

# МАШИНОСТРОЕНИЕ И МАШИНОВЕДЕНИЕ

DOI: 10.34031/2071-7318-2024-9-11-102-115

*\*Дююн И.А., Кабальянц П.С., Дююн Т.А., Рыбак Л.А.**Белгородский государственный технологический университет им. В.Г. Шухова**\*E-mail: duyun77@mail.ru*

## ИНТЕЛЛЕКТУАЛЬНАЯ СИСТЕМА УПРАВЛЕНИЯ ПЛАТФОРМОЙ СТЮАРТА НА ОСНОВЕ ПРИМЕНЕНИЯ МЕТОДА ОБУЧЕНИЯ С ПОДКРЕПЛЕНИЕМ

**Аннотация.** В статье представлена интеллектуальная система управления платформой Стюарта, являющейся механизмом параллельной кинематики с шестью степенями свободы. Основным недостатком таких механизмов является наличие особых положений рабочей области, в которых может произойти потеря устойчивости. Стандартные алгоритмы управления часто оказываются неэффективными при прохождении через потенциально нестабильные зоны, поэтому в качестве основного компонента предлагаемой системы управления выбран метод обучения с подкреплением (Reinforcement Learning), относящийся к виду машинного обучения. Выбор обусловлен эффективностью работы метода в непрерывных пространствах действий, что особенно важно для системы управления, обеспечивающей плавное движение в пределах рабочей области с переменными координатами. В условиях действия эксплуатационных нагрузок, вибраций и температурных колебаний необходима коррекция в реальном времени. Применение искусственной нейронной сети повысит точность и гибкость для адаптации к меняющимся условиям эксплуатации без снижения производительности на основе моделирования сложных нелинейных зависимостей и обучения с использованием накопленного опыта. Представлены следующие аспекты системы управления: схема работы и общая архитектура реализации метода обучения с подкреплением, схема алгоритма обучения, UML-диаграмма классов среды и агента, архитектура сети Актоар и Критика, процесс обучения сети и результаты ее тестирования, которые показали высокую эффективность ее применения.

**Ключевые слова:** платформа Стюарта, обучение с подкреплением, нейронные сети, параллельная кинематика, интеллектуальная система управления, симуляция в PyBullet

**Введение.** Современные производственные отрасли и другие сферы активно внедряют механизмы, основанные на параллельной кинематике. В таких механизмах выходное звено приводится в движение посредством нескольких кинематических цепей, что позволяет суммировать их степени свободы. В отличие от машин с последовательной кинематикой, такие устройства обеспечивают более высокую точность и грузоподъемность [1-5]. Благодаря параллельной кинематике эти механизмы имеют ряд эксплуатационных преимуществ: высокая жесткость конструкции при небольшом весе, простота сборки и масштабирования, хорошие динамические характеристики, высокая скорость и ускорение [6-10].

Механизмы параллельной кинематики используют в станкостроении, аэрокосмической области, в качестве координатно-измерительных машин, манипуляторов и подвижных тренажеров. Роботы-станки позволяют осуществлять высокоточную обработку деталей сложной геометрии, синхронную пятиосевую обработку, а также высокоскоростное фрезерование твердых материалов, обеспечивая более высокую производительность по сравнению с традиционными станками. Координатно-измерительные машины с параллельной кинематикой широко применяются

для точных измерений сложных пространственных форм. Конструкция таких машин представляет собой перевернутую пирамиду с основанием из сферических шарниров, которые служат точками отсчета для системы лазерных интерферометров, благодаря чему достигается высокая точность измерений. В аэрокосмической отрасли механизмы параллельной кинематики являются перспективным решением задач высокоточного позиционирования антенн в телескопах нового поколения. Роботизированные манипуляторы, использующие параллельную кинематику, находят широкое применение в погрузочно-разгрузочных работах, автоматизированных складах и сборочных линиях, особенно в производстве микропроцессорной техники. Подвижные тренажеры, построенные на основе параллельной кинематики, становятся все более популярными для обучения пилотов и водителей. Их главное преимущество перед неподвижными тренажерами заключается в возможности моделирования динамики реального движения [1-3].

Платформа Гью-Стюарта представляет собой современный этап развития механизмов параллельной структуры и относится к классу гексаподов, что предполагает наличие шести степеней свободы. Выходное звено (подвижная плат-

форма) перемещается относительно неподвижного основания по шести независимым координатам с помощью приводов линейных перемещений, движение обеспечивается шестью телескопически соединенными звеньями. Платформа способна перемещаться и вращаться вокруг трех координатных осей, что и обеспечивает ей шесть степеней свободы. Типовая конструкция платформы Стюарта с шестью степенями свободы представлена на рис. 1.

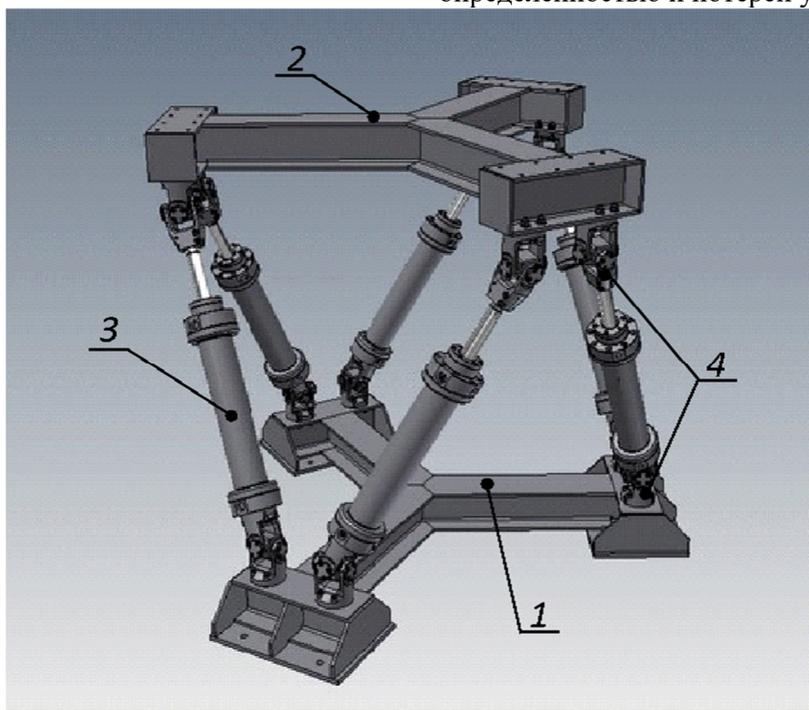


Рис. 1. Конструкция платформы Гью-Стюарта, основные механические компоненты: 1 – основание; 2 – подвижная платформа; 3 – электроцилиндры; 4 – шарниры

Планирование движения параллельного манипулятора предполагает решение прямой задачи кинематики, которая заключается в определении координат рабочего органа на основе длин выдвижения штоков, и обратной задачи кинематики — вычисления длин штоков по заданным координатам рабочего органа. Для решения этих задач традиционно применяются аналитические и численные методы. Однако в ситуациях с высокими требованиями к точности и быстродействию эти методы могут оказаться недостаточно эффективными. Под действием рабочей нагрузки, вибраций и температурных колебаний точность позиционирования может существенно снижаться. В таких условиях необходима коррекция в реальном времени, что требует от системы управления не только высокой точности, но и достаточной гибкости для адаптации к меняющимся условиям эксплуатации без снижения производительности [5, 6].

