**УДК 007.52, 519.878, 519.1, 004.942, 006.72**

# ПЕРЕНОС ПОДХОДА МАШИННОГО ОБУЧЕНИЯ С ПОДКРЕПЛЕНИЕМ С СИМУЛЯЦИОННОЙ МОДЕЛИ НА МОБИЛЬНОГО РОБОТА

А.Г. Сагитов (*sagitov@it.kfu.ru*)
Казанский Федеральный Университет, Институт Информационных технологий и интеллектуальных систем, Лаборатория интеллектуальных робототехнических систем, Казань

Tetsuto Takano (takanotetuto@gmail.com)
Университет Канадзавы, Канадзава, Япония

Shohei Muto (nsmkaa@gmail.com)
Университет Канадзавы, Канадзава, Япония

Р.О. Лавренов (lavrenov@it.kfu.ru)
Казанский Федеральный Университет, Институт Информационных технологий и интеллектуальных систем, Лаборатория интеллектуальных робототехнических систем, Казань

**Аннотация**. Обучение с подкреплением, как один из способов машинного обучения, показывает многообещающие результаты при его интеграции в различные робототехнические алгоритмы. Но для того, чтобы добиться оптимального поведения робота, требуется значительное количество времени и ресурсов. Используя виртуальные эксперименты, возможно значительно ускорить и улучшить производительность алгоритмов. Мы внедрили подход обучения с подкреплением для алгоритма локализации и картографирования, применяемого на мобильном роботе. Алгоритм был обучен в симуляционной среде Gazebo и перенесен на реального робота. В публикации показана целесообразность использования симуляции для обучения алгоритмов, применяемых мобильными роботами.[1]

**Ключевые слова**: алгоритм, мобильный робот, машинное обучение, моделирование, Gazebo.

[1] Работа выполнена при финансовой поддержке РФФИ (проект № 19-58-70002).

## Введение

Обучение с подкреплением (англ. reinforcement learning) — это метод машинного обучения, в основе которого лежит процесс корректировки модели поведения агента при взаимодействии с некоторой средой на основе функционала выигрыша (оценка действий агента). Процесс обучения, таким образом, является процедурой нахождения таких значений параметров модели поведения, которые обеспечивающих максимум выигрыша. Метод основан на принципах, аналогичных преподаванию в школе: сдача тестов и экзаменов, с получением оценок (положительных и отрицательных) для последующей работы над ошибками. В робототехнике такой метод является одним из способов проектирования и реализаций сложных моделей поведения, особенно предполагающих взаимодействия с неопределенной или динамичной средой.

Первоначальной целью исследования было создание колесного мобильного робота, выполняющего автономное исследование окружающей среды, на основе алгоритма обучения с подкреплением. Основным фактором, который должен быть учтён при использовании метода обучения с подкреплением — это общее количество времени, которое необходимо агенту на оптимизацию параметров поведения. Реальные эксперименты в таком случае являются достаточно медленными и затратными, и не исключают вероятности поломки агента или части окружения. Одним из возможных решений этой проблемы является имитационное моделирование эксперимента. В процессе обучения на модельных экспериментах алгоритм может быть апробирован в реальных условиях с последующей корректировкой параметров моделируемой среды.

Одним из основных инструментов разработки и верификации алгоритмов машинного обучения с подкреплением является система OpenAI Gym [Brockman, 2016]. Система представляет набор инструментов и библиотек для отладки и тестирования разработанных алгоритмов для различных окружений. В работе [Zamora et al., 2016] авторы интегрировали инструменты OpenAI Gym в Робототехническую Операционную Систему ROS и систему имитационного моделирования Gazebo, упростив тем самым интеграцию алгоритмов обучения в существующие РТС. Была также добавлена возможным отладки существующих алгоритмов на роботах в реальных условиях.

В рамках этой работы сделанный нами мобильный робот был интегрирован в среду ROS\Gazebo для решения задачи автономного исследования окружения. Разработанные интерфейсы управления были унифицированы для реального и виртуального робота. Таким образом, после завершения обучения в симуляции полученная стратегия сразу

может быть перенесена на реального робота для апробации. Обучение робота произведено на основе алгоритма Q-Learning, используя функционал вознаграждения робота, поощряющий исследование ранее неисследованных областей и штрафующий за столкновения с препятствиями окружающей среды и пассивность.

