УДК 519.6, 51-7

РЕШЕНИЕ ЗАДАЧИ ПЛАНИРОВАНИЯ ТРАЕКТОРИЙ В МНОГОАГЕНТНЫХ СИСТЕМАХ С ПРИМЕНЕНИЕМ МОДЕЛЕЙ МАШИННОГО ОБУЧЕНИЯ

Горина Елизавета Сергеевна[1], Ершов Николай Михайлович[2]

[1]Студент;
Московский государственный университет им. М. В. Ломоносова
Россия, 119991, г. Москва, ул. Ленинские горы, 1;
e-mail: s02220065@gse.cs.msu.ru

[2]Кандидат физико-математических наук, старший научный сотрудник;
Московский государственный университет им. М. В. Ломоносова;
Россия, 119991, г. Москва, ул. Ленинские горы, 1;
e-mail: ershov@cs.msu.ru.

В современных задачах автономной навигации и робототехники критически важным является создание систем управления, способных функционировать в режиме реального времени на устройствах с ограниченной вычислительной мощностью. Работа посвящена децентрализованному планированию траекторий в многоагентных системах с использованием архитектуры резервуарных вычислений. В рамках исследования реализован механизм онлайн-обновления весов выходного слоя нейросети, обеспечивающий адаптацию движения агентов к динамическим препятствиям и целям в реальном времени. Особое внимание уделено межагентному взаимодействию: разработан метод передачи сигналов через изменение визуальных характеристик агентов для координации маневров. Для проверки алгоритмов в среде Python создана имитационная модель, позволяющая оценивать поведение группы роботов при различной плотности окружения. Представлены результаты численных экспериментов и сравнительный анализ разработанного подхода с классическими алгоритмами планирования.

Ключевые слова: многоагентные системы, планирование траекторий, резервуарные вычисления, Echo State Networks, децентрализованное управление, межагентная коммуникация.

Для цитирования:

Горина Е. С., Ершов Н. М. Решение задачи планирования траекторий в многоагентных системах с применением моделей машинного обучения // Системный анализ в науке и образовании. 2026. № 3. С. 106-114. EDN: IEOSXI. URL: https://sanse.ru/index.php/sanse/article/view/738.

SOLVING THE TRAJECTORY PLANNING PROBLEM IN MULTI-AGENT SYSTEMS USING MACHINE LEARNING MODELS

Gorina Elizaveta S.[1], Ershov Nikolay M.[2]

[1]Student;
Lomonosov Moscow State University;
1 Leninskiye Gory, Moscow, 119991, Russia;
e-mail: s02220065@gse.cs.msu.ru.

[2]PhD in Physical and Mathematical Sciences, Senior Researcher;
Lomonosov Moscow State University;
1 Leninskiye Gory, Moscow, 119991, Russia;
e-mail: ershov@cs.msu.ru.

In modern tasks of autonomous navigation and robotics, it is critically important to create control systems capable of operating in real time on devices with limited computing power. The paper is devoted to the decentralized trajectory planning in multi-agent systems using reservoir computing architecture. As part of

Статья находится в открытом доступе и распространяется в соответствии с лицензией Creative Commons «Attribution» («Атрибуция») 4.0 Всемирная (CC BY 4.0) https://creativecommons.org/licenses/by/4.0/deed.ru

the research, a mechanism for online updating of the weights of the output layer of a neural network has been implemented, which ensures the adaptation of agent movement to dynamic obstacles and goals in real time. Special attention is paid to interagent interaction: a method of transmitting signals through changing the visual characteristics of agents to coordinate maneuvers has been developed. To test the algorithms in the Python environment, a simulation model has been created that allows evaluating the behavior of a group of robots at different environmental densities. The results of numerical experiments and a comparative analysis of the developed approach with classical planning algorithms are presented.

Keywords: multi-agent systems, trajectory planning, reservoir computing, Echo State Networks, decentralized management, inter-agent communication.

