

## Оригинальная статья / Original article

<https://doi.org/10.21869/2223-1560-2019-23-6-115-132>

## Настройка гиперпараметров искусственной нейронной сети для навигации мобильной платформы

Д. М. Дударенко<sup>1</sup> ✉, П. А. Смирнов<sup>1</sup>

<sup>1</sup> Санкт-Петербургский институт информатики и автоматизации Российской академии наук  
14 линия В.О., д. 39, Санкт-Петербург, 199178, Российская Федерация

✉ e-mail: dmitry@dudarenko.net

### Резюме

**Цель исследования.** Основной целью работы является повышение эффективности работы нейросетевой модели при навигации мобильной робототехнической платформы в статических и динамически сгенерированных средах.

**Методы.** Для решения поставленной задачи были предложены точная настройка и оптимизация гиперпараметров нейронной сети. Для стимулирования агентов исследовать окружающую среду была проведена корректировка системы вознаграждений, предполагающая повышение вознаграждения при уменьшении расстояния от агента до целевой точки и увеличение штрафа при движении в направлении, противоположном конечной точке, и при прохождении каждой последующей сцены. Такое распределение вознаграждений и штрафов побуждает агентов активно обучаться и способствует сокращению общего количества сцен. С целью уменьшения объема данных, обрабатываемых нейронной сетью, была введена нормализация входных векторов. Было уменьшено время обучения модели нейронной сети благодаря параллельному обучению агентов и, как следствие, увеличения опыта в результате исследования окружающей среды.

**Результаты.** Предложенный подход позволил сократить время обучения на 30% и повысить эффективность навигации мобильной платформы на 10% в динамически сгенерированной среде и на 22% в статической среде по сравнению с неоптимизированной моделью.

**Заключение.** Предложенное решение может быть использовано совместно с другими методами трассировки и навигации, когда обученная нейронная сеть работает одновременно с уже отработанными и проверенными алгоритмами навигации. Например, если мобильная платформа подключает обученную нейронную сеть только для корректировки положения в пространстве и предотвращения столкновений с другими объектами.

**Ключевые слова:** мобильный робот; навигация; искусственные нейронные сети; обучение с подкреплением.

**Конфликт интересов:** Авторы декларируют отсутствие явных и потенциальных конфликтов интересов, связанных с публикацией настоящей статьи.

**Для цитирования:** Дударенко Д. М., Смирнов П. А. Настройка гиперпараметров искусственной нейронной сети для навигации мобильной платформы // Известия Юго-Западного государственного университета. 2019; 23(6): 115-132. <https://doi.org/10.21869/2223-1560-2019-23-6-115-132>.

Поступила в редакцию 03.10.2019

Подписана в печать 26.11.2019

Опубликована 23.12.2019

# Setting Artificial Neural Network Hyperparameters for Mobile Platform Navigation

Dmitry M. Dudarenko <sup>1</sup> ✉, Petr A. Smirnov <sup>1</sup>

<sup>1</sup> St. Petersburg Institute for Informatics and Automation of the Russian Academy of Sciences, 14-th Linia VI, 39, St. Petersburg 199178, Russian Federation

✉ e-mail: dmitry@dudarenko.net

## Abstract

**Purpose of research.** The main purpose of this work is to increase the efficiency of a neural network model when navigating a mobile robotic platform in static and dynamically generated environments.

**Methods.** To solve this problem, precise setting and optimization of neural network hyperparameters were proposed. In order to encourage agents to explore the environment, the reward system was adjusted to increase the reward when the distance from the agent to the target point was reduced, and the penalty increased when moving in the opposite direction to the end point and passing each subsequent scene. This distribution of rewards and penalties encourages agents to learn actively and helps to reduce the total number of scenes. In order to reduce the amount of data processed by a neural network, normalization of input vectors was introduced. The learning time of the neural network model was reduced due to the parallel training of agents and, consequently, increased experience as a result of the environmental research.

**Results.** The proposed approach reduced the learning time by 30% and improved the navigation efficiency of the mobile platform by 10% in a dynamically generated environment and by 22% in a static environment compared to the non-optimized model.

**Conclusion.** The proposed solution can be used in conjunction with other methods of tracing and navigation, when the taught neural network works simultaneously with the already developed and proven navigation algorithms, for example, if the mobile platform connects a taught neural network only to adjust the position in space and to prevent collisions with other objects.

**Keywords:** mobile robot; navigation, artificial neural networks; reinforcement learning.

**Conflict of Interest:** The authors declare the absence of overt and potential conflicts of interest related to the publication of this article.

**For citation:** Dudarenko D. M., Smirnov P. A. Setting Artificial Neural Network Hyperparameters for Mobile Platform Navigation. *Izvestiya Yugo-Zapadnogo gosudarstvennogo universiteta = Proceedings of the Southwest State University*. 2019, 23(6): 115-132 (In Russ.). <https://doi.org/10.21869/2223-1560-2019-23-6-115-132>.

Received 03.10.2019

Accepted 26.11.2019

Published 23.12.2019

\*\*\*

## Введение

Мобильные робототехнические платформы и системы навигации ежедневно находят все большее применение в повседневной жизни. Их используют на складах, в торговых центрах, муниципальных, правительственных учреждениях и аэропортах [1, 2]. Платформа,

оснащенная навигационным программным обеспечением, может манипулировать объектами, собирать данные об окружающей среде, проверять безопасность маршрута для человека. Такая мобильная платформа и ее программное обеспечение должны отвечать определенным требованиям: иметь возмож-

ность обнаруживать динамические и статические препятствия, автономно анализировать объекты и обрабатывать большие данные, даже без подключения к сети. Использование бортовых видеокамер может быть не всегда оправдано из-за достаточно низкой скорости обработки изображений и видеосигнала, перспективных искажений. В таком случае решением является использование лидара. Лидар может быть использован как основной, так и как вспомогательный блок для измерительной системы робототехнической мобильной платформы. Благодаря отсутствию геометрических искажений, высокой производительности и точности измерения расстояния он является отличной заменой для видеокамеры.

Для решения задачи беспрепятственного движения мобильных платформ в стохастической среде ранее были разработаны системы навигации на основе двумерного лидара [3] и алгоритмы [4], обеспечивающие передвижение робототехнического средства к целевой точке в статическом и динамическом окружении [5]. Данная система основана на модели нейронной сети и позволяет учитывать присутствие статических и динамических препятствий на пути следования робота. Использование нейронной сети в динамически сгенерированном пространстве позволило повысить эффективность перемещения платформы, сделать алгоритм более адаптивным, по сравнению с классическими алгоритмами навигации.

