ĐIỀU HƯỚNG TỰ ĐỘNG CHO ROBOT DI ĐỘNG ĐỂ TRÁNH CHƯỚNG NGẠI VẬT TRONG MÔI TRƯỜNG KHÔNG XÁC ĐỊNH TRÊN CƠ SỞ THUẬT TOÁN Q-LEARNING

Trần Đức Chuyển[1*], Đào Huy Du[2*], Roãn Văn Hóa[1],
Nguyễn Đức Điển[1], Trần Ngọc Sơn[1], Vũ Thị Tố Linh[1]

Tóm tắt: *Bài báo này trình bày phương pháp xây dựng hệ thống điều hướng tự động thông minh cho robot di động đa hướng trong môi trường phẳng không xác định. Các nghiên cứu sử dụng công cụ lập trình là hệ điều hành cho robot di động (Robot Operating System - ROS). Từ các thông tin cập nhật về bản đồ, môi trường hoạt động, vị trí điều khiển của robot, và vật cản (Simultaneous Localization and Mapping - SLAM) để nhằm tính toán quỹ đạo cho robot trong hệ thống điều hướng tự động tránh các chướng ngại vật. Hệ thống điều hướng tính toán quỹ đạo toàn cục và quỹ đạo cục bộ cho robot trên cơ sở ứng dụng thuật toán Q-learning. Các kết quả nghiên cứu mô phỏng trong môi trường Gazebo và đặc biệt chạy thực nghiệm trên robot di động Turtlebot thật đã cho thấy tính hiệu quả thực tế của việc điều hướng tự động cho robot di động này.*

Từ khóa: Trí tuệ nhân tạo; Điều khiển robot di động; Thuật toán Q-learning; Học tăng cường.

1. MỞ ĐẦU

Thuật ngữ robot và điều khiển robot ngày nay trở nên thông dụng, đang từng bước gắn chặt với cuộc sống hàng ngày của con người như robot phục vụ (robot hút bụi, lau nhà, nấu ăn, pha chế,...), robot công nghiệp (các robot trong các dây chuyền sản xuất), robot trong y tế, robot trong lĩnh vực quân sự,... Kỹ thuật robot là một lĩnh vực đa ngành gồm cơ khí, điện, điện tử, điều khiển tự động và công nghệ thông tin. Mỗi lĩnh vực khi nghiên cứu đều đóng vai trò quan trọng trong quá trình nghiên cứu, thiết kế chế tạo robot, [1-4]. Trong ngành lĩnh vực kỹ thuật điều khiển và công nghệ thông tin, học tăng cường (reinforcement learning) là một lĩnh vực con của học máy, nghiên cứu cách thức một tác nhân trong một môi trường nên chọn thực hiện các hành động nào để cực đại hóa một khoản phần thưởng (reward) nào đó về lâu dài. Các thuật toán học tăng cường cố gắng tìm một chiến lược ánh xạ các trạng thái của môi trường tới các hành động mà tác nhân nên chọn trong các trạng thái đó, [8, 9, 11- 13, 15]. Môi trường hành động để điều khiển cho robot thường được biểu diễn dưới dạng một quá trình quyết định Markop trạng thái hữu hạn (Markov decision process - MDP), và các thuật toán học tăng cường cho ngữ cảnh này có liên quan nhiều đến các kỹ thuật quy hoạch động. Các xác suất chuyển trạng thái và các xác suất thu lợi trong MDP thường là ngẫu nhiên nhưng lại tĩnh trong quá trình của bài toán điều khiển robot. Khác với học có giám sát, trong học tăng cường không có các cặp dữ liệu vào/kết quả đúng, các hành động gần tối ưu cũng không được đánh giá đúng sai một cách tường minh. Hơn nữa, ở đây, hoạt động trực tuyến (on-line performance) được quan tâm, trong đó có việc tìm kiếm một sự cân bằng giữa khám phá (lãnh thổ chưa lập bản đồ) và khai thác (tri thức hiện có). Có hai phương pháp thường được sử dụng để giải các bài toán quyết định đó là tìm kiếm trong không gian chiến lược và tìm kiếm trong không gian hàm giá trị hay còn gọi là "phép lặp chiến lược" và "phép lặp giá trị". Hai phương pháp này chính là các giải thuật học tăng cường đặc trưng. Bên cạnh đó, trong những nghiên cứu gần đây các nhà khoa học đề xuất một phương pháp kết hợp giữa hai phương pháp trên đó chính là phương pháp Actor-Critic learning, [6, 7, 11, 13, 14].