For citation:

Gorina E. S., Ershov N. M. Solving the trajectory planning problem in multi-agent systems using machine learning models. *System analysis in science and education*, 2026;(3):106-114 (in Russ). EDN: IEOSXI. Available from: https://sanse.ru/index.php/sanse/article/view/738.

Введение

В последние годы многоагентные технологии достигли значительных успехов в широком спектре прикладных задач, включая автоматизацию складской логистики, управление группами беспилотных летательных аппаратов и проведение спасательных операций [1, 2]. Ключевым компонентом таких систем является алгоритм планирования траекторий (*Multi-Agent Path Finding*, *MAPF*), отвечающий за поиск набора бесконфликтных путей для группы агентов из начальных точек в целевые [2]. Процесс разработки систем управления для мобильных роботов включает несколько ключевых стратегий. Централизованное планирование обеспечивает строгие гарантии безопасности, однако сталкивается с проблемой экспоненциального роста вычислительной сложности при увеличении числа агентов [2]. Децентрализованное управление, напротив, позволяет агентам принимать решения на основе локального восприятия среды, что критически важно для обеспечения масштабируемости системы [3, 4, 5].

Настоящая работа посвящена децентрализованному планированию траекторий в многоагентных системах с использованием архитектуры резервуарных вычислений (*Echo State Networks*, *ESN*) [6, 7]. Применение эхо-сетей представляет собой перспективную альтернативу, так как эта архитектура обладает способностью обрабатывать временные последовательности данных при крайне низких затратах на обучение и выполнение [7]. Это открывает возможность реализации интеллектуального поведения на встраиваемых системах, где использование глубоких нейронных сетей может быть затруднено из-за аппаратных ограничений [1, 8].

1. Математическая постановка задачи планирования траекторий

Рассматривается непрерывная рабочая область $\Omega \subset \mathbb{R}^2$, в которой функционирует группа из N автономных агентов. Каждый агент $i \in 1, \dots, N$ в момент времени t характеризуется набором динамических параметров, к которым относятся положение (вектор координат центра агента $x_i(t) \in \Omega$) и скорость (вектор $v_i(t)$, модуль которого ограничен предельно допустимым значением $||v_i(t)|| \leq v_{max}$). Агенту последовательно назначаются целевые точки из заданного сценария. Локальная цель $g_i(t)$ считается достигнутой, если агент оказывается в ее малой окрестности:

$$||x_i(t) - g_i(t)|| < \epsilon$$

где ϵ — заданный порог точности. После выполнения этого условия агенту немедленно назначается новая цель, что позволяет исследовать потоковую эффективность системы на длинных временных промежутках.

В области Ω задано множество неподвижных препятствий $\mathcal{O} = \{O_k\}_{k=1}^{M}$. Физический контакт с объектами или другими агентами крайне нежелателен и минимизируется через систему штрафов. В модели выделяются два типа нежелательных событий:

1. Столкновение: Факт пересечения геометрии агента с препятствием O_k или телом другого агента j. Каждое такое событие фиксируется счетчиком C_i.
2. Опасное сближение: Нахождение агента в буферной зоне вблизи границ объектов, что формирует накопительный штраф P_i.

Для оценки эффективности работы алгоритма и формирования сигнала обучения встраивается обобщенная функция стоимости J, определяемая для всей группы агентов:

$$J = \sum_{i=1}^{N} (T_i + \lambda_1 C_i + \lambda_2 P_i),$$

где T_i – суммарное время, затраченное на прохождение маршрута, C_i – количество столкновений в процессе движения, P_i – штрафные баллы за нарушение безопасной дистанции, а λ_1, λ_2 – весовые коэффициенты, определяющие строгость алгоритма в отношении безопасности.

