роботов сместился от оценки базовой автоматизации к анализу глубоко интегрированных, интеллектуальных и адаптивных систем. Обзор новейших рецензируемых статей за период 2024-2026 годов демонстрирует эту качественную трансформацию на микро и макроуровнях. В сфере когнитивного управления и интеграции искусственного интеллекта прорывные исследования направлены на создание автономных производственных ячеек. Авторы [47] разработали инновационную архитектуру AI Fuzzy Node RED, которая формирует замкнутый когнитивный цикл управления. Эта система объединяет машинное зрение и модули логического вывода для управления промышленными роботами в режиме реального времени, позволяя существенно снизить энергопотребление и позиционные ошибки при одновременном повышении адаптивности.

Критическим вектором активного применения роботов остается развитие коллаборативной робототехники в контексте перехода к Индустрии 5.0. В своей работе [48] подчеркивают, что безопасное взаимодействие человека и машины больше не может опираться исключительно на физические барьеры. Оно требует бесшовного сотрудничества, основанного на анализе и прогнозировании человеческого поведения. В развитие этой концепции авторы [49] предложили метод EDNPOS - открытую архитектуру распознавания действий человека. Что дает коботам возможность интуитивно считывать намерения операторов в динамичной среде. Наконец, интеграция больших языковых моделей (LLM) в системы HRC, в обзоре [50], открывает горизонты для мультимодального взаимодействия. Роботы получают способность понимать сложные голосовые и текстовые команды операторов прямо на сборочной линии. На системном уровне роботизация требует принципиально новых архитектурных решений. Robot Digital Twin [51]

исследуют технологию цифровых двойников для промышленных роботов. В частности, отмечая её роль как критического связующего звена между физической и цифровой средой для симуляции. А также предиктивной аналитики и оптимизации производственных циклов. Исследовательская группа [52] предложила концепцию умного реконфигурируемого производства, где робототехника обеспечивает гибкую и динамичную перенастройку производственных линий. Для максимизации экономической выгоды от такого гибридного подхода [53] разработали структуру, объединяющую классическую методологию и технологии Индустрии 4.0, минимизирующую потери и способствующую целям устойчивого производства.

Макроэкономические аспекты активного внедрения роботов получают все более мощную эмпирическую аргументацию. Масштабное исследование [54], основанное на данных 476 европейских предприятий, подтвердило, что активное использование промышленных роботов статистически значимо повышает производительность труда. А также снижает процент производственного брака как следствие оптимизации процессов, содействует внедрению технологий. В глобальном масштабе анализ [55] доказывает, что внедрение робототехники усиливает устойчивость и статус стран в глобальные производственные пути. И даёт развивающимся экономикам реальный шанс быстро подтянуть технологии и сделать шаг вперёд в развитии.

Фундаментальные различия между коботами и классическими промышленными манипуляторами детерминируют специфику их экономического применения. Аналитический синтез технической и экономической литературы позволяет структурировать эти различия для более прозрачного понимания их функциональных ниш в таблице 4.

Таблица 4 – Сравнение традиционных и коллаборативных роботов

Анализируемый параметр	Традиционные промышленные роботы (SCARA / 6-осевые тяжелые манипуляторы)	Коллаборативные роботы (Коботы)
Основное технологическое преимущество	Экстремальная скорость перемещения, высокая грузоподъемность и микронная точность позиционирования в плоскостных операциях.	Высокая степень безопасности, мобильность, универсальность для работы в непосредственном контакте с человеком.
Особенности конструкции	Жесткая кинематика, часто 4-осевая архитектура (SCARA), оптимизированная для скоростных операций в плоскостях X-Y с жесткостью по оси Z.	Гибкая 6-осевая или 7-осевая архитектура, повсеместное использование высокочувствительных датчиков крутящего момента в сочленениях.

Продолжение таблицы 4

Обеспечение безопасности	Полная физическая изоляция (клетки, защитные экраны, световые завесы), системы экстренного обесточивания.	Программируемое ограничение силы и скорости; мгновенная безопасная остановка при контакте с любым препятствием.
Базовая стоимость оборудования	приб. $17,500 - $22,500 USD (8,000,000 - 20,000,000 тг) базовая стоимость механизма без учета ограждений и интеграции.	приб. $45,000 - $60,000 USD (21,000,000 - 28,000,000 тг) включая встроенные системы безопасности.
Программирование и интеграция	Требуют написания кода на проприетарных языках, найма высокооплачиваемых системных интеграторов, сложность в перенастройке.	Интуитивно понятные графические интерфейсы (Drag-and-Drop), возможность обучения методом ручного ведения.
Оптимальные производственные ниши	Массовое производство с высокой повторяемостью, конвейерная сборка электроники, высокоскоростные задачи pick-and-place, работа в тяжелых условиях (сварка, литье).	Производственные линии формата High-Mix/Low-Volume, обслуживание станков (machine tending), инспекция качества, сборочные операции вместе с человеком.
Экономическая эффективность процессов	На 30% быстрее и на 50% дешевле при выполнении монотонных горизонтальных перемещений, чем многоосевые аналоги.	Скорость искусственно ограничена протоколами безопасности; требуют обязательной оценки рисков всего рабочего места, несмотря на безопасность самого робота.

Эмпирические данные недвусмысленно связывают роботизацию с драматическим повышением качества продукции. В рамках исследования было подтверждено снижение уровня производственного брака. Одного из критически важного индикатора производственной неэффективности. Роботы, особенно задействованные в процессах прямой обработки, полностью исключают влияние человеческого фактора (усталость, дрожание рук, невнимательность) обеспечивая микронную точность и 100% повторяемость операций от первой до миллионной детали. Выдающимся и детально документированным примером практической реализации этих принципов является эмпирическое исследование завода TJ Automobile Manufacturing Company. Обладающего производственной мощностью 100 000 коммерческих автомобилей в год. Процесс окраски кузовов на данном предприятии страдал от целого комплекса унаследованных проблем. Проблемы как несбалансированности загрузки персонала, высоких экологических издержек,

нестабильного качества распыления краски и длительного времени ожидания в буферах [54].

Таблица 5 – Эмпирические результаты оптимизации производства на заводе TJ Automobile Manufacturing Company

Метрика производственного процесса	Количественное изменение	Экономический и технологический эффект
Уровень брака (Scrap rate)	Снижение на 8%	Радикальное сокращение расхода материалов и стоимости утилизации брака.
Уровень дефектов (Defect rate)	Снижение на 15%	Повышение качества конечной продукции, рост показателя «выход годных деталей» и прибыльности предприятия.
Доля операций, добавляющих ценность (Value-added time)	Увеличение на 9,5%	Рост эффективности использования рабочего времени, повышение соотношения ценности ко времени производственного цикла.
Доля операций, не добавляющих ценность (Non-value-added time)	Снижение на 12,5%	Сокращение скрытых потерь (ожидание, избыточная транспортировка, перепроизводство).
Штат операторов сборочной линии	Сокращение на 18,8% (ликвидация 3 рабочих постов)	Прямая экономия фонда оплаты труда, устранение ручного труда в токсичных зонах окраски и доводки; высвобожденный персонал переведен на другие задачи.
Время производственного цикла	Сокращение на 53 минуты	Существенное ускорение пропускной способности завода.

1.3 Воплощенный интеллект и фундаментальные модели

В обзоре [56] исследуют десятилетнюю динамику применения искусственного интеллекта, используя различные методы и тематическое моделирование BERTopic для выявления ключевых векторов развития. Авторы убедительно демонстрируют, как генеративный ИИ выступает в роли

уникального аппарата для задач, выявляя взаимосвязанные предметные кластеры в секторах здравоохранения, инженерии и бизнес-аналитики. Их детальный анализ подчеркивает, что различные исследовательские темы, такие как автоматизация управления или устойчивое развитие, проходят абсолютно циклы и этапы технологической зрелости. Особое внимание в работе уделяется растущим этическим и методологическим вызовам, включая необходимость жесткого смягчения алгоритмической предвзятости и обеспечения прозрачности генеративных моделей в обществе. На основе анализа масштабного массива данных исследование формирует целостное представление о ключевых закономерностях и взаимосвязях в области. А именно теоретическую и практическую основу для научных разработок. Результаты работы ориентируют последующие исследования на этически обоснованное применение технологий и развитие комплексных подходов, направленных на обеспечение устойчивого баланса развития.

Оценивая современную робототехнику в производственном секторе, в работе [57] рассматривается пересечение воплощенного искусственного интеллекта и концепции Индустрии 5.0 с акцентом на человеко-ориентированные роботизированные системы. Отмечается, что интеграция интеллектуальных моделей и комплексных систем позволяет существенно изменить подход к управлению роботами и производственными линиями, делая их более адаптивными и ориентированными на взаимодействие с человеком. Показано, что методы машинного обучения играют ключевую роль в развитии гибких роботизированных комплексов, обеспечивая возможность мелкосерийного и персонализированного производства, что снижает избыточное производство и уменьшает объем отходов. В то же время в [57] поднимаются важные вопросы, связанные с безопасностью, конфиденциальностью данных и справедливостью алгоритмов, особенно в контексте взаимодействия человека и робота. Это требует разработки строгих стандартов и нормативных механизмов для надежного внедрения таких систем. В целом подчеркивается, что переход к Индустрии 5.0 в робототехнике невозможен без комплексного подхода, включающего как технологическое развитие роботизированных систем, так и учет человеческого фактора, включая подготовку специалистов и обеспечение безопасного взаимодействия с интеллектуальными машинами.

Формируя теоретическую основу современных робототехнических систем, в работе [58] воплощенный интеллект рассматривается как тесная взаимосвязь физической структуры робота, его действий, сенсорного восприятия и способности к обучению. Подчеркивается, что интеллектуальное поведение возникает не из отдельных алгоритмов, а из их совместного взаимодействия в реальной среде, что требует перехода от изолированных моделей обучения к интегрированным системам управления. Также

рассматриваются подходы к предварительному обучению, направленные на улучшение восприятия роботами окружающей среды за счет объединения нескольких сенсорных каналов, включая различные визуальные представления. Это позволяет значительно повысить точность распознавания и адаптацию к сложным условиям.

Экстраполируя данную концепцию на уровень промышленной робототехники, в работе [59] анализируется внедрение воплощенного интеллекта в системы умного производства, основанные на современных моделях искусственного интеллекта. Отмечается, что традиционные алгоритмы ограничены при работе в условиях. Тогда как робототехнические системы с элементами воплощенного интеллекта способны адаптироваться к изменяющимся задачам. Подчеркивается, что такие системы могут постепенно совершенствовать стратегии управления и принятия решений за счет непрерывного взаимодействия с производственной средой. Это особенно важно для гибких производственных линий, где требуется обработка разнообразных и индивидуализированных задач. В [59] также вводится концепция коллективного взаимодействия роботизированных систем. При которой автономные устройства обмениваются сенсорными данными через промышленный IoT, обеспечивая оптимизацию производственных процессов.

Для более глубокого анализа методов обучения робототехнических систем в работе [60] представлен сравнительный обзор симуляторов и базовых задач в области воплощенного интеллекта. Подчеркивается переход от традиционных моделей, обучаемых на статических данных, к агентам, которые получают знания через активное взаимодействие с окружающей средой, как в симуляции, так и в реальности. Проводится систематическая оценка современных симуляционных платформ по ряду ключевых критериев, включая физическую достоверность моделирования и сложность реализуемой кинематики роботов. Это позволяет определить, какие инструменты наиболее подходят для разработки и тестирования алгоритмов управления.

Рассматривая задачи физического взаимодействия в робототехнике, в работе [61] представлен обзор методов захвата деформируемых и сложных объектов. Подчеркивается, что применение предварительно обученных мультимодальных моделей позволяет формировать более эффективные функции вознаграждения. Что существенно ускоряет обучение алгоритмов с подкреплением и повышает стабильность захвата.

В [61] проводится четкое различие между имитационным обучением, основанным на демонстрациях человека, и автономными методами, где робот самостоятельно вырабатывает стратегии через пробу и ошибку. Такой подход открывает возможности для нахождения новых, ранее не заданных решений в задачах манипулирования. Отмечается тенденция к переходу к сквозному управлению, при котором робот напрямую преобразует сенсорные данные.

Например, изображения с камер, в управляющие воздействия на приводы. Это позволяет уменьшить зависимость от промежуточных эвристических моделей и повысить адаптивность системы. Решая задачи адаптации интеллектуальных систем к требованиям промышленной робототехники, в работе [62] предлагается многоуровневая архитектура фундаментальных моделей, ориентированная на использование в реальных производственных условиях. Подчеркивается, что прямое применение универсальных генеративных моделей в промышленности ограничено из-за высоких требований к надежности, безопасности и точности управления роботами.

В [62] предлагаемый подход основан на интеграции сенсорных данных, физических моделей и экспертных знаний, что позволяет формировать более интерпретируемые и устойчивые системы управления. Это особенно важно для роботизированных комплексов, где требуется строгий контроль параметров движения и взаимодействия с объектами.

1.4 Примеры интеграции языковых моделей в робототехнику

В контексте интеграции больших языковых моделей в робототехнику в работе [63] рассматриваются как новые возможности, так и ограничения подобных систем. Подчеркивается, что такие модели следует воспринимать не только как инструменты обработки текста, а как элементы сложных управляющих систем. Системы где внутренние представления связаны с физическим управлением роботами. Отмечается, что подобные архитектуры позволяют реализовать высокоуровневое планирование, логический вывод и частичную автоматизацию программирования, снижая нагрузку на инженеров. Особое внимание уделяется подходам, которые улучшают восприятие и повышают эффективность работы в неструктурированных средах [63]. Продолжая данное направление, в работе [64] проводится анализ влияния фундаментальных моделей на развитие когнитивных возможностей роботов. Рассматриваются ключевые задачи как: формирование функций вознаграждения, управление приводами, планирование действий и интерпретацию сцены. Подчеркивается важность внедрения элементов восприятия контекста через настройку моделей, что позволяет повысить устойчивость обучения. Также описываются распределенные робототехнические системы, где агенты координируют действия с использованием языковых моделей [64]. Гибридная архитектура демонстрируется на рисунке 1.

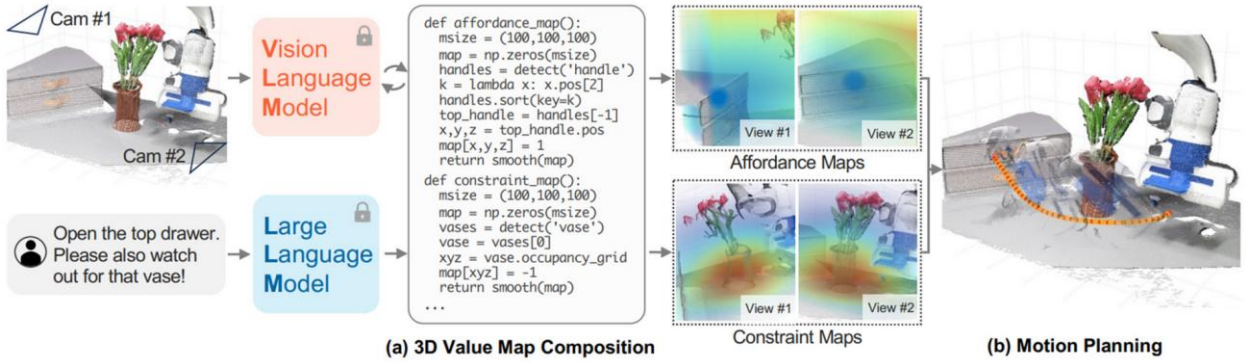

Рисунок 1 – Демонстрация гибрида архитектур для планирования движения [64]

С точки зрения инженерной реализации, в [65] представлена структурированная методология интеграции языковых моделей с сервисными роботами. Особое внимание уделяется преобразованию инструкций на естественном языке в формальные команды управления, а также обработке различных сенсорных данных, включая визуальные и тактильные сигналы. Приводятся практические подходы к построению архитектур и использованию промптов для управления поведением роботов [65]. Дополнительно в ряде работ [66, 67] показано, что использование визуально-языковых моделей повышает качество взаимодействия человека и робота, позволяя корректировать действия в реальном времени. В то же время исследования [68, 69] направлены на обеспечение безопасности, предлагая механизмы ограничения действий интеллектуальных агентов на уровне управления. В области практического применения [70] рассматриваются методы ускоренного развертывания роботизированных систем за счет использования языковых моделей, что особенно важно для динамичных сценариев. Наконец, в работах [71, 72] демонстрируется возможность автоматического планирования сложных последовательностей действий RoboGPT включая задачи сборки, с учетом требований безопасности и адаптации к окружающей среде.

1.5 Обзор работ с глубоким обучением и подкреплением

В области интеллектуального роботизированного производства в работе [73] представлен обзор применения методов машинного обучения для решения сложных задач автоматизации. Рассматривается концепция неявного контроля, при которой робот учитывает физические свойства материалов при выполнении операций. Также анализируются подходы к использованию обучения с подкреплением для оптимизации распределенных систем промышленного IoT. Который способствует повышению надежности и

безопасности вычислений. Подчеркивается необходимость перехода от жестко запрограммированных систем к адаптивным алгоритмам, способным работать в изменяющихся условиях производства. Развивая данное направление, в исследовании [74] анализируются возможности алгоритмов глубокого обучения с подкреплением в задачах оптимизации производственных процессов. Показано, что такие методы эффективно применяются для динамической настройки параметров обработки, а также для улучшения проектирования изделий. Особое внимание уделяется вопросам безопасности и необходимости включения обратной связи от человека при эксплуатации систем.

С точки зрения практической реализации, в работе [75] рассматриваются реальные примеры применения DRL в робототехнике. Отмечается высокая стоимость обучения в физической среде и проблемы с эффективностью выборки данных. Тем не менее, демонстрируются успехи в задачах манипуляции и сборки, а также подчеркивается необходимость разработки более устойчивых и универсальных архитектур.

На уровне управления производственными системами в [76] рассматриваются методы обучения с подкреплением для координации работы множества агентов в условиях умных фабрик. Анализируются архитектуры взаимодействия и подчеркиваются ограничения, связанные с нестабильностью коммуникаций в реальных условиях. Также выделяются перспективы использования объяснимого ИИ и методов автоматической диагностики.

Вопросы безопасности алгоритмов подробно изучаются в [77], где рассматриваются методы безопасного обучения с подкреплением. Подчеркивается конфликт между максимизацией награды и соблюдением ограничений, а также анализируются различные подходы к его разрешению. Отмечается важность разработки надежных алгоритмов для критических робототехнических приложений.

Наконец, в работе [78] рассматриваются методы применения DRL для управления манипуляторами. Подчеркивается проблема генерации данных и важность обобщающей способности моделей. В качестве эффективного решения предлагается комбинирование обучения с подкреплением с другими подходами, такими как имитационное обучение и использование симуляторов, что позволяет повысить точность и надежность робототехнических систем.

Дополняя теоретические основы обучения с подкреплением, в работе [79] представлен широкий обзор прикладных задач, решаемых с использованием RL. Рассматриваются базовые алгоритмы, такие как Q-learning и методы временных различий, с акцентом на итеративное обновление функции ценности. Показано, что методы на основе стратегий обладают преимуществами в непрерывных пространствах управления, что особенно важно для робототехнических задач. Также подчеркивается универсальность

RL для оптимизации сложных систем, включая адаптацию поведения в зависимости от внешних условий. В прикладных задачах робототехники в работах [80, 81] демонстрируется эффективность мультиагентных подходов DRL для объединения визуального восприятия и управления манипуляторами в динамичных средах. Это особенно актуально для человеко-ориентированных производственных процессов, где требуется высокая точность и безопасность взаимодействия. Дополнительно в исследованиях [82, 83] рассматриваются задачи динамического управления, включая прогнозирование траекторий движения объектов. В [84] показано, что DRL может успешно применяться для управления гибкими объектами, что требует учета сложной нелинейной динамики. В области координации роботизированных систем в [85] предлагаются методы синхронного управления несколькими роботами, направленные на предотвращение столкновений и оптимизацию энергозатрат, что критически важно для промышленных задач, таких как сварка.

Развивая направление фундаментальных моделей, в работах [86, 87] предложены архитектуры FOUNDER на рисунке 2, направленные на перенос знаний и формирование устойчивых представлений среды для принятия решений. Для повышения качества восприятия и манипуляции в [88, 89] вводятся методы FlowRAM и PDFactor, улучшающие пространственную осведомленность агентов.

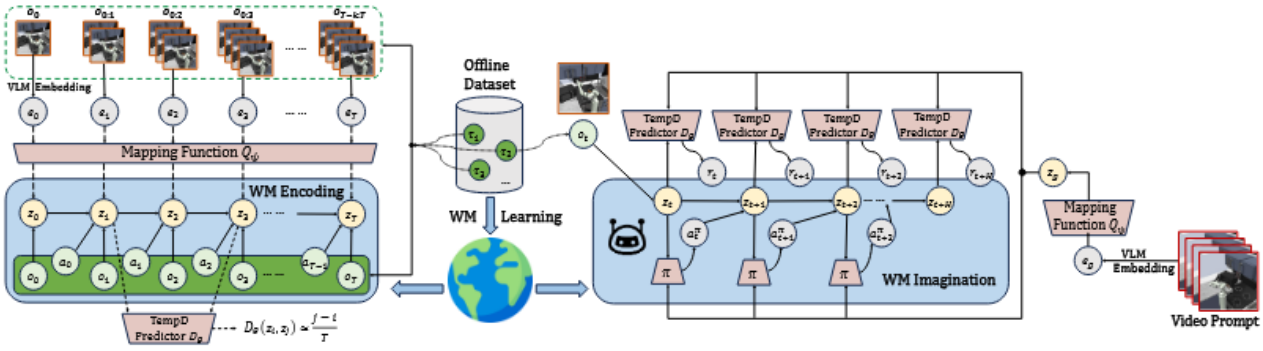

Рисунок 2 – Архитектура FOUNDER [86]

В задачах реального времени в [90] рассматриваются подходы к ускорению выполнения действий за счет пакетной генерации управляющих сигналов. Наконец, в работе [91] представлен бенчмарк для оценки способности робототехнических систем к непрерывному обучению, что является важным шагом к созданию адаптивных и долгоживущих интеллектуальных систем.

1.6 Анализ задачи планирования траекторий

Рассматривая задачи кинематики и планирования движений, в работе [92] проводится анализ современных подходов к управлению промышленными манипуляторами. Подчеркивается преимущество методов на основе глубокого обучения по сравнению с классическими эвристиками, особенно в условиях неизвестных препятствий. Также отмечается эффективность гибридных решений, объединяющих DRL и традиционные контроллеры для обеспечения устойчивости, и плавности траекторий. Развивая тему планирования движения, в исследовании [93] рассматриваются алгоритмы DRL для высокоскоростной навигации мобильных роботов и беспилотных систем. Показано, что методы семейства Actor-Critic и улучшенные версии Q-обучения позволяют эффективно работать в непрерывных пространствах, обеспечивая точное управление и адаптацию к динамическим условиям. В области восприятия среды в работе [94] анализируются методы семантического картирования, где особое внимание уделяется объединению геометрической и смысловой информации. Это позволяет роботам не только ориентироваться в пространстве, но и интерпретировать объекты и сцены, что важно для выполнения сложных задач.

Задачи локализации подробно рассматриваются в [95], где представлен обзор алгоритмов визуального SLAM. Отмечается значимость объединения данных с различных сенсоров и использования механизмов замыкания цикла для устранения накопленных ошибок, что критично для навигации в закрытых помещениях. В прикладных задачах планирования траекторий и обхода препятствий в [93, 96] показано, что методы глубокого обучения обеспечивают более надежное поведение по сравнению с традиционными алгоритмами. Дополнительно в [97, 98] рассматриваются иерархические подходы для управления сложными роботами, включая шагающие системы. Для задач реального времени в [99] предлагаются методы ускоренного вычисления управляющих воздействий, что снижает задержки и повышает устойчивость управления. В [100] демонстрируются гибридные подходы, объединяющие обратную кинематику и DRL для обеспечения безопасного движения манипуляторов. В условиях динамической среды в [101, 102] используются алгоритмы на основе Soft Actor Critic, позволяющие роботам адаптироваться к внешним возмущениям и точно следовать траекториям. В то же время в [103, 104] рассматриваются методы предотвращения столкновений и построения безопасных траекторий в реальном времени. Наконец, для оптимизации логистики в роботизированных системах в [105] применяются пространственно-временные методы DRL, позволяющие автономным

платформам учитывать движение людей и объектов, адаптируя маршруты без снижения эффективности производства.

1.7 Подходы по роботизированной сборке и манипуляции

В работе [106] рассматриваются современные подходы к выполнению контактных операций с требованиями к точности. Подчеркивается ограниченность классического позиционного управления и необходимость использования силового контроля в комбинации с вероятностными моделями. Например, частично наблюдаемые процессы принятия решений. Также отмечается перспективность методов восприятия, несмотря на их сложность. В сфере машинного зрения в [107] представлен анализ систем компьютерного зрения для промышленности. Рассматриваются методы сбора и аннотирования данных, а также интеграция сенсоров, включая камеры и LiDAR. Подчеркивается роль замкнутых систем управления, где визуальная информация напрямую влияет на корректировку действий робота. В задачах человеко-робот взаимодействия в [108] предлагается иерархическая модель передачи знаний от человека к роботу. Особое внимание уделяется формализации действий и использованию семантических представлений для повышения точности, и безопасности совместной работы. Развитие моделей Vision Language Action (VLA) рассматривается в [109, 110]. На рисунке 3 где показано, что трансформерные архитектуры позволяют напрямую преобразовывать визуально-языковую информацию в управляющие команды. Дополнительно в [111, 112] используются диффузионные политики для повышения обобщающей способности моделей.

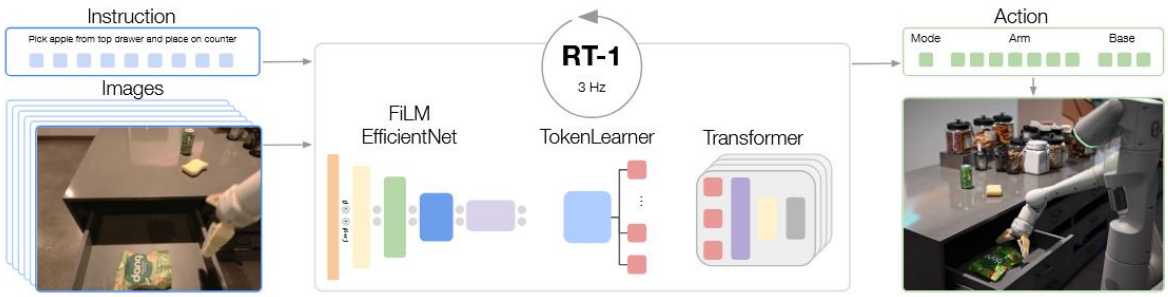

Рисунок 3 – Модель RT-1 [109]

В задачах манипуляции объектами в [113, 114] показана эффективность методов DRL для операций pick-and-place. Для работы с деформируемыми объектами в [115, 116] применяются адаптивные методы захвата, учитывающие физические свойства материалов. В классических задачах сборки в [117, 118] рассматриваются гибридные подходы, объединяющие

визуальные и тактильные данные. Для задач с непрерывным контактом в [119, 120] используются методы управления, позволяющие адаптировать жесткость системы в реальном времени. Вопросы безопасности рассматриваются в [121, 122], где предлагаются статистические методы фильтрации управляющих сигналов. На уровне интеграции систем в [123, 124] демонстрируются решения, объединяющие компьютерное зрение и обучение с подкреплением для точного управления. В [125, 126] рассматривается внедрение RL непосредственно на уровне ПЛК.

В специализированных задачах в [127, 128] разрабатываются методы управления в условиях ограниченной динамики, например, в космических системах. В [129] показаны подходы к снижению энергопотребления. Наконец, в [130, 131] рассматривается интеграция физических моделей в машинное обучение, что ускоряет обучение и повышает точность. В [132, 133] анализируются задачи высокоточной обработки поверхностей, а в [134, 135] рассматриваются методы планирования на основе графов знаний. Завершая обзор, в [136] демонстрируется возможность внедрения DRL непосредственно в промышленные контроллеры.

1.8 Анализ коллаборативной робототехники и безопасность

В области стандартизации и безопасности робототехнических систем в работе [137] рассматриваются подходы к валидации физической безопасности при взаимодействии человека и робота. Анализируются существующие стандарты и протоколы тестирования, направленные на снижение рисков, а также подчеркивается необходимость унификации процедур оценки безопасности для различных типов роботизированных систем. В контексте когнитивного взаимодействия человека и робота в [138] исследуется применение языковых моделей для повышения уровня автономности и качества планирования. Отмечаются как преимущества, так и ограничения, включая проблемы интерпретации, слияния сенсорных данных и обеспечения надежности решений. Проблема переноса моделей из симуляции в реальный мир рассматривается в [139], где анализируются методы уменьшения разрыва между виртуальной и физической средой. Подчеркивается важность рандомизации среды, робастных методов управления и консервативных алгоритмов оптимизации. Для обеспечения безопасности в динамических условиях в [140], [141] предлагаются методы предсказания поведения человека, позволяющие роботам адаптировать свои действия. В [142, 143] вводятся концепции безопасных зон, в которых агент может обучаться без риска выхода за допустимые пределы. Дополнительно в [144, 145] рассматриваются методы обучения без необходимости перезапуска, что повышает автономность систем. Наконец, в [146, 147] предлагаются подходы

к адаптации роботизированных процессов с учетом состояния человека, включая уровень нагрузки и усталости, что способствует повышению надежности и эффективности производственных систем. Дополнительно в работе [148] предлагается подход безопасного взаимодействия человека и робота на основе глубокого обучения с подкреплением. Авторы формализуют задачу предотвращения столкновений как задачу Марковского процесса принятия решений и демонстрируют возможность построения траекторий манипулятора в реальном времени с учетом положения оператора. В другом исследовании [149] рассматривается AR-assisted подход для взаимно-когнитивного безопасного взаимодействия человека и робота. Использование дополненной реальности совместно с DRL позволяет учитывать намерения человека и адаптировать траекторию робота в динамической среде. Дополнительно в [150] представлена концепция безопасного обучения с подкреплением, включающая безопасное исследование среды, выравнивание ценностей безопасности и безопасное сотрудничество человека и робота. Авторы подчеркивают важность интеграции механизмов ограничения действий и оценки риска при обучении роботизированных систем.

2 МЕТОДОЛОГИЯ

2.1 Концепция гибридного управления RL и MPC

В рамках данной работы предлагается гибридная архитектура управления роботизированным манипулятором, объединяющая Model Predictive Control (MPC) и Reinforcement Learning (RL). Основная идея подхода состоит в том, что MPC обеспечивает предсказуемое одновременно ограниченное и устойчивое управление. RL добавляет адаптивность к изменяющимся условиям среды или к неопределённостям модели и нештатным ситуациям.

Такой подход особенно актуален для промышленных приложений, где требуется одновременно: высокая точность движения, соблюдение физических и технологических ограничений, устойчивость к внешним возмущениям и способность к адаптации в изменяющейся среде.