Однако для разработанной системы были характерны застревания на месте. Поэтому целью работы является повышение эффективности системы навигации мобильной платформы. Данная задача была решена при помощи оптимизации и тонкой настройки гиперпараметров модели нейронной сети, корректировки системы вознаграждений и параллельного обучения моделей сразу нескольких мобильных платформ (агентов) на одной сцене.

Одной из важнейших задач при проектировании мобильных робототехнических средств является разработка системы навигации. Существует множество исследований, нацеленных на разработку методов и алгоритмов уточнения карты местности, движения по сложным траекториям, планирования движения с учетом присутствия препятствий на пути следования [6]. Перспективным направлением в данной научной области является использование искусственных нейронных сетей. Применение методов глубокого обучения обеспечивает плавность маневров роботов благодаря отслеживанию положения предметов в неизвестной окружающей среде [7]. В таком случае данные об окружающем пространстве могут быть получены посредством 2D-сканирования. Данные лидара поступают на вход сверточной нейронной сети, обученной при помощи метода Q-обучения, для последующей обработки. Такой подход позволяет мобильному роботу распознавать препятствия и избегать столкновения с ними при движе-

нии к целевой точке в загроможденном пространстве.

Принцип работы систем навигации на основе данных лидара базируется на определении следующего движения робота исходя только из значения расстояния от него до препятствия [8, 9]. Частые колебания измеряемого расстояния являются причиной резкого изменения скорости, что влечет за собой механические удары и неэффективное использование движущего момента двигателей. Данная проблема может быть решена при помощи использования буфера воспроизведения для хранения выходных данных сети [10]. То есть последние действия сохраняются в памяти и затем снова передаются на вход. Результаты тестирования, проведенного в симуляторе ROS-GAZEBO и в реальных условиях, показали более высокую точность такого подхода по сравнению с Leg Detector (LD), стандартным решением для роботов на основе операционной системы роботов (ROS).

Традиционный подход к навигации робота состоит из трех этапов. Первый этап подразумевает извлечение объектов из сцены, полученной при помощи камеры. Второй шаг заключается в определении текущей позиции робота с помощью классификатора извлеченных объектов. На последнем шаге определяется направление движения либо вручную, либо посредством обучения модули нейронной сети. В отличие от традиционного многошагового метода авторы работы [11] разработали визуально-

моторную навигационную систему на основе сверточной нейронной сети глубокого сквозного (End-to-end) обучения, которая позволяет определять линейные и угловые скорости робота за один шаг на основе входного изображения. Результаты экспериментов показали среднюю ошибку линейной скорости 2,2 см/с и среднюю ошибку угловой скорости 3,03 град/с. Предложенная система навигации может быть использована в реальных условиях и не требует применения дополнительных датчиков: лидара, радара, инфракрасной камеры, GPS.

Для повышения эффективности навигации мобильного робота в сложной среде модель нейронной сети может быть построена с учетом кинематических и динамических характеристик робототехнического устройства [12]. Например, в работе [13] предложена система навигации и построения карты окружающей среды на основе муравьиного алгоритма, которая учитывает переменную скорость движения мобильного робота. Такой подход позволяет робототехническому устройству снижать скорость в непосредственной близости к препятствиям и двигаться с высокой скоростью в открытых пространствах.

Навигация мобильной платформы осложняется отсутствием данных об окружающей среде особенно, в условиях пересеченной местности [14]. Для решения этой задачи авторы работы [15] разработали нейронную сеть глубокого обучения с подкреплением. Нейронная сеть получает в качестве вход-

ных данных необработанную информацию с датчиков, выходными данными модели являются локальные движения робота. Работоспособность предложенной модели была доказана при тестировании в уникальных динамически генерируемых трехмерных средах с различными размерами и уровнями проходимости.

Альтернативный подход при разработке систем управления заключается в использовании адаптивной сети на основе системы нечеткого вывода (adaptive neuro-fuzzy inference system, ANFIS) [17]. В этом случае для выбора параметров оценки используется генетический алгоритм, который генерирует данные посредством случайного подбора и комбинирования. Несмотря на то, что системы навигации на основе ANFIS могут быть встроены в малогабаритные платформы для движения в статическом пространстве [18], в загроможденных средах [19], данные системы не учитывают необходимость движения устройства в среде с динамическими препятствиями. Экспериментальные данные, представленные в [18, 19], демонстрируют высокую эффективность предложенного подхода только при передвижении на небольшие расстояния среди статических препятствий. Тем не менее, в среде с динамическими препятствиями, например в социо-киберфизических системах, скорость достижения роботом целевой точки не является ключевым показателем, что связано с: 1) массогабаритными

показателями робота; 2) обеспечением безопасности людей и/или животных, которые могут находиться на пути робота и также будут восприниматься им как динамические препятствия.

Представленные в обзоре исследования демонстрируют эффективность применения искусственных нейронных сетей глубокого обучения при разработке систем навигации мобильных роботов. Такой подход не нуждается в большом объеме размеченных входных данных и обеспечивает плавность движений робота. Использование двумерного лидара в качестве основного средства очувствления робота позволяет отслеживать положение других объектов в сцене. При повышении эффективности системы управления движением платформы за счет учета механических параметров робота необходимы дополнительные программные и аппаратные средства. В свою очередь, повышение эффективности навигации при помощи тонкой настройки гиперпараметров достигается путем оптимизации модели нейронной сети и не требует увеличения объема входных данных и дополнительных датчиков.

## Материалы и методы

Моделирование стохастической среды для действий робота проводилось на платформе Unity. В качестве статичной среды была выбрана 3D-модель одного из этажей здания научного учреждения СПИИРАН. 3D-модель включает в себя 33 комнаты, лестницы, проходы, коридоры

доры, двери. Динамически генерируемая среда представляет собой помещения с произвольной геометрией, которые были смоделированы автоматически в режиме реального времени. Основными средствами навигации платформы являются: лидар, закрепленный на платформе, и предварительно обученная модель нейронной сети. Траектория движения платформы строится по координатам, углам поворота и данным, считываемым с лидара с учетом подвижных и неподвижных объектов, встречающихся на пути следования. В качестве входных данных нейронной сети использовались данные, получаемые с лидара, координаты мобильной платформы, а также координаты начальной и конечной точек маршрута. Выходными данными модели являются координаты следующего движения робота. Для обучения модели нейронной сети использовался плагин Unity ML и библиотека TensorFlow. Для обучения модели использовался метод обучения с подкреплением, так как он поддерживается средой Unity напрямую и не требует большого объема размеченных данных. В этом случае мобильный робот (агент) обучается, взаимодействуя с некоторой окружающей средой через сигналы подкрепления [16]. Подобный подход может оказаться особенно эффективен при навигации без карты [17]. Одним из главных этапов работы по оптимизации системы навигации является более тонкая настройка гиперпарамет-

ров модели используемой нейронной сети. Различают следующие гиперпараметры:

- Параметр регуляризации  $\lambda$  (lambda) — используется при расчете общей оценки преимущества (generalized advantage estimation, GAE) и отражает степень доверия текущему значению функции стоимости при определении нового значения. Малые значения  $\lambda$  соответствуют сильной зависимости от текущей величины вознаграждения, полученной при исследовании роботом окружающей среды. Настройка  $\lambda$  позволяет достичь компромисса в определении значимости текущего и нового значений функции стоимости и повысить эффективность процесса обучения.