Задача планирования траекторий формулируется как поиск децентрализованной стратегии управления, которая для каждого агента в реальном времени отображает вектор текущих локальных наблюдений $u_i(t)$ в вектор управляющих воздействий $y_i(t)$ (приращение скорости):

$$y_i(t) = F\big(u_i(t)\big)$$

2. Резервуарные вычисления и модель Echo State Network

Резервуарные вычисления (*Reservoir Computing*) – это подход к разработке рекуррентных нейронных сетей, основанный на использовании динамической системы (резервуара) для обработки временных последовательностей данных [6, 7]. *Echo State Network* (*ESN*) отличается упрощенным процессом обучения, в котором веса входного слоя W_{in} и веса рекуррентных соединений внутри резервуара W остаются случайными и фиксированными, а обучению подвергаются исключительно веса выходного слоя W_{out}, что существенно снижает вычислительные затраты [7].

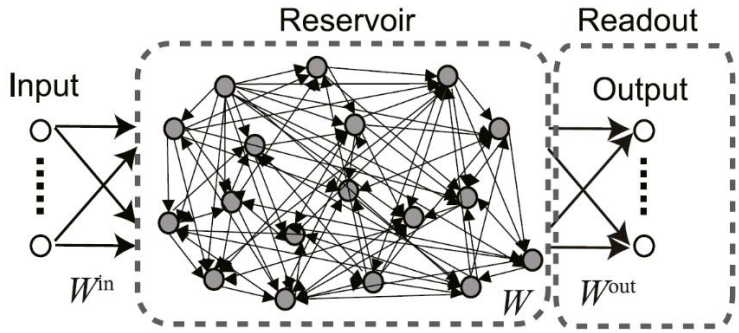


Рис. 1. Архитектура резервуарной нейронной сети

Обновление внутреннего состояния резервуара $x(n)$ и выходного вектора $y(n)$ на дискретном шаге времени n описывается следующей системой уравнений [7]:

$$\tilde{x}(n) = f(W \cdot x(n-1) + W_{in} \cdot [1; u(n)])$$

$$x(n) = (1-\alpha) \cdot x(n-1) + \alpha \cdot \tilde{x}(n)$$

$$y(n) = W_{out} \cdot [1; x(n)],$$

где $x(n) \in \mathbb{R}^N$ – вектор состояния резервуара, $\tilde{x}(n) \in R^N$ – обновление резервуара, $W \in \mathbb{R}^{N \times N}$ – матрица рекуррентных связей, α – коэффициент утечки (leaking rate), $W_{in} \in \mathbb{R}^{N \times (K+1)}$ – матрица входных весов, $[1; u(n)] \in \mathbb{R}^{K+1}$ – вертикальная конкатенация смещения и вектора входного сигнала, f – нелинейная функция активации (гиперболический тангенс), а $W_{out} \in \mathbb{R}^{M \times (N+1)}$ – выходная матрица весов [7].

Обновление весов выходного слоя в процессе онлайн-обучения агента для адаптации к динамическим препятствиям выполняется на основе алгоритмов градиентного спуска по следующей формуле [7]:

$$W_{out} = W_{out} - \eta \cdot r \cdot \nabla_{W_{out}} E = W_{out} - \eta \cdot r \cdot 2es_b^{\top},$$

где $s_b = [1; x(n)] \in \mathbb{R}^{N+1}$ – расширенное состояние резервуара со сдвигом, $y \in \mathbb{R}^M$ – предсказание сети, $d \in \mathbb{R}^M$ – желаемое целевое направление движения, $e = y - d$ – ошибка предсказания, $\eta > 0$ – коэффициент обучения (learning rate), r – внешняя скалярная награда от среды (reward), а E – минимизируемая целевая функция ошибки [7].

3. Проектирование децентрализованного управления и межагентного взаимодействия

Для эффективного принятия решений в реальном времени агент формирует 19-мерный вектор локальных признаков $u(n) \in \mathbb{R}^{19}$, который включает в себя: координаты текущего положения и цели, текущий вектор скорости, координаты ближайшего препятствия, коммуникационные сигналы, дистанции до физических объектов, измеренные лучами в секторе обзора агента в диапазоне 90°. Выходной слой нейросети ESN формирует вектор управления $y(n) \in \mathbb{R}^5$, в котором первые две компоненты задают вектор приращения скорости движения, а три последующие – визуальный отклик агента (сигнал RGB), транслируемый в окружающую среду для соседей.

