ТЕХНИЧЕСКИ УНИВЕРСИТЕТ – СОФИЯ

Факултет АВТОМАТИКА
Катедра АВТОМАТИЗАЦИЯ НА ЕЛЕКТРОЗАДВИЖВАНИЯТА

маг. инж. Анастасия Владимировна Славова

СИСТЕМИ С ДЪЛБОКО ОБУЧЕНИЕ ПРИ АВТОНОМНИТЕ МОБИЛНИ РОБОТИ

А В Т О Р Е Ф Е Р А Т

на дисертация за придобиване на образователна и научна степен
„ДОКТОР“

Област: 5. Технически науки

Професионално направление: 5.2. Електротехника, електроника и автоматика

Научна специалност: „Системи с изкуствен интелект“

Научен ръководител: доц. д-р Владимир Д. Христов

СОФИЯ, 2026 г.

Дисертационният труд е обсъден и насочен за защита от Катедрения съвет на катедра „АВТОМАТИЗАЦИЯ НА ЕЛЕКТРОЗАДВИЖВАНИЯТА“ към Факултет АВТОМАТИКА на ТУ-София на редовно заседание, проведено на 25.02.2026 г.

Публичната защита на дисертационния труд ще се състои на 06.07.2026 г. от 13:00 часа в Конферентната зала на БИЦ на Технически университет – София на открито заседание на научното жури, определено със заповед ОЖ-5.2-27 от 12.03.2026 г. на Ректора на ТУ-София в състав:

1. Доц. д-р инж. Марин Милков Жилевски – председател
2. Проф. д-р инж. Михо Рачев Михов – научен секретар
3. Доц. д-р инж. Никола Георгиев Шакев
4. Проф. д-р инж. Александра Иванова Грънчарова
5. Доц. д-р инж. Денис Сафидинов Чикуртев

Рецензенти:

1. Доц. д-р инж. Марин Милков Жилевски
2. Доц. д-р инж. Денис Сафидинов Чикуртев

Материалите по защитата са на разположение на интересуващите се в канцеларията на Факултет АВТОМАТИКА на ТУ-София, блок № 2, кабинет № 2340.

Дисертантът е редовен докторант към катедра „АВТОМАТИЗАЦИЯ НА ЕЛЕКТРОЗАДВИЖВАНИЯТА“ на факултет АВТОМАТИКА. Изследванията по дисертационната разработка са направени от автора, като някои от тях са подкрепени от научноизследователски проекти.

Автор: маг. инж. Анастасия Владимировна Славова
Заглавие: СИСТЕМИ С ДЪЛБОКО ОБУЧЕНИЕ ПРИ АВТОНОМНИТЕ МОБИЛНИ РОБОТИ
Тираж: 30 броя
Отпечатано в ИПК на Технически университет – София

I. ОБЩА ХАРАКТЕРИСТИКА НА ДИСЕРТАЦИОННИЯ ТРУД

Актуалност на проблема

Актуалността на разглеждания в дисертационния труд проблем се определя от стремителното развитие на автономните мобилни роботи (АМР) и нарастващата им роля в индустрията, логистиката, обслужващите дейности и работата в рискови среди. Съвременните производствени и складови системи изискват висока степен на автоматизация, гъвкавост и безопасност, което поставя повишени изисквания към методите за управление и навигация на АМР. Традиционните подходи, базирани на предварително изградени карти и класически алгоритми за планиране на пътя, показват ограничения при работа в динамични и частично наблюдаеми среди, където е необходима адаптация в реално време и устойчивост към несигурност в сензорните данни.

В този контекст дълбокото обучение с подсилване се утвърждава като перспективен инструмент за реализиране на автономна навигация без предварително изградени карти и без зависимост от GPS инфраструктура. Въпреки значителния напредък в областта остават нерешени въпроси, свързани със стабилността и скоростта на обучение, ефективността на използване на натрупания опит, избора на подходяща архитектура и хиперпараметри, както и с преноса на обучени модели от симулационна към реална среда. Тези проблеми имат както теоретично, така и ясно изразено практическо значение.

Допълнителна актуалност на изследването придава необходимостта от разработване на ресурсно ефективни решения, базирани на ограничен набор от сензори, като LiDAR и одометрия, с цел намаляване на хардуерната сложност и разходите за внедряване. Създаването и експерименталната верификация на модел за автономна навигация в реална среда – при минимална сензорна конфигурация и без използване на карти – представлява съществен принос към развитието на интелигентни, достъпни и приложими автономни мобилни системи.

Цел на дисертационния труд, основни задачи и методи за изследване

Целта на дисертационния труд е да се разработи система за управление на автономни мобилни роботи в условия на неопределеност, базирана на дълбоко обучение с подсилване, която гарантира безопасна работа и притежава голяма обобщаваща способност при използване на ограничен набор сензорни данни.

Задачите на дисертационния труд са:

1. Изследване и сравнителен анализ на приложението на модели за навигация чрез подхода обучение с подсилване (reinforcement learning, RL), включително техните алгоритмични особености, предимства и ограничения.

2. Избор, обучение и експериментална оценка на алгоритъм за автономна навигация съобразно заложените критерии.

3. Изследване на сходимостта на предложения подход при задачи за навигация.

4. Експериментално оценяване на приложимостта на предложените модели в реална среда с участие на автономен мобилен робот.

5. Систематичен анализ на устойчивостта на моделите при пренос от симулация в реална физическа среда.

Научна новост

Интегрирането на архитектурни и алгоритмични подобрения в обучението с подсилване (модифицирана невронна архитектура, адаптивна скорост на обучение и оптимизирано използване на буфер за опит) води до статистически значимо подобрение на сходимостта, стабилността и навигационната ефективност на колесен автономен мобилен робот спрямо базовата реализация на алгоритъма както в симулационна среда, така и в реална физическа постановка.

Практическа приложимост

Практическата приложимост на разработената в дисертационния труд система се изразява във възможността за внедряване на автономна навигация на мобилни роботи в реални индустриални и обслужващи среди – складове, производствени халета, логистични центрове и закрити пространства със сложна конфигурация – без необходимост от предварително изградени карти и скъпа сензорна инфраструктура. Предложеният подход, базиран на дълбоко обучение с подсилване и използване на ограничен набор от сензори (LiDAR и одометрия), позволява създаване на по-достъпни, адаптивни и икономически ефективни роботизирани решения, способни да работят в динамични и частично неизвестни среди, като същевременно се запазва висока степен на автономност, безопасност и гъвкавост при изпълнение на навигационни задачи.

Апробация

Резултатите са апробирани чрез поетапна експериментална верификация в симулационна и реална среда. Първоначално предложените алгоритми са обучени и сравнително анализирани в двумерна среда с използване на симулатора Flatland. Избран е най-подходящ модел за поставената задача и е реализирано неговото обучение в триизмерна симулационна среда с използване на ROS2 и Gazebo, където също така са изследвани влиянието на хиперпараметрите, архитектурата на невронната мрежа и функцията на възнаграждение върху сходимостта и ефективността на алгоритъма. Направени са алгоритмични и архитектурни подобрения на базовия модел. Впоследствие базовият и подобреният модели са внедрени и тествани върху реална роботизирана платформа Yahboom RDK X3, като са проведени поредица от експерименти в затворено пространство с различна конфигурация на препятствията, оценени чрез показатели като успеваемост при достигане на целта, оптималност на траекторията и устойчивост при пренос от симулация към реална среда.

Публикации

Основните постижения и резултати от дисертационния труд са публикувани в 7 научни статии, от които 4 са индексирани в Scopus, 2 в IEEE, а 1 е самостоятелно.

Структура и обем на дисертационния труд

Дисертационният труд е в обем от 151 страници, като включва увод, 4 глави за решаване на формулираните основни задачи, списък на основните приноси, списък на публикациите по дисертацията и използвана литература. Цитирани са общо 105 литературни източници, като 80 от тях са на латиница, а останалите са интернет адреси. Работата включва общо 80 фигури и 23 таблици. Номерацията на главите, точките, фигурите и таблиците в автореферата съответстват на тези в дисертационния труд.

II. СЪДЪРЖАНИЕ НА ДИСЕРТАЦИОННИЯ ТРУД

Глава 1. Дълбоко машинно обучение при автономни мобилни роботи – приложение, актуално състояние и проблеми

1.1 Приложение на дълбокото машинно обучение при автономни мобилни роботи

Направен обзор на прилагането на дълбокото машинно обучение в автономни мобилни роботи, като са описани основните използвани методи – конволюционни невронни мрежи, рекурентни невронни мрежи и обучение с подсилване за вземане на решения, управление и навигация. Представена е класификация на автономни мобилни роботи според средата на движение – наземни, въздушни и водни, като е акцентирано върху интеграцията на сензорни системи и алгоритми за управление, базирани върху дълбоко машинно обучение, позволяващи автономна работа в сложни и динамични среди.

1.2. Предимства на навигацията без предварително изградени карти и GPS, базирана на обучение с подсилване

Интеграцията на метода на обучение с подсилване значително разширява възможностите на мобилните роботи, като ги превръща от изпълнители на прости повтарящи се задачи в интелигентни, самостоятелни и колаборативни системи. Особено перспективна е автономната навигация в динамични и непознати среди, където традиционните методи като SLAM и алгоритмите за планиране на пътя могат да се окажат неефективни. Обучението с подсилване позволява навигация без предварително изградени карти или GPS, като агентът се адаптира динамично към препятствия и промени в средата, използвайки ограничен набор от сензори и локални наблюдения. Този подход осигурява адаптивност, устойчивост на шум и възможности за използване на сензорните данни в условия на несигурност, намалява зависимостта от външна инфраструктура и хардуерната сложност и предоставя гъвкава и ресурсно ефективна алтернатива на конвенционалните навигационни методи, което подчертава значимостта му за научни и практически приложения.

1.3. Обучение с подсилване – дефиниция, предимства и стратегии за обучение

Обучението с подсилване позволява автономно придобиване на оптимално поведение чрез итеративно взаимодействие с околната среда. Подходът се базира върху създаването на агенти, които се учат от околната среда, като взаимодействат с нея чрез проби и грешки и получават възнаграждение (положително или отрицателно) като уникална обратна връзка. Акцентира се върху трите основни стратегии за обучение: директни, индиректни и хибридни.

1.4. Съставяне на функцията на възнаграждение

Представено е значението на функцията на възнаграждение за обучението на агенти с дълбоко обучение с подсилване при навигация на автономни мобилни роботи, като са описани основните принципи за нейното проектиране – поощрения за приближаване към целта, достигане на целевата позиция и намиране на по-кратък път, както и наказания при сблъсък с препятствия и изтичане на времевите ограничения за изпълнение на задачата. Посочени са допълнителни похвати като ориентация спрямо целта, време за достигане и плавност на движението, както и използване на човешка обратна връзка за обучение на желано поведение.

1.5. Подходи за обучение с подсилване в задачата за навигация на автономни мобилни роботи

Разгледани са няколко перспективни подхода за обучение с подсилване. При анализ на посочените модели се използва набор от технически и практически критерии, които отразяват ограниченията на реалната система, средата и целите на задачата, като:

1. Сложност и динамика на средата, включваща наличието на частична наблюдаемост в средата и динамични обекти, шум в сензорите (LiDAR, одометрия);
2. Стабилност на обучението, описваща чувствителността на алгоритъма към неустойчиви актуализации на параметрите;
3. Изчислителни ресурси, необходими за обучение на алгоритмите;
4. Скорост на сходимост (време за обучение), отчитаща необходимостта от алгоритми с бърза сходимост при симулации, изискващи големи изчислителни ресурси, или при експерименти с физически среди;
5. Ефективност на използването на опита, определяща броя взаимодействия със средата за всеки алгоритъм;
6. Робастност и безопасност;
7. Чувствителност на алгоритъма към настройването на хиперпараметри;
8. Приложимост към поставената задача – навигация в среда с наличие на препятствия.

1.6. Обобщен анализ на алгоритмите и проблеми при използване на подхода DRL за решаване на задачата за навигация

Подходът на обучение с подсилване предоставя мощен инструмент за автономна навигация на мобилни роботи в сложни и динамични среди, позволявайки адаптивно поведение без предварителни карти или GPS. Основните му ограничения са дългото време за обучение, високите изчислителни изисквания и чувствителността към хиперпараметри. Алгоритмите, базирани на политика (като PPO и TRPO), се отличават с по-голяма стабилност, ефективност и приложимост в реални среди в сравнение с алгоритми, базирани на стойност (като DQN). Въпреки това остават нерешени редица съществени проблеми: ниска ефективност на използването на опита и бавна сходимост при някои алгоритми; силна зависимост от специфично дефинирана функция на възнаграждение; ограничена съпоставимост между различни изследвания; недостатъчно изследван пренос от симулационна към реална среда при използване на минимална сензорна конфигурация и без предварително изградени карти.

Цел и задачи

Въз основа на гореизложеното, **целта** на дисертационния труд е да се разработи система за управление на автономни мобилни роботи в условия на неопределеност, базирана на дълбоко обучение с подсилване, която гарантира безопасна работа и притежава голяма обобщаваща способност при използване на ограничен набор сензорни данни.

Формулирани са следните основни задачи: извършване на сравнителен анализ на подходящи алгоритми за навигация; избор и имплементация на подходяща симулационна среда; разработване и настройване на функция на възнаграждение; оптимизация на архитектурата и хиперпараметрите на невронната мрежа; обучение и сравнителна оценка на моделите в 2D и 3D симулация; експериментална верификация на подобрения модел в реална среда с анализ на неговата приложимост и устойчивост.

Глава 2. Избор на подходящи модели за навигация на АМР

2.1 Описание и имплементация на симулационната среда

Представена е реализация на метода на обучение с подсилване за автономна навигация на мобилен колесен робот в двумерна симулационна среда Flatland с интеграция с ROS2. Симулаторът поддържа физика, частична наблюдаемост и възприятие от първо лице, позволявайки бързо прототипиране и тестване на алгоритми с използване на библиотеки от програмния език Python. Описана е архитектурата на ROS2 с възли, теми, услуги, действия и параметри, както и tf2 за трансформации между координатните системи. Агентът е представен като диференциален двуколесен мобилен робот, оборудван с LiDAR сензор. Задачата на робота е намиране на най-бърз и безопасен път до целта в затворено пространство. Дефинирана е функцията на възнаграждение по такъв начин, че да насърчи бързо и безопасно приближаване на робота към целта.

2.2. Обучение и анализ на модели за навигация

Описани са структурите, архитектурите на невронните мрежи и използваните хиперпараметри, използвани за обучение на 4 алгоритъма по метода с подсилване: DQN, A2C, TRPO и PPO. Обучението на всеки от посочените модели се осъществява в една и съща затворена симулационна среда с наличие на препятствия. За анализа на успеваемостта на моделите за решаване на поставената задача са използвани следните метрики:

1. Кумулативно възнаграждение за епизод – измерва общото натрупано възнаграждение за всеки епизод и показва напредъка на агента при научаването на оптималната стратегия.
2. Брой стъпки за достигане на целта – оценява ефективността на навигацията и времето за изпълнение на задачата, като намаляването на броя стъпки показва по-ефективно поведение.
3. Крайни състояния на епизодите – класифицират се на успешно завършени епизоди, епизоди със сблъсък с препятствие или изтичане на времето, като така се оценява безопасността и надеждността на обучението.
4. Процент успешно завършени епизоди при тестване – измерва точността и стабилността на научената политика по време на тестови итерации (например 90% успех при PPO).
5. Дисперсия на кумулативното възнаграждение – оценява стабилността и последователността на резултатите между епизодите, като по-малката дисперсия показва по-надежден модел.

2.3 Обобщен анализ на получените резултати

Представен е обобщен статистически анализ на алгоритмите съобразно скоростта на сходимост, получените възнаграждения, продължителността на епизодите по време на обучение, както и параметрични оценки за всеки алгоритъм (Табл. 12).

Табл. 1. Статистически анализ на получени резултати

Показател	DQN	A2C	TRPO	PPO	Статистическа значимост

Скорост на сходимост (епизоди)	2814	1712	676	392	-
Подобрение на сходимостта спрямо предходния алгоритъм	-	39%	61%	42%	$p < 0.01$
Успешно завършени епизоди (%)	10	20	80	90	-
Средно възнаграждение	-180.8	-119.2	150.4	170.6	$p < 0.05$
Стандартно отклонение на възнаграждението	45.4	42.3	352.8	362.1	-
Среден брой стъпки/епизод	17.3	187.2	127.9	108.3	$p < 0.05$
Стандартно отклонение на броя стъпки	20.5	41.5	57.4	52.7	-

Резултатите показват ясно изразено подобрение в представянето на алгоритмите при преминаване от подходи, основани на стойности и актьор-критик (DQN, A2C), към основани на политики (TRPO, PPO). Скоростта на сходимост се увеличава последователно от DQN към PPO, като броят на необходимите епизоди намалява с 86% от 2814 при DQN до 392 при PPO. Успеваемостта при завършване на епизодите нараства рязко: от 10 – 20% при DQN и A2C до 80% при TRPO и 90% при PPO, което показва по-добра стабилност и надеждност на подходите, базирани на политики.

2.4 Изводи

Проведеният в настоящата глава сравнителен анализ на алгоритмите DQN, A2C, TRPO и PPO в двумерна симулация позволява формулирането на ясно обосновани научни изводи относно тяхната приложимост при задачата за автономна навигация. Изследването показва, че подходът, основан на стойности, DQN демонстрира ограничена ефективност, изразена в бавна сходимост, ниска успеваемост и отрицателни стойности на средното възнаграждение, което го прави неподходящ за решаване на сложни навигационни задачи в непрекъснати и динамични среди. Подходите, основани на политики и актьор-критик, A2C, TRPO и PPO показват значително по-добро представяне по всички разглеждани показатели. В частност, при TRPO и PPO се наблюдава съществено ускоряване на обучението, рязко повишаване на процента успешно завършени епизоди и значително по-високи стойности на средното възнаграждение. Това потвърждава теоретичните предимства на директната оптимизация на политиката при задачи, изискващи стабилно и последователно поведение.

Установено е, че алгоритъмът PPO демонстрира най-добър баланс между скорост на сходимост, стабилност на обучението и качество на получената навигационна политика. В сравнение с TRPO, PPO постига сходни или по-добри резултати при значително по-ниска алгоритмична сложност и по-лесна настройка,

което го прави по-подходящ за практическа реализация и последващо разширяване. На базата на тези научни изводи алгоритъмът PPO е избран като базов метод за по-нататъшните изследвания, представени в следващата глава. Този избор е мотивиран както от експерименталните резултати, така и от възможността алгоритъмът да бъде ефективно надграждан чрез архитектурни и алгоритмични подобрения с цел повишаване на ефективността и устойчивостта на автономната навигация.

Паралелно с избора на алгоритъм анализът обосновава и използването на двумерна симулационна среда за целите на настоящото изследване. Двумерната симулация позволява ефективно моделиране на основните трудности при навигация – избягване на препятствия, планиране на траектории и достигане на цел – при значително по-ниска изчислителна сложност в сравнение с тримерните среди. Това създава условия за провеждане на систематични експерименти, бърза итерация на модели и обективно сравнение на различни алгоритмични конфигурации, без загуба на обобщаваща сила на получените резултати.

В следващата глава 3 е описано имплементирането на доказано най-ефективния модел PPO в тримерна симулацията с интегриране на ROS2. Функционалността на модела е изцяло реализирана без участие на спомагателната библиотека SB3, за да се осигурят възможности за максимално свободно модифициране на структурата и параметрите. С цел получаване на реалистична симулация, възможно най-близка до условията, физическите характеристики и сензорните данни на реалното устройство, се използва и 3D модел на робота чрез URDF описание. Това подпомага последващото интегриране на обучения RL модел в мобилна 4-колесна платформа Yahboom RDK X3 с меканум колела, която е избрана за целите на експерименталното изследване в реална физическа среда.

Глава 3. Обучение на модела за навигация в триизмерна симулационна среда

3.1 Описание на 3D симулатора Gazebo и характеристики на използвания софтуер

Представено е описание на симулатора Gazebo за 3D симулация на обучение на робот чрез подхода с подсилване, като се подчертани неговите възможности за висококачествена физическа симулация, поддръжка на различни сензори и задвижвания, интеграция с ROS2 и програмиране на Python. Представени са софтуерните компоненти и библиотеки, използвани за конфигуриране и обучение на робота (Yahboom RDK X3 с меканум колела и LiDAR), включително PyTorch, Numpy, Tensorboard и rclpy, както и ролята на RViz за визуализация и отстраняване на грешки.

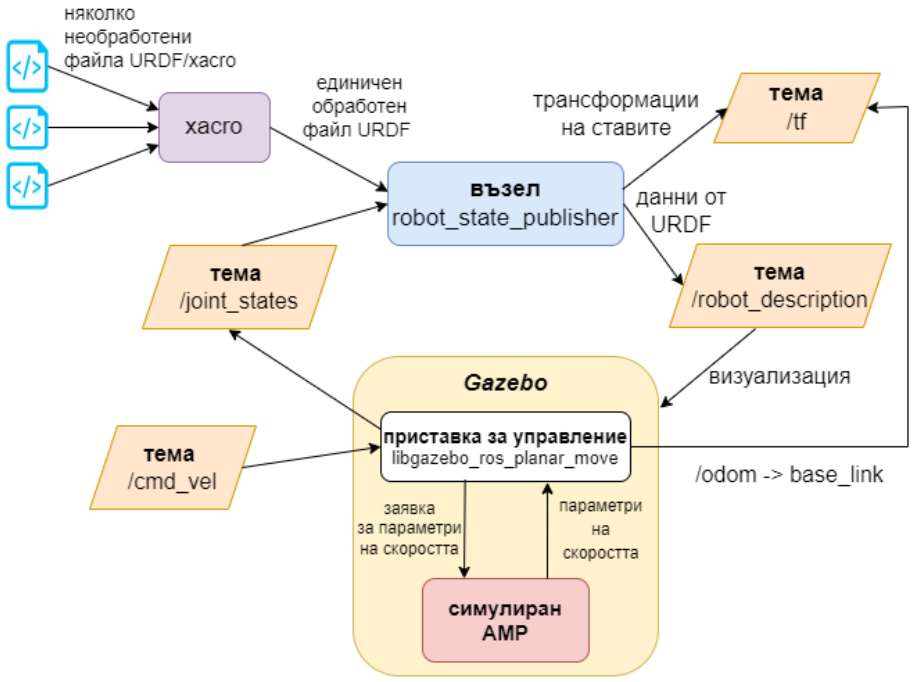
3.2 Подготовка на симулационната среда

Описана е цялостната подготовка и конфигурация на 3D симулационна среда за автономен мобилен робот в Gazebo и ROS2, която включва следните стъпки:

1. Визуализация на робота – създаване на 3D модел на Yahboom RDK X3 чрез URDF файлове, дефиниране на възли и стави, интеграция с RViz и конвертиране в SDF за Gazebo.
2. Построяване на симулирано затворено пространство – създаване на различни конфигурации с размери и препятствия за обучение на AMP.

3. Решаване на проблеми при симулацията – корекция на визуализация, цветове, позиции на камери и неподвижни звена, за да съвпадат с реалния робот.
4. Управление на автономен мобилен робот в симулация – използване на приставки за управление на колелата, публикуване на данни от LiDAR и други сензори чрез ROS2.
5. Позициониране на цел и автономен мобилен робот в симулация чрез случайно задаване на начални позиции на робота и целта в симулацията с цел научаване на адаптивни навигационни политики от агента.