Однако, несмотря на многочисленные преимущества, механизмы параллельной кинематики имеют определенные недостатки. Нелинейные зависимости кинематики и динамики от положения выходного звена в рабочей области создают анизотропию и неоднородность в динамических, упругих и скоростных характеристиках. Кроме того, существует риск интерференции, то есть пересечения кинематических цепей, что может обуславливать возникновение особых положений, характеризующихся геометрической неопределенностью и потерей устойчивости [3, 4].

Традиционные методы управления параллельными манипуляторами, как правило, опираются на классическую теорию управления и принципы аналитической механики, применяются линейные и нелинейные алгоритмы программирования, а обратная связь реализуется с помощью пропорционально-интегрально-дифференциальных (ПИД) регуляторов. Основное преимущество такого подхода заключается в его надежности и предсказуемости, он эффективен для выполнения производственных задач, требующих точного воспроизведения повторяющихся траекторий и движений в детерминированных условиях. Однако, в динамически изменяющихся или неопределенных условиях управление становится сложнее, так как необходимо учитывать множество внешних и внутренних факторов. Поэтому традиционные системы управления не всегда могут справиться с задачей поддержания устойчивости механизма в сложных эксплуата-

ционных условиях. Это связано с их неспособностью компенсировать динамические и кинематические нагрузки, что негативно сказывается на эффективности и надежности работы, особенно в роботизированных системах, где требуется высокая точность позиционирования. В таких случаях целесообразно использовать искусственные нейронные сети, способные моделировать сложные нелинейные зависимости и обучаться на основе накопленного опыта, что повышает точность и адаптивность системы управления [11, 12].

Одна из наиболее сложных задач — это управление манипулятором в зонах сингулярности. Стандартные алгоритмы управления часто оказываются неэффективными при прохождении через потенциально нестабильные области, что может привести к потере точности и снижению надежности работы. Применение искусственного интеллекта и машинного обучения для анализа данных с датчиков в реальном времени и адаптивной коррекции траектории движения позволяет обходить зоны сингулярности, значительно повышая эффективность и надежность отработки заданной траектории.

Рассмотрим современные подходы к решению проблем управления механизмами параллельной кинематики. В статье Мерле [13] предложен алгоритм для решения прямой задачи кинематики параллельного манипулятора Гофа с использованием интервального анализа. Этот метод рассматривает интервальные неопределенности как целостные объекты, что позволяет применять к ним арифметические и аналитические операции. Однако в условиях реального времени этот метод может оказаться слишком ресурсоемким, особенно по сравнению с нейронными сетями, которые способны быстрее и адаптивнее обрабатывать данные и учитывать неопределенности. Даш [14] в своей работе использует метод Денавита-Хартенберга и векторный анализ для решения обратной задачи кинематики манипулятора. Этот подход эффективен в условиях предсказуемой среды, но имеет ограничения в условиях динамически изменяющихся параметров, где учет множества стохастических факторов становится проблематичным. Нелинейности системы усложняют кинематический анализ и увеличивают время вычислений, что требует значительных усилий для точной настройки. В исследовании Виейры и соавторов [15] рассматривается метод планирования движения параллельных манипуляторов с использованием моделирования Монте-Карло. Авторы предлагают модель, в которой нейронные сети компенсируют вычислительные ограничения метода Монте-Карло,

прогнозируя вероятность отказа механизма в различных точках рабочей области. Однако данный подход тестировался только на манипуляторе 3RRR с тремя степенями свободы и не апробирован для более сложных механизмов, таких как платформа Стюарта. В работе предложено адаптивное нейронное управление параллельным манипулятором аэрокосмического применения. Система управления на основе нейронных сетей обеспечивает высокую точность позиционирования и ориентации манипулятора в условиях работы в открытом космосе.

Таким образом, внедрение элементов искусственного интеллекта и машинного обучения в систему управления параллельными манипуляторами является перспективным направлением, так как позволяет существенно повысить точность позиционирования, оптимизировать управление в реальном времени и обеспечить самообучение системы, это особенно важно в сферах, где ошибки могут иметь серьезные последствия.

**Основная часть.** В качестве основного компонента предлагаемой системы управления выбран метод обучения с подкреплением (Reinforcement Learning), который использован для создания и обучения нейронной сети, управляющей платформой Стюарта. Выбор DDPG (Deep Deterministic Policy Gradient) обусловлен его эффективностью в непрерывных пространствах действий, что особенно важно для разработки системы управления, обеспечивающей плавное движение платформы в пределах рабочей области с переменными координатами. Этот подход является одним из видов машинного обучения и основывается на взаимодействии двух основных компонентов: агента и среды (рис. 2) [16–18].

В качестве компонента среды выступает симуляционная модель платформы Стюарта, реализованная в PyBullet, который выбран в связи с тем, что он позволяет моделировать кинематику и динамику при движении платформы, предоставляя агенту реалистичную среду для обучения и тестирования.

Агент (нейронная сеть) принимает решения на основе текущего состояния среды (платформы), изменяя длины актуаторов для достижения заданного положения в рабочем пространстве. Агент взаимодействует со средой посредством двух элементов: состояния и действия. Состояние представляет собой текущее координатное положение платформы относительно рабочей области и выражается вектором, который включает три координаты по трем координатным осям и три угла Эйлера. Агент использует это состояние для принятия решений о следующих действиях.

Действие представляет собой изменение длины актуаторов платформы, которое в свою очередь приводит к изменению положения и ориентации платформы. Таким образом агент обучается выбирать такие изменения длин актуаторов, которые минимизируют отклонение от заданной траектории движения.

Элементы состояния и действия взаимодействуют между собой посредством двух других

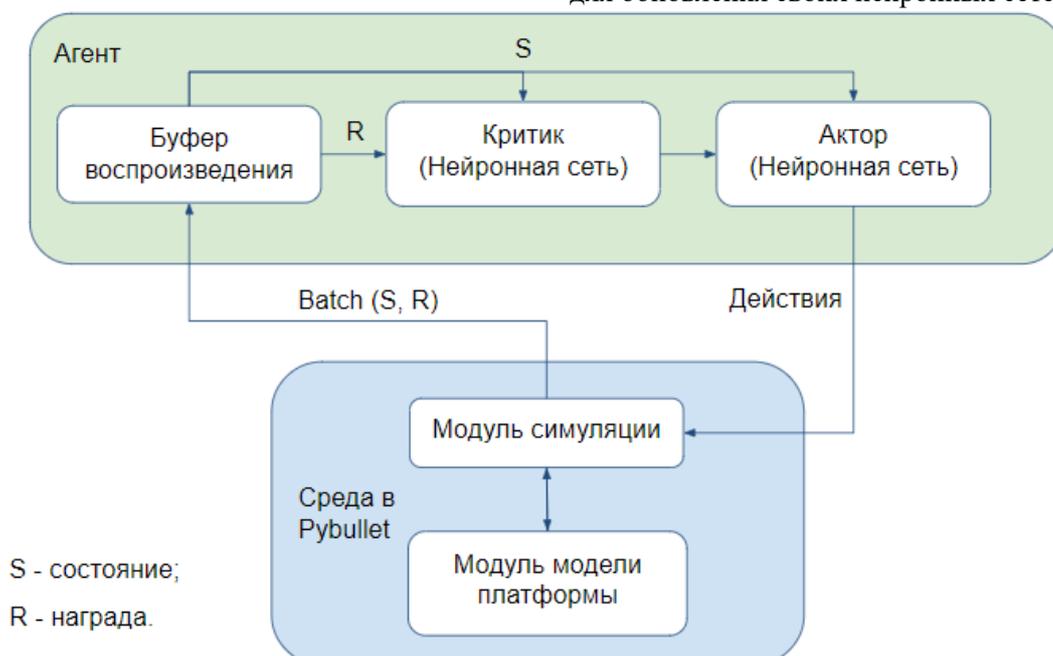
элементов – вознаграждения и штрафов. Вознаграждение является ключевым элементом процесса обучения. Агент получает вознаграждения за точное достижение целевых позиций, а также штрафы за ошибки или отклонения. Вознаграждение рассчитывается на основе отклонения текущего состояния платформы от целевого состояния, с учетом точности позиционирования.



