NGHIÊN CỨU PHƯƠNG PHÁP HỌC TĂNG CƯỜNG ỨNG DỤNG CHO BÀI TOÁN ĐỊNH VỊ ROBOT DI ĐỘNG LÀM VIỆC TRONG MÔI TRƯỜNG KHÔNG XÁC ĐỊNH

APPLICATION OF THE REINFORCEMENT LEARNING METHOD IN MOBILE ROBOT NAVIGATION WORKING IN AN UNKNOWN ENVIRONMENT

Nguyễn Hồng Sơn[1,*], Phan Thái Học[2], Nguyễn Anh Tú[3]

TÓM TẮT

Học tăng cường (RL) là một kỹ thuật của học máy với ưu thế tập trung vào việc đào tạo các mô hình học máy đưa ra một chuỗi các quyết định sao cho tác tử học đạt được mục tiêu trong một môi trường không chắc chắn, có thể là phức tạp. Những công trình nghiên cứu gần đây, phương pháp sử dụng học tăng cường đã được áp dụng nhằm nâng cao độ chính xác và linh hoạt cho robot di động giải quyết các bài toán về định vị. Mục đích của phương pháp chính là đào tạo cho robot khả năng tự học tự thích nghi với môi trường làm việc có thể là chưa được xác định sẵn. Bài báo sẽ đi sâu vào nghiên cứu vấn đề ứng dụng học tăng cường với thuật toán Q-learning cho quá trình định vị robot trong một môi trường chưa xác định.

Từ khoá: *Học tăng cường, định vị Robot di động, thuật toán Q-learning, môi trường không xác định.*

ABSTRACT

Reinforcement learning (RL) is a technique of machine learning with the preponderance of focusing on training machine learning models to make a sequence of decisions so that the learning agent achieves its goal in an uncertain or complicated environment. In recent research, the method of using reinforcement learning has been applied to improve the accuracy and flexibility of mobile robots to solve positioning problems. The purpose of the main method is to train the robot to learn to adapt itself to a working environment that may not be predefined. This paper will dive into the problem of applying reinforcement learning with the Q-learning algorithm to the robot positioning process in an unknown environment.

Keywords: *Reinforcement Learning, mobile robot Navigatio, Q-learning, unknown environment.*

[1]Lớp Cơ điện tử 03 - K14, Khoa Cơ khí, Trường Đại học Công nghiệp Hà Nội
[2]Lớp Cơ điện tử 04 - K14, Khoa Cơ khí, Trường Đại học Công nghiệp Hà Nội
[3]Khoa Cơ khí, Trường Đại học Công nghiệp Hà Nội
[*]Email: son94227@gmail.com

1. GIỚI THIỆU

Bước vào thời đại công nghiệp 4.0 với sự phát triển mạnh mẽ của khoa học công nghệ cùng khả năng tích hợp với các thiết bị ngoại vi, các loại cảm biến như thị giác, xúc giác, cảm biến lực/mô men làm tăng khả năng thích ứng với môi trường thay đổi của robot. Đồng thời việc ứng dụng các giải pháp điều khiển thông minh giúp robot có khả năng tự học và tự giải quyết vấn đề, tạo tiền đề cho robot di động ngày càng được ứng dụng rộng rãi trong nhiều lĩnh vực của đời sống như: Robot trong y tế, chăm sóc sức khỏe, nông nghiệp, đóng tàu, xây dựng, an ninh quốc phòng và trong các ứng dụng dân sinh. Quá trình tự động hóa và hiện đại hóa quy trình sản xuất trong các nhà máy công nghiệp được thực hiện một phần nhờ vào sự cải tiến các hoạt động vận chuyển trong các dây chuyền sản xuất. Sự ra đời của robot di động là một trong những bước chuyển hoá quan trọng trong quy trình tự động hóa hoạt động vận chuyển trong các dây chuyền sản xuất, lắp ráp cũng như hệ thống lưu kho. Bên cạnh đó, các loại robot di động phục vụ hoạt động logistic trong các bệnh viện, nhà ga và các trung tâm vận chuyển hàng hóa. Dựa vào các thông tin này, hệ thống điều khiển sẽ ra các quyết định phù hợp để robot có thể chuyển động theo quỹ đạo thiết kế với vận tốc tương ứng.

Trong robot di động gồm có bốn bài toán chính cần thực hiện, đầu tiên phải kể đến định vị, bài toán thứ hai là thiết kế quỹ đạo chuyển động, thứ ba là bài toán tránh vật cản và cuối cùng là điều khiển chuyển động. Định vị là xác định về hướng và vị trí của robot so với môi trường hoạt động để quá trình di chuyển đạt ổn định và chính xác, do đó định vị đóng vai trò vô cùng quan trọng, làm tiền đề để giải quyết các bài toán tiếp theo, sự thành công của một hệ robot hoạt động độc lập. Nếu robot không xác định được vị trí trong môi trường hoạt động sẽ khó khăn trong việc thực hiện nhiệm vụ tiếp theo. Trong nhiệm vụ nhận biết vị trí, các cảm biến được sử dụng để thu thập và sản xuất thông tin từ môi trường, nơi robot hoạt động. Có một loạt các cảm biến có thể được chọn để điều hướng tự động tùy thuộc vào nhiệm vụ của robot di động, nói chung có thể được phân thành hai nhóm là đinh vị tuyệt đối và định vị tương đối.

