NGHIÊN CỨU THUẬT TOÁN DEEP REINFORCEMENT LEARNING ĐIỀU KHIỂN MOBILE ROBOT ĐIỀU HƯỚNG TỰ ĐỘNG

STUDY ON DEEP REINFORCEMENT LEARNING ALGORITHM FOR AUTONOMOUS NAVIGATION OF MOBILE ROBOT

Lưu Thanh Tùng[1, 2], Nguyễn Hoàng Long[1, 2]
[1]Khoa Cơ khí, Trường Đại học Bách Khoa, Đại học Quốc gia Thành phố Hồ Chí Minh (HCMUT)
[2]Đại học Quốc gia Thành phố Hồ Chí Minh (VNU-HCM)

TÓM TẮT

Hoạch định chuyển động (motion planning) là yếu tố quan trọng để cho phép các robot di động hoạt động một cách tự động. Khi các trường hợp ứng dụng của robot trở nên phức tạp và khó đoán hơn, các bộ hoạch định chuyển động phân cấp truyền thống gặp phải những thách thức đáng kể. Tuy nhiên, với những tiến bộ trong học sâu, các bộ hoạch định chuyển động dựa trên học tăng cường sâu (DRL) đã trở thành một điểm nóng nghiên cứu nhờ vào nhiều tính năng ưu việt của chúng. Trong bài báo này, chúng tôi đề xuất một phương pháp học tăng cường sâu, tận dụng các dữ liệu laser thô và sử dụng thư viện PIC4rl-gym để xây dựng môi trường mô phỏng trên Gazebo. Các thí nghiệm cho thấy phương pháp đề xuất đã cho thấy khả năng điều hướng tự động hoàn toàn của mobile robot, và kết quả huấn luyện đã xác nhận tính hiệu quả của thuật toán.

Từ khóa: *Hoạch định chuyển động; Môi trường mô phỏng; Học tăng cường sâu.*

ABSTRACT

Motion planning is essential for enabling mobile robots to operate autonomously. As robot application scenarios become more complex and unpredictable, traditional hierarchical motion planners face significant challenges. However, with advancements in deep learning, motion planners based on deep reinforcement learning (DRL) have gained attention as a research hotspot due to their numerous advantageous features. In this paper, we propose a deep reinforcement learning approach, leveraging raw laser scans and PIC4rl-gym library to build simulation environment on Gazebo. The experiments show that the proposed method has enabled fully autonomous navigation of the agent, and the training results verify the effectiveness of the algorithm.

Keywords: *Motion planning; Environment Simulation; Deep reinforcement learning.*

1. TỔNG QUAN

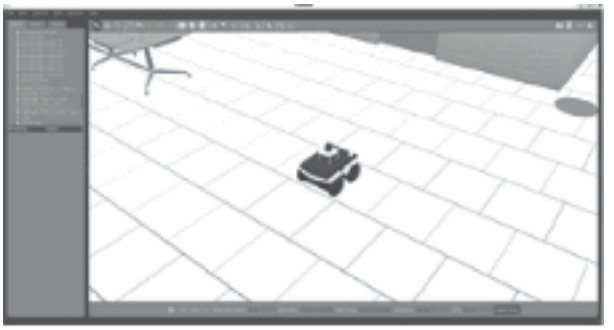
Kỹ thuật hoạch định chuyển động (MP) là một trong những mô-đun quan trọng nhất giúp cho robot di động có khả năng tự hành. Chức năng cụ thể của bộ hoạch định chuyển động là tích hợp thông tin trạng thái cục bộ hoặc toàn cục của hệ thống robot và đưa ra các quyết định lập kế hoạch tối ưu hoặc gần tối ưu trong các môi trường khác nhau, từ đó lập ra kế hoạch di chuyển cho robot. Mô-đun MP tiêu chuẩn bao gồm bộ lập kế hoạch chuyển động toàn cục (global planner) và bộ lập kế hoạch chuyển động cục bộ (local planner). Bộ lập kế hoạch chuyển động toàn cục chịu trách nhiệm tạo ra các quỹ đạo di chuyển, khả thi về động học, an toàn và có thể thực thi dựa trên thông tin bản đồ có sẵn. Bộ lập kế hoạch chuyển động cục bộ chịu trách nhiệm giúp mobile robot đưa ra các quyết định chuyển động theo thời gian thực trong các môi trường động cục bộ (ví dụ: môi trường có sự tham gia của người đi bộ, môi trường ngoài trời, v.v.).