Рис. 2. Схема работы метода обучения с подкреплением

На рис. 3 представлена архитектура взаимодействия агента и среды. Агент состоит из следующих трех ключевых компонентов: Буфер воспроизведения, Критик и Актор. В качестве Критика и Актора выступают две отдельные нейронные сети. Критик (нейронная сеть № 1) оценивает полезность действий агента, определяя, насколько хорошо данное действие способствует достижению цели. Актор (нейронная сеть № 2) определяет наилучшие действия на основе текущего состояния.

Буфер воспроизведения играет важную роль в процессе обучения агента. Он хранит историю взаимодействий агента со средой, включая состояния, действия, вознаграждения и новые состояния. Под состоянием здесь понимается определенное координатное положение платформы в пространстве рабочей области. Это позволяет агенту использовать прошлый опыт для обучения, обеспечивая более стабильное и эффективное обучение. Буфер воспроизведения используется для мини-батчей, которые агент использует для обновления своих нейронных сетей.



S - состояние;  
R - награда.

Рис. 3. Общая архитектура реализации метода обучения с подкреплением

Среда, реализованная в PyBullet, состоит из двух основных модулей: модуля симуляции и модуля модели платформы. Модуль симуляции отвечает за имитационное моделирование взаимодействий агента с окружающей средой и виртуально реализует кинематику и динамику платформы. Модуль модели платформы представляет собой детализированную модель платформы Стюарта, описывающую её точные конструктивные характеристики: геометрические размеры всех элементов (неподвижного основания, подвижной платформы, актуаторов, штоков, шарниров), физико-механические параметры материалов. Модель платформы используется модулем симуляции для вычисления ответов системы

на действия агента, обеспечивая точное моделирование поведения платформы.

На рис. 4 представлена блок-схема алгоритма DDPG, которая отображает этапы взаимодействия агента со средой и процесс обновления моделей. Процесс обучения начинается с инициализации всех компонентов: создание симулятора платформы Стюарта на основе PyBullet, формирование моделей нейронных сетей для Актора и Критика, а также их целевых версий, инициализация буфера воспроизведения для хранения опыта агента. Этот подготовительный этап необходим для корректной работы всех компонентов в процессе обучения.

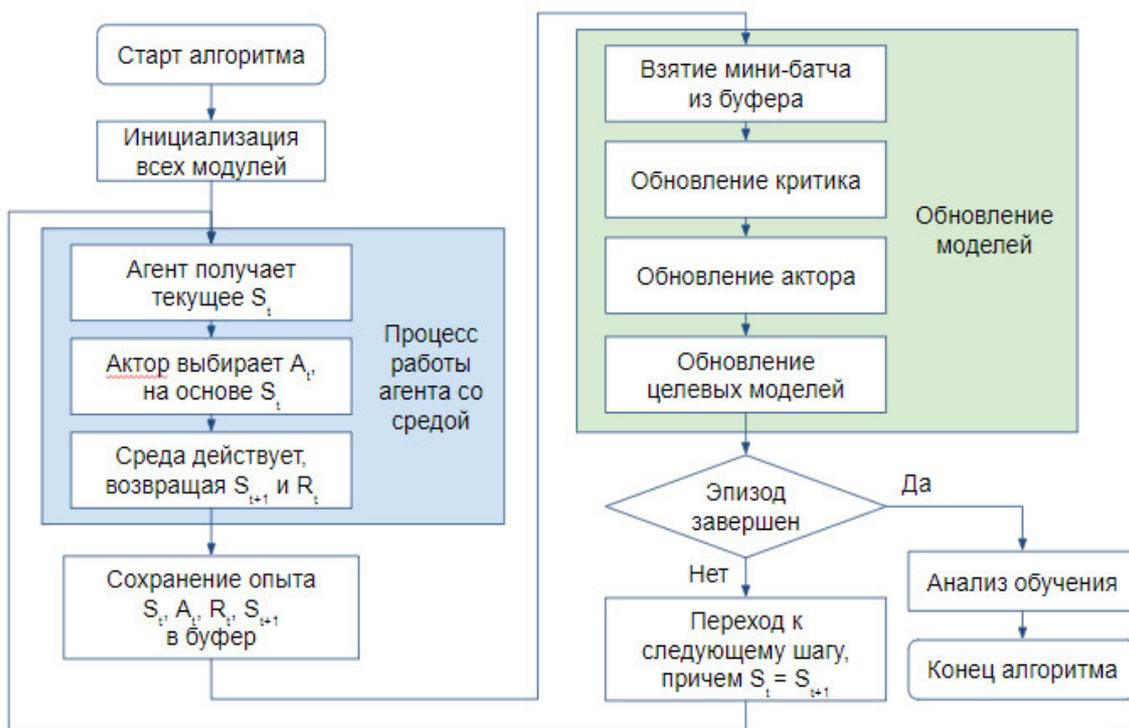


Рис. 4. Схема алгоритма обучения DDPG

Обучение состоит из множества эпизодов, каждый из которых включает два ключевых этапа: выбор начального и целевого состояний. В начале каждого эпизода агент выбирает случайное начальное положение платформы и задает фиксированную целевую позицию, которую платформа должна достичь. Далее агент взаимодействует со средой: Актор анализирует текущее состояние платформы и предсказывает действия, необходимые для достижения цели. Эти действия применяются к платформе, и симулятор обновляет её состояние. Взаимодействие агента с окружающей средой начинается с получения текущего состояния ( $S_t$ ), что включает координатное положение платформы и информацию о длинах выдвижения актуаторов. Затем Актор выбирает действие ( $A_t$ ) на основе текущего состояния, определяя, как изменять длины актуаторов плат-

формы. После выполнения действия симуляционная среда PyBullet обновляет состояние платформы, возвращая агенту новое состояние ( $S_{t+1}$ ) и вознаграждение ( $R_t$ ), которое оценивает эффективность действия. Следующий важный шаг — сохранение опыта. Данные ( $S_t$ ,  $A_t$ ,  $R_t$ ,  $S_{t+1}$ ) сохраняются в буфер воспроизведения, обеспечивая базу для последующего обучения. Затем происходит процесс обновления моделей Критика и Актора, который начинается после выполнения нескольких взаимодействий агента со средой. Этот процесс включает последовательные этапы: выбор мини-батча из буфера, обновление Критика, обновление Актора и синхронизацию целевых моделей.

На рис. 5 и 6 представлены классы, реализующие функционал среды и агента соответственно в виде UML-диаграммы, отражающей атрибуты и методы каждого класса.

