

$$A = C(F_p(R, U_c)F_p(R, U_r), CL)$$

This formula shows what action takes robot. C is comparison function to choose which of predicted results is better. But this function also has to consider who makes less errors, whom robot can trust. For that reason we've included a variable CL , which means a confidence level. It's a dynamic parameter that collects all result of previous evaluations and aggregates wrong decisions for each type of control. More actions are taken - more information about controllers is collected.

Moreover, in prediction function can be calculated not only appropriate state at the next moment of time, but some general goal. For instance, case when robot is to reach some point and complex of controllers commands leads other direction is set as wrong and robot has to search way by itself. Another of the ways to use such approach is controlling a group of robots by one controller. Controller doesn't need to communicate with every agent in group but sends some general commands.

References

- [1.] Dudek, G. Computational Principles Of Mobile Robotics. Second Edition / G.Dudek, M. Jenkin - UK: Cambridge University Press,2010. - 11 p.
- [2.] Norwig, P. Artificial Intelligence, A Modern Approach. Second Edition / S. Russel, P. Norwig – USA. Prantice Hall, 2003. - 557 p.

ИНТЕЛЛЕКТУАЛЬНАЯ СИСТЕМА УПРАВЛЕНИЯ АУТОНОМНЫМ МОБИЛЬНЫМ РОБОТОМ

Дёмин В.В., Кабыш А.С., Головки В.А.

Брестский государственный технический университет,
vvdemin@bstu.by

Рассматривается система эффективного управления мобильным роботом, в основе которой лежат алгоритмы обучения с подкреплением для сети колесных модулей. В рамках предлагаемого подхода эта сеть рассматривается как многоагентная система, в которой координация поведений агентов осуществляется виртуальным лидером. Предложена модифицированная модель обучения с подкреплением для адаптивной координации индивидуальных стратегий. Модифицированный Q learning алгоритм проводит обучение агентов эффективному управлению каждым колесом, в контексте группы, что позволяет агентам подстраиваться друг под друга.

Мобильные роботы, обучение с подкреплением, многоагентные системы, алгоритмы интеллектуального управления.

I. ВВЕДЕНИЕ

Эффективное управление мобильным роботом на производстве позволяет повысить общую производительность робота. Энергоэффективное управление роботом повышает время автономной работы, уменьшается общее энергопотребление, разумно расходуется общая энергия робота. Эффективная система управления обеспечивает прокладывание оптимальной траектории, что повышает маневренность при перевозке габаритных грузов в ограниченном пространстве, что в комбинации с энергоэффективным управлением дает возможность перевозки более тяжелых грузов на более длинные расстояния.

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 стратегией эффективного управления (оптимальная скорость мотора, оптимальный разгон, плавная функция торможения).

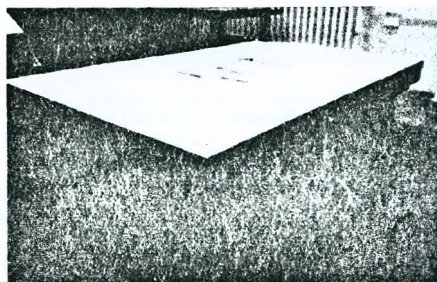


Рисунок 1 – Производственная грузовая мобильная платформа

Оптимальное планирование траектории, как правило, реализуется на уровне подсистемы планирования [2]. 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.

Задача эффективного управления при перевозке тяжелых грузов на современных производствах является актуальной для автономных мобильных грузовых платформ. Одна из таких платформ (рис. 1) – производственный грузовой робот, разработанный в лаборатории университета Равенсбург-Вайнгартена [4]. Основные характеристики платформы: размер 1200 см на 800 см, максимальная грузоподъемность 500 кг при комплектации 4-мя модулями, ёмкость аккумуляторов 52Ah, минимальная скорость 1 м/с, независимое управление каждым модулем. Представленная на рис. 1 платформа использует четыре модуля, но так же возможно собрать платформу и с большим количеством модулей.

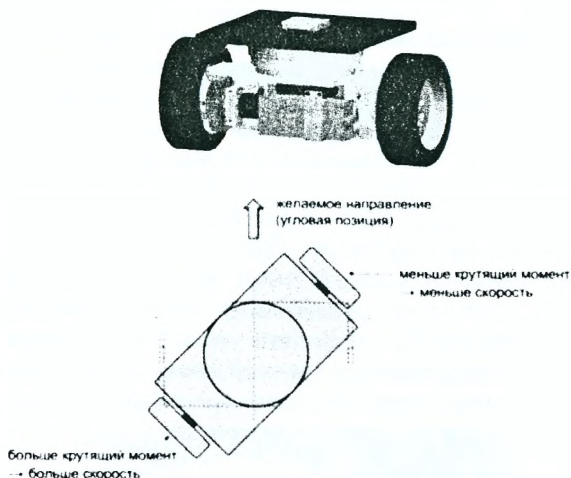


Рисунок 2 – Инновационный модуль

Платформа построена на базе инновационных модулей [4]. Такой модуль (рис. 2) состоит из двух колес, приводимых в движение двумя независимыми моторами, и имеет дифференциальную схему управления. К платформе такие модули подсоединены подшипником, что позволяет им поворачиваться относительно платформы на любой угол.