На Фиг. 47 е представена схема на реализираната комуникация с използване на приставката за управление на робота в симулатора Gazebo.



Фиг. 1. Схема на управлението на робота в симулатора Gazebo.

3.3 Създаване на система за обучение по метода RL

Процесът на обучение с метода RL е комплексен, в частност обучение по метода PPO включва следните стъпки:

1. Инициализиране на средата и агента с включени хиперпараметри.
2. Задаване на начално състояние на средата за получаване на първоначални наблюдения.
3. За всяка итерация на актуализирането:
   а) събиране на данни за траекториите на придвижване на агента;
   б) изчисляване на възнаграждението за всяка стъпка и предимствата на състоянията;
   в) неколкократно актуализиране на функциите на политиката и стойностите спрямо входните данни;
   г) запазване на съответните показатели с цел проследяване на напредъка на обучението, както и най-добрите стойности на теглата на модела, получени досега.

Дефинирана е функция на възнаграждението за целите на навигацията въз основа на подкрепа на действията, водещи до избор на къса и безопасна траектория за достигане до целта.

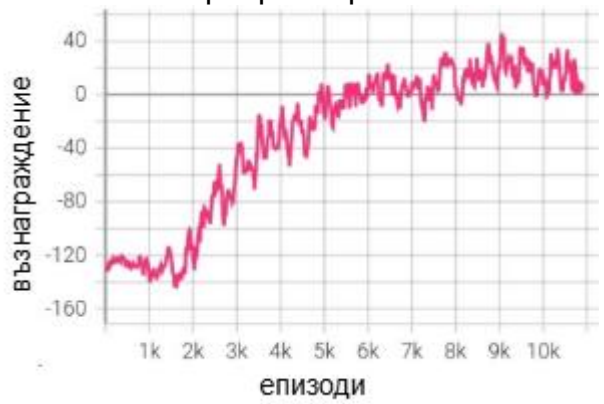
3.4 Обучение на модел за навигация по метода PPO

Проведена е експериментална настройка и фина оптимизация на хиперпараметрите и архитектурата на модела PPO за по-стабилно и ефективно обучение в среда на RL. По-конкретно са идентифицирани ключови хиперпараметри като скорост на обучение, размер на партидата, брой стъпки за събиране на данни, брой епохи на оптимизация, коефициент на ентропия, коефициент за стойностната функция, обхват на отрязване на политиката и нормализиране на предимствата. С цел определяне на тяхното влияние върху ефективността на обучението и стабилността на алгоритъма са проведени експерименти с различни стойности на хиперпараметрите. Резултатите са анализирани чрез сравнение на кумулативното възнаграждение за последните 100 епизода и показват влиянието на скоростта на обучение, размера на партидата и броя неврони в скритите слоеве върху качеството на обучението. Доказано е, че за поставената задача е целесъобразно да се използват по-бавна скорост на обучение, по-голям размер на партидите, по-малко епохи на итерация и по-плитка архитектура. След многобройни експерименти са избрани следните хиперпараметри за обучение на модела PPO за дадената задача:

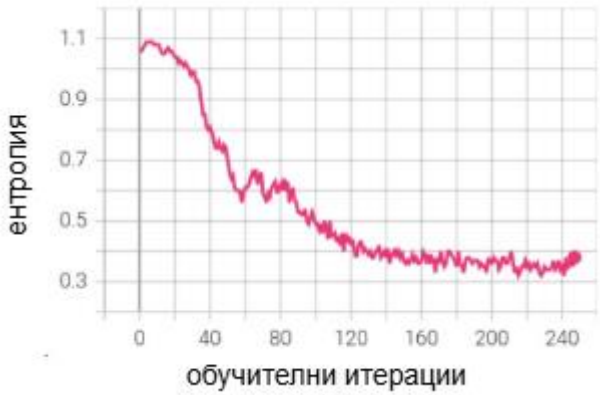
Табл. 15. Хиперпараметри, използвани за обучение на модела PPO в 3D симулационна среда

Хиперпараметър	Стойност
Брой стъпки	4096
Размер на партидата	512
Брой епохи	3
Обхват на отрязване	0.2
Коефициент на ентропия	0.001
Коефициент на обобщена оценка на предимствата	0.99
Скорост на обучение	0.003
Нормализиране на оценката на предимствата	True
Коефициент на стойност	True
Целеви показател на Кулбак-Лайблер	0.015

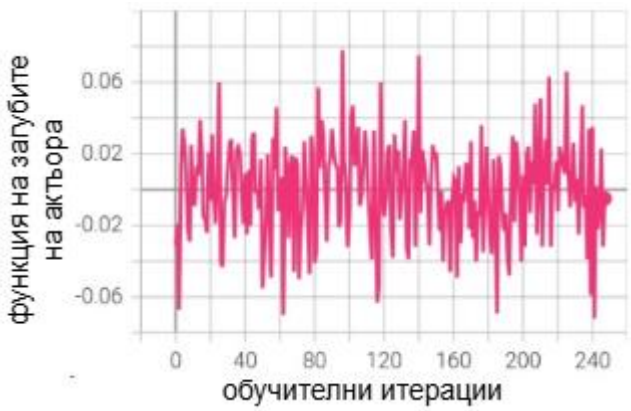
Характеристиките на показателите на обучената ДНМ с модел PPO с всички посочени хиперпараметри са показани на Фиг. 2.



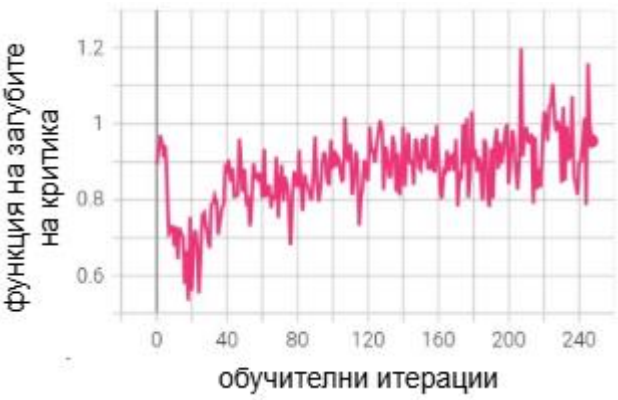
а.



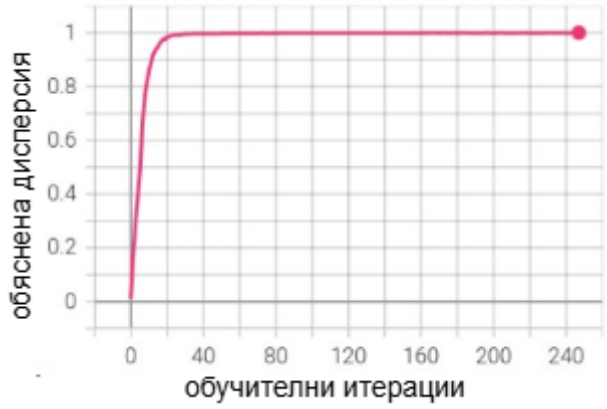
б.



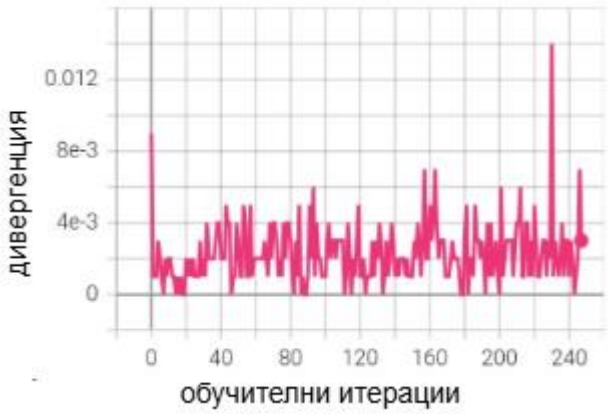
в.



г.

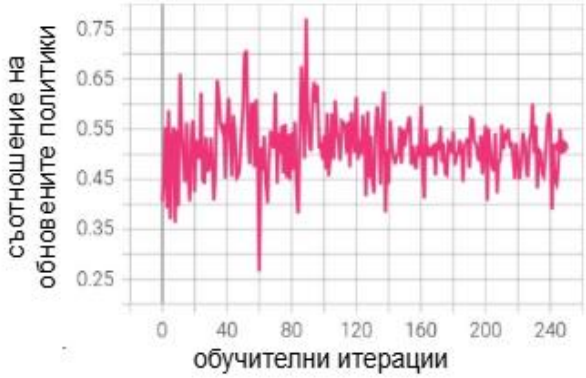


д.



е.

ж.



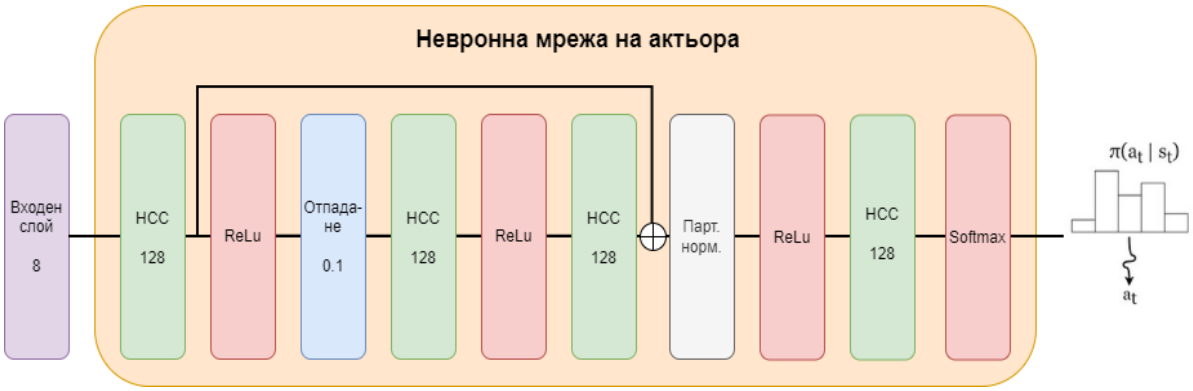
з.

Фиг. 2. Напредък на характеристиките на ДНМ по време на обучението: а) кумулативно възнаграждение, осреднено за последните 100 епизода; б) средна ентропия на разпределението на действията на актьора; в) функция на загубите на актьора; г) функция на загубите на критика; д) коефициент на обяснена дисперсия на критика; е) мярка за дивергенция на новата политика на Кулбак-Лайблер; ж) процентно съотношение на отрязаните вероятности за действия спрямо общия брой на действията в една партида; з) процентно съотношение на обновените политики спрямо общия им брой в една партида.

Агентът постига оптимално поведение с 90% точност между обучителните итерации след 9 000 епизода, като средната стойност на кумулативното възнаграждение доближава 45.

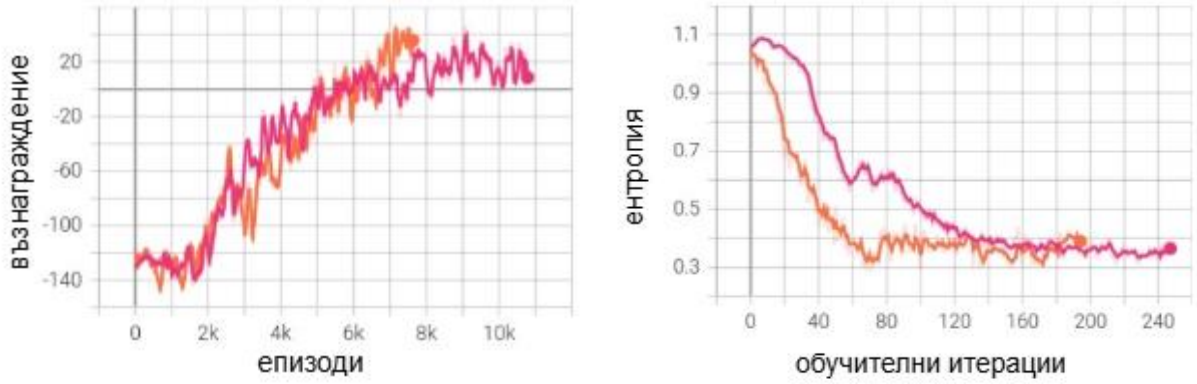
3.4.3 Промяна на архитектурата на ДНМ

С цел подобряване на скоростта на обучение по метода PPO е направено обновяване на архитектурата на ДНМ на актьора чрез добавяне на допълнително разклонение, базирано на подхода за обходни или остатъчни връзки (residual connections). Цялостната архитектура на ДНМ на актьора е показана на Фиг. 3.



Фиг. 3. Архитектура на ДНМ на актьора с обходна връзка

Така описаните подобрения в ДНМ допринасят за повишаване скоростта на обучение на модела PPO посредством намиране на нови зависимости на състоянията. Това по същество кара двете разклонения да научават различни представяния на входните данни, които впоследствие се сумират и преминават през последния скрит слой със 128 неврона, последван от активационна функция Softmax. Резултатите от експериментите с така описаната архитектура са показани на Фиг. 4.



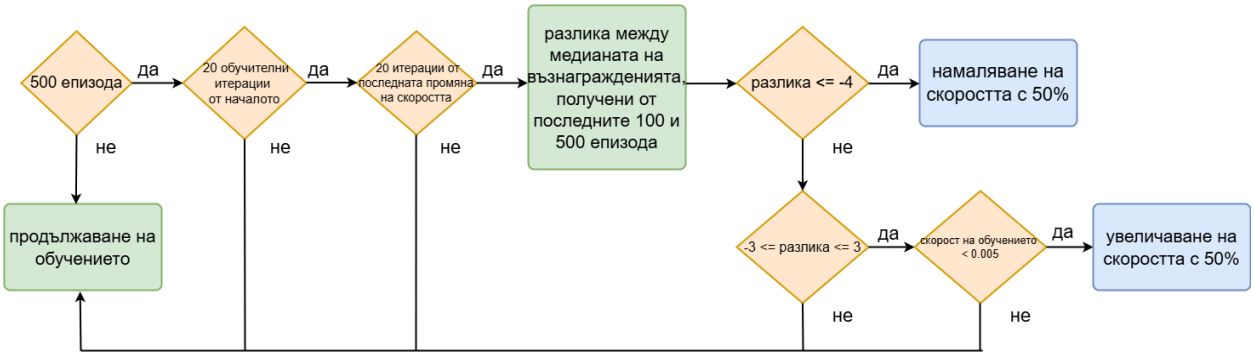
Фиг. 4. Обучение с обновената архитектура на актьора, използвайки подхода с обходни връзки: вляво – кумулативно възнаграждение, осреднено за последните 100 епизода; вдясно – средна ентропия на разпределението на действията на актьора

Процесът на обучение се ускорява след епизод 6 500 и достига максималната си средна стойност от 32 точки за последните 100 епизода към епизод 7 000. Същата стойност се достига от базовия модел 2 000 епизода по-късно.

3.4.4 Прецизиране на скоростта на обучение

В обучението на модела PPO за поставената задача са проведени експерименти с промяна на скоростта на обучение при актьора чрез оценка на

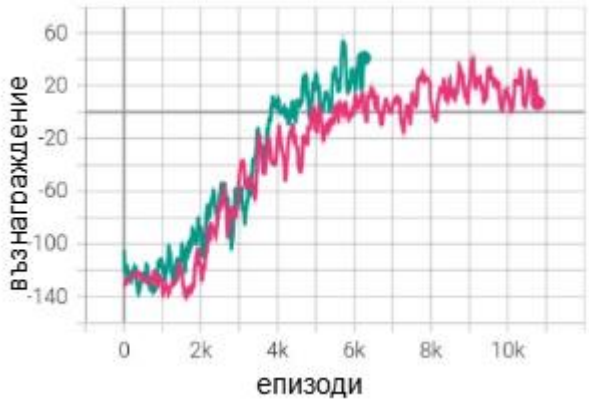
напредъка на обучението. Той се определя от представянето на агента през определен брой последни епизоди. Блок-схема на процеса е показана на Фиг. 5.



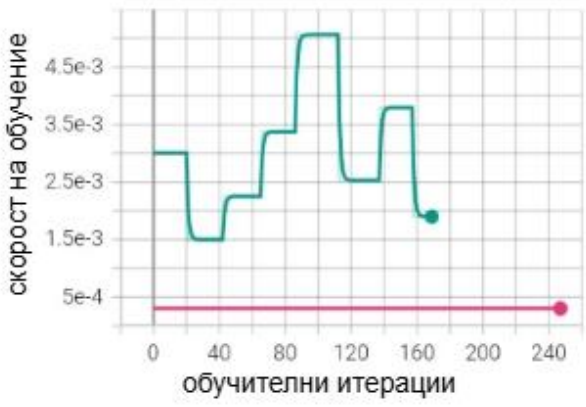
Фиг. 5. Алгоритъм на промяна на скоростта на обучението

Задават се минимални прагови стойности на брой изминали епизоди (500), обучителни итерации от началото на обучението (20) и последна промяна на скоростта на обучението (20). По този начин промяната на скоростта става плавно и въз основа на по-голяма извадка от данни. Сравнява се разликата на медианата на възнаграждения през последните 500 и 100 епизода (20%). Праговите стойности са намерени експериментално и се явяват хиперпараметри на алгоритъма. Статистическият параметър медиана е избрана с цел намаляване на влиянието на дисперсията на възнагражденията при откриване на непознати състояния от агента. При оценяване на напредъка на обучението, в случаите когато тя намалява (разликата е равна на -4), т.е. възнагражденията са с низходяща тенденция, скоростта на обучението следва да се намали двойно. От друга страна, когато агентът се намира в плато и функцията на възнагражденията не се променя (разликата е между -3 и 3), скоростта на обучение се увеличава с 50% спрямо предишната си стойност. Добавя се и допълнително условие – скоростта не трябва да надвишава 0.005, за да се избегнат прекалено големи стъпки в обновяването на теглата. По този начин при намаляване или спиране на напредъка на обучението се предприемат действия по промяна на скоростта на обучение, но когато е налице напредък, кривата на възнагражденията е възходяща и обучението продължава с избраната скорост.

На Фиг. 60 е показано намаляването на скоростта и средното възнаграждение на агента по време на обучение, сравнени с тези на базовия модел.



а)



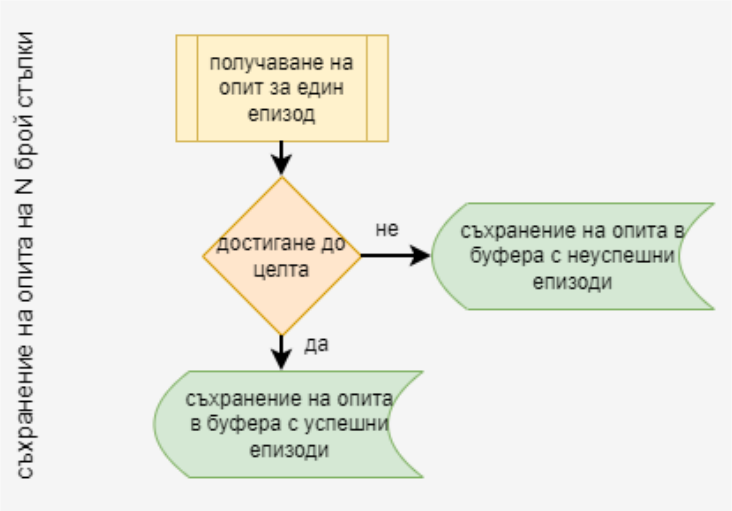
б)

Фиг. 6. Обучение при прилагане на алгоритъма за промяна на скоростта: а) кумулативно възнаграждение за последните 100 епизода; б) промяна на скоростта на обучението

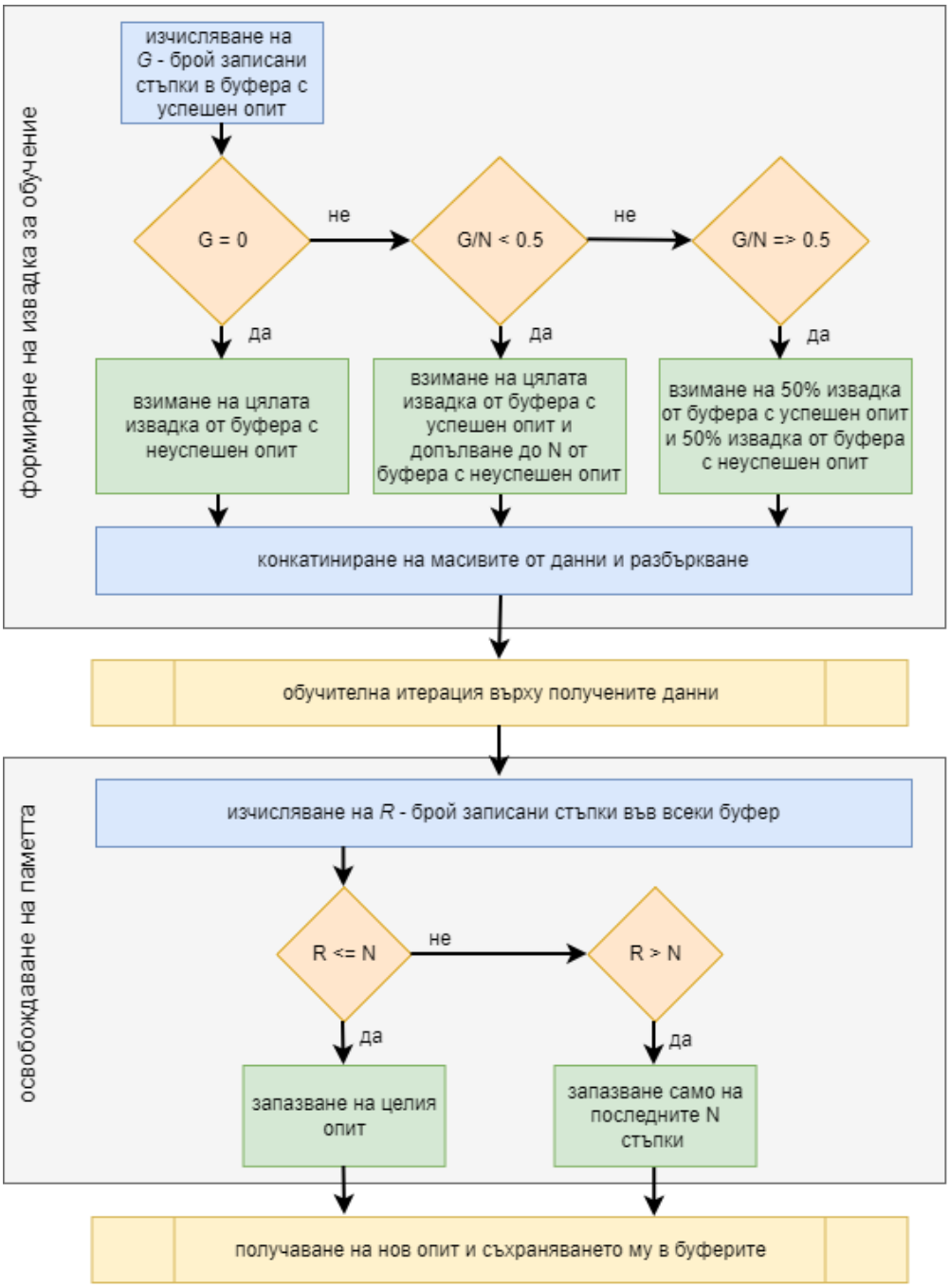
Обучението на алгоритъма до точност 90% с така описания подход за промяна на скоростта отнема значително по-малко време, а именно: 6 300 епизода при максимално възнаграждение 60 точки в сравнение с 9 000 епизода и възнаграждение 32 точки.