Cùng với sự cải thiện đáng kể về sức mạnh tính toán và khả năng lưu trữ của phần cứng hệ thống, công nghệ học sâu (Deep learning) đã được sử dụng rộng rãi trong học tăng cường. Trong những nghiên cứu gần đây, thuật toán học tăng cường sâu Deep Deterministic Policy Gradient (DDPG) [1] trở nên phổ biến vì có khả năng đưa ra quyết định hành động trong không gian liên tục, trong đó [2] đã ứng dụng thành công thuật toán với đầu vào là các dữ liệu laser thô và đầu ra là một vector hành động bao gồm vận tốc dài và vận tốc góc. Không chỉ với những dữ liệu đầu vào là laser, H. Surmann et al. [3] đã sử dụng kỹ thuật tổng hợp cảm biến (sensor fusion) để kết hợp thông tin thu về từ cảm biến Lidar và cả camera để tạo ra một không gian quan sát trực quan hơn cho robot. M. Everett et al. [4], [5] đã phát triển thuật toán DRL dành cho việc né tránh vật cản là con người. Vấn đề này thực sự gây nhiều khó khăn hơn so với việc né vật cản tĩnh như là các đồ vật trong văn phòng vì bên cạnh việc né tránh, chúng ta cần quan tâm đến sự ảnh hưởng của việc né đó đối với người đi bộ. Các tác giả đã sử dụng camera để phân biệt người đi bộ kết hợp với Lidar làm không gian quan sát cho thuật toán và phát triển phần thưởng (reward) là các quy luật xã hội (social norm) trong việc bắt chước hành vi né tránh của con người. Tuy nhiên, các thuật toán trên đòi hỏi sự phức tạp về mặt thiết kế mô hình, khối lượng tính toán lớn và thời gian huấn luyện lâu cùng với sự kết hợp của đa dạng các loại cảm biến, sẽ là không khả thi để áp dụng cho những trường hợp đơn giản, trong điều kiện mà mọi yêu cầu trên đều bị hạn chế. Trong nghiên cứu này, chúng tôi đưa ra một khung mô hình nghiên cứu đơn giản, chỉ sử dụng cảm biến Lidar để thu nhận dữ liệu đầu vào cho thuật toán DDPG, chúng tôi cũng tận dụng khả năng mô-đun hóa của thư viện PIC4rl-gym [6], giúp cho việc huấn luyện, kiểm tra và đánh giá mô hình một cách dễ dàng và trực quan hơn.

2. PHƯƠNG PHÁP

2.1. Mobile robot mô phỏng

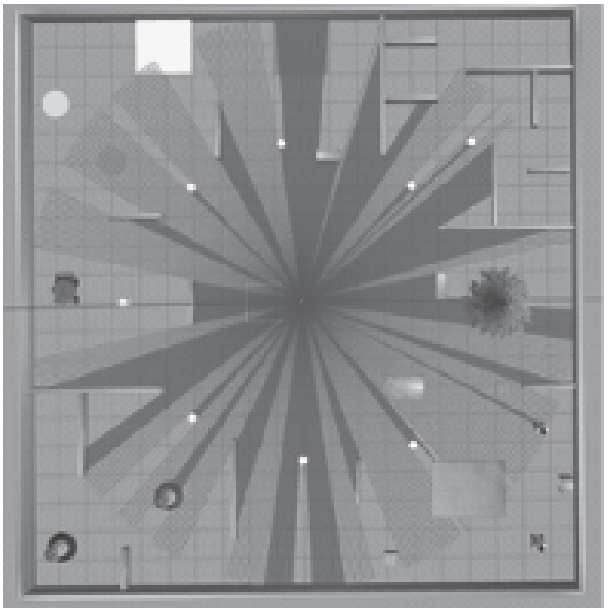
Để thực hiện công việc phát triển thuật toán, ta cần một mô hình robot đơn giản nhưng đầy đủ để có thể chạy được trong mô phỏng. Nền tảng robot di động mặt đất Jackal (UGV) là một giải pháp đa năng và mạnh mẽ cho nhiều ứng dụng robot khác nhau. Được thiết kế bởi Clearpath Robotics, Jackal thể hiện tính linh hoạt, độ tin cậy và khả năng mở rộng, làm cho nó phù hợp với nhiều dự án nghiên cứu, giáo dục và công nghiệp. Vì lý do đó, chúng tôi sử dụng mô hình Jackal làm robot đại diện cho quá trình mô phỏng. ☞



Hình 1. Mô hình Jackal mobile robot mô phỏng

2.2. Môi trường mô phỏng

Môi trường mô phỏng được xây dựng dựa trên phần mềm Gazebo (gọi là world). Các world là một file XML chứa các phần tử mô tả tính chất vật lý, các vật thể và sự chuyển động trong môi trường mô phỏng. Môi trường mô phỏng cần phải đáp ứng độ chính xác so với môi trường thực tế, vì vậy chúng tôi chọn môi trường indoor – môi trường mô phỏng văn phòng phổ biến gồm có các vật dụng và kích thước phù hợp để cảm biến có thể quét được nhiều thông tin nhất có thể.