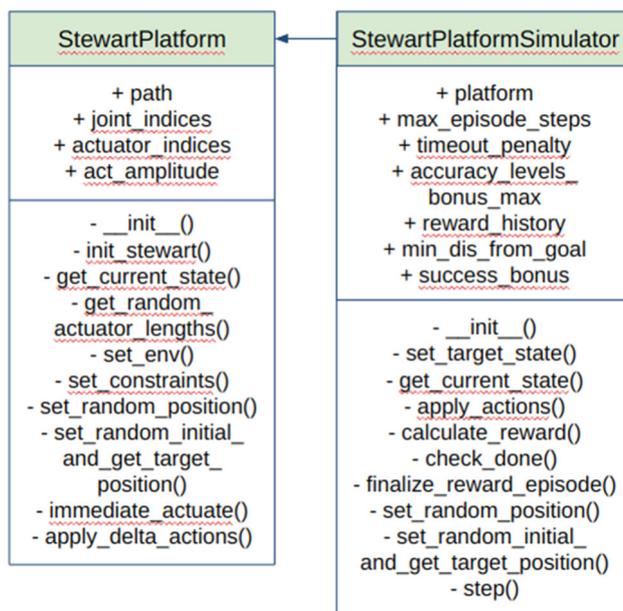


Рис. 5. UML-диаграмма классов среды

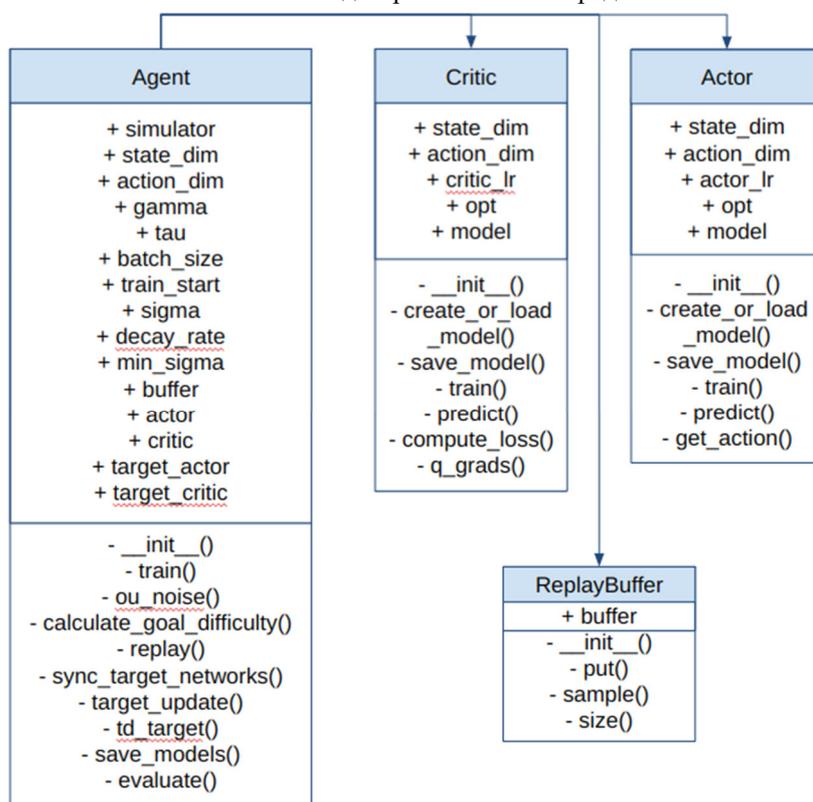


Рис. 6. UML-диаграмма классов агента

Рассмотрим подробнее архитектуру нейронных сетей, реализующих Актора и Критика в алгоритме DDPG, что позволит представить внутренние связи и процессы, происходящие во время обучения.

Актор (рис. 7) принимает на вход вектор состояния, который проходит через несколько слоев, обеспечивающих обработку информации: входной слой, первый полносвязный слой, второй полносвязный слой, выходной слой, Lambda слой. Входной слой принимает вектор состояния размерностью state\_dim. Первый полносвязный

слой содержит 400 нейронов с активацией ReLU, этот слой преобразует входные данные и выделяет из них важные признаки. Второй полносвязный слой содержит 300 нейронов с активацией ReLU, продолжает обработку данных, выделяя более сложные признаки. Выходной слой содержит action\_dim нейронов с активацией Tanh, выдавая действия. Lambda слой масштабирует выходные значения для получения допустимого диапазона действий, применяя функцию scaled\_delta.

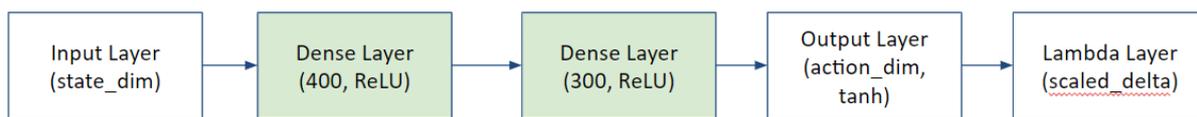


Рис. 7. Архитектура сети: Актор

Критик (рис. 8) оценивает качество действий, выдаваемых Актором, на основе текущего состояния, его архитектура имеет следующие слои: входные слои, первый полносвязный слой для состояния, второй полносвязный слой для состояния, полносвязный слой для действий, конкатенация слоев, полносвязный слой после конкатенации, выходной слой. Входные слои имеют два входа: вектор состояния размерностью state\_dim и вектор действий размерностью

action\_dim. Первый полносвязный слой для состояния содержит 400 нейронов, а второй полносвязный слой для состояния содержит 300 нейронов, оба имеют активацию ReLU. Полносвязный слой для действий содержит 32 нейрона и также имеет активацию ReLU. Конкатенация слоев объединяет обработанные состояния и действия. Полносвязный слой после конкатенации содержит 16 нейронов с активацией ReLU. Выходной слой содержит один нейрон с линейной активацией, представляющий оценку Q-value.

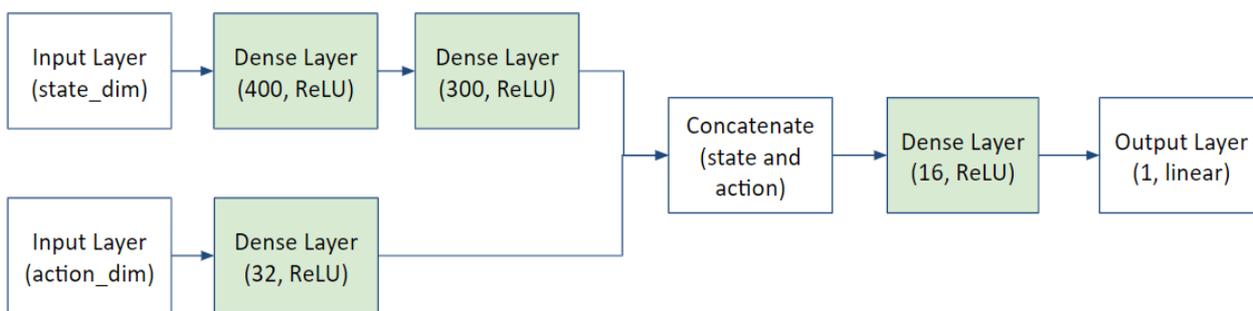


Рис. 8. Архитектура сети: Критик

**Результаты.** Рассмотрим результаты обучения нейронной сети, основанные на 500 эпизодах. На графике рис. 9 отображена средняя награда за каждый эпизод. Награда показывает общую эффективность действий агента, обучающегося достигать заданные координатные положения платформы. График демонстрирует рост

средней награды по мере увеличения количества эпизодов, что свидетельствует о прогрессе в обучении. Периоды колебаний награды указывают на адаптацию агента к новым условиям кинематического движения.