## 1 Существующие подходы

Математическая концепция обучения с подкреплением появилась в результате поиска оптимального управления Марковским процессом принятия решений в работах Беллмана [Bellman, 1957]. Дальнейшие принципы обучения с подкреплением были развиты в работах [Sutton, 1984], [Watkins, 1989]. В последние годы обучение с подкреплением стало одним из важных методов в робототехнике [Kober et al., 2013]. Этот метод может быть применен для передвижения робототехнических систем (РТС) [Kohl et al., 2004], [Endo et al., 2008], манипулирования объектами [Peters et al., 2008], [Theodorou et al., 2010], [Peters et al., 2010], [Kalakrishnan et al., 2011], планирования движения автономных летающих устройств [Abbeel et al., 2006] и других задач.

Комбинирование подхода обучения с подкреплением на основе нейронных сетей показало значительный потенциал при выполнении сложных взаимодействий, например, для управления в реальном времени семиосевыми манипуляторами [Pastor et al., 2009], [Schulman et al., 2015], [Levine et al., 2016]. Использование больших и глубинных нейронных сетей (англ. deep neural networks) позволило роботам освоить сложные манипуляции с минимальной корректировкой траекторий, ставя вопрос о применение подобного алгоритм к решению произвольных задач управления [Deisenroth et al., 2011], [Moldovan et al., 2015].

В исследовании [Sadeghi et al., 2018] сложная нейронная сеть (комбинация сверточной нейронной сети с моделью долго-краткосрочной памяти) успешно выполнила автокалибровку суставов на основе сохраненной истории произведенных действий и показаний датчиков. Помимо этого, обучив подобную модель на наборе синтетических (смоделированных) выборок, состоящих из набора конечных положений с рассчитанными траекториями, модель была использована для управления роботом манипулятором для достижений различных конечных положений в произвольных координатных системах.

## 2 Моделирование объекта и процесса обучения

Для организации процесса обучения виртуального робота в виртуальной среде мы использовали среду выполнения OpenAI gym-gazebo. Среда gym-gazebo является комбинацией системы машинного

обучения OpenAI Gym, системы управления ROS и среды физического моделирования Gazebo. OpenAI gym предоставляет набор интерфейсов для реализации, тестирования и отладки алгоритма обучения управляющего поведением реального и виртуального робота, в том числе и интерфейс взаимодействия со средой Gazebo [Afanasyev et al., 2015].

## 2.1 Обучаемый робот

В качестве обучаемого робота был выбран специально разработанный мобильный робот, управляемый с помощью установленного микроконтроллера Arduino Uno (Рис. 1). Обучаемый робот представляет собой мобильный четырехколесный робот с четырьмя двигателями постоянного тока (по одному на колесо, по два двигателя с каждой стороны). Каждое колесо оборудовано пружинной амортизацией. Для каждой оси был реализован программный контроллер скорости на основе библиотеки *ros_control* системы ROS. Робот реализует модель дифференциального рулевого управления при выполнении поворотов и полных разворотов на месте, создавая разницу в скоростях колес с левой или правой сторон. Для оценки окружения робота был использован лазерный дальномер Hokuyo LIDAR (UTM-30LX), установленный на верхней передней части робота.

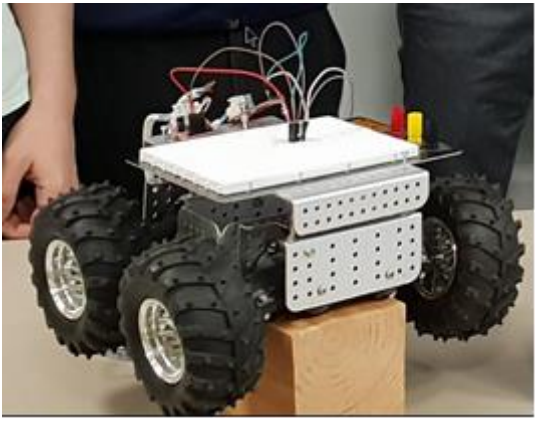

Рис. 1. Обучаемый робот на контроллере Arduino Uno.

Протоколом связи робота с ROS является цифровой интерфейс UART, реализованный на основе библиотеки *rosserial* системы ROS, с предоставлением доступа к коммуникационной подсистеме ROS, с интеграцией координатных систем, определенных в разных системах отсчёта. Также была выполнена синхронизация работы двигателя и

датчика с системным временем ROS. Использованные протоколы *rosserial* были адаптированы с помощью написания оберточных функций над стандартными типами сообщений ROS с последующим мультиплексированием несколько сообщений с микроконтроллера.

