УДК 681.883.67.001:621.396.677 10.25808/24094609.2019.30.4.003

О НЕКОТОРЫХ АЛГОРИТМАХ РЕШЕНИЯ ЗАДАЧИ ПАТРУЛИРОВАНИЯ ПОДВОДНОЙ СРЕДЫ С ПОМОЩЬЮ ГРУППЫ ВЗАИМОДЕЙСТВУЮЩИХ АНПА

М.С. Спорышев[1], А.Ф. Щербатюк[2]

Дальневосточный федеральный университет[1]
Федеральное государственное бюджетное учреждение науки
Институт проблем морских технологий ДВО РАН[2]

Рассматривается задача патрулирования области в подводной среде с помощью группы автономных необитаемых подводных аппаратов. Отмечаются проблемы, которые делают задачу вычислительно сложной и недоступной для точных методов оптимизации. Исследован приближенный метод, получивший в последние годы большую популярность, – обучение с подкреплением. Процесс патрулирования сформулирован в виде марковского процесса принятия решений. Рассмотрена децентрализованная стратегия, допускающая обмен информацией между аппаратами с возможной потерей данных и учитывающая сложности сетевого взаимодействия в подводной среде. Исследованы алгоритмы на основе методов обучения с подкреплением и проведен их сравнительный анализ. Описаны результаты моделирования работы предложенных алгоритмов в среде OpenAI gym.

ВВЕДЕНИЕ

Автономные необитаемые подводные аппараты (АНПА) все чаще применяются в задачах, связанных с поисковыми и наблюдательными миссиями. Примерами таких сценариев могут быть миссии по охране акваторий [1], поисковые и инспекционные операции [2, 3], отслеживание морских животных [4]. Большая часть таких задач может быть решена более эффективно при использовании группы взаимодействующих АНПА [5]. Однако использование группы также повлечет за собой необходимость решения новых проблем [6, 7].

В данной работе рассматривается сценарий миссии с патрулированием, в рамках которой аппаратам требуется построить маршрут обхода заданной области. При этом миссия не заканчивается при покрытии области полностью, а длится непрерывно. Частным случаем такой миссии можно считать разведывательную миссию, при которой аппаратам требуется совершить один цикл патрулирования, чтобы покрыть область полностью.

Существует большое количество работ, посвященных выполнению миссий с патрулированием и разведкой заданной области с помощью групп взаимодействующих мобильных роботов. Один из подходов основан на декомпозиции области на независимые связные части, которые распределяются между аппаратами. Каждый аппарат после разбиения отвечает за патрулирование только своей области. Примерами таких разбиений областей могут быть диаграммы Вороного [8], Equitable partitioning [9]. Для эффективной работы данных методов требуется наличие полной информации о расположении агентов и состоянии среды, что часто затруднено при использовании распределенных стратегий. Кроме того, в некоторых ситуациях области после разбиения получаются достаточно большими, что усложняет обмен данными между аппаратами на дистанциях больше 1 км.

Одним из популярных для решения данной задачи является теоретико-игровой подход. В этом случае в миссии предполагается наличие злоумышленника, и задача состоит в максимизации вероятности его обнаружения [10]. Теоретико-игровые методы, как правило, являются централизованными и вычислительно затратными [11], что ограничивает их применение на борту АНПА.

Отмеченными вычислительными сложностями обусловлено использование эвристических подходов для решения возникающих оптимизационных задач. Среди таких подходов могут быть эволюционные

[1] 690091, г. Владивосток, ул. Суханова, 8. Тел.: +7 (423) 243-23-15.
[2] 690091, г. Владивосток, ул. Суханова, 5а. Тел.: +7 (423) 243-25-78. E-mail: scherba@marine.febras.ru

оптимизационные методы [12] и методы на основе машинного обучения, обучения с подкреплением [13, 14].

С учетом успешных применений методов обучения с подкреплением [15, 16], в данной работе предлагаеется подход, получивший название proximal policy optimization [17]. Система управления каждым аппаратом моделируется как марковский процесс принятия решений, где функцией принятия решений выступает глубокая нейронная сеть. Такой алгоритм проходит процесс обучения в симуляционной среде, где моделируются некоторые сценарии проблем обмена данными между подводным аппаратами, а также возможные выходы аппаратов из строя.