Hình 2. Độ bao phủ của cảm biến Lidar trong môi trường mô phỏng văn phòng thực tế

2.3. Thuật toán học tăng cường sâu

Thuật toán học tăng cường tiêu chuẩn có thể được biểu diễn bằng một Quy trình quyết định Markov (Markov Decision Process – MDP). Một MDP được mô tả bởi 5 phần tử như sau: (S,A,P,R,γ). Trong đó:

S: Tập hợp các trạng thái (State) mà môi trường có thể tồn tại.

A: Tập hợp các hành động (Action) mà tác nhân (agent) có thể thực hiện trong mỗi trạng thái.

P: Hàm xác suất chuyển đổi trạng thái (Transition Probability), biểu diễn khả năng chuyển từ trạng thái hiện tại s_t sang trạng thái mới s_{t+1} khi thực hiện một hành động nhất định a_t.

R: Hàm thưởng (Reward), biểu diễn mức độ "tốt" hay "xấu" của việc thực hiện hành động a_t tại trạng thái s_t.

γ: Hệ số lạm phát (Discount Factor), dùng để đánh giá tầm quan trọng của các phần thưởng tương lai so với phần thưởng tức thời.

Khi một agent bắt đầu tương tác với môi trường, agent sẽ ở trong một trạng thái ban đầu s_0 được chọn từ phân phối xác suất cố định $p(s_0)$. Sau đó, theo chu kỳ, agent sẽ lựa chọn một hành động $a \in A$ tại trạng thái hiện tại s_t để di chuyển sang trạng thái mới s_{t+1} với xác suất chuyển đổi $P(s_{t+1} | s_t, a_t)$. Đồng thời, tác nhân nhận được phần thưởng $r_t = R(s_t, a_t)$.

Bộ điều khiển (hay policy) của thuật toán học tăng cường $\pi(a_t | s_t)$ là một hàm số có đầu vào là trạng thái quan sát được từ môi trường và đầu ra là một hành động. Bộ điều

ISSN 2615 - 9910 (bản in), ISSN 2815 - 5505 (online)
TẠP CHÍ CƠ KHÍ VIỆT NAM, Số 318, tháng 8 năm 2024
cokhivietnam.vn / tapchicokhi.com.vn

khiển của agent được huấn luyện để tối đa hóa tổng kỳ vọng về phần thưởng, hay nói cách khác, mục tiêu của ta là đạt được policy tối ưu π_θ^* với tham số là θ:

$$\pi_\theta^* = \underset{\pi}{\operatorname{argmax}}\, \mathbb{E}_{\tau\sim\pi} \sum_{t=0}^{T} \gamma^t r_t \quad (1)$$

Thuật toán DDPG giải quyết bài toán trên bằng kiến trúc Actor-Critic, trong đó Actor là một mạng với đầu vào là một state và đầu ra là một hành động; và Critic là một mạng với đầu vào là một cặp trạng thái và hành động và đầu ra là một giá trị đánh giá hành động đó là "tốt" hay "xấu" (Q-value). Chi tiết về kiến trúc của thuật toán DDPG được thể hiện trong Hình 4.

Để tối ưu tham số của Actor, chúng ta dùng công thức Policy Gradient:

$$\nabla J = \frac{1}{N}\sum_i \nabla Q \nabla \mu \quad (2)$$

Để tối ưu tham số của Critic, chúng ta tính hàm Loss:

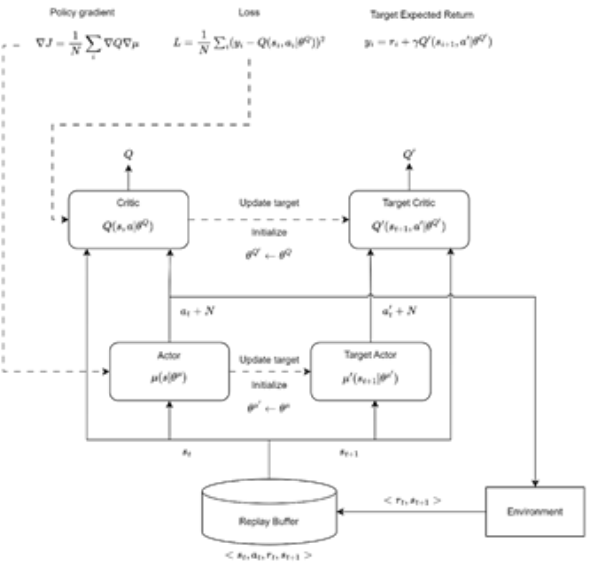
$$L = \frac{1}{N}\sum_i \left(y_i - Q(s_i, a_i|\theta^Q)\right)^2 \quad (3)$$

Trong đó, y_i là giá trị phần thưởng kỳ vọng mục tiêu và được tính bằng công thức:

$$y_i = r_i + \gamma Q'\left(s_{i+1}, a'\middle|\theta^{Q'}\right) \quad (4)$$

Hàm phần thưởng được thiết kế như sau: $r = r_d + r_{yaw} + r_{goal} + r_{col}$, trong đó:

$$\begin{cases} r_d = c_1(d_{t-1} - d_t) \\ r_{yaw} = c_2\left(1 - 2\sqrt{\left|\frac{\phi_t}{\pi}\right|}\right) \\ r_{goal} = 1000 \; if \; d_t \le d_{goal\ tolerance}, else\ 0 \\ r_{col} = -200 \; if \; d_t \le d_{collision\ tolerance}, else\ 0 \end{cases} \quad (5)$$

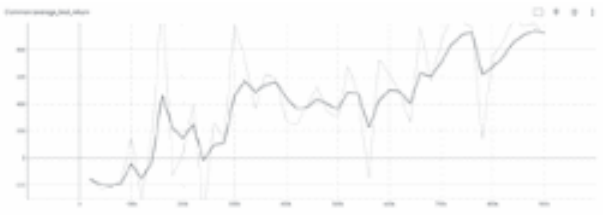


Hình 3. Kiến trúc của thuật toán DDPG

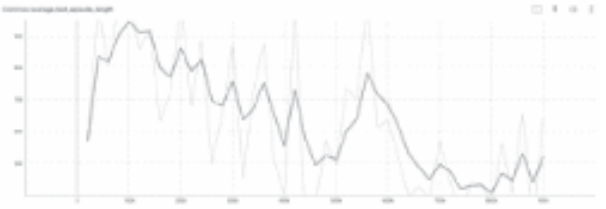
3. THÍ NGHIỆM

3.1. Huấn luyện thuật toán

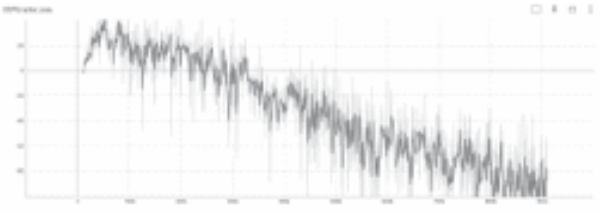
Phần cứng tính toán sử dụng để huấn luyện là GPU Nvidia 1060Ti Max-Q. Thuật toán được huấn luyện trong tổng cộng là 900 nghìn bước, thời gian huấn luyện xấp xỉ 12 giờ. Không gian trạng thái là một vector (38,) bao gồm 36 giá trị của tia laser và 2 giá trị biểu thị vị trí và hướng tương đối của mobile robot so với đích. Không gian hành động là một vector (2,) bao gồm vận tốc dài và vận tốc góc của robot. Đối với mạng Actor và Critic, chúng tôi chọn một mạng nơ ron gồm 2 lớp ẩn có số chiều như sau (256,256). Thuật toán được tối ưu bằng thuật toán Adam. Kết quả huấn luyện được thể hiện trên các Hình 4, Hình 5 và Hình 6: ☞



Hình 4. Phần thưởng nhận được trung bình trong pha kiểm tra



Hình 5. Độ dài của mỗi episode trong toàn bộ quá trình huấn luyện



Hình 6. Mất mát của mạng Actor trong bộ điều khiển

4. KẾT QUẢ VÀ THẢO LUẬN

Tổng phần thưởng tích lũy được xem là yếu tố đánh giá quan trọng nhất, thông số này phản ánh khả năng đạt được mục tiêu đã đề ra của thuật toán. Giá trị càng cao chứng tỏ trong một vòng lặp mô phỏng, thuật toán của ta đã điều khiển robot đi đúng hướng. Độ dài của một episode giảm cũng là một yếu tố đánh giá, nghĩa là theo thời gian robot sẽ thông thạo đường đi, khiến cho thời gian hoàn thành giảm. Hàm mất mát của Actor giảm dần cũng là một dấu hiệu cho thấy mô hình đã được huấn luyện tốt và đang hội tụ đến điểm tối ưu.