- Количество скрытых слоев.

- Количество эпизодов обучения.

Эпизод – итерация моделирования, то есть прохождения роботом пути от начальной до целевой точки.

- $\varepsilon$  (epsilon) — соответствует допустимому значению расхождения между старой и новой политикой в каждой итерации градиентного спуска. Чем меньше значение  $\varepsilon$ , тем большее время займет обучение.

- Временной горизонт (time horizon) – регулирует количество батчей (batches), которые нужно получить каждому агенту, прежде чем добавлять его в буфер опыта.

- Сила регуляризации энтропии  $\beta$  (beta) – отвечает за степень случайности обучения.

- Сила любопытства (curiosity strength) – отвечает за стимулирование исследования окружающего пространства.
- Количество эпох обучения.

## 1. Настройка гиперпараметров

Для повышения эффективности работы системы навигации агентов и уменьшения времени исследования окружающей среды была проведена настройка следующих гиперпараметров модели нейронной сети: числа скрытых слоев, временного горизонта, силы регуляризации энтропии  $\beta$ , силы любопытства и количества эпох обучения. С целью обеспечения баланса между скоростью обучения и качеством модели нейронной сети, количество скрытых слоев было уменьшено с 256 до 128. Изменение числа скрытых слоев позволяет контролировать количество нейронов в каждом полносвязном слое. Количество эпох (т.е. количество проходов через буфер опыта в каждой итерации градиентного спуска) было уменьшено с 8 до 3. Для возможности фиксации важных действий агента, таких как столкновения с другими объектами и застревание на месте, при частых вознаграждениях параметр «временной горизонт» был увеличен с 32 до 1000. Чтобы остановить непрерывный рост энтропии при обучении, параметры силы любопытства были отключены [20], а сила регуляризации энтропии  $\beta$  уменьшена с 0,005 до 0,001. Это, в свою очередь, гарантирует правильное исследование роботом пространства во время обучения.

На рис. 1 и 2 представлены графики изменения энтропии во время обучения нейронной сети для неоптимизированной и оптимизированной моделей соответственно.

Энтропия неоптимизированной модели увеличивается в течение обучения, что свидетельствует о том, что при попытке избежать столкновения с другими объектами платформа не достигает целевой точки. Уменьшение параметра способствует поощрению той политики агента, которая стимулирует его к изучению окружающего пространства. Благодаря отключению параметра силы любопытства и оптимизации  $\beta$  удалось достичь постепенного уменьшения энтропии.

С целью систематизации формы представления входных данных нейронной сети было уменьшено количество векторов координат точек пространства. Вместо координат начальной и целевой точек в виде  $(x, y, z)$  использовался трехмерный вектор. Такое решение позволило сократить объем обрабатываемых нейронной сетью данных.

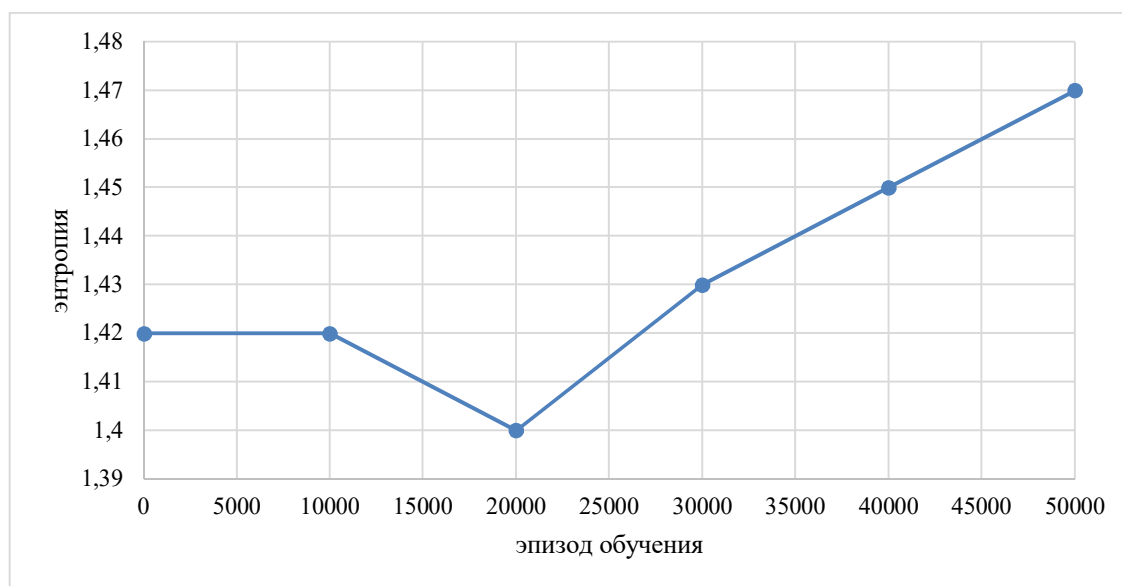
## 2. Корректировка системы распределения вознаграждений

Для оптимизации системы распределения вознаграждений было предложено несколько решений. Для обеспечения более плавного движения платформы среди динамических и статических объектов был исключен модуль выдачи вознаграждений за сближение с динамическими и статическими объектами. Данный модуль штрафует аген-

та  $(-0,01f)$  или назначал вознаграждение  $(0,001f)$ , если расстояние между лучами лидара и динамическими или статическими объектами было меньше 1 метра (в единицах измерения Unity).