Механизм коммуникации реализован полностью децентрализованно и автономно: агенты не обмениваются между собой точными пространственными координатами или глобальными траекториями напрямую. Вместо этого взаимодействие организовано посредством трансляции текущего состояния резервуара эхо-сети в форме неявного визуального сигнала в формате *RGB*, который транслируется в окружающую среду. Для этого каждый робот в радиусе своей локальной видимости считывает цветовые параметры соседей, формируя 6 коммуникационных признаков входного вектора (средние и максимальные значения по каждому цветовому каналу). В процессе функционирования целевые значения цвета формируются динамически на основе безопасности и успеха агента: канал *R* (красный) принимает высокие значения при критическом сближении с объектами, выступая сигналом тревоги; канал *G* (зеленый) ассоциируется с успешным продвижением к цели, а канал B (синий) отражает стабильность текущей стратегии движения.

Поскольку веса выходного слоя W_{out} обучаются одновременно для генерации команд движения и изменения цвета, между физической траекторией и визуальным сигналом устанавливается устойчивая математическая связь. В результате окружающие агенты, считывая *RGB*-параметры соседа, получают прямую информацию о векторе его состояния и намерениях, что позволяет им упреждающе координировать совместные маневры в узких проходах и эффективно предотвращать возникновение взаимных блокировок.

4. Программная реализация и результаты экспериментов

Для проверки алгоритмов в среде *Python* создана модульная имитационная модель, включающая расчет физических коллизий, менеджер сценариев и управляющие скрипты. Также для сравнительного анализа дополнительно были обучена модель на базе алгоритма обучения с подкреплением *PPO* и модель реактивного поведения.

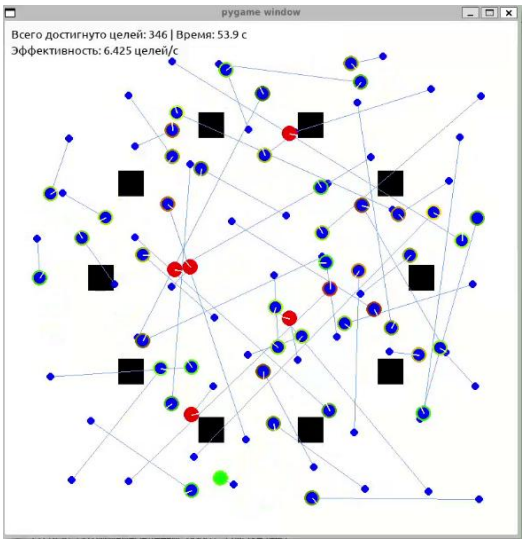


Рис. 2. Модульный симулятор

Проведены численные эксперименты для различных навигационных сценариев: встречные и перекрестные потоки агентов, прохождение узких проходов и зон со сложной конфигурацией препятствий.