Với sự tiến bộ của công nghệ, các giải pháp thông minh ứng dụng cho lĩnh vực robotics nói chung và robot di động nói riêng đã phần nào khắc phục được những nhược điểm của các phương pháp xây dựng robot truyền thống. Điều quan trọng là phải hiểu bản chất của việc học để đạt được mục tiêu của thông minh của robot. Mặc dù có rất nhiều thuật toán được phát triển dưới dạng phương pháp học có giám sát và không giám sát trong lĩnh vực học máy, nhưng ý tưởng cơ bản của việc sử dụng học tăng cường (RL) là học từ tương tác. Để có được khả năng tương tác, robot phải thu thập dữ liệu từ các cảm biến khác nhau được gắn trên robot, bao gồm cả tia hồng ngoại. Năm 1957, [3] giới thiệu lập trình động (dynamic programming), dẫn đến điều khiển tối ưu và sau đó là quy trình quyết định Markov (MDP). Sự khác biệt về thời gian đã mang lại một khía cạnh mới cho RL, và vào năm 1989 Q-learning được khám phá bởi [4] là một bước đột phá quan trọng trong lĩnh vực AI. Cuối cùng, thuật toán được Sara [5] giới thiệu vào năm 1994 với tên gọi 'modifined Q-learning'. Có những nghiên cứu khác nhằm nâng cao kỹ thuật RL để có kết quả học tập nhanh hơn và chính xác hơn [2, 6 - 8]. Điểm mạnh của phương pháp học tăng cường là giúp cho mô hình ở đây là robot có tính tự học và tích luỹ kinh nghiệm sau những lần thử dựa vào quy trình thu thập và tối ưu các phần thưởng dựa vào tương tác với môi trường; cùng thuật toán Q-learning có chức năng cập nhật giá trị sau những tương tác giữa robot và môi trường nhằm tính toán đưa ra các hành động tối ưu cho robot di chuyển tới điểm mục tiêu trong môi trường. Từ những đặc điểm kể trên, nhóm nghiên cứu quyết định lựa chọn giải pháp học tăng cường giải quyết bài toán định vị của robot trong môi trường không xác định.

2. CƠ SỞ LÝ THUYẾT PHƯƠNG PHÁP HỌC TĂNG CƯỜNG VÀ THUẬT TOÁN Q-LEARNING

❖ Cơ sở lý thuyết phương pháp học tăng cường

Giải pháp học tăng cường tập trung nhiều hơn vào việc học hướng tới mục tiêu từ sự tương tác hơn là các giải pháp khác của kỹ thuật học máy. Thực thể học tập (the learning entity) không được cho biết trước những hành động phải thực hiện, nhưng thay vào đó phải tự khám phá ra hành động nào tạo ra phần thưởng lớn nhất, hay chính là mục tiêu cuối cùng của nó, bằng cách kiểm tra các hành động này thông qua phương pháp "thử và sai".

Các thuật ngữ thường dùng trong học tăng cường bao gồm 7 thuật ngữ:

▪ Tác nhân (Agent): đại diện cho "giải pháp", là một chương trình máy tính với một vai trò duy nhất là đưa ra quyết định để giải quyết các vấn đề ra quyết định, tác động phức tạp dưới sự không chắc chắn.

▪ Môi trường (Environment): đó là đại diện của một trường "vấn đề", là mọi thứ xảy ra sau tác động của tác nhân.

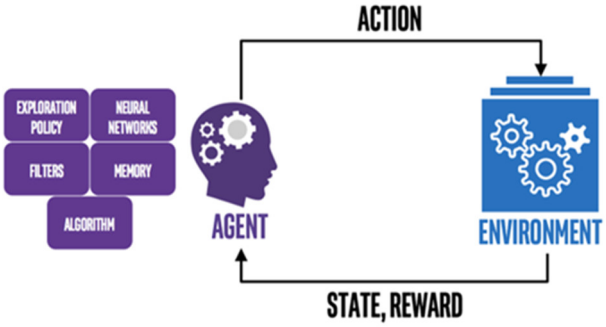
▪ Hành động (Actions): tập hợp các phương thức mà tác nhân tác động đến môi trường.

▪ Trạng thái (State): trạng thái của tác nhân được trả lại qua tác động đến môi trường.

▪ Phần thưởng (Reward): phần thưởng tương ứng với mỗi trạng thái của tác nhân nhận được trong môi trường.

▪ Tập (Episode): một chu kỳ bao gồm các tương tác giữa tác nhân và môi trường từ thời điểm bắt đầu đến kết thúc.

▪ Chính sách (Policy): chính sách hay là một luật thiết lập cho tác nhân hoàn thành được mục tiêu đặt ra trong môi trường.