3.4.5 Буфер за съхранение на опита

Разглежда се нов начин за съхранение на опита, базиран на представянето на агента в конкретни епизоди и по-ефективното му използване. Процесът е разделен на три основни стъпки: съхранение на опита, формиране на извадка за обучение и почистване на паметта. За целта се създават два буфера за съхранение на опита с различно предназначение. Схемата на процеса е показана на Фиг. 61 и Фиг. 62.

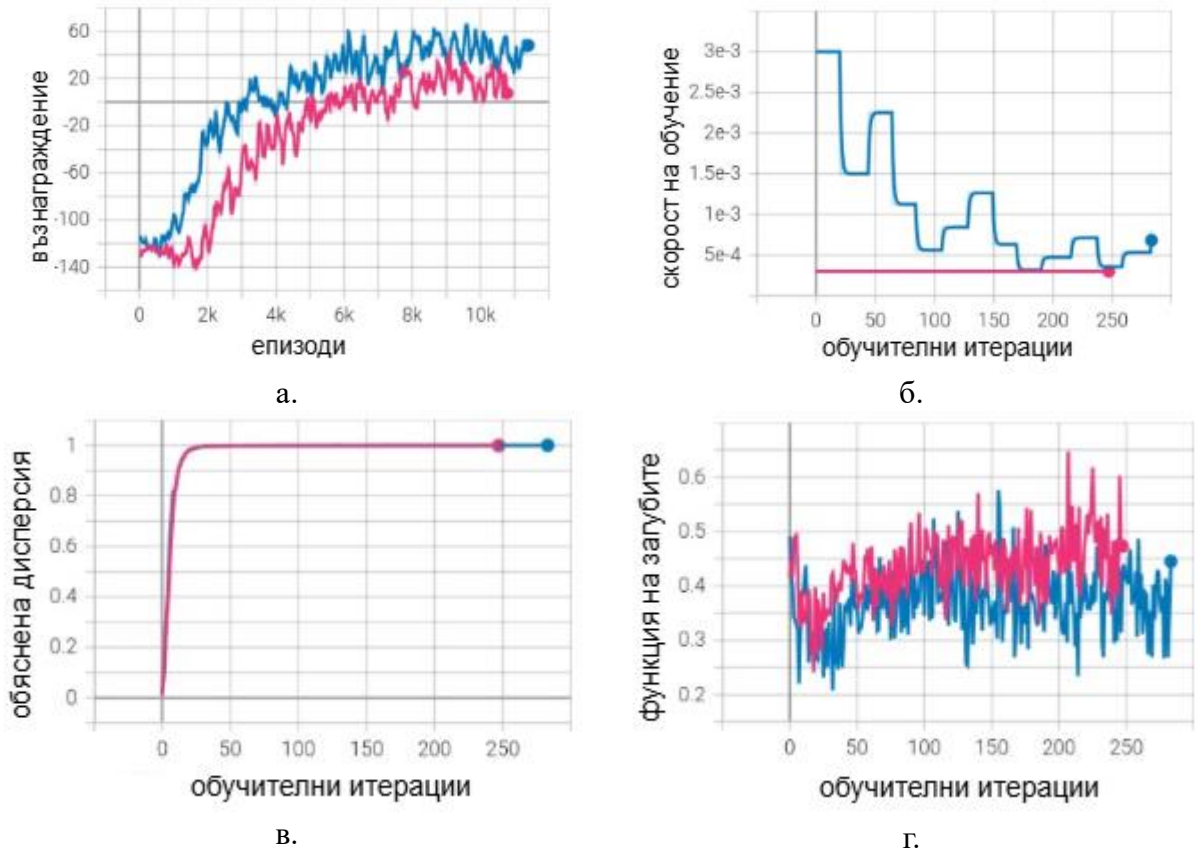


Фиг. 7. Процес на съхранение на получения опит



Фиг. 8. Формиране на извадка от буферите и освобождаване на паметта

Всеки епизод се съхранява с подробни данни за наблюдаваните състояния, действията, вероятностите, получените награди и крайния резултат. Когато се натрупат достатъчно стъпки (4 096), от двата буфера се формира обучителна извадка, като се подбират по-нови епизоди, съчетани според съотношението успешни/неуспешни. След това неактуалният опит се премахва, за да се гарантира, че агентът винаги се обучава с най-актуалните данни от средата, което повишава ефективността и адаптивността на обучението. Резултатите от така описания подход са показани на Фиг. 9.



Фиг. 9. Процес на обучение при прилагане на алгоритъма със съхранение на опита: а) кумулативно възнаграждение за последните 100 епизода; б) промяна на скоростта на обучение; в) коефициент на обяснена дисперсия на критика; г) функция на загубите

Резултатите от използването на така описания метод за управление на буфера на паметта показват по-ефективната му работа, което се наблюдава от кривата на кумулативното възнаграждение спрямо базовия модел – напредъкът на обучението настъпва значително по-рано и с по-възходяща тенденция, отколкото при базовия модел.

3.5 Обобщен сравнителен анализ на направените подобрения

В Табл. 2 е показан обобщен сравнителен анализ на базовия модел с предложените подобрени конфигурации, включващи промяна на архитектурата на ДНМ, адаптивна скорост на обучение и използване на получения опит.

Табл. 2. Количествена оценка на сравнителния анализ на базовия и подобрените модели

Показател	Базов модел	Модел с променена архитектура на ДНМ	Модел с адаптивна скорост на обучение	Модел с подобрено използване на получения опит
Скорост на сходимост (епизоди)	~9000	~7000	~6300	~6100
Успешно завършени тестови епизоди (%)	90%	90%	90%	95%
Средно възнаграждение	40	53	51	62
Стандартно отклонение на възнаграждението	10.8	10.3	9.7	10.1

Минимална стойност на ентропията	0.3	0.3	0.2	0.2
Итерация, при която се постига минимална ентропия на разпределението на действията на актьора	210	70	68	53
Минимална стойност на функцията на загубите на актьора	-0.07	-0.05	0.01	0.05
Стандартно отклонение на функцията на загубите на актьора	0.6	0.54	0.38	0.35
Минимална стойност на функцията на загубите на критика	0.65	0.67	0.49	0.21
Стандартно отклонение на функцията на загубите на критика	0.73	0.82	0.56	0.23
Максимален коефициент на обяснена дисперсия на критика	0.99	0.99	0.99	0.99
Итерация, при която се постига максимален коефициент на обяснена дисперсия на критика	20	19	18	18

По отношение на скоростта на обучение се наблюдава подобрение при всички модифицирани модели спрямо базовия модел. Докато базовият модел достига максималната успеваемост 90% след приблизително 9 000 епизода, моделът с променена архитектура с използване на обходна връзка и допълнителна регуляризация на ДНМ на актьора подобрява скоростта на сходимост с 22% (до 7 000 епизода) при същата успеваемост при тестови епизоди (90%). Допълнителното внедряване на адаптивност в скоростта на обучение води до намиране на същата успеваемост след 6 300 епизода, а най-добри резултати се постигат при модела с подобрено използване на буфера за съхранение и използване на опит, който се обучава за приблизително 6 100 епизода (32% подобрение спрямо базовия модел), като освен това достигната успеваемост при тестови епизоди се повишава с 5% (95%) спрямо базовия модел (90%).

3.6 Аблационен анализ на подобрения модел

С цел количествена оценка на приноса на предложените подобрения върху ефективността на алгоритъма PPO е проведен аблационен анализ, основната идея на който е систематично деактивиране на отделни компоненти от подобрения модел и измерване на влиянието им върху скоростта на обучение на алгоритъма, стабилността и крайното представяне на агента в тестови епизоди. Като референтна основа е използван базовият обучен модел PPO с първоначалната архитектура на актьора и критика, фиксирана скорост на обучение и стандартен начин на събиране на обучителните примери. Резултатите от проведените експерименти са показани в Табл. 3.

Табл. 3. Резултати от аблационния анализ на направените подобрения

Конфигурация на модела	Промяна на архитектурата	Адаптивна скорост на обучение	Подобрен буфер за опит	Скорост на сходимост (епизоди)	Минимална ентропия	Средно възнаграждение	Максимална достигната точност (%)
Базов модел	X	X	X	~9000	0.3	40	90
Без промяна на архитектурата	X	✓	✓	~8200	0.3	45	92
Без адаптивна скорост	✓	X	✓	~7500	0.25	48	90
Без подобрен буфер за опит	✓	✓	X	~7000	0.25	50	90
Подобрен модел (пълен)	✓	✓	✓	~6100	0.2	62	95

Резултатите от така проведените експерименти показват, че всяко от предложените подобрения допринася самостоятелно за повишаване на ефективността на алгоритъма PPO. Най-съществено влияние върху скоростта на обучение оказват промяната на архитектурата на ДНМ на актьора и адаптивната скорост на обучение, докато подходът за управление на буфера за опит подобрява устойчивостта и стабилността на обучението. Комбинирането на всички предложени подобрения води до най-добър баланс между висока скорост на обучение, стабилност и подобрена точност.

3.7 Изводи

В настоящата глава е разгледано реализирането на подхода за RL в 3D симулация Gazebo с участие на 3D модел на роботизираната мобилна 4-колесна платформа Yahboom RDK X3 с меканум колела и сензор LiDAR. Подготовката на системата за задачата на автономна навигация е многостъпков процес и включва създаване на симулационна среда, конфигуриране на сензори, настройване и внедряване на управление на робота с ROS2, създаване на система за обучение по метода RL. Реализираната симулационна среда създава контролирани и възпроизводими условия за обучение и тестване на алгоритми RL за поставената задача.

Проведените експерименти потвърждават възможността за успешно обучение на навигационна политика с подхода RL, използвайки данни от LiDAR и одометрия, без предварително изградени карти. Комбинацията от симулатор Gazebo с ROS2 и програмен език Python предоставя висока степен на модулност, гъвкавост и възможност за пълнофункционално управление на всички процеси от симулацията и обучението.

Реализирано е фино донастройване на алгоритъма PPO и е изграден базов модел, който се използва за последващите подобрения на алгоритъма. Достигнатата успеваемост от 90% при тестване потвърждава научната хипотеза

за високата ефективност на този подход при изпълнение на задачи за автономна навигация на АМР. Чрез внимателен подбор на хиперпараметрите, архитектурата на невронната мрежа и функцията на възнаграждението се постига устойчиво обучение и висока точност на агента при вземане на решения.

Извършен е задълбочен анализ и експериментално изследване на възможностите за подобряване на алгоритъма PPO в контекста на задачата за автономна навигация. Чрез въвеждането на архитектурни изменения в ДНМ, адаптивна скорост на обучение и усъвършенстван механизъм за управление на натрупания опит е постигнато съществено повишаване на ефективността и стабилността на обучението. Резултатите от сравнителния и аблационния анализ показват, че всяко от предложените подобрения има самостоятелен положителен принос към представянето на модела. Внедряването на обходни връзки и нормализация в архитектурата на актьора подобрява скоростта на обучение и спомага за по-бързо усвояване на оптималната политика на поведение. Адаптивната промяна на скоростта съобразно напредъка на обучението на агента повишава устойчивостта на процеса и намалява необходимото време за достигане на оптимално поведение. Използването на по-ефективен механизъм за съхранение и използване на натрупания опит повишава обобщителните способности на модела, като успеваемостта при тестовите епизоди достига 95%. Също така, подобрената версия на модела демонстрира по-бърза скорост на обучение с 32%, по-високо средно възнаграждение с 55%, по-добра точност с 5% и по-устойчиво поведение по време на обучението спрямо базовия модел. Освен това анализът на ентропията и функциите на загубите на актьора и критика доказват по-ранното формиране на стабилна политика. Критикът във всички разглеждани случаи успешно апроксимира функцията на стойността, като при подобрения модел това се случва за по-малък брой итерации.

От концептуална гледна точка предимството на предложената методика спрямо утвърдените DRL навигационни подходи се състои в нейната систематичност и възможности за интегриране и надграждане. Предложена е комбинация от няколко целенасочени подобрения, които адресират различни аспекти на процеса на обучение – скорост, устойчивост и ефикасност при използване на данните. В сравнение с други DRL подходи, които често изискват сложно настройване за постигане на добри резултати, предложената методика запазва стабилността и простотата на алгоритъма PPO, като същевременно преодолява неговите ограничения, свързани със скоростта на сходимост и еднотипно управление на обучителния опит. Проведените експерименти доказват, че прилагането на тези подобрения води до по-бързо формиране на стабилна навигационна политика. Също така описаният комплексен подход демонстрира, че подобрения в навигационната ефективност могат да бъдат постигнати не непременно чрез замяна на алгоритъма, а също така чрез интелигентно адаптиране и комбиниране на архитектурни и обучителни механизми в рамките на утвърдена парадигма за DRL.

Постигнатите резултати в симулация представляват основа за последващото интегриране и тестване на модела в реална физическа среда, описано в глава 4, където също така се извършва оценка на неговите обобщаващи способности и устойчивост в реални условия.

Глава 4. Експериментални изследвания и анализ на приложимостта на получения модел за изпълнение на задачата за навигация на АМР

4.1 Техническо описание на използваното оборудване

За целите на експерименталните изследвания е използвана роботизирана платформа, разработена въз основата на междинния интеграционен софтуер с отворен код ROS2. Като основен контролер се използва RDK X3, чиито изчислителни ресурси осигуряват обработване на данни в реално време и изпълнение на сложни изчислителни задачи. Платформата е оборудвана с високопроизводителни хардуерни устройства, колела Mecanum за реализиране на сложни движения, LiDAR за измерване на разстоянието до обекта (time of flight, ToF) и картографиране, 3D камера (Табл. 4).

Табл. 4. Общи параметри на хардуера

централен процесор	ARM Cortex-A53, 4 ядра по 1.2 GHz
операционна система	Ubuntu 20.04 и ROS-Foxy
сензори	LiDAR MS200, камера CSI, 3D камера
захранване	постояннотоково, 7.4 V
живот на батерията	3.5 часа
радиоуправление	джойстик, клавиатура, мобилен телефон
комуникация	локална мрежа (LAN), точка на достъп (WiFi)
материал на шасито	алуминиева сплав
безопасност	защита срещу свързване, защита срещу късо съединение, защита срещу блокиране на роторите
външни размери	236.11 x 181.10 x 184.9 мм
маса	1.93 кг

4.2 Конфигуриране на софтуера с ROS2 съобразно задачата за навигация

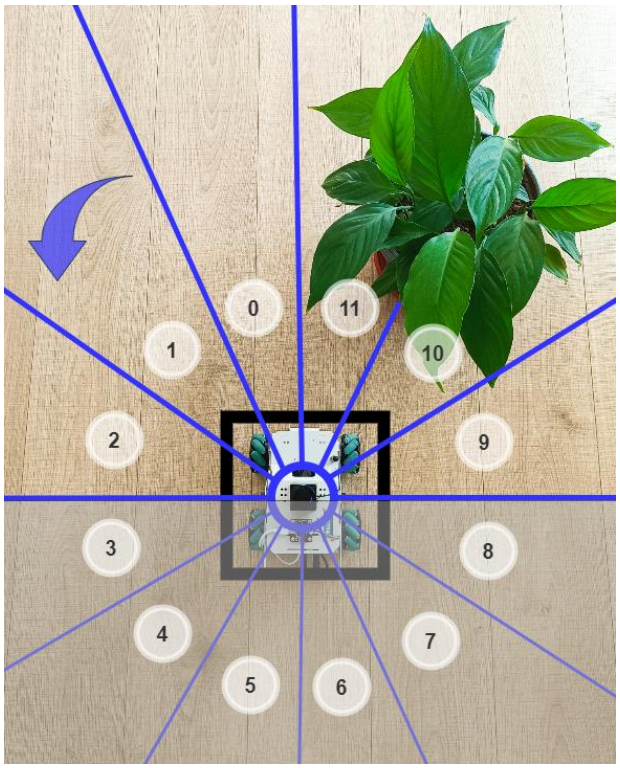
За целите на експерименталното изследване в платформата Yahboom RDK X3 е инсталиран следният софтуер:

- операционна система Ubuntu 20.04.6 LTS;
- междинен интеграционен софтуерен пакет за роботи ROS2 Foxy;
- Python 3.8.10 със следните библиотеки:
  - PyTorch 2.4.1 – за дълбоко обучение;
  - NumPy 1.24.3, math, squaternion – за математически изчисления;
- rclpy – клиентската библиотека на Python за ROS 2;
- ros-foxy-geometry-msgs, ros-foxy-nav_msgs, ros-foxy-sensor_msgs, ros-foxy-visualization_msgs – пакети за съобщения на ROS2 Foxy.

За свързване с робота е използван софтуер за отдалечен достъп за настолни компютри и мобилни устройства RealVNC.

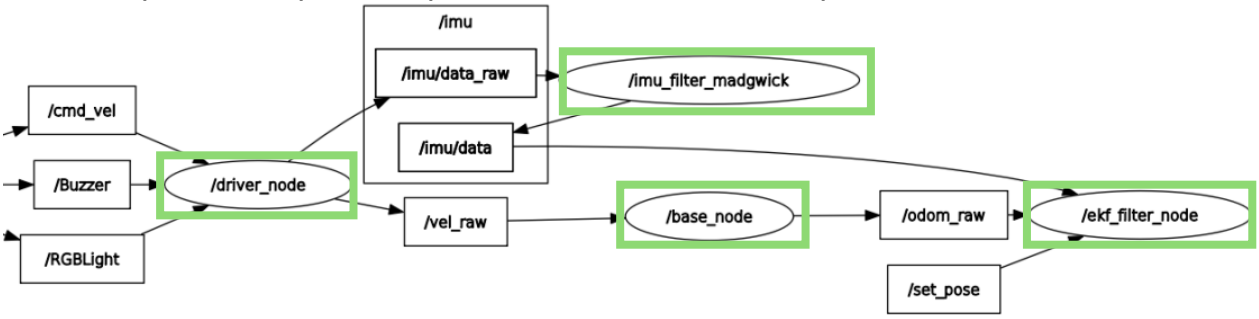
Сензорът LiDAR MS200 е конфигуриран в диапазон на отчитания от 0° до 360° с въртене в посока, обратна на часовниковата стрелка, и с разстояние от 0.05 до 20 м. Избраната честота на предаване на сигнала е 10 Hz. Показанията на LiDAR се публикуват в темата MS200/scan. Схематичното представяне на реализацията е показано на Фиг. 10. Едно отчитане в диапазона до 360° се разделя на 12 еднакви по размер зони с ъгъл на видимост от 30°, като се взема минималното разстояние до препятствието за всяка от 6-те зони с индекси [0, 1, 2, 9, 10, 11]. По този начин показанията на LiDAR представляват вектор от 6

променливи с плаващ десетичен разделител, описващи минималните разстояния до препятствията, равномерно разпределени по фронталната част на робота с ъглов диапазон от 180°.



Фиг. 10. Конфигурация на отчитанията от LiDAR като входен параметър на алгоритъма

Данните от одометрията се получават чрез комбиниране на imu данните, прочетени от ROS2, и данните за скоростта. На Фиг. 11 е показана реализация на локализирането на робота чрез използване на одометрия.

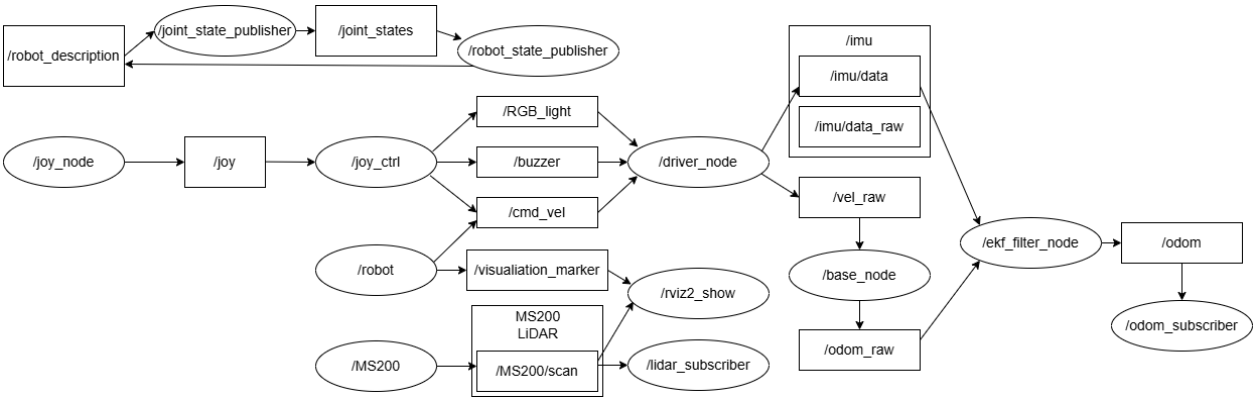


Фиг. 11. Част от архитектурата по предаване на съобщения за целите на локализация

На всяка стъпка със зададени линейна и ъглова скорост мобилният робот конфигурира входните данни за алгоритъма по начина, описан във формула (11):

S_t = 6 отчитания от LiDAR + разстояние до целта + ориентация спрямо целта(1)

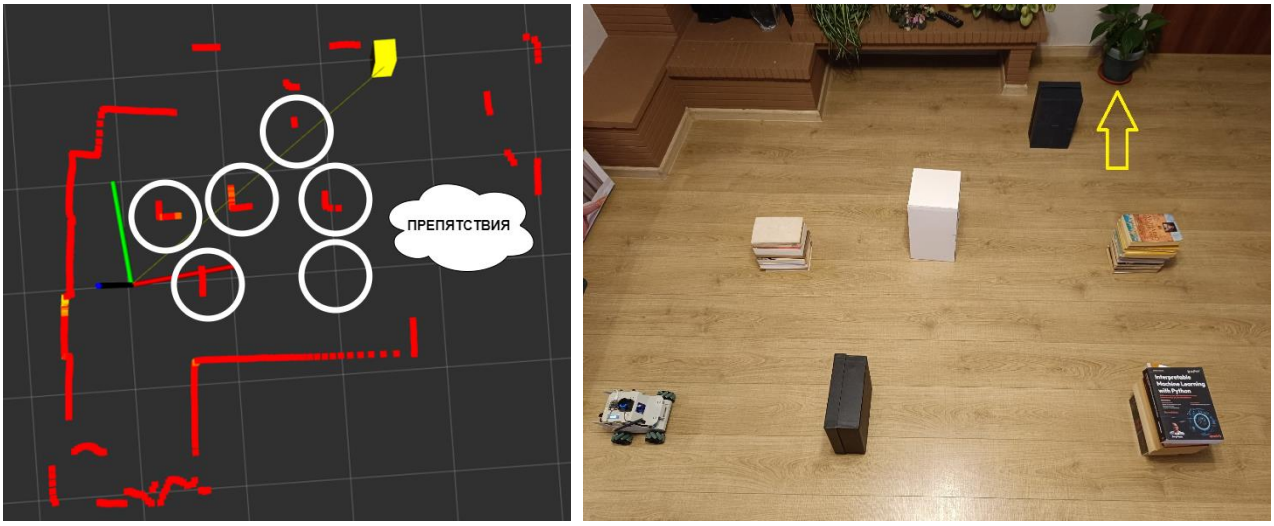
На Фиг. 12 е показана цялостната изградена разпределена архитектура в ROS 2, която включва отчитането на показанията на сензора (LiDAR), контролера за движение и алгоритмите за локализиране заедно с трансформациите, реализирани за поставената задача по навигация.