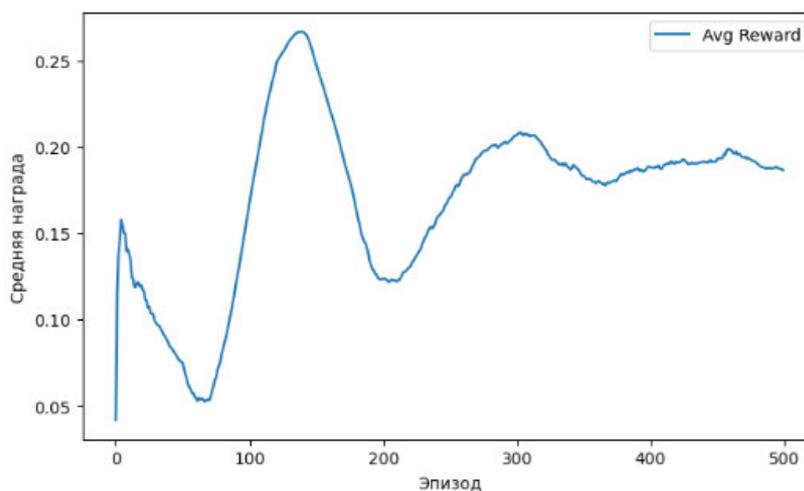


Рис. 9. Процесс обучения сети – средняя награда за эпизод

График рис. 10 показывает средний бонус за точность в пределах одного эпизода. Бонусы за точность выдаются, когда агент достигает требуемого положения платформы с высокой точностью. Рост среднего бонуса по мере обучения

указывает на то, что агент научился более точно достигать целевых позиций, улучшая свою производительность.

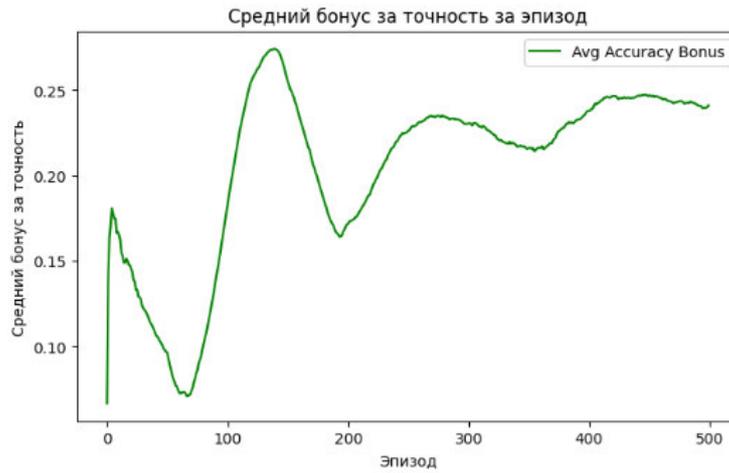


Рис. 10. Процесс обучения сети – средний бонус за точность за эпизод

График рис. 11 иллюстрирует количество шагов, необходимых для достижения целевого положения. Снижение количества шагов свидетельствует об улучшении эффективности агента. Агент учится обеспечивать требуемое целевое положение платформы быстрее, что является показателем успешного обучения и оптимизации

действий. Диаграмма рис. 12 показывает распределение успешных эпизодов по 25-эпизодным интервалам. Успешные эпизоды – это те, в которых агент достиг цели до истечения максимального количества шагов. Видно, что успешные эпизоды чаще встречаются ближе к концу обучения, что подтверждает улучшение модели.

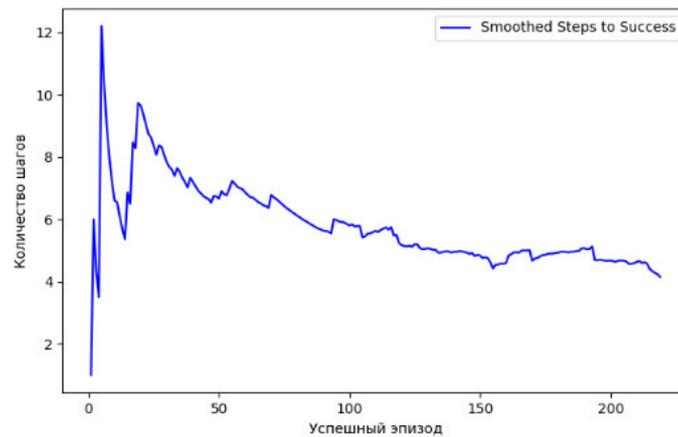


Рис. 11. Процесс обучения сети – количество шагов для достижения целевого положения по эпизодам

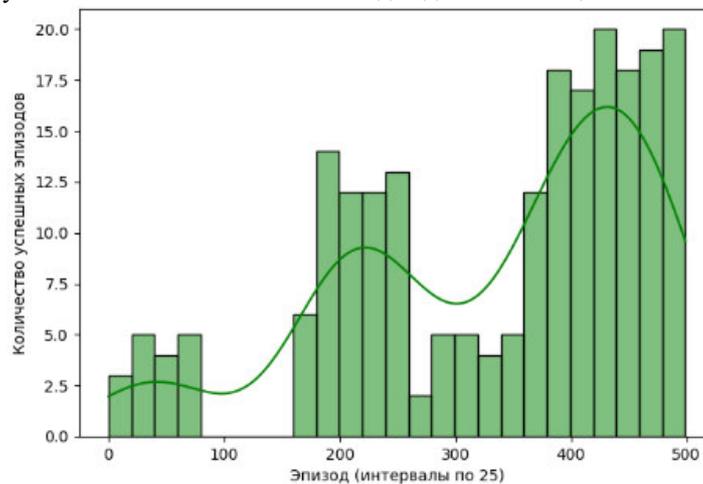


Рис. 12. Процесс обучения сети – распределение успешных эпизодов в интервалах по 25 эпизодов

В процессе обучения каждого эпизода записывался лог, включающий следующие ключевые данные: награды за каждый шаг и общая награда

за эпизод, бонусы за точность, если они были применены, штрафы за отклонения от целевых

значений, общее количество шагов, использованных для достижения цели, конечное и начальное положение платформы и её ориентация, выбор действий агентом на каждом шаге. Эти данные сохранялись для последующего анализа, что позволяло оценивать и корректировать параметры модели, а также следить за её прогрессом. Благодаря этому подробному логированию можно было отслеживать каждое изменение и его влияние на процесс обучения.

После завершения процесса обучения нейронной сети было выполнено тестирование обученной модели. Эксперимент тестирования

заключался в проверке способности нейронной сети достигать заданных целевых положений платформы Стюарта в координатном пространстве рабочей области симуляционной среды PyBullet. Было проведено 100 тестовых эпизодов, из которых 91 завершился успешно, что соответствует 91 % успеха. Ключевые метрики процесса представлены на рис. 13. Эти показатели демонстрируют высокую эффективность модели в выполнении поставленных задач. Модель достигала цели в среднем за 5.95 шагов, что свидетельствует о её высокой точности и скорости.

Успешные эпизоды	91/100	Средняя награда	8.77
Процент успеха	91.00%	Средний бонус за точность	5.74
Макс. количество шагов	20	Средний модуль действия	0.24
Среднее количество шагов	5.95	Средняя сложность цели	0.38

Рис. 13. Основные метрики процесса тестирования

На графиках рис. 14 показаны начальные положения и ориентации платформы. Зеленые звездочки обозначают целевые позиции, синие точки – успешные эпизоды, а красные крестики –

неудачные эпизоды. Видно, что модель успешно справляется с разнообразными начальными условиями, демонстрируя свою адаптивность и надежность.