Hình 1. Sơ đồ quan hệ giữa Tác nhân và Môi trường trong kỹ thuật học tăng cường [10]

Trong đó, tác nhân và môi trường là hai thành phần cốt lõi của một hệ thống học tăng cường. Hai thành phần cốt lõi này tương tác liên tục theo cách mà tác nhân cố gắng tác động đến môi trường thông qua các hành động (hay quyết định) và môi trường phản ứng lại với các hành động của tác nhân.

Môi trường thường chứa đựng một nhiệm vụ được xác định rõ ràng và có thể cung cấp cho tác nhân một tín hiệu khen thưởng (phần thưởng) như một câu trả lời trực tiếp cho các hành động của tác nhân. Phần thưởng này là phản hồi về mức độ hiệu quả của hành động cuối cùng của tác nhân trong việc nó góp phần đạt được nhiệm vụ. Phần thưởng này được cung cấp bởi môi trường.

❖ Thuật toán Q-learning

Q-learning là một thuật toán học tăng cường không có mô hình (Model-free). Nó học tập dựa trên các giá trị (values-based). Các thuật toán dựa trên giá trị cập nhật hàm giá trị dựa trên một phương trình (đặc biệt là phương trình Bellman). Trong khi loại còn lại, dựa trên chính sách (policy-based) ước tính hàm giá trị với một chính sách tham lam có được từ lần cải tiến chính sách cuối cùng [9].

Phương trình Bell-man:

$$Q^*(s,a) = E_{s'}[r + \gamma \max_{a'} Q(s',a')^* \mid s,a] \quad (1)$$

Q learning là một thuật toán off-policy: Có nghĩa là nó học được giá trị của chính sách (policy) tối ưu một cách độc lập với các hành động của chủ thể (agent). Mặt khác, on-policy learner tìm hiểu giá trị của chính sách đang được thực hiện bởi chủ thể, bao gồm các bước thăm dò và họ sẽ tìm ra một chính sách tối ưu, có tính đến việc khám phá (exploration) vốn có trong chính sách.

$$Q_{st,at} = Q_{st,at} + \alpha * [r_t + \gamma * \max Q(st+1,a) - Q_{st,at}] \quad (2)$$

Trong đó:

· $Q^*(s, a)$ là giá trị kỳ vọng (phần thưởng của chiết khấu tích lũy) trong việc thực hiện hành động (action) *a* ở trạng thái (state) *s* và sau đó tuân theo chính sách tối ưu (optimal policy).

· α là tốc độ học của thuật toán thể hiện sự hội tụ của mô hình học;

· γ là hằng số chiết khấu xác định mức độ quan trọng được trao cho phần thưởng trước mắt và phần thưởng trong tương lai;

· s_t và a_t là giá trị trạng thái và hành động thu được tại thời điểm t.

Hành động từ mỗi trạng thái thu được của thuật toán Q-learning được xác định bởi Quy trình đưa quyết định Markov (MDP) nhằm đưa ra hành động phù hợp trong tương lai nhờ phần thưởng hiện tại.

Với bộ giá trị (S, A, P, R, γ), với:

· S là một tập hợp các trạng thái,

· A là tập hợp các hành động mà tác nhân có thể chọn để thực hiện,

· P là Ma trận xác suất chuyển đổi,

· R là Phần thưởng được tích lũy bởi các hành động của tác nhân,

· γ là hệ số chiết khấu.

- Ma trận xác suất chuyển đổi:

$$P^a_{ss'} = P[S_{t+1} = s' \mid S_t = s, A_t = a] \quad (3)$$

- Hàm phần thưởng:

$$R^a_s = E[R_{t+1} \mid S_t = s, A_t = a] \quad (4)$$

❖ Lưu đồ thuật toán Q-learning:

Thuật toán: Q-learning

Đầu vào:

Tập trạng thái $S = \{1, \ldots, s_n\}$;

Tập hành động $A = \{1, \ldots, a_n\}$;

Hàm phần thưởng: $S \times A \rightarrow []$

Khởi tạo các siêu tham số của thuật toán: $\alpha, \gamma \in [0,1]$;

Phương thức:

Khởi tạo *Q*: $S \times A \rightarrow$ [] ngẫu nhiên;

for giá trị *Q* chưa hội tụ:

do: đặt trạng thái $s \in S$;

for *s* không phải là trạng thái cuối:

do:

Lựa chọn hành động *a* mới (dựa vào chính sách tối ưu);

Thực hiện hành động *a*;

Quan sát trạng thái *s* mới và thu nhận phần thưởng *R*;

Cập nhật;

$$Q_{st,at} = Q_{st,at} + \alpha * [r_t + \gamma * \max Q(st+1, a) - Q_{st,at}]$$

Cập nhật trạng thái $s \rightarrow s'$;

End for

End for

Đầu ra:

Hành động tốt nhất được lựa chọn *a'*.