1. Задача патрулирования

1.1. Патрулирование как марковский процесс принятия решений

Предполагается, что имеется n аппаратов. Каждый аппарат имеет m управляющих сигналов, которые могут подаваться в его систему управления движением в любой момент времени, $u_{ij} \in [0,1]$, i – номер аппарата, j – номер сигнала. Патрулирующий процесс задается пятью параметрами:

$$(S, A, \pi, P, R),$$

где S – множество состояний среды. В нашем случае в качестве среды рассматривается ограниченная двумерная область. Предполагается, что каждому аппарату доступно полное состояние среды и положения в ней всех аппаратов. В модели состояние $s \in S$ представляет из себя прямоугольное изображение среды с нанесенными фигурами аппаратов.

A – множество действий, которые могут совершать АНПА. В нашем случае $A = [0,1]^{n \times m} = \{u_{ij}\}$ – значения управляющих сигналов каждого аппарата.

$\pi(a_t \mid s_0, s_1, \ldots, s_t)$ – функция распределения действий a_t, которые должны совершить аппараты в момент времени t, по текущей истории состояний среды $s_0, s_1, \ldots, s_t$. В данной работе функция-стратегия аппроксимируется глубокой нейронной сетью с параметрами θ, которые вычисляются путем решения задачи оптимизации. Так как данная функция зависит от параметров θ, далее будем обозначать ее π_θ.

$P(s_{t+1} \mid s_t, a_t)$ – функция распределения следующего состояния среды, принимающая текущее состояние s_t и действия агентов a_t в момент времени t. Использование функции распределения в данном случае необходимо, так как в среде возможны случайные события, такие как выход аппаратов из строя. Таким образом, $s_{t+1} \sim P(s_{t+1} \mid s_t, a_t)$, т.е. следующее состояние среды является случайной величиной, зависимой от текущего состояния и действий аппаратов.

$R(s_0, s_1, \ldots, s_t, a_t) \rightarrow \mathbf{R}$ – функция, возвращающая числовую награду по текущей истории состояний и последнему действию. Значения этой функции на каждом шаге будут использованы алгоритмом оптимизации для настраивания параметров функции π_θ.

Рассматривается дискретный процесс, начинающийся в некотором состоянии s_0. На шаге $t \in [0,T]$ среда, включающая положения всех аппаратов, находится в состоянии s_t. С помощью функции-стратегии выбирается случайное действие a_t согласно распределению $\pi_\theta(a_t \mid s_t)$, которое в данный момент времени совершат аппараты. Среда переходит в случайное состояние $s_{t+1} \sim P(s_{t+1} \mid s_t, a_t)$, распределенное согласно текущему состоянию, действию и функции P, которая однозначно задается симулятором, среда также сообщает награду $R(s_0, \ldots, s_t, a_t)$ в текущий момент t. Такой процесс продолжается некоторое конечное число шагов T.

Задача состоит в том, чтобы найти параметры функции π_θ, при которых достигается максимум математического ожидания суммы всех наград, полученных за конечное число шагов T:

$$\max_\theta E\Big[\sum_{t=0}^{T} R(s_0, \ldots, s_t, a_t)\Big],$$

$$a_t \sim \pi_\theta(a_t \mid s_t),$$

$$s_t \sim P(s_t \mid s_{t-1}, a_{t-1}).$$

В вышеописанной постановке распределение действий аппаратов зависит от всей истории состояний среды при выполнении миссии, $s_0, \ldots, s_t$. Аппроксимировать такую функцию распределения весьма затруднительно, так как такая функция будет иметь переменное число аргументов, количество которых может быть достаточно велико. Поэтому в рамках принятой модели вышеописанный процесс принятия решений сводится к марковскому, в котором функция награды и функция распределения действий зависят только от предыдущего состояния, вместо всей истории.

