УДК 004.8

СИСТЕМА УПРАВЛЕНИЯ РОБОТОТЕХНИЧЕСКИМИ КОМПЛЕКСАМИ НА ОСНОВЕ ОБУЧЕНИЯ С ПОДКРЕПЛЕНИЕМ

Ф.А. Хуссейн (*houssein.firas@gmail.com*)
В.Б. Пархоменко (*vparhomenko@sfedu.ru*)
М.Ю. Бутенко (*butenko@sfedu.ru*)
Т.А. Гайда (*tgayda@sfedu.ru*)
Южный федеральный университет, г. Таганрог

Аннотация. В статье исследуется вопрос создания интеллектуальных систем управления робототехническими комплексами верхнего уровня, ориентированных на формирование тактики решения поставленных задач. Недавние успехи в области игрового искусственного интеллекта, основанного на глубоких искусственных нейросетях (ИНС), продемонстрировали их высокий потенциал в решении тактических задач в сложных динамических средах. Трансфер этих технологий в реальные робототехнические системы может послужить шагом к созданию новых прикладных решений в области функционирования автономных робототехнических комплексов верхнего уровня, ориентированных на формирование их тактики, успешно сочетающих классические алгоритмические и современные статистические методы.

Ключевые слова: робототехнические комплексы (РТК), искусственный интеллект, интеллектуальный агент, обучение с подкреплением.

Введение

В соответствии со взглядами отечественных и зарубежных специалистов в боевых действиях будущего одними из наиболее перспективных видов вооружения и военной техники будут робототехнические комплексы (РТК) военного назначения. При этом ряд специалистов предполагает, что широкомасштабное внедрение роботов и технологий робототехники изменит способы ведения операций и технический облик перспективных систем вооружения и военной техники, повысит эффективность их применения, а также обеспечит сокращение потерь личного состава [Цариченко и др., 2019], [Верба и др., 2016].

Одним из подходов к созданию и управлению этими комплексами являются жесткие неадаптивные алгоритмы. Несмотря на интеллектуализацию систем управления РТК, тактический уровень принятия решений основан на сложно структурированных комбинациях классических алгоритмов, заранее определяющих характер поведения автономных РТК в зависимости от условий функционирования. Примером реализации систем управления на основе жестких алгоритмов являются американские системы «Squad X», которые разрабатываются для предоставления разведывательной информации на уровне батальона. Наземные и воздушные роботы обеспечивают личный состав подразделений морской пехоты разведывательной информацией, поддерживая полную ситуационную осведомлённость как на пересечённой местности, так и в условиях городского боя [DARPA 2016].

Разработка подобных систем – чрезвычайно сложный и наукоемкий процесс, требующий глубокого погружения в предметную область и высокоточного описания всех возможных сценариев и их вариаций, которые могут возникнуть в динамичной непредсказуемой среде.

Под эмпирическими алгоритмами понимаются методы, результат функционирования которых зависит от полученного ранее опыта. Их применение требует обучения на заранее сформированных массивах обработанных и необходимым образом структурированных данных.

Однако, использование этих технологий в настоящее время ограничивается конкретными задачами управления движением. Например, модуль распознавания объектов в беспилотном автомобиле использует алгоритм YOLO [Yu Huang и др., 2020], выход которого передается на системы управления.

Искусственный интеллект показывает превосходный результат в стратегических и тактических задачах DotA2 (представляет собой многопользовательскую игру в жанре MOBA («многопользовательская онлайновая боевая арена»), в которой сражаются две команды по пять игроков) [OpenAI 2019], где одновременно осуществляется управление действиями нескольких объектов по результатам анализа информации о среде, описываемой более чем двадцатью тысячами параметров.

В связи со сложностью создания систем управления на основе жестких алгоритмов и последними успехами в области машинного оучения, в данной статье исследуется вопрос создания систем управления группой РТК на основе технологии обучения с подкреплением, которая способна обрабатывать всю доступную информацию о среде и формировать рациональную последовательность действий РТК, направленных на решение поставленной задачи.

1 Постановка задачи

Обучение с подкреплением — один из способов машинного обучения, в ходе которого система управления верхнего уровня (интеллектуальный агент) обучается, взаимодействуя с некоторой средой. Данный подход, с точки зрения кибернетики, является одним из видов кибернетического эксперимента [Sutton и др. 2018].

В целях определения схемы полного цикла обучения интеллектуального агента, в статье проводится исследование примера прикладной задачи управления, которая может быть описана следующим образом. Необходимо создать систему управления группой РТК для случая ведения боя против группы обороняющихся противников в заданном участке городской среды на основе технологии обучения с подкреплением и определить границы применимости этой системы при формировании тактики ведения боя.