Фиг. 12. Реализирана архитектура за комуникация в ROS2

4.3 Експериментално изследване на приложимостта на алгоритмите с RL в реална среда

Опитната постановка в реална среда представлява затворено пространство с неправилна форма с размери 4 x 3.8 м и разположени различни по форма и размер предмети (препятствия) в нея. AMP е разположен в единия край на затвореното пространство, а целта е на разстояние 3.15 м от него. Опитните резултати се събират при наличие на 0, 1, 2, 3, 4, 5 и 6 препятствия с цел анализ на представянето на базовия и подобрения модел, получени в глава 3. С цел сравнителен анализ на обучените модели позициите на AMP и целта остават постоянни. На Фиг. 13 е показана опитна постановка при наличие на максимален брой от 6 препятствия и реална снимка на постановката.



а) б)

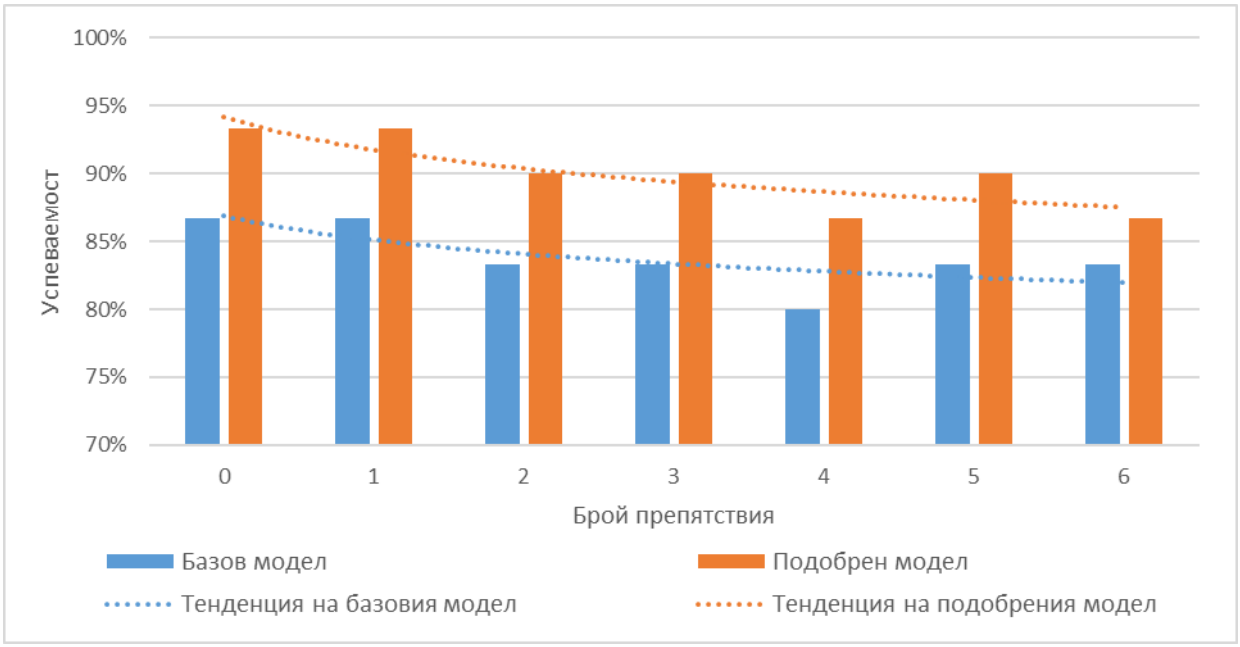
Фиг. 13. Визуализация на опитната постановка в Rviz (а) и реална среда (б) с 6 препятствия

Основните показатели, събирани по време на опитите, са булева стойност за успешно завършване на епизода, минимален и среден брой направени стъпки за всеки епизод и крайно състояние на агента при приключване на епизода. В Табл. 5 са показани количествени характеристики при анализа на двата модела.

Табл. 5. Експериментални резултати в реална среда

Модел	Бр. препятствия	Успешни епизоди	Успеваемост	Брой стъпки за епизод при успешно завършени епизоди		
				Ср.	Мин.	Макс.
Базов	0	26	87%	68	59	85
Базов	1	26	87%	69	63	86
Базов	2	25	83%	76	68	88
Базов	3	25	83%	78	73	92
Базов	4	24	80%	80	73	96
Базов	5	25	83%	83	76	98
Базов	6	25	83%	89	76	98
Подобрен	0	28	93%	66	58	83
Подобрен	1	28	93%	66	61	87
Подобрен	2	27	90%	74	65	88
Подобрен	3	27	90%	78	70	93
Подобрен	4	26	87%	78	68	95
Подобрен	5	27	90%	83	75	98
Подобрен	6	26	87%	85	76	99

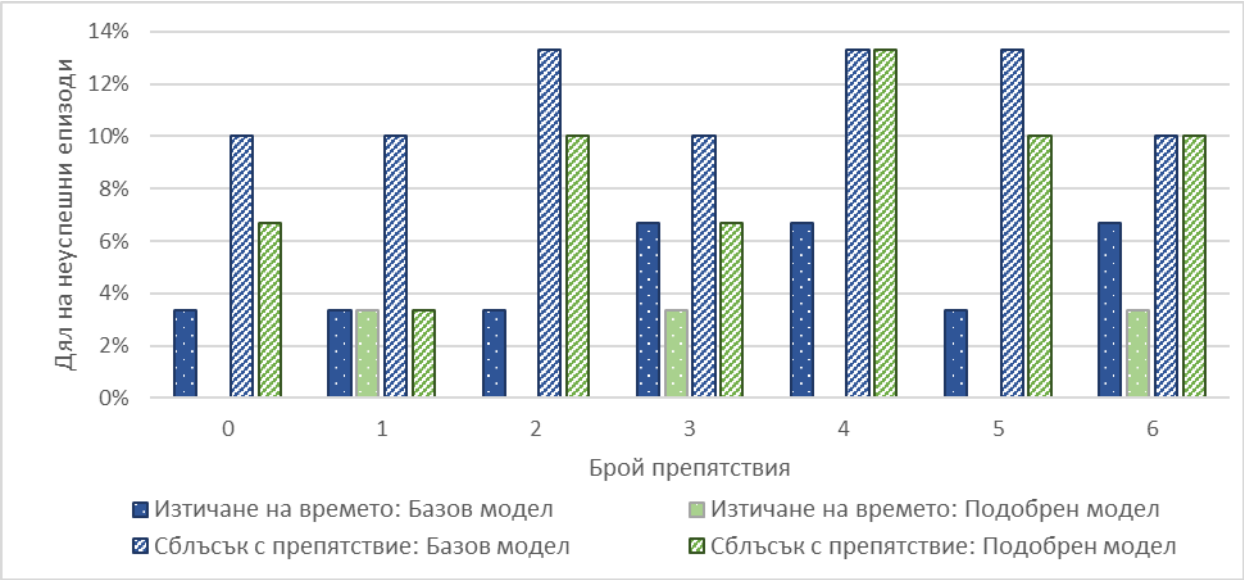
Най-надеждният показател за устойчивостта на обучените модели към шумове и грешки в сензорните показания, както и за тяхната обобщаваща способност, е успеваемостта им при използване за изпълнение на поставената задачата. На Фиг. 14 са показани получените резултати при така описаната опитна постановка с 0, 1, 2, 3, 4, 5 и 6 препятствия.



Фиг. 14. Сравнение на успеваемостта на моделите

И при двата модела се наблюдава висока и относително стабилна успеваемост при всички експериментални постановки. Подобреният модел превъзхожда базовия при всеки брой препятствия, като разликата в успеваемостта варира между 4% и 7% в полза на подобрения модел.

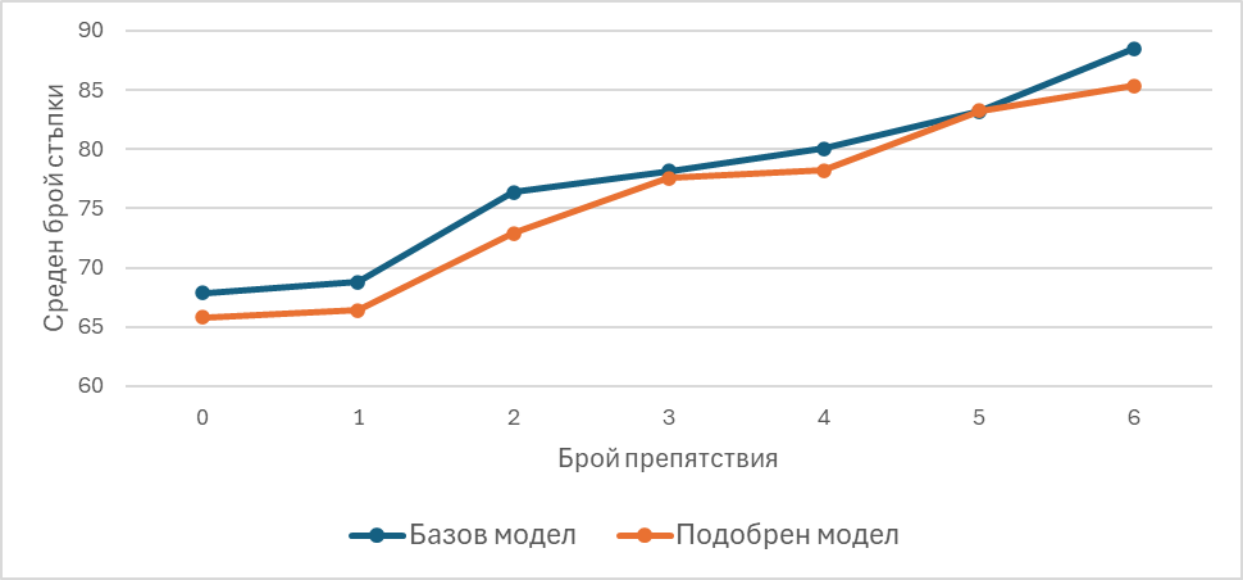
Резултативните състояния на AMP при неуспешно завършените епизоди, крайното състояние на които е сблъсък с препятствие или изтичане на времето (при гранична стойност от 100 стъпки), са визуализирани на Фиг. 15.



Фиг. 15. Сравнителен анализ на крайните състояния при неуспешно завършените епизоди

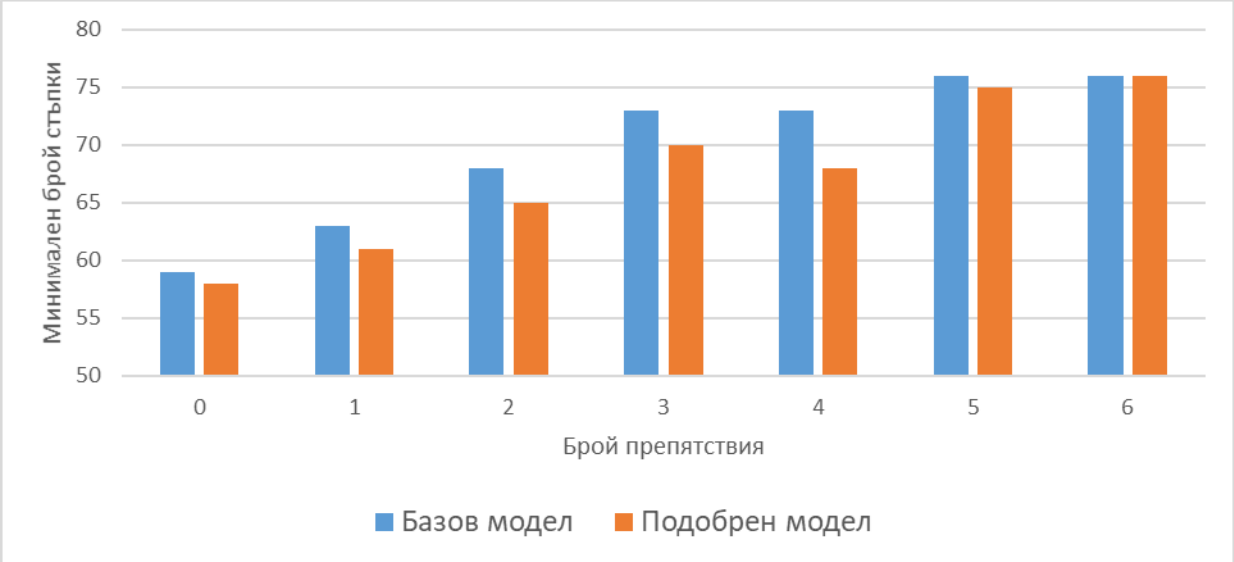
Сблъсъците представляват основната причина за неуспешни епизоди и при двата модела. За базовия модел честотата на сблъсъци варира между 10% и 13%, като не се наблюдава ясно изразена монотонна зависимост спрямо броя препятствия. При подобрения модел честотата на сблъсъци е по-ниска от тази на базовия модел във всички постановки освен при 4 и 6 препятствия, където двата модела имат еднаква стойност (13%).

От съществено значение за определяне на оптималността е да бъдат разгледани траекториите на движение на AMP при експериментите с всеки от двата модела. За целта се сравнява средният брой стъпки при опити с различен брой препятствия, както и минималният брой стъпки за достигане до целта. На Фиг. 16 са показани разликите между средния брой стъпки за успешно завършване на епизодите при двата модела.



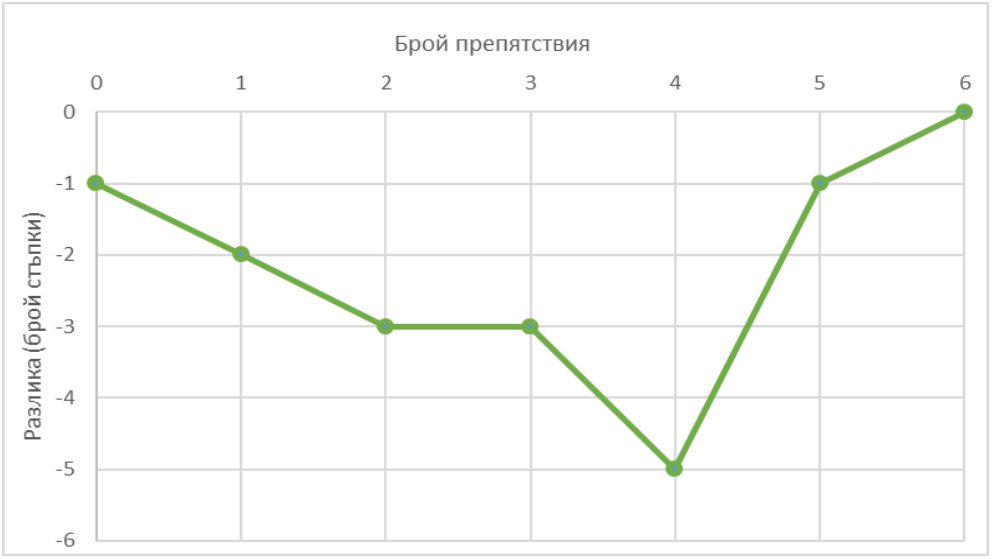
Фиг. 16. Сравнение на средния брой стъпки при двата модела

При добавяне на препятствия пътят на придвижване става по-дълъг – средно от 67 до 87 стъпки. Видно е, че пътят при подобрения модел е по-кратък с около 2 стъпки при всяка една от постановките. Разликата не се откроява единствено в постановката с 5 препятствия, в която средният брой стъпки съставя 83 и при двата модела. От гледна точка на оптималността на траекториите за всяка постановка, на Фиг. 17 е показан минималният брой стъпки при различен брой препятствия. Разликата между експериментите с 0 и 1 препятствие тук се открояват по-ясно – броят стъпки в най-кратките траектории се увеличават с нарастване на препятствията в пространството. Размерът на траекториите за достигане до целта нараства постъпателно от около 58 при 0 препятствия до 76 при 6 препятствия.



Фиг. 17. Сравнение на минималния брой стъпки

Открояват се по-големи нарастванията на минималните траектории при 2 и 5 препятствия, като разликата се състои в около 5 допълнителни стъпки. Тази разлика на минималния брой стъпки между двата модела при всяка постановка е показана на Фиг. 18.



Фиг. 18. Разлика на минималния брой стъпки при сравняване на моделите

Траекториите на подобрения модел обикновено са с 1 и повече стъпки по-кратки тези на от агента, чиито действия се формират от базовия модел. Изключение прави само постановката с 6 препятствия, където показанията са еднакви. От гореизложеното може да се направи изводът, че при увеличаване на броя препятствия ефективността на базовия модел намалява по-рязко, докато тази на подобрения модел запазва устойчивостта си и показва значително по-слаби колебания в резултатите.

Количествената оценка на трансфера на моделите от симулация към реална среда се базира на резултатите от проведените експерименти, които са обобщени в Табл. 5. Основната метрика за функционална приложимост е успеваемостта – относителният дял на епизодите, при които AMP достига целта без сблъсък и в рамките на определеното времево ограничение. В Табл. 6 е представено сравнение на получените резултати в симулационна и физическа среда на базовия и подобрения модел.

Табл. 6. Сравнение на точността на моделите в симулация и реална среда

Модел	Среда	Средна успеваемост
Базов	Симулация	90%
	Реална среда	84%
Подобрен	Симулация	95%
	Реална среда	90%

Резултатите показват, че и двата модела запазват висока успеваемост при експериментите при директен пренос от симулация към реална среда без допълнително обучение. Точността на моделите спада в реална среда с 6% и 5% съответно при базовия и подобрения модел. Въпреки това подобреният модел запазва по-висока абсолютна успеваемост в реална среда спрямо базовия, което потвърждава по-добрата му обобщаваща способност.

4.4 Изводи

В настоящата глава е реализирано практическо пренасяне на обучените модели от симулационната 3D среда във физическа постановка с участие на AMP Yahboom RDK X3. Експериментите показват, че за успешен преход на

виртуалната опитна постановка в реална такава са необходими следните стъпки, действия и инструменти:

- Прецизно конфигуриране на входните данни на модела в реална среда с цел повишаване на способността за обобщаване от наблюдавани състояния в симулационна среда до ненаблюдавани в реална среда.
- Постигане на висока точност на окончателния модел чрез обучение с подсилване в симулационна среда, която трябва да отговаря на параметрите на реалната такава.
- Оптимизиране на конфигурацията на архитектурата на AMP в ROS2 чрез създаване на възли абонати и издатели към необходимите за целта теми в установен формат на съответните съобщения.
- Необходимост от използване на помощни приставки в ROS2, като Rviz за визуализиране на пространството от гледна точка на робота и Rqt за представяне на цялостната архитектура, които едновременно спомагат за визуализация на цялостната система и откриване на грешки, неточности и несъответствия в нея.

Възникналите трудности при пренасяне от симулационна в реална среда са свързани най-вече с грешки на сигналите, получавани от LiDAR и одометрията. В този смисъл при експериментите се установява, че отчитанията на LiDAR съдържат неправилни нулеви стойности и грешки в отчитането на разстоянията до препятствия, което налага тяхната допълнителна обработка и конфигуриране преди получаване на крайното състояние. Управлението на AMP също така изисква периодично калибриране на ъгловата и линейната скорост с цел намаляване на грешките в управлението. Разработената в рамките на настоящия дисертационен труд софтуерна архитектура в ROS2 осигурява надеждна комуникация между възлите за сензорни данни, локализация, управление и визуализация, което позволява обработка и реализиране на адаптивно управление на робота в реално време.

Проведени са експерименти с различен брой препятствия и е изградена методика за анализ на базовия и подобрения модел чрез редица метрики, в т.ч.: успех при достигане до целта, крайно състояние на всеки епизод и оптималност на получените траектории. Експерименталните резултати ясно открояват по-ефективното управление на AMP чрез използването на модел, обучен до по-висока точност (95%) в симулацията, отколкото при базовия модел, обучен с точност 90%. В реална среда подобреният модел има по-висока обща точност спрямо базовия: 90% спрямо 84%, както и по-кратки траектории на движение, измерени чрез средния брой стъпки в 180 експериментални епизода (78 при подобрения и 76 при базовия).

Проведените експерименти с физическия AMP потвърждават възможността за успешно прилагане на навигационна политика, базирана върху подхода DRL, обучена в симулационна среда, без използване на предварително съставени карти и GPS позициониране. Наблюдаваното отклонение на точността на моделите при пренос в реална среда е -5% и представлява умерено понижение в ефективността спрямо симулационните резултати, което е очаквано предвид наличието на сензорен шум, неточно моделирана динамика и акумулиращи се одометрични грешки. Въпреки това количественият анализ показва, че тази разлика е ограничена и не възпрепятства практическата приложимост на предложеното решение.

Получените резултати показват, че предложената система може да бъде използвана като основа за разработване на интелигентни навигационни модули в роботизирани платформи, работещи в условия на динамична среда и ограничена

сензорна информация. Изследваният подход създава предпоставки за бъдещо разширяване към многоагентни системи, интеграция с допълнителни сензори и адаптиране към по-сложни навигационни сценарии.

НАУЧНО-ПРИЛОЖНИ И ПРИЛОЖНИ ПРИНОСИ

Научно-приложни приноси:

1. Разработен е оригинален модел за автономна навигация на мобилен робот без използване на предварително изградени карти и GPS позициониране, базиран на локално възприемане на средата чрез дълбоко обучение с подсилване, който позволява формиране на устойчива навигационна политика в динамична и частично наблюдаема среда.
2. Формулиран и експериментално валидиран е модел на зависимостта между входните сензорни параметри, структурата на функцията на възнаграждение и характеристиките на научената навигационна политика, като е доказано тяхното ключово влияние върху безопасността, плавността и стабилността на движението.
3. Извършен е задълбочен анализ на сходимостта и робастността на навигационен модел, базиран върху метода PPO, включващ количествена оценка на влиянието на ключови хиперпараметри върху стабилността и ефективността на обучението в условия на стохастичност и сензорен шум.
4. Предложен е модифициран механизъм за ефективно използване на буфер на опит, който оптимизира разпределението на обучителните извадки, ускорява сходимостта и повишава точността на научената политика, което е потвърдено чрез сравнителни експериментални резултати.
5. Разработен е адаптивен механизъм за динамично регулиране на скоростта на обучение, който намалява колебанията в постигането на подобрения и повишава стабилността на базирания върху PPO алгоритъм при сложни навигационни сценарии.

Приложни приноси:

1. Разработена е интегрирана симулационна среда (Flatland/Gazebo) за изследване на навигацията чрез обучение с подсилване посредством реалистичен модел на мобилна платформа.
2. Реализирана е експериментална система за автономна навигация с ограничен набор сензори, приложима в индустриални затворени пространства.
3. Изградена е реална експериментална установка за валидиране на навигационни политики чрез физически робот.

4. Успешно е осъществен и експериментално валидиран пренос на обучен в симулационна среда модел към физическа роботизирана платформа, като е демонстрирано запазване на навигационните характеристики и устойчивост на политиката при реални физически ограничения и шумове.

5. Разработени са метрики за оценка на безопасността и обобщаемостта на модели, базирани на обучение с подсилване, в реална среда.

6. Реализирана е система за управление въз основа на ROS2, демонстрираща възможност за намаляване на хардуерната сложност чрез използване на бюджетни сензори.