Пусть состояние роботизированной системы в момент времени *t* задаётся вектором (1).

$$x_t \in \mathbb{R}^n \tag{1}$$

а управляющее воздействие имеет вид (2).

$$u_t \in \mathbb{R}^m \tag{2}$$

Тогда динамика системы в дискретном времени может быть записана (3).

$$x_{t+1} = f(x_t, u_t) \tag{3}$$

где функция описывает нелинейную динамику манипулятора и взаимодействие с рабочей средой. В гибридной архитектуре итоговое управление может формироваться следующим образом (4).

$$u_t = u_t^{\text{MPC}} + u_t^{\text{RL}} \tag{4}$$

u_t^{MPC}является управляющим воздействием, найденным методом предиктивного управления, u_t^{RL} является корректирующей переменной, формируемой агентом. В более общем виде RL может не только добавлять поправку, но и генерировать целевые ориентиры для MPC (5).

$$x_t^{\text{ref}} = \pi_\theta(x_t), x_t^{\text{ref}} = \pi_\theta(x_t), \quad u_t = \text{argmin} J_{\text{MPC}} \tag{5}$$

π_θ является стратегией RL с параметрами θ. MPC решает задачу построения за этой целевой траекторией.

Таким образом, гибридный подход может реализовываться в трёх основных вариантах (4). RL как генератор опорной траектории, MPC как исполнительный контроллер (5). MPC как защитный слой для RL предлагает действие u_t^{RL}, а MPC или блок ограничений проверяет его допустимость (6).

$$u_t = \Pi_{\mathcal{U}}(u_t^{\mathrm{RL}}) \tag{6}$$

$\Pi_{\mathcal{U}}$ обозначает проекцию на множество допустимых управлений $\mathcal{U}$. Для промышленной робототехники наиболее обоснованным является второй и третий вариант, поскольку они позволяют сохранить интерпретируемость и безопасность системы.

2.2 Разделение уровней управления

В рамках разрабатываемой системы управления роботизированным манипулятором предлагается использовать двухуровневую архитектуру, основанную на сочетании методов MPC и RL. Такая структура позволяет эффективно разделить задачи управления на два принципиально различных уровня: уровень точного исполнения и уровень адаптивного принятия решений.

Нижний уровень системы представлен контроллером на основе MPC, который отвечает за формирование управляющих воздействий. В частности, на подаваемых на исполнительные механизмы робота. Основной функцией данного уровня является обеспечение стабильного, предсказуемого и допустимого поведения системы. Это достигается за счёт решения на каждом шаге времени задачи оптимизации, учитывающей как динамику системы и ограничения на состояния и управления.

Пусть состояние системы в момент времени t задаётся вектором x_t, а управление обозначается как u_t. Тогда MPC формирует последовательность управляющих воздействий, минимизируя функционал качества вида (7).

$$J = \sum_{k=0}^{N-1} \left((x_{t+k|t} - x_{t+k}^{\mathrm{ref}})^T Q (x_{t+k|t} - x_{t+k}^{\mathrm{ref}}) + u_{t+k|t}^T R u_{t+k|t} \right) \tag{7}$$

при выполнении динамических ограничений (8) и ограничений на допустимые области состояний и действий (9).

$$x_{t+k+1|t} = f(x_{t+k|t}, u_{t+k|t}) \tag{8}$$

$$x_{t+k|t} \in \mathcal{X}, \quad u_{t+k|t} \in \mathcal{U} \tag{9}$$

Таким образом MPC гарантирует, что система не выйдет за пределы допустимых режимов, включая ограничения по углам суставов, скоростям, ускорениям и условиям безопасности (избегание столкновений, точное позиционирование). При этом управление реализуется по принципу скользящего горизонта, что означает что на каждом шаге применяется только первое оптимальное воздействие $u_t = u^{\star}_{t|t}$, после чего задача пересчитывается заново.

Верхний уровень системы представлен моделью (RL-агент), который выполняет принципиально другую функцию. В отличие от MPC, который опирается на заданную модель системы, RL-агент формирует свою стратегию управления на основе взаимодействия со средой и накопленного опыта. Его задача заключается не в непосредственном управлении приводами, а в адаптивной корректировке поведения системы в условиях неопределённости.

Формально процесс обучения с подкреплением задаётся как марковский процесс принятия решений (МППР), в котором агент реализует стратегию (10).

$$z_t = \pi_\theta(x_t) \tag{10}$$

π_θ является параметризованной политикой, а z_t представляет собой выход верхнего уровня управления. В отличие от классического подхода, в предлагаемой архитектуре данный выход не интерпретируется напрямую как управляющее воздействие. Вместо этого он используется для модификации параметров или целей нижнего уровня. Так, в зависимости от выбранной схемы интеграции, вектор z_t может интерпретироваться как: корректировка опорной траектории, смещение целевого положения схвата, желаемая скорость движения изменение весов функционала оптимизации MPC.

Например, если RL-агент формирует поправку к целевому состоянию, то опорная точка для MPC определяется как (11).

$$x_t^{\text{ref}} = x_t^{\text{nominal}} + z_t \tag{11}$$

В этом случае MPC решает задачу слежения уже за адаптивной траекторией, учитывающей текущие условия среды. Как альтернатива RL может воздействовать на структуру целевой функции MPC. Тогда матрицы весов принимают вид (12)

$$Q_t = Q_0 + \Delta Q_t^{\text{RL}}, \quad R_t = R_0 + \Delta R_t^{\text{RL}} \tag{12}$$

что позволяет динамически изменять приоритеты управления, например, усиливать точность позиционирования или, наоборот, снижать энергозатраты в зависимости от текущей ситуации. Такое взаимодействие формирует иерархическую архитектуру, которую можно представить в следующем виде (13).

$$x_t \rightarrow \pi_\theta(x_t) \rightarrow z_t \rightarrow \text{MPC}(x_t, z_t) \rightarrow u_t \rightarrow x_{t+1} \quad (13)$$

Принципиально важно, что RL в данной структуре не заменяет MPC, а дополняет его. MPC остаётся ответственным за безопасное и корректное управление системой. RL же обеспечивает адаптацию и улучшение стратегии в долгосрочной перспективе. С точки зрения функционального разделения это означает, что MPC решает задачу локального оптимального управления при заданных условиях, минимизируя отклонение от цели и соблюдая ограничения; RL решает задачу глобальной оптимизации поведения, формируя такие условия, при которых выполнение задачи становится более эффективным. Такое разделение особенно важно в промышленной робототехнике. Прямое управление манипулятором исключительно с использованием RL может приводить к нестабильным или небезопасным действиям, особенно на ранних этапах обучения. Введение MPC в качестве нижнего уровня позволяет ограничить пространство допустимых действий (14)

$$u_t \in \mathcal{U} \quad (14)$$

и тем самым гарантировать соблюдение физических и технологических требований. Дополнительно, двухуровневая структура позволяет повысить интерпретируемость системы. Действия MPC могут быть напрямую связаны с решением оптимизационной задачи, тогда как поведение RL можно анализировать через влияние на опорные параметры системы.

Таким образом, предложенная архитектура объединяет преимущества двух подходов. MPC обеспечивает строгое соблюдение ограничений и высокую точность управления, тогда как RL позволяет системе адаптироваться к изменяющимся условиям, неопределённостям модели и сложным сценариям взаимодействия со средой. В результате формируется гибкая и надёжная система управления, пригодная для применения в задачах промышленной робототехники, где критически важны как безопасность, так и эффективность выполнения операций.

2.3 Model Predictive Control

Современные робототехнические системы, особенно промышленные манипуляторы, функционируют в условиях высокой динамической сложности, наличия ограничений и необходимости точного позиционирования. В таких условиях традиционные методы управления, основанные на статических регуляторах, оказываются недостаточно гибкими. В этой связи особое значение приобретает метод управления с предсказанием модели, известный как MPC, который обеспечивает возможность оптимального управления с учётом будущего поведения системы. В основе MPC лежит математическая модель динамики объекта управления, описываемая в общем виде как (15)

$$x_{k+1} = f(x_k, u_k), \tag{15}$$

x_k представляет состояние системы в момент времени k, u_k управляющее воздействие. В контексте робототехнических систем вектор состояния может включать углы суставов, их скорости, а также положение и ориентацию рабочего органа. Управление соответствует прикладываемым моментам или силам.

Ключевой особенностью MPC является формирование оптимального управления на конечном горизонте предсказания длины N. Это приводит к следующей постановке задачи оптимизации (16) при ограничениях (17).

$$\min_{u_0,\ldots,u_{N-1}} \sum_{k=0}^{N-1} \ell\,(x_k, u_k) + \ell_f(x_N) \tag{16}$$

$$x_{k+1} = f(x_k, u_k), \quad x_k \in \mathcal{X}, \quad u_k \in \mathcal{U} \tag{17}$$

Функция стоимости $\ell(x_k, u_k)$ задаёт критерий качества управления, обычно включающий отклонение от желаемой траектории и штраф за использование управляющих воздействий. Терминальная функция $\ell_f(x_N)$ обеспечивает корректное завершение горизонта предсказания и влияет на устойчивость системы.

Принципиально важным является то, что оптимизация выполняется на каждом шаге времени, однако применяется только первое управление из найденной последовательности (18).

$$u_k = u_0^* \tag{18}$$

После этого горизонт предсказания сдвигается вперёд, и задача решается заново. Такой подход получил название скользящего горизонта и обеспечивает адаптацию управления к текущему состоянию системы.

Особую значимость MPC приобретает в задачах управления роботами благодаря возможности явного учёта ограничений. В реальных системах ограничения имеют критическое значение, поскольку превышение допустимых значений может привести к повреждению оборудования или созданию опасных ситуаций. Эти ограничения формализуются следующим образом (19).

$$u_{min} \leq \mathrm{u_k} \leq u_{max}, x_{min} \leq x_k \leq x_{max} \tag{19}$$

Таким образом, MPC не только оптимизирует поведение системы, но и гарантирует её безопасную эксплуатацию. Для линейных систем динамика может быть представлена в виде (20).

$$x_{k+1} = Ax_k + Bu_k \tag{20}$$

В этом случае задача MPC с квадратичной функцией стоимости сводится к задаче квадратичного программирования. Это позволяет использовать эффективные численные методы и реализовывать MPC в режиме реального времени, что особенно важно для робототехнических приложений (21).

$$J = \sum_{k=0}^{N-1} (x_k^T Q x_k + u_k^T R u_k) + x_N^T Q_f x_N \tag{21}$$

Тем не менее, несмотря на свои преимущества, MPC имеет ограничения, связанные прежде всего с необходимостью точной модели системы и высокой вычислительной сложностью. В реальных условиях параметры модели могут быть неизвестны или изменяться, что снижает эффективность управления. Именно в этом контексте становится актуальным использование методов обучения с подкреплением.

Обучение с подкреплением формулирует задачу управления как задачу максимизации накопленной награды:

$$\pi^* = \underset{\pi}{\mathrm{argmax}} \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t \, r(x_t, u_t)\right] \tag{22}$$

π – стратегия управления, $r(x,u)$ – функция награды. В отличие от MPC, методы RL не требуют явной модели динамики и способны адаптироваться к изменяющейся среде за счёт накопления опыта. Интересным является тот

факт, что между MPC и RL существует глубокая теоретическая связь. Если рассматривать функцию стоимости MPC как отрицательную награду (23)

$$J \approx -\sum r(x_k, u_k), \tag{23}$$

то становится очевидно, что обе парадигмы решают схожую задачу, но различными способами: MPC выполняет локальную оптимизацию на каждом шаге, тогда как RL стремится к глобальной оптимизации поведения.

Интеграция MPC и RL позволяет объединить преимущества обоих подходов. Один из способов заключается в использовании нейросетевой модели динамики (24)

$$x_{k+1} = \hat{f}_\theta(x_k, u_k), \tag{24}$$

которая обучается методом RL. Такая модель затем используется внутри MPC что позволяет компенсировать ошибки модели. Другой подход предполагает использование функции ценности, полученной в RL, в качестве терминальной функции MPC (25).

$$\ell_f(x_N) = V(x_N) \tag{25}$$

Это позволяет учитывать долгосрочные последствия действий, что значительно улучшает качество управления. Кроме того, MPC может выступать в роли самой политики (26).

$$\pi(x) = \underset{u_{0:N}}{\operatorname{argmin}} J(x, u) \tag{26}$$

RL может использоваться для настройки параметров MPC, таких как матрицы весов Q, R или длина горизонта предсказания N.

В задачах управления роботизированными манипуляторами такая интеграция особенно эффективна. Например, при задаче перемещения объекта в заданную точку целевая функция может быть записана как (27)

$$J = \sum_{k=0}^{N} \| x_k - x_{goal} \|^2 + \lambda \| u_k \|^2 \tag{27}$$

где первый аргумент отвечает за точность позиционирования, а второй - за плавность движения. MPC обеспечивает точное следование траектории и соблюдение ограничений, тогда как RL адаптирует стратегию к неопределённостям среды, таким как изменения массы объекта или внешние возмущения. Таким образом, MPC представляет собой фундаментальный метод оптимального управления, обладающий высокой точностью и

способностью учитывать ограничения системы. В то же время его сочетание с методами обучения с подкреплением открывает новые возможности для создания интеллектуальных адаптивных систем управления, способных эффективно функционировать в сложных и динамических условиях промышленной среды.

2.4 Методы обучения с подкреплением: DDPG и TD3

2.4.1 Deep deterministic policy gradient

Развитие робототехнических систем требует методов управления, способных адаптироваться к неопределённостям среды и обучаться на основе взаимодействия с ней. В отличие от MPC, который опирается на явную модель динамики, методы обучения с подкреплением формируют стратегию управления непосредственно через опыт, что делает их особенно перспективными для задач с высокой степенью неопределённости. В общем виде задача обучения с подкреплением формулируется как МППР, в рамках которого агент взаимодействует со средой, получая состояния x_t, выбирая действия u_t и получая награду $r(x_t, u_t)$. Целью является нахождение оптимальной стратегии (28).

$$\pi^* = \underset{\pi}{\text{argmax}}\mathbb{E}\left[\sum_{t=0}^{\infty}\gamma^t\, r(x_t, u_t)\right] \tag{28}$$

$\gamma \in (0,1)$ – коэффициент дисконтирования, определяющий важность будущих наград.

Ключевым понятием является функция ценности действия (Q-функция), которая оценивает качество выбора действия u в состоянии x (29).

$$Q^{\pi}(x, u) = \mathbb{E}\left[\sum_{t=0}^{\infty}\gamma^t\, r(x_t, u_t) \mid x_0 = x, u_0 = u\right] \tag{29}$$

В задачах управления роботами пространство действий является непрерывным (30).

$$u \in \mathbb{R}^m \tag{30}$$

Это исключает использование классических методов RL, таких как Q-learning, поскольку поиск максимума (31) становится вычислительно сложным. Для решения этой проблемы используются методы как actor-critic, в которых политика параметризуется явно (31)

$$\max_{u} Q(x,u) \tag{31}$$

Алгоритм DDPG представляет собой детерминированный вариант actor-critic подхода, в котором политика задаётся как (32).

$$u = \mu_\theta(x), \tag{32}$$

μ_θ - нейросеть (actor), параметризованная весами θ. Критик обучается аппроксимировать Q-функцию (33)

$$Q_\phi(x,u). \tag{33}$$

Обучение критика осуществляется путём минимизации функции ошибки Беллмана (34)

$$L(\phi) = \mathbb{E}\left[\left(Q_\phi(x_t,u_t) - y_t\right)^2\right], \tag{34}$$

где целевое значение определяется как (35)

$$y_t = r_t + \gamma Q_{\phi'}(x_{t+1}, \mu_{\theta'}(x_{t+1})). \tag{35}$$

Здесь ϕ' и θ' - параметры целевых сетей, обновляемых по правилу Поляка (36).

$$\theta' \leftarrow \tau\theta + (1-\tau)\theta' \tag{36}$$

Политика обучается через градиент по Q-функции (37).

$$\nabla_\theta J \approx \mathbb{E}\left[\nabla_u Q_\phi(x,u) \mid_{u=\mu_\theta(x)} \nabla_\theta \mu_\theta(x)\right] \tag{37}$$

Таким образом, actor обновляется так, чтобы выбирать действия, максимизирующие оценку критика. Несмотря на эффективность, DDPG обладает рядом недостатков, включая переоценку Q-функции и нестабильность обучения.

2.4.2 Алгоритм Twin Delayed Deep Deterministic Policy Gradient (TD3)

Алгоритм TD3 был предложен как улучшение DDPG с целью устранения его основных недостатков. Ключевые модификации TD3 направлены на уменьшение переоценки Q-функции и стабилизацию обучения. Вместо одной Q-функции используются две независимые оценки (38).

$$Q_{\phi_1}(x,u), \quad Q_{\phi_2}(x,u) \tag{38}$$

Целевое значение вычисляется как минимум двух оценок (39).

$$y_t = r_t + \gamma \min_{i=1,2} Q_{\phi_{i'}}(x_{t+1}, \tilde{u}) \tag{39}$$

Это снижает эффект переоценки и делает обучение более устойчивым. В TD3 вводится шум в целевое действие (40) с ограничением (41).

$$\tilde{u} = \mu_{\theta'}(x_{t+1}) + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma) \tag{40}$$

$$\epsilon \in [-c, c] \tag{41}$$

Это предотвращает переобучение критика к узким областям пространства действий. Actor обновляется реже, чем критик (42).

$$\theta \leftarrow \theta - \alpha \nabla_\theta J \quad \text{каждые } d \text{ шагов} \tag{42}$$

Это позволяет критикам сначала стабилизироваться, что улучшает качество градиента. В задачах управления манипуляторами состояние может включать (43).

$$x = [q, \dot{q}, x_{TCP}, x_{target}] \tag{43}$$

q – углы суставов, x_{TCP} – положение рабочего органа. Действие задаётся как (44).

$$u = \tau \in \mathbb{R}^m \tag{44}$$

Таким образом, агент обучается минимизировать ошибку позиционирования и затраты управления. TD3 особенно эффективен в таких задачах благодаря: устойчивости обучения способности работать в непрерывных пространствах и уменьшению переоценки Q-функции.

Алгоритмы DDPG и TD3 представляют собой эффективные инструменты для управления роботами в непрерывных пространствах действий. Они позволяют обучать сложные стратегии управления без явной модели динамики, что делает их особенно полезными в условиях неопределённости. Тем не менее, наилучшие результаты достигаются при интеграции с методами MPC, где алгоритм обеспечивает точность и соблюдение ограничений, а RL – адаптивность и способность к обучению. Такой синергетический подход формирует основу современных интеллектуальных систем управления роботами.

2.4.3 Иерархическое обучение с подкреплением

Иерархическое обучение с подкреплением представляет собой расширение классического RL, направленное на декомпозицию сложной задачи управления на несколько уровней абстракции. В контексте управления роботизированными манипуляторами это особенно важно, поскольку задачи часто имеют многосоставную структуру: захват объекта, перенос, позиционирование и взаимодействие с окружающей средой. Пусть стандартная задача RL задаётся МППР (45).

$$\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, R, \gamma) \quad (45)$$

- $\mathcal{S}$ – пространство состояний;
- $\mathcal{A}$ – пространство действий;
- $P(s' \mid s, a)$ – вероятностная динамика;
- $R(s, a)$ – функция награды;
- $\gamma \in [0,1]$ – коэффициент дисконтирования.

В HRL вводится дополнительная структура в виде иерархии политик (46).

$$\pi = \{\pi^{high}, \pi^{low}\} \quad (46)$$

Высокоуровневая политика π^{high} оперирует абстрактными действиями или подзадачами (47)

$$g_t \sim \pi^{high}(g \mid s_t), \quad (47)$$

g_t – подцель. Например, как: поднести захват к объекту, переместить к корзине). Низкоуровневая политика π^{low} реализует управление (48).

$$a_t \sim \pi^{low}(a \mid s_t, g_t) \quad (48)$$

Таким образом, действие зависит не только от состояния, но и от текущей подцели. Целевая функция для иерархии может быть записана как (49).

$$J(\pi) = \mathbb{E}_{\pi^{high}, \pi^{low}} \left[\sum_{t=0}^{T} \gamma^t R(s_t, a_t) \right] \quad (49)$$

Однако обучение проводится раздельно: π^{low} оптимизируется для достижения подцелей (50).

$$r_t^{low} = -\| s_t - g_t \| \quad (50)$$

π^{high} оптимизируется по итоговой задаче (51).

$$r_t^{high} = R(s_t, a_t) \quad (51)$$

В задачах робототехники HRL позволяет естественно моделировать поведение манипулятора как последовательность фаз: поиск объекта, захват, перенос и размещение. Идея универсальной политики в обучении с подкреплением предполагает существование одной стратегии, способной решать широкий спектр задач без повторного обучения. Однако на практике это сталкивается с фундаментальными барьерами. Главная трудность – различие распределений состояний и функций вознаграждения. Оптимальная политика для одной среды редко переносится в другую. Универсальная стратегия должна охватывать сразу все варианты, что ведёт к чрезмерно

большому пространству действий и неопределённости в целях оптимизации. К этому добавляется проблема катастрофического забывания: при последовательном обучении новая задача разрушает старые навыки. А стремление охватить множество сред требует колоссального исследования, делая обучение универсальной политики крайне долгим и ресурсоёмким. Поэтому современные подходы всё чаще используют не единую стратегию, а модульные иерархии, ансамбли политик и мета-обучение, где универсальность достигается через адаптацию и разделение задач. Таким образом, сама идея одной политики оказывается скорее теоретическим идеалом, чем практическим инструментом.

2.4.4 Многоуровневый TD3 (Multi-level TD3)

Алгоритм Twin Delayed Deep Deterministic Policy Gradient (TD3) является одним из наиболее стабильных алгоритмов для непрерывного управления. Однако в сложных задачах, таких как манипуляции с объектами, классический TD3 может испытывать трудности из-за разреженных наград и сложной динамики среды. Multi-Level TD3 представляет собой расширение TD3, сочетающее идеи HRL и фазового разбиения задачи. В классическом TD3 используются: две функции ценности $Q_{\theta_1}, Q_{\theta_2}$, актор π_ϕ, целевые сети. Целевая функция критика задается следующим образом (52).

$$y = r + \gamma \min_{i=1,2} Q_{\theta_{i'}}(s', \pi_{\phi'}(s') + \epsilon), \tag{52}$$

$\epsilon \sim \text{clip}(\mathcal{N}(0, \sigma), -c, c)$. В таком случае функция потерь будет иметь следующий вид (53)

$$\mathcal{L}(\theta_i) = \mathbb{E}\left[(Q_{\theta_i}(s, a) - y)^2\right]. \tag{53}$$

Актор обновляется реже (54)

$$\nabla_\phi J \approx \mathbb{E}\left[\nabla_a Q_{\theta_1}(s, a) \nabla_\phi \pi_\phi(s)\right]. \tag{54}$$

Иерархическое обучение с подкреплением и Multi-Level TD3 позволяют перейти от монолитного обучения к структурированному управлению, что критически важно для сложных робототехнических задач. Разделение на уровни и фазы уменьшает сложность оптимизации, повышает стабильность обучения и делает систему более интерпретируемой и масштабируемой, особенно при интеграции с методами управления. Дополнительно такой подход позволяет локализовать принятие решений на каждом уровне. Это особенно эффективно в задачах манипуляции, где требуется сочетание глобального выбора цели и локального управления суставами робота. В результате повышается адаптивность системы к изменениям внешней среды и улучшается качество выполнения сложных последовательных операций.

Алгоритм 1: Multi-Level TD3

1 **Инициализация**
- параметр актёра ϕ, критиков θ_1, θ_2
- таргет-копии $\phi', \theta_{1'}, \theta_{2'}$
- коэффициенты обучения α, β
- коэффициент дисконтирования γ
- Polyak-коэффициент τ
- частота обновления актёра d
- шумы: action noise $\epsilon \sim \mathcal{N}(0, \sigma)$, target policy smoothing $\epsilon' \sim clip(\mathcal{N}(0, \tilde{\sigma}), -c, c)$
- replay-буфер $\mathcal{D}$

2 **Для каждого управляющего цикла $t = 0, 1, 2, \ldots$ до T**
- Выбрать действие $a_t = \pi_\phi(s_t) + \epsilon$
- Выполнить действие a_t, получить r_t, s_{t+1}, сохранить $(s_t, a_t, r_t, s_{t+1}) \in \mathcal{D}$
- Выбрать minibatch из $\mathcal{D}$
- Сгенерировать зашумлённое таргет-действие $\tilde{a}_{t+1} = \pi_{\phi'}(s_{t+1}) + \epsilon'$
- Посчитать таргет: $y = r_t + \gamma \cdot \min_{i=1,2} Q_{\theta_{i'}}(s_{t+1}, \tilde{a}_{t+1})$
- Обновить критики: $\theta_i \leftarrow \theta_i - \beta \nabla_{\theta_i}(Q_{\theta_i}(s_t, a_t) - y)^2, \quad i = 1,2$

3 **Если t mod d == 0**
- Обновить актёра: $\phi \leftarrow \phi + \alpha \nabla_\phi Q_{\theta_1}(s_t, \pi_\phi(s_t))$
- Обновить таргет-сети: $\theta_{i'} \leftarrow \tau\theta_i + (1-\tau)\theta_{i'}, \quad \phi' \leftarrow \tau\phi + (1-\tau)\phi'$

2.5 Постановка задачи для планирования траектории

2.5.1 MPC для построения траектории в 3D-пространстве

Ставится задача построения траектории движения объекта в непрерывном трехмерном пространстве с использованием управления (55).

$$\begin{aligned} \mathrm{P}_1 &= (x_1, y_1, z_1) \\ \mathrm{P}_2 &= (x_2, y_2, z_2) \\ \mathrm{P}_3 &= (x_3, y_3, z_3), \end{aligned} \tag{55}$$

P_1 – начальная точка, P_2 – промежуточная целевая точка, P_3 – конечная целевая точка.

Система управления должна построить траекторию, начиная из P_1, направляясь последовательно к P_2 и P_3, основываясь на модели движения и принципе MPC. Если необходимо пройти не три, а произвольное число целевых точек $\mathrm{P}_1, \mathrm{P}_2, \ldots, \mathrm{P}_N$, модель MPC легко масштабируется. В этом случае

целевая точка на каждом шаге времени t выбирается из списка в соответствии с текущим прогрессом вдоль траектории и функционал стоимости дополняется суммой отклонений от каждой промежуточной цели (56)

$$J = \sum_{t=1}^{N} \| \mathrm{x}_t - \mathrm{p}_t^{\text{target}} \|^2 + \lambda \sum_{t=0}^{N-1} \| \mathrm{a}_t \|^2, \tag{56}$$

Последовательность $\mathrm{p}_t^{\text{target}}$ строится по заранее заданным точкам пути $[\mathrm{P}_1, \mathrm{P}_2, \ldots, \mathrm{P}_N]$, возможно с интерполяцией или разбиением по времени. Разберем целевую функцию в дальнейших разделах.

2.5.2 Кинематическая модель движения

Предполагается, что объект подчиняется кинематике второго порядка. Состояние системы включает положение x_t и скорость v_t, а управляющее воздействие - ускорение a_t (57).

$$\mathrm{x}_t = \begin{bmatrix} x_t \\ y_t \\ z_t \end{bmatrix}, \quad \mathrm{v}_t = \begin{bmatrix} v_{x,t} \\ v_{y,t} \\ v_{z,t} \end{bmatrix}, \quad \mathrm{a}_t = \begin{bmatrix} a_{x,t} \\ a_{y,t} \\ a_{z,t} \end{bmatrix} \tag{57}$$

Кинематика второго порядка предполагает, что положение объекта зависит от второй производной по времени, то есть от ускорения. Таким образом, изменение положения объекта определяется текущей скоростью и ускорением, а скорость – под действием управляющего ускорения. Такая модель отражает инерционные свойства движения и обеспечивает реалистичную динамику. Динамика системы в дискретной форме при шаге Δt (58).

$$\begin{gathered} \mathrm{x}_{t+1} = \mathrm{x}_t + \Delta t \cdot \mathrm{v}_t + \frac{1}{2} \Delta t^2 \cdot \mathrm{a}_t \\ \mathrm{v}_{t+1} = \mathrm{v}_t + \Delta t \cdot \mathrm{a}_t \end{gathered} \tag{58}$$

Использование дискретной модели обусловлено цифровой природой вычислительных систем и необходимостью численного решения задачи оптимизации. Модель дискретизируется с постоянным шагом времени Δt, что обеспечивает совместимость с методами численного программирования, такими как квадратичное программирование (QP), применяемое в MPC.

2.5.3 Граничные условия и целевая функция

Начальное состояние известно т.е. начальная точка – первые начальные координаты, при скорости равной нулю (59).

$$\mathrm{x}_0 = \mathrm{P}_1, \quad \mathrm{v}_0 = 0 \tag{59}$$

Управление направлено на минимизацию отклонения от целевых точек, а также на минимизацию управляющего воздействия. Моделирование функции состоит из необходимости минимизировать расстояние между предсказанной позицией x и целевой точкой p_t^{target} на каждом шаге. Это будет означать что объект будет стараться следовать по нужной траектории используя квадрат евклидовой нормы разности между двумя векторами (60).

$$\sum_{t=1}^{N} \| \mathrm{x}_t - \mathrm{p}_t^{\text{target}} \|^2 \tag{60}$$

$\mathrm{p}_t^{\text{target}}$ – целевая точка на шаге времени t, выбираемая из множества заданных точек, Для плавности добавим параметр, который будет штрафовать за слишком быстрое ускорение (61).

$$\lambda \sum_{t=0}^{N-1} \| \mathrm{a}_t \|^2 \tag{61}$$

$\lambda > 0$ – вес, коэффициент штрафа за резкие ускорения. Итоговая целевая функция будет выглядеть как квадратичное уравнение (QP) (62).

$$J = \sum_{t=1}^{N} \| \mathrm{x}_t - \mathrm{p}_t^{\text{target}} \|^2 + \lambda \sum_{t=0}^{N-1} \| \mathrm{a}_t \|^2 \tag{62}$$

Формулировка задачи оптимального управления выглядит следующим образом. Цель минимизировать J при данных вышеописанных условиях (63).

$$\begin{aligned} &\min_{\mathrm{x}_{0:N}, \mathrm{v}_{0:N}, \mathrm{a}_{0:N-1}} \quad \sum_{t=1}^{N} \| \mathrm{x}_t - \mathrm{p}_t^{\text{target}} \|^2 + \lambda \sum_{t=0}^{N-1} \| \mathrm{a}_t \|^2 \\ &\text{при условиях:} \qquad \mathrm{x}_0 = \mathrm{P}_1, \quad \mathrm{v}_0 = 0 \\ &\mathrm{x}_{t+1} = \mathrm{x}_t + \Delta t \cdot \mathrm{v}_t + \frac{1}{2} \Delta t^2 \cdot \mathrm{a}_t \\ &\mathrm{v}_{t+1} = \mathrm{v}_t + \Delta t \cdot \mathrm{a}_t, \quad \forall t = 0, \dots, N-1 \end{aligned} \tag{63}$$

2.5.4 Сравнительные модели для траектории

Гладкие функции – функции, которые точно проходит через заданные точки (interpolation), минимизирует суммарную кривизну или изменение ускорения, но не учитывает динамические ограничения, например, максимальные ускорения, скорость объекта, инерцию или модель движения. То есть, они оптимальны с точки зрения геометрии, но не обязательно с точки зрения управления. MPC, напротив, оптимизирует движение с учётом динамики, позволяет учитывать ограничения на ускорение, скорость и

положение и может отклоняться от точек, но делает это в рамках физических ограничений.