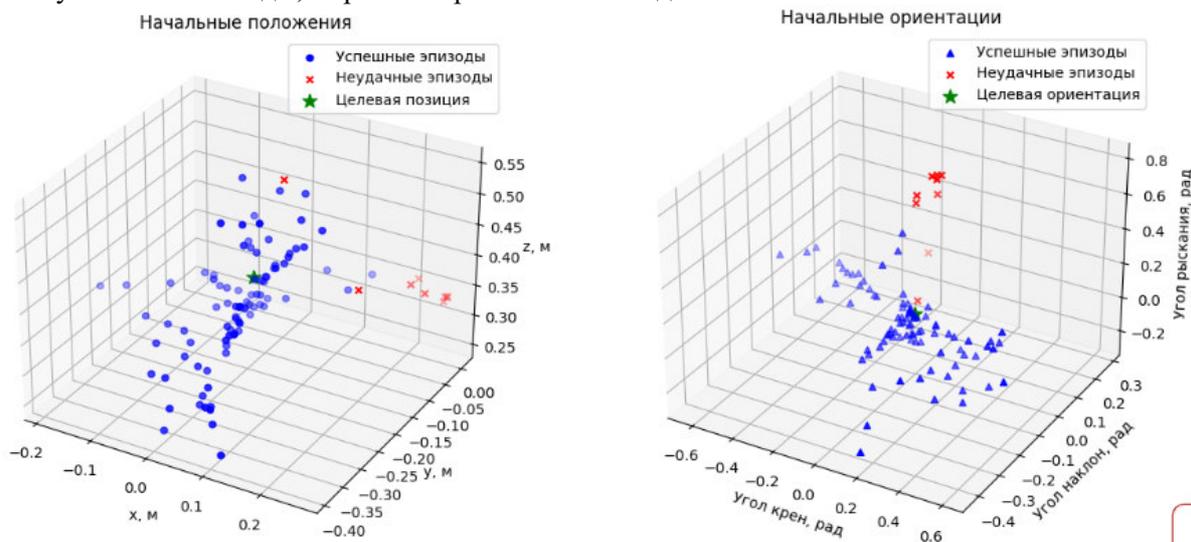


Рис. 14. Результаты тестирования нейронной сети

Для реализации процесса тестирования в программном коде использован метод evaluate, который выполняет следующие основные задачи:

1. Инициализация – определение начальных и целевых состояний платформы для каждого эпизода.
2. Запуск симуляции – запуск эпизода с заданным начальным состоянием и фиксированной целевой позицией и ориентацией.
3. Применение действий – Актор (нейронная сеть) предсказывает действия для текущего состояния, и эти действия применяются к платформе.

4. Обновление состояния – платформа переходит в новое состояние, и алгоритм оценивает награду за этот переход.

5. Запись результатов – результаты каждого шага и эпизода записываются для последующего анализа, включая награды, бонусы за точность, штрафы за отклонения и количество шагов.

6. Анализ успеха – проверка условий успешного завершения эпизода (достижение целевой позиции) или неудачи (превышение максимального количества шагов).

Метод логирует все ключевые метрики, что позволяет анализировать эффективность модели по каждому эпизоду. В результате были полу-

чены данные, подтверждающие высокую точность и скорость работы нейронной сети в различных условиях.

На рис. 15 представлен график зависимости количества шагов от сложности положения. Этот

график позволяет оценить, каким образом сложность начального состояния влияет на количество шагов, необходимых для достижения целевой позиции.



Рис. 15. Зависимость количества шагов от сложности положения

В эксперименте использованы 100 тестовых эпизодов. Успешность эпизода оценивалась по количеству необходимых шагов до достижения цели. Допустимым считалось количество шагов для достижения цели менее 20, в этом случае эпизод считался успешным. Если платформа не достигала цели за 20 шагов, то эпизод считался провальным. На графике синими точками обозначены успешные эпизоды, а красными – неудачные. Синие горизонтальные линии и красные горизонтальные линии представляют среднее количество шагов для успешных и неудачных эпизодов соответственно. Из графика видно, что среднее количество шагов для успешных эпизодов составляет 4.56 шага, что подтверждает высокую эффективность и точность модели.

Для оценки сложности начального положения платформы применялся метод `calculate_goal_difficulty`. При реализации метода используются два основных параметра: начальное состояние платформы и заданное целевое положение в координатной области. Оба этих параметра представляют собой одномерные массивы, содержащие позиции и ориентации платформы.

Под позицией здесь понимается координата рабочей области, а под ориентацией значение углов Эйлера. Сначала рассчитывается разница в позициях (векторное расстояние между начальными и целевыми координатами) и разница в ориентациях (векторное расстояние между начальными и целевыми углами ориентации). Затем эти две величины складываются для получения итогового значения сложности положения. Таким образом, сложность положения рассчитывается как сумма евклидовых расстояний между начальными и целевыми позициями и начальными и целевыми ориентациями платформы. Итоговое значение сложности положения характеризует, насколько далеко начальное координатное положение платформы находится от требуемого целевого положения в пространстве координатной области.

Как видно из графика, модель демонстрирует высокую стабильность и способность достигать цели даже при увеличении сложности начального состояния. Таким образом, результаты тестирования показывают, что разработанная нейронная сеть способна эффективно управлять платформой Стюарта, достигая требуемого

координатного положения за минимальное количество шагов, независимо от сложности начального состояния.

На рис. 16 показано для примера начальное и конечное положение платформы Стюарта в процессе выполнения тестирования. Начальное

положение является отправной точкой для инициализации процесса управления с использованием нейронной сети. Платформа находится в произвольном случайном положении, которое используется для начальных условий в процессе тестирования.

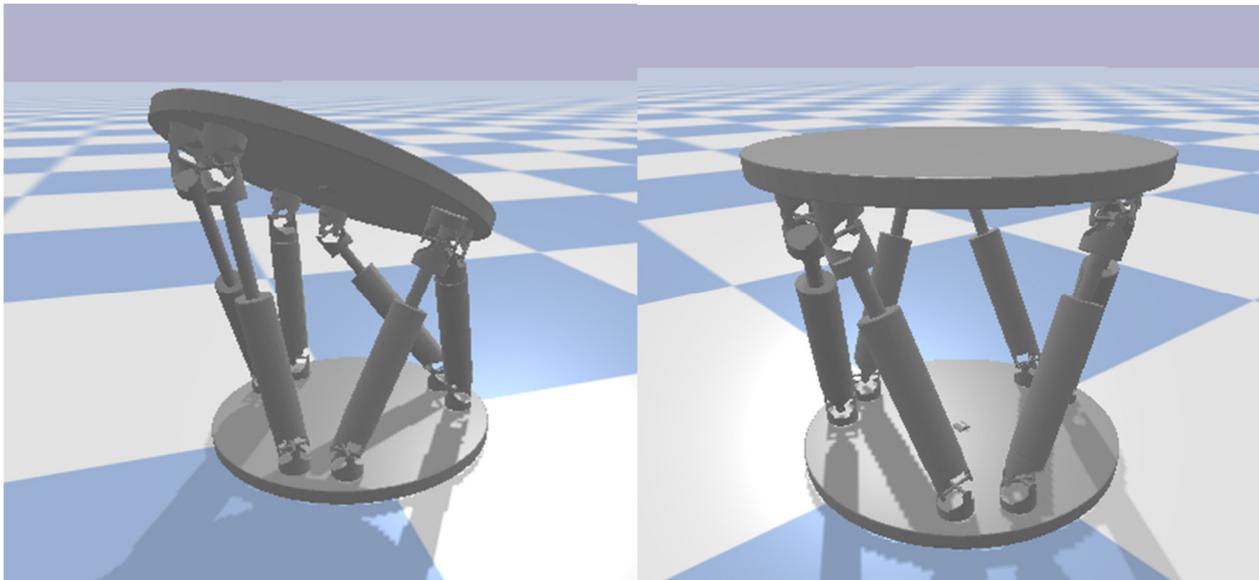


Рис. 16. Начальное и конечное положения платформы в процессе тестирования

**Выводы.** Предложенная интеллектуальная система управления платформой Стюарта расширяет ее функциональные возможности и обеспечивает высокую точность воспроизведения заданных траекторий движения. Интегрированный в систему управления алгоритм глубокого обучения с подкреплением (DRL) позволил системе адаптироваться к изменяющимся условиям и оптимизировать траекторию движения исполнительного элемента в реальном времени. Результаты тестирования модели показали, что предложенная система управления способна эффективно и точно управлять платформой Стюарта, достигая целевых позиций за минимальное количество шагов. Модель продемонстрировала высокую адаптивность и устойчивость к изменяющимся начальным условиям, что подтверждает перспективность использования методов машинного обучения в управлении механизмами параллельной кинематики.