❖ Thuật toán Q-learning sử dụng hàm chính sách tối ưu:

Thuật toán 1: Hàm chính sách tối ưu Epsilon-Greedy

Đầu vào: *α*: tỉ lệ học, γ: hệ số chiết khấu, *ε*: một số nhỏ

Kết quả: Một bảng *Q* chứa các giá trị *Q (S, A)* xác định chính sách ước tính tối ưu π^*

Khởi tạo *Q (s, a)* tùy ý, ngoại trừ các giá trị đầu và cuối *Q* (terminal,);

Q (terminal,) ← 0;

For chạy mỗi tập episode **do:**

Khởi tạo trạng thái S;

For *mỗi bước trong tập*

do:

A ← Lựa chọn hành động (Q, S, ε)

Thực hiện hành động A, sau đó quan sát phần thưởng R và trạng thái tiếp theo S'

$Q(S,A) \leftarrow Q(S,A) + \alpha^*[R + \gamma max_a\ Q(S',a) - Q(S,A)]$;

$S \leftarrow S'$

While S không phải là giá trị cuối cùng;

End

End

Thuật toán 2: Epsilon-Greedy lựa chọn hành động

Đầu vào: *Q*: Bảng *Q* đã được tạo cho đến thời điểm hi hiện tại

Kết quả: Hành động được lựa chọn

Phương thức: Lựa chọn hành động (*Q, S, ε*) is

n ← chọn 1 số ngẫu nhiên trong khoảng 0 đến 1

if $n < \varepsilon$ **then**

A ← hành động ngẫu nhiên từ tập không gian chứ

Else

A←*maxQ(S,)*;

End

Return hành động đã lựa chọn *A*;

End

Thuật toán 3: Hàm chính sách tối ưu Softmax function

Công thức của hàm softmax là:

$$\sigma(\vec{z})_i = \frac{e^{Zi}}{\sum_{j=1}^{K} e^{Zi}} \quad (5)$$

· $\vec{Z}$: Giá trị vector nhập vào cho hàm softmax (từ z_0 đến z_k)

· Zi: Tất cả các giá trị z đều là giá trị vector nhập cho hàm softmax. Chúng có thể là bất cứ số thực nào, số dương, số âm hay số 0. Ví dụ một mạng thần kinh nhân tạo có thể có giá trị vector ra là (-0,62; 8,12; 2,53). Đây không phải là phân phối xác suất đúng. Đó là vì sao ta cần đến hàm softmax.

· e^{Zi}: Hàm lũy thừa tiêu chuẩn được áp dụng cho mỗi giá trị nhập. Nó sẽ đưa ra một giá trị dương lớn hơn 0. Giá trị này sẽ rất nhỏ nếu giá trị nhập là âm, và rất lớn nếu giá trị nhập dương. Tuy nhiên, nó sẽ không cố định trong khoảng (0:1]. Đây là yêu cầu của một xác suất.

· $\sum_{j=1}^{K} e^{Zi}$: Dòng phía dưới của công thức là một cụm chuẩn hóa. Nó đảm bảo rằng tổng của các giá trị ra sẽ luôn bằng 1 và nằm trong khoảng (0:1]. Như vậy, sẽ xuất hiện phân phối xác suất chính xác.

3. XÂY DỰNG MÔ HÌNH THUẬT TOÁN HỌC TĂNG CƯỜNG ỨNG DỤNG CHO BÀI TOÁN ĐỊNH VỊ ROBOT DI ĐỘNG

❖ Thiết lập môi trường:

Đối với tác nhân ở đây là robot di động dạng hai bánh vi sai, việc thiết lập cho quá trình tự học của robot là một công đoạn hết sức quan trọng. Để thuận tiện cho robot hoạt động, tiến hành thực nghiệm trên hai dạng môi trường là môi trường thực và môi trường ảo. Mục đích là sẽ thực hiện quá trình training thu thập dữ liệu của robot trên môi trường ảo, từ đó sử dụng bộ dữ liệu đã training để giúp robot triển khai kết quả thuật toán trong môi trường thực.

- Môi trường ảo: Phần mềm Gazebo kết hợp Rviz, sử dụng mô hình mô phỏng robot thực training lấy dữ liệu;

- Môi trường vật lý: Xây dựng dựa trên môi trường ảo.

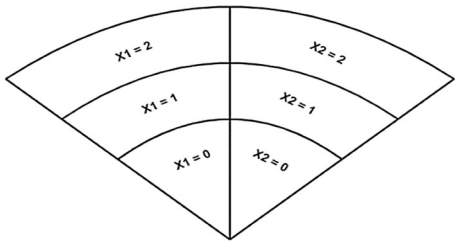
Đối với môi trường ảo, thực hiện xây dựng mô hình robot dạng vi sai với kích thước gần đúng để đạt được kết quả như mong đợi.