Кубический сплайн. Каждая компонента $x(t), y(t), z(t)$ аппроксимируется отдельно (64).

$$S_k(t) = a_k + b_k(t - t_k) + c_k(t - t_k)^2 + d_k(t - t_k)^3, \quad t \in [t_k, t_{k+1}] \qquad (64)$$

k – номер сегмента между точками P_k и P_{k+1}, t – параметр времени или длины дуги, определяющий положение вдоль сегмента, t_k – значение параметра t, соответствующее точке P_k, a_k, b_k, c_k, d_k – коэффициенты полинома, вычисляемые так, чтобы обеспечить непрерывность траектории и её производных.

Линейная интерполяция. Простая интерполяция прямыми отрезками между точками (65).

$$L_k(t) = \frac{t_{k+1} - t}{t_{k+1} - t_k} \cdot \mathrm{P}_k + \frac{t - t_k}{t_{k+1} - t_k} \cdot \mathrm{P}_{k+1}, \quad t \in [t_k, t_{k+1}] \qquad (65)$$

t – параметр на интервале между узлами, t_k, t_{k+1} – значения параметра, соответствующие точкам P_k и P_{k+1}.

Обеспечивает непрерывность только самой траектории, но не её производных.

B-сплайн. Обобщённая форма аппроксимации, не обязательно проходящая через точки. Траектория выражается через базисные функции (66).

$$B(t) = \sum_{i=0}^{n} N_{i,p}\,(t) \cdot \mathrm{P}_i \qquad (66)$$

$N_{i,p}(t)$ – B-сплайн базисные функции степени p, определяемые рекурсивно; P_i - контрольные точки (не обязательно лежат на траектории); t – параметр вдоль кривой. Для кубических B-сплайнов: $p = 3$.

Параметризация и выбор узлов влияют на форму кривой: B-сплайн обычно гладкий, но не проходит точно через точки.

2.5.5 Поиск оптимальных параметров MPC

Параметрами в MPC считается две переменные dt, N. Где dt – дискретный шаг времени между предсказаниями, N – количество шагов предсказания в горизонте планирования. Выбор может быть зависим от множества факторов. Так как MPC повторяет оптимальное поведение с учетом аспектов системы, MPC свойственно опираться на физические свойства реальных объектов.

Выбор частоты обновления модели зависит от физики системы (например, скорости объекта, инерции), должен быть достаточно малым, чтобы точно аппроксимировать динамику. Но не слишком маленьким, чтобы не перегружать вычислитель. Обычно, значение берется с 0.01-0.1.

Горизонт планирования – сколько шагов мы предсказываем вперёд. Он основывается на еще параметре планирование по времени T (67).

$$T = N \cdot dt \tag{67}$$

Параметр T должен сохраняться в диапазоне нескольких секунд. Это время – отвечает и за плавность модели. Чем больше N – тем больше времени требуется для расчета и как результат плавней. Для больших маршрутов рекомендуется использовать большее значение N, и наоборот для быстрого решения – небольшое.

Для того чтобы найти оптимальные параметры, можно использовать определенные критерии, после которого мы получим значения, которые затем можно будет сравнить как оптимальные и нет. Для первого можно использовать поиск по сетке – итерация с каждым наблюдаемым значением по каждому с каждым. Для выборки были выбраны: параметр N находится от 40 до 100 с шагом 20, когда dt равняется от 0.05,0,1 и 0.2.

2.5.6 Парето-фронт

Для того чтобы найти оптимальные параметры, можно использовать определенные критерии, после которого мы получим значения, которые затем можно будет сравнить как оптимальные и нет. Для первого можно использовать поиск по сетке – итерация с каждым наблюдаемым значением по каждому с каждым. Также для повышения эффективности поиска оптимальных решений в многокритериальной постановке может быть применён эволюционный алгоритм NSGA-II. В отличие от полного перебора параметров (grid search), данный подход позволяет работать с большим пространством параметров и находить приближённый парето-фронт за разумное время.

Алгоритм NSGA-II основан на популяционном подходе, где на каждом шаге формируется множество решений, которые эволюционируют посредством операций отбора, скрещивания и мутации. Ключевой особенностью является использование процедуры недоминирующей сортировки, позволяющей разбить все решения на уровни (фронты) в зависимости от их доминирования. Дополнительно применяется механизм сохранения разнообразия решений, что предотвращает их сходимость в одну точку и обеспечивает равномерное покрытие Парето-фронта.

Таким образом, NSGA-II позволяет одновременно учитывать несколько критериев оптимальности, таких как точность (MAE) и плавность траектории, и получать набор компромиссных решений. Это особенно важно в задачах управления роботизированными манипуляторами, где требуется баланс между качеством выполнения задачи и физическими ограничениями системы. Для выборки были выбраны: параметр N находится от 40 до 100 с шагом 20, когда *dt* равняется от 0.05,0,1 и 0.2.

Для нахождения оптимальных параметров было использовано так называемая парето-фронт. Пусть у нас есть множество точек (решений) $\mathcal{S} = \{\mathrm{s}_1, \mathrm{s}_2, \dots, \mathrm{s}_n\}$, где каждая точка $\mathrm{s}_i = \left(f_1^{(i)}, f_2^{(i)}, \dots, f_k^{(i)}\right)$ - значения k критериев. Решение s_a доминирует над s_b (68).

$$\forall j \in \{1, \dots, k\}: f_j^{(a)} \leq f_j^{(b)} \quad \text{и} \quad \exists j: f_j^{(a)} < f_j^{(b)} \tag{68}$$

То есть: s_a не хуже по всем критериям, и строго лучше хотя бы по одному. Тогда Парето-фронт – это подмножество $\mathcal{P} \subset \mathcal{S}$, состоящее из всех точек, которые не доминируются никакими другими (69). Множество P состоит из всех элементов s_i из множества S, для которых существует другой элемент s_j из множества S, отличный от s_i, такой что $s_j \prec s_i$.

$$\mathcal{P} = \{\, s_i \in S \mid \nexists s_j \in S \setminus \{s_i\} : s_j \prec \mathrm{s_i}\} \tag{69}$$

В нашем случае (2 критерия): пусть: $f_1 = \mathrm{MAE}$, $f_2 =$ Средняя норма ускорения. Тогда точка $\left(f_1^{(a)}, f_2^{(a)}\right)$ доминирует $\left(f_1^{(b)}, f_2^{(b)}\right)$, если $f_1^{(a)} \leq f_1^{(b)}$ и $f_2^{(a)} \leq f_2^{(b)}$ и при этом хотя бы одно строгое неравенство.

2.5.7 Метрики оценки

Для анализа эффективности и реалистичности траектории, построенной с помощью MPC, проводится сравнение с классической сплайн-интерполяцией. Кубический сплайн строится по всем точкам $\mathrm{P}_1, \dots, \mathrm{P}_N$ и представляет собой гладкую функцию, минимизирующую кривизну траектории. Однако кубический сплайн не учитывает физику движения (скорость, ускорение, ограничения). Несмотря на то, что, всегда проходит через точки, но может требовать нереалистичных манёвров, оптимален по геометрии, но не по управлению. Для количественного анализа введены следующие метрики сравнения. Пусть:

$\mathbf{x}_t^{\mathrm{mpc}}$ – положение по MPC на шаге t,

$\mathbf{x}_t^{\mathrm{spline}}$ – положение по сплайну на шаге t,

T – общее количество шагов времени.
Ошибка на каждом шаге (евклидова) определяет отклонение текущей точки траектории до соответствующей точки сплайновой траектории. Чем меньше значение, тем ближе траектории друг к другу (70).

$$e_t = \| \mathrm{x}_t^{\mathrm{mpc}} - \mathrm{x}_t^{\mathrm{spline}} \|_2 \tag{70}$$

Средняя ошибка (MAE) показывает среднее отклонение между траекторией MPC и сплайновой траекторией за весь временной интервал. Используется как общий показатель точности аппроксимации (71).

$$\mathrm{MAE} = \frac{1}{T} \sum_{t=0}^{T-1} e_t \tag{71}$$

Максимальная ошибка является значением наибольшего отклонения между траекториями MPC и сплайна среди всех временных шагов. Позволяет оценить худший случай расхождения (72).

$$\mathrm{MaxError} = \max_t (e_t) \tag{72}$$

Норма скорости определяет величину скорости перемещения объекта между двумя соседними точками траектории (73).

$$\| \mathrm{v}_t \|_2 = \frac{1}{\Delta t} \| \mathrm{x}_{t+1} - \mathrm{x}_t \|_2 \tag{73}$$

Норма ускорения (74) показывает изменение скорости между соседними временными шагами, характеризуя плавность движения траектории (74).

$$\| \mathrm{a}_t \|_2 = \frac{1}{\Delta t} \| \mathrm{v}_{t+1} - \mathrm{v}_t \|_2 \tag{74}$$

Эти метрики позволяют количественно оценить, насколько MPC траектория близка к сплайну, и насколько она более «физически корректна» по скорости и ускорению.

Графики ошибок, скорости и ускорения визуализируются для полного анализа различий между подходами. Итоговый pipeline выглядит следующим образом на рисунке 4.

Рисунок 4 – Pipeline связи UR5 и MPC

Алгоритм 2: Model Predictive Control для построения траектории

1 **Инициализация**

- Задать начальную позицию x_0 и скорость $v_0 = 0$
- Определить горизонт предсказания N, шаг по времени Δt и параметр регуляризации λ
- Задать целевые точки (waypoints): $P = \{p_1, \dots, p_M\}$

2 **Для каждого управляющего цикла $t = 0, 1, 2, \dots$:**

2.1 Построение опорной (референсной) траектории

- Определить текущую подцель $x_t^{ref} \in P$ на основе близости к точкам и прогресса движения

2.2 **Формулировка задачи оптимизации**

- Минимизировать функцию стоимости:

$$J = \sum_{k=1}^{N} \| x_k - x_k^{ref} \|_2^2 + \lambda \sum_{k=0}^{N-1} \| a_k \|_2^2$$

- При ограничениях динамики системы:

$$x_{k+1} = x_k + v_k \Delta t + \frac{1}{2} a_k \Delta t^2$$
$$v_{k+1} = v_k + a_k \Delta t$$

2.3 **Решение задачи оптимизации**

- Использовать метод квадратичного программирования (QP) для нахождения оптимальной последовательности управлений:

$$\{a_t^*, \dots, a_{t+N-1}^*\}$$

2.4 **Применение управляющего воздействия**

- Применить только первое управление:

$$a_t = a_t^*$$

2.5 **Обновление состояния системы**

- Вычислить:

$$v_{t+1} = v_t + a_t \Delta t$$
$$x_{t+1} = x_t + v_t \Delta t + \frac{1}{2} a_t \Delta t^2$$

2.6 **Повторить процесс с шага 2.1**

Увеличить t пока траектория не будет завершена или не будет достигнута последняя целевая точка.

2.6 Определение задачи для обучения с подкреплением

2.6.1 Постановка задачи

В рамках задачи рассматривается движение агента в непрерывном трехмерном пространстве, где заданы две ключевые точки: начальная и целевая. Основная цель состоит в построении траектории, позволяющей агенту добраться от начальной позиции до целевой по наиболее короткому пути.

Для решения данной задачи используется подход обучения с подкреплением, в рамках которого формируется функция награды. Эта функция определяет поведение агента, поощряя его за действия такие как приближающие к цели и штрафуя за нежелательные состояния. Таким образом, задача формулируется как задача максимизации суммарной награды.

Для усложнения сценария в пространство могут быть добавлены препятствия, ограничивающие движение агента. В результате можно выделить два типа задач:

- Движение к одной целевой точке;
- Последовательное достижение нескольких целевых точек.

Во втором случае для наглядности оптимальной траектории между целями может применяться аппроксимация с использованием кубических сплайнов.

2.6.2 Определение функции награды

В предложенном подходе функция награды включает три основных компонента. Первый компонент связан с расстоянием до цели. Агент получает штраф, пропорциональный расстоянию до текущей цели, что стимулирует его сокращать это расстояние. Чем ближе агент к цели, тем меньше штраф и, соответственно, выше итоговая награда (75).

$$R_{distance} = -\|current\_pos - goal\| \quad (75)$$

$current\_pos$ – текущая позиция агента, а $goal$ – координаты текущей цели.
Второй компонент отвечает за взаимодействие с препятствиями. Вводятся параметры расстояния до препятствия и штрафа за опасное приближение. Если агент оказывается ближе заданного порога (например, 0.2 единицы), ему начисляется штраф фиксированной величины (например, 10). Это формирует поведение избегания столкновений (76).

$$R_{obstacle} = \begin{cases} -10, & if\ \|current\_pos - obstacle\| < 0.2 \\ 0, & else \end{cases} \quad (76)$$

Третий компонент поощряет достижение цели. Если агент оказывается вблизи цели на расстоянии менее заданного порога, он получает положительную награду. Это позволяет явно закрепить успешное выполнение подзадачи (77).

$$R_{goal\_reward} = \begin{cases} +10, & if\ \|current - goal\| < 0.1 \\ 0, & else \end{cases} \quad (77)$$

Итоговая функция награды представляет собой комбинацию всех описанных компонентов (78).

$$R_{total} = R_{distance} + \sum_{obstacle} R_{obstacle} + R_{goal\_reward} \quad (78)$$

2.7 Применение в симуляционной среде

Webots является универсальной симуляционной средой, разработанной компанией Cyberbotics, и ориентированной как на научные исследования, так и на прикладные задачи. Основное назначение данной платформы заключается в предоставлении пользователю инструмента для создания виртуальных моделей роботов, окружающей среды и сценариев взаимодействия с высокой степенью реалистичности. Благодаря интеграции физического движка Webots позволяет учитывать такие аспекты, как динамика, кинематика, столкновения и силы трения, что делает моделирование максимально приближенным к реальным условиям эксплуатации. Одним из ключевых преимуществ Webots является его ориентация на возможность переноса разработанных и обученных моделей из симуляции в реальный мир как показано на рисунке 5. Это особенно важно в задачах, связанных с управлением роботизированными манипуляторами, где ошибки в реальной системе могут привести к значительным затратам или даже повреждению оборудования. Использование Webots позволяет предварительно протестировать алгоритмы управления, включая методы

управления на основе Model Predictive Control и алгоритмы обучения с подкреплением, такие как DDPG или TD3, в безопасной виртуальной среде. Таким образом, достигается снижение рисков и ускорение процесса разработки.

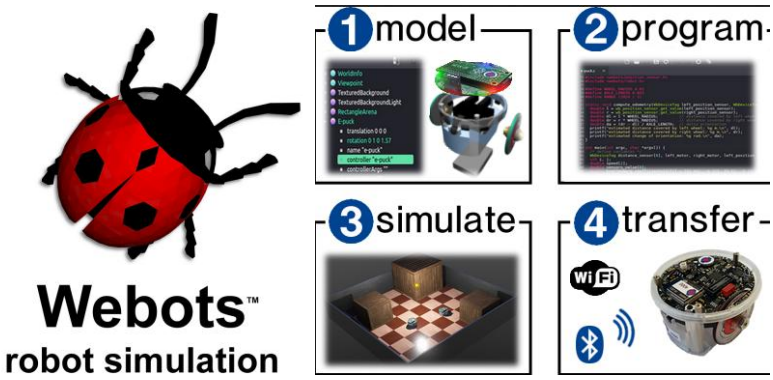

Рисунок 5 – Логотип Webots

Важным аспектом Webots является его поддержка широкого спектра языков программирования, включая Python, C и C++. Особенно значимой является интеграция с Python, поскольку именно этот язык широко используется в области машинного обучения и искусственного интеллекта. Благодаря этому открывается возможность интегрировать алгоритмы обучения с подкреплением непосредственно в симуляцию. Это делает Webots удобной платформой для разработки интеллектуальных робототехнических систем, где требуется тесная связь между восприятием, принятием решений и управлением. В библиотеке платформы представлены различные типы манипуляторов, мобильных роботов, камер, лидаров и других устройств. Это позволяет сократить время разработки, поскольку не требуется создавать все компоненты с нуля. Интерфейс представлен на рисунке 6.

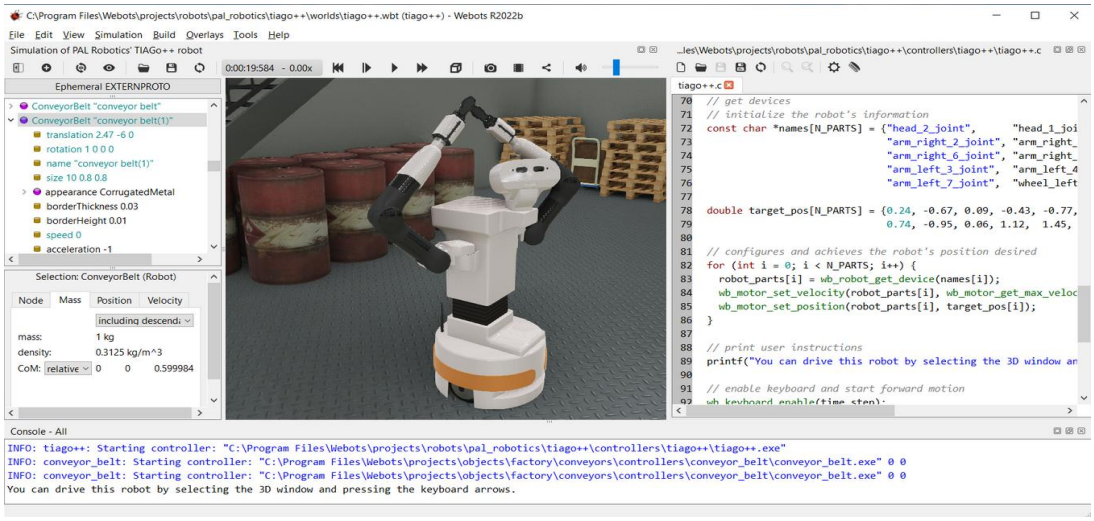

Рисунок 6 – Интерфейс Webots

Например, при работе с промышленными манипуляторами можно использовать уже готовые модели, такие как UR5, и сосредоточиться непосредственно на разработке алгоритмов управления и оптимизации движения. Кроме того, платформа обеспечивает удобные средства визуализации и отладки. Пользователь может в реальном времени наблюдать за поведением робота, анализировать траектории движения, взаимодействие с объектами и реакцию на изменения среды. Это особенно важно при разработке сложных алгоритмов, где необходимо понимать, как именно принимаются решения и какие факторы влияют на поведение системы. Возможность пошагового выполнения симуляции и изменения параметров делает процесс отладки более прозрачным и эффективным. Следует также отметить поддержку стандарта ROS который широко используется в робототехнике. Интеграция Webots с ROS позволяет использовать существующие пакеты и инструменты, а также облегчает перенос разработанных решений на реальные робототехнические платформы. Это делает платформу не просто симулятором, а полноценной частью экосистемы разработки робототехнических систем. С точки зрения физического моделирования, платформа использует современный физический движок, обеспечивающий реалистичное поведение объектов. Это включает моделирование сил, инерции, контактов и других физических эффектов. В результате пользователь получает возможность тестировать алгоритмы в условиях, максимально приближенных к реальным. Это особенно важно для задач, связанных с манипуляцией объектами, где точность взаимодействия играет ключевую роль.

Отдельного внимания заслуживает возможность создания пользовательских сценариев и сред в Webots предоставляет гибкие инструменты для построения виртуальных миров, включая импорт моделей из внешних CAD-систем, таких как SolidWorks. Это позволяет моделировать реальные производственные линии, рабочие зоны и другие сложные среды. Таким образом, разработчик может создавать симуляции, максимально соответствующие конкретным условиям применения, что повышает достоверность получаемых результатов.

В контексте современных исследований, связанных с обучением с подкреплением, Webots выступает как эффективная платформа для тестирования алгоритмов. Возможность многократного воспроизведения сценариев, контроля параметров среды и автоматизации экспериментов делает его особенно полезным для обучения моделей, требующих большого количества итераций. Таким образом, Webots представляет собой мощный инструмент, объединяющий в себе возможности моделирования, разработки и тестирования робототехнических систем. Его преимущества заключаются в реалистичности симуляции, удобстве интеграции с современными

алгоритмами искусственного интеллекта, поддержке стандартов робототехники и гибкости настройки среды. В совокупности эти характеристики делают Webots важным элементом в процессе разработки современных роботизированных решений, особенно в задачах, связанных с промышленными манипуляторами и интеллектуальным управлением. В таблице 6 демонстрируется сравнение Webots с другими программами.

Таблица 6 – Анализ по критериям разных симуляционных программ

Критерий	**Webots**	**Gazebo**	**CoppeliaSim**	**MuJoCo**
Тип системы	Универсальный симулятор	ROS-ориентированная среда	Модульная платформа	Физический движок
Физическая точность	Высокая	Средняя	Средняя/высокая	Очень высокая
Скорость симуляции	Средняя	Средняя	Средняя	Очень высокая
Удобство использования	Высокое	Среднее (сложный setup)	Среднее	Низкое (требует настройки)
Интеграция с Python/RL	Отличная	Хорошая	Хорошая	Отличная
Поддержка ROS	Есть	Основная функция	Есть	Ограниченная
Готовые модели роботов	Много	Много	Много	Мало
Визуализация	Хорошая	Хорошая	Очень хорошая	Минимальная
Sim-to-Real	Хорошо поддерживается	Хорошо	Средне	Ограниченно
Настройка среды	Простая	Сложная	Гибкая	Требует программирования
Подходит для RL	Да	Да	Да	Да
Подходит для промышленности	Да	Да	Да	Частично
Требует лицензии	Нет (open-source)	Нет (open-source)	Да	Нет (open-source)

2.8 Моделирование симуляционной среды для задачи RL

В непрерывном пространстве будет робот-манипулятор с установленным хватом. Задача взять куб, лежащий в пространстве досягаемости. После захвата – сбросить его в корзину. В самой среде кроме куба и корзины и робота ничего не будет, а следовательно – робот будет

понимать все через кинематику (расстояния, хват). Для теоретического решения задачи ее вполне достаточно. Для симуляционной среды был выбран open-source программа Webots. Позволяющая в режиме реального времени запускать симуляции с учетом динамики и физики объектов. Среда позволяет создавать роботов для взаимодействия с объектами. В добавок к этому есть встроенный редактор контроллеров, и консоль вывода. Для демонстрации возможности была взята задача Pick and drop.

Задача pick and drop – это одна из самых фундаментальных задач в робототехнике и автоматизации. Её важность можно рассмотреть с нескольких сторон: Складская логистика: роботы берут товары и кладут их в коробки, корзины, конвейеры; Производство: манипуляторы перекладывают детали между станками, подают заготовки или собирают изделия; Фармацевтика и электроника: требуется аккуратное перемещение маленьких и хрупких предметов; Сервисные роботы: от роботов-официантов до медицинских ассистентов – все опираются на навык pick-and-drop. Pick-and-drop - это основа всех более сложных манипуляционных действий: сборка, сортировка, упаковка, монтаж. Если робот не может уверенно взять объект и положить его в нужное место, то любые более сложные операции становятся невозможными. В то же время это хорошая модельная задача для тестирования алгоритмов: управления (MPC, RL) – как робот строит траекторию и принимает решение. Задача также дает демонстрацию как именно агент учится последовательным действиям: подойти, захватить, перенести, отпустить. Обобщения – если алгоритм научился pick-and-drop на одном объекте, можно проверить его способность справляться с разными размерами, весами, формами.

2.9 Объекты в симуляционной среде

Робот Universal Robots 5 (UR5) – это 6-осевой коллаборативный робот-манипулятор от датской компании Universal Robots, предназначенный для автоматизации повторяющихся задач весом до 5 кг. Модель отображена на рисунке 7.

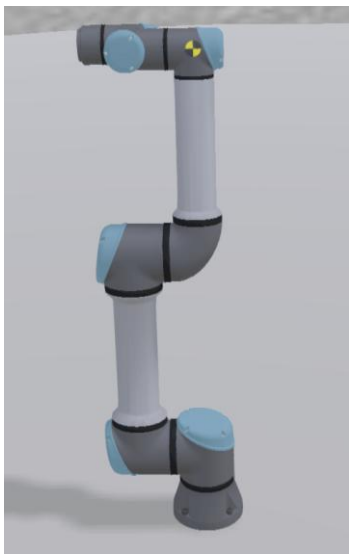

Рисунок 7 – Модель UR5e в Webots

В Webots уже есть полностью готова модель UR5e которая позволит упростить процесс создания робота на данном этапе. В таблицах 7 и 8 расписаны характеристики UR5e.

Таблица 7 – Характеристики UR5e

Раздел	Параметр	Значение
Общие характеристики	Максимальная полезная нагрузка	5 кг
Общие характеристики	Дальность (reach)	850 мм
Общие характеристики	Количество степеней свободы	6 вращающихся суставов
Общие характеристики	Повторяемость (pose repeatability)	0,03 мм
Скорость и обновление	Максимальная скорость суставов	180 °/c
Скорость и обновление	Скорость TCP	≈ 1 м/с
Скорость и обновление	Частота обновления системы	500 Hz
Вес и конструкция	Вес робота (без контроллера)	~ 20,7 кг

Продолжение таблицы 7

Вес и конструкция	Отпечаток основания / диаметр	149 мм
Контроллер и электроника	Размеры блока управления	460 × 449 × 254 мм
Контроллер и электроника	Порты I/O контроллера	16 DI, 16 DO, 2 AI, 2 AO
Контроллер и электроника	Порты I/O инструмента (Tool I/O)	2 DI, 2 DO, 2 AI
Контроллер и электроника	Питание инструмента	12 V / 24 V, до 1,5 A (dual pin), 1 A (single pin)
Датчик силы и момента	Точность (accuracy)	4 N, 0,3 Nm
Безопасность и режимы	Конфигурируемые функции безопасности	17 функций
Безопасность и режимы	Соответствие стандартам	EN ISO 13849-1 (PLd, Cat.3), EN ISO 10218-1

Таблица 8 – Название суставов UR5e

Название сустава	Диапазон (в рад.)
elbow_joint	[-3.14;3.14]
shoulder_lift_joint	[-6.28;6.28]
shoulder_pan_joint	[-6.28;6.28]
wrist_1_joint	[-6.28;6.28]
wrist_2_joint	[-6.28;6.28]
wrist_3_joint	[-6.28;6.28]

В задачах RL робот обучается через множество взаимодействий с окружающей средой - попыток захвата, ошибок, обратной связи. В таких сценариях простота, детерминированность и надёжность захвата становятся критичными. EPick как вакуумный захватчик имеет встроенный

электрический насос, поэтому ему не нужен внешний источник сжатого воздуха на рисунке 8.

Таблица 9 – Robotiq E Pick Gripper

Раздел	Параметр	Значение
Общие характеристики	Масса инструмента	~ 0,71 кг
Общие характеристики	Уровень вакуума	~ 80 %
Производительность	Вакуумный поток	~ 12 л/мин
Шум	Уровень шума	~ 64 дБ(А)
Привод / питание	Тип привода	Электрический вакуумный насос (без внешнего воздуха)
Температурные условия	Рабочая температура	5 °C - 40 °C
Температурные условия	Температура хранения	-30 °C - +60 °C
Относительная влажность	Рабочая влажность	20-80 % (без конденсации)
Защита	Класс защиты	IP4X
Электроника и интерфейс	Напряжение / связь	24 V DC, Modbus RTU (RS-485)
Конфигурация присосок	Количество присосок	1 / 2 / 4 (конфигурации)
Механика и ограничения	Центр масс / момент ограничения	Указаны в мануале (зависит от конфигурации)

Это упрощает конфигурацию среды RL и уменьшает внешние параметры, которые могут вносить шум или нестабильность. Меньше “внешних степеней свободы” – меньше источников неопределённости. Рисунок 9 показывает UR5 с установкой.

Рисунок 8 – Robotiq E Pick Gripper в Webots

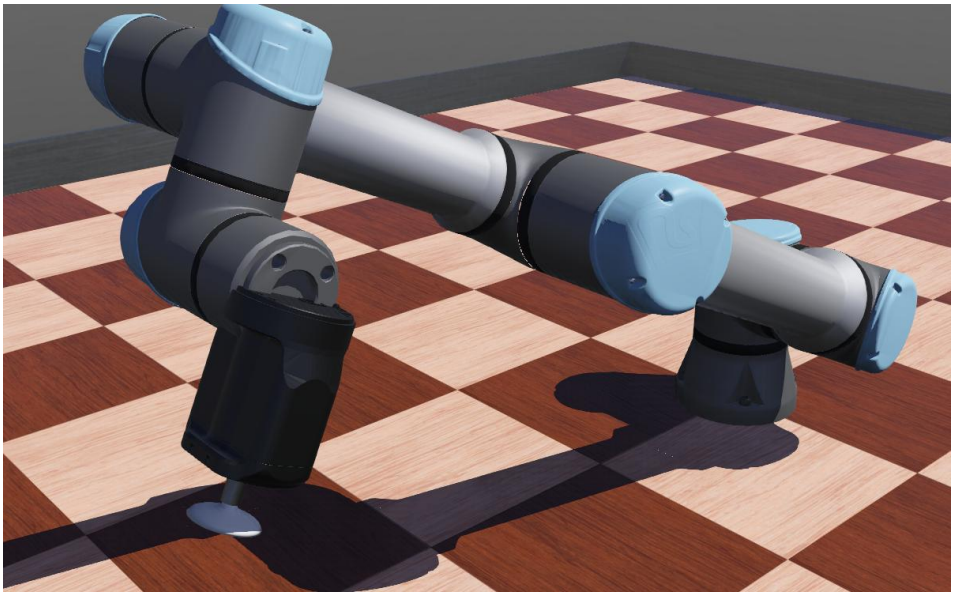

Рисунок 9 – UR5e с установленным Robotiq E Pick Gripper в Webots

На рисунке 10 показаны объекты в среде после инициализации робота. Их параметры указаны в таблице 10.

Таблица 10 – Объекты в среде

Название	Параметры
Куб	Масса: 0.15 kg, плотность: 1200 кг/м^3, цвет: зеленый, тип: Solid.
Корзина	Конструкция, состоящая из 5 объектов (BASKET_FLOOR, BASKET_W1-W4) формирующая корзину. Цвет: черно-красный.