Виртуальное представление обучаемого робота было создано в среде Gazebo с реализацией контроллеров управления, моделей датчиков и интерфейсов модели с адаптацией под поведение реального робота, (Рис. 2), используя подход, примененный нами ранее для моделирования гусеничного робота [Sokolov et al., 2016], [Лавренов и др., 2018].

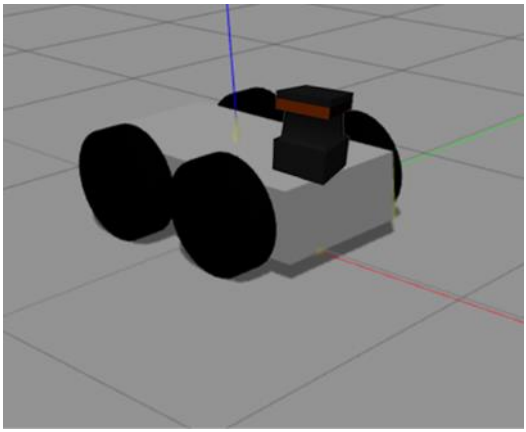

Рис. 2. Виртуальный робот в Gazebo.

## 2.2. Алгоритм обучения

Для организации процесса обучения РТС процедуре исследования окружающей среды был использован широко распространенный многошаговый метод обучения с подкреплением Q-Learning [Watkins, 1989]. Поведение робота в модели определяется набором правил, определяющих возможные действия в различных обстоятельствах, отношение между действиями и вознаграждениями при их выполнении являются вероятностными переходами.

Процесс исследования роботом окружающей среды был представлен как модель конечного Марковского процесса принятия решений (англ. FMDP - finite Markov decision process). Используя алгоритм Q-learning, был произведен поиск оптимальных параметров модели, определенных максимумом ожидаемого значения выигрыша (в нашем случае отношения исследованной области к общей) в каждом из действий робота. Начиная с некоторой начальной точки в пространстве принятия решений, алгоритм Q-learning позволил найти оптимальные параметры для модели поведения

(в виде формализованного FMDP). В качестве функционала выигрыша было принято отношение меры исследованного пространства к общему. Вычисление отношения производилось на основе сравнения карты, построенной роботом на основе показаний лидара, и модельной карты лабиринта.

Робот строит свою карту используя исключительно показания лидара. Для каждой модельной среды определена модельная карта, представляющая из себя желаемое состояние – полностью исследованная карта.

Обучаемый робот — это кортеж состояний S и набора возможных действий из этих состояний A. Состояние робота S представляет собой набор текущих координат робота, простроенной на основе показаний лидара карты и текущей оценке. В случае выбора действия $\mathbf{a} \in A$, робот выполняет переходы из текущего состояния в состояние, определенное действием **a.** После каждого перехода, робот подсчитывает суммарное вознаграждение на основе функционала. По завершению определенного количества шагов, или в случае, когда робот застревает и не может двигаться дальше, проводится расчет окончательного вознаграждения.

Робот изменяет параметры переходов для максимизации суммарного вознаграждения, в результате поиска максимума прогнозируемого вознаграждения в будущих состояниях. Общее потенциальное суммарное вознаграждение определяется как взвешенная сумма ожидаемых значений вознаграждений за все будущие действия, начиная с текущего состояния, и является целевой функцией для поиска оптимальных параметров поведения.

После выполнения действия целевая функция обновляется:

$$Q(s_t,a_t) \leftarrow Q(s_t,a_t) + \alpha \, |r_t + \gamma \max Q(s_{t+1},a_t) - Q(s_t,a_t)| \qquad (1)$$

где $r_t$ – вознаграждение за шаг t, параметры $\alpha$ и $\gamma$ являются гиперпараметрами алгоритма обучения, $\alpha$ - скорость обучения, определяющая влияние новой полученной информации, $\gamma$ коэффициент дисконтирования, определяющий выбор максимума будущего или текущего вознаграждения.