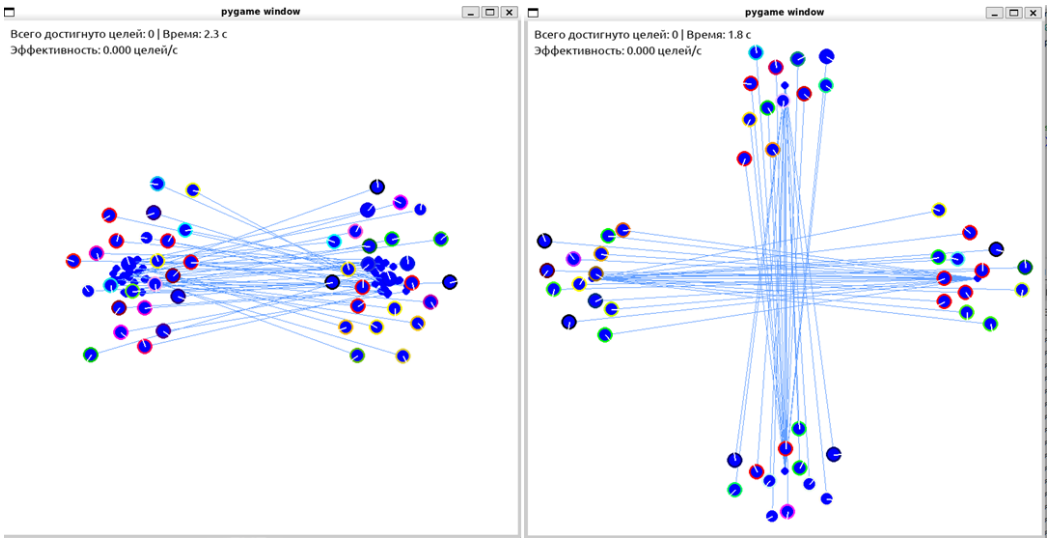


Рис. 3. Миссии с встречными и перекрестными потоками агентов

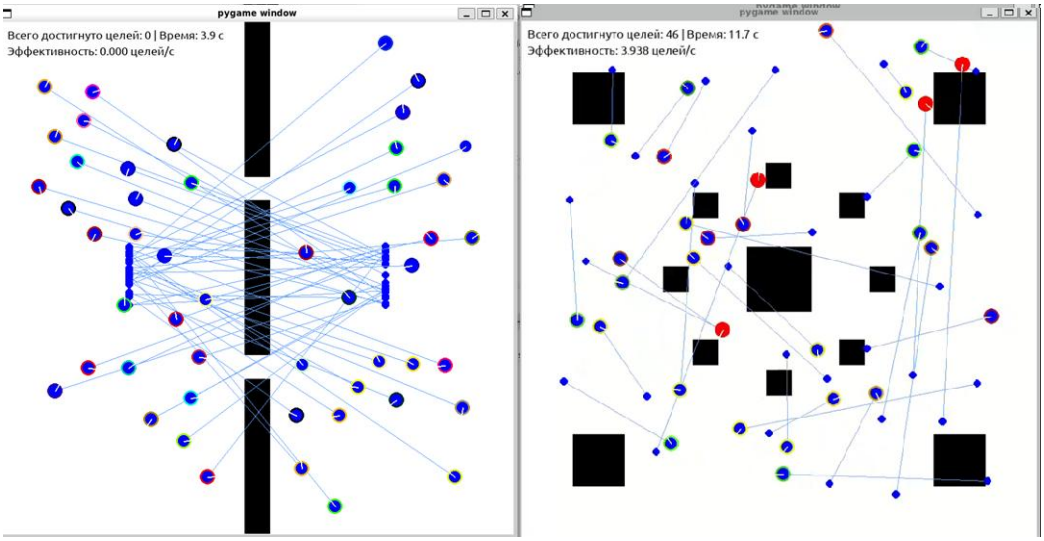


Рис. 4. Миссии с узкими проходами и сложной конфигурацией препятствий

Численные результаты показали, что разработанный подход на базе *ESN* характеризуется минимальной вычислительной нагрузкой, так как время принятия решений обученной моделью крайне низкое и сводится к линейным операциям над весами выходного слоя. Благодаря распределенной обработке данных добавление новых участников не вызывает экспоненциального роста нагрузки, что обеспечивает высокую масштабируемость системы. Наличие рекуррентных связей в *ESN* позволило успешно избегать тупиков и взаимных блокировок в узких проходах, которые характерны для классических реактивных методов.

При этом сравнительный анализ эффективности выявил выраженную зависимость результатов от топологии навигационного сценария. В миссиях на открытом пространстве (например, в сценарии с интенсивными встречными горизонтальными потоками) классический алгоритм обучения с подкреплением *PPO* удерживает лидерство по абсолютному количеству успешно достигнутых целей за зафиксированный отрезок времени. Это обусловлено способностью *PPO* выстраивать глобально оптимизированные траектории на свободных участках, в то время как *ESN*-подход в подобных условиях идет вторым, демонстрируя сопоставимую, но чуть меньшую финальную продуктивность.

