ОБУЧЕНИЕ С ПОДКРЕПЛЕНИЕМ КАК МЕТОД УПРАВЛЕНИЯ МОБИЛЬНЫМ РОБОТОМ: СРАВНЕНИЕ С ПИ-РЕГУЛЯТОРОМ

Дружинин Н.С.[1], Беляев А.С.[2]
[1] ТПУ, ИШИТР, гр.8Е02, e-mail: nsd11@tpu.ru
[2] ТПУ, ИШИТР, к.т.н., ст. преподаватель, e-mail: asb22@tpu.ru

Аннотация

В исследовании рассматривается применение метода обучения с подкреплением (RL) для управления мобильным роботом Festo Robotino V1.6, в сравнении с традиционным подходом использования PI-регулятора. Основное внимание уделено техническим аспектам реализации и безопасности обучения, с особым акцентом на использовании нейросетевых моделей в контролируемой среде. Экспериментальная часть демонстрирует, что RL позволяет достигать высокой точности в выполнении заданной траектории движения, превосходя результаты, получаемые с помощью PI-регулятора. Анализ показывает значительные перспективы применения обучения с подкреплением в робототехнике для повышения адаптивности и эффективности управления в динамичных и неопределенных условиях. Отдельно отмечается важность дальнейшего изучения и развития методов RL, учитывая их потенциал в интеграции искусственного интеллекта в реальные условия эксплуатации робототехнических систем.

Ключевые слова: обучение с подкреплением (RL), мобильные роботы, многорукие бандиты, ПИ-регулятор, нейросетевая модель.

Введение

В современной робототехнике применяется широкий спектр методов управления, позволяющих достигать высокой эффективности и адаптивности в различных условиях эксплуатации. Одним из передовых подходов является обучение с подкреплением (Reinforcement Learning, RL), которое позволяет мобильным роботам самостоятельно находить оптимальные стратегии поведения в неопределенной среде.

Одним из ключевых преимуществ RL является возможность минимизации затрат на разработку управляющих систем для сложных объектов. В традиционных подходах значительные ресурсы уходят на математическое моделирование объектов управления, что включает в себя описание множества параметров, влияния внешних возмущений и неопределенностей. В случае с RL, требуется значительно меньше усилий для начального этапа разработки, так как алгоритм способен самостоятельно "изучить" объект управления в процессе обучения. Это существенно снижает как временные, так и финансовые затраты на разработку, делая RL экономически выгодным решением.

Тем не менее, несмотря на преимущества, существуют и определенные сложности в применении RL. Одна из главных проблем – необходимость первичного обучения в безопасной среде, чтобы избежать возможного выхода объекта из строя в процессе эксплуатации. В научной литературе широко описаны методики обучения в симуляторах, что позволяет предварительно "подготовить" алгоритмы управления к работе в реальных условиях с целью уменьшения риска для оборудования [1–3].

Целью данной работы является разработка и тестирование алгоритма обучения с подкреплением для задачи управления мобильным роботом Festo Robotino V1.6, который должен эффективно передвигаться по различным типам поверхности [4, 5]. Особое внимание уделяется безопасности обучения – используется заранее разработанная нейросетевая модель мобильного робота, что позволяет проводить обучение в контролируемой и безопасной среде. Кроме того, необходимо провести сравнение эффективности и экономической выгоды применения обучения с подкреплением в сравнении с традиционными методами управления, такими как PI-регулятор, с акцентом на возможности оптимизации процесса управления мобильными роботами в сложных условиях.

Разработка многоруких бандитов для Robotino V2

Для первичного обучения алгоритмов обучения с подкреплением используется одна из разработанных и протестированных ранее «цифровых теней» мобильного робота. Структурная схема цифровой тени представлена на рисунке 1, где V^3_x, V^3_y, V^3_z – заданные проекции заданных скоростей в локальной системе координат; w^3_1, w^3_2, w^3_3 – заданные угловые скорости моторов; w^P_1, w^P_2, w^P_3 –

реальные угловые скорости моторов; I_1, I_2, I_3 – потребляемые силы токов моторов; $w^э_1$, $w^э_2$, $w^э_3$ – эффективная скорость колёс с учётом эффекта проскальзывания; Δx, Δy, Δz – перемещение мобильного робота по осям X и У, изменение его угловой ориентации.

