ЭФФЕКТИВНОЕ УПРАВЛЕНИЕ ПРОИЗВОДСТВЕННЫМ МОБИЛЬНЫМ РОБОТОМ НА ОСНОВЕ ПОДКРЕПЛЯЮЩЕГО ОБУЧЕНИЯ

В. В. Дёмин[1], А. С. Кабыш[1], В. А. Головко[1], R. Stetter[2]

[1] *Брестский государственный технический университет*
Брест, Беларусь
E-mail: spas.work@gmail.com
[2]*Университет Равенсбург-Вайнгартен,*
Вайнгартен, Германия

Описывается применение подкрепляющего обучения для мульти-агентной системы, цель которой – эффективное управления роботом. В рамках предлагаемого подхода используется модифицированный Q-learning алгоритм для множества агентов, позволяющий эффективно управлять каждым колесом, при этом агенты подстраиваются друг под друга.

Ключевые слова: мульти-агентные системы, подкрепляющее обучение, Q-learlning, эффективное управление роботом, интеллектуальное управление.

Введение

Эффективное управление мобильным роботом на производстве позволяет экономить множество ресурсов: время автономной работы, возможность перевозки более тяжелых грузов на более длинные расстояния, маневренность при перевозке габаритных грузов в ограниченном пространстве. Важными задачами является **оптимизация энергопотребления** и **оптимальное планирование траектории**. Задача энергосбережения в общем случае должна обеспечиваться подсистемами управления. Например, проблема энергопотребления моторов решается при их проектировании [1]. Подсистема управления не сможет влиять на КПД моторов, но должна обладать политикой эффективного управления [2] (оптимальная скорость мотора, оптимальный разгон, плавная функция торможения).

Оптимальное планирование траектории реализуется на уровне подсистемы планирования [3], [4], [5], [6]. Такая подсистема строит траекторию до цели и разбивает ее на части, которые могут быть представлены в виде кривых определенного радиуса и прямолинейных промежутков. Система управления роботом позволяет передвигаться (по возможности без остановок) по этой траектории, затрачивая как можно меньше энергии батарей.

В данной работе рассматривается разработка интеллектуального метода эффективного управления производственным роботом, разработанным в лаборатории университета Равенсбург-Вайнгартен. 3D модель робота изображена на рис. 1. Эта плат-

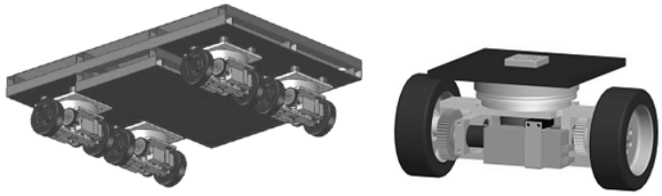


Рис. 1. 3D модель платформы

форма построена на основе инновационных модулей с низким энергопотреблением [7]. Интеллектуальная система управления основана на методах мультиагентных систем и обучения с подкреплением.

Обучение колесного модуля ориентации

Проведем декомпозицию роботизированной платформы на независимые колесные модули-агенты. Агенты располагаются в двумерной среде с привязкой к маяку, как показано на рис. 2*а*. Местоположение маяка определяется координатами (x_b, y_b). Радиус разворота ρ – расстояние от центра модуля до маяка.

Ошибка угла поворота вычисляется по формуле ϕ ($_{center}$ и φ_{robot} известны из среды).

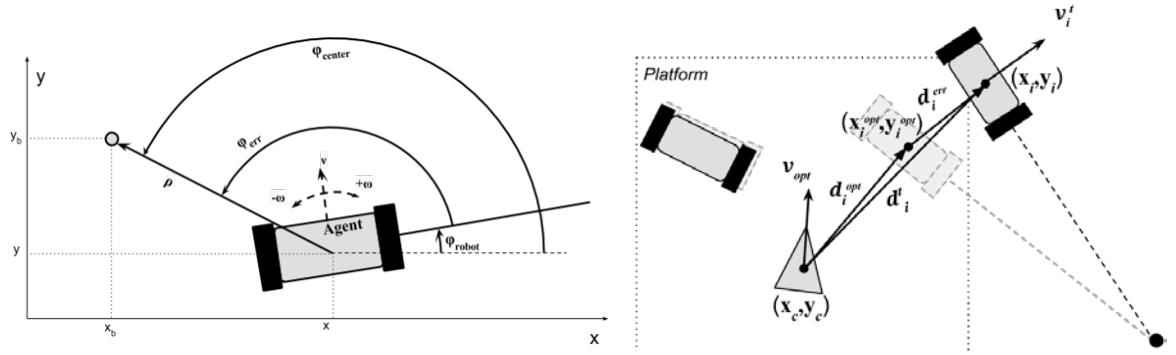
$$\varphi_{err} = \varphi_{center} - \varphi_{robot}\,. \tag{1}$$

Каждый модуль является самостоятельной автономной единицей. При объединении их в группу агентам необходимо координировать свои действия для поддержания формации. Одним из способов управления формацией агентов является создание виртуальной структуры [8]. Основная идея – определить *виртуального лидера* и *виртуальные координаты*, расположенные в центре формации относительно всей группы, и состояние каждого агента будет определяться относительно виртуального лидера или виртуального центра координат.