Состоянием среды s_t будем считать не только двумерное изображение среды, но и двумерную бинарную маску, на которой отмечена та часть области, которая уже попадала в зону видимости одного из аппаратов, а также бинарную маску препятствий (вместе с фигурами аппаратов) (см. рис 4). Такое описание состояния позволит вычислять функцию награды (доля посещенной части области) только по текущему состоянию и действию $R(s_t, a_t)$, без учета

всех предыдущих состояний. Также и функция распределения действий аппаратов может быть построена как функция только текущего состояния $\pi_\theta(a_t \mid s_t)$. Получаем марковский процесс принятия решений, описываемый пятью параметрами:

$$(S, A, \pi_\theta(a_t \mid s_t), P(s_{t+1} \mid s_t, a_t), R(s_t, a_t)).$$

1.2. Централизованная стратегия

Вышеописанная функция распределения действий аппаратов $\pi_\theta(a_t \mid s_t)$ владеет информацией обо всех АНПА в группе через состояние среды и предсказывает управляющее воздействие для всех аппаратов в группе, т.е. входит в состав централизованного алгоритма.

В предложенном алгоритме функция распределения действий аппроксимируется нейронной сетью с комплексной архитектурой. На рис. 1 отражена архитектура для децентрализованной стратегии, которая отличается от централизованной только тем, что вместо одной сущности для координат всех АНПА в группе используются две сущности: координаты текущего АНПА и вектор координат с временными метками остальных АНПА в группе.

Для сверточной части архитектуры были выбраны ResNet блоки [18], которые хорошо показали себя при обработке двумерных изображений. После полносвязных слоев от всех наблюдаемых величин использовался механизм внимания, описанный в работе [19]. Результат подавался на вход рекуррентной нейронной сети LSTM [20]. Далее, чтобы получить управляющие сигналы (действия, блок Actions), используется полносвязный слой, выходом которого является вектор размерности $2 \times n \times m$ – матожидания и дисперсии гауссовского распределения для каждого управляющего сигнала каждого из аппаратов.

1.3. Децентрализованный сценарий

В случае децентрализованного алгоритма функция распределения действий будет запущена локально на каждом аппарате и будет предоставлять распределение действий соответствующего аппарата. Таким образом, множество действий $A = [0,1]^m = \{u_i\}$ является множеством управляющих сигналов одного аппарата.

Кроме того, каждый аппарат владеет только информацией о своем положении, бинарной маской своей пройденной части области. У аппаратов имеется возможность обмениваться положениями и бинарными масками пройденной области. Каждый аппарат регулярно рассылает список покрытых им координат области. Таким образом, все аппараты хранят бинарную маску области, покрытой другими аппаратами, согласно полученным в результате обмена данным. Также каждый аппарат хранит heartbeat вектор для каждого аппарата в группе. Это вектор количества временных отсчетов с последнего обновления информации от каждого другого аппарата в группе.

Возможность получать сообщения аппаратами моделируется двумя способами. В первом случае аппараты могут получать сообщения от другого ап-