Trong lĩnh vực điều khiển robot, thuật toán Q-learning thuộc nhóm phương pháp dựa trên giá trị (value-based method), nghĩa là thuật toán này đã tìm cách tính toán hàm giá trị, rồi từ hàm giá trị đưa ra chính sách tối ưu. Dù đây là thuật toán được cho là đơn giản nhưng Q-learning lại là nền tảng của hầu hết các thuật toán học tăng cường quan trọng sau này. Trong một số các nghiên cứu gần đây, ở tài liệu [5] các tác giả chỉ dừng lại vấn đề áp dụng thuật toán một cách máy móc cho bài toán tìm đường ngắn nhất, mà chưa đi sâu vào xử lý các tình huống thực tế trong môi

trường làm việc rộng lớn có nhiều yếu tố bất định.

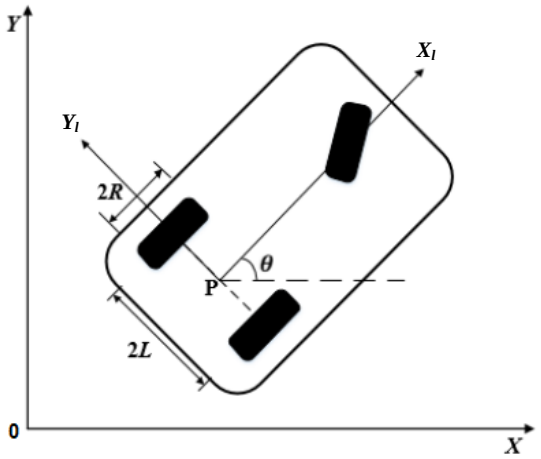
Trong bài báo này sử dụng thuật toán Q-learning để nhằm giải quyết bài toán tìm đường đi và tránh vật cản (vật cản cố định, vật cản có thể là di động) cho robot di động trong môi trường làm việc và không gian hành động thực tế.

Phần còn lại của bài báo được tổ chức như sau. Phần thứ hai là nội dung nghiên cứu; trong đó có mô hình được đề xuất, ứng dụng thuật toán học tăng cường Q-learning để điều hướng robot di động để tránh chướng ngại vật trong môi trường không xác định. Kết quả nghiên cứu sẽ được đưa ra trong phần thứ ba. Cuối cùng, phần thứ tư là kết luận của bài báo này.

2. NỘI DUNG NGHIÊN CỨU

2.1. Xây dựng mô hình điều khiển cho robot di động đa hướng

Robot là một hệ cơ học phức tạp, nhiều khối lượng và có thể nhiều bậc tự do. Mỗi bậc tự do thực hiện một chuyển động và được điều khiển bởi một hệ truyền động điện. Hơn nữa robot là đối tượng điều khiển chứa nhiều động cơ có liên hệ ràng buộc lẫn nhau. Để xây dựng mô hình hệ thống điều khiển cho robot ta xem xét một rô bốt di động như hình 1 với ba bánh xe, hai trong số đó là bánh xe tiêu chuẩn lái xe nằm ở phía sau khung và một là bánh xe phía trước có thể làm cho robot di động giữ thăng bằng và không gây ra bất kỳ hạn chế chuyển động nào đối với robot di động, [2, 8, 13, 15].



Hình 1. *Mô hình robot di động đa hướng.*

Hai khung tọa độ có thể được sử dụng để mô tả chuyển động của rô bốt di động. Một là khung tọa độ toàn cục (X, Y) được xác định trên mặt phẳng thực tế của robot sẽ di chuyển và khung còn lại là khung tọa độ cục bộ (Xl, Yl) được xác định trên rô bốt di động. Góc giữa hai khung tọa độ được ký hiệu là θ. Chuyển động của robot sẽ được xác định cho ngăn xếp điều hướng. Như tọa độ tổng thể được chọn trong hình 1, rõ ràng là vận tốc của rô bốt chứa ba thành phần: vận tốc thẳng dọc theo trục OX và tốc độ góc theo trục OY.

Để chỉ định vị trí của rô bốt di động đa hướng, tọa độ P được chọn trên khung của rô bốt làm điểm tham chiếu vị trí trung tâm điều khiển của nó. Điểm P được định vị bởi tọa độ (x, y) trong khung toàn cục của toàn bộ môi trường điều khiển robot. Để mô tả chuyển động của rô bốt di động là các chuyển động thành phần, cần phải xác định ánh xạ chuyển động dọc theo các trục của khung hình cầu với chuyển động dọc theo các trục tọa độ của khung cục bộ mà toàn bộ môi trường điều khiển rô bốt. Ánh xạ này được biểu diễn dưới dạng biểu thức sau:

$$\begin{bmatrix} \dot{x}_l \\ \dot{y}_l \\ \dot{\theta}_l \end{bmatrix} = \begin{bmatrix} cos\theta & sin\theta & 0 \\ -sin\theta & cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} \quad (1)$$