На рис. 2*б* (x_i, y_i) и (x_i^{opt}, y_i^{opt}) представляют координаты целевого и реального положения *i*-ого модуля, соответственно d_i^{err} представляет отклонение для *i*-ого модуля от правильного положения в платформе (2).

$$d_i^{err} = d_i^t - d_i^{opt}\,, \tag{2}$$

где d_i^t – расстояние до виртуального центра от текущего положения модуля, и d_i^{opt} – эталонное расстояние между виртуальным центром и *i*-ым агентом, которое получено из топологии платформы.



а *б*

Рис. 2. Состояние *агента а:* по отношению к маяку
б: Состояние платформы для *i*-го модуля

Интеллектуальная система управления модулей

Интеллектуальная система управления построена на основе обучения с подкреплением и решает две задачи: (1) позиционирует модули относительно точки вращения и (2) координирует согласованное движение модулей. Обучение с подкреплением является методом обучения оптимальному управлению автономных агентов в неизвестной

среде [9]. Используя *Q-learning* правило, ошибка временной разности между двумя последующими состояниями агентов вычисляется по следующей формуле:

$$\delta^t = r^t - \gamma \max_{a \in A(s^{t+1})} Q(s^{t+1}, a) - Q(s^t, a^t), \quad (3)$$

где r^t – значение награды полученное за выбор действия α^t в состоянии s^t, γ – коэффициент обесценивания отдаленных ценностей, $Q(s^t, a^t)$ – ценность выбора действия a^t в состоянии s^{t+1}. После каждого временного шага ценность прошлого состояния корректируется согласно ошибке временной разницы:

$$Q(s^t, a^t) = Q(s^t, a^t) + \alpha\delta^t. \quad (4)$$

Обучение агента позиционированию означает положительное подкрепление тех действий, которые минимизируют угол φ_{err}. Благодаря обучению и обобщению, агент способен поддерживать значение угла $\varphi_{err} \to 0$ при больших отклонениях, позиционироваться относительно любых углов, даже если они динамически изменяются с течением времени движения.

Для координации используется многоагентное расширение обучения с подкреплением. Основные аспекты подхода изложены в [10]–[12]. Идея разработанного подхода заключается в использовании значения влияния для координации между модулями и виртуальным лидером платформы. Цель – определение последовательности правильных действий. Правильное влияние должно награждать, отрицательное – наказывать. Проблемой проектирования является определение таких влияний в рамках индивидуальных наград.

Архитектура подкрепляющего обучения, решающая задачу кооперативного движения, изображена на рис. 3.

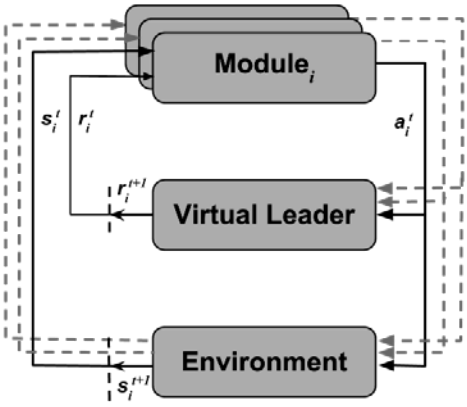


Рис. 3. Архитектура подкрепляющего обучения
для мультиагентной системы

Модуль i, находясь в состоянии s_i^t, выбирает действие α_i^t, используя текущую стратегию выбора действий, и переходит в следующее состояние s_i^{t+1}. Платформа получает данные об изменениях после выполнения действия, вычисляет и присваивает награду r_i^{t+1} модулю как обратную связь успешности данного действия.

Схожий *Q*-learning алгоритм (3) может быть использован для обновления политики модуля. Главное их отличие в том, что во втором случае награда назначается виртуальным лидером вместо окружающей среды:

$$\Delta Q_i(s_i^t, a_i^t) = \alpha[r_{p \to i}^{t+1} + \gamma \max_{a \in A(s_i^{t+1})} Q_i(s_i^{t+1}, a) - Q_i(s_i^t, a_i^t)], \quad (5)$$

Результаты моделирования

Первый этап моделирования заключается в позиционировании модулей относительно маяка. Таким образом, они занимают правильное положение для езды по кругу. Обучение происходит один раз для одного модуля перед кооперативным этапом моделирования. Изученные правила сохраняются и копируются для других агентов. Топология *Q*-функции, которая обучалась в течении 720 эпох, показана на рис. 4.