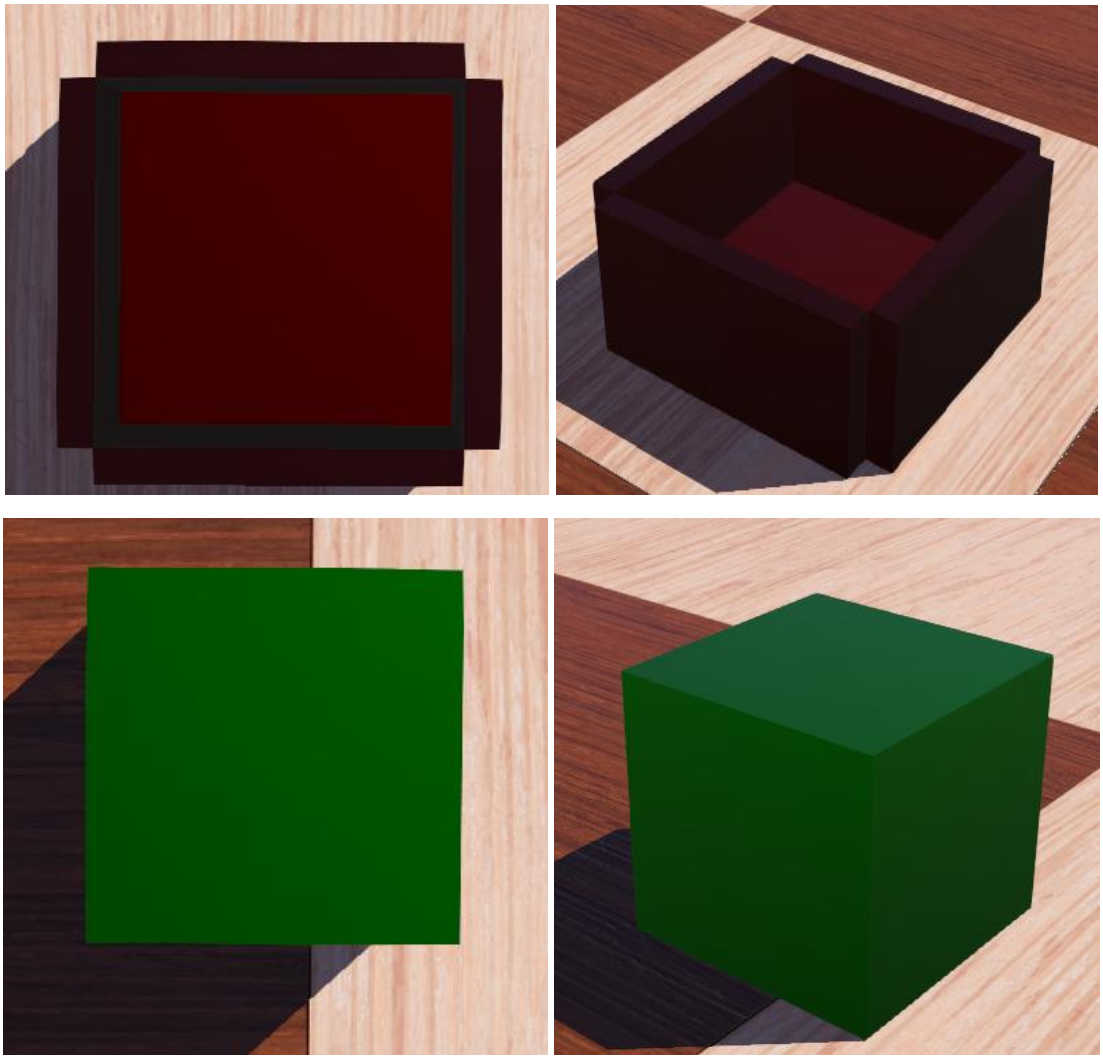

Рисунок 10 – Корзина и куб (вид сверху и сбоку)

Куб является объектом “Solid” в Webots что означает, что он рассматривается симулятором как физическое тело с массой, геометрией и инерционными свойствами, участвующее во всех вычислениях динамики. Такой объект подчиняется законам физики в движке Webots: на него действуют силы гравитации, трения, столкновений, и он может взаимодействовать с другими телами через контакты. Атрибут Solid также подразумевает, что у объекта есть своя коллизия (collision shape), определяющая, как он будет взаимодействовать при столкновениях, и своя

физическая модель (масса, центр масс, тензор инерции), что позволяет использовать его в задачах манипуляции, захвата и перемещения.

2.10 Кинематика и геометрия среды

Для разных подсчетов, связанных с расстоянием, используется прямая кинематика UR5e через библиотеку Modern Robotics (python).

- $q = [q_1, q_2, \dots, q_6]^\top$ - вектор углов суставов;
- $M \in SE(3)$ - поза TCP при нулевых углах;
- $S_i \in \mathbb{R}^6$ - винтовые оси в пространственных координатах;
- $Slist = [S_1, \dots, S_6]$.

Общая формула в таком случае является (79).

$$T_{\text{tcp}}(q) = T_{\text{base}} \cdot \exp([S_1]q_1) \cdot \exp([S_2]q_2) \cdot \dots \cdot \exp([S_6]q_6) \cdot M \cdot T_{\text{tool}} \quad (79)$$

где

- $[S_i]$ – матрица Ли из вектора винта,
- T_{tool} – сдвиг на оффсет TCP (0.16 м по Z).

TCP-координаты заданы следующей формулой (80).

$$p_{\text{tcp}}(q) = (T_{\text{tcp}}(q))_{0:3,\,3} \quad (80)$$

Для рассчета расстояния есть три дистанции.

- p_{basket} – верхняя точка центра корзины.
- p_{tcp} – позиция сустава.

Таблица 11 – Расстояния и формулы

Расстояние	Формула
TCP и Куб	$d_{\text{tcp,cube}} = \| p_{\text{tcp}} - p_{\text{cube}} \|_2$
TCP и Корзина	$d_{\text{tcp,basket}} = \| p_{\text{tcp}} - p_{\text{basket}} \|_2$
Куб и Корзина	$d_{\text{cube,basket}} = \| p_{\text{cube}} - p_{\text{basket}} \|_2$

2.11 Правила среды и ограничения

С помощью ограничении можно помочь агенту следовать определенным путям чем давать полностью свободную для перемещения среду. Ограничением можно помочь модели существенно уменьшить кол-во

начальных эпизодов для получения первых положительных сигналов от среды. В таблице 12 и 13 расписаны жесткие и мягкие ограничения.

Таблица 12 – «Жесткие» ограничения для агента

Название терминации	Описание	Формула/условие	Название штрафа
sensor_limit_violation	Лимиты суставов Если агент пытается выйти за предел указанных ему action.	Любая фаза: $q_i \notin [q_{i,\min}, q_{i,\max}]$ иначе terminate	PENALTY_LIMIT_VIOL
phase1_forbidden_motion	Разрешён только shoulder_lift_joint. Если двигаются остальные то штраф.	Фаза PICK: движение любых других суставов кроме shoulder_lift_joint. иначе terminate	PENALTY_PICK_FORBID
phase1_range_violation	Выход за пределы диапазона плеча.	Фаза PICK $q_{\text{shoulder\_lift}} \notin [-0.120, +0.120]$ иначе terminate	PENALTY_PICK_RANGE
tcp_ground_contact	TCP касается земли.	Любая фаза: $z_{\text{tcp}} \leq z_{\text{ground}} + \varepsilon$ иначе terminate	PENALTY_GROUND_HIT
too_far_from_cube	TCP слишком далеко от куба.	Фаза PICK: $d_{\text{tcp,cube}} > 1.0$ м иначе terminate	PENALTY_TOO_FAR
wrong_attach	Неправильный захват. Если TCP «держит» куб, но на самом деле далеко	Фаза PICK: $d_{\text{tcp,cube}} > R_{\text{suction}} + 0.03$ иначе terminate	PENALTY_WRONG_ATTACH

Продолжение таблицы 12

released_before_goal	Ранний сброс куба не над корзиной.	over_xy = $(x_{\text{cube}} \in [x_b - s_x/2 - m,\ x_b + s_x/2 + m]) \ \wedge \ (y_{\text{cube}} \in [y_b - s_y/2 - m,\ y_b + s_y/2 + m])$ above_rim = $(z_{\text{cube}} \geq z_b + h - m)$ ¬(over_xy ∧ above_rim) иначе terminate	PENALTY_RELEASE
time_limit	Если шагов больше чем EPISODE_MAX_STEPS (N)	steps > EPISODE_MAX_STEPS (N) иначе terminate	TIME_LIMIT_PENALTY

Таблица 13 – «Мягкие» ограничения

Название ограничения	Описание	Формула
Шаговый штраф	Штраф за выполнение и подталкивание к завершению эпизода.	$r \mathrel{+}= STEP\_PENALTY = -0.1$
Близость TCP к кубу	Чем ближе вакуумный хват тем больше награда.	$r_{\text{near\_cube}} = W_{\text{cube}} \cdot e^{-\lambda_{\text{cube}} d_{\text{tcp,cube}}}$
Близость куба к корзине	Чем ближе куб тем больше награда.	$r_{\text{near\_basket}} = W_{\text{basket}} \cdot e^{-\lambda_{\text{basket}} d_{\text{cube,basket}}}$
Близость TCP к точке сброса	Чем ближе он оказался к точке сброса тем больше награда.	$r_{\text{tcp\_drop}} = W_{\text{drop}} \cdot e^{-\lambda_{\text{drop}} d_{\text{tcp,drop}}}$
Прогресс к корзине	Бонус за уменьшение расстояния.	$r_{\text{progress}} = W_{\text{prog}} \cdot (d_{\text{prev}} - d_{\text{now}})$
Выравнивание движения TCP к корзине	Заставляет следовать к корзине.	$r_{\text{align}} = W_{\text{align}} \cdot \cos\theta$
«Воронка» (funnel) Если TCP попадает в область воронки:	Заставляет следовать к корзине.	$r_{\text{funnel}} = W_{\text{funnel}}$

Для повышения обучаемости в задаче манипуляции с объектами используется механизм геометрической воронки (funnel reward shaping). Данный метод служит для формирования промежуточного сигнала вознаграждения, который направляет агента в сторону целевого контейнера (корзины) ещё до непосредственного достижения им её отверстия.

Корзина имеет ограниченное отверстие (порядка 0.1 на 0.1 м), поэтому вероятность успешного случайного попадания манипулятора в область допустимого сброса объекта крайне мала. Чтобы смягчить дискретность награды и обеспечить dense reward, вокруг корзины вводится расширяющаяся по высоте область в виде воображаемого конуса («воронки»). Радиус дна (самой корзины) указан в (81).

$$R = 1/2\sqrt{s_x^2 + s_y^2} \tag{81}$$

Допустимый радиус на высоте z (82).

$$r_{\text{allow}}(z) = \begin{cases} R, & z \le z_{\text{rim}} \\ R + \tan(\alpha)\,(z - z_{\text{rim}}), & z > z_{\text{rim}} \end{cases} \tag{82}$$

где α - угол раскрытия «воронки» (в коде = 30 градусов). Условие: TCP внутри funnel, если (83).

$$\sqrt{(x_{\text{tcp}} - x_b)^2 + (y_{\text{tcp}} - y_b)^2} \;\le\; r_{\text{allow}}(z_{\text{tcp}}) \tag{83}$$

2.12 Пространство действии и наблюдении агента

Каждый вектор наблюдения формируется конкатенацией признаков (таблица-14).

Таблица 14 – Векторы наблюдения

Название признака	Обозначение и описание	Размер
Состояние робота	q - позиции 6 суставов UR5e (по значениям сенсоров).	6
Положение TCP	tcp - мировые координаты TCP (x,y,z).	3
Положение кубика	cube - мировые координаты центра кубика (x,y,z).	3
Центр корзины (XY)	bask_xy - координаты корзины (только x,y).	2
Фаза задачи (one-hot)	phase - вектор: [1,0] для PICK, [0,1] для DROP.	2

Для basket signal используются дополнительные особенности (таблица-15).

Таблица 15 – Особенности BasketSignal

Название признака	Обозначение и описание	Размер
Нормированный вектор между TCP и корзиной	vec_tcp2basket_norm	3
Нормированный вектор между кубом и корзиной	vec_cube2basket_norm	3
Нормированная дистанция между TCP и корзиной	dist_tcp2basket	1
Нормированная дистанция между кубом и корзиной	dist_cube2basket	1
TCP над цилиндром корзины	over_basket	1
TCP в «воронке» корзины	inside_funnel	1
Косинус выравнивания скорости TCP к цели	tcp_goal_align_cos	1

Суммарно 27 признаков. Для обучения используется унифицированный вектор 7D (6 суставов + 1 вакуум). В фазе PICK заполняется только одно поле (shoulder_lift) и вакуум, остальные нули. В фазе DROP используется всё. Суммарно размерность составляет 9 (или 2 для PICK и 7 для DROP). Такой подход позволяет унифицировать входное пространство модели и избежать необходимости построения отдельных архитектур для разных фаз манипуляции. Заполнение неиспользуемых признаков нулевыми значениями упрощает обработку данных и обеспечивает согласованность структуры входного вектора при обучении. В таблице 16 указано пространство действии.

Таблица 16 – Action space

Фаза	Признак	Размер	Описание	Диапазон
PICK	Δshoulder_lift	1	Приращение угла в суставе shoulder_lift_joint (движение вверх/вниз)	[-1,1] масштабируется на DELTA_SCALE_PICK = 0.02 рад
PICK	vaccum	1	Управление вакуумным захватом (0 = OFF, 1 = ON)	[0,1]
DROP	Δjoints	6	Приращения для всех 6 суставов (shoulder, elbow, wrist)	[-1,1] для каждого масштабируется на DELTA_SCALE_DROP = 0.04 рад
DROP	vacuum	1	Управление вакуумным захватом (0 = OFF, 1 = ON)	[0,1]

2.13 Функция награды

В каждом шаге агент получает награду как сумму нескольких компонентов (84).

$$r_t = r_{\text{near\_cube}} + r_{\text{near\_basket}} + r_{\text{tcp\_drop}} + r_{\text{progress}} + r_{\text{align}} + r_{\text{funnel}} + r_{\text{idle}} + r_{\text{step}} \quad (84)$$

Таблица 17 – Компоненты функции награды

Название	Формула	Описание
Близость TCP к кубику	$r_{\text{near\_cube}} = W_{\text{cube}} \cdot \exp(-\lambda_{\text{cube}} \cdot d_{\text{tcp,cube}})$	Чем ближе TCP к кубу тем больше положительная награда.
Близость куба к корзине (только если куб захвачен)	$r_{\text{near\_basket}} = W_{\text{basket}} \cdot \exp(-\lambda_{\text{basket}} \cdot d_{\text{cube,basket}})$	Куб должен оказаться рядом с корзиной

Продолжение таблицы 17

Близость TCP к точке сброса (в фазе DROP)	$r_{\text{tcp\_drop}} = W_{\text{drop}} \cdot \exp\left(-\lambda_{\text{drop}} \cdot d_{\text{tcp,drop}}\right)$	TCP должен тянуться к центру корзины сверху.
Прогресс (динамический бонус в DROP)	$r_{\text{progress}} = W_{\text{prog}} \cdot \left(d_{\text{prev}} - d_{\text{now}}\right)$	Если куб приближается к корзине то положительный ревард.
Выравнивание движения TCP к корзине	$r_{\text{align}} = W_{\text{align}} \cdot \cos\theta$ $\cos\theta = \frac{(p_{\text{tcp}}(t) - p_{\text{tcp}}(t-1)) \cdot (p_{\text{basket}} - p_{\text{tcp}})}{\| p_{\text{tcp}}(t) - p_{\text{tcp}}(t-1) \| \| p_{\text{basket}} - p_{\text{tcp}} \|}$	Если TCP движется в сторону корзины, то положительный ревард.
Funnel reward	$r_{\text{funnel}} = \begin{cases} W_{\text{funnel}}, & p_{\text{tcp}} \in \text{funnel} \\ 0, & \text{иначе} \end{cases}$	Ревард за нахождение TCP в «воронке» корзины
Idle penalty	$r_{\text{idle}} = 0$	Игнорируется, штраф за повторяющие действия.
Step penalty	$r_{\text{step}} = \text{STEP\_PENALTY}$	Отрицательный, чтобы подталкивать к быстрому выполнению задачи.

2.14 Применение роботов для промышленных задач

2.14.1 Задача дуговой сварки

В условиях современной промышленности особое значение приобретает автоматизация технологических процессов, требующих высокой точности, повторяемости и стабильности качества. К таким процессам относятся дуговая

сварка и промышленная маркировка изделий, включая нанесение идентификационных обозначений, таких как VIN-коды. Использование коллаборативных роботов, в частности ERA Cobot, позволяет эффективно решать данные задачи, обеспечивая сочетание гибкости, безопасности и высокой производительности.

Одним из ключевых направлений применения ERA Cobot является дуговая сварка. Данный процесс предъявляет строгие требования к точности позиционирования инструмента, равномерности движения и синхронизации подачи сварочного тока. Робот позволяет реализовать эти требования за счёт программируемого управления траекторией и параметрами движения.

Применение робота в сварке начинается с этапа обучения. Оператор вручную задаёт траекторию движения сварочной горелки, перемещая манипулятор в нужные позиции. Это позволяет быстро формировать сложные сварочные пути без необходимости глубокого программирования. После сохранения траектории робот способен воспроизводить её с высокой точностью и стабильностью. Демонстрация показана на рисунке 11.

В процессе выполнения сварки робот обеспечивает постоянную скорость движения вдоль шва также, как и точное соблюдение геометрии траектории и синхронизацию движения с включением и отключением сварочной дуги.

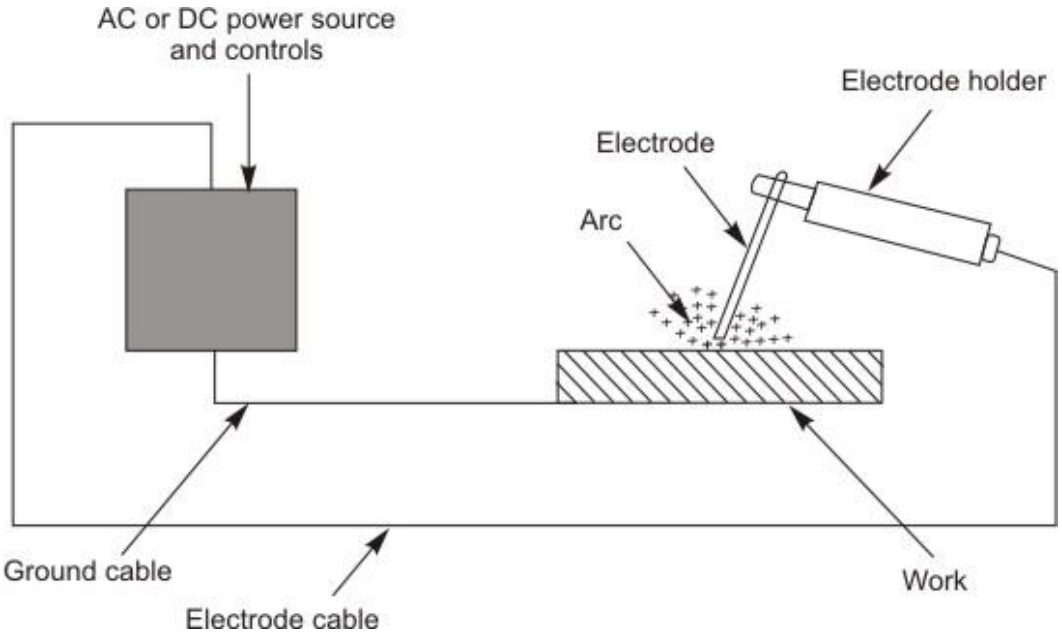

Рисунок 11 – Схема демонстрация дуговой сварки

Это позволяет существенно снизить влияние человеческого фактора, который часто приводит к дефектам сварных соединений. Кроме того, робот

может работать в условиях, опасных для человека, таких как высокая температура, яркое излучение и вредные испарения.

Важным преимуществом кобота является возможность настройки параметров безопасности, включая контроль столкновений и адаптивное поведение при возникновении внештатных ситуаций. Это делает его пригодным для использования в коллаборативной среде, где робот может работать рядом с оператором.

Вторым важным направлением применения кобота является промышленная маркировка. В отличие от сварки, где основное внимание уделяется термическому воздействию, в задачах маркировки ключевую роль играет точность позиционирования и плавность движения.

Маркировка может включать нанесение текстовых символов, серийных номеров или графических элементов на поверхность изделия. Для этого требуется точное следование заданной траектории с минимальными отклонениями. Робот обеспечивает это за счёт высокой повторяемости движения и возможности точного контроля скорости.

Процесс внедрения робота в задачу маркировки аналогичен сварке. Сначала оператор задаёт траекторию вручную или с помощью программного интерфейса. Затем эта траектория сохраняется и может многократно воспроизводиться без потери качества. Это особенно важно при серийном производстве, где требуется одинаковое нанесение маркировки на большое количество изделий.

Дополнительным преимуществом является возможность интеграции робота с системами компьютерного зрения. Это позволяет автоматически определять положение объекта и корректировать траекторию маркировки в реальном времени. Таким образом, достигается высокая адаптивность системы к изменениям в рабочей среде.

Использование ERA Cobot M5 для сварки и маркировки демонстрирует универсальность данного решения. Один и тот же робот может выполнять различные технологические операции при смене инструмента и программы. Это значительно снижает затраты на оборудование и повышает гибкость производства. К основным преимуществам применения робота можно отнести:

- высокую точность и повторяемость выполнения операций
- снижение количества ошибок, связанных с человеческим фактором
- повышение безопасности производственного процесса
- сокращение времени обучения и перенастройки
- возможность интеграции с современными цифровыми системами управления

Таким образом, ERA Cobot представляет собой эффективное решение для автоматизации задач сварки и маркировки. Его использование позволяет

не только повысить качество продукции, но и обеспечить устойчивое развитие производственных процессов в условиях цифровой трансформации промышленности.

2.14.2 Задача маркировки

Маркировка транспортных средств, в частности нанесение VIN-номеров, является критически важным этапом производства, обеспечивающим идентификацию изделия, его прослеживаемость и соответствие международным стандартам. VIN-номер представляет собой уникальный код, который должен быть нанесён с высокой точностью, читаемостью и долговечностью. В традиционном производстве данная операция часто выполняется вручную или с использованием полуавтоматических инструментов, что приводит к ряду существенных проблем. В связи с этим всё более актуальным становится внедрение роботизированных решений, позволяющих автоматизировать данный процесс.

Процесс нанесения VIN-номера требует соблюдения ряда строгих требований. Во-первых, необходимо обеспечить точное позиционирование инструмента относительно поверхности изделия. Даже незначительные отклонения могут привести к ухудшению читаемости символов или их несоответствию стандартам. Во-вторых, важно поддерживать постоянное усилие и скорость движения инструмента. Неравномерное давление или изменение скорости приводит к различной глубине или ширине символов, что негативно сказывается на качестве маркировки. В-третьих, необходимо учитывать геометрию поверхности. В реальных условиях поверхность, на которую наносится VIN-номер, может иметь сложную форму, что требует адаптации траектории движения инструмента. Таким образом, задача нанесения VIN-номера является не просто механической операцией, а комплексным процессом, требующим высокой точности, повторяемости и адаптивности.

Робот обеспечивает высокую точность позиционирования и повторяемость движений. Это позволяет наносить символы с одинаковыми параметрами на всех изделиях, обеспечивая стабильное качество маркировки.

Робот способен поддерживать постоянную скорость движения и усилие инструмента, что исключает вариативность, характерную для ручного труда. В результате достигается равномерность нанесения символов.

Однажды заданная траектория может воспроизводиться неограниченное количество раз без потери точности. Это особенно важно в условиях массового производства.

Робот может быть оснащён системой компьютерного зрения, которая позволяет автоматически определять положение объекта и корректировать

траекторию нанесения VIN-номера. Это обеспечивает адаптацию к возможным отклонениям в размещении изделия.

2.14.3 Технические характеристики робота

Робот ERA Cobot M5 представляет собой шестиосевой коллаборативный манипулятор, разработанный для выполнения высокоточных операций в промышленной среде. Его конструкция ориентирована на достижение баланса между грузоподъёмностью, точностью и гибкостью применения, что делает его подходящим для задач сварки, маркировки, сборки и обработки. Характеристики показаны на рисунке 12.

ERA	Model	ERA-M5	
Specification	Payload	5kg	
	Reach	922mm	
	Range/Axis	6	
Movement	Repeatability	±0.02mm	
	Axis Movement	Working Range	Working Speed
	Axis 1	±175°	±180°/s
	Axis 2	+85°/-265°	±180°/s
	Axis 3	±160°	±180°/s
	Axis 4	+85°/-265°	±180°/s
	Axis 5	±175°	±180°/s
	Axis 6	±175°	±180°/s
	Typical TCP Speed	1m/s	
Physical	Footprint	150mm	
	Weight	about 22kg	
	Operation Temperature	0-45℃	
	Operation Humidity	90%RH(non consensing)	
	Materials	Aluminum, Steel	

Рисунок 12 – Характеристики ERA Cobot M5

Манипулятор обладает шестью степенями свободы, что обеспечивает возможность работы в сложных пространственных условиях и выполнение операций с произвольной ориентацией рабочего инструмента. Одной из ключевых характеристик робота является его рабочий радиус, составляющий 922 мм. Это позволяет охватывать значительную рабочую зону без необходимости перемещения основания, что особенно важно при выполнении непрерывных операций, таких как сварка длинных швов или маркировка крупных деталей. Онтология и крепление робота демонстрируются на рисунке 13 и 14.

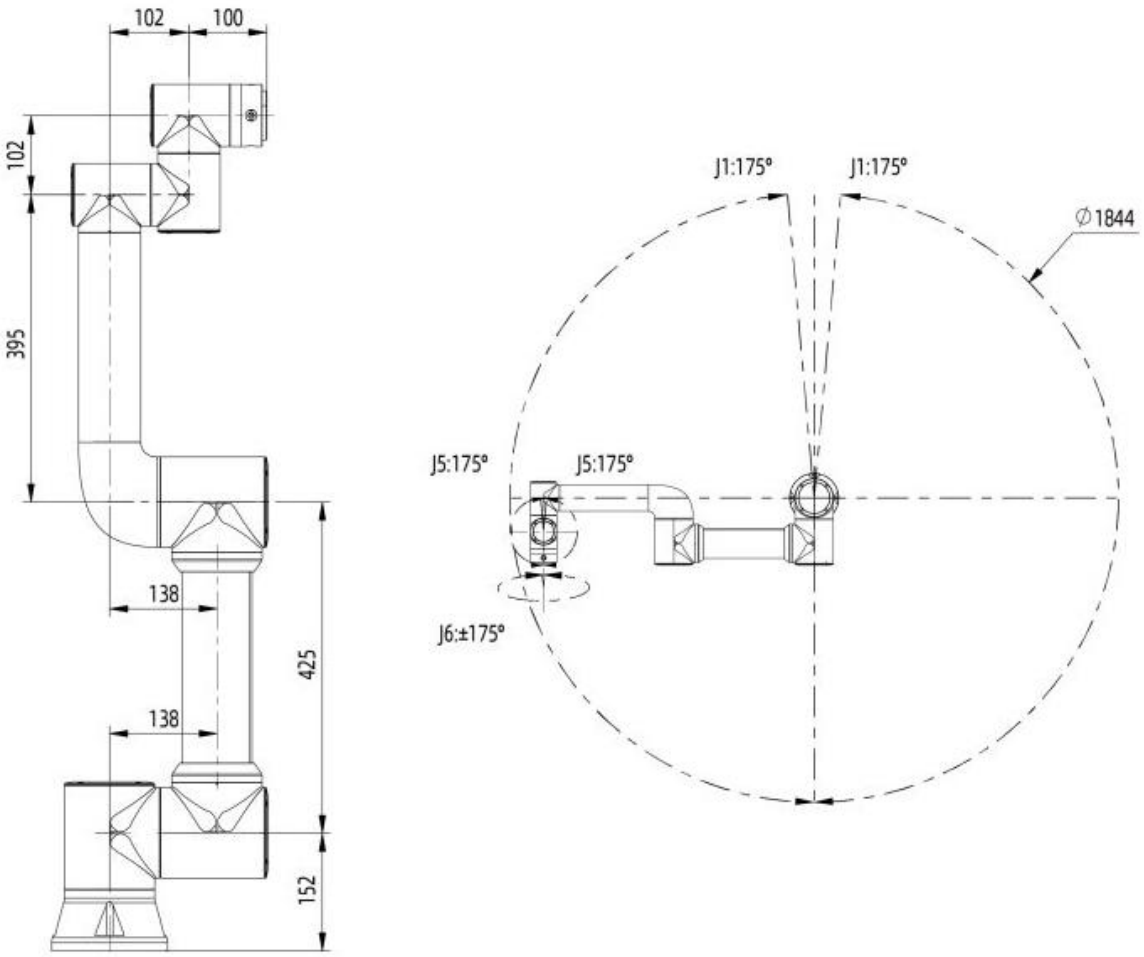

Рисунок 13 – Онтология робота

Грузоподъёмность робота составляет 5 кг, что является достаточным для большинства инструментов, используемых в промышленности, включая сварочные горелки, маркеры, гравировальные устройства и лёгкие захваты. Высокая точность обеспечивается повторяемостью на уровне 0.02 мм. Данный показатель критически важен для задач, где требуется стабильное воспроизведение траекторий, например, при нанесении маркировки или выполнении точных сварных соединений. Такая конфигурация позволяет роботу выполнять сложные пространственные движения, обходить препятствия и работать с объектами под различными углами.

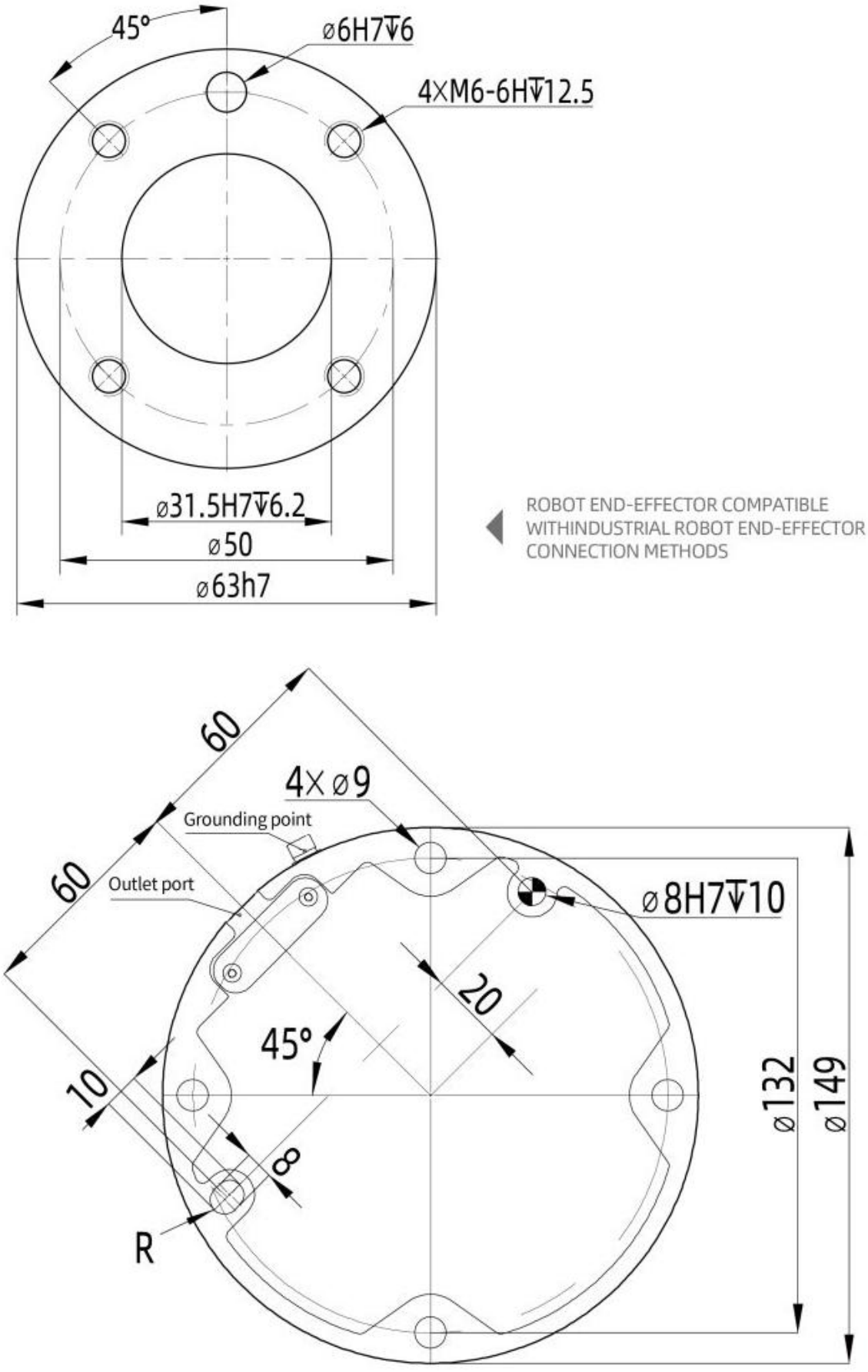

Рисунок 14 – Концевой фланец и крепление основания робота

Максимальная скорость вращения каждой оси достигает 180 градусов/с, что обеспечивает высокую динамику работы и сокращает время выполнения операций. Скорость перемещения рабочего органа (TCP) достигает 1 м/с, что позволяет эффективно использовать робот в задачах, требующих быстрого перемещения между точками, например, в серийном производстве.