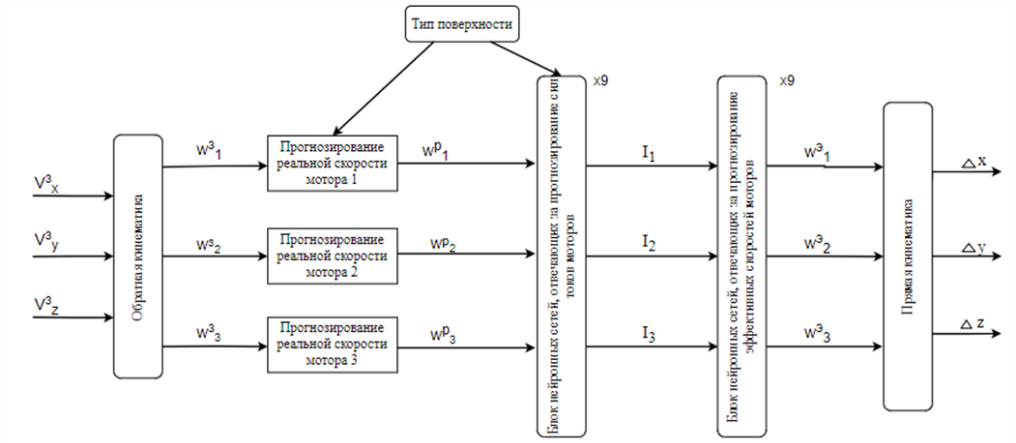

Рис. 1. «Нейросетевая» модель мобильного робота

Для реализации системы управления на основе обучения с подкреплением был выбран метод «многорукий бандит». При этом структура системы управления представляет собой два последовательно применённых «многоруких бандита». Таким образом, для работы двух связанных между собой бандитов необходимо задавать последовательно точки, лежащие на требуемой траектории. На рисунке 2 представлена структурная схема алгоритма управления мобильным роботом при помощи многоруких бандитов.

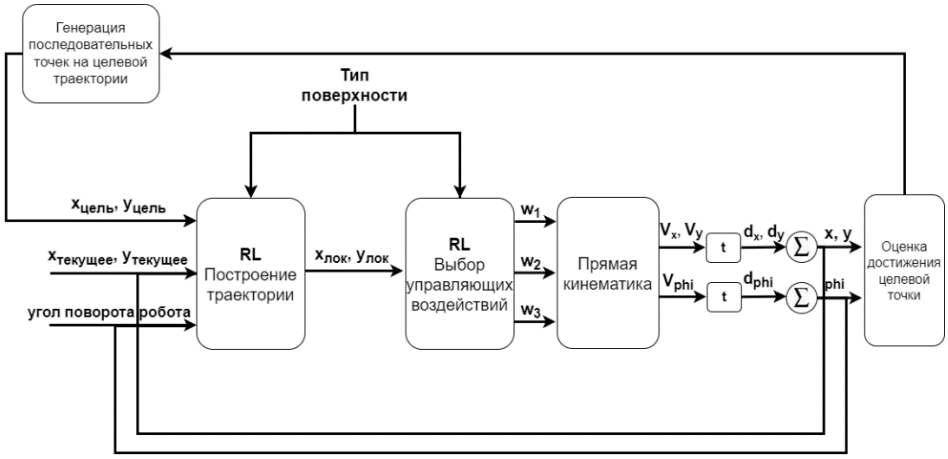

Рис. 2. Структурная схема алгоритма управления на основе многоруких бандитов

Бандит нижнего уровня (RL выбор управляющих воздействий) выбирает оптимальные скорости мобильного робота для достижения точек, лежащих на окружности, описанной вокруг него. На вход этого бандита идет целевая точка, в которую должен переместиться робот с учётом эффекта проскальзывания, на выходе бандит принимает решение, какие управляющие воздействия ему подать на моторы робота, чтобы оказаться в заданной точке. При обучении использовалась нейросетевая модель, описанная ранее на рисунке 1. Возможные точки, в которые бандит может переместиться, представлены на рисунке 3.