В статье рассматривается проблема эффективного управления многоколесным роботом на примере производственной грузовой мобильной платформы. Ключевым вкладом этой статьи является предложенная модель координации колесных модулей на основе виртуального лидера и обучения с подкреплением. Представленная модель решает проблему кругового движения платформы относительно центра разворота, даже если он динамически меняет свое положение. Подход требует лишь информации о положении агентов относительно центра платформы. По сравнению с аналогичными подходами кругового движения [3], предложенный алгоритм позволяет повысить эффективность потребления энергии роботом.

II. УПРАВЛЕНИЕ МОДУЛЕМ

Традиционный подход для управления платформой – расчет кинематики и моделирование инверсной кинематики [4]. В таком случае, если будут добавлены или демонтированы модули платформы, это потребует повторных расчетов и повторной конфигурации подсистемы управления. Существующий расчет кинематики может быть применен только для симметричного движения по

кругу, при движении по которому центр разворота находится на центральной оси SG (пример предыдущего расчета на рис. 3). К примеру, невозможно использовать автомобильную схему управления или любую другую.

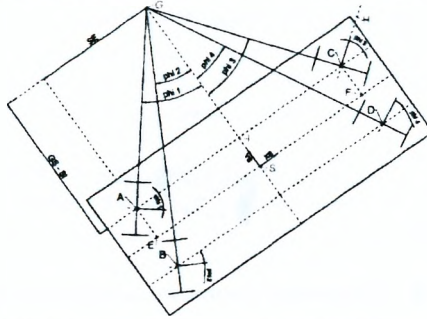


Рисунок 3 – Кинематическая модель платформы с четырьмя модулями для симметричного кругового движения

Кинематическая схема управления модулем описывается дифференциальным уравнением движения [5]:

$$\dot{x}(t) = \frac{R}{2}(v_r + v_l) \cos \theta(t) \quad (1)$$

$$\dot{y}(t) = \frac{R}{2}(v_r + v_l) \sin \theta(t) \quad (2)$$

$$\dot{\theta}(t) = \frac{R}{L}(v_r - v_l) \quad (3)$$

Здесь $[x, y, \vartheta] \in R^2 \times [-\pi, \pi]$ определяют положение робота, v_r и v_l – скорости правого и левого колес соответственно, R – радиус колеса, L – расстояние между левым и правым колесом [5]. Параметры v – линейная скорость, и w – угловая скорость используются для управления модулем:

$$v = \frac{R}{2}(v_r + v_l) \quad (4)$$

$$w = \frac{R}{2}(v_r - v_l) \quad (5)$$

III. ОБУЧЕНИЕ КОЛЕСНОГО МОДУЛЯ

Проведем декомпозицию роботизированной платформы на независимые колесные модули-агенты. Каждый модуль является самостоятельной автономной единицей, с индивидуальным поведением. Агенты располагаются в двумерной среде с привязкой к маяку, как показано на рисунке 4. Местоположение маяка определяется координатами (x_b, y_b) . Маяк – точка в пространстве,

обозначающая центр разворота робота. Радиус разворота ρ – это расстояние от центра модуля до маяка. Ошибка угла поворота вычисляется по формуле 6 (φ_{center} и φ_{robot} известны из среды).

$$\varphi_{err} = \varphi_{center} - \varphi_{robot} \quad (6)$$

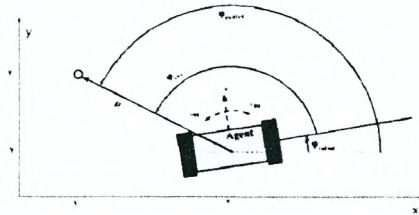


Рисунок 4 – Состояние агента по отношению к маяку

При объединении модулей в целостную группу, агентам необходимо координировать свои действия для поддержания целостности формации. Одним из способов управления формацией агентов является создание виртуальной структуры [6]. Основная идея данного подхода состоит в определении виртуального лидера, задающего виртуальные координаты. Таким образом, состояние каждого агента будет определяться относительно виртуального лидера или виртуального центра координат.