Для нашего обучаемого робота определены четыре возможных действия в любом состоянии: движение вперед (одновременное вращение обеих осей со скоростью +0,2 м/с), левый поворот (левая ось -0,1 м/с, правая ось +0,1 м/с), правый поворот (левая ось +0,1 м/с, правый ось -0,1 м/с) и движение задним ходом (обе оси -0,2 м/с). Вознаграждение робота штрафовалось в случаях касания роботом препятствий. Шаг по времени выбран размером в 1 секунду, по выполнению выбранной формы

движения, на основе данных лидара карта робота достраивалась и рассчитывалось текущее вознаграждение.

### 2.3 Среда исследования

В качестве тестовой среды была выбрана одна из стандартных симуляций Gazebo: GazeboCircuit2TurtlebotLidar-v0. Эта среда состоит из закольцованного коридора для робота с пятью правыми поворотами и одним левым поворотом (Рис. 3). Был использован датчик LIDAR для локализации и картографирования окружающей среды. Больше никакой информации, с помощью которой можно было бы судить о положении робота, не использовалось.

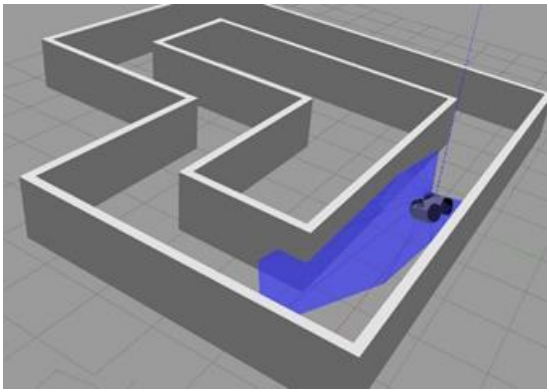

Рис. 3. Среда для тестирования в ROS/Gazebo.

## 3 Результаты

Использование смоделированного робота в виртуальной среде позволило обучить алгоритм примерно в 10 раз быстрее, чем с помощью обучения в режиме реального времени. Таким образом виртуальный робот может выполнять в десять раз больше обучающих экспериментов, чем настоящий робот. Так, более пяти тысяч запусков алгоритма исследования виртуальной среды были завершены в течение 6 часов.


Рис. 4. Суммарное вознаграждение в ходе запусков алгоритма.

На рисунке 4 продемонстрировано, что средняя общая награда постепенно увеличилась, следовательно, алгоритм исследования среды значительно улучшился с течением времени. Даже с тренировками, выполненными полностью в симуляторе, были получены стратегии исследования, которые хорошо работают на физическом роботе. Мы связываем успешный переход, в первую очередь, с унифицированным интерфейсом управления между реальным и виртуальным роботом. Однако, были замечены несколько факторов, которые способствовали различию между симуляцией и реальностью во время выполнения алгоритма. Реальный робот, в отличии от виртуального, демонстрирует проскальзывание колеса в поперечном и продольном направлениях, накапливает неточности скорости вращения колеса и более склонен к столкновениям при движении возле стен.

## 4 Планы дальнейшей работы

В перспективе мы заинтересованы проверить метод на более сложных роботах с использованием большего набора различных сенсоров. Для дальнейшего улучшения процесса обучения мы планируем распараллелить процесс расчетов и реализовать автоматические рекомендации из результирующей стратегии. Также будет проведена дальнейшая диверсификация среды и внедрены дополнительные инструменты для расчета показателей производительности для различных алгоритмов.

## Заключение

Обучение с подкреплением играет важную роль в растущей области машинного обучения. Чтобы преодолеть трудности, связанные с обучением, использование робототехнических симуляторов, таких как Gazebo, позволяет существенно снизить стоимость и повысить скорость разработки. В данной работе было продемонстрировано, что алгоритмы обучения с подкреплением способны изучать сложные навыки исследования с нуля и без целенаправленно разработанных траекторий. Тем не менее, наш метод имеет ряд существенных ограничений, включая упрощенные модели робота и тестовой среды.

## Список литературы

**[Brockman, 2016]** G. Brockman, et.al., OpenAI gym (arXiv preprint, arXiv:1606.01540, 2016).

**[Zamora et al., 2016]** I. Zamora, et.al. Extending the OpenAI Gym for robotics: a toolkit for reinforcement learning using ROS and Gazebo (arXiv preprint arXiv:1608.05742, 2016).

**[Bellman, 1957]** R. Bellman, A Markovian decision process, in *Journal of Mathematics and Mechanics* (1957) pp. 679-684.

**[Sutton, 1984]** R.S. Sutton, Temporal credit assignment in reinforcement learning, (PhD Thesis, University of Massachusetts, Amherst, MA., 1984).