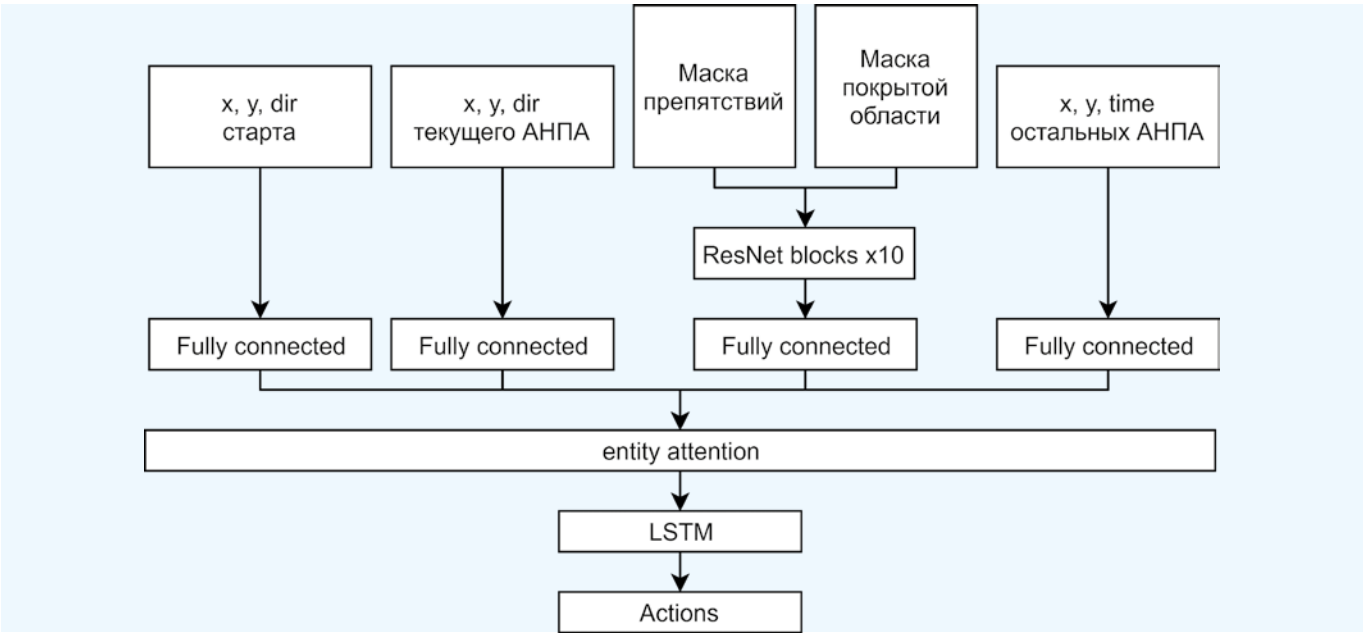

Рис. 1. Структура функции-стратегии в децентрализованном случае

парата, только если расстояние между ними меньше заданной заранее константы p^*. Если расстояние достаточное, все данные, отправленные аппаратами, будут корректно получены в полном объеме. Во втором случае успешное получение пакета данных одного аппарата от другого моделируется как случайная величина, нормально распределенная вокруг текущего положения аппарата. Более подробно, вероятность успешно передать данные от аппарата, находящегося в позиции $pos_s = (x_s, y_s)$, аппарату в позиции $pos_t = (x_t, y_t)$, имеет вид:

$$\frac{1}{\sqrt{2\pi\sigma}} exp\left(\frac{-\rho(pos_s pos_t)}{2\sigma^2}\right),$$

где ρ – евклидово расстояние между точками . Таким образом, каждая отправленная аппаратом координата может быть получена или нет в зависимости от случайного события с заданным распределением.

1.4. Процедура обучения

В процессе обучения агенты получают награду $-np / C$, где np – количество новых покрытых пикселов, а C – нормировочная константа. Отрицательная награда используется для того, чтобы аппараты минимизировали длину траектории в процессе обучения. За полное покрытие территории агенты получают награду R^*, которая настраивается для различных сценариев. Патрулирование достигается повторением миссии покрытия территории. В процессе обучения используется метод Proximal Policy Optimization [17], который предполагает наличие двух нейронных сетей – actor network, которая выдает распределение следующего действия агента и critic network, которая предсказывает будущую награду, полученную в течение дальнейшей части эпизода. Децентрализованная стратегия в процессе обучения использует централизованный подход. Critic network принимает на вход полное состояние среды. Агенты же используют одинаковую actor network с одинаковыми весами, однако все действия и наблюдения у агентов разные и вычисляются независимо друг от друга.

2. Вычислительный эксперимент

Для выполнения вычислительного эксперимента был разработан симулятор на основе платформы OpenAI Gym [21], вместе с библиотекой двумерной физики твердого тела Box-2D [22]. Каждый аппарат моделировался как двумерное твердое тело треугольной формы, для которого алгоритм обучения с подкреплением мог управлять только угловой скоростью. Движение вперед выполнялось с постоянной, заранее заданной скоростью. Среда представляла из себя закрытую область со случайной кусочно-линейной границей. В каждом эпизоде случайно генерировалась область внутри квадрата 400 × 400, каждый эпизод длился не более 500 шагов, после чего миссия завершалась.

В вычислительном эксперименте тестировались несколько сценариев.

Обход области со случайной границей одним аппаратом.

На рис. 2, *а* приведен пример траектории аппарата, а на рис. 2, *б* показано количество эпизодов и шагов, которые потребовались, чтобы аппарат начал стабильно обходить область без столкновений с ее границами за нужное число шагов.