Анализ технологии обучения интеллектуальных агентов позволяет установить, что для решения указанной задачи необходимо:

- сформировать среду для обучения;
- выбрать, реализовать и исследовать ряд алгоритмов обучения;
- выбрать, исследовать и модернизировать структуру ИНС;
- сформировать рациональную систему вознаграждений.

2 Реализация

2.1 Среда обучения

Функционирование объекта в среде соответствует базовым принципам Марковского процесса принятия решений (МППР) с конечным множеством состояний, включающего вероятности выигрыша и перехода в состояние, которые обычно являются случайными, стационарными величинами в рамках задачи.

Агент может обучаться в реальном мире, где он ограничен законами физики, или в виртуальной среде, имитирующей реальную среду функционирования. Последнее является предпочтительным в связи с большим количеством нерациональных действий агента при обучении, которые могут нанести физический ущерб самому РТК и окружающей среде.

Примером реализации обучения в виртуальной среде является работа [Andrychowicz и др., 2017], в которой продемонстрирована высокая эффективность обучения нейросети для роботизированной руки в виртуальной среде, с последующим применением обученной нейросети к управлению реальной робо-рукой.

С целью обучения агентов, была разработана виртуальная среда обучения (Рис. 1) с возможностью случайной генерации начального направления и положения агентов, положений препятствий и их размеров. Агент управляет своей линейной ($\in [-70,70]$ метр/секунду) и угловой ($\in [-\pi,\pi]$ радиан/секунду) скоростями. Сектор поражения противника больше, чем у собственного РТК. Эпизод завершается при: столкновении с препятствием; выходе за пределы среды; поражении вражеских РТК; поражении собственных РТК противником.

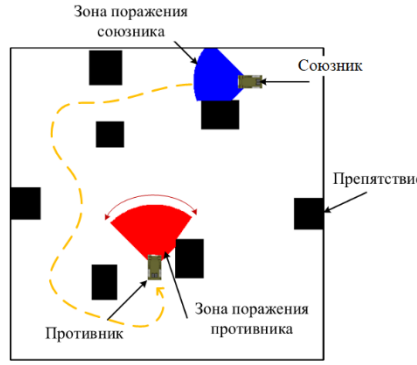

Рис. 1. Среда обучения агента.

2.2 Определение алгоритмов обучения

К основным группам подходов, представляющих алгоритмы обучения с подкреплением (Reinforcement Learning, RL), относятся: безмодельные и основанные на моделях. Под моделью подразумевается функция, которая предсказывает вероятности перехода между состояниями и полученные вознаграждения. Подходы на основе моделей статистически более эффективны, чем безмодельные методы [Dayana и др., 2008], но в связи с тем, что среда, с которой мы имеем дело, является динамической и может сильно изменяться, создание модели, которая описывает все возможные варианты, является очень сложной задачей.

Безмодельные подходы используют опыт для непосредственного изучения одной или двух более простых величин. Функция политики $\pi(s|a)$ определяется как отображение состояний на вероятности выбора каждого возможного действия. Функция ценности состояния-действия $q(s,a)$ оценивает, насколько целесообразно выполнение какого-либо действия в каком-либо состоянии [Sutton и др. 2018]). Эти подходы могут достичь того же оптимального по заданному критерию поведения, но без использования модели.

Безмодельный подход включает в себе ряд алгоритмов. В связи с тем, что пространство принятых решений является недескретным (ленейная скорость $\in [-70{,}70]$ метр/секунду, угловая скорость $\in [-\pi, \pi]$ рад/секунду), то для нас подходят следующие алгоритмы обучения:

- The Proximal Policy Optimization (PPO) [Schulman и др., 2017];
- Soft Actor-Critic (SAC) [Haarnoja и др., 2017]
- Deep Deterministic Policy Gradient (DDPG) [Lillicrap и др., 2019];
- Advantage Actor Critic (A2C) [Mnih и др., 2016].

Выбор наиболее подходящих алгоритмов для решения нашей задачи осложняется тем, что для этого не существует определённой методики.

Для осуществления выбора необходимо провести исследование безмодельных алгоритмов со всеми возможными вариациями их гиперпараметров, то есть, параметров, значения которых используется для управления процессом обучения, например: количество шагов в среде, состовляющих выборку обучения; размер выборки для обучения; количество эпох; коэффициент дисконтирования; скорость обучения и т.д.