**Источник финансирования.** Работа выполнена в рамках реализации федеральной программы поддержки университетов «Приоритет 2030» с использованием оборудования на базе Центра высоких технологий БГТУ им. В. Г. Шухова.

#### БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Дуюн И.А., Корнилов А.В., Дуюн Т.А. Применение интерфейса Python-Adams для моделирования работы платформы Гью-Стюарта // Вестник БГТУ им. В.Г. Шухова. 2022. № 10. С.

96–106. DOI: 10.34031/2071-7318-2022-7-10-96-106

2. Дуюн И.А., Дуюн Т.А., Кабелянц П.С. Моделирование особых положений роботизированной платформы и определение параметров ее управляемости // СТИН. 2023. № 5. С. 2–6.

3. Дуюн И.А., Кабелянц П.С., Зуев С.В., Дуюн Т.А., Малышев Д.И. Определение индекса управляемости и выявление особых положений параллельного манипулятора // Вестник БГТУ им. В.Г. Шухова. 2023. № 5. С. 116–125. DOI: 10.34031/2071-7318-2023-8-5-116-125

4. Дуюн И.А., Горлов А.С., Дуюн Т.А. Совместное моделирование движения параллельного манипулятора с использованием Adams-Matlab // Вестник БГТУ им. В.Г. Шухова. 2022. № 11. С. 108–119. DOI: 10.34031/2071-7318-2022-7-11-108-119.

5. Duyun T.A., Duyun I.A., Rybak L.A., Perevuznik V.V. Simulation of the Structural and Force Parameters of a Robotic Platform Using Co-Simulation // Procedia Computer Science. 2022. Vol. 213. Pp. 720–727. DOI: 10.1016/j.procs.2022.11.126

6. Duyun I.A., Duyun T.A., Kabalyants P.S., Rybak L.A. Optimization of a 6-DOF Platform for Simulators Based on an Analysis of Structural and Force Parameters // Machines. 2023. Vol. 11. Pp. 1–18. DOI: 10.3390/machines11080814

7. Леонов Г.А., Зегжда С.А., Зуев С.М. Динамика платформы Стюарта и управление ее движением // Доклады академии наук. 2014. Т. 458. № 1. С. 36–41. DOI: 10.7868/S0869565214250094

8. Ribeiro P.R., Andre M., Philipp K., Karel K. A VR Truck Docking Simulator Platform for Developing Personalized Driver Assistance // *Applied Sciences*. 2021. 11(19). 8911. DOI: 10.3390/app11198911
9. Vieira H.L., de Carvalho Fontes J.V., Beck A.T., da Silva M.M. Robust Critical Inverse Condition Number for a 3RRR Robot Using Failure Maps // Springer International Publishing. 2018. Pp. 285–294. DOI: 10.1007/978-3-319-67567-1\_27
10. Dasmahapatra S., Ghosh M. Workspace Identification of Stewart Platform // *International Journal of Engineering and Advanced Technology*. 2020. Vol. 9. Pp. 1903–1907. DOI: 10.35940/ijeat.C5338.029320
11. Peng Y., Dai H., Zhang H., Yue X. Dynamics and control of a bio-inspired Stewart platform // *Xibeigongye Daxue Xuebao / Journal of Northwestern Polytechnic University*. 2021. Vol. 39. Pp. 258–266. DOI: 10.1051/jnwpu/20213920258
12. Karakas B., Senay B. Kinematics of Supination and Pronation with Stewart Platform // *J. Math. Sci. Model*. 2021. Vol. 4. Pp. 1–6. DOI: 10.33187/jmsm.815125
13. Merlet J.P. Jacobian, Manipulability, Condition Number, and Accuracy of Parallel Robots // *Journal of Mechanical Design*. 2006. Vol. 128. Pp. 199–206. DOI: 10.1115/1.2121740
14. Dash K.K., Choudhury B.B., Senapati S.K. Inverse Kinematics Solution of a 6-DOF Industrial Robot // *Advances in Intelligent Systems and Computing*. 2019. Vol. 758. Pp. 183–192. DOI: 10.1007/978-981-13-0514-6\_19
15. Vieira H.L., da Silva M.M., Beck A.T. Reliable Motion Planning for Parallel Manipulators // *Mechanism and Machine Theory*. 2019. Vol. 140. Pp. 553–566. DOI: 10.1016/j.mechmachtheory.2019.06.022
16. Zhukov Y.A., Korotkov E.B., Moroz A.V., Zhukova V.V., Abramov A.M. Adaptive Neural Network Control of Hexapod for Aerospace Application // *IOP Conference Series: Materials Science and Engineering*. 2018. Vol. 441. 012001. DOI: 10.1088/1757-899X/441/1/012001
17. Valentino K., Kevin K., Singgih J., End D. Dynamic Models Based Virtual Reality Flight Simulator // *Internetworking Indones. J.* 2022. Vol. 11. Pp. 23–28.
18. Li S., Jin L., Mirza M. Neural Network Based Stewart Platform Control // *Wiley-IEEE Press*. 2019. Vol. 7. Pp. 105–129. DOI: 10.1002/9781119557005.ch7

*Информация об авторах*

**Дуюн Иван Александрович**, студент кафедры программного обеспечения вычислительной техники и автоматизированных систем. E-mail: duyun77@mail.ru. Белгородский государственный технологический университет им. В.Г. Шухова. Россия, 308012, Белгород, ул. Костюкова, д. 46.

**Кабалянц Петр Степанович**, кандидат технических наук, доцент кафедры программного обеспечения вычислительной техники и автоматизированных систем. Белгородский государственный технологический университет им. В.Г. Шухова. Россия, 308012, Белгород, ул. Костюкова, д. 46.

**Дуюн Татьяна Александровна**, доктор технических наук, заведующий кафедрой технологии машиностроения. E-mail: tanduun@mail.ru. Белгородский государственный технологический университет им. В.Г. Шухова. Россия, 308012, Белгород, ул. Костюкова, д. 46.

**Рыбак Лариса Александровна**, доктор технических наук, профессор кафедры технологии машиностроения. E-mail: rl\_bgtu@intbel.ru. Белгородский государственный технологический университет им. В.Г. Шухова. Россия, 308012, Белгород, ул. Костюкова, д. 46.

Поступила 05.09.2024 г.