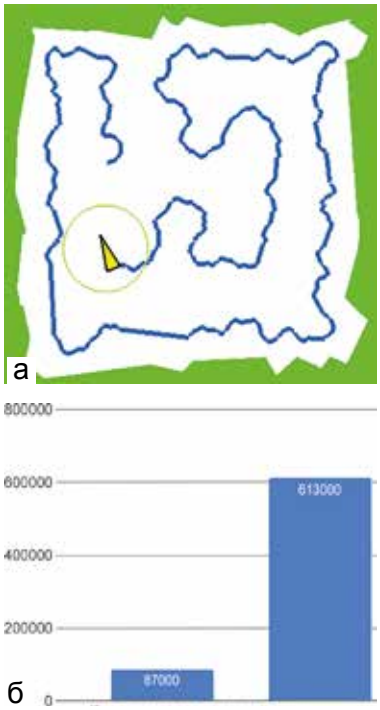

Рис. 2. *а* – пример траектории аппарата; *б* – количество эпизодов, необходимых для достижения приемлемой стратегии обхода в случае централизованной стратегии для одного и двух аппаратов

Обход области с полной информацией.

Во втором сценарии использовались многоагентные централизованная и децентрализованная стратегии с полной информацией. В эксперименте использовались 2 аппарата. В централизованной стратегии наблюдением каждого аппарата являлась маска покрытия области всеми аппаратами в каждый момент времени. На рис. 3 можно увидеть пример наблюдения и примеры траекторий аппаратов. Видно, что распределенная стратегия использует меньшее число шагов для обучения той же задаче.

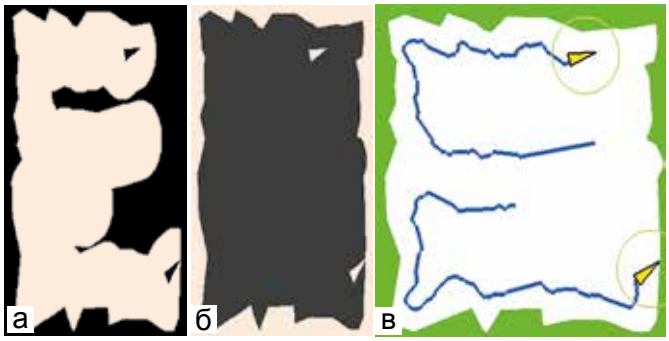

Рис. 3. Наблюдение-маска покрытой области (*а*), наблюдение-маска препятствий (*б*), траектории АНПА (*в*)

Обход области с неполной информацией и обменом данными без потерь.

В данном сценарии использовалась децентрализованная стратегия с неполной информацией, но идеальным обменом. Таким образом, наблюдением каждого аппарата являлась только его маска покрытия области и части масок других аппаратов, актуальные на момент обмена данными с ними. Обмен данными осуществлялся автоматически при сближении аппаратов на достаточное, заранее заданное расстояние. На рис. 4 можно видеть количество эпизодов для обучения в сравнении с подходами, использующими полную информацию о среде.

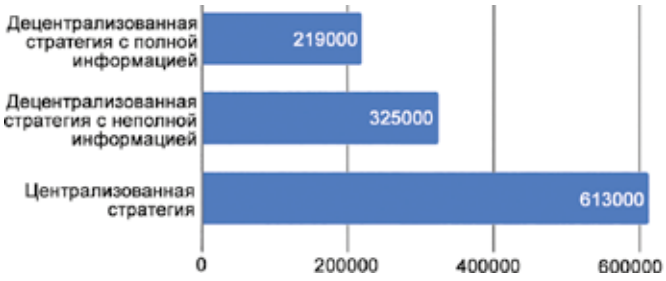

Рис. 4. Количество эпизодов для обучения соответствующих функций-стратегий

Сценарий с выходами аппаратов из строя.

Использовался сценарий с неполной информацией, в котором в случайные моменты времени любой аппарат мог выйти из строя. Хотя бы один аппарат всегда оставался в рабочем состоянии до конца миссии. В данном сценарии использовались несколько подходов для сравнения. В первом использовалась децентрализованная стратегия, обученная на эпизодах с неполной информацией, но без выходов аппаратов из строя. Во втором подходе использовалась аналогичная стратегия, но обученная на сценариях с выходами аппаратов из строя. В третьем подходе использовалась стратегия, описанная в разделе 1.3, использующая heartbeat временные метки каждого аппарата. На рис. 5 можно наблюдать примеры траекторий и средние показатели каждой из стратегий.