СПИСЪК НА ПУБЛИКАЦИИТЕ ПО ДИСЕРТАЦИОННИЯ ТРУД

1. A. Slavova, V. Hristov, “Mapless Navigation with Deep Reinforcement Learning in Indoor Environment,” *International Scientific Conference “TechSys 2025” – ENGINEERING, TECHNOLOGIES AND SYSTEMS*, Plovdiv, Bulgaria, July 2025, DOI: 10.3390/engproc2025100063, http://techsys.tu-plovdiv.bg/ – **Scopus**.
2. A. Slavova and V. Hristov, "Policy Interpretation for Deep Reinforcement Learning", *2025 International Conference Automatics, Robotics and Artificial Intelligence (ICARAI)*, Sozopol, Bulgaria, 2025, pp. 1-4, DOI: 10.1109/ICARAI67046.2025.11137898 – **Scopus**.
3. A. Slavova and V. Hristov, "Application of Reinforcement Learning in Autonomous Mobile Robots", *2024 32nd National Conference with International Participation (TELECOM)*, Sofia, Bulgaria, 2024, pp. 1-4, DOI: 10.1109/TELECOM63374.2024.10812227 – **Scopus**.
4. D. Slavov, V. Hristov and A. Slavova, "Distributed Machine Learning through Transceiver Competitive Connectivity of Remote Computing Systems", *2023 International Scientific Conference on Computer Science (COMSCI)*, Sozopol, Bulgaria, 2023, pp. 1-7, DOI: 10.1109/COMSCI59259.2023.10315948 – **Scopus**.
5. A. Slavova, D. Slavov and V. Hristov, " Research on Computer Vision models for Deep Learning in Autonomous Mobile Robots", *2024 International Conference Automatics, Robotics and Artificial Intelligence (ICARAI)*, Sozopol, Bulgaria, 2024, DOI: 10.1088/1757-899X/1317/1/012011.
6. A. Slavova and D.Slavov, " Task Execution and Dynamic Re-Planning with a Mobile Robot and Manipulator: A Real-Robot Study Using RDK X3 and myCobot 320 – Part 1", *2025 33rd National Conference with International Participation (TELECOM)*, Sofia, Bulgaria, 2025 – **IEEE**.
7. A. Slavova and D.Slavov, " Task Execution and Dynamic Re-Planning with a Mobile Robot and Manipulator: A Real-Robot Study Using RDK X3 and myCobot 320 – Part 2", *2025 33rd National Conference with International Participation (TELECOM)*, Sofia, Bulgaria, 2025 – **IEEE**.

DEEP LEARNING SYSTEMS FOR AUTONOMOUS MOBILE ROBOTS

Anastasiya Slavova

Abstract

This PhD thesis elaborates on the research, design, and implementation of reinforcement learning algorithms for the navigation of autonomous mobile robots.

Chapter 1 provides a comprehensive introduction to the application of deep learning, particularly deep reinforcement learning (DRL), in autonomous mobile robots (AMRs). The chapter highlights the advantages of navigation without pre-built maps or GPS, showing how reinforcement learning allows AMRs to adapt to unknown and dynamic settings using limited sensor input. Several reinforcement learning approaches for AMR navigation are analyzed based on practical and technical criteria such as environmental complexity, learning stability, computational requirements, convergence speed, experience efficiency, robustness, hyperparameter sensitivity, and task applicability. The chapter concludes with a critical analysis of DRL methods, noting strengths such as adaptive behavior without prior maps, and limitations including long training times, high computational costs, sensitivity to hyperparameters, inefficient use of experience, and challenges in transferring models from simulation to real-world environments.

Chapter 2 presents the design and evaluation of reinforcement learning models for autonomous navigation of a wheeled mobile robot in a 2D Flatland simulation integrated with ROS2. The robot, equipped with a LiDAR sensor, is trained to reach a target safely and efficiently, guided by a carefully designed reward function. Four algorithms—DQN, A2C, TRPO, and PPO—are implemented, and their performance is compared using various metrics. The results show a clear progression in performance from value-based methods (DQN, A2C) to policy-based methods (TRPO, PPO), with PPO achieving the best balance of convergence speed, training stability, and navigation quality.

Chapter 3 describes the implementation and training of PPO navigation model for the Yahboom RDK X3 four-wheeled mobile robot in a 3D Gazebo simulation integrated with ROS2. The chapter covers setting up the 3D environment, creating accurate robot and obstacle models, configuring sensors. The PPO algorithm is implemented, with extensive fine-tuning of hyperparameters, neural network architecture, and learning mechanisms. Key improvements include actor network residual connections, adaptive learning rate adjustments based on agent progress, and a memory buffer system that prioritizes recent and successful experiences. These enhancements significantly increase training speed, stability, and performance, raising test episode success rates from 90% to 95% and improving cumulative rewards and convergence speed.

In chapter 4 experimental tests were conducted in a real environment with varying numbers of obstacles (0–6) to evaluate the performance of both the baseline and improved models. Key metrics included episode success rate, trajectory optimality, and number of steps to reach the goal. Results showed that the improved model consistently outperformed the baseline, achieving higher success rates (90% vs. 84%), shorter trajectories, and better robustness to sensor noise. The experiments confirmed that policies trained in simulation can be effectively transferred to real-world scenarios without pre-built maps or GPS, with only minor performance drops due to real-world uncertainties.

TECHNICAL UNIVERSITY OF SOFIA

Faculty of Automatics

Department of Electrical Motion Automation Systems

Anastasiya Vladimirovna Slavova, MEng

DEEP LEARNING SYSTEMS IN AUTONOMOUS MOBILE ROBOTS

A B S T R A C T

of a dissertation for obtaining an educational and scientific degree

Doctor of Philosophy

Area: 5. Technical sciences

Professional field: 5.2. Electrical engineering, electronics and automation

Scientific specialty: Artificial intelligence systems

Supervisor: Assoc. Prof. Vladimir D. Hristov

SOFIA, 2026

The dissertation thesis is discussed and directed for defense by the Council of the Department of Electrical Motion Automation Systems, Faculty of Automatics, Technical University of Sofia on a regular session held on 25.02.2026.

The public defense of the dissertation thesis will take place at 13:00 hours on 06.07.2026 in the Conference room of LIC, Technical University of Sofia at an open session of the scientific jury determined by Order ОЖ-5.2-27 / 12.03.2026 of the Rector of Technical University of Sofia with the following members:

1. Assoc. Prof. Marin Milkov Zhilevski, PhD, Eng. – chairman
2. Prof. Miho Rachev Mihov, PhD, Eng. – scientific secretary
3. Assoc. Prof. Nikola Georgiev Shakev, PhD, Eng.
4. Prof. Alexandra Ivanova Grancharova, PhD, Eng.
5. Assoc. Prof. Denis Safidinov Chikurtev, PhD, Eng.

Reviewers:

1. Assoc. Prof. Marin Milkov Zhilevski, PhD, Eng.
2. Assoc. Prof. Denis Safidinov Chikurtev, PhD, Eng.

The thesis defense documents are available to those interested in the office of the Faculty of Automatics, Technical University of Sofia, block 2, room 2340.

The doctoral candidate is an extramural PhD student at the Department of Electrical Motion Automation Systems, Faculty of Automatics. The research for the dissertation was conducted by the author, with some parts supported by research projects.

Author: Anastasiya Vladimirovna Slavova, Meng
Title: DEEP LEARNING SYSTEMS
IN AUTONOMOUS MOBILE ROBOTS
Print circulation: 30
Printed by the Publishing House of Technical University of Sofia

I. GENERAL CHARACTERISTICS OF THE DISSERTATION THESIS

Relevance of the problem

The relevance of the problem addressed in the dissertation is determined by the rapid development of autonomous mobile robots (AMRs) and their increasing role in industry, logistics, service activities, and operations in hazardous environments. Modern manufacturing and warehouse systems require a high degree of automation, flexibility, and safety, which places increased demands on the control and navigation methods of AMRs. Traditional approaches based on pre-built maps and classical path-planning algorithms demonstrate limitations when operating in dynamic and partially observable environments, where real-time adaptation and robustness to uncertainty in sensor data are required.

In this context, deep reinforcement learning has established itself as a promising tool for achieving autonomous navigation without pre-built maps and without reliance on GPS infrastructure. Despite significant progress in the field, unresolved issues remain related to training stability and speed, the efficiency of utilizing accumulated experience, the selection of appropriate architectures and hyperparameters, as well as the transfer of trained models from simulation to real-world environments. These problems have both theoretical and clearly defined practical significance.

The study gains additional relevance from the need to develop resource-efficient solutions based on a limited set of sensors, such as LiDAR and odometry, in order to reduce hardware complexity and deployment costs. The development and experimental verification of a model for autonomous navigation in a real environment—using a minimal sensor configuration and without relying on maps—constitute a significant contribution to the advancement of intelligent, affordable, and practically applicable autonomous mobile systems.

Purpose of the dissertation thesis, main tasks and research methods

The purpose of this dissertation is to develop a control system for autonomous mobile robots operating under uncertainty, based on deep reinforcement learning, which ensures safe operation and demonstrates strong generalization capability when using a limited set of sensor data.

The tasks for the thesis are:

1. Investigation and comparative analysis of the application of navigation models based on the reinforcement learning (RL) approach, including their algorithmic features, advantages, and limitations.

2. Selection, training, and experimental evaluation of an algorithm for autonomous navigation in accordance with the defined criteria.

3. Investigation of the convergence of the proposed approach in navigation tasks.

4. Experimental evaluation of the applicability of the proposed models in a real environment involving an autonomous mobile robot.

5. Systematic analysis of the robustness of the models when transferred from simulation to a real physical environment.

Scientific novelty

The integration of architectural and algorithmic improvements in reinforcement learning (a modified neural architecture, adaptive learning rate, and optimized use of an experience replay buffer) leads to a statistically significant improvement in the convergence, stability, and navigation efficiency of a wheeled autonomous mobile robot compared to the baseline implementation of the algorithm, both in a simulation environment and in a real physical setup.

Practical applicability

The practical applicability of the system developed in the dissertation lies in the possibility of implementing autonomous navigation for mobile robots in real industrial and service environments—such as warehouses, production halls, logistics centers, and indoor spaces with complex layouts—without the need for pre-built maps or expensive sensor infrastructure. The proposed approach, based on deep reinforcement learning and the use of a limited set of sensors (LiDAR and odometry), enables the development of more affordable, adaptive, and cost-effective robotic solutions.

Approbation

The results were validated through stepwise experimental verification in both simulation and real environments. Initially, the proposed algorithms were trained and comparatively analyzed in a two-dimensional environment using the Flatland simulator. The most suitable model for the given task was selected, and its training was implemented in a three-dimensional simulation environment using ROS2 and Gazebo, where the influence of hyperparameters, neural network architecture, and reward function on algorithm convergence and efficiency was also investigated. Algorithmic and architectural improvements were made to the baseline model. Subsequently, both the baseline and improved models were deployed and tested on a real robotic platform, Yahboom RDK X3, with a series of experiments conducted in an enclosed space featuring different obstacle configurations, evaluated using metrics such as success rate in reaching the goal, trajectory optimality, and robustness in transfer from simulation to real environment.

Publications

The main achievements and results of the dissertation are published in 7 scientific papers of which 4 are indexed in Scopus, 2 in IEEE while 1 are of single authorship.

Structure and volume of the dissertation

The dissertation thesis has a volume of 151 pages, including an introduction, 4 chapters for solving the formulated main problems, a list of key contributions, a list of dissertation publications, a list of citations to dissertation publications, and references. A total of 105 literature sources are cited, 80 of which are in Latin script, while the remainder are internet sources. The thesis includes 80 figures and 23 tables. The figure and table numbering in the abstract corresponds to the thesis numbering.

II. CONTENTS OF THE DISSERTATION THESIS

Chapter 1. Deep machine learning in autonomous mobile robots – application, current state, and challenges

1.1 Application of deep machine learning in autonomous mobile robots

A review of the application of deep machine learning in autonomous mobile robots was conducted, describing the main methods used — convolutional neural networks, recurrent neural networks, and reinforcement learning for decision-making, control, and navigation. A classification of autonomous mobile robots according to their operating environment is presented — ground, aerial, and underwater — with emphasis on the integration of sensor systems and control algorithms based on deep machine learning, enabling autonomous operation in complex and dynamic environments.

1.2. Advantages of navigation without pre-built maps and GPS, based on reinforcement learning

The integration of reinforcement learning significantly expands the capabilities of mobile robots, transforming them from executors of simple repetitive tasks into intelligent, autonomous, and collaborative systems. Autonomous navigation in dynamic and unknown environments is particularly promising, where traditional methods such as SLAM and path planning algorithms may prove ineffective. Reinforcement learning enables navigation without pre-built maps or GPS, allowing the agent to dynamically adapt to obstacles and environmental changes using a limited set of sensors and local observations. This approach ensures adaptability, robustness to noise, and the ability to utilize sensory data under conditions of uncertainty, reduces dependence on external infrastructure and hardware complexity, and provides a flexible and resource-efficient alternative to conventional navigation methods, highlighting its significance for scientific and practical applications.

1.3. Reinforcement learning – definition, advantages, and training strategies

Reinforcement learning enables the autonomous acquisition of optimal behavior through iterative interaction with the environment. The approach is based on creating agents that learn from the environment by interacting with it through trial and error and receiving rewards (positive or negative) as a unique form of feedback. Emphasis is placed on three main training strategies: direct, indirect, and hybrid approaches.

1.4. Construction of the reward function

The importance of the reward function for training deep reinforcement learning agents in the navigation of autonomous mobile robots is presented, describing the fundamental principles for its design — rewards for approaching the target, reaching the goal position, and finding a shorter path, as well as penalties for collisions with obstacles and exceeding time limits for task execution. Additional techniques are outlined, such as goal-oriented orientation, time efficiency in reaching the target, and movement smoothness, along with the use of human feedback to train desired behavior.

1.5. Reinforcement learning approaches for navigation in autonomous mobile robots

Several promising reinforcement learning approaches are considered. The analysis of the presented models uses a set of technical and practical criteria reflecting the constraints of the real system, environment, and task objectives, such as:

1. Complexity and dynamics of the environment, including partial observability and dynamic objects, and sensor noise (LiDAR, odometry);
2. Training stability, describing the algorithm's sensitivity to unstable parameter updates;
3. Computational resources required for algorithm training;
4. Convergence speed (training time), accounting for the need for fast-converging algorithms in simulations requiring significant computational resources or in experiments with physical environments;
5. Efficiency of experience utilization, determining the number of interactions with the environment for each algorithm;
6. Robustness and safety;
7. Algorithm sensitivity to hyperparameter tuning;
8. Applicability to the given task – navigation in an environment with obstacles.

1.6. Generalized analysis of algorithms and challenges in using the DRL approach for navigation

The deep reinforcement learning (DRL) approach provides a powerful tool for autonomous navigation of mobile robots in complex and dynamic environments, enabling adaptive behavior without pre-existing maps or GPS. Its main limitations include long training times, high computational requirements, and sensitivity to hyperparameters. Policy-based algorithms (such as PPO and TRPO) demonstrate greater stability, efficiency, and applicability in real-world environments compared to value-based algorithms (such as DQN). However, several significant challenges remain unresolved: low efficiency in experience utilization and slow convergence in some algorithms; strong dependence on a specifically designed reward function; limited comparability between different studies; and insufficient research on transfer from simulation to real environments when using minimal sensor configurations and operating without pre-built maps.

Goal and tasks

Based on the above, the aim of the dissertation is to develop a control system for autonomous mobile robots operating under uncertainty, based on deep reinforcement learning, which ensures safe operation and possesses strong generalization capability when using a limited set of sensory data.

The following main tasks have been formulated: conducting a comparative analysis of suitable navigation algorithms; selection and implementation of an appropriate simulation environment; development and tuning of a reward function; optimization of the neural network architecture and hyperparameters; training and comparative evaluation of models in 2D and 3D simulation; experimental verification of the improved model in a real environment with analysis of its applicability and robustness.

Chapter 2. Selection of appropriate models for navigation of autonomous mobile robots

2.1 Description and implementation of the simulation environment

The implementation of reinforcement learning for autonomous navigation of a mobile wheeled robot in a two-dimensional simulation environment was carried out using Flatland with integration through ROS2. The simulator supports physics, partial observability, and first-person perception, enabling rapid prototyping and testing of

algorithms using Python libraries. The architecture of ROS2 is described, including nodes, topics, services, actions, and parameters, as well as tf2 for coordinate frame transformations. The agent is represented as a differential two-wheeled mobile robot equipped with a LiDAR sensor. The task of the robot is to find the fastest and safest path to the target within an enclosed space. The reward function is defined in such a way as to encourage quick and safe movement of the robot toward the goal.

2.2. Training and analysis of navigation models

The structures, neural network architectures, and hyperparameters used for training four reinforcement learning algorithms — DQN, A2C, TRPO, and PPO — are described. Training of each model is conducted in the same closed simulation environment containing obstacles. The following metrics are used to analyze model performance in solving the given task:

1. Cumulative reward per episode – measures the total accumulated reward for each episode and indicates the agent's progress in learning the optimal strategy.
2. Number of steps to reach the goal – evaluates navigation efficiency and task execution time, where a reduction in the number of steps indicates more efficient behavior.
3. Terminal states of episodes – classified as successfully completed episodes, episodes with collisions with obstacles, or episodes due to time expiration, thus evaluating the safety and reliability of training.
4. Success rate during testing – measures the accuracy and stability of the learned policy during test iterations (for example, 90% success for PPO).
5. Dispersion of cumulative reward – evaluates stability and consistency of results between episodes, where lower dispersion indicates a more reliable model.

2.3 Generalized analysis of the obtained results

A generalized statistical analysis of the algorithms is presented, considering convergence speed, obtained rewards, episode duration during training, as well as parametric evaluations for each algorithm (Table 12).

Table 12. Statistical analysis of obtained results

Indicator	DQN	A2C	TRPO	PPO	Statistical significance
Convergence speed (episodes)	2814	1712	676	392	-
Convergence improvement vs. previous algorithm	-	39%	61%	42%	$p < 0.01$
Successfully completed episodes (%)	10	20	80	90	-
Average reward	-180.8	-119.2	150.4	170.6	$p < 0.05$
Standard deviation of reward	45.4	42.3	352.8	362.1	-

Average steps per episode	17.3	187.2	127.9	108.3	$p < 0.05$
Standard deviation of steps	20.5	41.5	57.4	52.7	-

The results show a clear improvement in algorithm performance when transitioning from value-based and actor-critic approaches (DQN, A2C) to policy-based approaches (TRPO, PPO). The convergence speed increases progressively from DQN to PPO, with the number of required episodes decreasing by 86%, from 2814 for DQN to 392 for PPO. The success rate in episode completion rises sharply: from 10–20% for DQN and A2C to 80% for TRPO and 90% for PPO, demonstrating greater stability and reliability of policy-based approaches.

2.4 Conclusions

The comparative analysis of the DQN, A2C, TRPO, and PPO algorithms in a two-dimensional simulation conducted in this chapter allows the formulation of clearly substantiated scientific conclusions regarding their applicability to the task of autonomous navigation. The study shows that the value-based approach, DQN, demonstrates limited effectiveness, expressed in slow convergence, low success rates, and negative average reward values, making it unsuitable for solving complex navigation tasks in continuous and dynamic environments. Policy-based and actor-critic approaches (A2C, TRPO, and PPO) exhibit significantly better performance across all evaluated metrics. In particular, TRPO and PPO show substantial acceleration of training, a sharp increase in the percentage of successfully completed episodes, and significantly higher average reward values. This confirms the theoretical advantages of direct policy optimization for tasks requiring stable and consistent behavior.

It has been established that the PPO algorithm demonstrates the best balance between convergence speed, training stability, and the quality of the resulting navigation policy. Compared to TRPO, PPO achieves similar or better results with significantly lower algorithmic complexity and easier tuning, making it more suitable for practical implementation and further extension. Based on these scientific conclusions, the PPO algorithm has been selected as the baseline method for further research presented in the next chapter. This choice is motivated both by experimental results and by the possibility of effectively enhancing the algorithm through architectural and algorithmic improvements to increase the efficiency and robustness of autonomous navigation.

Alongside the algorithm selection, the analysis also justifies the use of a two-dimensional simulation environment for the purposes of the present study. Two-dimensional simulation allows effective modeling of the main challenges in navigation — obstacle avoidance, trajectory planning, and goal reaching — with significantly lower computational complexity compared to three-dimensional environments. This creates conditions for systematic experimentation, rapid model iteration, and objective comparison of different algorithmic configurations without loss of generalizability of the obtained results.

In the next chapter (Chapter 3), the implementation of the proven most effective PPO model in a three-dimensional simulation with ROS2 integration is described. The model's functionality is fully implemented without the use of the SB3 library to ensure maximum flexibility in modifying its structure and parameters. To achieve a realistic simulation as close as possible to the conditions, physical characteristics, and sensor data of the real device, a 3D robot model described via URDF is also used. This

supports the subsequent integration of the trained reinforcement learning model into a mobile four-wheel platform with mecanum wheels, Yahboom RDK X3, which has been selected for experimental validation in a real physical environment.

Chapter 3. Training the navigation model in a three-dimensional simulation environment

3.1 Description of the Gazebo 3D simulator and characteristics of the used software

A description of the Gazebo simulator for 3D simulation of robot training using reinforcement learning is provided, highlighting its capabilities for high-quality physical simulation, support for various sensors and actuators, integration with ROS2, and Python programming. The software components and libraries used for configuring and training the robot (Yahboom RDK X3 with mecanum wheels and LiDAR) are presented, including PyTorch, NumPy, TensorBoard, and rclpy, as well as the role of RViz for visualization and debugging.

3.2 Preparation of the simulation environment

The overall preparation and configuration of a 3D simulation environment for an autonomous mobile robot in Gazebo and ROS2 is described, including the following steps:

1. Robot visualization – creation of a 3D model of the Yahboom RDK X3 using URDF files, definition of links and joints, integration with RViz, and conversion to SDF for Gazebo.
2. Construction of a simulated enclosed environment – creation of different configurations with varying dimensions and obstacles for training the autonomous mobile robot.
3. Troubleshooting in the simulation – correction of visualization, colors, camera positions, and fixed components to match the real robot.
4. Control of the autonomous mobile robot in simulation – use of plugins for wheel control and publishing data from LiDAR and other sensors via ROS2.
5. Positioning of the target and autonomous mobile robot in simulation – random initialization of the robot's and target's positions within the simulation to enable the learning of adaptive navigation policies by the agent.

Figure 47 presents a scheme of the implemented communication using the robot control plugin in the Gazebo simulator.