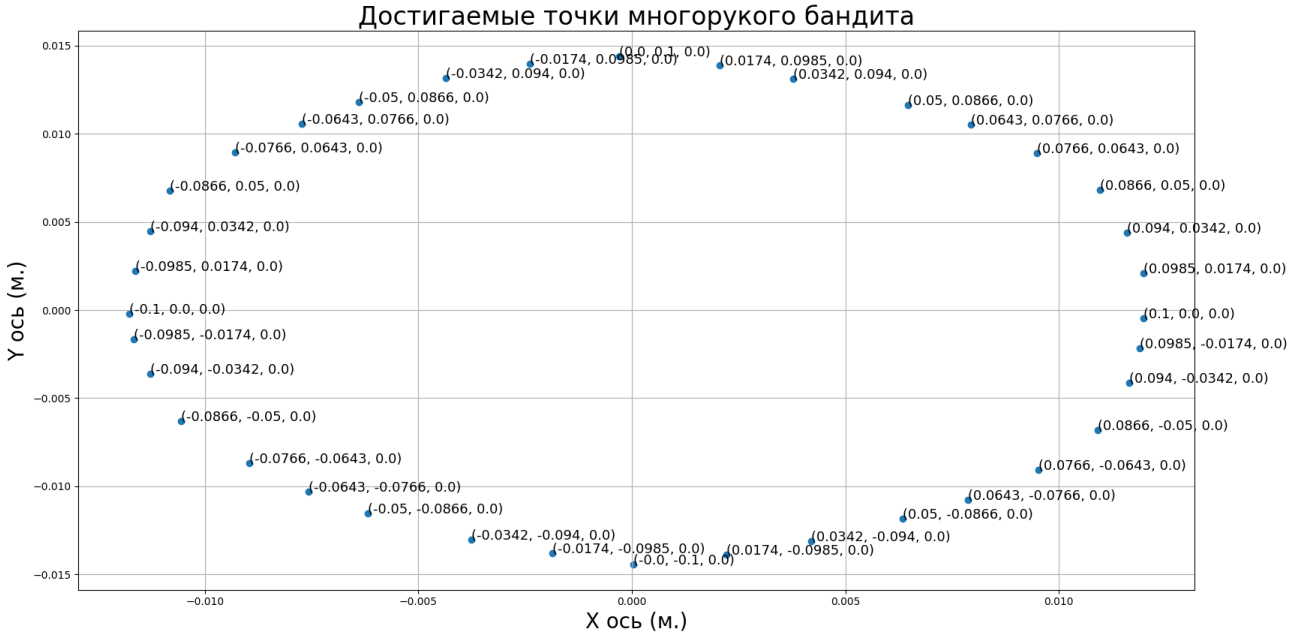

Рис. 3. Достигаемые точки первого многорукого бандита

Бандит верхнего уровня (RL построение траекторий) выполняет выбор оптимальной (по уменьшению расстояния) траектории для достижения целевой точки. На вход бандиту идет целевая точка, в которой должен оказаться мобильный робот после выполнения нескольких управляющих воздействий, на выходе бандит выбирает состояние, в котором он должен оказаться на следующем шаге, чтобы приблизиться к целевой точке. Полученная точка идет на вход первому бандиту. Данный бандит обучен доходить из любой точки дискретизированного полигона в любую точку поля.

Разработка PI регулятора для Robotino V2

Для оценки полученных результатов также реализовано классическое управление мобильным роботом на основе PI регулятора по положению. Настройка коэффициентов регулятора производилась на основе показаний нейросетевой модели робота. На рисунке 4 представлена структурная схема алгоритма управления мобильным роботом на основе PI регулятора.

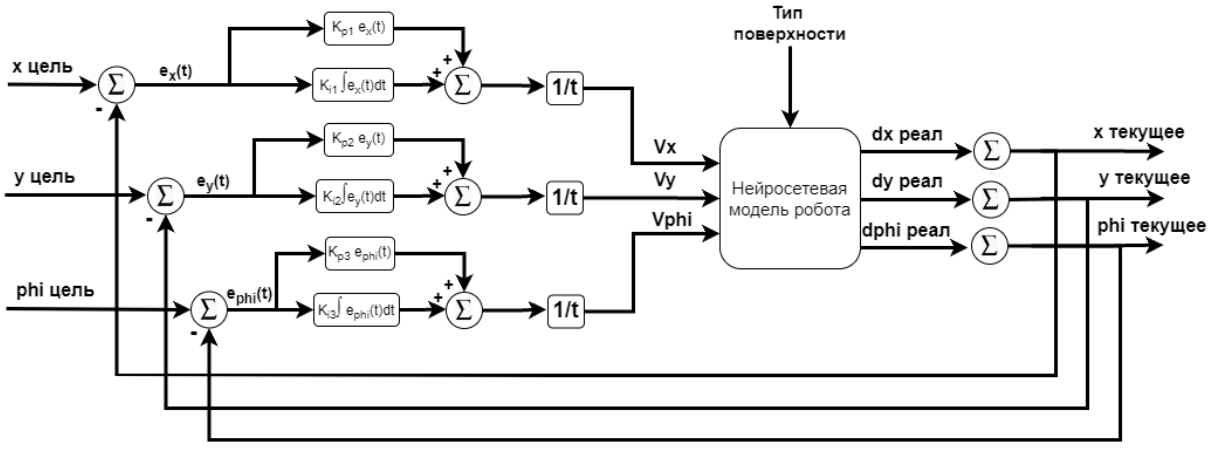

Рис. 4. Структурная схема алгоритма управления на основе PI регулятора по положению

Разработав два метода управления, следующим этапом стала апробация результатов на прохождение траектории квадрата.