2.3 Описание архитектуры нейронной сети

Свёрточная нейронная сеть использовалась для обработки состояния среды, которое передавалось на вход в виде изображения. Параллельно две сети прямого распространения использовались для обработки координат и направлений движения агентов и противников. Данные на выходе этих трёх ИНС являются входом четвёртой ИНС, которая аппроксимирует функцию политики. В итоге, последняя ИНС определяет значения линейной и угловой скоростей агента на следующем шаге (рис. 2). Достоинство выбранной архитектуры состоит в том, что агент получает дополнительную информацию о своём состоянии и о состоянии противников, что положительно влияет на сходимость процесса обучения.

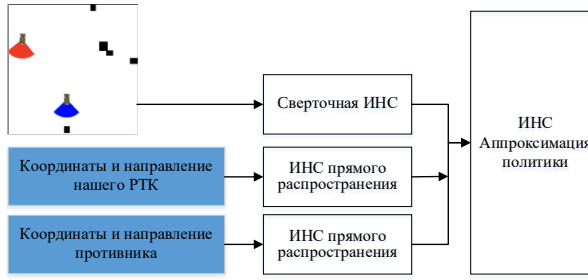

Рис. 2. Архитектура ИНС.

2.4 Функция вознаграждения

Все безмодельные алгоритмы RL нацелены на максимизацию ожидаемого вознаграждения, поскольку максимизация вознаграждения лежит в основе цели RL. Например, агент получает награду +1, если эпизод завершился достижением заданной цели и -1, если по окончании эпизода цель не была достигнута. Именно токую схему использовали DeepMind, чтобы обучить AlphaZero играть в шахматы, и победили нейросеть StockFish, которая обучалась на партиях 10000 лучших игроков мира. Однако, такая схема вознаграждений требует продолжительного процесса обучения, чтобы политика сходилось к оптимальной, так как агент получает награду только в конце эпизода, и неизвестно, в каком моменте было принято неправильное решение. Решение этой проблемы – вознаградить агента во время эпизода за то, что он приблизился к поставленной перед ним цели.

В процессе обучения агент получает следующие вознаграждения:

- –0,1 – при каждом выполненном шаге; это вознаграждение предназначено, чтобы агент научился как можно быстрее достигать цели, так как, чем меньше шагов он осуществляет, тем меньше отрицательного вознаграждения получает;
- значение, вычисленное в соответствии с уравнением (1) – при приближении к противнику сзади агент получает положительное вознаграждение; за приближение к препятствиям агент получает отрицательное вознаграждение;
- –30 – при столкновении с препятствием или выходе за пределы среды;
- –100 – при поражении агента противником;
- +100 – при поражении противника агентом (уравнение 2).

$$r_i = -0.1 + \alpha.\varphi.dist(goal) - \beta.\frac{1}{dist(obstacles)}, \quad (1)$$

$$r_1 = \begin{cases} 100, \text{приуспешномзавершении} \\ -30, \text{пристолкновении} \\ -100, \text{припоражении} \end{cases}, \quad (2)$$

где r_i – вознаграждение за каждый шаг i; α – весовой коэффициент влияния расстояния до противника (>1); φ– угол между векторами ориентации союзника и противника (радиан); $dist(goal)$– расстояние до противника; β– весовой коэффициент влияния расстояния до препятствий (>1); $dist(obstacles)$ – расстояние до препятствий.

Итоговое вознаграждение за игру рассчитывается следующим образом:

$$r = \sum_{i=1}^{T} r_i + r_1 , \quad (3)$$

где T – количество шагов в эпизоде.

3 Моделирование

В связи со сложностью поставленной задачи было принято решение начать с самого простого варианта (один РТК против одного РТК в среде с препятствиям) и поэтапно усложнять задачу (группа из двух РТК против группой из двух и т.д.)

Использовались следующие гиперпараметры: количество шагов для обучения = 1e6; количество шагов, состовляющих выборку обучения =2048 ; скорость обучения = 0.0001; коэффициент дисконтирования = 0.99; размер выборки для обучения = 64; количество эпох = 10

На рисунке 3 изображено среднее вознаграждение (ось y), полученное агентами с разными алгоритмами в процессе обучения за милион шагов (ось x). Как можно видеть, все агенты обучились, так как значения вознаграждений стремятся к асимптоте, алгоритм PPO быстрее всех сходится.