Tác nhân-robot thực hiện thu thập dữ liệu của môi trường bằng các quan sát (observation) từ Cảm biến Lidar SICK Scan NAV, xây dựng các ràng buộc không gian quan sát. Từ đó tổng hợp và tương tác với môi trường bằng các hành động được tối ưu bởi chính sách thiết lập. Đầu quá trình thu thập dữ liệu từ môi trường cần khởi tạo một số thông số liên quan tới: Vị trí khởi tạo; góc quét giới hạn, độ rộng dải quét, các vùng thiết lập của lidar; Vị trí điểm goal khởi tạo cho robot…

❖ Thiết lập, ràng buộc tập các trạng thái, các hành động khả dụng cho robot:

actions[5] = [tien_thang, re_trai, re_phai, dung_lai, quay_tron]

Không gian trạng thái của robot sẽ dựa vào phép đo từ cảm biến laser Sick Nav245. Vì cảm biến laser sử dụng có thể đo khoảng cách trong phạm vi 50 cm đến 50 mét và góc đo là 270 độ xung quanh robot. Điều này sẽ dẫn đến không gian trạng thái của robot là một không gian lớn vô hạn. Chúng ta cần căn chỉnh để giới hạn không gian giúp thuật toán có thể hội tụ.

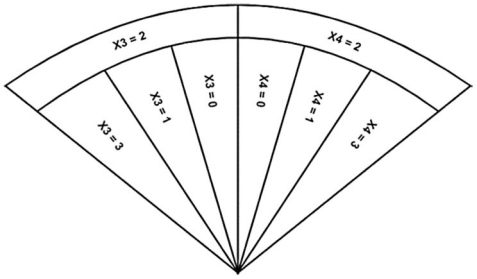


Hình 2. Ràng buộc tập không gian trạng thái của robot [1]

Đầu tiên, phép đo khoảng cách sẽ được lựa chọn là 1 mét và góc quét của cảm biến laser trong phạm vi [-135°,135°] theo hướng của robot đang di chuyển. Đây là một giả định hợp lý vì các chướng ngại vật cách xa robot hơn 1 mét và phía sau robot sẽ không gây nguy hiểm tới robot.

Không gian trạng thái sẽ bao gồm 4 phần tử (x_1, x_2, x_3, x_4) được xác định dựa trên khoảng cách và vị trí của chướng ngại vật đối với robot. Các trạng thái biến đối x1 và x2 được xác định dựa trên khoảng cách giữa vật cản đối với robot. Hai trạng thái này sẽ được xác định như sau:

$$x_i = \begin{cases} 0, 50\text{cm} \le d < 80\text{cm} \\ 1, 80\text{cm} \le d < 130\text{cm} \\ 2, 130\text{cm} \le d < 180\text{cm} \end{cases} \quad (i = 1,2)$$



Hình 3. Ràng buộc tập không gian trạng thái của robot [1]

Khoảng cách d là khoảng cách được tính từ vật cản đến rìa trái và rìa phải của robot. Ta có x_1 tương ứng sẽ là khoảng cách từ rìa trái robot đến vật cản còn x_2 là khoảng cách từ rìa phải robot đến vật cản.

Hai trạng thái còn lại là x_3 và x_4 sẽ được xác đinh dựa trên vị trí của chướng ngại vật so với robot. Chúng ta đánh dấu vị trí chướng ngại vật là p, p là đại diện cho góc mà robot phát hiện ra vật cản bởi cảm biến laser. Khoảng s1: $0° \le p \le \frac{h}{3}$; $s2 = \frac{h}{3} \le p \le 2\frac{h}{3}$ với h là 1/2 góc quét của cảm biến laser (=270°/2 = 135°). Hai trạng thái x_3, x_4 sẽ được xác định như sau:

$$x_i = \begin{cases} 0, \text{Robot phát hiện ra vật cản} \\ 1, \text{Robot phát hiện ra vật cản} \\ 2, \text{Robot bị vật cản bao phủ toàn phần} \\ 3, \text{Vật cản nằm ngoài phạm vi của robot} \end{cases} \quad (i = 3,4)$$

❖ Giới hạn hành động:

$$v = \begin{cases} 0{,}08\ \frac{m}{s}\text{, robot đi thẳng về phía trước} \\ 0{,}06\ \frac{m}{s}\text{, robot rẽ trái} \\ 0{,}06\ \frac{m}{s}\text{, robot rẽ phải} \end{cases}$$

$$\omega = \begin{cases} 0\ \frac{rad}{s}\text{, robot đi thẳng về phía trước} \\ +0{,}4\ \frac{rad}{s}\text{, robot rẽ trái} \\ -0{,}4\ \frac{rad}{s}\text{, robot rẽ phải} \end{cases}$$

❖ Xây dựng hàm chức năng phần thưởng

Để đạt được những điều trên, chức năng phần thưởng được định nghĩa là sự kết hợp của 3 phần thưởng khác nhau:

$$r_1 = \begin{cases} +0{,}2\ , at = đi\ thẳng \\ -0{,}1\ ,\ at = rẽ\ trái\ /\ rẽ\ phải \end{cases}$$

$$r_2 = \begin{cases} +0{,}2\ , Wr2.\Delta d < 0 \\ -0{,}2\ ,\ Wr2.\Delta d \geq 0 \end{cases}$$