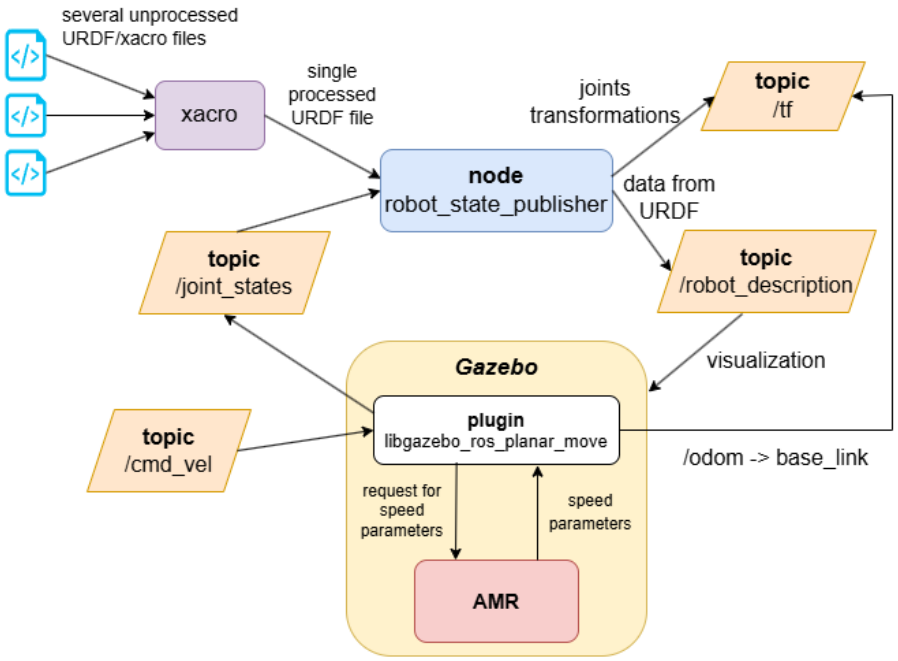


Fig. 47. Scheme of robot control in the Gazebo simulator.

3.3 Creating an RL training system

The RL training process is complex; in particular, training using the PPO method includes the following steps:

1. Initializing the environment and agent with configured hyperparameters.
2. Setting the initial state of the environment to obtain initial observations.
3. For each update iteration:
   - а) collecting trajectory data of the agent's movement;
   - б) calculating rewards for each step and state advantages;
   - в) repeatedly updating the policy and value functions based on the input data;
   - г) saving relevant metrics to track training progress, as well as the best model weights obtained so far.

A reward function for navigation purposes is defined based on encouraging actions that lead to selecting a short and safe trajectory to reach the goal.

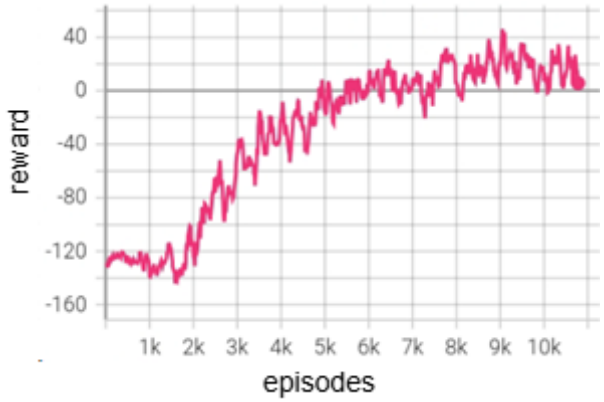
3.4 Training a navigation model using the PPO method

Experimental tuning and fine optimization of the hyperparameters and architecture of the PPO model were performed to achieve more stable and efficient training in the reinforcement learning environment. Specifically, key hyperparameters were identified, such as learning rate, batch size, number of data collection steps, number of optimization epochs, entropy coefficient, value function coefficient, policy clipping range, and advantage normalization. To determine their influence on training efficiency and algorithm stability, experiments were conducted with different hyperparameter values. The results were analyzed by comparing the cumulative reward over the last 100 episodes and demonstrated the impact of learning rate, batch size, and the number of neurons in hidden layers on training quality. It was shown that for the given task, it is advisable to use a lower learning rate, a larger batch size, fewer optimization epochs, and a shallower network architecture. After numerous experiments, the following hyperparameters were selected for training the PPO model for the given task:

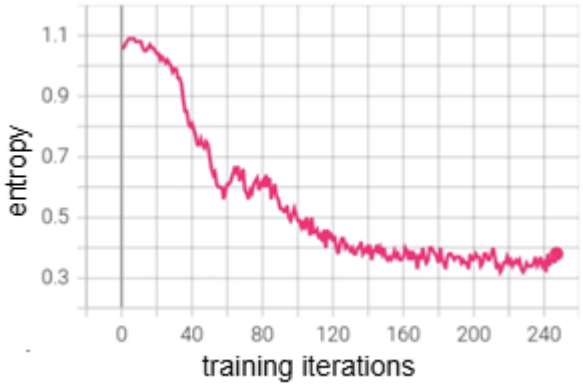
Table 15. Hyperparameters used for training the PPO model in the 3D simulation environment

Hyperparameter	Value
Steps count	4096
Batch size	512
Number of epochs	3
Clipping range	0.2
Entropy coefficient	0.001
Generalized advantage estimation coefficient	0.99
Learning rate	0.003
Advantage normalization	True
Value function coefficient	True
Target Kullback-Leibler indicator	0.015

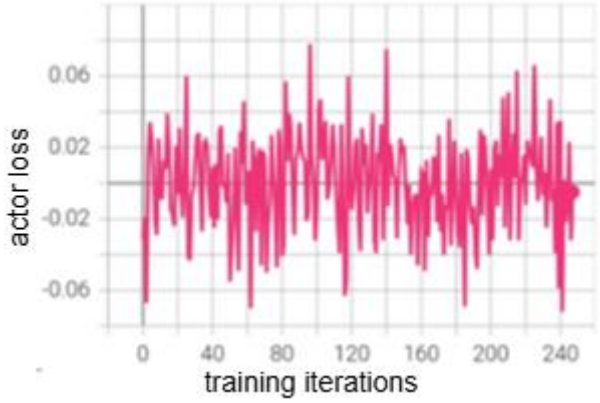
The characteristics of the indicators of the trained deep neural network using the PPO model with all specified hyperparameters are shown in Fig. 55.



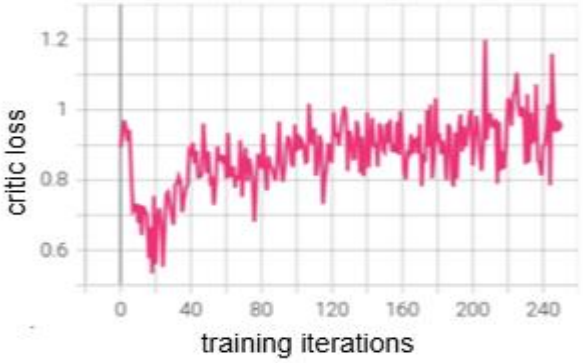
a.



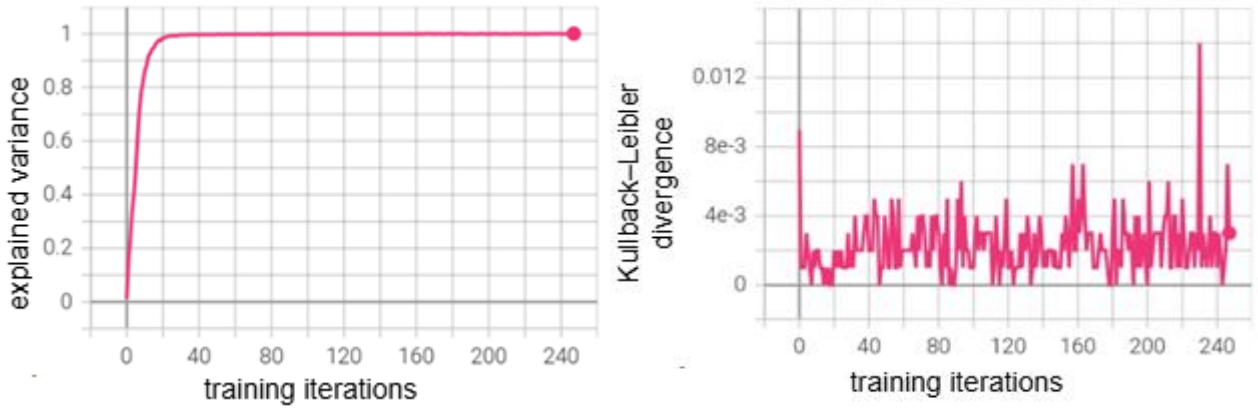
b.



c.



d.



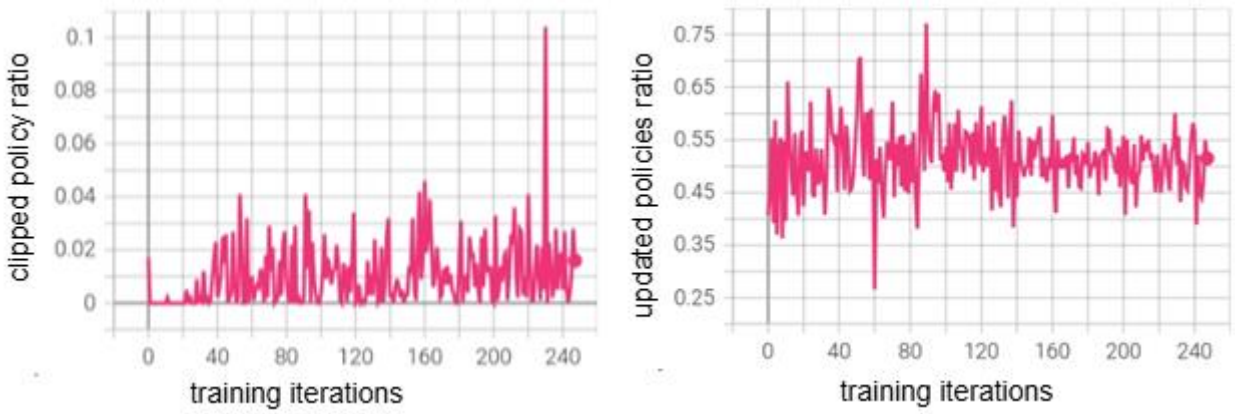


Fig. 55. Progress of DNN characteristics during training: (a) cumulative reward averaged over the last 100 episodes; (b) average entropy of the actor's action distribution; (c) actor loss function; (d) critic loss function; (e) explained variance coefficient of the critic; (f) measure of Kullback–Leibler divergence of the new policy; (g) percentage ratio of clipped action probabilities relative to the total number of actions in one batch; (h) percentage ratio of updated policies relative to their total number in one batch.

The agent achieves optimal behavior with 90% accuracy between training iterations after 9,000 episodes, with the average cumulative reward approaching 45.

3.4.3 Modification of the DNN architecture

To improve the training speed using the PPO method, the DNN architecture of the actor was updated by adding an additional branch based on the residual connections approach. The overall architecture of the actor's DNN is shown in Fig. 56.

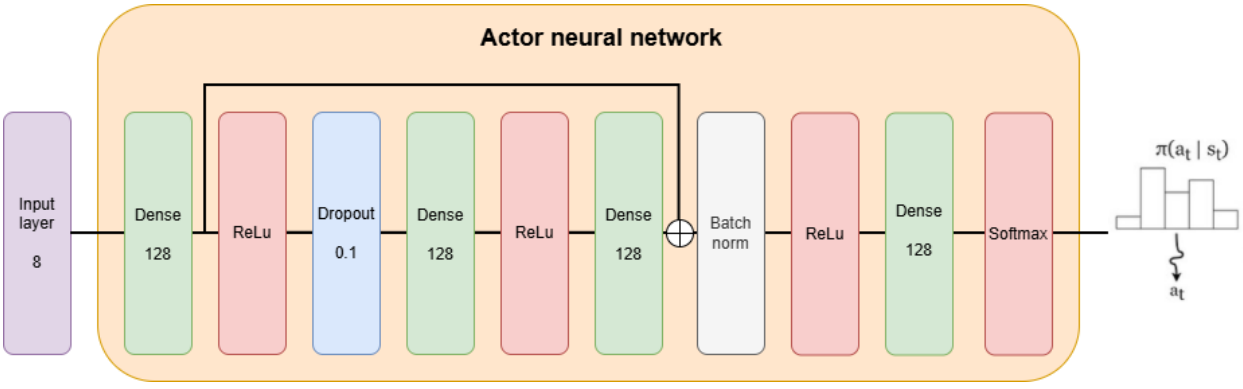


Fig. 56. Actor DNN architecture with residual connection

The described improvements to the DNN contribute to increasing the training speed of the PPO model by discovering new relationships in the states. Essentially, this causes the two branches to learn different representations of the input data, which are subsequently summed and passed through the final hidden layer with 128 neurons, followed by a Softmax activation function. The results of experiments with the described architecture are shown in Fig. 57.

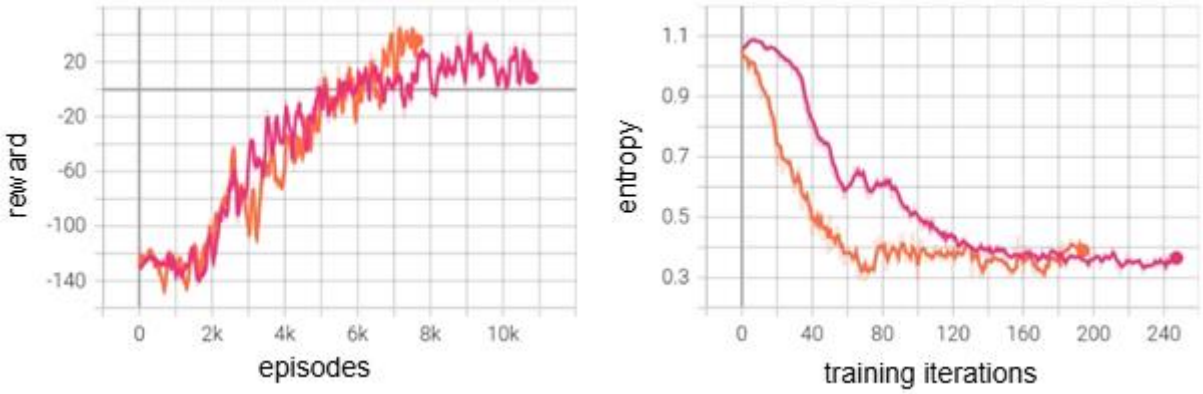


Fig. 57. Training with the updated actor architecture using residual connections: left – cumulative reward averaged over the last 100 episodes; right – average entropy of the actor's action distribution.

The training process accelerates after episode 6,500 and reaches its maximum average value of 32 points over the last 100 episodes by episode 7,000. The same value is reached by the baseline model 2,000 episodes later.

3.4.4 Refinement of the learning rate

In training the PPO model for the given task, experiments were conducted by modifying the actor's learning rate and evaluating training progress. Progress is determined by the agent's performance over a specified number of recent episodes. A flowchart of the process is shown in Fig. 59.

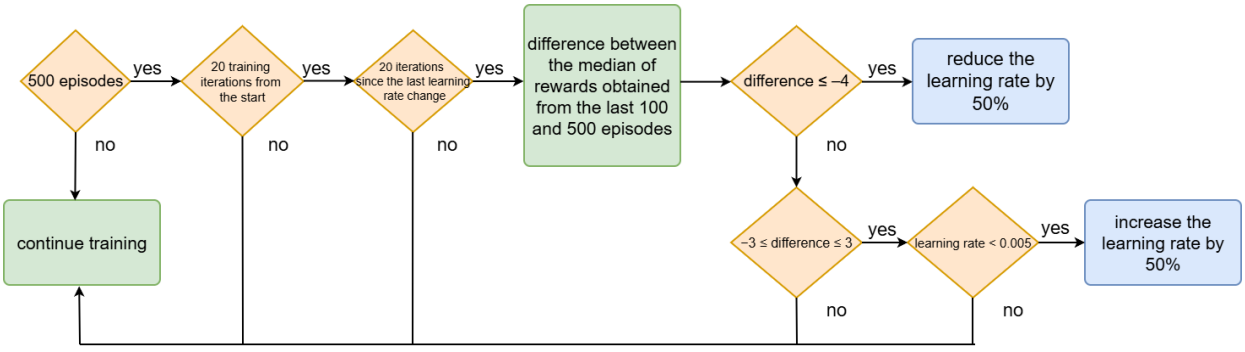
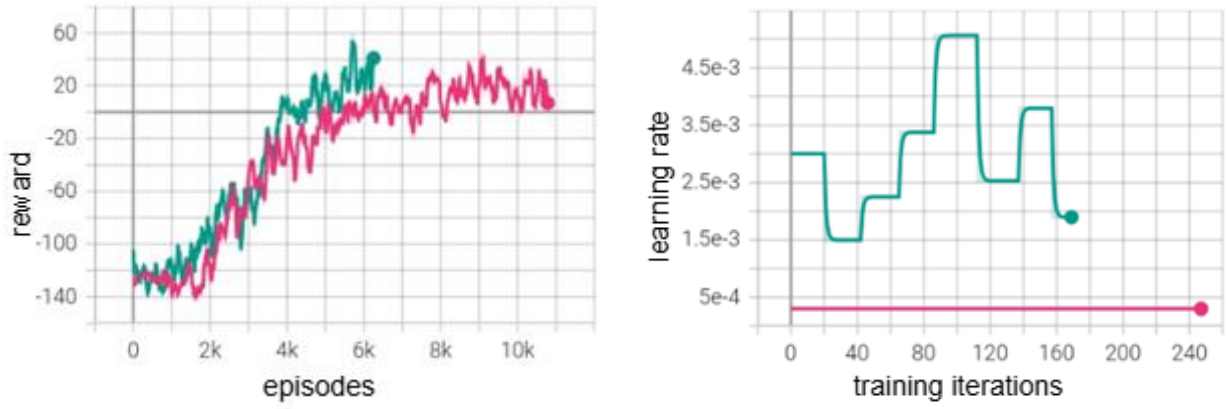


Fig. 59. Algorithm for learning rate adjustment

The minimum threshold values are set for the number of elapsed episodes (500), training iterations from the start of training (20), and iterations since the last learning rate change (20). This ensures that learning rate adjustments occur gradually and based on a larger sample of data. The difference between the median rewards of the last 500 and 100 episodes is compared (20%). These threshold values were determined experimentally and serve as hyperparameters of the algorithm. The statistical parameter median was chosen to reduce the influence of reward dispersion when detecting unknown states by the agent.

When evaluating training progress, if the median reward decreases (difference equal to –4), indicating a downward trend in rewards, the learning rate is reduced by half. Conversely, when the agent is in a plateau phase and the reward function shows no significant change (difference between –3 and 3), the learning rate is increased by 50% relative to its previous value. An additional constraint is introduced — the learning rate must not exceed 0.005 to prevent excessively large weight updates. Thus, when training progress slows or stops, learning rate adjustments are applied, while if progress is observed and the reward curve is upward, training continues with the chosen rate.

Figure 60 shows the decrease in learning rate and the agent's average reward during training, compared with those of the baseline model.



a) b)

Fig. 60. Training using the learning rate adjustment algorithm: (a) cumulative reward over the last 100 episodes; (b) learning rate changes.

Training the algorithm to 90% accuracy using the described learning rate adjustment approach requires significantly less time — namely, 6,300 episodes with a maximum reward of 60 points — compared to 9,000 episodes and a reward of 32 points.

3.4.5 Experience replay buffer

A new approach to experience storage is considered, based on the agent's performance in specific episodes and more efficient utilization of stored data. The process is divided into three main steps: experience storage, formation of a training sample, and memory cleanup. For this purpose, two experience buffers with different purposes are created. The scheme of the process is shown in Fig. 61 and Fig. 62.