Компактность робота является одним из его преимуществ. Размер основания составляет всего 150 мм, что позволяет устанавливать его в ограниченном пространстве и интегрировать в существующие производственные линии без значительной модификации.

Масса робота составляет около 22 кг, что облегчает его транспортировку, установку и обслуживание. Это особенно важно для гибких производственных систем, где требуется частая перенастройка оборудования.

Корпус робота выполнен из алюминия и стали, что обеспечивает сочетание прочности и малого веса, а также устойчивость к механическим нагрузкам и промышленным условиям эксплуатации.

Робот рассчитан на работу в широком диапазоне температур от 0 до 45C, что делает его пригодным для большинства производственных помещений.

Технические характеристики ERA Cobot M5 демонстрируют его универсальность и пригодность для широкого спектра промышленных задач. Высокая точность, достаточная грузоподъёмность, широкий диапазон движений и компактные размеры делают его эффективным инструментом для автоматизации процессов, требующих как гибкости, так и стабильности выполнения операций.

3 РЕЗУЛЬТАТЫ

3.1 Результаты задачи MPC

Для эксперимента будут выбраны оптимальные по парето-фронту значения. На рисунке 15 выделено красным. Для репрезентативности все данные отображены в таблице 18.

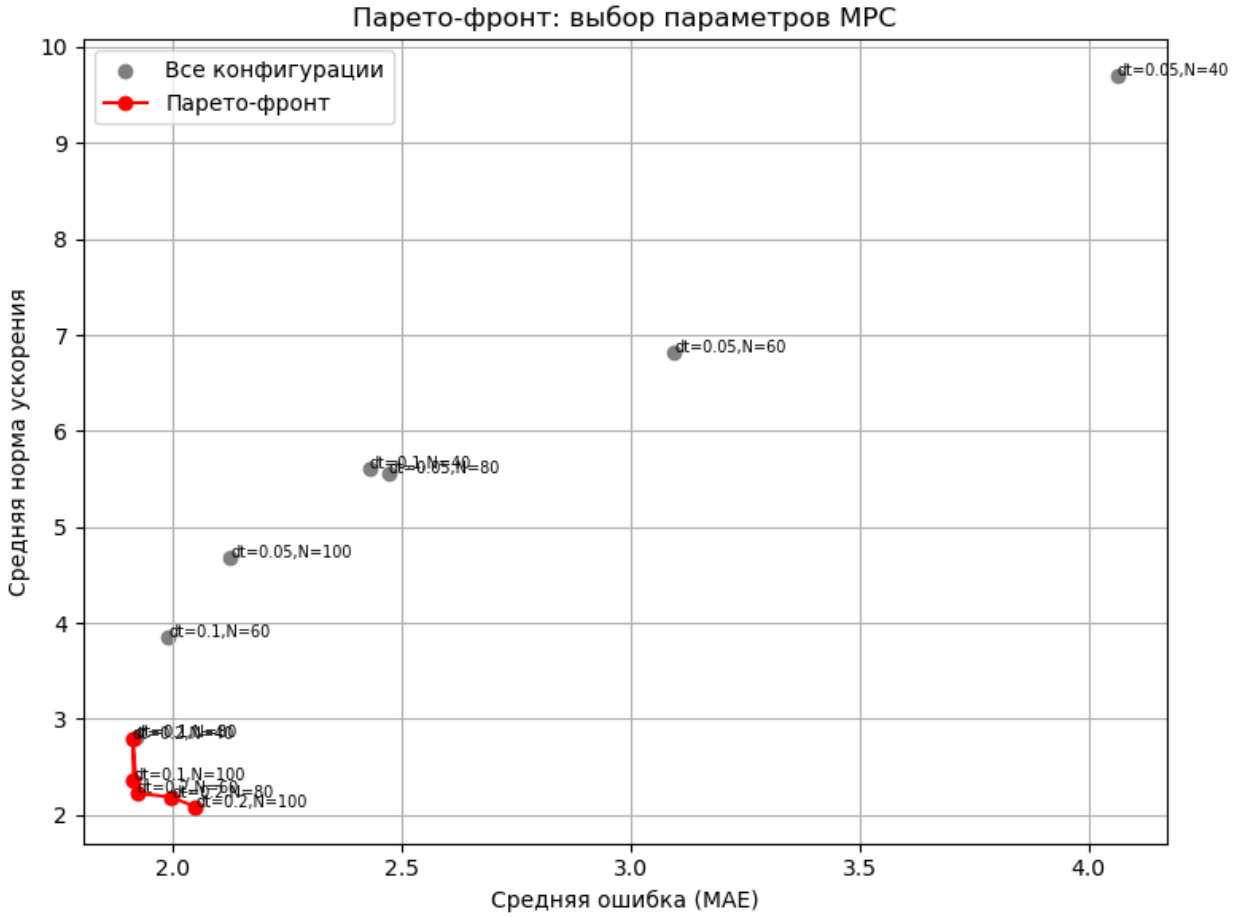

Рисунок 15 – Парето-фронт для критериев MSE и средней нормы ускорения

Таблица 18 – Оптимальные значения по парето-фронту

№	dt	N	MAE	Mean_Accel
0	0.05	40	4.062607	9.693824
1	0.05	60	3.095144	6.812185
2	0.05	80	2.471065	5.565629
3	0.05	100	2.126994	4.681773
4	0.10	40	2.429257	5.612403
5	0.10	60	1.991383	3.858621
6	0.10	80	1.919954	2.804109
7	0.10	100	1.915239	2.359426
8	0.20	40	1.913472	2.784551

Продолжение таблицы 18

9	0.20	60	1.922416	2.226548
10	0.20	80	1.997126	2.185949
11	0.20	100	2.050234	2.085049

Отсюда можно вынести следующие выводы: чем больше значение шага времени – тем ускорение маленькое (11 запись), из-за чего стоит брать N – больше. Наибыстрейшей траектории ускорения можно через наименьшие параметры, однако ошибка будет высокой сравнительно с другими. Наименьшее же значение является в 6 по 8 записи. Однако, если важно ускорение с ошибкой, можно учитывать при N = 40 или 80 (+/- 0.02) и более плавное через N=100.

Для дальнейшей интерпретации можно взять запуски №0 (наиболее высокое ускорение), №11 (наименьшая скорость) и №6 (среднее между ускорением и ошибки).

3.1.1 Эксперимент с наиболее высоким ускорением

Как видно на рисунках 16, 17, 18 и 19 при таких значениях, MPC не может полностью пройтись по всем точкам что означает что высокое ускорение достигается при помощи игнорирования цели. Что касается метрик, видно, как MPC на фоне остальных не успевает достичь результата и имеет наивысшую ошибку. А скорость - результат среднего на всем отрезке. Помимо этого, метрика Max Error показывает 6.097 у MPC, сравнительно у Linear – 2.226, Сплайны – одинаково по нулям.

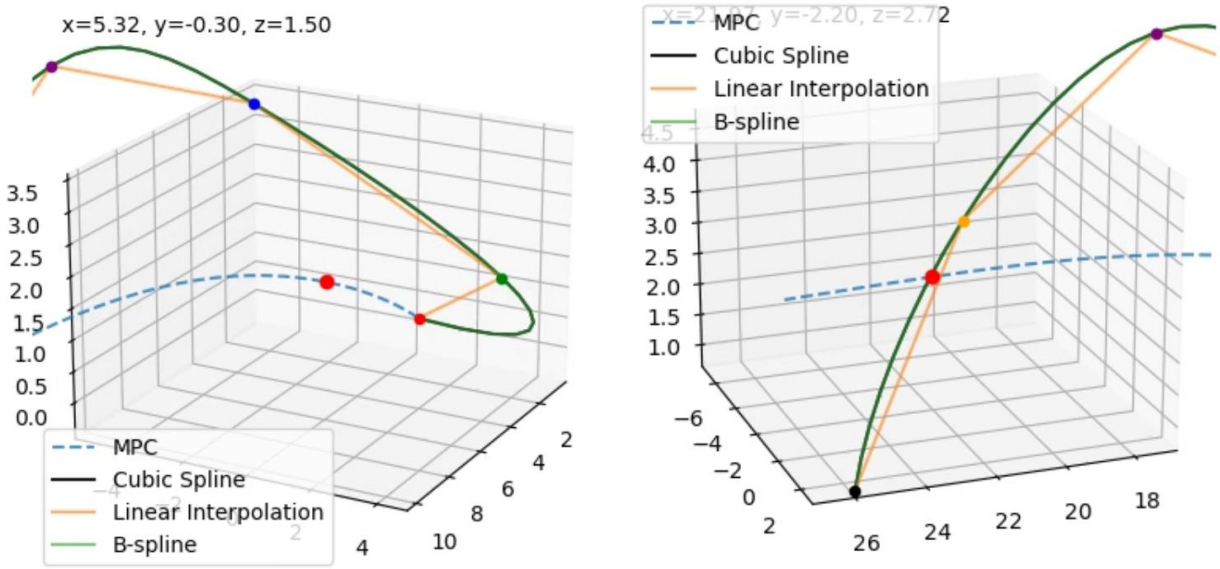

Рисунок 16 – Визуализация траектории для эксперимента-0

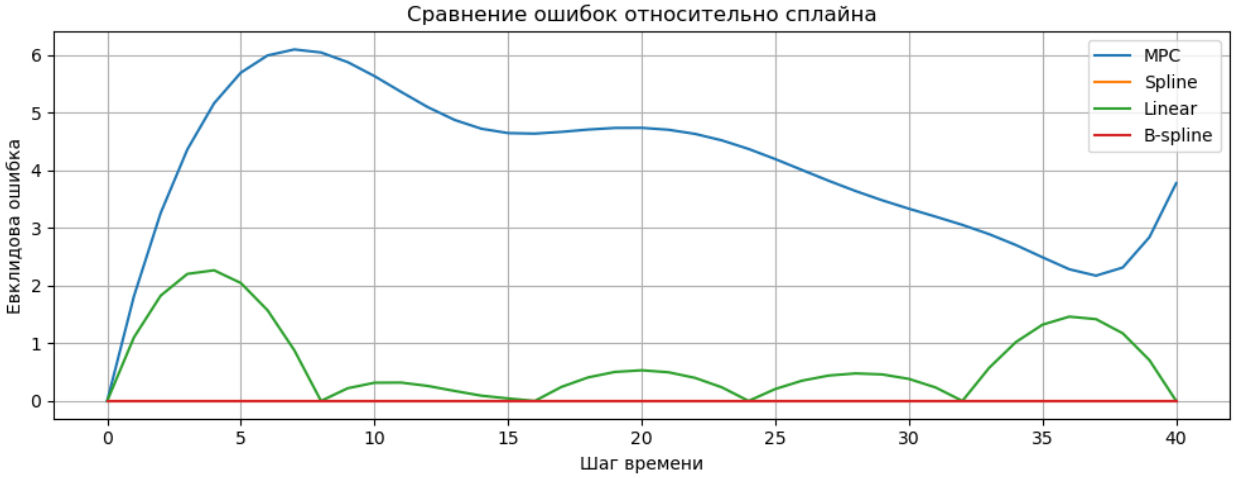

Рисунок 17 – Визуализация сравнения ошибок для эксперимента-0

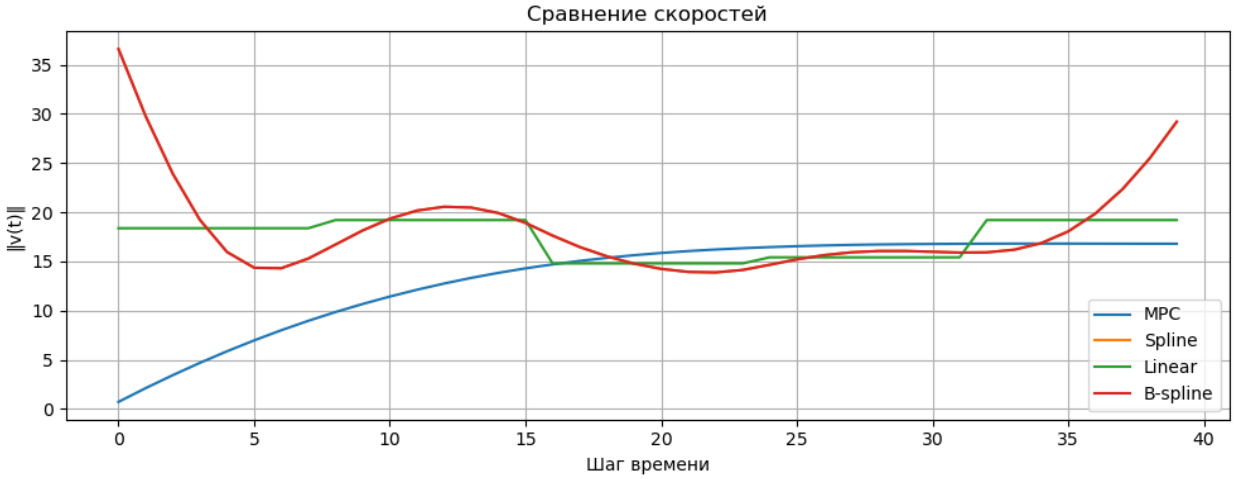

Рисунок 18 – Визуализация сравнения скоростей для эксперимента-0

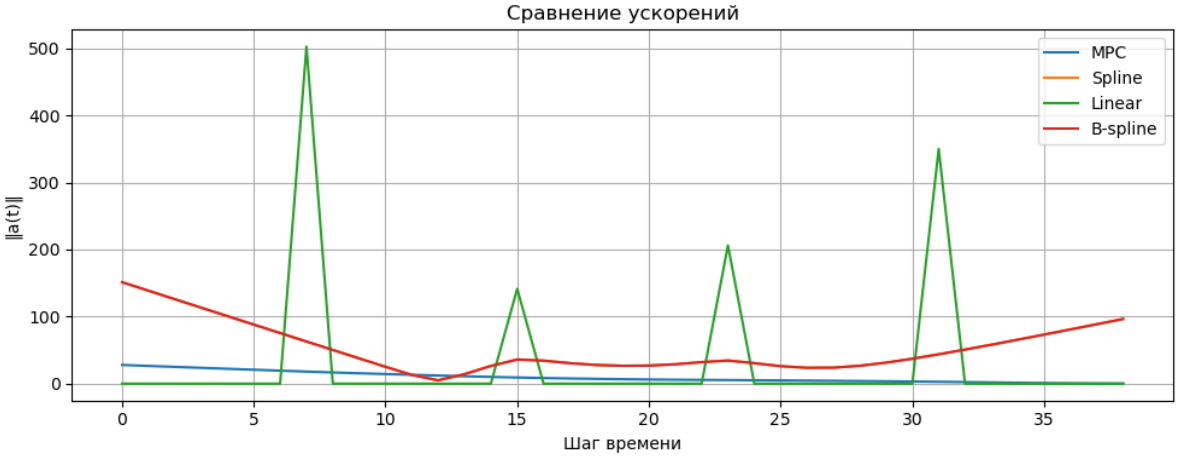

Рисунок 19 – Визуализация ускорений для эксперимента-0

3.1.2 Эксперимент с минимальной средней ошибкой

Теперь, при таких значениях, MPC на рисунках 20, 21, 22 и 23 уже успешней проходит по заданным точкам. И имеет более оптимальную для

задач манипулятора “кривую”. Которую он может проходить с приближением близкой к сплайну. Видно, как MPC оптимальной может пройтись по точкам особенно в начале. Метрика Max Error снизилась до 4.456 сравнительно на ~1.5 пункта.

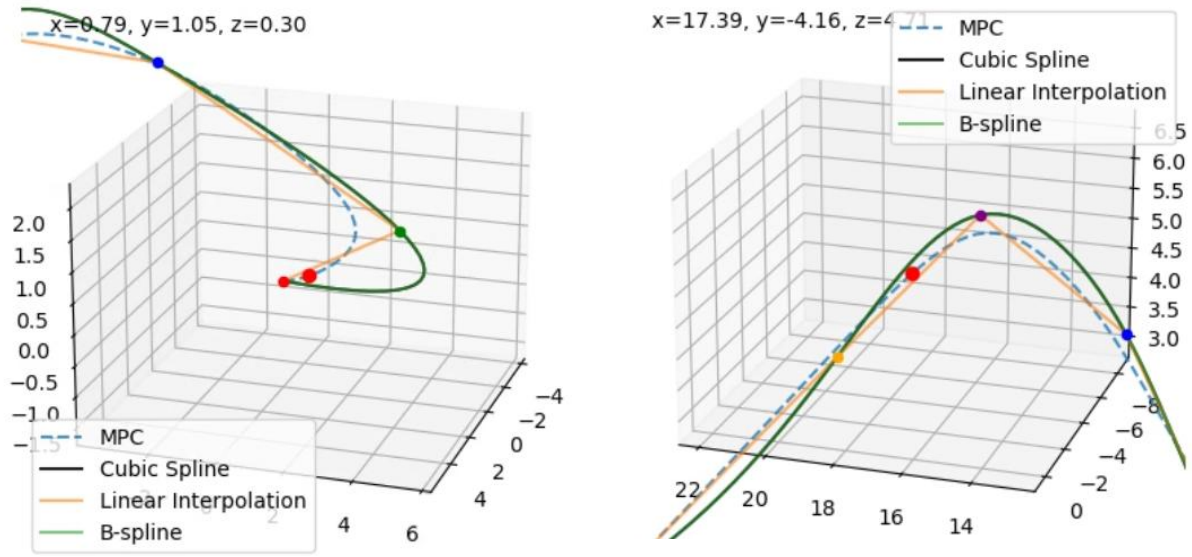

Рисунок 20 – Визуализация ускорений для эксперимента-6

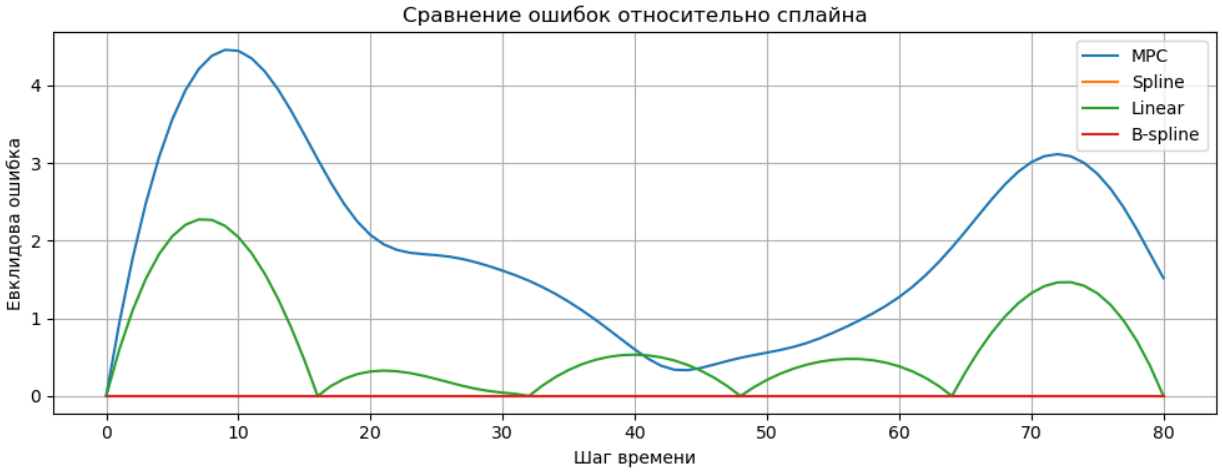

Рисунок 21 – Визуализация сравнения ошибок для эксперимента-6

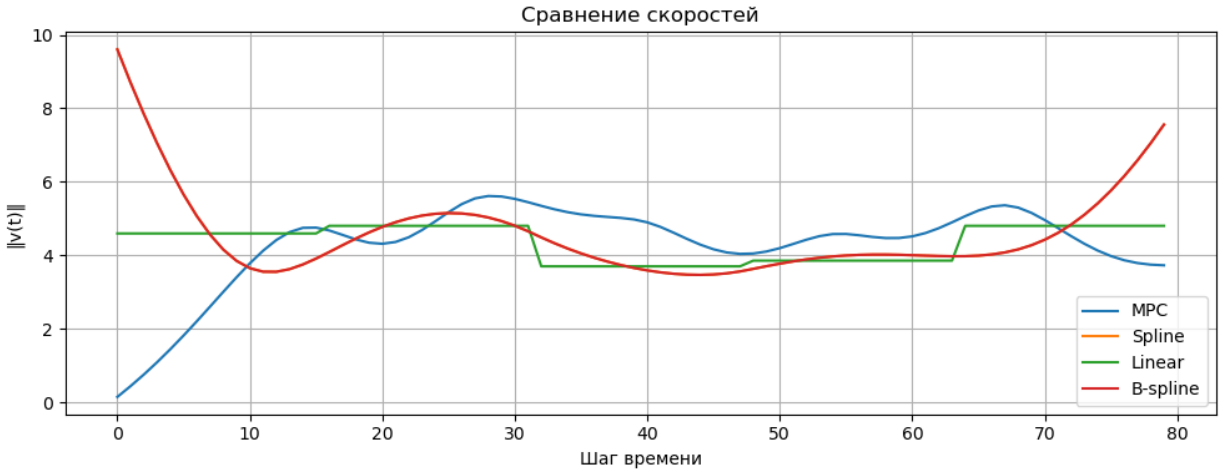

Рисунок 22 – Визуализация сравнения скоростей для эксперимента-6

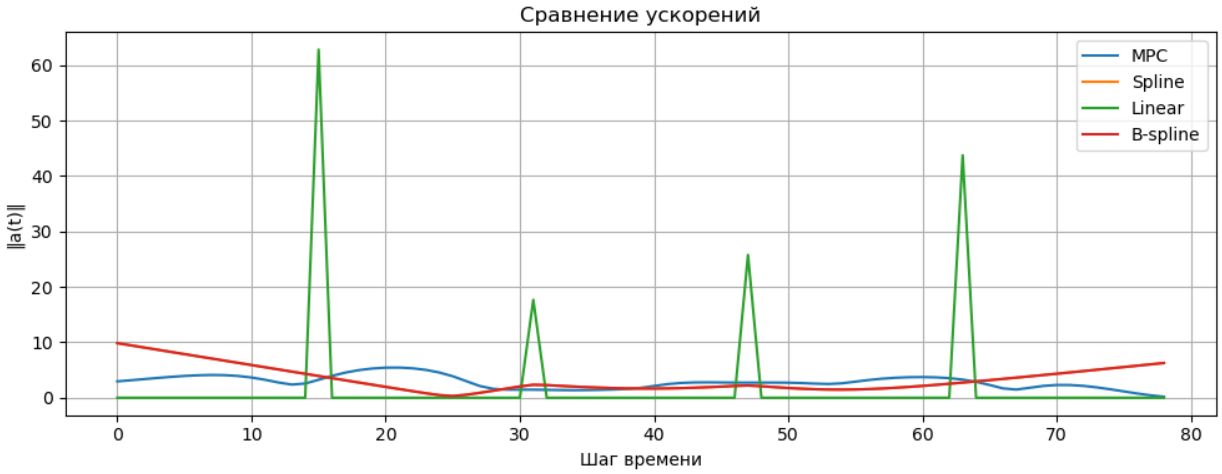

Рисунок 23 – Визуализация ускорений для эксперимента-6

3.1.3 Эксперимент с минимальным ускорением

MPC “повторяет” траекторию линейной интерполяции, тем самым лишь немного отклоняясь от его траектории. На рисунках 24, 25, 26 и 27 Можно заметить, как MPC используется те же траектории. Max error при этом не увеличилась до 5.821 что нельзя считать пока оптимальной в сравнении с экспериментом-6. В целом на всем отрезке MAE показывает 2.050.

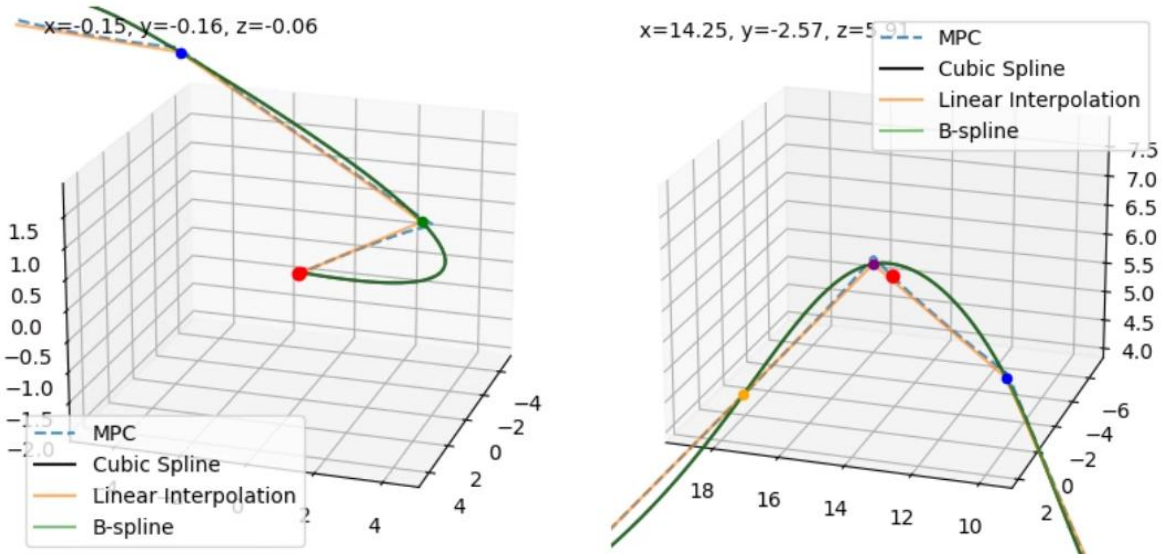

Рисунок 24 – Визуализация траектории для эксперимента-11

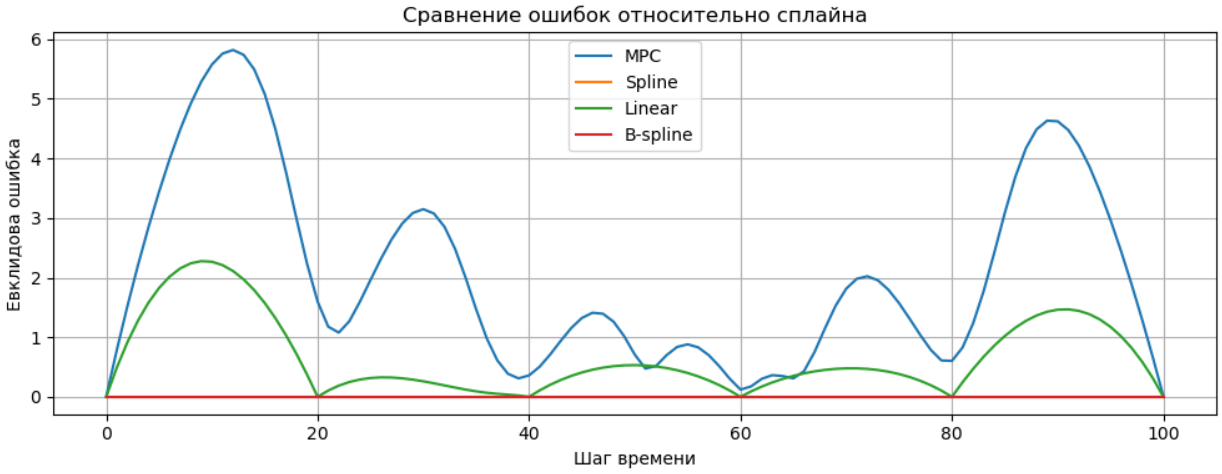

Рисунок 25 – Визуализация сравнения ошибок для эксперимента-11

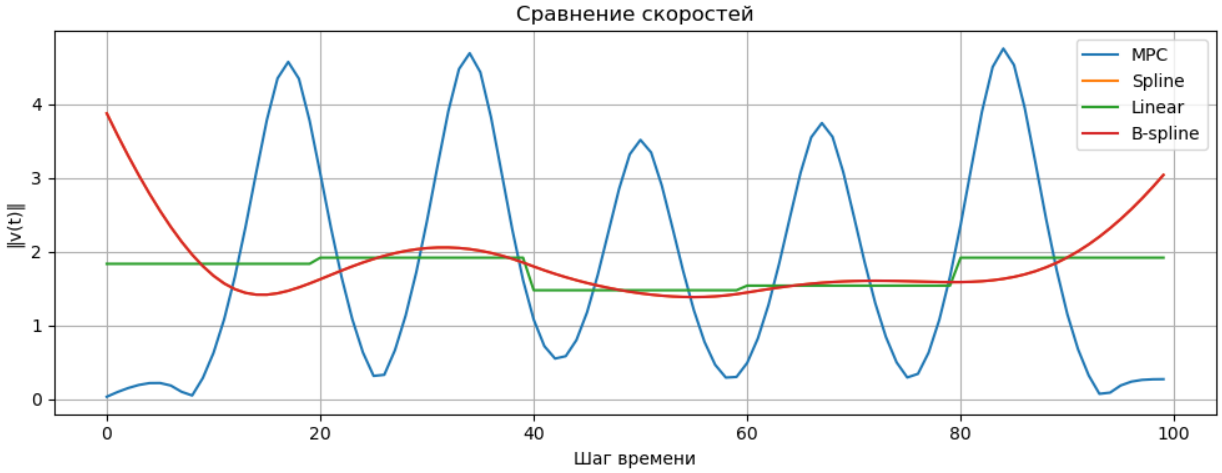

Рисунок 26 – Визуализация сравнения скоростей для эксперимента-6

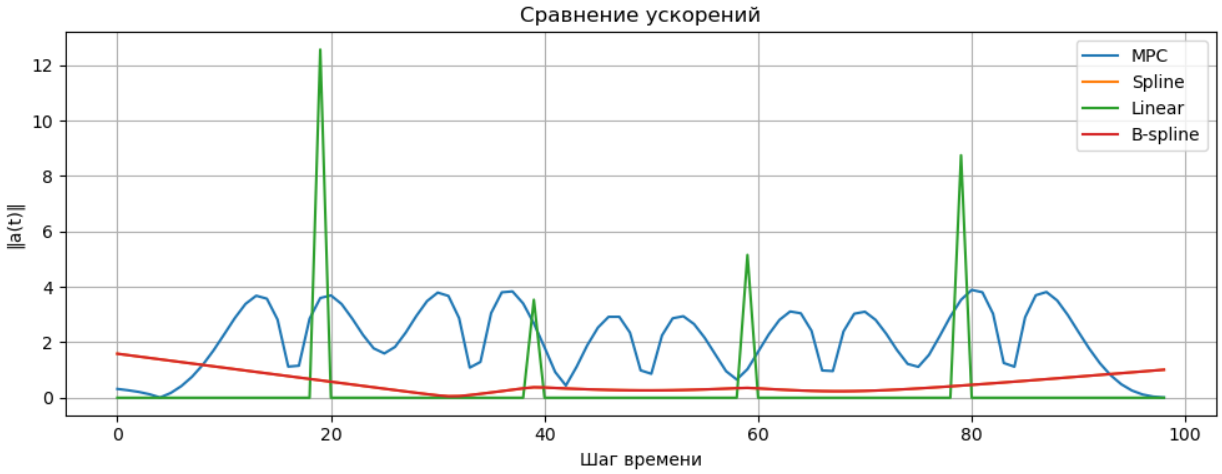

Рисунок 27 - Визуализация ускорений для эксперимента-6

3.2 Результаты задачи обучения с подкрепления

Траектория для одной цели. Обучение производилось в два этапа. Для репрезентативности, у агентов есть лишь 200 эпизодов. Для первого эксперимента у агента лишь одна цель, и его задача, лишь обойдя препятствия достичь цели. Тестирование проходило на видеокарте NVIDIA RTX 4070 12GB с использованием CUDA, тем самым время на обучение заняло около 2-х часов. Результаты отображены на рисунке 28. Из рисунка можно понять, как первые эпизоды агент пытается понять в какую сторону двигаться и получать положительную награду. Начиная с 60-го эпизода видно, как агенту удается найти идеальную траекторию. На 100-ом эпизоде агент также сохраняет тенденцию идеальной прямой. Однако с 144 по 156-е эпизоды агент пытается найти и другой путь до цели, однако к итоговому 200-му эпизода он возвращается к прямой траектории. Данная динамика свидетельствует о постепенной стабилизации политики агента и формировании устойчивого поведения в среде. Кратковременные отклонения траектории в промежуточных эпизодах могут быть связаны с процессом исследования альтернативных маршрутов и попыткой поиска потенциально более эффективных решений. В конечном итоге возврат к прямолинейной траектории подтверждает, что модель смогла определить оптимальный путь достижения цели с минимальными затратами времени и ресурсов.