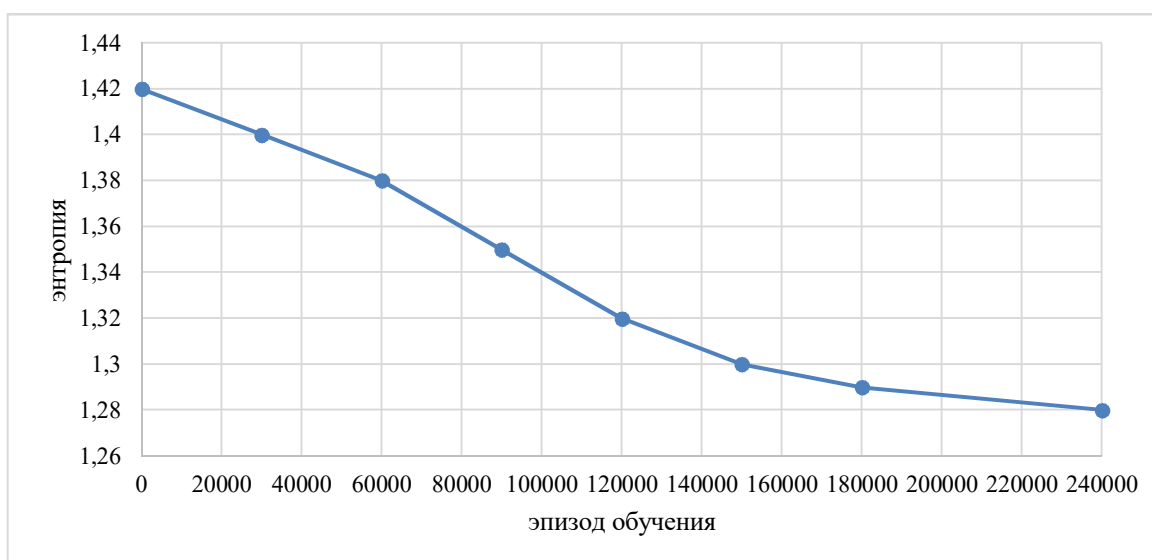
Кроме того, было увеличено вознаграждение агента с  $0,25f$  до  $0,4f$  за

уменьшение расстояния до конечной точки. Для предотвращения движения агента в сторону, противоположные целевой точке был также увеличен  $0,1f$  до  $0,2f$ . Штраф за каждую следующую пройденную платформой сцену был установлен на уровне  $-0,005f$ .



**Рис. 1.** Энтропия неоптимизированной модели нейронной сети

**Fig. 1.** Entropy of a non-optimized neural network model



**Рис. 2.** Энтропия оптимизированной модели нейронной сети

**Fig. 2.** Entropy of an optimized neural network model



Подобное распределение вознаграждений побуждает платформу более активно обучаться на сцене и стремиться к сокращению общего количества сцен.

### 3. Параллельное обучение агентов

Время обучения нейронной сети напрямую зависит от объема полученного опыта в исследовании окружающей среды. Поэтому для сокращения времени обучения было увеличено количество агентов, одновременно находящихся на сцене. Среда Unity позволяет добавлять практически неограничен-

ное количество агентов на сцену. Тем не менее, большое количество агентов значительно увеличит время завершения сцены, так как для завершения обучения необходимо завершение всех действий каждого агента. Оптимальным балансом является использование не более 5 или 10 агентов одновременно. В настоящем исследовании было использовано 5 агентов для динамически сгенерированной среды и 5 агентов для смоделированной среды. На рис. 3 представлен процесс параллельного обучения нескольких агентов.

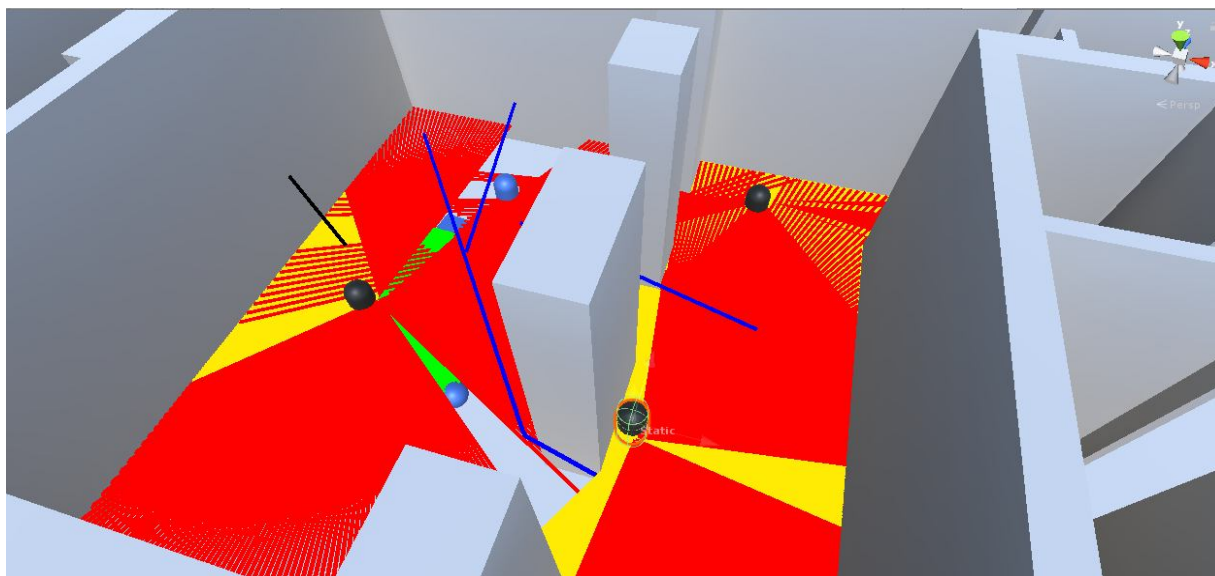


Рис. 3. Параллельное обучение агентов

Fig. 3. Parallel training of agents

На рисунке показаны лучи лидара: зеленого цвета — до динамических препятствий, желтого — до статических препятствий, лучи красного цвета означают отсутствие препятствий в пределах дальности измерений (5-10 м). Каждый агент движется по случайной траектории с переменной скоростью и внезапными остановками, что имитирует движение объектов в реальной среде. Агенты исследуют окружающую мест-

ность, аккумулируя весь свой опыт в общем интерфейсе Unity ML и не обмениваясь информацией между собой напрямую. Опыт робота представляет собой совокупность вознаграждений, штрафов, векторов координат точек траектории движения. Эти данные подвергаются нормализации и поступают на вход модели нейронной сети для дальнейшей обработки.