На рисунке 5 (x_i, y_i) и (x_i^{opt}, y_i^{opt}) представляют координаты реального и целевого положения i -го модуля соответственно, \vec{d}_i^{err} представляет вектор отклонения для i -го модуля от правильного положения в платформе (7).

$$\vec{d}_i^{err} = \vec{d}_i' - \vec{d}_i^{opt} \quad (7)$$

где \vec{d}_i' – вектор расстояния до виртуального центра от текущего положения модуля, и \vec{d}_i^{opt} – вектор эталонного расстояния между виртуальным центром и i -м агентом, которое получено из топологии платформы.

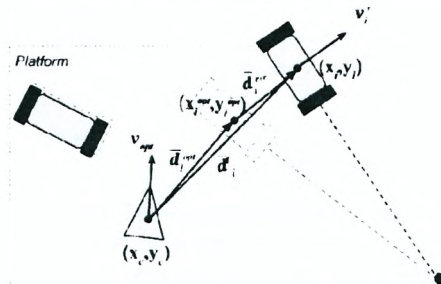


Рисунок 5 – Состояние платформы для i ого модуля

IV. ИНТЕЛЛЕКТУАЛЬНАЯ СИСТЕМА УПРАВЛЕНИЯ МОДУЛЕЙ

Интеллектуальная система управления [7] построена на основе обучения с подкреплением и решает две задачи: позиционирует модули относительно точки вращения (6) и координирует согласованное движение модулей (7). Обучение с подкреплением является методом обучения автономных агентов для нахождения оптимальных стратегий поведения в неизвестной среде [8]. Метод основан на исследовании агентом пространства состояний и нахождении тех пар состояние-действие, которые обеспечивают суммарный ожидаемый максимум награды.

Обучение агента позиционированию означает положительное подкрепление тех действий, которые минимизируют угол ошибки. В результате обучения и обобщения полученной стратегии, агент способен поддерживать значение угла поворота при больших отклонениях, позиционироваться относительно любых углов поворота, даже если они изменяются во время движения. Обученные агенты способны держать строй формации, минимально мешая друг другу при движении реального робота и тем самым увеличивая энергоэффективность управления.

Для обучения скоординированному поведению агентов используется расширение стандартной модели многоагентного обучения с подкреплением, основанное на использовании виртуального лидера [9]–[11]. Идея подхода в том, что виртуальный лидер оценивает влияние поведения каждого агента на общую целостность платформы. Конструктивные влияние рассматриваются как положительные подкрепления, а деструктивные – как негативные.

Модель многоагентного обучения с подкреплением с использованием виртуального лидера, решающая задачу кооперативного движения, изображена на рисунке 6.

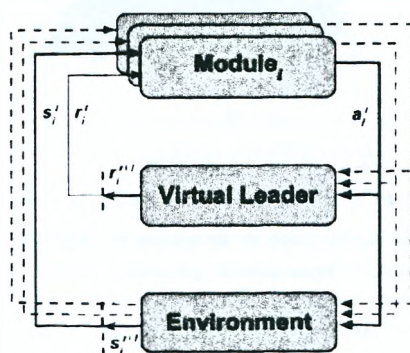


Рисунок 6 – Архитектура подкрепляющего обучения для мультиагентной системы

Модуль i , находясь в состоянии s_i^t , выбирает действие a_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 с Q-learning алгоритм (8) используется для обновления стратегии модуля. Главное отличие заключается в том, что в (8) награда назначается виртуальным лидером, а не окружающей средой:

$$\Delta Q_i(s_i^t, a_i^t) = \alpha[r_i^{t+1} + \gamma \max_{a \in A(s_i^{t+1})} Q_i(s_i^{t+1}, a) - Q_i(s_i^t, a_i^t)] \quad (8)$$

V. РЕЗУЛЬТАТЫ МОДЕЛИРОВАНИЯ

Первый этап моделирования заключается в позиционировании модулей относительно маяка. В результате агенты занимают правильное положение для движения по кругу. Обучение происходит один раз для одного агента перед кооперативным этапом моделирования [7]. Полученные правила сохраняются и копируются для других агентов. Топология Q-функции, которая обучалась в течение 720 эпох, показана на рис. 7.

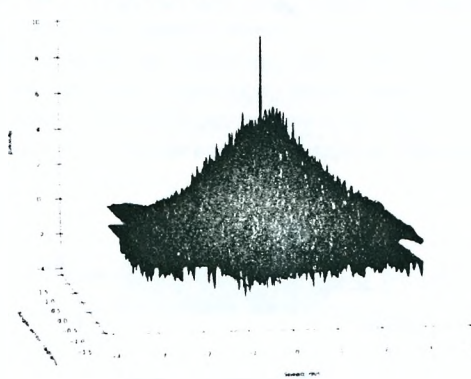


Рисунок 7 – Топология Q-функции после обучения одного модуля

На рис. 8а показано начальное положение платформы, на рис. 8б показан результат автоматического позиционирования агентов, используя обученную стратегию.