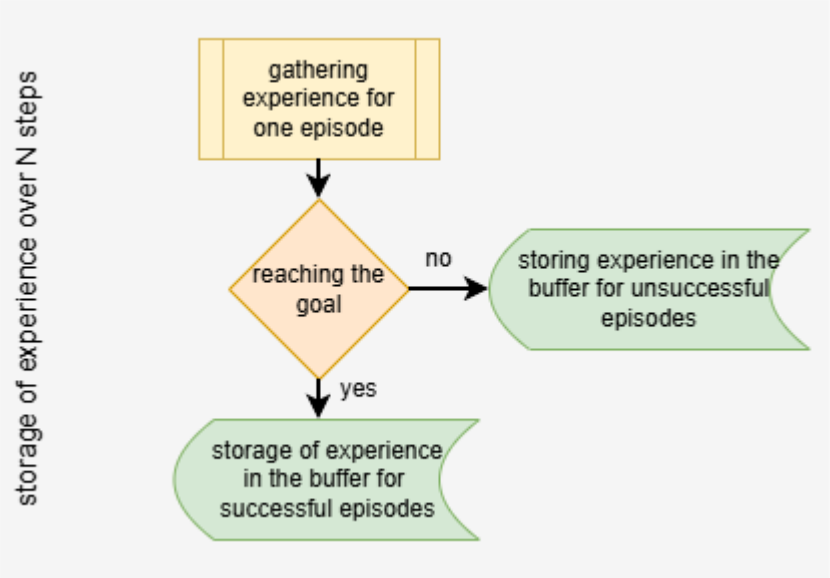


Fig. 61. Process of storing the obtained experience

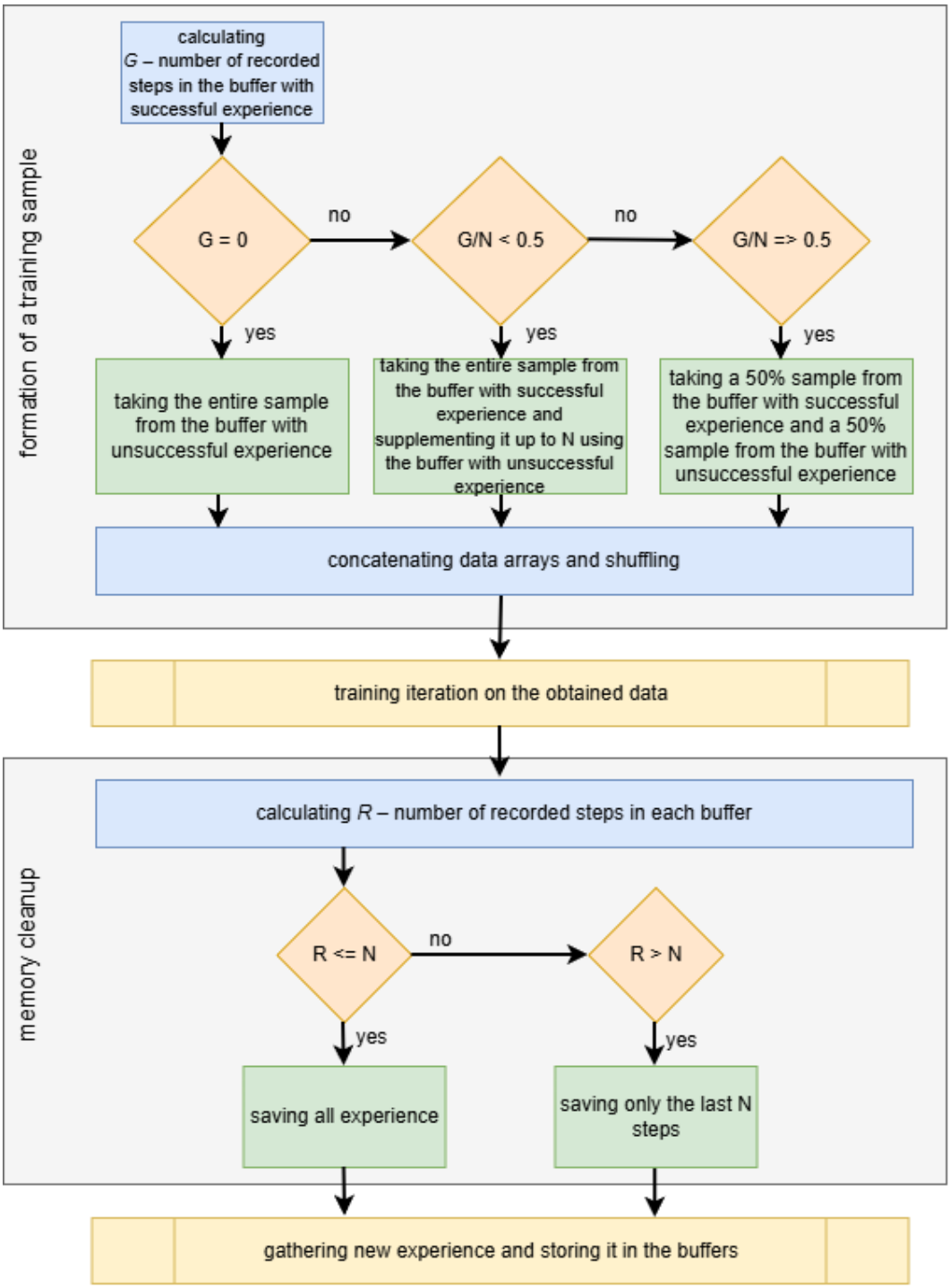
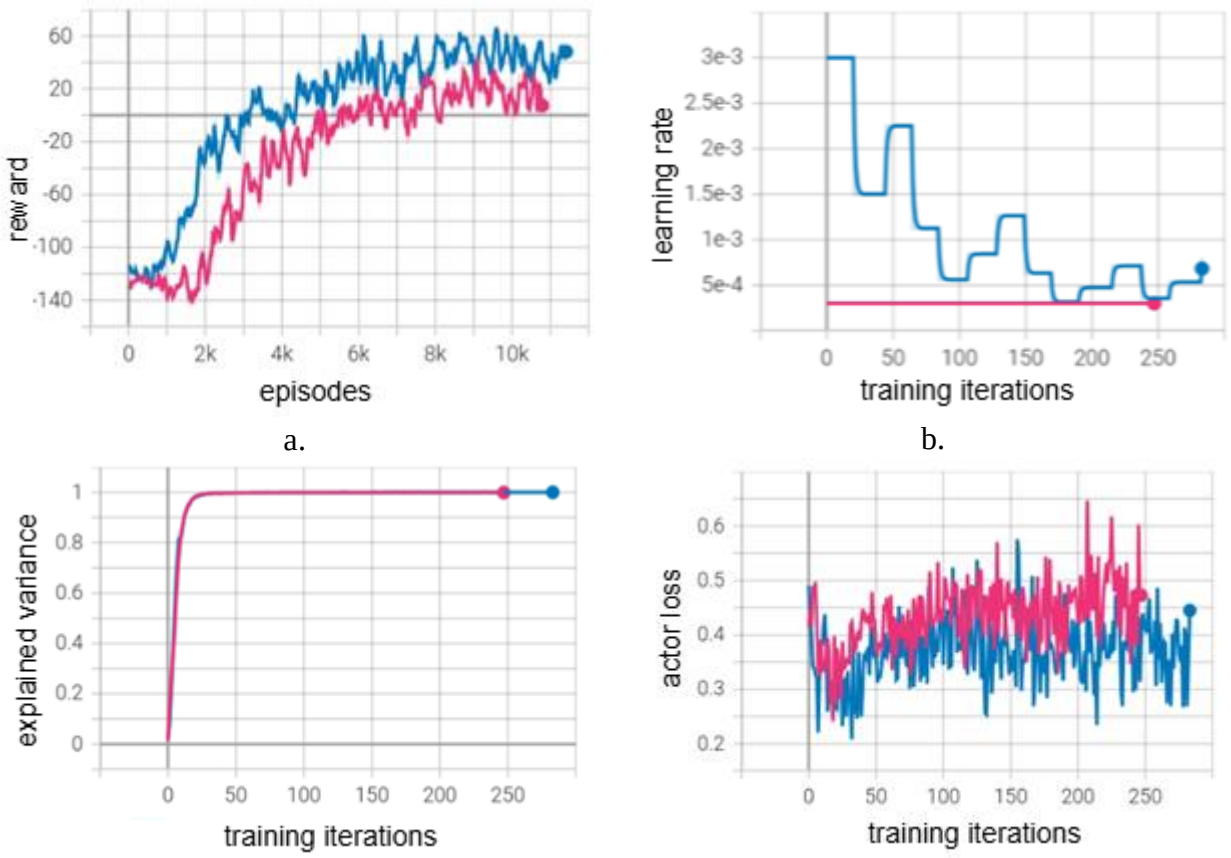


Fig. 62. Formation of a sample from the buffers and memory cleanup

Each episode is stored with detailed data on observed states, actions, probabilities, received rewards, and the final outcome. When a sufficient number of steps is accumulated (4,096), a training sample is formed from both buffers by selecting newer episodes and combining them according to the ratio of successful to unsuccessful cases. After that, outdated experience is removed to ensure that the agent always trains on the most recent data from the environment, thereby improving training efficiency and adaptability. The results of the described approach are shown in Fig. 63.



a. b. c. d.

Fig. 63. Training process using the experience storage algorithm: (a) cumulative reward over the last 100 episodes; (b) learning rate changes; (c) explained variance coefficient of the critic; (d) loss function.

The results of using the described memory buffer management method demonstrate more efficient operation, as evidenced by the cumulative reward curve compared to the baseline model — training progress occurs significantly earlier and exhibits a more upward trend than in the baseline model.

3.5 Generalized comparative analysis of the implemented improvements

Table 16 presents a generalized comparative analysis of the baseline model and the proposed improved configurations, including modifications to the NN architecture, adaptive learning rate, and utilization of stored experience.

Table 16. Quantitative evaluation of the comparative analysis between the baseline and improved models

Indicator	Baseline model	Model with modified NN architecture	Model with adaptive learning rate	Model with improved experience utilization
Convergence speed (episodes)	~9000	~7000	~6300	~6100
Successfully completed test episodes (%)	90%	90%	90%	95%
Average reward	40	53	51	62
Standard deviation of reward	10.8	10.3	9.7	10.1
Minimum entropy value	0.3	0.3	0.2	0.2

Iteration at which minimum entropy of actor action distribution is achieved	210	70	68	53
Minimum actor loss function value	-0.07	-0.05	0.01	0.05
Standard deviation of actor loss function	0.6	0.54	0.38	0.35
Minimum critic loss function value	0.65	0.67	0.49	0.21
Standard deviation of critic loss function	0.73	0.82	0.56	0.23
Maximum explained variance coefficient of the critic	0.99	0.99	0.99	0.99
Iteration at which maximum explained variance coefficient of the critic is achieved	20	19	18	18

In terms of training speed, an improvement is observed in all modified models compared to the baseline model. While the baseline model reaches maximum success of 90% after approximately 9,000 episodes, the model with a modified architecture using residual connections and additional regularization of the actor's DNN improves convergence speed by 22% (to 7,000 episodes) while maintaining the same success rate in test episodes (90%). The additional introduction of adaptive learning rate results in achieving the same success rate after 6,300 episodes, and the best results are obtained with the model featuring improved experience buffer utilization, which trains in approximately 6,100 episodes (a 32% improvement compared to the baseline model). Furthermore, the achieved success rate in test episodes increases by 5% (95%) compared to the baseline model (90%).

3.6 Ablation analysis of the improved model

To quantitatively evaluate the contribution of the proposed improvements to the effectiveness of the PPO algorithm, an ablation analysis was performed. The main idea of the analysis is systematic deactivation of individual components of the improved model and measurement of their impact on training speed, algorithm stability, and final agent performance in test episodes. The baseline reference is the original PPO model with the initial actor and critic architecture, fixed learning rate, and standard experience collection method. The results of the experiments are shown in Table 17.

Table 17. Results of the ablation analysis of the implemented improvements

Model configuration	Architecture change	Adaptive learning rate	Improved experience buffer	Convergence speed (episodes)	Min. entropy	Average reward	Max. achieved accuracy (%)
Baseline model	X	X	X	~9000	0.3	40	90
Without architecture change	X	✓	✓	~8200	0.3	45	92
Without adaptive	✓	X	✓	~7500	0.25	48	90

learning rate							
Without improved experience buffer	✓	✓	X	~7000	0.25	50	90
Improved model (full)	✓	✓	✓	~6100	0.2	62	95

The results of the conducted experiments demonstrate that each of the proposed improvements independently contributes to enhancing the efficiency of the PPO algorithm. The most significant impact on training speed is observed from the modification of the actor's DNN architecture and the adaptive learning rate, while the experience buffer management approach improves training robustness and stability. The combination of all proposed improvements yields the best balance between high training speed, stability, and improved accuracy.

3.7 Conclusions

This chapter examined the implementation of the reinforcement learning approach in the Gazebo 3D simulation, involving a 3D model of the mobile four-wheel robotic platform Yahboom RDK X3 with mecanum wheels and a LiDAR sensor. The preparation of the system for autonomous navigation is a multi-step process and includes the creation of a simulation environment, sensor configuration, tuning and deployment of robot control with ROS2, and the development of a reinforcement learning training system. The implemented simulation environment provides controlled and reproducible conditions for training and testing reinforcement learning algorithms for the given task.

The conducted experiments confirm the possibility of successfully training a navigation policy using the reinforcement learning approach with LiDAR and odometry data, without the use of pre-built maps. The combination of Gazebo with ROS2 and Python offers a high degree of modularity, flexibility, and full control over simulation and training processes.

Fine tuning of the PPO algorithm was implemented and a baseline model was built, which is used for subsequent improvements of the algorithm. The achieved success rate of 90% in testing confirms the scientific hypothesis about the high efficiency of this approach for autonomous navigation tasks of mobile robots. Through careful selection of hyperparameters, neural network architecture, and reward function, stable learning and high decision-making accuracy of the agent were achieved.

A thorough analysis and experimental investigation of possibilities for improving the PPO algorithm in the context of autonomous navigation were conducted. By introducing architectural modifications to the DNN, adaptive learning rate, and an enhanced mechanism for managing stored experience, a significant improvement in training efficiency and stability was achieved. The results of the comparative and ablation analyses show that each of the proposed improvements contributes independently and positively to model performance. The implementation of residual connections and normalization in the actor's architecture improves training speed and facilitates faster acquisition of the optimal behavior policy. Adaptive learning rate adjustment based on training progress enhances process stability and reduces the time required to reach optimal behavior. The use of a more efficient experience storage and utilization mechanism improves model generalization capabilities, with test episode success reaching 95%. Additionally, the improved model demonstrates a 32% faster training speed, a 55% higher average reward, 5% better accuracy, and more stable

behavior during training compared to the baseline model. Furthermore, analysis of entropy and loss functions for the actor and critic proves earlier formation of a stable policy. The critic successfully approximates the value function in all considered cases, with the improved model achieving this in fewer iterations.

From a conceptual standpoint, the advantage of the proposed methodology compared to established DRL navigation approaches lies in its systematic nature and potential for integration and enhancement. A combination of targeted improvements is proposed, addressing different aspects of the training process — speed, stability, and data utilization efficiency. Compared to other DRL approaches that often require complex tuning to achieve good results, the proposed methodology maintains the stability and simplicity of the PPO algorithm while overcoming its limitations related to convergence speed and homogeneous experience management. The conducted experiments demonstrate that these improvements lead to faster formation of a stable navigation policy. The described comprehensive approach also shows that navigation efficiency improvements can be achieved not only by replacing the algorithm but also by intelligently adapting and combining architectural and training mechanisms within an established DRL paradigm.

The results achieved in simulation provide a foundation for subsequent integration and testing of the model in a real physical environment, described in Chapter 4, where its generalization capabilities and robustness under real-world conditions are also evaluated.

Chapter 4. Experimental research and analysis of the applicability of the obtained model for autonomous mobile robot navigation

4.1 Technical description of the equipment used

For the purposes of the experimental studies, a robotic platform was used, developed based on the open-source middleware ROS2. The RDK X3 controller serves as the main computational unit, providing real-time data processing and execution of complex computational tasks. The platform is equipped with high-performance hardware components, mecanum wheels for complex movement capabilities, a LiDAR sensor for distance measurement (time of flight, ToF) and mapping, and a 3D camera (Table 18).

Table 18. General hardware parameters

processor	ARM Cortex-A53, 4 cores at 1.2 GHz
operating system	Ubuntu 20.04 and ROS-Foxy
sensors	LiDAR MS200, CSI camera, 3D camera
power supply	DC, 7.4 V
battery life	3.5 hours
remote control	joystick, keyboard, mobile phone
communication	local network (LAN), access point (WiFi)
chassis material	aluminum alloy
safety	connection protection, short-circuit protection, rotor blockage protection
external dimensions	236.11 x 181.10 x 184.9 mm
weight	1.93 kg

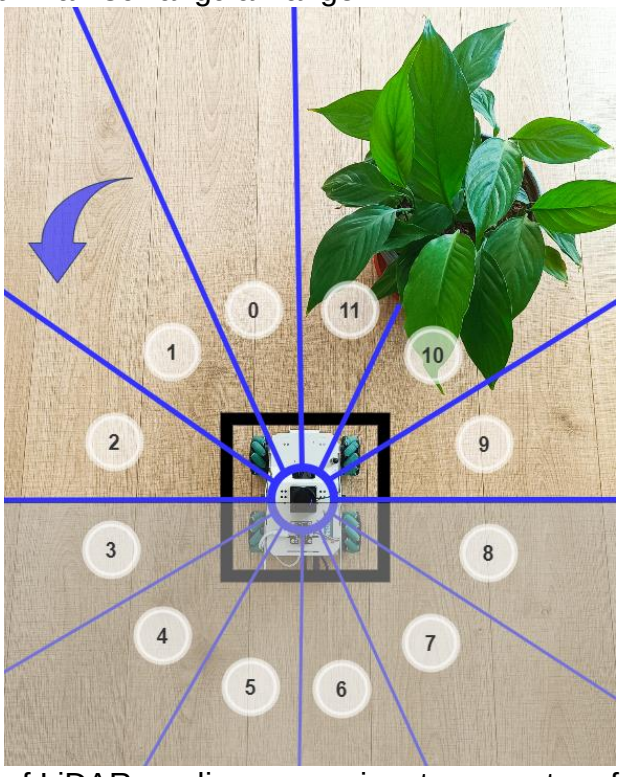
4.2 Software configuration with ROS2 for navigation

For the experimental study, the following software was installed on the Yahboom RDK X3 platform:

- operating system: Ubuntu 20.04.6 LTS;;
- robotic middleware package: ROS2 Foxy;
- Python 3.8.10 with the following libraries:
  - PyTorch 2.4.1 – for deep learning;
  - NumPy 1.24.3, math, squaternion – for mathematical computations;
- rclpy – Python client library for ROS2;
- ros-foxy-geometry-msgs, ros-foxy-nav_msgs, ros-foxy-sensor_msgs, ros-foxy-visualization_msgs – ROS2 Foxy message packages.

For remote connection to the robot, the RealVNC remote desktop software was used for both desktop and mobile devices.

The LiDAR MS200 sensor was configured with a measurement range from 0° to 360°, rotating counterclockwise, and a distance range from 0.05 to 20 m. The selected signal transmission frequency is 10 Hz. LiDAR readings are published in the topic MS200/scan. A schematic representation of the implementation is shown in Fig. 69. One 360° scan is divided into 12 equal zones with a field of view of 30°, and the minimum distance to obstacles is extracted for each of the six zones with indices [0, 1, 2, 9, 10, 11]. Thus, the LiDAR readings form a vector of six floating-point variables describing the minimum distances to obstacles, evenly distributed across the frontal section of the robot within a 180° angular range.



Configuration of LiDAR readings as an input parameter of the algorithm

Data from odometry are obtained by combining IMU data read via ROS2 and speed data. Figure 70 shows the implementation of robot localization using odometry.

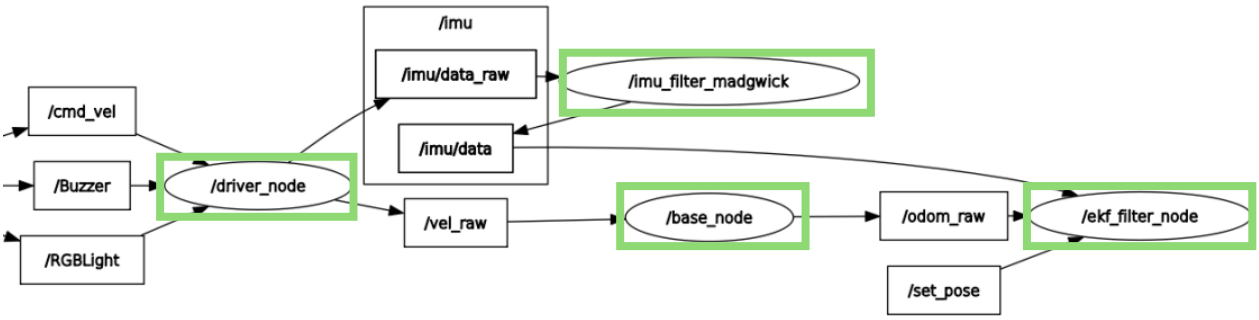


Fig. 70. Part of the message transmission architecture for localization purposes

At each step with specified linear and angular velocity, the mobile robot configures the input data for the algorithm as described in formula (11).

S_t = 6 LiDAR readings + distance to the goal + orientation relative to the goal (1)

Figure 71 shows the overall distributed architecture built in ROS 2, which includes sensor readings (LiDAR), the motion controller, and localization algorithms, together with the transformations implemented for the navigation task.

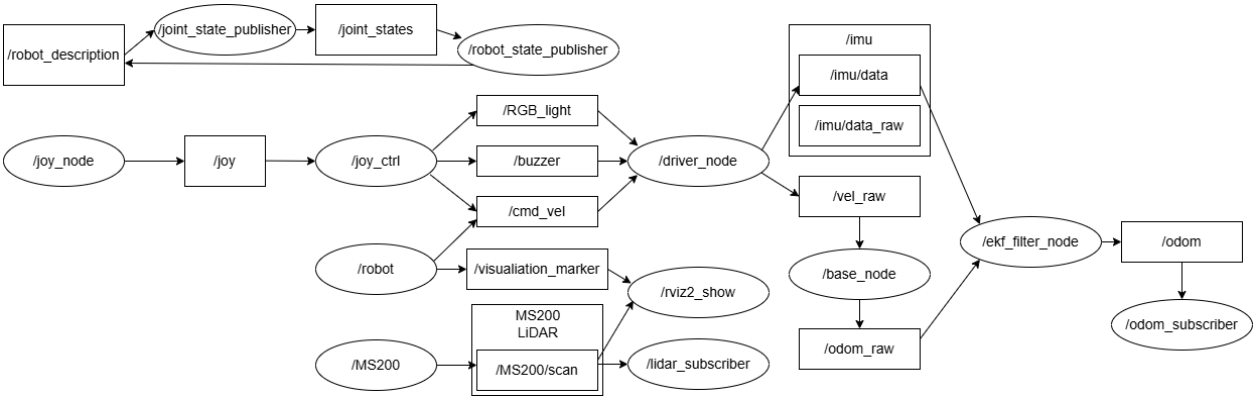
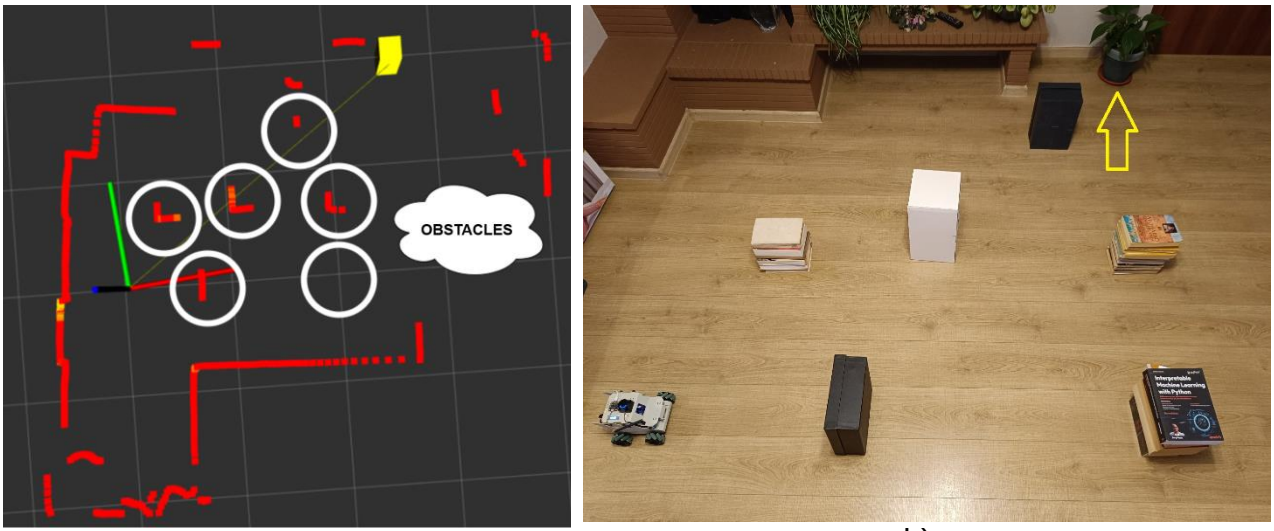


Fig. 71. Implemented communication architecture in ROS2

4.3 Experimental study of the applicability of reinforcement learning algorithms in a real environment

The experimental setup in the real environment represents an enclosed space with an irregular shape measuring 4 × 3.8 m, containing various objects (obstacles) of different shapes and sizes. The autonomous mobile robot (AMR) is positioned at one end of the enclosed space, while the goal is located 3.15 m away from it. Experimental results are collected under conditions with 0, 1, 2, 3, 4, 5, and 6 obstacles in order to analyze the performance of the baseline and improved models obtained in Chapter 3. For comparative analysis of the trained models, the positions of the AMR and the goal remain constant. Figure 75 shows the experimental setup with the maximum number of six obstacles and a real photograph of the setup.



a) b)

Fig. 75. Visualization of the experimental setup in RViz (a) and real environment (b) with 6 obstacles

The main indicators collected during the experiments were a boolean value for successful episode completion, the minimum and average number of steps taken for each episode, and the final state of the agent upon episode termination. Table 22 presents quantitative characteristics from the analysis of both models.

Table 22. Experimental results in the real environment

Model	Obstacles count	Successful episodes	Success rate	Number of steps per episode (for successfully completed episodes)		
				Avg.	Min.	Max.
Baseline	0	26	87%	68	59	85
Baseline	1	26	87%	69	63	86
Baseline	2	25	83%	76	68	88
Baseline	3	25	83%	78	73	92
Baseline	4	24	80%	80	73	96
Baseline	5	25	83%	83	76	98
Baseline	6	25	83%	89	76	98
Improved	0	28	93%	66	58	83
Improved	1	28	93%	66	61	87
Improved	2	27	90%	74	65	88
Improved	3	27	90%	78	70	93
Improved	4	26	87%	78	68	95
Improved	5	27	90%	83	75	98
Improved	6	26	87%	85	76	99

The most reliable indicator of the robustness of the trained models to noise and errors in sensor readings, as well as their generalization capability, is their success rate in performing the given task. Figure 76 shows the results obtained in the described experimental setup with 0, 1, 2, 3, 4, 5, and 6 obstacles.