## Результаты и их обсуждение

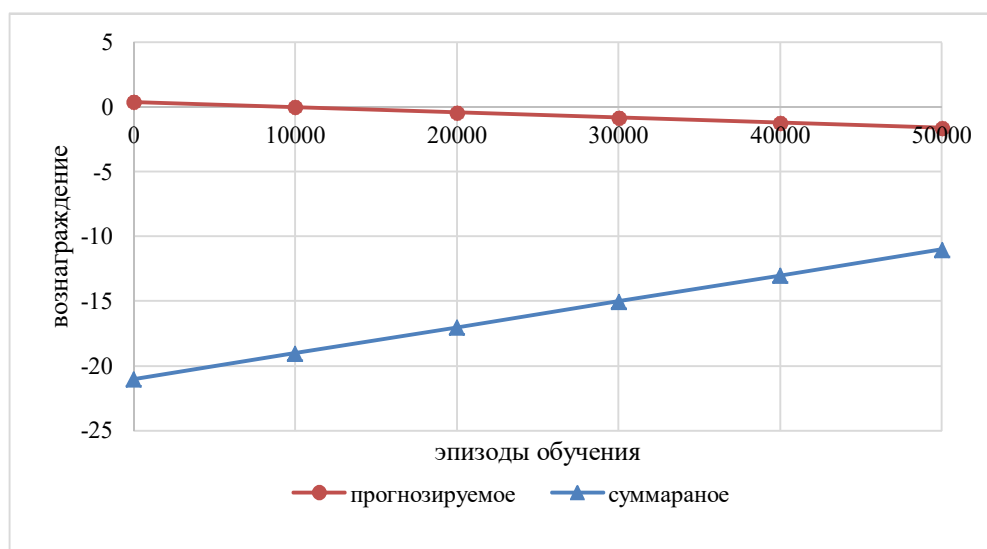
Обучение неоптимизированной модели нейронной сети осуществлялось на небольших объемах данных — 50 000 эпизодов. Предложенная тонкая настройка гиперпараметров нейронной сети позволила увеличить скорость обучения на 30%, что, в свою очередь, дало возможность расширить обучающий набор данных до 250 000 эпизодов. Такая настройка способствовала повышению стабильности во время обучения.

На рис. 4 и 5 представлены зависимости средних прогнозируемых и суммарных вознаграждений за все предыдущие эпизоды от времени обучения для неоптимизированной и оптимизированной моделей соответственно.

Увеличение прогнозируемого вознаграждения в течение обучения свидетельствует о накоплении опыта агентов о предыдущих эпизодах. Прогнозируемое вознаграждение использует этот

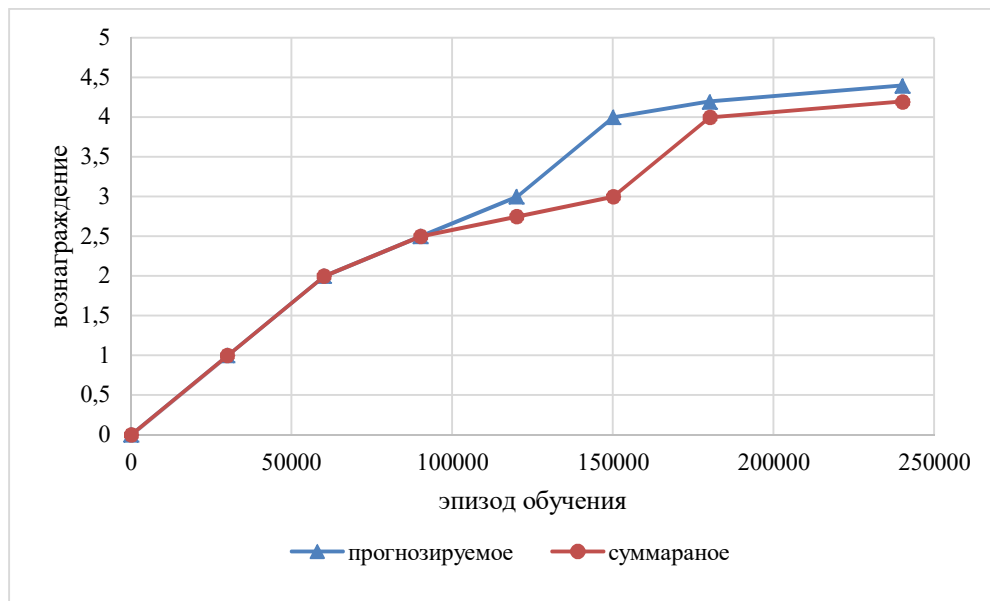
опыт для предсказания последующих действий. Плавная возрастающая кривая суммарного вознаграждения отражает стабильность процесса обучения.

Для оценки эффективности предложенной оптимизации гиперпараметров было проведено по 5 экспериментов в каждой среде. Результаты сравнительного тестирования эффективности модели представлены в табл. Эффективность модели, как и в предыдущих работах, представляет собой долю успешного достижения агентами целевых точек, выраженную в процентах. Достижение считается успешным, если агент достиг конечной точки менее чем за 5 минут, не столкнувшись при этом с другими статическими и динамическими объектами, а также параллельно обучаемыми агентами. Следует отметить, что максимальная скорость платформы в экспериментах была ограничена значением 2 м/с.



**Рис. 4.** Зависимость среднего прогнозируемого и суммарного вознаграждений от времени обучения неоптимизированной модели

**Fig. 4.** Dependency of mean predicted and aggregate rewards on the time of non-optimized model training



**Рис 5.** Зависимость среднего прогнозируемого и суммарного вознаграждений от времени обучения оптимизированной модели

**Fig. 5.** Dependency of mean predicted and aggregate rewards on the time of optimized model training

При проведении экспериментов было выбрано именно такое максимальное значение скорости, поскольку предполагается, что оно отвечает предполагаемым условиям эксплуатации моделируемых мобильных платформ с точки зрения их массогабаритных показателей и требований безопасности. Параметр точности для данной модели нейронной сети рассчитывается как отклонение пути агента от кратчайшего пути между двумя точками, рассчитанного по алгоритму  $A^*$ , который используется в среде Unity в качестве базового алгоритма расчета пути (в условных единицах расстояния в среде Unity). Параметр MSE для данной модели нейронной сети рассчитывается как среднеквадратичная ошибка координат (разница между фактическим и прогнозируемым местоположением агента).

Как видно из таблицы, для оптимизированной модели, обученной в динамически сгенерированном окружении, характерно более чем двукратное повышение эффективности (с 20% до 42%) по сравнению с неоптимизированной моделью, обученной в статичном окружении. Для модели, обученной в динамически сгенерированной среде и тестируемой в динамически сгенерированном окружении, прирост эффективности составляет 10% (с 40% до 50%). Для модели, обученной в статичном окружении и тестируемой в статичной среде, процент достижения целевой точки не изменился и составил 100%. Для модели, обученной в статичном окружении и тестируемой в динамическом окружении, улучшение, прирост эффективности составляет 10% (с 60% до 70%).