Апробация многоруких бандитов и PI регулятора

В качестве оценки работоспособности алгоритмов, а также для их сравнительного анализа была поставлена задача проехать по траектории квадрата. На рисунке 5 представлены результаты работы алгоритмов.

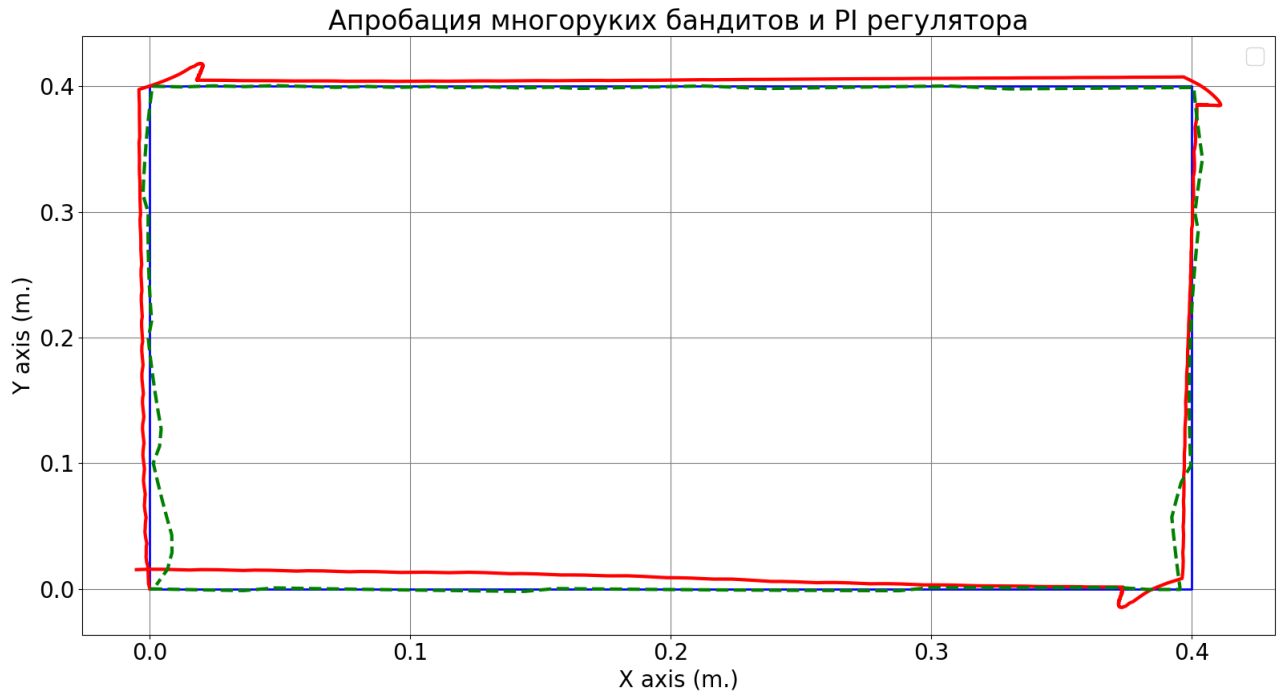

Рис. 5. Апробация алгоритмов на траектории «квадрат»

На рисунке 5 синяя линия – целевая траектория, красная линия – траектория с управлением через PI регулятор, зеленая пунктирная линия – траектория с управлением посредством многоруких бандитов.

Для численной оценки рассчитана метрика абсолютной интегральной ошибки по осям X и У. Поскольку построенные траектории включают в себя разное количество точек, то для оценки был использован инструмент интерполяции. В таблице 1 представлены численные результаты ошибок для рассматриваемых алгоритмов управления.

Таблица 1

Сравнение разработанных алгоритмов управления

	Многорукие бандиты	PI регулятор
Абсолютная интегральная ошибка по оси X, метры	0.042	0.091
Абсолютная интегральная ошибка по оси У, метры	0.013	0.114

Исходя из полученных результатов, можно сделать вывод о том, что в данной постановке задачи многорукие бандиты справляются с задачей управления лучше классического PI регулятора. Однако стоит отметить, что результаты регулятора в данном случае неоднозначные, поскольку настройка коэффициентов пропорционально и интегральной составляющей происходила вручную итеративным методом. Однако стоит отметить, что управление мобильным роботом при помощи многоруких бандитов является конкурентно способным методом.