Bằng cách xem xét giới hạn trượt tiêu chuẩn của bánh xe, $\dot{y}_l = 0$, nghĩa là bánh xe không thể trượt trực giao với mặt phẳng bánh xe, khi đó, ta có thể thu được:

$$-\dot{x}\sin\theta + \dot{y}\cos\theta = 0 \tag{2}$$

Biểu thị vận tốc chuyển động $\dot{x}_l$ của rô bốt di động đa hướng là *v* và tốc độ quay θ là *w*, khi đó, động học của rô bốt di động đa hướng trở thành:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix} \tag{3}$$

Vận tốc chuyển động tịnh tiến và tốc độ quay của rô bốt di động nhiều chiều có mối quan hệ sau với vận tốc thẳng của hai bánh:

$$v = \frac{1}{2}(v_l + v_r), \qquad w = \frac{v_l - v_r}{2L} \tag{4}$$

Nếu xét vận tốc thẳng tại các bánh xe và vận tốc góc của hai bánh xe đủ tiêu chuẩn thì ta được: $v_l = w_l R$ và $v_r = w_r R$. Khi đó, ta có thể biểu diễn vận tốc góc của hai bánh xe chuẩn theo vận tốc chuyển động tịnh tiến và tốc độ quay của rô bốt di động đa hướng sẽ là:

$$w_l = \frac{v + wL}{R}, \quad w_r = \frac{v - wL}{R} \tag{5}$$

Bằng cách xem xét gia tốc của rô bốt di động đa hướng trong giới hạn tọa độ làm việc cục bộ của nó, thì mô hình động có thể được biểu diễn như [6]:

$$\begin{bmatrix} \dot{v} \\ \dot{w} \end{bmatrix} = \begin{bmatrix} \frac{1}{Rm} & \frac{1}{Rm} \\ \frac{L}{RJ} & \frac{L}{RJ} \end{bmatrix} \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} \tag{6}$$

trong đó, m và J lần lượt là khối lượng và mô men quán tính của rô bốt di động; R là bán kính của hai bánh xe tiêu chuẩn; L là nửa khoảng cách giữa hai bánh xe tiêu chuẩn; $\tau = [\tau_1 \ \tau_2]^T$ là vectơ mô-men xoắn đầu vào từ động cơ tác dụng đến hai bánh xe tiêu chuẩn, [1- 4].

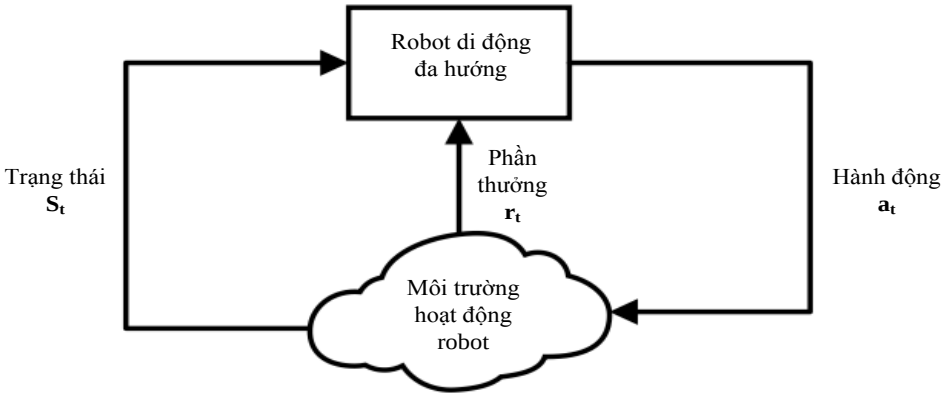
Mục tiêu là dạy cho rô bốt di động đa hướng bám theo chương trình điều khiển trên cơ sở nhận dạng và xử lý dữ liệu, thông qua các thiết bị nhận dạng như camera thông minh; cảm biến thông minh và thuật toán điều khiển, sao cho robot đi theo các quỹ đạo nhất định trong không gian và môi trường làm việc phù hợp với các tham số $e_{x,k} \approx 0,\ e_{y,k} \approx 0,\ e_{\theta,k} \approx 0$.

2.2. Ứng dụng thuật toán Q-learning để điều khiển robot di động đa hướng

2.2.1. Phương pháp học tăng cường với thuật toán Q-learning

Phương pháp học tăng cường với thuật toán Q-learning là một nhánh của học máy được phát triển để phục vụ cho việc tính toán thông minh cho lĩnh vực khoa học kỹ thuật nói chung và trên phương diện điều khiển học nói riêng. Nơi mà "giá trị hành động bằng giá trị hiện tại cộng với tổng các giá trị tối ưu trong tương lai". Với Q-learning nói riêng và học tăng cường nói chung, thì mọi thứ được chia thành "trạng thái - s_t" và "hành động - a_t" với thời gian được biểu thị bằng một chuỗi các bước thời gian (t = 0, 1, 2,...). Đối với môi trường làm việc liên tục như điều khiển robot di động thì việc đầu tiên cần làm là lượng tử hóa không gian trạng thái để có cập nhật $S = \{S_1, S_2, ...S_m\}$ và lượng tử hóa không gian hành động thành tập $A = \{a_1, a_2, ...a_n\}$, kết quả là

môi trường tạo ra phần thưởng $r_t = r(s_t, a) \in R$, để hiểu rõ hơn ta có sơ đồ tương tác môi trường học tập như hình 2.