Bảng 1. Bảng thông số đánh giá độ hiệu quả của thuật toán sau 3 lần chạy

Tình trạng	Thời gian	Góc tích lũy trung bình	Khoảng cách trung bình đến vật cản	Khoảng cách	Đường đi	Tỉ số khoảng cách trên đường đi
Success	28.13s	0.34 rad	4.54	8.96	10.17	0.88
Success	1s	0.7 rad	6.41	0.77	0.45	1.71
Success	16.89s	0.33 rad	5.46	7.40	7.74	0.96

ISSN 2615 - 9910 (bản in), ISSN 2815 - 5505 (online)
TẠP CHÍ CƠ KHÍ VIỆT NAM, Số 318, tháng 8 năm 2024
cokhivietnam.vn / tapchicokhi.com.vn

Từ bảng đánh giá trên, ta có thể thấy rằng robot đều hoàn thành việc đi đến điểm đích trong khoảng thời gian chấp nhận được. Các giá trị góc tích lũy trung bình thấp nghĩa là trong quá trình di chuyển robot luôn giữ một hướng cố định. Khoảng cách trung bình đến vật cản thấp nghĩa là robot đã luôn duy trì khoảng cách an toàn đến vật cản. Cuối cùng, thông số quan trọng liên quan đến hoạch định đường đi cho robot là tỉ số khoảng cách trên đường đi, trong đó khoảng cách chính là khoảng cách Euclid từ vị trí ban đầu của robot đến vị trí đích. Con số này càng gần 1 nghĩa là quãng đường mà robot đã đi gần trùng với khoảng cách ban đầu giữa chúng.

5. KẾT LUẬN

Thuật toán huấn luyện đã điều khiển robot di chuyển đến điểm đích được cho trước với tỉ lệ bị va chạm với vật cản tĩnh gần như bằng không. Kết quả thử nghiệm trên mô phỏng cho thấy robot có thể hoàn thành nhiệm vụ di chuyển đến mục tiêu trong thời gian ngắn và an toàn. Với độ phức tạp thấp và thời gian huấn luyện không cao, đây chính là nền móng cho việc triển khai thuật toán vào thiết bị phần cứng dễ dàng và nhanh gọn hơn.

Lời cảm ơn:

Nghiên cứu này được tài trợ bởi Trường Đại học Bách khoa, Đại học Quốc gia Thành phố Hồ Chí Minh trong khuôn khổ đề tài mã số SVOISP-2023-CK-12. Chúng tôi xin cảm ơn Trường Đại học Bách khoa, ĐHQG-HCM đã hỗ trợ cho nghiên cứu này. ❖

Ngày nhận bài: **15/5/2024**
Ngày phản biện: **18/6/2024**

Tài liệu tham khảo:

[1]. Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D. & Wierstra, D. (2016). "*Continuous control with deep reinforcement learning*". In Y. Bengio & Y. LeCun (eds.), ICLR.

[2]. L. Tai, G. Paolo and M. Liu, "*Virtual-to-real deep reinforcement learning: Continuous control of mobile robots for mapless navigation*". 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).

[3]. Surmann, H., Jestel, C., Marchel, R., Musberg, F., Elhadj, H., & Ardani, M. (2020). "*Deep Reinforcement learning for real autonomous mobile robot navigation in indoor environments*". ArXiv, abs/2005.13857.

[4]. Y. F. Chen, M. Everett, M. Liu and J. P. How, "*Socially aware motion planning with deep reinforcement learning*". 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).

[5]. M. Everett, Y. F. Chen and J. P. How, "*Motion Planning Among Dynamic, Decision-Making Agents with Deep Reinforcement Learning*". 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).

[6]. M. Martini, A. Eirale, S. Cerrato and M. Chiaberge, "*PIC4rl-gym: a ROS2 Modular Framework for Robots Autonomous Navigation with Deep Reinforcement Learning*". 2023 3rd International Conference on Computer, Control and Robotics (ICCCR).