Заключение

В ходе выполненной работы был проведен сравнительный анализ двух методов управления мобильным роботом Festo Robotino V1.6: классического PI-регулятора и обучения с подкреплением,

реализованного через многорукие бандиты. Исследование показало, что алгоритмы обучения с подкреплением могут быть не только экономически выгодными, но и технически эффективными в задачах управления сложными системами, такими как мобильные роботы, в условиях неопределенности и динамично изменяющейся среды.

Следует отметить, что для более точной оценки работоспособности алгоритмов управления требуется провести ряд тестов на реальном роботе. Данный этап является следующим в разработке рассматриваемого подхода. В обучении с подкреплением он носит название sim-to-real.

Благодаря использованию нейросетевой модели для первичного обучения в безопасной симулированной среде, удалось значительно снизить риски выхода реального робота из строя и оптимизировать процесс настройки алгоритмов управления. Результаты апробации показали, что многорукие бандиты способны обеспечить более точное следование заданной траектории по сравнению с PI-регулятором, что подтверждает их потенциал в решении подобных задач.

Однако необходимо отметить, что выбор метода управления должен основываться на конкретных требованиях и условиях эксплуатации робота. Несмотря на преимущества обучения с подкреплением, в некоторых случаях традиционные методы могут оказаться предпочтительнее из-за их предсказуемости и стабильности.

Заключительно, результаты данной работы подчеркивают важность дальнейших исследований в области применения обучения с подкреплением в робототехнике и разработки новых методов, которые позволят эффективно сочетать возможности искусственного интеллекта с требованиями реального мира.

Список использованных источников

1. X.B. Peng, M. Andrychowicz, W. Zaremba and P. Abbeel, "Sim-to-Real Transfer of Robotic Control with Dynamics Randomization," 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, QLD, Australia. – 2018. – C. 3803-3810. – doi: 10.1109/ICRA.2018.8460528.
2. Tan, Jie & Zhang, Tingnan & Coumans, Erwin & Iscen, Atil & Bai, Yunfei & Hafner, Danijar & Bohez, Steven & Vanhoucke, Vincent. Sim-to-Real: Learning Agile Locomotion For Quadruped Robots. – 2018
3. Wahlström, Niklas & Schön, Thomas & Deisenroth, Marc. From Pixels to Torques: Policy Learning with Deep Dynamical Models. – 2015
4. A.S. Belyaev, O.A. Brylev, E.A. Ivanov, Slip Detection and Compensation System for Mobile Robot in Heterogeneous Environment, IFAC-PapersOnLine. – Volume 54. – Issue 13. – 2021. – Pages 339-344. – ISSN 2405-8963. – URL: https://doi.org/10.1016/j.ifacol.2021.10.470.
5. Andrakhanov A., Belyaev A. Navigation learning system for mobile robot in heterogeneous environment: Inductive modeling approach, (2017), Proceedings of the 12th International Scientific and Technical Conference on Computer Sciences and Information Technologies, CSIT. – 2017. 1, art. – no. 8098846. – pp. 543 – 548. Cited 2 times. – DOI: 10.1109/STC-CSIT.2017.8098846.
6. F. Cursi, W. Bai, W. Li, E.M. Yeatman and P. Kormushev, "Augmented Neural Network for Full Robot Kinematic Modelling in SE(3)," in IEEE Robotics and Automation Letters. – vol. 7. – no. 3. – pp. 7140-7147. – July 2022. – doi: 10.1109/LRA.2022.3180428.
7. Gu Fang and M.W., M.G. Dissanayake, "Neural networks for modelling robot forward dynamics," Proceedings of ICNN'95 - International Conference on Neural Networks. – 1995. – pp. 2715-2719 – vol.5. – doi: 10.1109/ICNN.1995.487965.
8. Singh, B., Kumar, R. & Singh, V.P. Reinforcement learning in robotic applications: a comprehensive survey. Artif Intell – Rev 55. – 945-990. – 2022. – URL: https://doi.org/10.1007/s10462-021-09997-9.
9. Polydoros, Athanasios & Nalpantidis, Lazaros. (2017). Survey of Model-Based Reinforcement Learning: Applications on Robotics. Journal of Intelligent & Robotic Systems. –86. –153-. 10.1007/s10846-017-0468-y.
10. S. Shao et al., "Towards Hardware Accelerated Reinforcement Learning for Application-Specific Robotic Control," 2018 IEEE 29th International Conference on Application-specific Systems, Architectures and Processors (ASAP), Milan, Italy. – 2018. – pp. 1-8. – doi: 10.1109/ASAP.2018.8445099.