Табл. 1. Количество достигнутых целей в миссии со встречными потоками агентов

Время (сек)	PPO	ESN	ESN_simple	Baseline
50	94	94	65	26
100	266	231	173	81
150	461	390	295	137
200	629	531	405	175

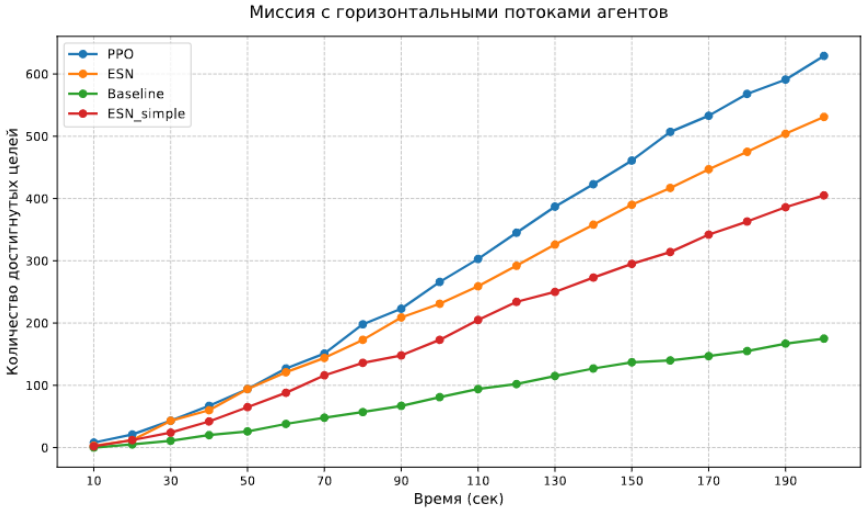


Рис. 5. График зависимости количества достигнутых целей от времени в миссии со встречными потоками агентов

Табл. 2. Количество достигнутых целей в миссии с перекрестными потоками агентов

Время (сек)	PPO	ESN	ESN_simple	Baseline
50	67	43	11	34
100	181	149	55	111
150	321	244	110	186
200	422	353	154	269

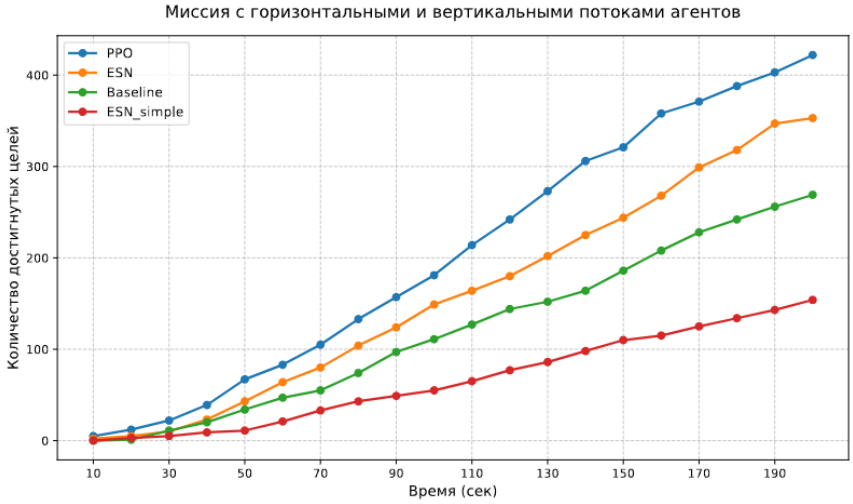


Рис. 6. График зависимости количества достигнутых целей от времени в миссии с перекрестными потоками агентов

В сложных навигационных сценариях (характеризующихся высокой плотностью скопления роботов, заторами и стесненными пространственными условиями) разработанный алгоритм на базе *ESN* продемонстрировал результаты, превосходящие базовые модели, а в некоторых конфигурациях даже превзошел алгоритм *PPO* по среднему темпу достижения целей. В условиях жестких геометрических ограничений и узких мест *PPO* может сталкиваться с проблемой неэффективного исследования среды и избыточной нестабильности траекторий, тогда как динамическая память резервуара *ESN* позволяет сохранять плавность хода и непрерывность потока.