**[Watkins, 1989]** C.J.C.H. Watkins, Learning from Delayed Rewards (Ph.D. thesis, Cambridge University, 1989).

**[Kober et al., 2013]** J.Kober, J.A. Bagnell, J. Peters, Reinforcement learning in robotics: A survey, in *The Int. J. of Robotics Research*, 32, (2013), pp. 1238-1274.

**[Kohl et al., 2004]** N. Kohl and P. Stone, Policy gradient reinforcement learning for fast quadrupedal locomotion, in *Int. Conf. on Robotics and Automation* 3 (2004), pp. 2619-2624.

**[Endo et al., 2008]** G. Endo, J. Morimoto, T. Matsubara, J. Nakanishi, and G. Cheng, Learning CPG-based biped locomotion with a policy gradient method: Application to a humanoid robot, in *Int. J. of Robotic Research*, **27**(2) (2008), pp. 213–228.

**[Peters et al., 2008]** J. Peters and S. Schaal, Reinforcement learning of motor skills with policy gradients, in *Neural Networks*, **21**(4) (2008), pp. 682–697.

**[Theodorou et al., 2010]** E. Theodorou, J. Buchli, and S. Schaal, Reinforcement learning of motor skills in high dimensions, in *Int. Conf. on Robotics and Automation* (2010) pp. 2397-2403.

**[Peters et al., 2010]** J. Peters, K. Mulling, and Y. Altun, Relative entropy policy search, in *AAAI Conference on Artificial Intelligence* (2010), pp. 1607-1612.

**[Kalakrishnan et al., 2011]** M. Kalakrishnan, L. Righetti, P. Pastor, and S. Schaal, Learning force control policies for compliant manipulation, in *Int. Conf. on Intelligent Robots and Systems* (2011), pp. 4639-4644.

**[Abbeel et al., 2006]** P. Abbeel, A. Coates, M. Quigley, and A. Ng, An application of reinforcement learning to aerobatic helicopter flight, in *Advances in Neural Information Processing Systems* (2006) pp. 1-8.

**[Pastor et al., 2009]** P. Pastor, H. Hoffmann, T. Asfour, and S. Schaal, Learning and generalization of motor skills by learning from demonstration, in *Int. Conf. on Robotics and Automation* (2009), pp. 763-768.

**[Schulman et al., 2015]** J. Schulman, S. Levine, P. Moritz, M. Jordan, and P. Abbeel, Trust region policy optimization, in *Int. Conf. on Machine Learning* (2015), pp. 1889-1897.

**[Levine et al., 2016]** S. Levine, C. Finn, T. Darrell, and P. Abbeel, End-to-end training of deep visuomotor policies, *J. of Machine Learning Research*, **17**(1) (2016), pp. 1334-1373.

**[Deisenroth et al., 2011]** M. Deisenroth and C. Rasmussen, PILCO: a model-based and data efficient approach to policy search, in *Int. Conf. on Machine Learning* (2011), pp. 465-472.

**[Moldovan et al., 2015]** T. Moldovan, S. Levine, M. Jordan, and S. Abbeel, Optimism-driven exploration for nonlinear systems, in *Int. Conf. on Robotics and Automation* (2015), pp. 3239-3246.

**[Sadeghi et al., 2018]** F. Sadeghi, A. Toshev, E. Jang and S. Levine, Sim2Real Viewpoint Invariant Visual Servoing by Recurrent Control, in *IEEE Conf. on Computer Vision and Pattern Recognition* (2018), pp. 4691-4699.

**[Afanasyev et al., 2015]** I., Afanasyev, A., Sagitov, E., Magid. ROS-based SLAM for a Gazebo-simulated mobile robot in image-based 3D model of indoor environment. In *Int. Conf. on Advanced Concepts for Intelligent Vision Systems*, (Springer, Cham, 2015), pp. 273-283.

**[Sokolov et al., 2016]** Sokolov M. et al. 3D modelling and simulation of a crawler robot in ROS/Gazebo //*Proceedings of the 4th International Conference on Control, Mechatronics and Automation*. – ACM, 2016. – С. 61-65.

**[Лавренов и др., 2018]** Лавренов Р. О. и др., Робот "Сервосила Инженер": Разработка сервера передачи видеопотока и интерфейса управления под фреймворк ROS // *Известия ЮФУ. Технические науки*. – 2018. – №. 1. – С. 294-309.