**Таблица.** Показатели эффективности моделей нейронной сети, обученных в различных средах**Table.** Performance parameters of neural network models, trained in different environments

Среда тестирования / Test environment	Неоптимизированная модель / Non-optimized model		Оптимизированная модель / Optimized model	
	Обучение в статичной среде / Training in static environment	Обучение в динамически сгенерированной среде / Training in dynamically generated environment	Обучение в статичной среде / Training in static environment	Обучение в динамически сгенерированной среде / Training in dynamically generated environment
Эффективность, % / Efficiency, %				
Статичная Static	100	20	100	42
Динамически сгенерированная Dynamically generated	60	40	70	50
Отклонение от заданного пути / Deflection from the specified path				
Статичная / Static	0.4	5	0.3	2.5
Динамически сгенерированная / Dynamically generated	3.0	2.5	2.7	2.25
MSE				
Статичная / Static	1.0	1.82	1.0	1.58
Динамически сгенерированная / Dynamically generated	1.38	1.74	1.33	1.51

Несмотря на незначительное увеличение эффективности для моделей, протестированных в динамически сгенерированной среде, стоит отметить сокращение общего времени достижения конечных точек. При использовании неоптимизированной модели этот параметр составил около 250-300 секунд

(3-4 минуты), тогда как для оптимизированной модели характерно значение 150-200 секунд (2-3 минуты).

Как видно из таблицы, для оптимизированной модели, обученной в динамически сгенерированном окружении, отклонение от заданного пути снизилось в два раза (с 5 до 2,5 единиц изме-

рения Unity) по сравнению с неоптимизированной моделью, обученной в статичном окружении. Для модели, обученной в динамически сгенерированной среде и тестируемой в динамически сгенерированном окружении, также наблюдается увеличение точности (отклонение снизилось с 2,5 до 2,25). Для модели, обученной в статичном окружении и тестируемой в динамическом окружении, среднее отклонение снизилось на 0,3 единицы. Для модели, обученной в статичном окружении и тестируемой в статичной среде, отклонение снизилось на 0,1 единицы. Значения средней ошибки, представленные в таблице, демонстрируют общее повышение точности модели, особенно для агентов, обученных в динамически сгенерированной среде.

## Выводы

Предложенная тонкая настройка и оптимизация гиперпараметров позволила повысить эффективность навигации мобильной платформы на 10% в динамически сгенерированной среде и на 22% в статической среде. Эксперименты, проводимые в статичном помещении, показали эффективность навигации на 80% выше, чем при тестировании, проводимом в динамически генерируемом помещении. Параллельное обучение агентов на одной сцене позволило повысить скорость обучения. Тон-

кая настройка параметров нейронной сети позволила уменьшить энтропию и увеличить ожидаемое вознаграждение, что позитивно сказалось на принятии решений агентом во время передвижения. Отклонение от намеченной траектории при использовании оптимизированной модели нейронной сети уменьшилось по сравнению с неоптимизированной моделью. Базовая точность рассчитывается на основе встроенного в среду Unity алгоритма поиска пути A\* и системы предотвращения столкновений, результаты работы которых сравниваются с результатами работы неоптимизированной и оптимизированной моделей. Отклонение агентов от заданного пути при использовании оптимизированной модели уменьшилось, при этом количество их столкновений с динамическими и статическими объектами не увеличилось, что говорит об эффективности применяемых в работе подходов. Результаты исследования могут быть применены для совершенствования алгоритмов навигации мобильных платформ [23, 24] в реальных условиях. Особый интерес предложенное решение может представлять при использовании с другими методами трассировки и навигации, когда обученная нейронная сеть действует совместно с уже отработанными и проверенными алгоритмами навигации.

### Список литературы

1. Artificial intelligence and collaborative robot to improve airport operations / F. Donadio, J. Frejaville, S. Larnier, S. Vetault // *In Online Engineering & Internet of Things*. 2018. P. 973-986. [https://doi.org/10.1007/978-3-319-64352-6\\_91](https://doi.org/10.1007/978-3-319-64352-6_91).
2. Kejia robot—an attractive shopping mall guider / Y. Chen, F. Wu, W. Shuai, N. Wang, R. Chen, X. Chen // *In International Conference on Social Robotics*. 2015. P. 145-154. [https://doi.org/10.1007/978-3-319-25554-5\\_15](https://doi.org/10.1007/978-3-319-25554-5_15).
3. Robot Navigation System in Stochastic Environment based on Reinforcement Learning on Lidar Data / D. Dudarenko, A. Kovalev, I. Tolstoy, I. Vatamaniuk // *Proceedings of 14th International Conference on Electromechanics and Robotics Zavalishin's Readings*. 2019; 525-536. [https://doi.org/10.1007/978-981-13-9267-2\\_44](https://doi.org/10.1007/978-981-13-9267-2_44).
4. Dudarenko, Rubtsova, Kovalev, Sivchenko Reinforcement Learning Approach for Navigation of Ground Robotic Platform in Statically and Dynamically Generated Environments. *19th IFAC conference on technology, culture and international stability TECIS* ( в печати).
5. Reignier P., Hansen V., Crowley J.L. Incremental supervised learning for mobile robot reactive control // *Robotics and autonomous systems*. 1997. № 19(3-4). P. 247-257. [https://doi.org/10.1016/S0921-8890\(96\)00054-1](https://doi.org/10.1016/S0921-8890(96)00054-1)
6. Deep Learning for Visual Navigation of Unmanned Ground Vehicles: A review / N. O'Mahony, S. Campbell, L. Krpalkova, D. Riordan, J. Walsh, A. Murphy, C. Ryan // *In 2018 29th Irish Signals and Systems Conference (ISSC)*. 2018. №1-6. <https://doi.org/10.1109/ISSC.2018.8585381>
7. Kim K.S., Kim D.E., Lee J.M. Deep Learning Based on Smooth Driving for Autonomous Navigation. *2018 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*. 2018. P. 616-621.
8. Woods J.O., Christian J.A. Lidar-based relative navigation with respect to non-cooperative objects. *Acta Astronautica*. 2016. № 126. P. 298-311. <https://doi.org/10.1016/j.actaastro.2016.05.007>
9. Multisensor Online Transfer Learning for 3D LiDAR-based Human Detection with a Mobile Robot / Z. Yan, L. Sun, T. Duckct, N. Bellotto // *In 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2018. P. 7635-7640. <https://doi.org/10.1109/IROS.2018.8593899>
10. Tracking People in a Mobile Robot From 2D LIDAR Scans Using Full Convolutional Neural Networks for Security in Cluttered Environments / Á.M. Guerrero-Higueras, C. Álvarez-Aparicio, M.C.C. Olivera, F.J. Rodríguez-Lera, C. Fernández-Llamas, F.M. Rico, V. Matellán // *Frontiers in neurorobotics*. 2018. №12. P. 1-13. <https://doi.org/10.3389/fnbot.2018.00085>