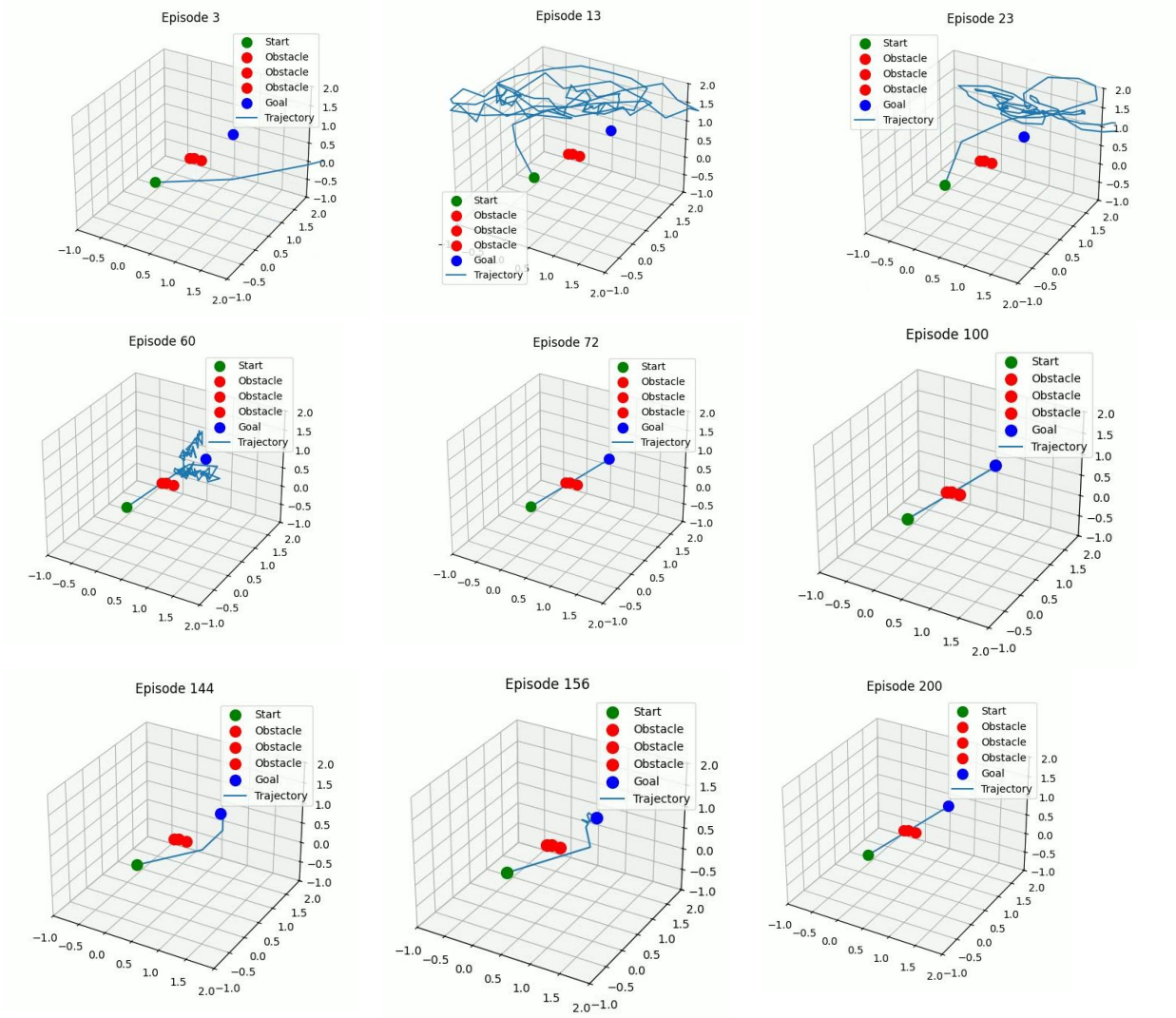

Рисунок 28 – Демонстрация генерация траектории агента для одной целевой точки

Траектория для нескольких целей. Для второго эксперимента было увеличено количества наград, тем самым агенту теперь стоит пройтись по всем целевым точкам. Для простоты визуализации угол обзора был изменен и показан на рисунке 29. Здесь стоит обратить внимание на то, как агент старается пройтись по всем точкам. На эпизоде 74 видно, как агент находит траектории для прохождения по целевым точкам. В эпизоде 141 как он стабильно начинает проходить и наконец на 199-м эпизоде продолжает тенденцию. Однако стоит упомянуть что можно достичь и наилучших результатов увеличив эпизоды.

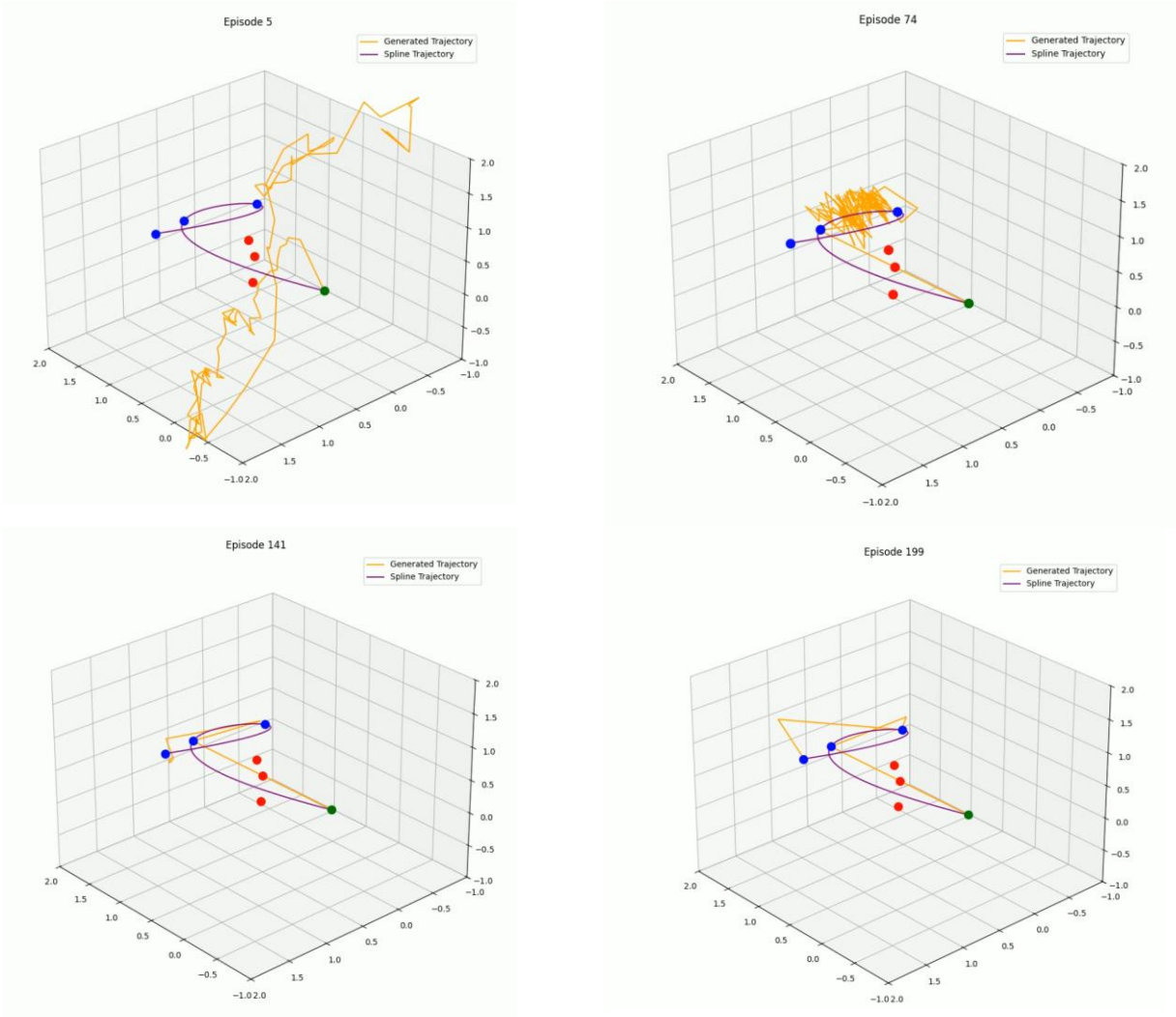

Рисунок 29 – Демонстрация генерация траектории агента для множества целевых точек

3.3 Параметры запуска эксперимента в сим. среде

По представленной таблице были определены ключевые параметры среды обучения и ограничения управления роботизированным манипулятором. Установлены диапазоны допустимого движения суставов, условия завершения эпизода и параметры возврата робота в исходное положение, что обеспечивает стабильность и безопасность процесса обучения. Система вознаграждений и штрафов была настроена таким образом, чтобы стимулировать успешный захват объекта, точное перемещение к корзине и минимизацию ошибок, связанных с выходом за пределы суставов, неправильным отпусканием объекта или столкновением с землёй. Дополнительно введены коэффициенты близости к кубу, корзине и целевой точке сброса, что позволило направлять траекторию движения агента и улучшать качество управления. Ограничение на количество шагов эпизода и критерий успешных последовательных эпизодов использовались для контроля

завершения обучения и оценки устойчивости модели. В совокупности данные параметры формируют сбалансированную среду обучения для задач захвата, перемещения и точного позиционирования объекта роботизированным манипулятором.

Таблица 19 – Параметры запуска эксперимента

Параметр	Значение	Описание
REWARD_SCALE	0.001 (по умолчанию)	Масштабирование (уже применено в формулах ниже)
JOINT_LIMITS	shoulder_pan: (-0.240,1.290) shoulder_lift: (-0.690, 0.120) elbow: (-1.800, 0.240) wrist_1: (-0.500, 2.400) wrist_2: (-1.200, 1.100) wrist_3: (-0.900, 1.760)	Диапазоны движения суставов
TERMINATE_ON_LIMIT_VIOLATION	True	Эпизод завершается при выходе за пределы суставов
DISABLE_TCP_GROUND_CONTACT	True (по умолчанию)	Можно отключить терминацию при касании TCP земли
PICK_SHOULDER_LIFT_RANGE	(-0.120, 0.120)	Диапазон для подъёма плеча при захвате
PICK_OTHER_JOINT_TOL	0.02	Допуск для других суставов в фазе Pick
JOINT_HOME	pan=0.0, lift=-0.690, elbow=0.0, wrist1=0.0, wrist2=0.0, wrist3=0.0	Домашнее положение робота
HOME_TOL	0.01	Допуск отклонения от Home
HOME_STABLE_STEPS	10	Кол-во шагов для стабилизации в Home
SUCTION_ENABLE_RADIUS	0.3	Радиус активации присоса
SUCTION_ATTACH_RADIUS_HARD	0.06	Жёсткий радиус захвата куба
FAR_DISTANCE_TERMINATE	1.00	Прекращение эпизода при слишком большом удалении
GROUND_Z	0.0	Уровень земли
GROUND_EPS	0.004	Допуск для касания земли
EPISODE_MAX_STEPS	1500	Лимит шагов в эпизоде
WARMUP_STEPS	350	Шаги разогрева
POST_RELEASE_GRACE_STEPS	30	Шаги после отпускания объекта без штрафа
RELEASE_OVER_BASKET_MARGIN	0.03	Допустимый зазор при отпускании над корзиной
REWARD_PICK_SUCCESS	+0.2	Награда за успешный захват
TIME_LIMIT_PENALTY	-10.0	Штраф за превышение лимита времени
REWARD_SUCCESS	+22.0	Награда за успешное выполнение задачи
PENALTY_WRONG_ATTACH	-2.0	Штраф за неправильный захват

Продолжение таблицы 19

PENALTY_LIMIT_VIOL	-15.0	Штраф за выход за пределы суставов
PENALTY_TOO_FAR	-0.5	Штраф за слишком большое удаление
PENALTY_RELEASE	-2.4	Штраф за отпускание вне корзины
PENALTY_PICK_FORBID	-1.2	Штраф за захват в запрещённой зоне
PENALTY_PICK_RANGE	-0.8	Штраф за выход из допустимого диапазона захвата
PENALTY_GROUND_HIT	-0.6	Штраф за удар о землю
STEP_PENALTY	-0.1	Штраф за каждый шаг (time penalty)
W_DROP_TCP	10	Вес для экспоненты близости TCP к точке сброса
DROP_TCP_DECAY	9.0	Крутизна экспоненты для TCP-drop
W_NEAR_CUBE	0.10	Вес близости к кубу
W_NEAR_BASKET	0.15	Вес близости к корзине
CUBE_NEAR_DECAY	3.0	Декей экспоненты для куба
BASKET_NEAR_DECAY	4.0	Декей экспоненты для корзины
W_PROGRESS	0.08 (env)	Вес прогресса до корзины
W_ALIGN	0.03 (env)	Вес выравнивания
W_FUNNEL	0.02 (env)	Вес воронки (фокусировка траектории)
MAX_EPISODES	10000	Максимальное количество эпизодов обучения. Если `0` то ограничения нет.
SUCCESS_STREAK_REQ	5	Количество успешных эпизодов подряд, необходимых для завершения обучения.
REWARD_SCALE	0.001 (по умолчанию)	Масштабирование (уже применено в формулах ниже)
JOINT_LIMITS	shoulder_pan: (-0.240,1.290) shoulder_lift: (-0.690, 0.120) elbow: (-1.800, 0.240) wrist_1: (-0.500, 2.400) wrist_2: (-1.200, 1.100) wrist_3: (-0.900, 1.760)	Диапазоны движения суставов
TERMINATE_ON_LIMIT_VIOLATION	True	Эпизод завершается при выходе за пределы суставов
DISABLE_TCP_GROUND_CONTACT	True (по умолчанию)	Можно отключить терминацию при касании TCP земли
PICK_SHOULDER_LIFT_RANGE	(-0.120, 0.120)	Диапазон для подъёма плеча при захвате
PICK_OTHER_JOINT_TOL	0.02	Допуск для других суставов в фазе Pick
JOINT_HOME	pan=0.0, lift=-0.690, elbow=0.0, wrist1=0.0, wrist2=0.0, wrist3=0.0	Домашнее положение робота
HOME_TOL	0.01	Допуск отклонения от Home
HOME_STABLE_STEPS	10	Кол-во шагов для стабилизации в Home
SUCTION_ENABLE_RADIUS	0.3	Радиус активации присоса
SUCTION_ATTACH_RADIUS_HARD	0.06	Жёсткий радиус захвата куба

Продолжение таблицы 19

FAR_DISTANCE_TERMINATE	1.00	Прекращение эпизода при слишком большом удалении
GROUND_Z	0.0	Уровень земли
GROUND_EPS	0.004	Допуск для касания земли
EPISODE_MAX_STEPS	1500	Лимит шагов в эпизоде
WARMUP_STEPS	350	Шаги разогрева
POST_RELEASE_GRACE_STEPS	30	Шаги после отпускания объекта без штрафа
RELEASE_OVER_BASKET_MARGIN	0.03	Допустимый зазор при отпускании над корзиной
REWARD_PICK_SUCCESS	+0.2	Награда за успешный захват
TIME_LIMIT_PENALTY	-10.0	Штраф за превышение лимита времени
REWARD_SUCCESS	+22.0	Награда за успешное выполнение задачи
PENALTY_WRONG_ATTACH	-2.0	Штраф за неправильный захват
PENALTY_LIMIT_VIOL	-15.0	Штраф за выход за пределы суставов
PENALTY_TOO_FAR	-0.5	Штраф за слишком большое удаление
PENALTY_RELEASE	-2.4	Штраф за отпускание вне корзины
PENALTY_PICK_FORBID	-1.2	Штраф за захват в запрещённой зоне
PENALTY_PICK_RANGE	-0.8	Штраф за выход из допустимого диапазона захвата
PENALTY_GROUND_HIT	-0.6	Штраф за удар о землю
STEP_PENALTY	-0.1	Штраф за каждый шаг (time penalty)
W_DROP_TCP	10	Вес для экспоненты близости TCP к точке сброса
DROP_TCP_DECAY	9.0	Крутизна экспоненты для TCP-drop
W_NEAR_CUBE	0.10	Вес близости к кубу
W_NEAR_BASKET	0.15	Вес близости к корзине
CUBE_NEAR_DECAY	3.0	Декей экспоненты для куба
BASKET_NEAR_DECAY	4.0	Декей экспоненты для корзины
W_PROGRESS	0.08 (env)	Вес прогресса до корзины
W_ALIGN	0.03 (env)	Вес выравнивания
W_FUNNEL	0.02 (env)	Вес воронки (фокусировка траектории)
MAX_EPISODES	10000	Максимальное количество эпизодов обучения. Если `0` то ограничения нет.
SUCCESS_STREAK_REQ	5	Количество успешных эпизодов подряд, необходимых для завершения обучения.

3.4 Анализ результатов

В ходе обучения алгоритма TD3 с разделением на две стадии – Pick и Drop – наблюдаются характерные закономерности, отражённые на представленных графиках. Эти данные позволяют глубже понять, как агент осваивает задачу и какие трудности возникают в различных фазах. Агент, спустя 225 эпизодов и выполнения 5 подряд успешных эпизодов (SUCCESS_STREAK_REQ) завершил работу потратив на выполнение около 2 часов реального времени или приблизительно ~4 часа в симуляции на видеокарте NVIDIA RTX 4070 12GB и процессоре Intel i7-12700KF на Pytorch

с CUDA. На рисунке 30, видно, как со временем агент начал справляться и получать награду выше чем мог получать до этого.

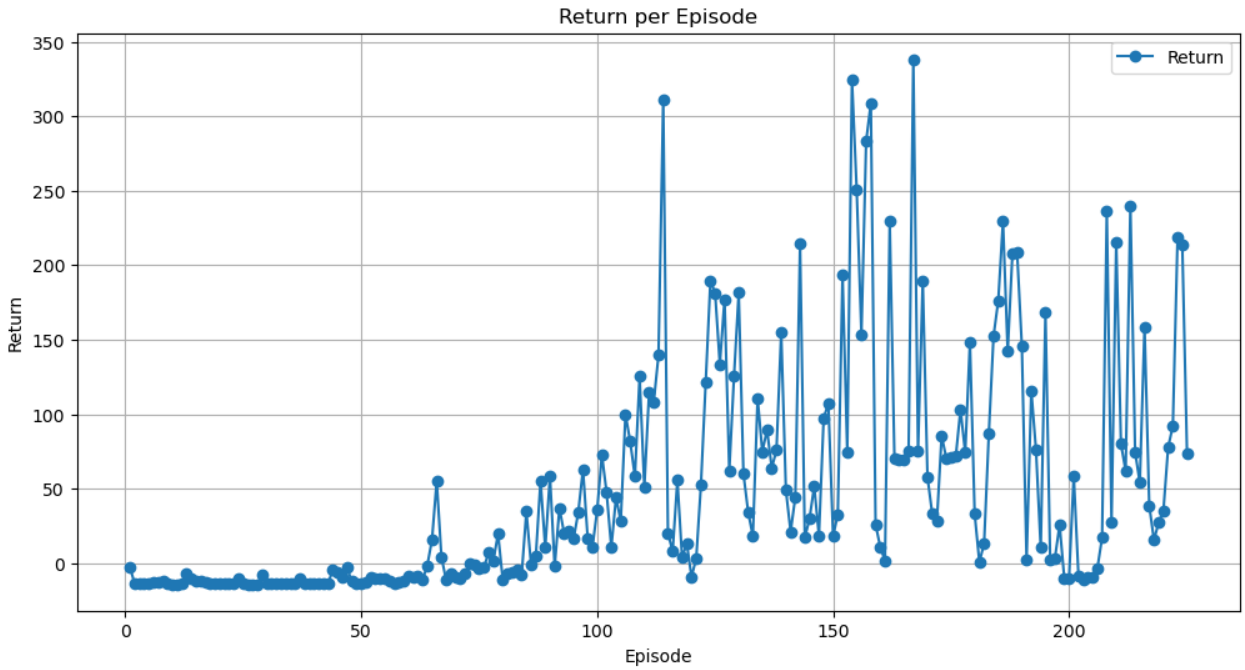

Рисунок 30 – Награды по эпизодам

Успешные эпизоды начали появляться с 150 эпизода, но смогли повторить 5 подряд лишь ближе к концу терминального эпизода как показано на рисунке 31.

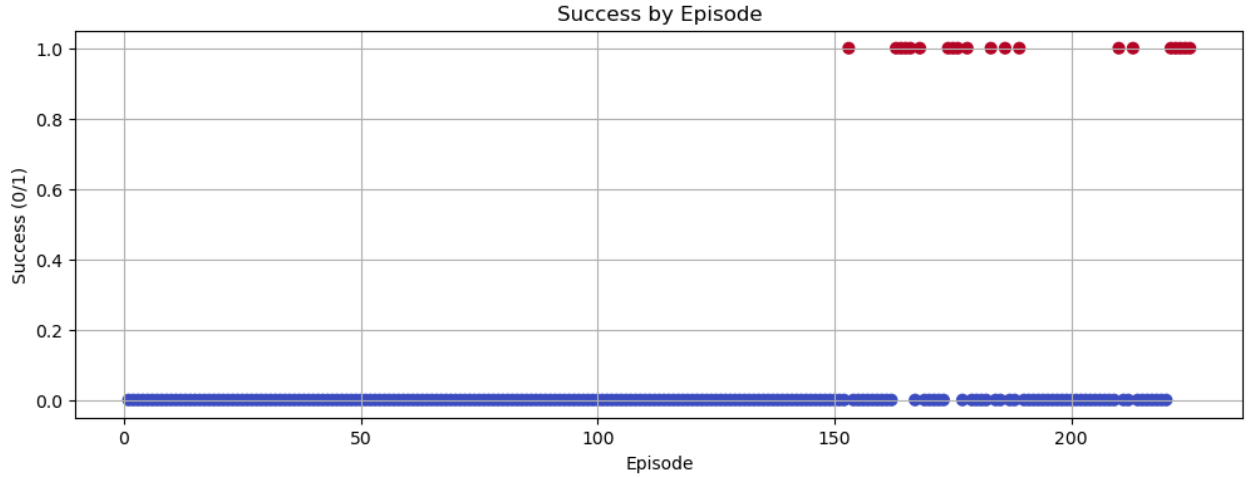

Рисунок 31 – Успешные эпизоды

Длина эпизода в моменте исследования составила почти 450 шагов перед завершением, однако ближе концу обрела более конкретный интервал как показано на рисунке 32.

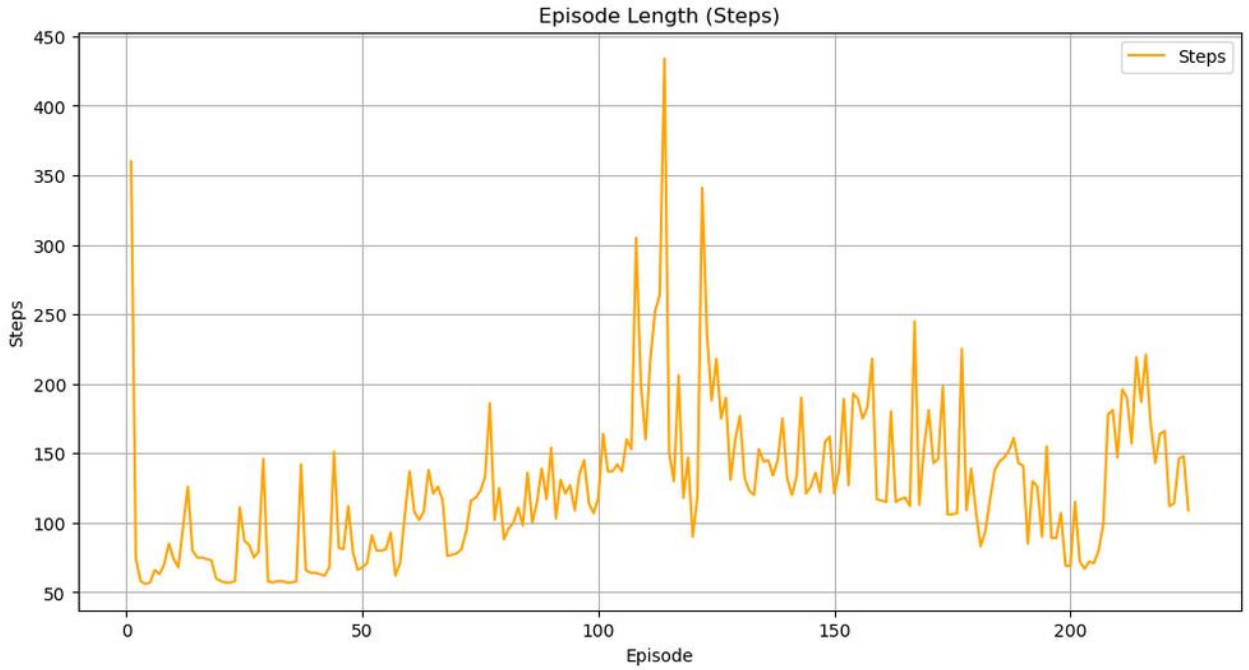

Рисунок 32 – Продолжительность эпизода

На рисунке 33 демонстрируются основные причины завершения эпизодов. Самым частым стал released_before_goal которое фиксировалась в большинстве случаев

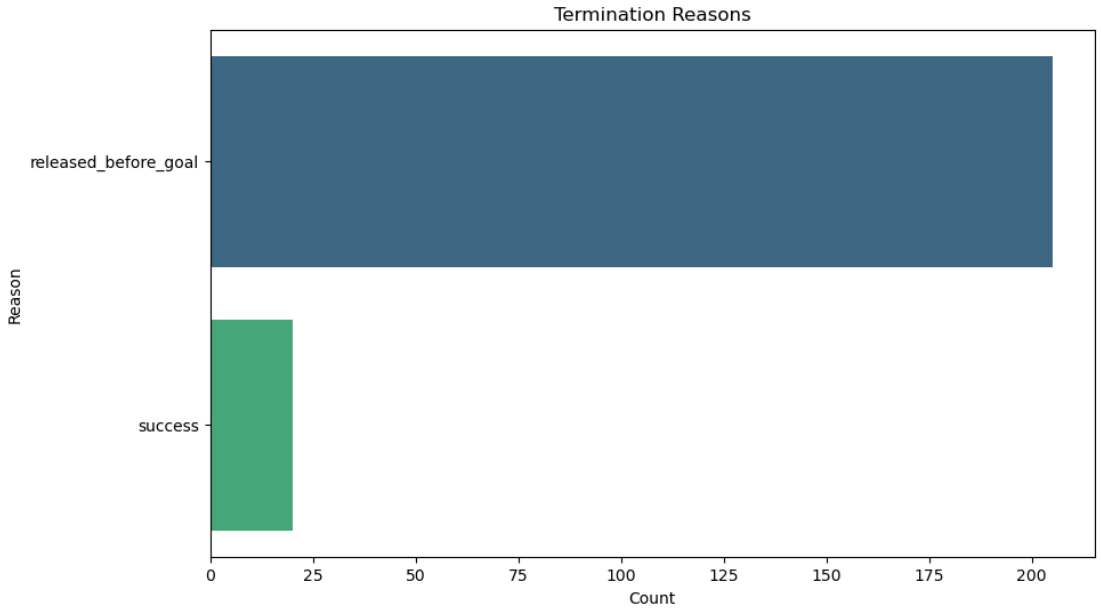

Рисунок 33 – Причины остановки эпизода

По мере роста числа шагов, после примерно 200 итераций, происходит резкое снижение ошибок и постепенная стабилизация вблизи 0–5. Это важный признак: критики достигли согласованности и начали выдавать устойчивые и согласованные оценки. Такая динамика указывает на то, что агент сформировал базовое понимание структуры среды и научился различать хорошие и плохие действия. Вторая диаграмма демонстрирует потери акторов.