Табл. 3. Количество достигнутых целей в миссии с узкими проходами

Время (сек)	PPO	ESN	ESN_simple	Baseline
50	25	31	18	4
100	58	64	52	10
150	92	101	81	17
200	114	124	103	22

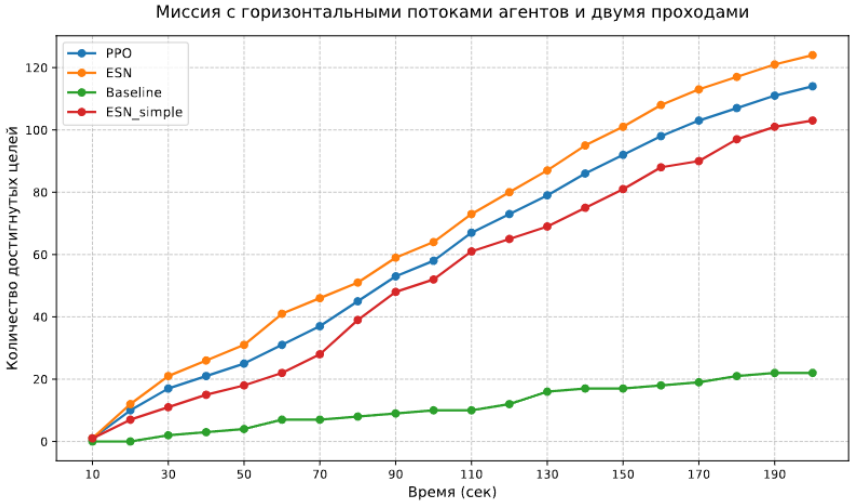


Рис. 7. График зависимости количества достигнутых целей от времени в миссии с узкими проходами

Табл. 4. Количество достигнутых целей в миссии со сложной конфигурацией препятствий

Время (сек)	PPO	ESN	ESN_simple	Baseline
50	100	153	129	79
100	196	319	257	157
150	267	515	381	225
200	313	745	512	285

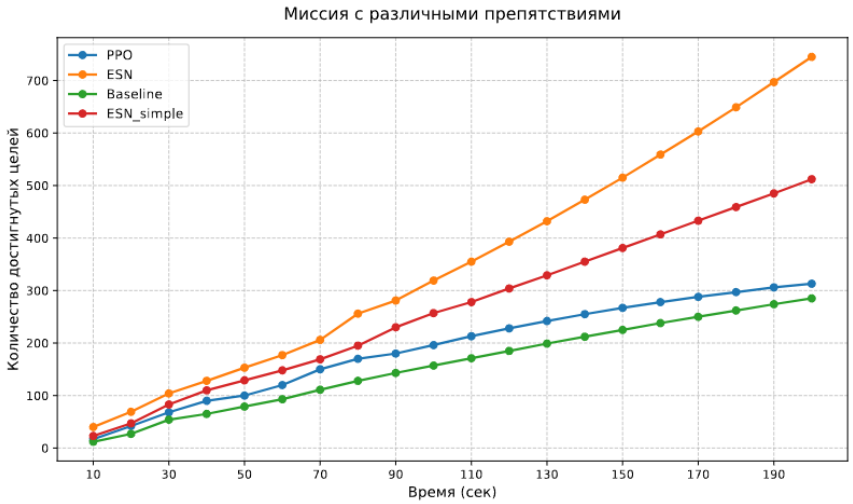


Рис. 8. График зависимости количества достигнутых целей от времени в миссии со сложной конфигурацией препятствий