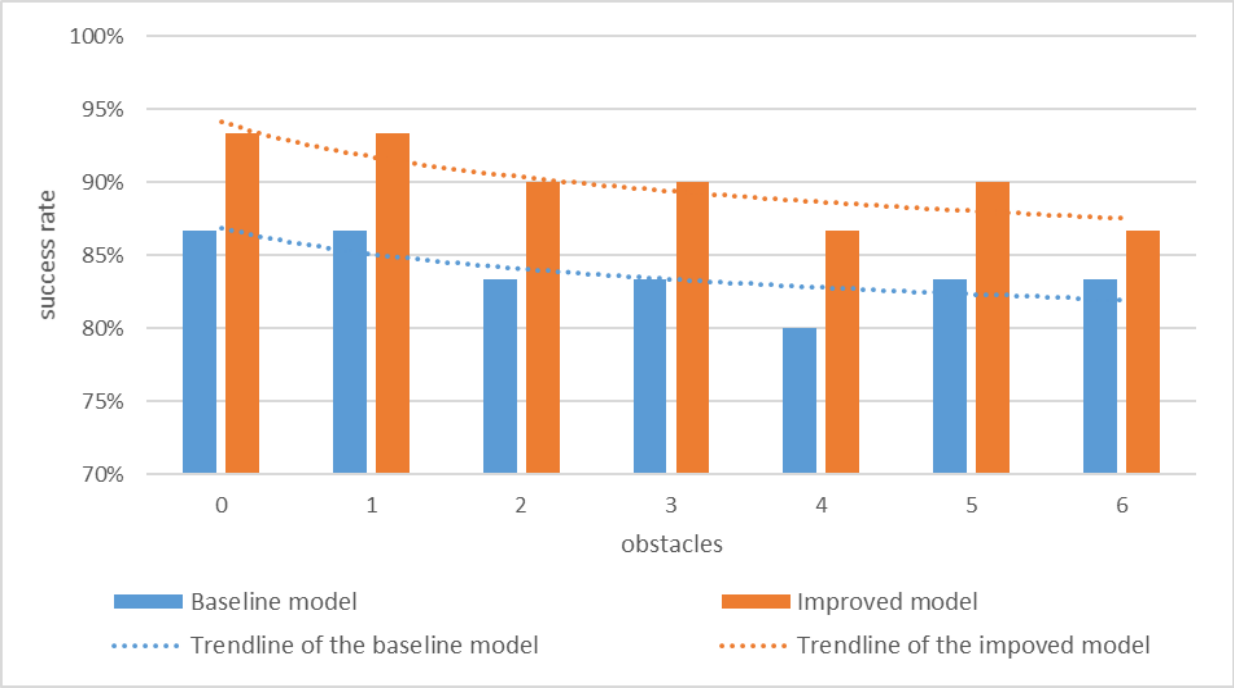


Fig. 76. Comparison of model success rates

Both models demonstrate high and relatively stable success across all experimental setups. The improved model outperforms the baseline in every obstacle scenario, with the difference in success rate ranging between 4% and 7% in favor of the improved model.

The resulting states of the autonomous mobile robot in unsuccessfully completed episodes—where termination occurred due to collision with an obstacle or time expiration (with a limit of 100 steps)—are visualized in Fig. 77.

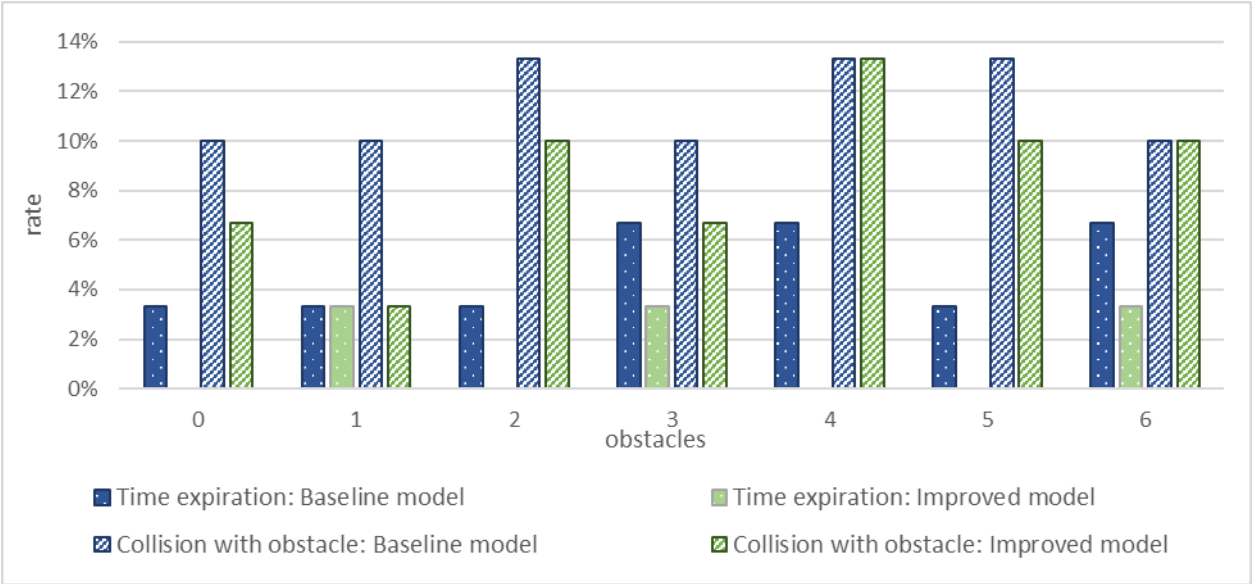


Fig. 77. Comparative analysis of terminal states in unsuccessfully completed episodes

Collisions represent the main cause of unsuccessful episodes for both models. For the baseline model, the collision frequency varies between 10% and 13%, with no clearly defined monotonic dependence on the number of obstacles. For the improved model, the collision frequency is lower than that of the baseline model in all setups except for cases with 4 and 6 obstacles, where both models show the same value (13%).

It is essential for determining optimality to examine the movement trajectories of the autonomous mobile robot in experiments with each model. For this purpose, the average number of steps in trials with different numbers of obstacles and the minimum number of steps required to reach the goal are compared. Figure 78 shows the differences in the average number of steps required to successfully complete episodes for both models.

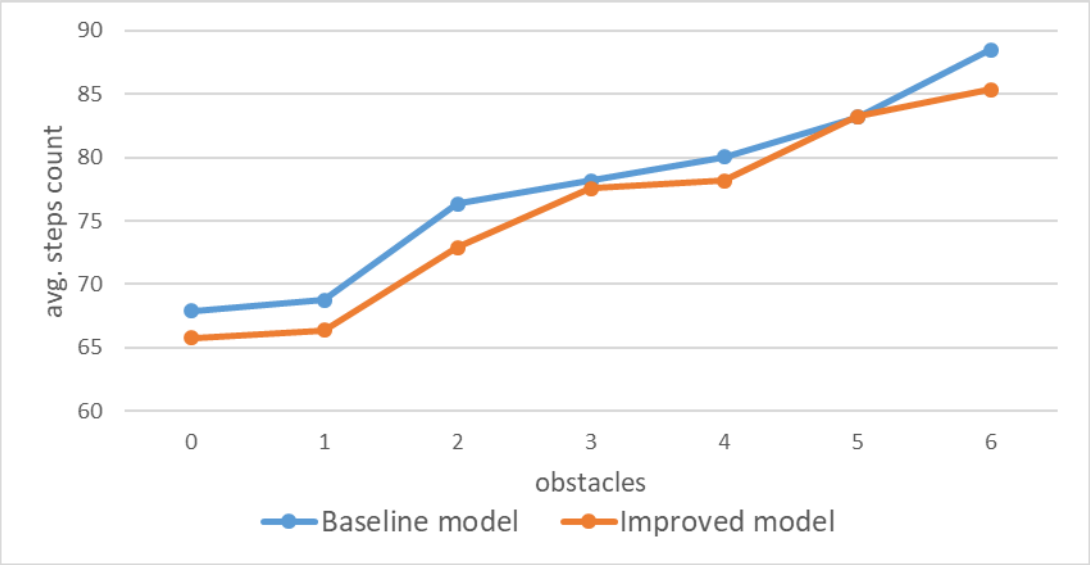


Fig. 78. Comparison of the average number of steps for both models

When obstacles are added, the movement path becomes longer — averaging from 67 to 87 steps. It is evident that the path of the improved model is shorter by about two steps in each setup. The difference is not noticeable only in the configuration with five obstacles, where the average number of steps is 83 for both models. Regarding trajectory optimality for each setup, Figure 79 shows the minimum number of steps for different numbers of obstacles. The difference between experiments with 0 and 1 obstacle is more pronounced here — the number of steps in the shortest trajectories increases as obstacles in the environment grow. The trajectory length required to reach the goal increases gradually from about 58 steps with no obstacles to 76 steps with six obstacles.

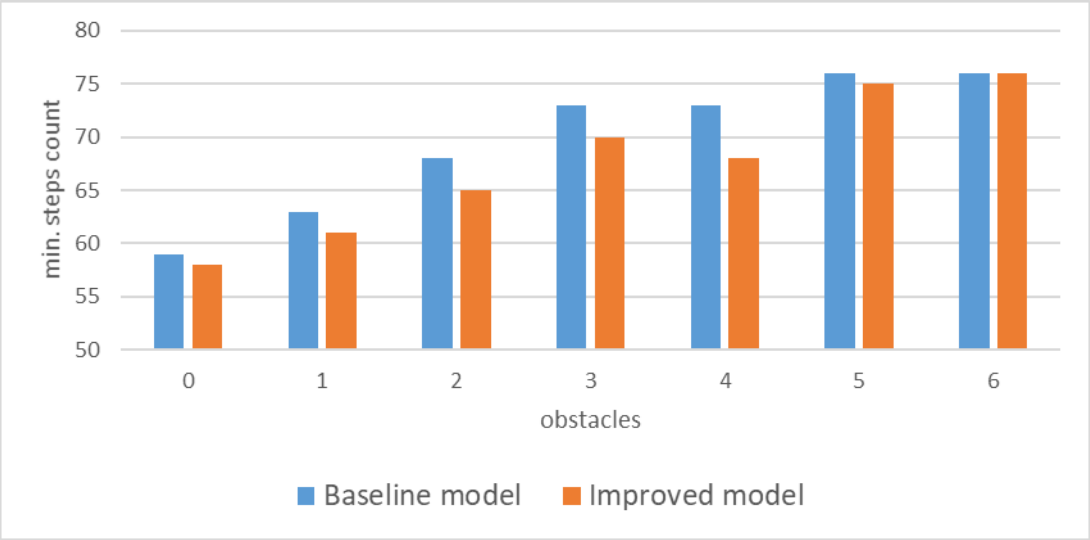


Fig. 79. Comparison of the minimum number of steps

Larger increases in the minimum trajectories are observed with 2 and 5 obstacles, where the difference amounts to approximately five additional steps. This difference in the minimum number of steps between the two models for each setup is shown in Figure 80.

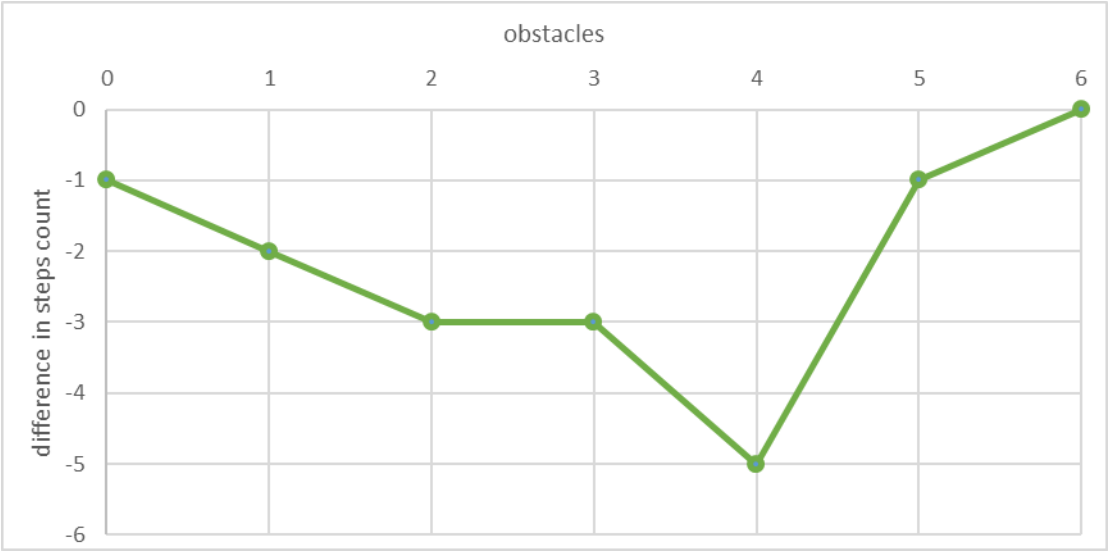


Fig. 80. Difference in the minimum number of steps when comparing the models

The trajectories of the improved model are generally one or more steps shorter than those of the agent whose actions are determined by the baseline model. The only exception is the configuration with six obstacles, where the results are identical. From the above, it can be concluded that as the number of obstacles increases, the efficiency of the baseline model decreases more sharply, whereas the improved model maintains its robustness and exhibits significantly smaller fluctuations in performance.

The quantitative evaluation of model transfer from simulation to real environment is based on the results of the conducted experiments, summarized in Table 22. The main metric for functional applicability is success rate — the proportion of episodes in which the autonomous mobile robot reaches the goal without collision and within the defined time limit. Table 23 presents a comparison of the results obtained in simulation and in the physical environment for both the baseline and improved models.

Table 23. Comparison of model accuracy in simulation and real environment

Model	Environment	Average success rate
Baseline	Simulation	90%
	Real	84%
Improved	Simulation	95%
	Real	90%

The results show that both models maintain high success rates in experiments during direct transfer from simulation to the real environment without additional training. The accuracy of the models decreases in the real environment by 6% and 5% for the baseline and improved models, respectively. However, the improved model retains higher absolute success in the real environment compared to the baseline, confirming its superior generalization capability.

4.4 Conclusions

In this chapter, the practical transfer of the trained models from the 3D simulation environment to a physical setup involving the Yahboom RDK X3 autonomous mobile robot was implemented. The experiments show that successful transition from the virtual experimental setup to a real one requires the following steps, actions, and tools:

- Precise configuration of the model's input data in the real environment to enhance generalization from observed states in the simulation to unseen states in the real environment.
- Achievement of high final model accuracy through reinforcement learning in a simulation environment that matches the parameters of the real one.
- Optimization of the AMR architecture configuration in ROS2 by creating subscriber and publisher nodes for the required topics using the appropriate message formats.
- Use of auxiliary ROS2 tools such as RViz for spatial visualization from the robot's perspective and RQt for displaying the overall architecture, which simultaneously assist in system visualization and the detection of errors, inaccuracies, and inconsistencies.

The difficulties encountered when transferring from the simulation environment to the real environment are mainly related to errors in the signals obtained from the LiDAR and odometry. In this context, experiments revealed that LiDAR readings contain incorrect zero values and errors in distance measurements to obstacles, which necessitates their additional processing and configuration before obtaining the final state. Control of the autonomous mobile robot also requires periodic calibration of angular and linear velocity to reduce errors in motion control. The ROS2 software architecture developed within this dissertation provides reliable communication between nodes for sensor data, localization, control, and visualization, enabling data processing and adaptive real-time robot control.

Experiments with different numbers of obstacles were conducted, and a methodology for analyzing the baseline and improved models was developed using several metrics, including: success in reaching the goal, final episode outcome, and optimality of the obtained trajectories. The experimental results clearly demonstrate more effective control of the autonomous mobile robot using the model trained to higher accuracy (95%) in simulation compared to the baseline model trained with 90% accuracy. In the real environment, the improved model achieves higher overall accuracy than the baseline — 90% versus 84% — as well as shorter movement trajectories,

measured by the average number of steps in 180 experimental episodes (78 for the improved model and 76 for the baseline).

The experiments conducted with the physical autonomous mobile robot confirm the possibility of successfully applying a navigation policy based on the reinforcement learning (DRL) approach, trained in a simulation environment, without the use of pre-built maps and GPS positioning. The observed deviation in model accuracy during transfer to the real environment is -5%, representing a moderate decrease in performance compared to simulation results, which is expected given the presence of sensor noise, inaccurately modeled dynamics, and accumulating odometry errors. Nevertheless, the quantitative analysis shows that this difference is limited and does not hinder the practical applicability of the proposed solution.

The obtained results demonstrate that the proposed system can be used as a foundation for developing intelligent navigation modules in robotic platforms operating in dynamic environments with limited sensor information. The investigated approach creates prerequisites for future expansion toward multi-agent systems, integration with additional sensors, and adaptation to more complex navigation scenarios.

III. AUTHOR'S REPORT ON CONTRIBUTIONS

Scientific-applied contributions:

1. An original model for autonomous navigation of a mobile robot without the use of pre-built maps and GPS positioning has been developed, based on local environmental perception through deep reinforcement learning, which enables the formation of a stable navigation policy in a dynamic and partially observable environment.
2. A model describing the relationship between input sensor parameters, the structure of the reward function, and the characteristics of the learned navigation policy was formulated and experimentally validated, demonstrating their key influence on movement safety, smoothness, and stability.
3. A thorough analysis of the convergence and robustness of a navigation model based on the PPO method was conducted, including a quantitative assessment of the influence of key hyperparameters on training stability and efficiency under conditions of stochasticity and sensor noise.
4. A modified mechanism for efficient experience buffer utilization was proposed, which optimizes the distribution of training samples, accelerates convergence, and improves policy accuracy — findings confirmed by comparative experimental results.
5. An adaptive mechanism for dynamic learning rate adjustment was developed, reducing fluctuations in performance improvement and enhancing the stability of the PPO-based algorithm in complex navigation scenarios.

Applied contributions:

1. An integrated simulation environment (Flatland/Gazebo) was developed for studying navigation through reinforcement learning using a realistic model of a mobile platform.

2. An experimental system for autonomous navigation with a limited set of sensors was implemented, applicable in industrial enclosed spaces.
3. A real experimental setup was built for validating navigation policies using a physical robot.
4. Successful transfer and experimental validation of a model trained in simulation to a physical robotic platform was achieved, demonstrating preservation of navigation characteristics and policy robustness under real physical constraints and noise.
5. Metrics for evaluating the safety and generalization of reinforcement learning-based models in real environments were developed.
6. A ROS2-based control system was implemented, demonstrating the possibility of reducing hardware complexity through the use of low-cost sensors.

IV. LIST OF PUBLICATIONS RELATED TO THE DISSERTATION THESIS

1. A. Slavova, V. Hristov, “Mapless Navigation with Deep Reinforcement Learning in Indoor Environment,” *International Scientific Conference “TechSys 2025” – ENGINEERING, TECHNOLOGIES AND SYSTEMS*, Plovdiv, Bulgaria, July 2025, DOI: 10.3390/engproc2025100063, http://techsys.tu-plovdiv.bg/ – **Scopus**.
2. A. Slavova and V. Hristov, "Policy Interpretation for Deep Reinforcement Learning", *2025 International Conference Automatics, Robotics and Artificial Intelligence (ICARAI)*, Sozopol, Bulgaria, 2025, pp. 1-4, DOI: 10.1109/ICARAI67046.2025.11137898 – **Scopus**.
3. A. Slavova and V. Hristov, "Application of Reinforcement Learning in Autonomous Mobile Robots", *2024 32nd National Conference with International Participation (TELECOM)*, Sofia, Bulgaria, 2024, pp. 1-4, DOI: 10.1109/TELECOM63374.2024.10812227 – **Scopus**.
4. D. Slavov, V. Hristov and A. Slavova, "Distributed Machine Learning through Transceiver Competitive Connectivity of Remote Computing Systems", *2023 International Scientific Conference on Computer Science (COMSCI)*, Sozopol, Bulgaria, 2023, pp. 1-7, DOI: 10.1109/COMSCI59259.2023.10315948 – **Scopus**.
5. A. Slavova, D. Slavov and V. Hristov, " Research on Computer Vision models for Deep Learning in Autonomous Mobile Robots", *2024 International Conference Automatics, Robotics and Artificial Intelligence (ICARAI)*, Sozopol, Bulgaria, 2024, DOI: 10.1088/1757-899X/1317/1/012011.
6. A. Slavova and D.Slavov, " Task Execution and Dynamic Re-Planning with a Mobile Robot and Manipulator: A Real-Robot Study Using RDK X3 and myCobot 320 – Part 1", *2025 33rd National Conference with International Participation (TELECOM)*, Sofia, Bulgaria, 2025 – **IEEE**.
7. A. Slavova and D.Slavov, " Task Execution and Dynamic Re-Planning with a Mobile Robot and Manipulator: A Real-Robot Study Using RDK X3 and myCobot 320 – Part 2", *2025 33rd National Conference with International Participation (TELECOM)*, Sofia, Bulgaria, 2025 – **IEEE**.

DEEP LEARNING SYSTEMS FOR AUTONOMOUS MOBILE ROBOTS

Anastasiya Slavova

Abstract

This PhD thesis elaborates on the research, design, and implementation of reinforcement learning algorithms for the navigation of autonomous mobile robots.

Chapter 1 provides a comprehensive introduction to the application of deep learning, particularly deep reinforcement learning (DRL), in autonomous mobile robots (AMRs). The chapter highlights the advantages of navigation without pre-built maps or GPS, showing how reinforcement learning allows AMRs to adapt to unknown and dynamic settings using limited sensor input. Several reinforcement learning approaches for AMR navigation are analyzed based on practical and technical criteria such as environmental complexity, learning stability, computational requirements, convergence speed, experience efficiency, robustness, hyperparameter sensitivity, and task applicability. The chapter concludes with a critical analysis of DRL methods, noting strengths such as adaptive behavior without prior maps, and limitations including long training times, high computational costs, sensitivity to hyperparameters, inefficient use of experience, and challenges in transferring models from simulation to real-world environments.

Chapter 2 presents the design and evaluation of reinforcement learning models for autonomous navigation of a wheeled mobile robot in a 2D Flatland simulation integrated with ROS2. The robot, equipped with a LiDAR sensor, is trained to reach a target safely and efficiently, guided by a carefully designed reward function. Four algorithms—DQN, A2C, TRPO, and PPO—are implemented, and their performance is compared using various metrics. The results show a clear progression in performance from value-based methods (DQN, A2C) to policy-based methods (TRPO, PPO), with PPO achieving the best balance of convergence speed, training stability, and navigation quality.

Chapter 3 describes the implementation and training of PPO navigation model for the Yahboom RDK X3 four-wheeled mobile robot in a 3D Gazebo simulation integrated with ROS2. The chapter covers setting up the 3D environment, creating accurate robot and obstacle models, configuring sensors. The PPO algorithm is implemented, with extensive fine-tuning of hyperparameters, neural network architecture, and learning mechanisms. Key improvements include actor network residual connections, adaptive learning rate adjustments based on agent progress, and a memory buffer system that prioritizes recent and successful experiences. These enhancements significantly increase training speed, stability, and performance, raising test episode success rates from 90% to 95% and improving cumulative rewards and convergence speed.

In chapter 4 experimental tests were conducted in a real environment with varying numbers of obstacles (0–6) to evaluate the performance of both the baseline and improved models. Key metrics included episode success rate, trajectory optimality, and number of steps to reach the goal. Results showed that the improved model consistently outperformed the baseline, achieving higher success rates (90% vs. 84%), shorter trajectories, and better robustness to sensor noise. The experiments confirmed that policies trained in simulation can be effectively transferred to real-world scenarios without pre-built maps or GPS, with only minor performance drops due to real-world uncertainties.