На рис. 9 показан результат эксперимента совместного движения платформы после обучения. Такое обучение в среднем занимает 11000 эпох. Внешние параметры моделирования: шаг обучения $\alpha = 0,4$, коэффициент обесценивания $\gamma = 0,7$, оптимальная скорость $\omega_{opt} = 0,8$ рад/с, угол торможения $\varphi_{stop} = 0,16$ рад.

Для моделирования ω_{opt} используется как константное значение скорости, чтобы показать применимость такого подхода. Для реального робота должны производиться расчёты такой функции используя документацию на моторы [12] и другие параметры (передаточное число, размер, загруженность). φ_{stop} в данном случае подбирается вручную. В дальнейших исследованиях планируется разработать алгоритм автоматического подбора угла торможения.

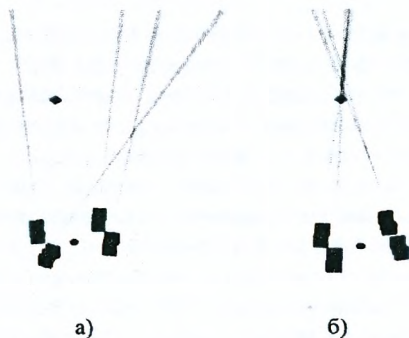


Рисунок 8 – Результаты моделирования обученных агентов платформы задаче поворота



Рисунок 9 – Результаты моделирования обученных агентов платформы задаче совместного движения

VI. ВЫВОДЫ

Экспериментальная часть демонстрирует успешное применение многоагентного подхода на основе виртуального лидера с использованием обучения с подкреплением для задачи эффективного управления многоколесной роботизированной платформой. Предлагаемый подход включает множество Q-learning агентов, которые определяют оптимальное управление модулями относительно виртуального лидера. Достоинства разработанного подхода:

- Декомпозиция обозначает, что вместо построения глобальной Q-функции мы строим множество локальных;

- Адаптивность – платформа адаптирует свое поведение для динамически изменяемого маяка и перенастраивает свою траекторию;
- Масштабируемость и обобщающая способность – один метод обучения используется для множества агентов, для любой позиции маяка и для любой позиции агента на платформе.

REFERENCES

- [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] Mei Y., Lu Y.-H., Hu Y. C., Lee G. Energy-Efficient Motion Planning for Mobile Robots // *Robotics and Automation. Proceedings of IEEE International conference on ICRA'04*. IEEE. – 2004. – Vol. 5. – P. 4344–4349.
- [3] Benedettelli, D., Garulli, A. and Giannitrapani, A. Experimental validation of collective circular motion for nonholonomic multi-vehicle systems // *Robotics and Autonomous Systems*. 2010. No 58. P. 1028–1036.
- [4] 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. – Vol. 156. – P. 130–140.
- [5] G. Dudek and M. Jenkin, *Computational Principles of Mobile Robotics*. Cambridge University Press. – 2010.
- [6] Ren, W. Distributed coordination architecture for multi-robot formation control / W. Ren, N. Sorensen // *Robotics and Autonomous Systems*. – 2008. – Vol. 56, № 4. – P. 324–333.
- [7] Dziomin, U.; Kabysh, A. A multi-agent reinforcement learning approach for the efficient control of mobile robot / U. Dziomin; A. Kabysh, V. Golovko; R. Stetter // *Intelligent Data Acquisition and Advanced Computing Systems (IDAACS)*. – 2013. – Vol.02. – P. 867–873.
- [8] Sutton, R.S. *Reinforcement Learning: An Introduction* / R.S. Sutton, A.G. Barto // MIT Press. – 1998. – 322 p.
- [9] Kabysh, A. General model for organizing interactions in multi-agent systems / A. Kabysh, V. Golovko // *International Journal of Computing*. – 2012. – Vol. 11. Issue 3. – P. 224–233.
- [10] Kabysh, A. Influence Learning for Multi-Agent Systems Based on Reinforcement Learning / A. Kabysh, V. Golovko // *International Journal of Computing*. – 2012. – Vol. 11, Issue 1. – P. 39–44.
- [11] Kabysh, A. Influence model and reinforcement learning for multi agent coordination / A. Kabysh, V. Golovko, K. Madani // *Journal of Qafqaz University, Mathematics and Computer Science*. – 2012. – № 33. – P. 58-64.
- [12] 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.