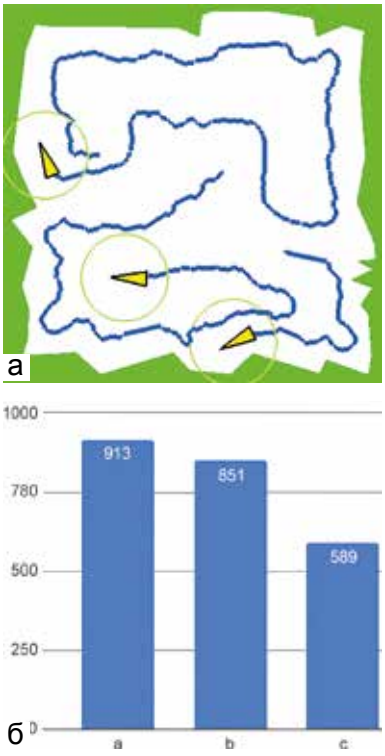

Рис. 5. Слева: пример траекторий. Справа: среднее количество шагов для покрытия территории: *а* – децентрализованная стратегия обученная на эпизодах без выхода АНПА из строя, *б* – стратегия, не использующая heartbeat временные метки, *с* – полная децентрализованная стратегия

По длинам траектории можно понять моменты выходов АНПА из строя (все АНПА двигаются с одинаковой скоростью).

Сценарий с выходами аппаратов из строя и обменом с потерей информации.

В данном сценарии моделировались потери переданных аппаратами сообщений вместо сообщений со случайными выходами аппаратов из строя. В сценарии использовались 3 аппарата. Стратегия была обучена на аналогичных сценариях, также с тремя аппаратами. В каждом сценарии к концу миссии оставался только один аппарат. На рис. 6 можно увидеть таймлайн выходов аппарата из строя и средний итоговый процент покрытия территории по всем эпизодам.

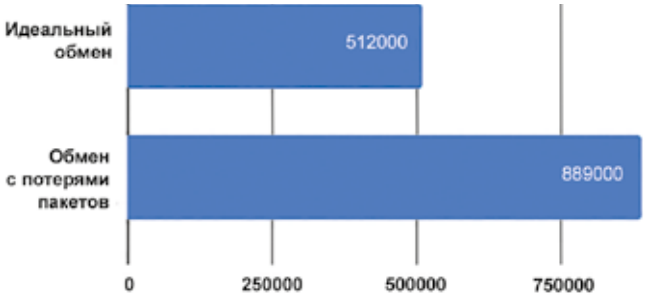

Рис. 6. Количество эпизодов для обучения функции стратегии в сценарии с группой из трех АНПА и выходами аппаратов из строя

ЗАКЛЮЧЕНИЕ

Представлен новый подход к задаче патрулирования области в подводной среде, учитывающий сложности обмена данными между подводными аппаратами и возможные выходы аппаратов из строя во время выполнения миссии. Хорошие результаты по организации управления в двумерной среде показала модель децентрализованной стратегии на основе обучения с подкреплением. В будущем предполагается рассмотреть более сложную трехмерную среду для испытаний и учитывать пространственную динамику движения АНПА. Также планируется провести эксперимент в морской среде с АНПА «ММТ-3000» [23].