11. Kim Y.H., Jang J.I., Yun S. End-to-end deep learning for autonomous navigation of mobile robot // *2018 IEEE International Conference on Consumer Electronics (ICCE)*. 2018. №1-6. <https://doi.org/10.1109/ICCE.2018.8326229>.
12. Khnissi K., Seddik C., Seddik H. Smart Navigation of Mobile Robot Using Neural Network Controller // *2018 International Conference on Smart Communications in Network Technologies (SaCoNeT)*. 2018. P. 205-210. <https://doi.org/10.1109/SaCoNeT.2018.8585616>.
13. Variable Speed Robot Navigation by an ACO Approach / T. Lei, C. Luo, G.E. Jan, K. Fung // *In International Conference on Swarm Intelligence*. 2019. P. 232-242. [https://doi.org/10.1007/978-3-030-26369-0\\_22](https://doi.org/10.1007/978-3-030-26369-0_22).
14. Traversability Assessment and Trajectory Planning of Unmanned Ground Vehicles with Suspension Systems on Rough Terrain / Y.K. Zhang, Yang, M. Fu, M. Wang // *Sensors*. 2019. №19(20). P. 4372. <https://doi.org/10.3390/s19204372>.
15. Robot navigation of environments with unknown rough terrain using deep reinforcement learning / K. Zhang, F. Niroui, M. Ficocelli, G. Nejat // *In 2018 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*. 2018. P. 1-7. <https://doi.org/10.1109/SSRR.2018.8468643>.
16. Learning navigation behaviors end-to-end with autorl / HT. L. Chiang, A. Faust, M. Fiser, A. Francis. *IEEE Robotics and Automation Letters*. 2019. № 4(2). P. 2007-2014. <https://doi.org/10.1109/LRA.2019.2899918>.
17. Jang J.S.R. ANFIS: adaptive-network-based fuzzy inference system // *IEEE transactions on systems, man, and cybernetics*. 1993. № 23(3). P. 665-685. <https://doi.org/10.1109/21.256541>.
18. Mobile robot navigation in unknown static environments using ANFIS controller / A. Pandey, S. Kumar, K.K. Pandey, D.R. Parhi // *Perspectives in Science*. 2016. №8. P. 421-423. <https://doi.org/10.1016/j.pisc.2016.04.094>
19. Pandey A., Parhi D.R. Multiple mobile robots navigation and obstacle avoidance using minimum rule based ANFIS network controller in the cluttered environment // *International Journal of Advanced Robotics and Automation*. 2016. № 1(1). P. 1-11. <https://doi.org/10.15226/2473-3032/1/1/00102>
20. Smart W.D., Kaelbling L.P. Effective reinforcement learning for mobile robots // *Proceedings 2002 IEEE International Conference on Robotics and Automation* (Cat. No. 02CH37292). 2002. № 4. P. 3404-3410. <https://doi.org/10.1109/ROBOT.2002.1014237>
21. Tai L., Paolo G., Liu M. Virtual-to-real deep reinforcement learning: Continuous control of mobile robots for mapless navigation // *In 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2017. P. 31-36. <https://doi.org/10.1109/IROS.2017.8202134>
22. Curiosity-driven exploration by self-supervised prediction / D. Pathak, P. Agrawal, A.A. Efros, T. Darrell // *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 2017. P. 16-17. URL: [http://openaccess.thecvf.com/content\\_cvpr\\_2017\\_workshops/w5/papers/Pathak\\_Curiosity-Driven\\_Exploration\\_by\\_CVPR\\_2017\\_paper.pdf](http://openaccess.thecvf.com/content_cvpr_2017_workshops/w5/papers/Pathak_Curiosity-Driven_Exploration_by_CVPR_2017_paper.pdf).

23. Proactive Localization System Concept for Users of Cyber-Physical Space / A. Saveliev, D. Malov, A. Edemskii, N. Pavliuk // *In International Conference on Interactive Collaborative Robotics. ICR 2018. Lecture Notes in Computer Science*. 2018; 11097: 233-238. [https://doi.org/10.1007/978-3-319-99582-3\\_24](https://doi.org/10.1007/978-3-319-99582-3_24)

24. Ватаманюк И.В., Савельев А.И. Мобильная робототехническая платформа как компонент киберфизического интеллектуального пространства // *Экстремальная робототехника*. 2017. №1. Р. 37-42.

## References

1. Donadio F., Frejaville J., Larnier S., Vetault S. Artificial intelligence and collaborative robot to improve airport operations. *In Online Engineering & Internet of Things*, 2018, pp. 973-986. [https://doi.org/10.1007/978-3-319-64352-6\\_91](https://doi.org/10.1007/978-3-319-64352-6_91).

2. Chen Y., Wu F., Shuai W., Wang N., Chen R., Chen X. Kejia robot—an attractive shopping mall guider. *In International Conference on Social Robotics*, 2015, pp. 145-154. [https://doi.org/10.1007/978-3-319-25554-5\\_15](https://doi.org/10.1007/978-3-319-25554-5_15).

3. Dudarenko D., Kovalev A., Tolstoy I., Vatamaniuk I. Robot Navigation System in Stochastic Environment based on Reinforcement Learning on Lidar Data. *Proceedings of 14th International Conference on Electromechanics and Robotics Zavalishin's Readings*, 2019, pp. 525-536. [https://doi.org/10.1007/978-981-13-9267-2\\_44](https://doi.org/10.1007/978-981-13-9267-2_44).

4. Dudarenko, Rubtsova, Kovalev, Sivchenko Reinforcement Learning Approach for Navigation of Ground Robotic Platform in Statically and Dynamically Generated Environments. *19th IFAC conference on technology, culture and international stability TECIS* (in press).

5. Reignier P., Hansen V., Crowley J.L. Incremental supervised learning for mobile robot reactive control. *Robotics and autonomous systems*, 1997, no. 19(3-4), pp. 247-257. [https://doi.org/10.1016/S0921-8890\(96\)00054-1](https://doi.org/10.1016/S0921-8890(96)00054-1)

6. O'Mahony N., Campbell S., Krpalkova L., Riordan D., Walsh J., Murphy A., Ryan C. Deep Learning for Visual Navigation of Unmanned Ground Vehicles: A review. *In 2018 29th Irish Signals and Systems Conference (ISSC)*, 2018, pp. 1-6. <https://doi.org/10.1109/ISSC.2018.8585381>

7. Kim K.S., Kim D.E., Lee J.M. Deep Learning Based on Smooth Driving for Autonomous Navigation. *2018 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, 2018, pp. 616-621.

8. Woods J.O., Christian J.A. Lidar-based relative navigation with respect to non-cooperative objects. *Acta Astronautica*, 2016; no. 126, pp. 298-311. <https://doi.org/10.1016/j.actaastro.2016.05.007>.