Здесь разделены две модели Pick и Drop. Actor Pick отвечает за фазу захвата объекта а Actor Drop отвечает за фазу сброса объекта в корзину .Для Pick характерно быстрое достижение стабильности: ошибки актора остаются близкими к нулю на протяжении всего обучения. Это говорит о том, что задача захвата куба проще для агента. Робот относительно быстро освоил стратегию приближения и фиксации объекта. Совершенно иная картина у Drop. На ранних этапах потери актора находятся на крайне низком уровне (до –20…–25), что означает резкие и неустойчивые обновления стратегии. Агент испытывает значительные трудности в определении момента отпускания куба. Лишь после примерно 200 шагов кривая начинает подниматься и стабилизироваться ближе к –5. Этот процесс отражает постепенное обучение робота правильному таймингу и траектории сброса. Именно в этой фазе объясняются частые ошибки типа released_before_goal, зафиксированные в логах эпизодов. На рисунке 34 и 35 показывается значение функции потери.

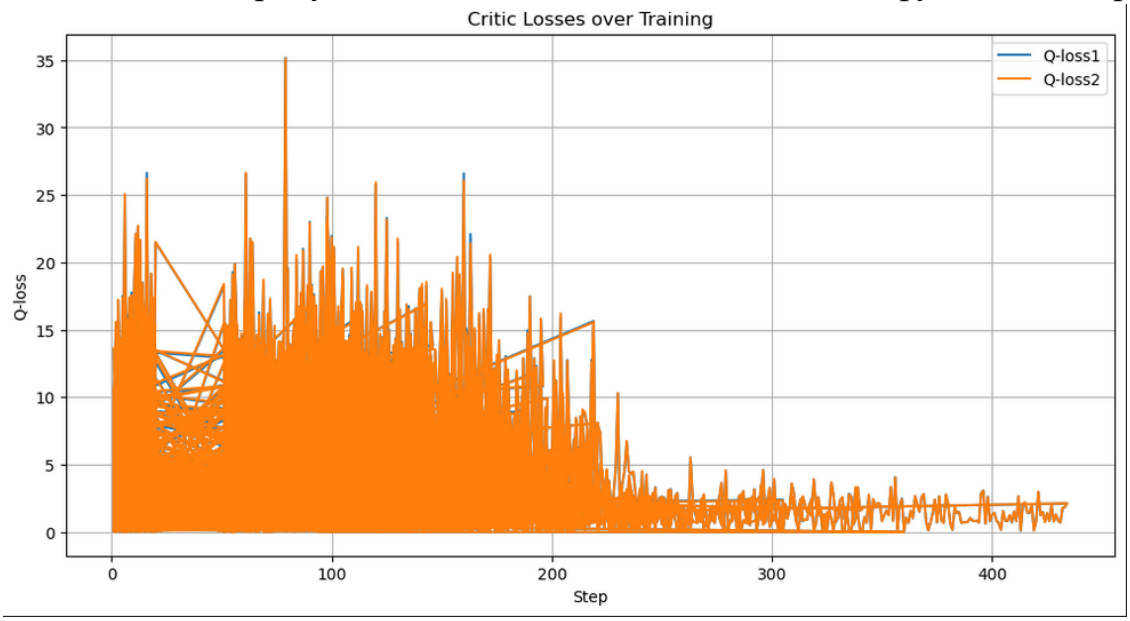

Рисунок 34 – Значения функции потерь для critic

Рисунок 35 – Значения функции потерь для actor

Рисунок 34 и 35 демонстрирует динамику шумов, добавляемых к действиям в фазах Pick и Drop. Для Pick начальный уровень шума выше (≈0.14), что стимулирует широкий спектр действий и ускоряет освоение базового поведения захвата. С течением времени шум постепенно снижается до 0.02–0.04, позволяя агенту сосредоточиться на эксплуатации найденной стратегии. Для Drop в начале шум имеет устойчивый уровень около 0.10 и снижается гораздо медленнее. Это отражает необходимость более долгого исследования: агент должен найти правильный тайминг отпускания куба, что является сложной задачей. Постепенное снижение шума к концу обучения указывает на переход от хаотичных проб к более уверенным и целенаправленным действиям. Таким образом, характер изменения шумов демонстрирует различия в сложности подзадач: Pick быстро стабилизируется, тогда как Drop требует длительной и интенсивной фазы исследования. На рисунке-36 показан рост числа обновлений акторов и критиков.

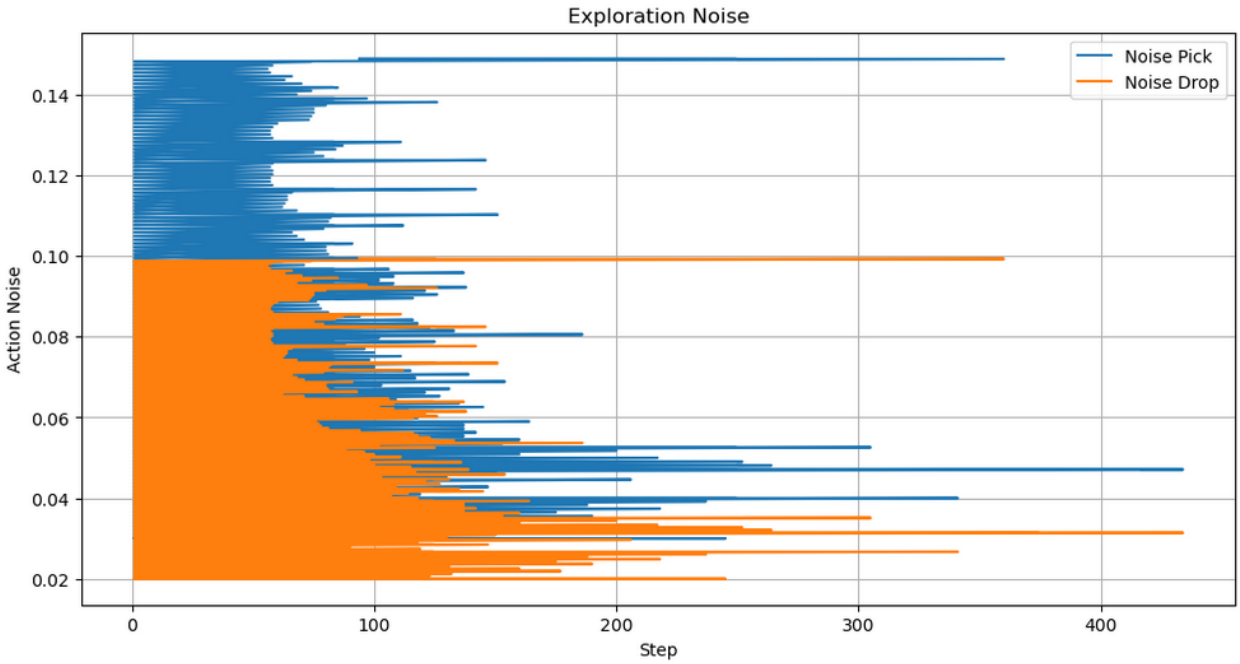

Рисунок 36 – Значение шума в действиях

На рисунке 37 видно, как обновления растут скачкообразно, отражая регулярные итерации обучения на выборках из буфера воспроизведения. На ранних этапах рост обновлений особенно интенсивен, что связано с высокой частотой корректировок параметров. По мере продвижения количество обновлений стабилизируется на уровне ~10 000, что свидетельствует о насыщении опыта и переходе модели к тонкой настройке. Неравномерные скачки указывают на задержки обновления акторов (характерные для TD3), когда критики обучаются чаще, а акторы обновляются реже (delayed policy updates).

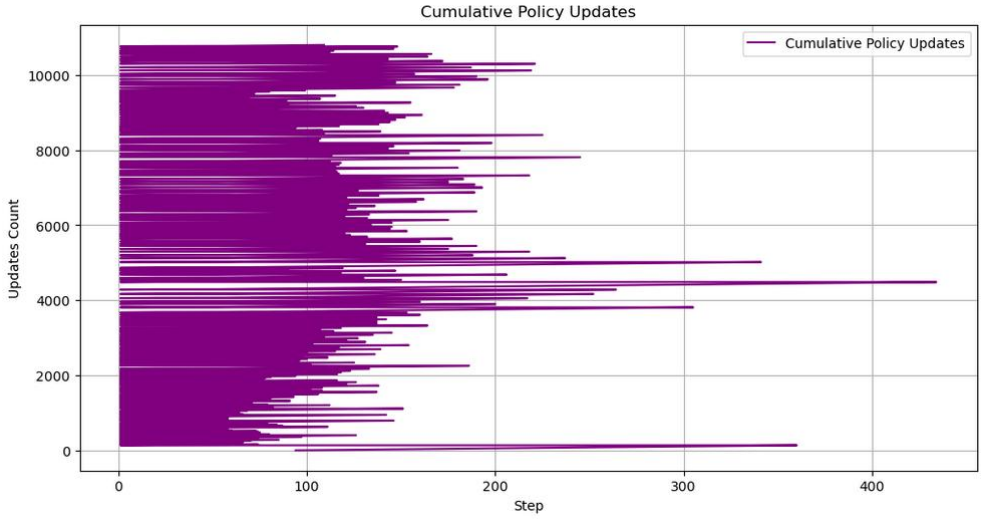

Рисунок 37 – Кол-во обновлении политики (для pick и drop)

На рисунке 38 представленный график отражает динамику компонентов функции вознаграждения в задаче pick and place. Основное внимание привлекает компонент r_tcp_drop, которая заметно доминирует над всеми остальными сигналами. Её значения достигают пиков до семи единиц, что указывает на то, что агент чаще всего взаимодействует именно с фазой сброса объекта. Однако форма кривых демонстрирует хаотичность и резкие колебания, что говорит о неустойчивости поведения и отсутствии выработанной стабильной стратегии на старте.

Остальные компоненты как r_near_cube, r_near_basket, r_progress, r_align, r_funnel - практически не проявляют себя: их значения остаются близкими к нулю. Это указывает либо на низкую значимость их весов в общей функции вознаграждения, либо на то, что агент редко достигает соответствующих состояний (нахождение у кубика, корзины, выравнивание и поступательное движение).

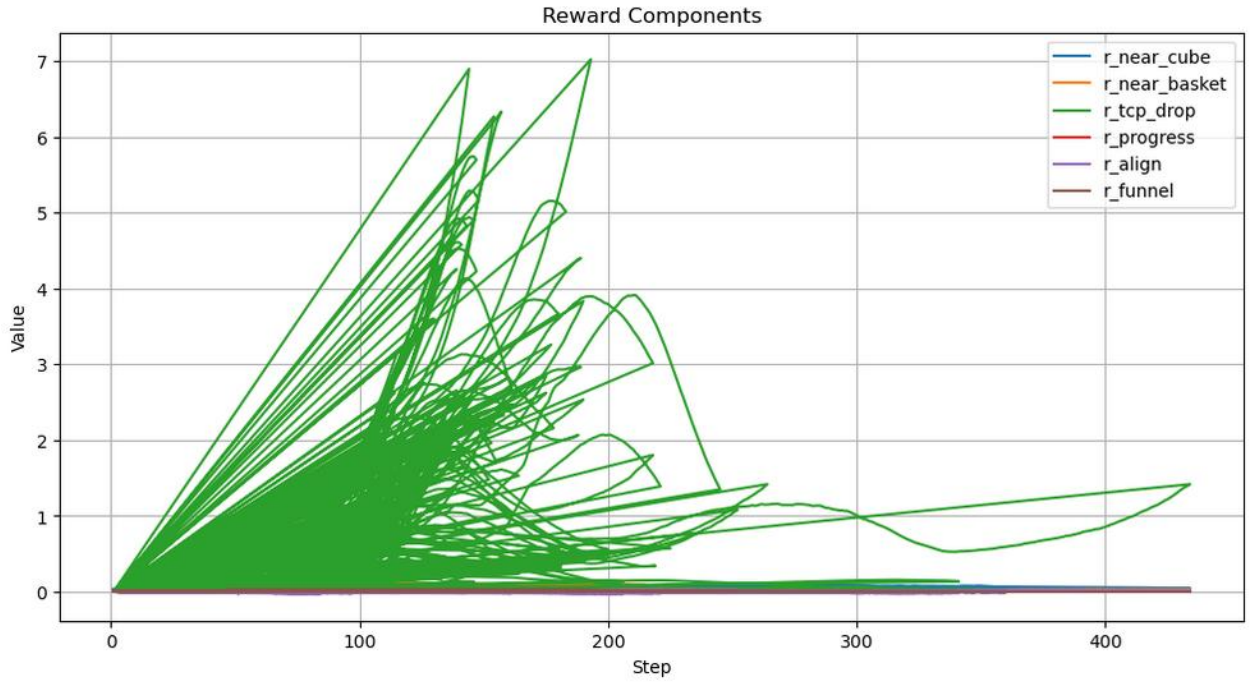

Рисунок 38 – Вклад каждой из компонентов награды в эпизоде

Одна из успешных попыток броска кубика в корзину в симуляции Webots продемонстрировано на рисунке 39.

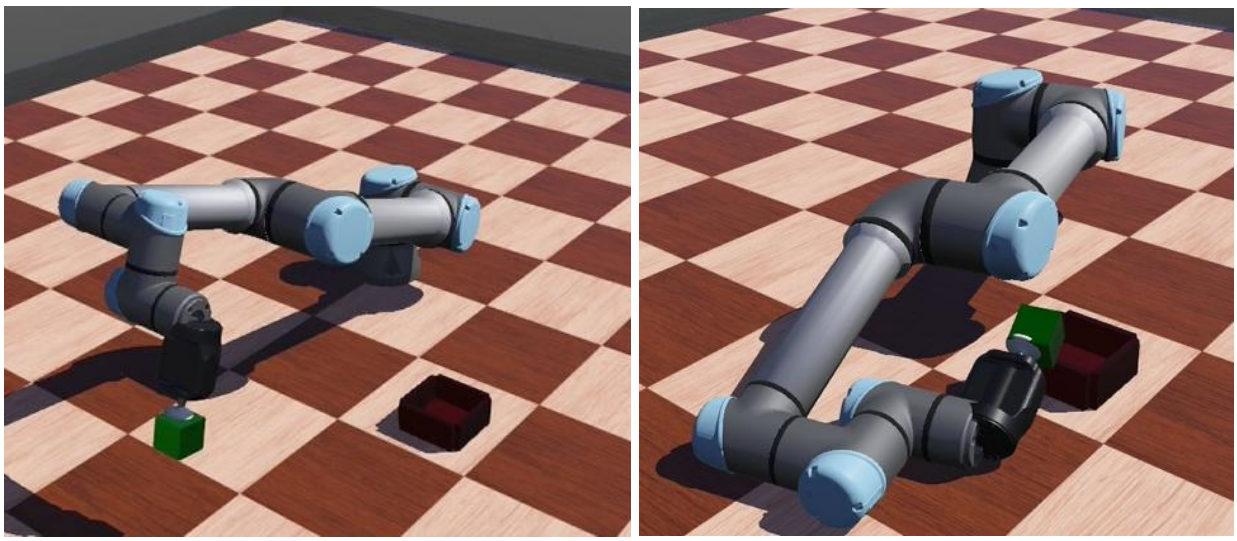

Рисунок 39 – Демонстрация Pick-and-drop в Webots

3.5 Применение в задачах сварки и маркировки

Для задач маркировки был для начала установлен 3D напечатанный макет, а затем установлен настоящий лазерный аппарат как показано на рисунке 40.

Рисунок 40 – Демонстрация ERA Cobot M5 с 3D–макетом лазерного аппарата

На изображениях 41, 42, 43 видно, как двигается робот после установки. Далее происходит сам процесс нанесения.

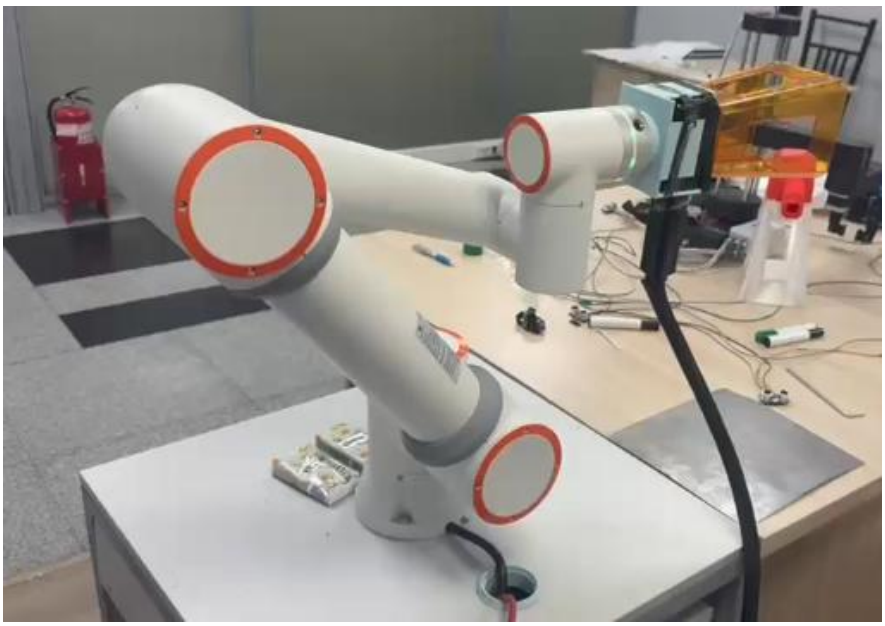

Рисунок 41 – Демонстрация ERA Cobot M5 с установленным лазерным аппаратом

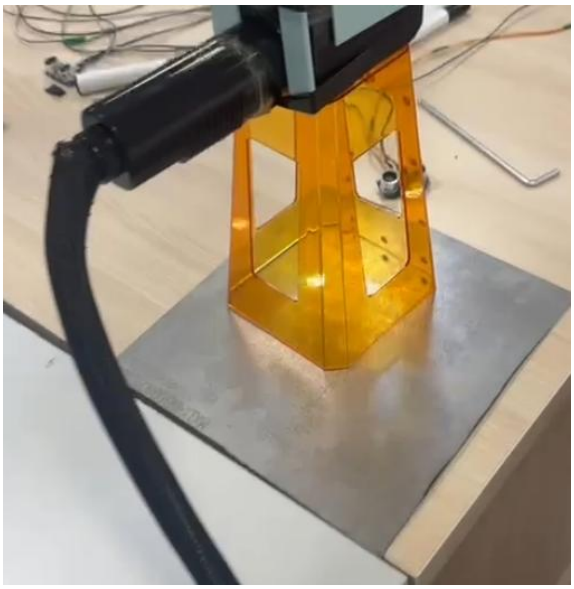

Рисунок 42 – Демонстрация процесса нанесения лазера на металлический образец

Рисунок 43 – Демонстрация процесса нанесения лазера на металлический образец (полный вид)

В целях обеспечения безопасности при проведении демонстрационных испытаний сварочный аппарат не был введён в рабочий режим и фактически не использовался. Вместо этого для имитации технологического процесса был применён маркер, закреплённый на исполнительном органе роботизированного манипулятора как показано на рисунке 44. Такой подход позволил наглядно отработать траекторию движения, проверить точность позиционирования и корректность управляющих алгоритмов без риска, связанного с высокими температурами, электрической дугой и возможным повреждением оборудования или окружающей среды.

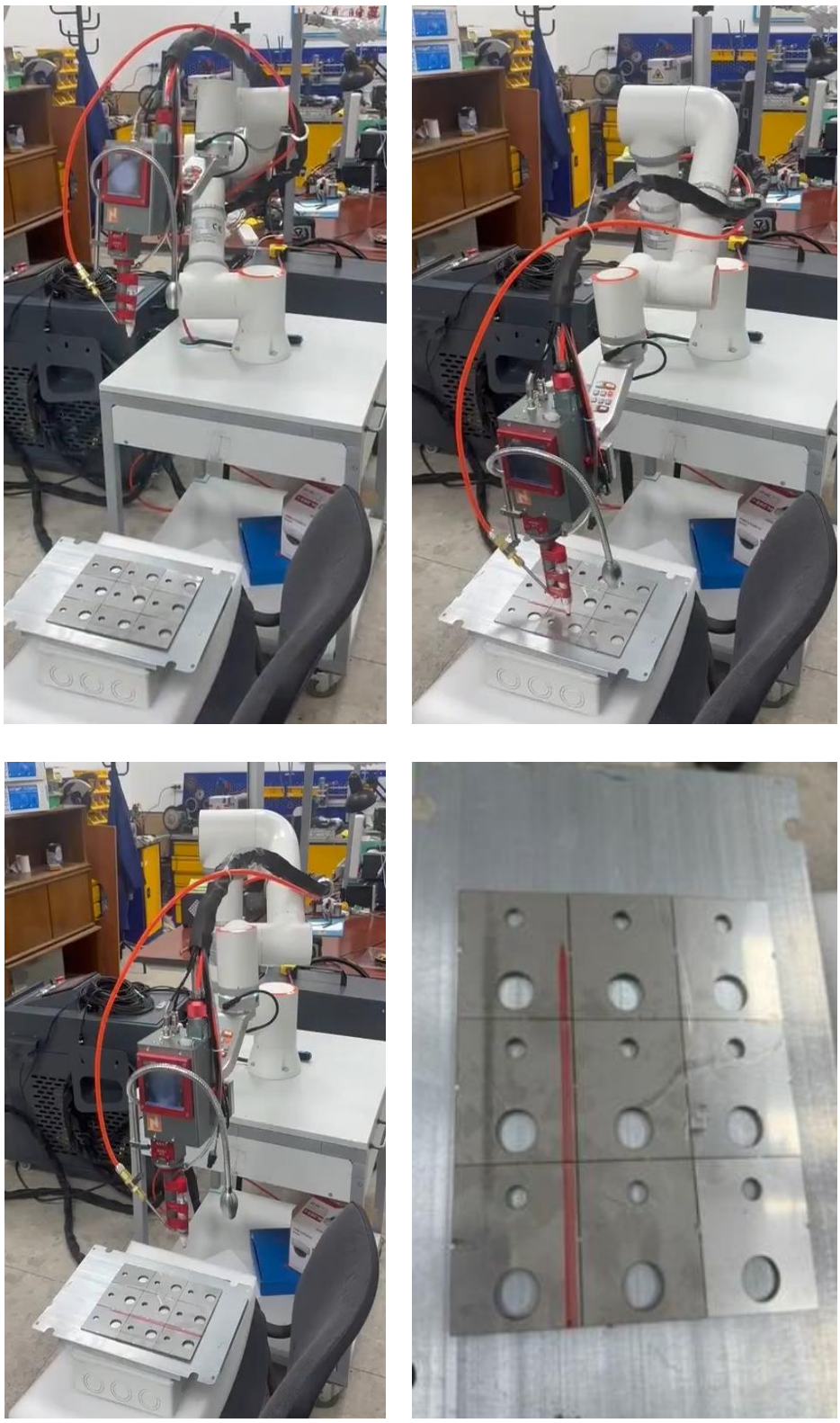

Рисунок 44 – Демонстрация движения робота с установленным сварочным аппаратом по заданной траектории и результат сварочной траектории

ЗАКЛЮЧЕНИЕ

В заключение следует отметить, что в рамках проведённого исследования были рассмотрены и проанализированы современные подходы к разработке моделей глубокого обучения с подкреплением для управления роботизированными манипуляторами в промышленных условиях. Особое внимание было уделено вопросам построения систем управления, способных адаптироваться к динамически изменяющейся среде, обеспечивать высокую точность выполнения задач и сохранять устойчивость при наличии неопределённостей и внешних возмущений. В работе была исследована эволюция алгоритмов обучения с подкреплением от дискретных методов к современным алгоритмам с непрерывным пространством действий, таким как MPC, DDPG, TD3. Установлено, что данные подходы позволяют эффективно решать задачи управления роботизированными манипуляторами, обеспечивая плавное и точное управление в многомерных пространствах состояний. Особую значимость представляет интеграция методов обучения с подкреплением с классическими подходами оптимального управления, такими как Model Predictive Control, что позволяет объединить преимущества предсказуемости и устойчивости с адаптивностью и обучаемостью. Практическая часть исследования продемонстрировала возможность применения разработанных моделей в симуляционной среде с последующим ориентированием на перенос в реальные промышленные условия. Были проанализированы ключевые аспекты построения среды обучения, включая формирование функции награды, выбор пространства состояний и действий, а также влияние параметров среды на процесс обучения агента. Полученные результаты подтверждают, что корректная постановка задачи и грамотная настройка алгоритмов играют решающую роль в достижении требуемого уровня качества управления. Несмотря на существующие ограничения, современные подходы демонстрируют значительный потенциал в решении данной задачи и открывают перспективы для дальнейших исследований.

Таким образом, проведённая работа подтверждает высокую актуальность и перспективность использования методов глубокого обучения с подкреплением для управления роботизированными манипуляторами в промышленности. Разработанные подходы позволяют повысить уровень автоматизации, улучшить точность и надёжность выполнения технологических операций, а также сократить необходимость ручного вмешательства. В дальнейшем целесообразно сосредоточить усилия на повышении устойчивости алгоритмов, снижении требований к объёму обучающих данных, а также на разработке методов, обеспечивающих гарантированную сходимость и безопасность работы в реальных условиях.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1) Bistak, P., Huba, M., Vrancic, D., & Chamraz, S. (2023). IPDT Model-Based Ziegler–Nichols Tuning Generalized to Controllers with Higher-Order Derivatives. Sensors, 23(8), 3787. https://doi.org/10.3390/s23083787

2) Raibert, M. H., & Craig, J. J. (1981). Hybrid Position/Force control of Manipulators. Journal of Dynamic Systems Measurement and Control, 103(2), 126–133. https://doi.org/10.1115/1.3139652

3) Hutchinson, S., Hager, G., & Corke, P. (1996). A tutorial on visual servo control. IEEE Transactions on Robotics and Automation, 12(5), 651–670. https://doi.org/10.1109/70.538972

4) Macenski, S., Foote, T., Gerkey, B., Lalancette, C., & Woodall, W. (2022). Robot Operating System 2: Design, architecture, and uses in the wild. Science Robotics, 7(66), eabm6074. https://doi.org/10.1126/scirobotics.abm6074

5) Kober, J., Bagnell, J. A., & Peters, J. (2013). Reinforcement learning in robotics: A survey. The International Journal of Robotics Research, 32(11), 1238–1274. https://doi.org/10.1177/0278364913495721

6) Chemweno, P., Pintelon, L., & Decre, W. (2020). Orienting safety assurance with outcomes of hazard analysis and risk assessment: A review of the ISO 15066 standard for collaborative robot systems. Safety Science, 129, 104832. https://doi.org/10.1016/j.ssci.2020.104832

7) Luh, J. (1983). An anatomy of industrial robots and their controls. IEEE Transactions on Automatic Control, 28(2), 133–153. https://doi.org/10.1109/tac.1983.1103216

8) Denavit, J., & Hartenberg, R. S. (1955). A kinematic notation for Lower-Pair mechanisms based on matrices. Journal of Applied Mechanics, 22(2), 215–221. https://doi.org/10.1115/1.4011045

9) Bobrow, J., Dubowsky, S., & Gibson, J. (1985). Time-Optimal control of robotic manipulators along specified paths. The International Journal of Robotics Research, 4(3), 3–17. https://doi.org/10.1177/027836498500400301

10) De Santis, A., Siciliano, B., De Luca, A., & Bicchi, A. (2007). An atlas of physical human–robot interaction. Mechanism and Machine Theory, 43(3), 253–270. https://doi.org/10.1016/j.mechmachtheory.2007.03.003

11) Carron, A., Arcari, E., Wermelinger, M., Hewing, L., Hutter, M., & Zeilinger, M. N. (2019). Data-Driven Model predictive control for trajectory tracking with a

robotic arm. IEEE Robotics and Automation Letters, 4(4), 3758–3765. https://doi.org/10.1109/lra.2019.2929987

12) Spong, M. (1992). On the robust control of robot manipulators. IEEE Transactions on Automatic Control, 37(11), 1782–1786. https://doi.org/10.1109/9.173151

13) Janabi-Sharifi, F., Deng, L., & Wilson, W. J. (2010). Comparison of basic visual servoing methods. IEEE/ASME Transactions on Mechatronics, 16(5), 967–983. https://doi.org/10.1109/tmech.2010.2063710

14) Kavraki, L., Svestka, P., Latombe, J., & Overmars, M. (1996). Probabilistic roadmaps for path planning in high-dimensional configuration spaces. IEEE Transactions on Robotics and Automation, 12(4), 566–580. https://doi.org/10.1109/70.508439

15) Feng, Z., Hu, G., Sun, Y., & Soon, J. (2020). An overview of collaborative robotic manipulation in multi-robot systems. Annual Reviews in Control, 49, 113–127. https://doi.org/10.1016/j.arcontrol.2020.02.002

16) Arimoto, S., Kawamura, S., & Miyazaki, F. (1984). Bettering operation of Robots by learning. Journal of Robotic Systems, 1(2), 123–140. https://doi.org/10.1002/rob.4620010203

17) Wilson, J., Charest, M., & Dubay, R. (2016b). Non-linear model predictive control schemes with application on a 2 link vertical robot manipulator. Robotics and Computer-Integrated Manufacturing, 41, 23–30. https://doi.org/10.1016/j.rcim.2016.02.003

18) Ziegler, J. G., & Nichols, N. B. (1993). Optimum settings for automatic controllers. Journal of Dynamic Systems Measurement and Control, 115(2B), 220–222. https://doi.org/10.1115/1.2899060

19) Ziegler, J. G., & Nichols, N. B. (1942). Optimum settings for automatic controllers. Transactions of the American Society of Mechanical Engineers, 64(8), 759–765. https://doi.org/10.1115/1.4019264

20) Hogan, N. (1984). Impedance control of industrial robots. Robotics and Computer-Integrated Manufacturing, 1(1), 97–113. https://doi.org/10.1016/0736-5845(84)90084-x

21) Khatib, O. (1987). A unified approach for motion and force control of robot manipulators: The operational space formulation. IEEE Journal on Robotics and Automation, 3(1), 43–53. https://doi.org/10.1109/jra.1987.1087068

22) Ortega, R., & Spong, M. W. (1989). Adaptive motion control of rigid robots: A tutorial. Automatica, 25(6), 877–888. https://doi.org/10.1016/0005-1098(89)90054-x

23) Chiaverini, S., Siciliano, B., & Villani, L. (1999b). A survey of robot interaction control schemes with experimental comparison. IEEE/ASME Transactions on Mechatronics, 4(3), 273–285. https://doi.org/10.1109/3516.789685

24) LaValle, S. M., & Kuffner, J. J. (2001). Randomized kinodynamic planning. The International Journal of Robotics Research, 20(5), 378–400. https://doi.org/10.1177/02783640122067453

25) Gerkey, B. P., & Matarić, M. J. (2004). A Formal analysis and taxonomy of task allocation in Multi-Robot Systems. The International Journal of Robotics Research, 23(9), 939–954. https://doi.org/10.1177/0278364904045564

26) Olfati-Saber, R., Fax, J. A., & Murray, R. M. (2007). Consensus and cooperation in networked Multi-Agent systems. Proceedings of the IEEE, 95(1), 215–233. https://doi.org/10.1109/jproc.2006.887293

27) Argall, B. D., Chernova, S., Veloso, M., & Browning, B. (2008). A survey of robot learning from demonstration. Robotics and Autonomous Systems, 57(5), 469–483. https://doi.org/10.1016/j.robot.2008.10.024