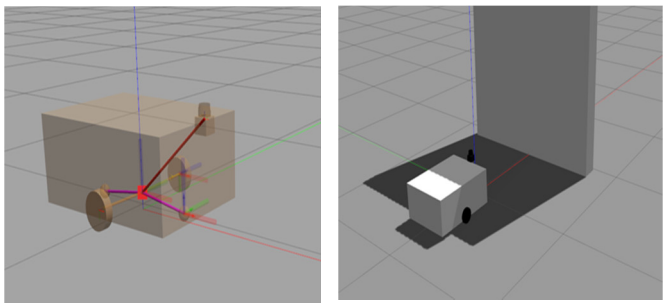
$$r_3 = \begin{cases} -0{,}8\ , robot\ thay\ đổi\ hướng\ quay \\ 0\ ,\ robot\ không\ thay\ đổi\ hướng\ quay \end{cases}$$

Tổng phần thưởng rt sẽ bằng tổng của ba phần trước trong trường hợp không va chạm, trong khi phần thưởng âm rất lớn là −100 trong trường hợp va chạm.

$$r_t = \begin{cases} -100\ , nếu\ có\ xảy\ ra\ va\ chạm \\ r1 + r2 + r3, nếu\ không\ xảy\ ra\ va\ chạm \end{cases}$$

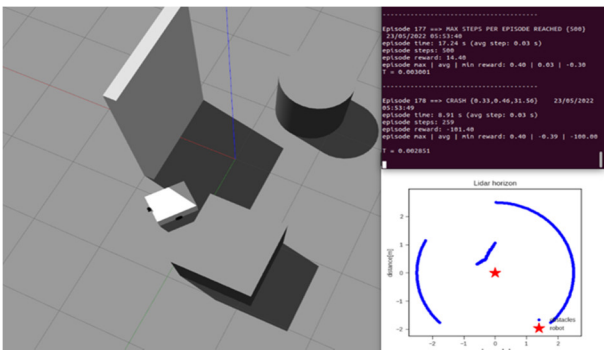
4. MÔ PHỎNG, THỰC NGHIỆM MÔ HÌNH HỌC TĂNG CƯỜNG

❖ Xây dựng mô hình mô phỏng:



Hình 4. Xây dựng mô hình và môi trường ảo trên phần mềm môi trường giả lập Gazebo

❖ Quy trình huấn luyện mô phỏng và thu thập dữ liệu:

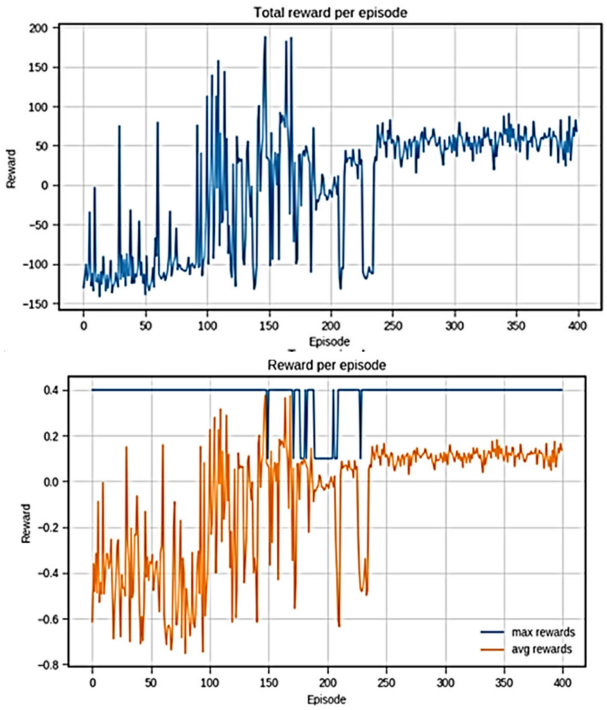


Hình 5. Quá trình huấn luyện mô hình và thu thập dữ liệu trên Gazebo

▪ Mục tiêu của quá trình là huấn luyện để mô hình có khả năng lưu lại các giá trị phần thưởng được trả về từ các biến trạng thái sau khi thực hiện một hành động, từ đó có khả năng vượt qua các vật cản.

▪ Sử dụng các hàm chính sách tối ưu khác nhau trong quá trình huấn luyện

▪ Quá trình huấn luyện sử dụng hàm chính sách tối ưu Epsilon-greedy:



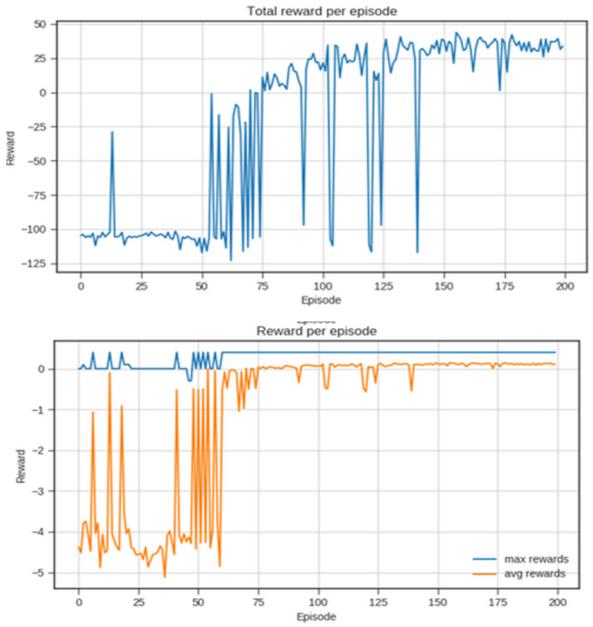
Hình 6. Đồ thị thể hiện phần thưởng nhận được trên từng episode và toàn episodes sử dụng hàm tối ưu Epsilon-Greedy