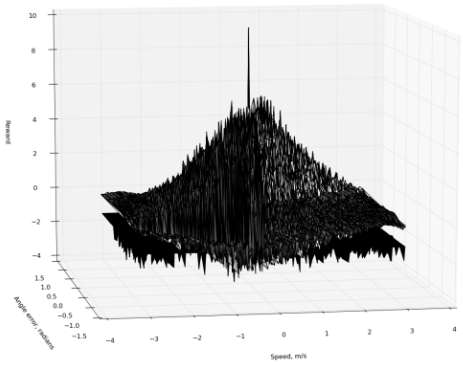


Рис. 4. Топология *Q*-функции после обучения

На рис. 5 показано начальное положение платформы (левое) и автоматическое позиционирование агентов (правое), используя обученную политику.

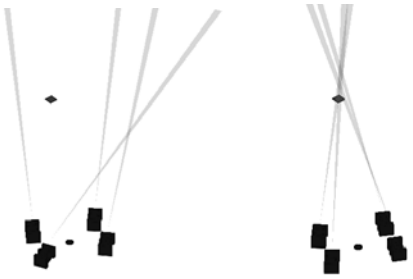


Рис. 5. Начальные и конечные позиции агентов

На рис. 6 показан результат эксперимента совместного движения платформы после обучения. Такое обучение в среднем занимает 11 000 эпох.

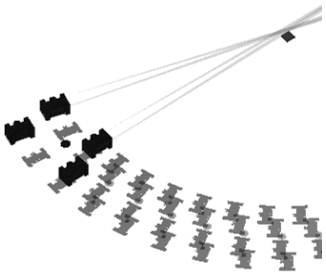


Рис. 6. Процесс совместного кругового движения платформы

Внешние параметры моделирования: шаг обучения a = **0.4**, коэффициент обесценивания γ = **0.7**, оптимальная скорость ω_{opt} = **0.8** рад/с, угол торможения φ_{stop} = **0.16** рад.

Заключение

Экспериментальная часть демонстрирует успешное применение мульти-агентного подхода на основе подкрепляющего обучения для задачи эффективного управления многоколесной роботизированной платформы. Предлагаемый подход включает множество *Q*-learning агентов, которые определяют оптимальное управление модулями относительно виртуального лидера. Достоинства разработанного подхода заключаются в адаптивности к изменению целей и масштабируемости по количеству агентов.

Для моделирования ω_{opt} и v_{opt} используются константные значения оптимальных угловой и линейной скоростей, чтобы показать применимость такого подхода. Для реального робота необходимо проводить расчеты такой функции, используя документацию на моторы [13] и параметры модулей.

Библиографические ссылки

1. *Walters D. G.* The Whole Life Efficiency of Electric Motors // Energy Efficiency Improvements in Electric Motors and Drives. Springer, 1997. P. 81–94.
2. *Barili A., Ceresa M., Parisi C.* Energy-saving motion control for an autonomous mobile robot // Industrial Electronics. Proceedings of the IEEE International Symposium on ISIE'95. IEEE, 1995. V. 2. P. 674–676.
3. *Balkcom D. J., Matthew T. M.* Extremal trajectories for bounded velocity differential drive robots // Robotics and Automation. Proceedings of IEEE International Conference on ICRA'00. IEEE, 2000. V. 3. P. 2479–2484.
4. *Duleba I., Sasiadek J. Z.* Nonholonomic Motion Planning Based on Newton Algorithm With Energy Optimization // IEEE Trans. Control Syst. Technol. 2003, V. 11. № 3. P. 355–363.
5. *Mei Y., Lu Y.-H., Hu Y. C., George C. S.* Lee Energy-Efficient Motion Planning for Mobile Robots // Robotics and Automation. Proceedings of IEEE International conference on ICRA'04. IEEE, 2004. V. 5. P. 4344–4349.
6. *Kaliukhovich D., Golovko V., Paczynski A.* Control algorithms for the mobile robot "Max" on a task of line following provided by intelligent image processing // Solid state phenomena. 2009. V 147. P. 35–42.
7. *Stetter R., Ziemniak P., Pachinski A.* Realization and Control of a Mobile Robot // Research and Education in Robotics-EUROBOT 2010, Communication in Computer and Information Science. Springer, 2011. V. 156. P. 130–140.
8. *Ren W., Sorensen N.* Distributed coordination architecture for multi-robot formation control // Robotics and Autonomous Systems, 2008. V. 56. № 4. P. 324–333.
9. *Sutton R. S., Barto A. G.* Reinforcement Learning: An Introduction // MIT Press, 1998. 322 pages.
10. *Kabysh A., Golovko V.* General model for organizing interactions in multi-agent systems // International Journal of Computing. 2012. V. 11. Issue 3. P. 224–233.
11. *Kabysh A., Golovko V.* Influence Learning for Multi-Agent Systems Based on Reinforcement Learning // International Journal of Computing. 2012. V. 11. Issue 1. P. 39–44.
12. *Kabysh A., Golovko V., Madani K.* Influence model and reinforcement learning for multi agent coordination // Journal of Qafqaz University, Mathematics and Computer Science. 2012. № 33. P. 58–64.
13. Maxon motor uk ltd. Brushless motor RE35 Graphite Brushes, 90 Watt, 12V DC motor datasheet. Available: http://www.maxonmotor.com/medias/sys_master/8806653427742/13_104_EN.pdf.