Эксперименты также наглядно доказали, что внедрение механизма межагентной коммуникации оказывает исключительно положительное влияние на общую эффективность системы. Использование неявных цветовых *RGB*-сигналов позволило существенно повысить пропускную способность всей группы агентов по сравнению с изолированно действующей конфигурацией. Считывание визуальных параметров соседей позволяет роботам инициировать маневры уклонения упреждающе, еще до фиксации угрозы собственным лидаром, что радикально снижает время простоя в заторах и минимизирует риски взаимной блокировки.

Заключение

В работе успешно реализован и исследован децентрализованный метод планирования траекторий многоагентных систем на базе *Echo State Networks*. Разработанный алгоритм цветовой неявной коммуникации показал свою эффективность в задачах координации совместного движения в плотных средах. Проведенные эксперименты подтверждают масштабируемость системы и низкую вычислительную трудоемкость разработанной архитектуры.

К направлениям будущих исследований могут относится:

- Масштабирование разработанного метода на трехмерные пространства для управления группами беспилотных летательных аппаратов.
- Анализ робастности и устойчивости алгоритма трансляции визуальных сигналов в условиях зашумления каналов связи, неполной видимости среды и оптических помех
- Теоретическое и экспериментальное исследование пространств параметров резервуара эхо-сети.

Список источников

1. Learn to Follow: Decentralized Lifelong Multi-agent Pathfinding via Planning and Learning / A. Skrynnik, A. Andreychuk, M. Nesterova [et al.] // arXiv : [electronic archive]. – arXiv:2310.01207v1 [cs.AI]. – https://doi.org/10.48550/arXiv.2310.01207. – Submitted on 3 oct 2023.
2. Silver D. Cooperative pathfinding /// Proceedings of the First AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment (1 – 2 Jun, 2005 in Marina Del Rey, California USA). – Menlo Park, California: AAAI Press, 2005. – Vol. 1. – No. 1. – P. 117-122. – https://doi.org/10.1609/aiide.v1i1.18726.
3. Environment induced emergence of collective behaviour in evolving swarms with limited sensing / F. Van Diggelen, J. Luo, T. A. Karagüzel [et al.] // arXiv : [electronic archive]. – arXiv:2203.11585. – https://doi.org/10.48550/arXiv.2203.11585. – Submitted on 22 mar 2022.
4. Empirical assessment and comparison of neuro-evolutionary methods for the automatic off-line design of robot swarms / K. Hasselmann, A. Ligot, J. Ruddick, M. Birattari // Nature Communications. – 2021. – Vol. 12. – P. 4345. – https://doi.org/10.1038/s41467-021-24642-3.
5. Reciprocal n-body collision avoidance / J. Van den Berg, S. M. Guy, M. Lin, D. Manoch // Robotics Research: The 14th International Symposium of Robotics Research ISRR (Lucerne, Switzerland, 31.08 – 03.09 2009). - Berlin, Heidelberg : Springer Berlin Heidelberg, 2011. – P. 3-19. – https://doi.org/10.1007/978-3-642-19457-3_1.
6. Recent advances in physical reservoir computing / G. Tanaka, T. Yamane, J. B. Héroux [et al.] // Neural Networks. – 2019. – Vol. 115. – P. 100-123. – https://doi.org/10.1016/j.neunet.2019.03.005.
7. Lukoševičius M. A Practical Guide to Applying Echo State Networks / ed. By G. Montavon, G. D. Orr // Neural Networks: Tricks of the Trade, Reloaded. – Berlin, Heidelberg: Springer, 2012. – P. 659-686. – https://doi.org/10.1007/978-3-642-35289-8_36.
8. Sutton R. S. Reinforcement Learning: An Introduction / R. S. Sutton, A. G. Barto. – Second edition. – Cambridge, Massachusetts; London, England: MIT Press, 2018. – 552 p.