9. Yan Z., Sun L., Duckctr T., Bellotto N. Multisensor Online Transfer Learning for 3D LiDAR-based Human Detection with a Mobile Robot. *In 2018 IEEE/RSJ International Con-*



*ference on Intelligent Robots and Systems (IROS)*, 2018, pp. 7635-7640. <https://doi.org/10.1109/IROS.2018.8593899>

10. Guerrero-Higueras Á.M., Álvarez-Aparicio C., Olivera M.C.C., Rodríguez-Lera F.J., Fernández-Llamas C., Rico F.M., Matellán V. Tracking People in a Mobile Robot From 2D LIDAR Scans Using Full Convolutional Neural Networks for Security in Cluttered Environments. *Frontiers in neurorobotics*, 2018, no. 12, pp. 1-13. <https://doi.org/10.3389/fnbot.2018.00085>.

11. Kim Y.H., Jang J.I., Yun S. End-to-end deep learning for autonomous navigation of mobile robot. *2018 IEEE International Conference on Consumer Electronics (ICCE)*. 2018; 1-6. <https://doi.org/10.1109/ICCE.2018.8326229>.

12. Khnissi K., Seddik C., Seddik H. Smart Navigation of Mobile Robot Using Neural Network Controller. *2018 International Conference on Smart Communications in Network Technologies (SaCoNeT)*. 2018, pp. 205-210. <https://doi.org/10.1109/SaCoNeT.2018.8585616>.

13. Lei T., Luo C., Jan G.E., Fung K. Variable Speed Robot Navigation by an ACO Approach. *In International Conference on Swarm Intelligence*, 2019, pp. 232-242. [https://doi.org/10.1007/978-3-030-26369-0\\_22](https://doi.org/10.1007/978-3-030-26369-0_22).

14. Zhang K., Yang Y., Fu M., Wang M. Traversability Assessment and Trajectory Planning of Unmanned Ground Vehicles with Suspension Systems on Rough Terrain. *Sensors*, 2019, no. 19(20), 4372 p. <https://doi.org/10.3390/s19204372>.

15. Zhang K., Niroui F., Ficocelli M., Nejat G. Robot navigation of environments with unknown rough terrain using deep reinforcement learning. *In 2018 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, 2018, pp. 1-7. <https://doi.org/10.1109/SSRR.2018.8468643>.

16. Chiang HT. L., Faust A., Fiser M., Francis A. Learning navigation behaviors end-to-end with autorl. *IEEE Robotics and Automation Letters*, 2019, no. 4(2), pp. 2007-2014. <https://doi.org/10.1109/LRA.2019.2899918>.

17. Jang J.S.R. ANFIS: adaptive-network-based fuzzy inference system. *IEEE transactions on systems, man, and cybernetics*, 1993, no. 23(3), pp. 665-685. <https://doi.org/10.1109/21.256541>.

18. Pandey A., Kumar S., Pandey K.K., Parhi D.R. Mobile robot navigation in unknown static environments using ANFIS controller. *Perspectives in Science*, 2016, no. 8, pp. 421-423. <https://doi.org/10.1016/j.pisc.2016.04.094>.

19. Pandey A., Parhi D.R. Multiple mobile robots navigation and obstacle avoidance using minimum rule based ANFIS network controller in the cluttered environment. *International Journal of Advanced Robotics and Automation*, 2016, no. 1(1), pp. 1-11. <https://doi.org/10.15226/2473-3032/1/1/00102>.

20. Smart W.D., Kaelbling L.P. Effective reinforcement learning for mobile robots. *Proceedings 2002 IEEE International Conference on Robotics and Automation* (Cat. No. 02CH37292). 2002; 4: 3404-3410. <https://doi.org/10.1109/ROBOT.2002.1014237>.
21. Tai L., Paolo G., Liu M. Virtual-to-real deep reinforcement learning: Continuous control of mobile robots for mapless navigation. *In 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 31-36. <https://doi.org/10.1109/IROS.2017.8202134>.
22. Pathak D., Agrawal P., Efros A.A., Darrell T. Curiosity-driven exploration by self-supervised prediction. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2017, pp. 16-17. Available at: [http://openaccess.thecvf.com/content\\_cvpr\\_2017\\_workshops/w5/papers/Pathak\\_Curiosity-Driven\\_Exploration\\_by\\_CVPR\\_2017\\_paper.pdf](http://openaccess.thecvf.com/content_cvpr_2017_workshops/w5/papers/Pathak_Curiosity-Driven_Exploration_by_CVPR_2017_paper.pdf).
23. Saveliev A., Malov D., Edemskii A., Pavliuk N. Proactive Localization System Concept for Users of Cyber-Physical Space. *In International Conference on Interactive Collaborative Robotics. ICR 2018. Lecture Notes in Computer Science*, 2018, no. 11097, pp. 233-238. [https://doi.org/10.1007/978-3-319-99582-3\\_24](https://doi.org/10.1007/978-3-319-99582-3_24).
24. Vatamanyuk I.V., Saveliev A.I. Mobil'naya robototekhnicheskaya platforma kak komponent kiberfizicheskogo intellektual'nogo prostranstva [Mobile robotic platform as a component of cyber-physical intellectual space]. *Ekstremal'naya robototekhnika = Extreme Robotics*, 2017, no. 1, pp. 37-42 (In Russ.).

---

### Информация об авторах / Information about the Authors

**Дударенко Дмитрий Михайлович**, младший научный сотрудник лаборатории технологий больших данных социокберфизических систем, Санкт-Петербургский институт информатики и автоматизации Российской академии наук, г. Санкт-Петербург, Российская Федерация, e-mail: [dmitry@dudarenko.net](mailto:dmitry@dudarenko.net)

**Dmitry M. Dudarenko**, Junior Researcher, Laboratory of Big Data and Socio-Cyberphysical Systems, St. Petersburg Institute for Informatics and Automation of the Russian Academy of Sciences, St. Petersburg, Russian Federation, e-mail: [dmitry@dudarenko.net](mailto:dmitry@dudarenko.net)

**Смирнов Петр Алексеевич**, младший научный сотрудник лаборатории автономных робототехнических систем, Санкт-Петербургский институт информатики и автоматизации Российской академии наук, г. Санкт-Петербург, Российская Федерация, e-mail: [petruha.smirnov.1994@gmail.com](mailto:petruha.smirnov.1994@gmail.com)

**Petr A. Smirnov**, Junior Researcher, Laboratory of Autonomous Robotic Systems, St. Petersburg Institute for Informatics and Automation of the Russian Academy of Sciences, St. Petersburg, Russian Federation, e-mail: [petruha.smirnov.1994@gmail.com](mailto:petruha.smirnov.1994@gmail.com)