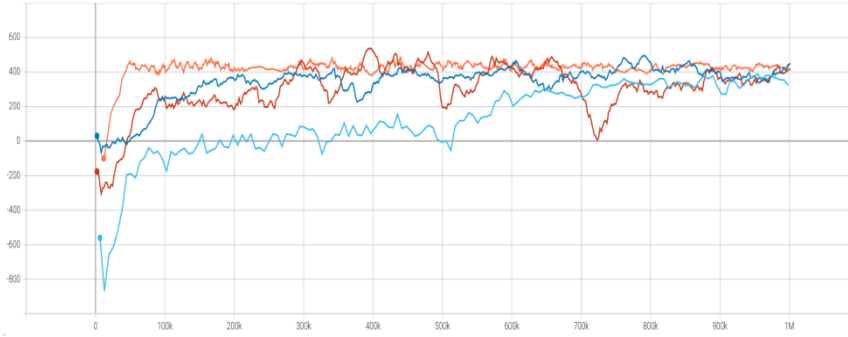

Рис. 3. Усредненное значение вознаграждения за эпизод, оранжевый график – PPO, синий график – A2C, красный график – DDPG, голубой график – SAC.

На рисунке 3 представлено среднее количество действий (ось y), совершаемых в среде за один эпизод в процессе обучения за милион

шагов (ось x).

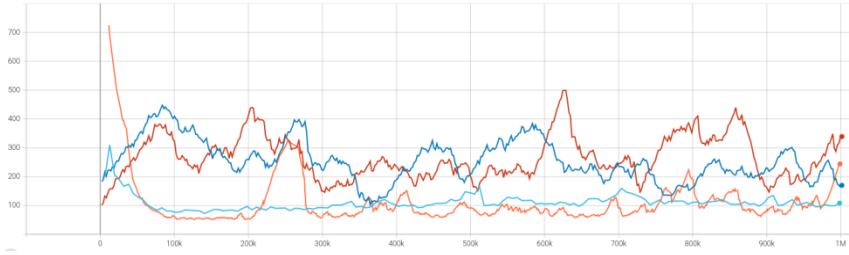

Рис. 4. Среднее количество действий за эпизод, оранжевый график – PPO, синий график – A2C, красный график – DDPG, голубой график – SAC.

После обучения агента, на тестировании был достигнут результат 60% успеха в поражение противника в среде с препятствиями.

Анализ поведения агента показал, что в большинстве случаев агент проигрывает из-за столкновения с препятствием или выхода за пределы среды. В связи с этим можно сделать следующие выводы: нейронная сеть переобучается на некоторых ситуациях, либо её обобщающей способности недостаточно для определения политики агента во всех разнообразных ситуациях; необходима дальнейшая настройка системы вознаграждений и определение оптимальных значений гиперпараметров.

Заключение

В статье рассмотрен процесс разработки системы управления группой РТК на основе обучения с подкреплением, который является весьма трудоёмкой задачей.

Основные преимущества систем управления на основе RL заключаются в том, что они могут быть свободными от моделей т.е, не требуют знания динамической модели управляемого объекта и явного программирования всеобъемлющих правил, могут работать с произвольно абстрагированными управляющими входами, многомерными пространствами состояний.

На основе анализа полученных результатов были определены направления дальнейших исследований для улучшения разработанной системы, а именно, - для повышения процента успеха в достижении заданной цели. Определена необходимость экспериментов со структурой нейронной сети; различными функциями вознаграждений и гиперпараметрами алгоритмов обучения.

Список литературы

[Цариченко и др., 2019] Цариченко С.Г., Постников Е.В., Пантелеев М.Г. Концепция виртуального полигона нового поколения для отработки программного обеспечения автономных робототехнических комплексов на основе мультиагентных технологий // Робототехника и техническая кибернетика. 2019.

[Верба и др., 2016] Верба В.С., Татарский Б.Г. Комплексы с беспилотными летательными аппаратами // монография. 2016.

[DARPA 2016] DARPA’s OFFensive Swarm-Enabled Tactics (OFFSET) program // URL: Swarm Tactics Tools & Technologies for DARPA's OFFSET Program

[OpenAI 2019] OpenAI, Dota 2 with Large Scale Deep Reinforcement Learning. 2019.

[Huang и др., 2020] Yu Huang, Yue Chen. Autonomous Driving with Deep Learning: A Survey of State-of-Art Technologies // arxive.org. 2020.

[Sutton и др. 2018] Sutton R.S., Barto A.G. Reinforcement Learning: An Introduction // MIT press. 2018.

[Andrychowicz и др., 2017] Andrychowicz M. et al. Hindsight experience replay // arXiv. 2017.

[Dayana и др., 2008] Dayana P., Niv Y. — Reinforcement learning: The Good, The Bad and The Ugly, 2008

[Mnih и др., 2016] Mnih, Volodymyr, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning // International conference on machine learning. 2016.

[Schulman и др., 2017] Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O., Proximal Policy Optimization Algorithms // ArXiv. 2017.

[Lillicrap и др., 2019] Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., et al. Continuous Control with Deep Reinforcement Learning // ArXiv. 2019.

[Haarnoja и др., 2017] Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S., Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor // ArXiv. 2017.

Расширенные тезисы