28) Osa, T., Pajarinen, J., Neumann, G., Bagnell, J. A., Abbeel, P., & Peters, J. (2018). An Algorithmic Perspective on Imitation Learning. Foundations and Trends in Robotics, 7(1–2), 1–179. https://doi.org/10.1561/2300000053

29) Tang, C., Abbatematteo, B., Hu, J., Chandra, R., Martín-Martín, R., & Stone, P. (2024). Deep Reinforcement Learning for Robotics: A Survey of Real-World Successes. Annual Review of Control Robotics and Autonomous Systems, 8(1), 153–188. https://doi.org/10.1146/annurev-control-030323-022510

30) Jin, L., Li, S., Yu, J., & He, J. (2018). Robot manipulator control using neural networks: A survey. Neurocomputing, 285, 23–34. https://doi.org/10.1016/j.neucom.2018.01.002

31) Karoly, A. I., Galambos, P., Kuti, J., & Rudas, I. J. (2020). Deep Learning in Robotics: Survey on model structures and training strategies. IEEE Transactions on Systems Man and Cybernetics Systems, 51(1), 266–279. https://doi.org/10.1109/tsmc.2020.3018325

32) Ratliff, N., Zucker, M., Bagnell, J. A., & Srinivasa, S. (2009). CHOMP: Gradient optimization techniques for efficient motion planning. 2009 IEEE International Conference on Robotics and Automation, 489–494. https://doi.org/10.1109/robot.2009.5152817

33) Schulman, J., Duan, Y., Ho, J., Lee, A., Awwal, I., Bradlow, H., Pan, J., Patil, S., Goldberg, K., & Abbeel, P. (2014). Motion planning with sequential convex optimization and convex collision checking. The International Journal of Robotics Research, 33(9), 1251–1270. https://doi.org/10.1177/0278364914528132

34) Sucan, I. A., Moll, M., & Kavraki, L. E. (2012). The Open Motion Planning Library. IEEE Robotics & Automation Magazine, 19(4), 72–82. https://doi.org/10.1109/mra.2012.2205651

35) Krämer, M., Rösmann, C., Hoffmann, F., & Bertram, T. (2020). Model predictive control of a collaborative manipulator considering dynamic obstacles. Optimal Control Applications and Methods, 41(4), 1211–1232. https://doi.org/10.1002/oca.2599

36) Mitsioni, I., Tajvar, P., Kragic, D., Tumova, J., & Pek, C. (2023). Safe Data-Driven Model predictive control of systems with complex dynamics. IEEE Transactions on Robotics, 39(4), 3242–3258. https://doi.org/10.1109/tro.2023.3266995

37) Mariotti, E., Magrini, E., & De Luca, A. (2019). Admittance Control for Human-Robot Interaction Using an Industrial Robot Equipped with a F/T Sensor. 2019 International Conference on Robotics and Automation (ICRA), 6130–6136. https://doi.org/10.1109/icra.2019.8793657

38) Magrini, E., Ferraguti, F., Ronga, A. J., Pini, F., De Luca, A., & Leali, F. (2019). Human-robot coexistence and interaction in open industrial cells. Robotics and Computer-Integrated Manufacturing, 61, 101846. https://doi.org/10.1016/j.rcim.2019.101846

39) Alonso-Mora, J., Baker, S., & Rus, D. (2017). Multi-robot formation control and object transport in dynamic environments via constrained optimization. The International Journal of Robotics Research, 36(9), 1000–1021. https://doi.org/10.1177/0278364917719333

40) Da Costa Barros, Í. R., & Nascimento, T. P. (2021). Robotic Mobile Fulfillment Systems: A survey on recent developments and research opportunities. Robotics and Autonomous Systems, 137, 103729. https://doi.org/10.1016/j.robot.2021.103729

41) Chan, M. L., & Hvass, P. (2017). ROS-Industrial: Expanding Industrial Application Capabilities. Journal of the Robotics Society of Japan, 35(4), 307–310. https://doi.org/10.7210/jrsj.35.307

42) Terei, N., Wiemann, R., & Raatz, A. (2024). ROS-Based control of an industrial Micro-Assembly robot. Procedia CIRP, 130, 909–914. https://doi.org/10.1016/j.procir.2024.10.184

43) Urrea, C., & Kern, J. (2025). Recent Advances and Challenges in Industrial Robotics: A Systematic Review of Technological Trends and Emerging Applications. Processes, 13(3), 832. https://doi.org/10.3390/pr13030832

44) Wu, K., Tang, Z., & Zhang, L. (2025). A study on the impact of industrial robot applications on labor resource allocation. Systems, 13(7), 569. https://doi.org/10.3390/systems13070569

45) Dompey, A. M. A., Yussif, A., Dai, A., & Zayed, T. (2026). Productivity assessment of assembly robots for prefabricated construction: trends and performance factors. Construction Innovation, 1–28. https://doi.org/10.1108/ci-06-2025-0264

46) IFR International Federation of Robotics. (n.d.). World Robotics 2025 report – INDUSTRIAL ROBOTS – released by IFR. https://ifr.org/ifr-press-releases/news/global-robot-demand-in-factories-doubles-over-10-years

47) Egamberdiev, K., Suvonov, B., Khayriddinov, S., & Juraev, D. (2026). Integration of Artificial Intelligence and Robotics in Industrial Engineering. In Next-Generation Industrial Engineering: Artificial Intelligence, Smart Manufacturing, and Sustainable Industrial Engineering (English edition). https://doi.org/10.62486/978-9915-704-21-0.ch06

48) Du, Y., Zhang, B., Zheng, C., & Tang, Y. (2026). Towards seamless and safe human-robot collaboration in Industry 5.0: advances in human behaviour prediction. International Journal of Production Research, 1–38. https://doi.org/10.1080/00207543.2026.2639732

49) Song, C., Wang, B., Li, X., Yang, H., & Wang, L. (2026). EDNPOS: An open-set skeleton-based human action recognition approach for human-robot collaboration enabled by outlier exposure. Robotics and Computer-Integrated Manufacturing, 101, 103278. https://doi.org/10.1016/j.rcim.2026.103278

50) Yuan, G., Liu, X., Xiao, M., Xiao, J., & Wang, L. (2026). Large language models in human-robot collaboration: A systematic review, trends, and challenges. Journal of Manufacturing Systems, 85, 249–268. https://doi.org/10.1016/j.jmsy.2026.01.011

51) Qin, Q., Liu, Z., Zhong, R., Wang, X. V., Wang, L., Wiktorsson, M., & Wang, W. (2025). Robot digital twin systems in manufacturing: Technologies, applications, trends and challenges. Robotics and Computer-Integrated Manufacturing, 97, 103103. https://doi.org/10.1016/j.rcim.2025.103103

52) Li, X., Gu, X., Huang, S., Wang, B., Leng, J., & Wang, L. (2026). Smart reconfigurable manufacturing towards Industry 5.0. Journal of Manufacturing Systems, 86, 178–202. https://doi.org/10.1016/j.jmsy.2026.02.023

53) Ibrahim, A., & Kumar, G. (2025). A framework for integrating Lean Six Sigma and Industry 4.0 for sustainable manufacturing. International Journal of Production Research, 1–23. https://doi.org/10.1080/00207543.2025.2571202

54) Yang, Q., Wang, X., & Wu, H. (2025). Study on lean production management of new energy vehicle body painting based on the dual perspectives of digital transformation and VSM. PLoS ONE, 20(2), e0318253. https://doi.org/10.1371/journal.pone.0318253

55) Li, Y., Song, Y., & Lee, C. (2026). Industrial robot adoption and the resilience of manufacturing global value chains. Structural Change and Economic Dynamics, 77, 93–109. https://doi.org/10.1016/j.strueco.2026.01.003

56) Raman, R., Pattnaik, D., Hughes, L., & Nedungadi, P. (2024). Unveiling the dynamics of AI applications: A review of reviews using scientometrics and BERTopic modeling. Journal of Innovation & Knowledge, 9(3), 100568. https://doi.org/10.1016/j.jik.2024.100517

57) Xu, J., Sun, Q., Han, Q. L., & Tang, Y. (2025). When Embodied AI Meets Industry 5.0: Human-Centered Smart Manufacturing. IEEE/CAA Journal of Automatica Sinica, 12(3), 485-501. https://doi.org/10.1109/JAS.2025.125327

58) Liu, H., Guo, D., & Cangelosi, A. (2025). Embodied Intelligence: A Synergy of Morphology, Action, Perception and Learning. ACM Computing Surveys, 57. https://doi.org/10.1145/3717059

59) Ren, L., Dong, J., Liu, S., Zhang, L., & Wang, L. (2024). Embodied Intelligence Toward Future Smart Manufacturing in the Era of AI Foundation Model. IEEE/ASME Transactions on Mechatronics, 30(4), 2632-2642. https://doi.org/10.1109/ 10.1109/TMECH.2024.3456250

60) Duan, J., Yu, S., Tan, H. L., Zhu, H., & Tan, C. (2022). A Survey of Embodied AI: From Simulators to Research Tasks. IEEE Transactions on Emerging Topics in Computational Intelligence, 6(2), 230-244. DOI: https://doi.org/10.1109/TETCI.2022.3141105

61) Sun, J., Mao, P., Kong, L., & Wang, J. (2025). A Review of Embodied Grasping. Sensors, 25(3), 852. DOI: https://doi.org/10.3390/s25030852

62) Ren, L., Wang, H., Dong, J., et al. (2025). Industrial foundation model. IEEE Transactions on Cybernetics, 55(5), 2286-2301. https://doi.org/10.1109/TCYB.2025.3527632

63) Wang, J., Shi, E., Hu, H., et al. (2025). Large language models for robotics: Opportunities, challenges, and perspectives. Journal of Automation and Intelligence, 4(1), 52-64. DOI: 10.1016/j.jai.2024.12.003

64) Jeong, H., Lee, H., Kim, C., & Shin, S. (2024). A Survey of Robot Intelligence with Large Language Models. Applied Sciences, 14(19), 8868. DOI: https://doi.org/10.3390/app14198868

65) Kim, Y., Kim, D., Choi, J., et al. (2024). A survey on integration of large language models with intelligent robots. Intelligent Service Robotics, 17. DOI: https://doi.org/10.1007/s11370-024-00550-5

66) Shirai, K., Beltran-Hernandez, C. C., Hamaya, M., et al. (2024). Vision-Language Interpreter for Robot Task Planning. 2024 IEEE ICRA. DOI: https://doi.org/10.1109/ICRA57147.2024.10611112

67) Liu, H., Zhu, Y., Kato, K., et al. (2024). Enhancing the LLM-Based Robot Manipulation Through Human-Robot Collaboration. IEEE Robotics and Automation Letters, 9(8), 6904-6911. DOI: https://doi.org/ 10.1109/LRA.2024.3415931

68) Wang, Z., Liu, Q., Qin, J., & Li, M. (2024). Ensuring Safety in LLM-Driven Robotics: A Cross-Layer Sequence Supervision Mechanism. 2024 IEEE/RSJ IROS. DOI: https://doi.org/10.1109/IROS58592.2024.10801576

69) Yang, Z., Raman, S. S., Shah, A., & Tellex, S. (2024). Plug in the Safety Chip: Enforcing Constraints for LLM-driven Robot Agents. 2024 IEEE ICRA. DOI: https://doi.org/10.1109/ICRA57147.2024.10611447

70) Macdonald, J. P., Mallick, R., Wollaber, A. B., et al. (2024). Language, Camera, Autonomy! Prompt-engineered Robot Control for Rapidly Evolving Deployment. Proceedings of the 2024 ACM/IEEE HRI. DOI: https://doi.org/10.1145/3610978.3640671

71) Li, S., Ma, Z., Liu, F., et al. (2025). Safe Planner: Empowering Safety Awareness in Large Pre-Trained Models for Robot Task Planning. Proceedings of the AAAI Conference on Artificial Intelligence. DOI: https://doi.org/10.1609/aaai.v39i14.33602

72) You, H., et al. (2023). Robot-Enabled Construction Assembly with Automated Sequence Planning Based on ChatGPT: RoboGPT. Buildings, 13(7). DOI: https://doi.org/10.3390/buildings13071772

73) Liu, Z., Liu, Q., Xu, W., Wang, L., & Zhou, Z. (2022). Robot learning towards smart robotic manufacturing: A review. Robotics and Computer-Integrated Manufacturing, 77, 102360. DOI: https://doi.org/10.1016/j.rcim.2022.102360

74) Li, C., Zheng, P., Yin, Y., Wang, B., & Wang, L. (2023). Deep reinforcement learning in smart manufacturing: A review and prospects. CIRP Journal of Manufacturing Science and Technology, 40. DOI: https://doi.org/10.1016/j.cirpj.2022.11.003

75) Tang, C., Abbatematteo, B., Hu, J., et al. (2025). Deep Reinforcement Learning for Robotics: A Survey of Real-World Successes. Annual Review of Control, Robotics, and Autonomous Systems, 8, 153-188. DOI: https://doi.org/10.1146/annurev-control-030323-022510

76) Alginahi, Y. M., Sabri, O., & Said, W. (2025). Reinforcement Learning for Industrial Automation: A Comprehensive Review of Adaptive Control and Decision-Making in Smart Factories. Machines, 13(12), 1140. DOI: https://doi.org/10.3390/machines13121140

77) Xie, Y. (2025). A Survey of Safe Reinforcement Learning Methods in Robotics. ITM Web of Conferences. DOI: https://doi.org/10.1051/itmconf/20257801014

78) Liu, R., et al. (2021). Deep Reinforcement Learning for the Control of Robotic Manipulation: A Focussed Mini-Review. Robotics, 10(1), 22. DOI: https://doi.org/10.3390/robotics10010022

79) Sivamayil, K., et al. (2023). A systematic study on reinforcement learning based applications. Energies, 16(3), 1512. DOI: https://doi.org/10.3390/en16031512

80) Khan, F., Feng, W., Wang, Z., & Wang, W. (2025). Safe reinforcement learning for vision-based robotic manipulation in human-centered environments. International Journal of Intelligent Robotics and Applications. DOI: https://doi.org/10.1007/s41315-025-00507-6

81) Cao, G., & Bai, J. (2025). Multi-agent deep reinforcement learning-based robotic arm assembly research. PLoS ONE, 20(2), e0311550. DOI: https://doi.org/10.1371/journal.pone.0311550

82) Al Homsi, M., Trumić, M., Fagiolini, A., & Cirrincione, G. (2025). Comparative analysis of deep Q-learning algorithms for object throwing using a robot manipulator. Frontiers in Robotics and AI. DOI: https://doi.org/10.3389/frobt.2025.1567211

83) Calderon-Cordova, C., & Sarango, R. (2023). A Deep Reinforcement Learning Algorithm for Robotic Manipulation Tasks in Simulated Environments. Engineering Proceedings, 47(1), 12. DOI: https://doi.org/10.3390/engproc2023047012

84) Bai, X., Wang, J., Xiong, X., & Boukas, E. (2025). Deep Reinforcement Learning of Robotic Manipulation for Whip Targeting. 2025 IEEE SIMPAR. DOI: https://doi.org/10.1109/SIMPAR62925.2025.10979109

85) Xu, L.-X., & Chen, Y.-Y. (2021). Deep Reinforcement Learning Algorithms for Multiple Arc-Welding Robots. Frontiers in Control Engineering, 2, 632417. DOI: https://doi.org/10.3389/fcteg.2021.632417

86) Zhang, Z., Peng, Z., Liu, W., Zhou, H., & Zhou, B. (2024). Robotic Policy Learning via Human-assisted Action Preference Optimization. Conference on Robot Learning (CoRL) 2024.

87) Wang, Y., Yu, R., Wan, S., Gan, L., & Zhan, D.-C. (2025). FOUNDER: Grounding Foundation Models in World Models for Open-Ended Embodied Decision Making. PMLR 267. DOI: https://doi.org/10.48550/arXiv.2507.12496

88) Wang, S., Wang, L., Zhou, S., et al. (2025). FlowRAM: Grounding Flow Matching Policy with Region-Aware Mamba Framework for Robotic Manipulation. CVPR 2025 Proceedings.

89) Tian, J., Wang, L., Zhou, S., et al. (2025). PDFactor: Learning Tri-Perspective View Policy Diffusion Field for Multi-Task Robotic Manipulation. CVPR 2025 Proceedings.

90) Black, K., Galliker, M. Y., & Levine, S. (2025). Real-Time Execution of Action Chunking Flow Policies. NeurIPS 2025.

91) Liu, B., Zhu, Y., Gao, C., et al. (2023). LIBERO: Benchmarking Knowledge Transfer for Lifelong Robot Learning. NeurIPS 2023. DOI: https://doi.org/10.48550/arXiv.2306.03310

92) Tamizi, M. G., Yaghoubi, M., & Najjaran, H. (2023). A Review of Recent Trend in Motion Planning of Industrial Robots. Int. J. of Intelligent Robotics and Applications, 7(2). DOI: https://doi.org/10.1007/s41315-023-00274-2

93) Xu, L., & Zhang, W. (2025). Survey on Path Planning Based on Deep Reinforcement Learning. Proceedings of Machine Learning Research, 278, 685-695.

94) Han, X., Li, S., Wang, X., & Zhou, W. (2021). Semantic Mapping for Mobile Robots in Indoor Scenes: A Survey. Information, 12(2), 92. DOI: https://doi.org/10.3390/info12020092

95) Barros, A. M., et al. (2022). A Comprehensive Survey of Visual SLAM Algorithms. Robotics, 11(1), 24. DOI: https://doi.org/10.3390/robotics11010024

96) Shi, Y., Zhang, G., & Kong, M. (2024). Path planning of welding robot based on deep learning. Paladyn, 15, 20240002. DOI: https://doi.org/10.1515/pjbr-2024-0002

97) Wang, X., Fu, H., Deng, G., et al. (2023). Hierarchical Free Gait Motion Planning for Hexapod Robots Using Deep Reinforcement Learning. IEEE Transactions on Industrial Informatics, 19(11). DOI: https://doi.org/10.1109/TII.2023.3240758

98) Zhang, Y., & Chen, P. (2023). Path Planning of a Mobile Robot for a Dynamic Indoor Environment Based on an SAC-LSTM Algorithm. Sensors, 23(24), 9802. DOI: https://doi.org/10.3390/s23249802

99) Chai, R., Niu, H., Carrasco, J., et al. (2024). Design and Experimental Validation of Deep Reinforcement Learning-Based Fast Trajectory Planning and Control for Mobile Robot in Unknown Environment. IEEE Transactions on Neural Networks and Learning Systems, 35(4). DOI: https://doi.org/10.1109/TNNLS.2022.3209154

100) Zhong, J., Wang, T., & Cheng, L. (2022). Collision-Free Path Planning for Welding Manipulator via Hybrid Algorithm of Deep Reinforcement Learning and Inverse Kinematics. Complex & Intelligent Systems, 8(3), 1899-1912. DOI: https://doi.org/10.1007/s40747-021-00366-1

101) Hu, J., Wang, F., & Li, X. (2024). Trajectory Tracking Control for Robotic Manipulator Based on Soft Actor–Critic and Generative Adversarial Imitation Learning. Biomimetics, 9(12), 779. DOI: https://doi.org/10.3390/biomimetics9120779

102) Wang, Z., Chen, Z., & Li, H. (2025). A Heterogeneous Time-Series Soft Actor–Critic Method for Quadruped Locomotion. Drones, 9(8), 569. DOI: https://doi.org/10.3390/drones9080569

103) Liu, Y., et al. (2025). Obstacle avoidance planning for body in white spot welding robot based on deep reinforcement learning. Proceedings of SPIE, 13953, 1395314. DOI: https://doi.org/10.1117/12.3084388

104) Ren, Z., et al. (2024). Research on obstacle avoidance path planning of robotic manipulator based on deep reinforcement learning. Proceedings of the 4th International Conference on Advanced Algorithms and Neural Networks, 46. DOI: https://doi.org/10.1117/12.3049584

105) Fan, F., et al. (2024). Spatiotemporal path tracking via deep reinforcement learning of robot for manufacturing internal logistics. Journal of Manufacturing Systems, 74, 112-125. DOI: https://doi.org/10.1016/j.jmsy.2024.01.005

106) Qin, L. (2026). Recent progress and challenges of key technologies in robotic assembly. Chinese Journal of Mechanical Engineering, 39, 100032. DOI: https://doi.org/10.1016/j.cjme.2025.100032

107) Zhang, H., et al. (2022). Deep Learning-Based Robot Vision: High-End Tools for Smart Manufacturing. IEEE Instrumentation & Measurement Magazine, 25(3), 23-30. DOI: https://doi.org/10.1109/MIM.2022.9756392

108) Lee, R. K.-J., Zheng, H., & Lu, Y. (2024). Human-Robot Shared Assembly Taxonomy: A step toward seamless human-robot knowledge transfer. Robotics and

Computer-Integrated Manufacturing, 86, 102618. DOI: https://doi.org/10.1016/j.rcim.2023.102686

109) Brohan, A., et al. (2023). RT-1: Robotics Transformer for Real-World Control at Scale. Robotics: Science and Systems. DOI: https://doi.org/10.15607/RSS.2023.XIX.025

110) Kim, M. J., et al. (2025). OpenVLA: An Open-Source Vision-Language-Action Model. PMLR (CoRL), 270, 2679-2713.

111) Chi, C., et al. (2024). Diffusion policy: Visuomotor policy learning via action diffusion. The International Journal of Robotics Research. DOI: https://doi.org/10.1177/02783649241273668

112) Lin, X., et al. (2024). SpawnNet: Learning Generalizable Visuomotor Skills from Pre-trained Network. 2024 IEEE ICRA. DOI: https://doi.org/10.1109/ICRA57147.2024.10610356

113) Iqdymat, A., & Stamatescu, G. (2025). Reinforcement Learning of a Six-DOF Industrial Manipulator for Pick-and-Place Application Using Efficient Control in Warehouse Management. Sustainability, 17(2), 432. DOI: https://doi.org/10.3390/su17020432

114) Honelign, L., & Tullu, A. (2025). Implementation of the deep deterministic policy gradient algorithm in a simulated environment to improve the target-reaching performance of a robotic arm. Actuators, 14(4), 165. DOI: https://doi.org/10.3390/act14040165

115) Ferreira, H. C., & Barbosa, R. S. (2025). Deep Reinforcement Learning for Adaptive Robotic Grasping and Post-Grasp Manipulation in Simulated Dynamic Environments. Future Internet, 17(10), 437. DOI: https://doi.org/10.3390/fi17100437

116) Zhang, C., et al. (2024). Heterogeneous knowledge graph-driven subassembly identification with ensemble deep learning in Industry 4.0. International Journal of Production Research, 62(15), 5432-5450. DOI: https://doi.org/10.1080/00207543.2024.2430450

117) Lee, R. K.-J. (2025). A robotic peg-in-hole assembly method based on visual-tactile fusion and reinforcement learning. Industrial Robot, 52(1), 1-14. DOI: https://doi.org/10.1108/IR-11-2024-0526

118) Tsang, Y.P., et al. (2025). A deep reinforcement learning approach for online and concurrent operations optimization. Computers in Industry, 164, 104202. DOI: https://doi.org/10.1016/j.compind.2024.104202

119) Ma, H., Li, Z., & Wu, X. (2024). Research on 3C compliant assembly strategy method of manipulator based on deep reinforcement learning. Computers and Electrical Engineering, 119, 109605. DOI: https://doi.org/10.1016/j.compeleceng.2024.109605

120) Zhang, S., Liang, S., & Jiang, Z. (2025). Research on robotic peg-in-hole assembly method based on variable admittance. Applied Sciences, 15(4), 2143. DOI: https://doi.org/10.3390/app15042143

121) Hickman, X., & Prince, D. (2025). Hybrid safe reinforcement learning: tackling distribution shift and outliers with the Student-t's process. Neurocomputing, 634, 129912. DOI: https://doi.org/10.1016/j.neucom.2024.129912

122) Beltran-Hernandez, C. C., et al. (2020). Variable compliance control for robotic peg-in-hole assembly: A deep-reinforcement-learning approach. Applied Sciences, 10(19), 6923. DOI: https://doi.org/10.3390/app10196923

123) Li, M., Huang, J., Xue, L., & Zhang, R. (2023). A guidance system for robotic welding based on an improved YOLOv5 algorithm with a RealSense depth camera. Scientific Reports, 13, 21299. DOI: https://doi.org/10.1038/s41598-023-21299-1

124) Chen, X., & Cai, Y. (2020). Vision-Based Robotic Object Grasping—A Deep Reinforcement Learning Approach. Sensors, 20(5), 1555. DOI: https://doi.org/10.3390/machines11020275

125) Li, Y., et al. (2025). Visuo-tactile feedback policies for terminal assembly facilitated by reinforcement learning. Frontiers in Robotics and AI, 12. DOI: https://doi.org/10.3389/frobt.2025.1660244

126) Ali, H., et al. (2025). Deep Reinforcement Learning Based Robotic Arm Control. International Journal of Advanced Computer Science and Applications, 16(6). DOI: https://doi.org/10.14569/IJACSA.2025.0160666

127) Arshad, M., & Bazzocchi, M. (2024). Collision Avoidance and Disturbance Minimization Through Deep Reinforcement Learning Control of a Free-Floating Space Manipulator. 22nd IAA Symposium on Space Debris, 1881-1887. DOI: https://doi.org/10.52202/078360-0181

128) Chen, Y., & Jinfeng, C., (2024). Research on Applications of Deep Learning in Robotic Automation for Assembly Lines. 2024 6th International Academic Exchange Conference on Science and Technology Innovation, 1-5. DOI: 10.1109/IAECST64597.2024.11118082

129) Shao, Y., et al. (2023). A Control Method of Robotic Arm Based on Improved Deep Deterministic Policy Gradient. 2023 International Conference on Mechatronics Technology and Intelligent Manufacturing, 1-6. DOI: https://doi.org/10.1109/ICMA57826.2023.10215662

130) Deng, W., et al. (2024). Physics informed machine learning model for inverse dynamics in robotic manipulators. Applied Soft Computing, 163, 111877. DOI: https://doi.org/10.1016/j.asoc.2024.111877

131) Dharmawan, A. G., et al. (2020). A Model-Based Reinforcement Learning and Correction Framework for Process Control of Robotic Wire Arc Additive Manufacturing. 2020 IEEE ICRA, 1-7. DOI: https://doi.org/10.1109/ICRA40945.2020.9197222

132) Shan, S. (2024). Fine robotic manipulation without force/torque sensor. IEEE Robotics and Automation Letters, 9(2), 1206-1213. DOI: https://doi.org/10.1109/LRA.2023.3341770

133) Zhou, H., et al. (2021). A hybrid control strategy for grinding and polishing robot based on adaptive impedance control. Advances in Mechanical Engineering, 13(4). DOI: https://doi.org/10.1177/16878140211004034

134) Jiang, M., et al. (2023). A novel fine-grained assembly sequence planning method based on knowledge graph and deep reinforcement learning. Journal of Manufacturing Systems, 68, 269-282. DOI: https://doi.org/10.1016/j.jmsy.2024.08.001

135) Li, J., et al. (2022). A flexible manufacturing assembly system with deep reinforcement learning. Control Engineering Practice, 124, 105151. DOI: https://doi.org/10.1016/j.conengprac.2021.104957

136) Solowjow, E., et al. (2020). Industrial Robot Grasping with Deep Learning using a Programmable Logic Controller (PLC). 2020 IEEE 16th CASE, 1-6. DOI: https://doi.org/10.1109/CASE48305.2020.9216744

137) Valori, M., et al. (2021). Validating Safety in Human–Robot Collaboration: Standards and New Perspectives. Robotics, 10(2), 65. DOI: https://doi.org/10.3390/robotics10020065

138) Yuan, G., Liu, X., Xiao, M., Xiao, J., & Wang, L. (2026). Large language models in human-robot collaboration: A systematic review, trends, and challenges. Journal of Manufacturing Systems, 85, 249-268. DOI: https://doi.org/10.1016/j.jmsy.2026.01.011

139) Salvato, E., Fenu, G., Medvet, E., & Pellegrino, F. A. (2021). Crossing the reality gap: A survey on sim-to-real transferability of robot controllers in reinforcement learning. IEEE Access, 9. DOI: https://doi.org/10.1109/access.2021.3126658

140) Du, Y., Zhang, B., Zheng, C., & Tang, Y. (2026). Towards seamless and safe human-robot collaboration in Industry 5.0: advances in human behaviour prediction.

International Journal of Production Research. DOI: https://doi.org/10.1080/00207543.2026.2639732

141) Song, C., Wang, B., Li, X., et al. (2026). EDNPOS: An open-set skeleton-based human action recognition approach for human-robot collaboration enabled by outlier exposure. Robotics and Computer-Integrated Manufacturing, 101, 103278. DOI: https://doi.org/10.1016/j.rcim.2026.103278

142) Thananjeyan, B., Balakrishna, A., Nair, S., et al. (2021). Recovery RL: Safe Reinforcement Learning With Learned Recovery Zones. IEEE Robotics and Automation Letters. DOI: https://doi.org/10.1109/LRA.2021.3070252

143) Thumm, J., & Althoff, M. (2022). Provably Safe Deep Reinforcement Learning for Robotic Manipulation in Human Environments. 2022 IEEE ICRA. DOI: https://doi.org/10.1109/ICRA46639.2022.9811698

144) Namasivayam, K., et al. (2024). Learning to Recover from Plan Execution Errors during Robot Manipulation: A Neuro-symbolic Approach. 2024 IEEE/RSJ IROS. DOI: https://doi.org/10.1109/IROS58592.2024.10801831

145) Zhou, X., et al. (2024). Reset-Free Reinforcement Learning via Multi-State Recovery and Failure Prevention for Autonomous Robots. Tsinghua Science and Technology. DOI: https://doi.org/10.26599/TST.2023.9010117

146) Bi, Z., Miao, Z., & Zhang, W. (2021). Safety assurance mechanisms of collaborative robotic systems in manufacturing. Robotics and Computer-Integrated Manufacturing, 67, 102022. DOI: https://doi.org/10.1016/j.rcim.2020.102022

147) Oliff, H., et al. (2020). Reinforcement learning for facilitating human-robot-interaction in manufacturing. Journal of Manufacturing Systems, 56, 326-340. DOI: https://doi.org/10.1016/j.jmsy.2020.06.018

148) Q. Liu, Z. Liu, B. Xiong, W. Xu, and Y. Liu, “Deep reinforcement learning-based safe interaction for industrial human-robot collaboration using intrinsic reward function,” Advanced Engineering Informatics, vol. 49, p. 101360, Jul. 2021, doi: 10.1016/j.aei.2021.101360.

149) C. Li, P. Zheng, Y. Yin, Y. M. Pang, and S. Huo, “An AR-assisted Deep Reinforcement Learning-based approach towards mutual-cognitive safe human-robot interaction,” Robotics and Computer-Integrated Manufacturing, vol. 80, p. 102471, Oct. 2022, doi: 10.1016/j.rcim.2022.102471.

150) S. Gu, A. Kshirsagar, Y. Du, G. Chen, J. Peters, and A. Knoll, “A human-centered safe robot reinforcement learning framework with interactive behaviors,” Frontiers in Neurorobotics, vol. 17, p. 1280341, Nov. 2023, doi: 10.3389/fnbot.2023.1280341.