© Дуюн И.А., Кабалянц П.С., Дуюн Т.А., Рыбак Л.А., 2024

**\*Duyun I.A., Kabalyants P.S., Duyun T.A., Rybak L.A.**  
Belgorod State Technological University named after V.G. Shukhova  
\*E-mail: duyun77@mail.ru

## REINFORCEMENT LEARNING-BASED INTELLIGENT CONTROL FOR STEWART PLATFORM

**Abstract.** This article presents an intelligent control system for the Stewart platform, a parallel kinematic mechanism with six degrees of freedom. One of the main challenges of such mechanisms is the presence of singular positions within the workspace, which can lead to instability. Standard control algorithms often prove ineffective when navigating these potentially unstable zones. Therefore, the proposed control system utilizes Reinforcement Learning, a type of machine learning, as its core component. The choice of this method is

*motivated by its effectiveness in continuous action spaces, which is crucial for ensuring smooth movement within a workspace with variable coordinates. In conditions involving operational loads, vibrations, and temperature fluctuations, real-time correction is necessary. The application of an artificial neural network enhances accuracy and flexibility, enabling the system to adapt to changing operational conditions without compromising performance by modeling complex nonlinear dependencies and learning from accumulated experience. The paper discusses the following aspects of the control system: the working scheme and overall architecture of the reinforcement learning method, the learning algorithm scheme, UML diagrams of the environment and agent classes, the architecture of the actor-critic network, the network training process, and the results of its testing, which demonstrate high efficiency in its application.*

**Keywords:** Stewart platform, reinforcement learning, neural networks, parallel kinematics, intelligent control system, simulation in PyBullet

## REFERENCES

1. Duyun I.A., Kornilov A.V., Duyun T.A. Application of Python-Adams Interface for Modeling the Operation of the Gough-Stewart Platform [Primenenie interfeysa Python-Adams dlya modelirovaniya raboty platformy Gyuy-Syuarta]. Bulletin of BSTU named after V.G. Shukhov. 2022. No. 10. Pp. 96–106. DOI: 10.34031/2071-7318-2022-7-10-96-106. (rus)
2. Duyun I.A., Duyun T.A., Kabalyants P.S. Modeling of Singular Positions of a Robotic Platform and Determination of its Controllability Parameters [Modelirovanie osobykh polozheniy robotizirovannoy platformy i opredelenie parametrov eye upravlyaemosti]. STIN. 2023. No. 5. Pp. 2–6. (rus)
3. Duyun I.A., Kabalyants P.S., Zuev S.V., Duyun T.A., Malyshev D.I. Determination of the Controllability Index and Identification of Singular Positions of a Parallel Manipulator [Opredelenie indeksa upravlyaemosti i vyiyavlenie osobykh polozheniy parallel'nogo manipulyatora]. Bulletin of BSTU named after V.G. Shukhov. 2023. No. 5. Pp. 116–125. DOI: 10.34031/2071-7318-2023-8-5-116-125. (rus)
4. Duyun I.A., Gorlov A.S., Duyun T.A. Co-Simulation of Parallel Manipulator Motion Using Adams-Matlab [Sovmestnoe modelirovanie dvizheniya parallel'nogo manipulyatora s ispol'zovaniem Adams-Matlab]. Bulletin of BSTU named after V.G. Shukhov. 2022. No. 11. Pp. 108–119. DOI: 10.34031/2071-7318-2022-7-11-108-119. (rus)
5. Duyun T.A., Duyun I.A., Rybak L.A., Perezvuznik V.V. Simulation of the Structural and Force Parameters of a Robotic Platform Using Co-Simulation. Procedia Computer Science. 2022. Vol. 213. Pp. 720–727. DOI: 10.1016/j.procs.2022.11.126.
6. Duyun I.A., Duyun T.A., Kabalyants P.S., Rybak L.A. Optimization of a 6-DOF Platform for Simulators Based on an Analysis of Structural and Force Parameters. Machines. 2023. Vol. 11. Pp. 1–18. DOI: 10.3390/machines11080814.
7. Leonov G.A., Zegzhda S.A., Zuev S.M. Dynamics of the Stewart Platform and Control of its Motion [Dinamika platformy Styuarata i upravlenie
8. Ribeiro P.R., Andre M., Philipp K., Karel K. A VR Truck Docking Simulator Platform for Developing Personalized Driver Assistance. Applied Sciences. 2021. 11(19). 8911. DOI: 10.3390/app11198911.
9. Vieira H.L., de Carvalho Fontes J.V., Beck A.T., da Silva M.M. Robust Critical Inverse Condition Number for a 3RRR Robot Using Failure Maps. Springer International Publishing. 2018. Pp. 285–294. DOI: 10.1007/978-3-319-67567-1\_27.
10. Dasmahapatra S., Ghosh M. Workspace Identification of Stewart Platform. International Journal of Engineering and Advanced Technology. 2020. Vol. 9. Pp. 1903–1907. DOI: 10.35940/ijeat.C5338.029320.
11. Peng Y., Dai H., Zhang H., Yue X. Dynamics and Control of a Bio-Inspired Stewart Platform. Xibei gongye Daxue Xuebao / Journal of Northwestern Polytechnic University. 2021. Vol. 39. Pp. 258–266. DOI: 10.1051/jnwpu/20213920258.
12. Karakas B., Senay B. Kinematics of Supination and Pronation with Stewart Platform. J. Math. Sci. Model. 2021. Vol. 4. Pp. 1–6. DOI: 10.33187/jmsm.815125.
13. Merlet J.P. Jacobian, Manipulability, Condition Number, and Accuracy of Parallel Robots. Journal of Mechanical Design. 2006. Vol. 128. Pp. 199–206. DOI: 10.1115/1.2121740.
14. Dash K.K., Choudhury B.B., Senapati S.K. Inverse Kinematics Solution of a 6-DOF Industrial Robot.
15. Vieira H.L., da Silva M.M., Beck A.T. Reliable Motion Planning for Parallel Manipulators. Mechanism and Machine Theory. 2019. Vol. 140. Pp. 553–566. DOI: 10.1016/j.mechmachtheory.2019.06.022.
16. Zhukov Y.A., Korotkov E.B., Moroz A.V., Zhukova V.V., Abramov A.M. Adaptive Neural Network Control of Hexapod for Aerospace Application. IOP Conference Series: Materials Science and Engineering. 2018. Vol. 441. P. 012001. DOI: 10.1088/1757-899X/441/1/012001.

17. Valentino K., Kevin K., Singgih J., End D. Dynamic Models Based Virtual Reality Flight Simulator. *Internetworking Indones. J.* 2022. Vol. 11. Pp. 23–28.

18. Li S., Jin L., Mirza M. *Neural Network Based Stewart Platform Control*. Wiley-IEEE Press. 2019. Vol. 7. Pp. 105–129. DOI: 10.1002/9781119557005.ch7.

*Information about the authors*

**Duyun, Ivan A.** Student. E-mail: duyun77@mail.ru. Belgorod State Technological University named after V.G. Shukhov. Russia, 308012, Belgorod, st. Kostyukova, 46.

**Kabalyants, Petr S.** Candidate of Engineering Sciences. Belgorod State Technological University named after V.G. Shukhov. Russia, 308012, Belgorod, st. Kostyukova, 46.

**Duyun, Tatyana A.** Doctor of Engineering Sciences. E-mail: tanduun@mail.ru. Belgorod State Technological University named after V.G. Shukhov. Russia, 308012, Belgorod, st. Kostyukova, 46.

**Rybak, Larisa A.** Doctor of Engineering Sciences. E-mail: rl\_bgtu@intbel.ru. Belgorod State Technological University named after V.G. Shukhov. Russia, 308012, Belgorod, st. Kostyukova, 46.

---

*Received 05.09.2024*

**Для цитирования:**

Дуюн И.А., Кабелянц П.С., Дуюн Т.А., Рыбак Л.А. Интеллектуальная система управления платформой Стюарта на основе применения метода обучения с подкреплением // Вестник БГТУ им. В.Г. Шухова. 2024. № 11. С. 102–115. DOI: 10.34031/2071-7318-2024-9-11-102-115

**For citation:**

Duyun I.A., Kabalyants P.S., Duyun T.A., Rybak L.A. Reinforcement learning-based intelligent control for Stewart platform. *Bulletin of BSTU named after V.G. Shukhov.* 2024. No. 11. Pp. 102–115. DOI: 10.34031/2071-7318-2024-9-11-102-115