ЛИТЕРАТУРА

1. Caiti A., Munafo A., Vettori G. A geographical information system (gis)-based simulation tool to assess civilian harbor protection levels // IEEE J. of Oceanic Engineering. 2012. Vol. 37, No. 1. P. 85–102.
2. Inzartsev A., Pavin A., Panin M., Tolstonogov A., Eliseenko G. Detection and Inspection of Local Bottom Objects with the Help of a Group of Special-Purpose AUVs // Proc. of the OCEANS 2018 MTS/IEEE Conference. Kobe, Japan, 2018.
3. Inzartsev A., Pavin A. AUV Behavior Algorithm While Inspecting of Partly Visible Pipeline // Proc. of the OCEANS 2006 MTS/IEEE Conference. Boston, MA, USA, 2006.
4. Lin Y. et al. A Multi Autonomous Underwater Vehicle System for Autonomous Tracking of Marine Life // J. of Field Robotics. 2017. Vol. 34, No. 4. P. 757–774.
5. Cao Xiang, Daqi Zhu, Simon X. Yang. “Multi-AUV target search based on bioinspired neurodynamics model in 3-D underwater environments” // IEEE transactions on networks and learning systems 27. 2016. Vol. 11 (2016). P. 2364–2374.
6. Ferri G. et al. Cooperative robotic networks for underwater surveillance: an overview // IET Radar, Sonar & Navigation. 2017. Vol. 11, No. 12. P. 1740–1761.
7. Pavin A., Inzartsev A., Eliseenko G. Reconfigurable Distributed Software Platform for a Group of UUVs (Yet Another Robot Platform) // Proc. of the OCEANS 2016 MTS/IEEE Conference & Exhibition. California, USA, September 19-23, 2016.
8. Antonelli G. et al. Harbour protection strategies with multiple autonomous marine vehicles // Int. Workshop on Modelling and Simulation for Autonomous Systems. Springer, Cham, 2014. P. 241–261.
9. Pavone M., Arsie A., Frazzoli E., Bullo F. Equitable partitioning policies for robotic networks // 2009 IEEE Int. Conf. on Robotics and Automation. 2016. May. P. 2356–2361.
10. Agmon Noa, Sarit Kraus, Gal A. Kaminka. Multi-robot perimeter patrol in adversarial settings // 2008 IEEE Int. Conf. on Robotics and Automation. Pasadena, CA, USA, 2008.
11. Asghar A.B., Smith S.L. A Patrolling Game for Adversaries with Limited Observation Time // 2018 IEEE Conference on Decision and Control (CDC). IEEE, 2018. P. 3305–3310.
12. Portugal David et al. Finding optimal routes for multi-robot patrolling in generic graphs // 2014 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems. IEEE. Chicago, Illinois, USA, 2014.
13. Adepegba A.A., Miah S., Spinello D. Multi-agent area coverage control using reinforcement learning // The Twenty-Ninth International Flairs Conference. Florida, USA, 2016. P. 368–373.
14. Sporyshev M., Scherbatyuk A. Reinforcement learning approach for cooperative AUVs in underwater surveillance operations // Proc. of the 2019 IEEE Underwater Technology (UT). IEEE. Kaohsiung, Taiwan. 2019. P. 74–77.
15. Schulman J. et al. Trust Region Policy Optimization // Proc. of Int. Conf. Machine Learning. Lille, France. 2015. Vol. 37.
16. Mnih V. et al. Asynchronous methods for deep reinforcement learning // Proc. of 33th Int. conf. on machine learning. USA . 2016. Vol. 48. P. 1928–1937.
17. Schulma J. et al. Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347 (2017). Helsinky, Finland. 2017.
18. He K., Zhang X., Ren S., Sun, J. Deep residual learning for image recognition // Proc. of the IEEE conf. on computer vision and pattern recognition. Las Vegas, Nevada, USA. 2016. P. 770–778.
19. Vaswani Ashish et al. Attention is all you need. Advances in neural information processing systems // Proc. of 31th Conf. on Neural Information Processing Systems (NIPS 2017). Long Beach, CA, USA.
20. Hochreiter S., Schmidhuber J. Long short-term memory. Neural computation. 1997. Vol 9 (8). P. 1735–1780.
21. Brockman Gr. et al. Openai gym. – URL: http://www.gym.openai.com (дата обращения: 18.11.2019).
22. Catto E. Box2d: A 2d physics engine for games. – URL: http://www.box2d.org (дата обращения: 18.11.2019).
23. Горнак В.Е., Инзарцев А.В., Львов О.Ю., Матвиенко Ю.В., Щербатюк А.Ф. ММТ-3000 – новый малогабаритный автономный необитаемый подводный аппарат ИПМТ ДВО РАН // Подводные исследования и робототехника. 2007. № 1 (3). С. 12–20.