▪ Quá trình huấn luyện sử dụng hàm chính sách tối ưu Softmax-function:

▪ Đánh giá quá trình huấn luyện:

· **Quá trình huấn luyện sử dụng hàm chính sách tối ưu Epsilon-greedy:** Biên dạng đồ thị phần thưởng từ những episode nhỏ thể hiện sự kém ổn định do quá trình ban đầu, hàm epsilon càng lớn ưu tiên kích hoạt phương thức khám phá; nhưng từ các episode lớn trong khoảng [250,400], phần thưởng thu được có biên dạng ổn định do epsilon giảm dần, lựa chọn hành động tham lam, đồng thời các bước di chuyển của robot tăng dần trong quá trình huấn luyện thu được kết quả khả quan.

· **Quá trình huấn luyện sử dụng hàm chính sách tối ưu Softmax-function:** Biên dạng đồ thị phần thưởng từ những episode nhỏ cũng thể hiện sự kém ổn định tuy nhiên tối ưu hơn so với Epsilon-greedy do mỗi giá trị của tham số T mang trọng số của mỗi hành động được lựa chọn, từ những episode [150,200] biên dạng của phần thưởng tăng dần do với giá trị T càng tiến về 0 hành động tham lam được lựa chọn ưu tiên, đồng thời các bước di chuyển của robot tăng dần trong quá trình huấn luyện thu được kết quả khả quan.

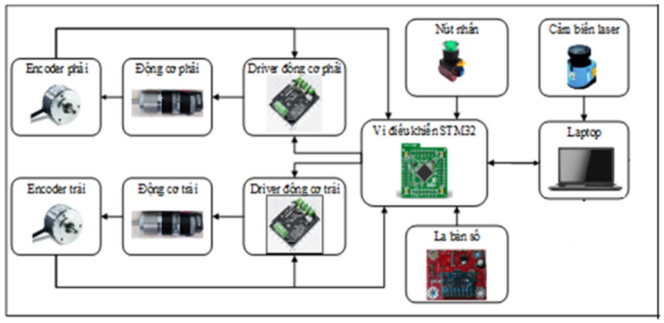


Hình 7. Đồ thị thể hiện phần thưởng nhận được trên từng episode và toàn episodes sử dụng hàm tối ưu Softmax function

❖ Thực nghiệm:

Sau khi thực hiện quá trình huấn luyện mô hình trong trường mô phỏng để thu thập dữ liệu, tiến hành thực nghiệm trên mô hình thực để chứng minh tính hiệu quả của phương pháp.

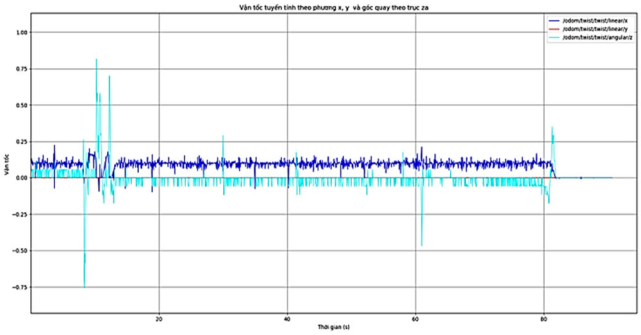
▪ Cấu trúc hệ thống điều khiển phần cứng robot:



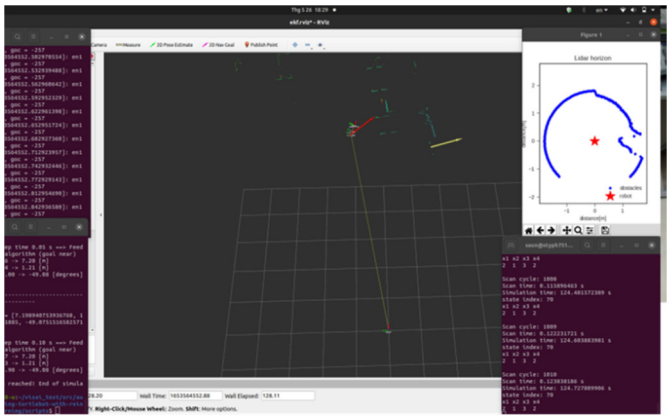
Hình 8. Sơ đồ phần cứng hệ thống robot

▪ Kết quả thực nghiệm:

· Đồ thị biểu diễn sự thay đổi vận tốc theo phương (x, y) và góc quay theo trục z khi di chuyển tới đích của robot:



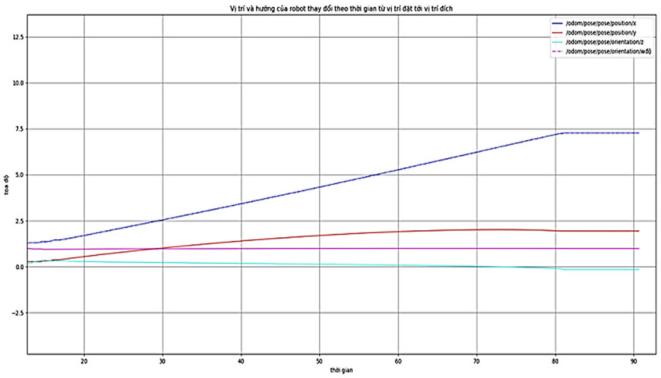
Hình 9. Đồ thị biểu diễn sự thay đổi của vận tốc robot theo phương (x,y) và góc quay z



Hình 10. Khởi tạo vị trí đặt và vị trí điểm đích cho robot

Hình 11. Robot di chuyển tới vị trí điểm đích đã đặt trước

· Đồ thị biểu diễn đáp ứng tọa độ của robot tại vị trí điểm đích A *(7,1989; 1,8065; -0,8565):*



Hình 12. Đồ thị đáp ứng theo vị trí và hướng của robot

Đánh giá kết quả thử nghiệm:

▪ Hệ thống đã hoạt động ổn định, cho thời gian đáp ứng đến vị trí điểm đặt nhanh;

▪ Thực hiện được yêu cầu phát hiện và tránh các vật cản trên đường di chuyển với môi trường không xác định, robot chưa biết vị trí vật cản cũng như biên dạng của môi trường làm việc;

▪ So sánh với phương pháp định vị robot truyền thống khác, phương pháp đề xuất có độ ổn định cao, thời gian đáp ứng nhanh và tích hợp thêm được tác vụ tránh vật cản khi di chuyển, robot không bị ảnh hưởng bởi các biên dạng vật cản khác nhau.

▪ Đồng thời phương pháp mới giúp giảm bớt bước thiết lập bản đồ trong môi trường cố định, giải quyết được bài toán định vị robot trong môi trường không xác định trước.

5. KẾT LUẬN

Nghiên cứu phương pháp học tăng cường đã giải quyết bài toán định vị cho robot trong môi trường không xác định; Tích hợp sử dụng các hàm chính sách tối ưu epsilon-greedy và softmax function giúp mô hình thuật toán có kết quả với độ tin cậy cao hơn. Trong quá trình thực nghiệm, robot thực hiện được nhiệm vụ định vị điểm đích và tự phát hiện cũng như tránh vật cản trong quá trình di chuyển.

Trong thời gian tới, nhóm nghiên cứu tiếp tục nghiên cứu mở rộng về các giải pháp định vị và kết hợp với laser hoặc camera nhằm xác lập được vị trí của robot, các vật cản để có thể ứng dụng trực tiếp với nhiều loại robot khác nhau; Nghiên cứu các phương pháp học tăng cường mở rộng như học tăng cường sâu (deep reinforcement learning), kết hợp sử dụng mạng thần kinh nhân tạo (neural network), phát triển ứng dụng các giải pháp điều khiển thông minh cho robot ngoài trời; Nghiên cứu hoàn thiện giải pháp điều khiển để giảm các sai lệch, đồng thời phát triển giải pháp cho hệ thống robot di động làm việc trong môi trường ngoài trời (outdoor), nhóm các robot đa nhiệm làm việc đồng thời.

TÀI LIỆU THAM KHẢO

[1]. Aleksa Lukoviÿ, Kosta Jovanoviÿ, 2020, *Application of Artificial Intelligence in Mobile Robotics and Autonomous Driving*. Master rad. University of Belgrade, Faculty of Electrical Engineering.

[2]. Sebag M. 2014. *A tour of machine learning: an AI perspective*. AI Commun; 27: 11-23.

[3]. Bellman RE., 1957. *A Markov decision process.* J Math Mech; 6: 679-684.

[4]. Watkins CJCH, 1989. *Learning from delayed rewards.* PhD, Cambridge University, Cambridge, UK.

[5]. Rummery GA, Niranjan M. 1994. *On-line Q-learning Using Connectionist Systems.* Cambridge, UK: Cambridge University Engineering Department.

[6]. Muhammad J, Bucak IO, 2013. *An improved Q-learning algorithm for an autonomous mobile robot navigation problem.* In: IEEE 2013 International Conference on Technological Advances in Electrical, Electronics and Computer Engineering; 9-11 May; Konya, Turkey. New York, NY, USA: IEEE. pp. 239-243.

[7]. Yun SC, Parasuraman S, Ganapathy V. 2013. *Mobile robot navigation: neural Q-learning*. Adv Comput Inf ; 178: 259-268.

[8]. Hwang KS, Lo CY, 2013. *Policy improvement by a model-free dyna architecture.* IEEE T Neural Network 2013; 24: 776-788.

[9]. http://www.beelab.info/2021/12/rl-series-thuat-toan-q-learning.htm

[10]. https://intellabs.github.io/coach/_images/design.png.