Hình 2. *Sơ đồ tương tác môi trường học tập của robot.*

Khi đó, cách Q-learning hoạt động là tính toán và lưu giữ giá trị Q trên một hành động và trạng thái cụ thể, Q(s, a). Tất cả những thông tin, kinh nghiệm tích lũy từ những lần tính toán trước đó sẽ được mã hóa thành một bảng đánh giá.

Chúng ta tính toán tổng phần thưởng thu được sau thời gian t là R_t được hoàn trả về như sau:

$$R_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k} \tag{7}$$

trong đó, $0 \le \gamma < 1$ là hệ số khấu trừ cho các phần thưởng. Giá trị của γ càng nhỏ thì phần thưởng càng được chú trọng trong khi thực hiện hành động. Khi đó, hàm giá trị hành động (hàm Q) được xác định như sau:

$$Q^n(s,a) = E_\pi \{R_t | s_t = s, a_t = a\} \tag{8}$$

trong đó, $E_\pi\{...\}$ đại diện cho kỳ vọng theo chính sách ngẫu nhiên. Hàm $Q^n(s,a)$ đại diện cho tổng phần thưởng chiết khấu dự kiến khi ta chọn hành động a dưới trạng thái s và sau đó chọn hành động theo chính sách π. Hàm Q được mô tả dưới dạng công thức đệ quy sau:

$$Q^\pi(s,a) = \sum_{s' \in S} \Pr(s'|s,a) \left(r(s,a,s') + \gamma \sum_{a' \in A} \pi(a'|s') Q^\pi(s',a') \right) \tag{9}$$

trong đó, S và A lần lượt là tập trạng thái và tập hành động. Từ công thức này, chúng ta có thể xác định rằng, hàm Q theo chính sách tối ưu π^*, tức là hàm Q tối ưu, thỏa mãn phương trình sau, được gọi là phương trình tối ưu Bellman:

$$Q^*(s,a) = E_{s'} \{ r_t + \gamma \max_{a'} Q^*(s',a') \} \tag{10}$$

Trong thuật toán Q-learning, bằng cách cập nhật lặp đi lặp lại hàm Q bằng cách sử dụng (10) dựa trên các dữ liệu thực nghiệm, hàm Q có thể được hội tụ ngẫu nhiên thành $Q^*(s,a)$ và do đó, chính sách tối ưu có thể được xác định là hàm chính sách đối với $Q^* : a^* = argmax_a(s,a)$. Trong thực tế, tác nhân học tập của robot khi di chuyển phải khám phá môi trường hành động vì hàm Q không đáng tin cậy và cần phải lựa chọn hành động để được sử dụng một cách rộng rãi như một chính sách ngẫu nhiên, khi đó cho phép để chọn một hành động có xác suất cho một

trạng thái đầu vào s. Cụ thể hơn, chính sách μ sẽ tham gia lựa chọn một hành động nhằm tối đa hóa hàm Q ở trạng thái s với xác suất a của 1- μ, μ ∈ [0, 1] và cho phép lựa chọn một hành động ngẫu nhiên với xác suất còn lại. Khi các trạng thái và hành động là rời rạc và khác nhau, một cách đơn giản để biểu diễn hàm Q là sử dụng như một bảng giá trị cho tất cả các cặp trạng thái và hành động như sau:

$$Q(s,a) \leftarrow Q(s,a) + \alpha.\left((r + \gamma \max_{a'} Q(s',a') - Q(s',a') \right) \tag{11}$$

Trong đó, $0 < \alpha \leq 1$ là tốc độ học và tốc độ học càng lớn thì dữ liệu mới cập nhật càng nhanh. Với thuật toán này, bảng Q hội tụ đến hàm Q tối ưu trong điều kiện hội tụ của xấp xỉ ngẫu nhiên. Mặt khác, vì điều này dựa trên phương pháp xấp xỉ ngẫu nhiên, nên cần có một số lượng dữ liệu thích hợp cho tất cả các cặp trạng thái (s, a) thay đổi.

Trong phương pháp Q-learning dạng bảng, khi số lượng phần tử trong trạng thái hoặc không gian hành động là rất lớn hay trạng thái hoặc không gian hành động là liên tục, chúng ta thường biểu diễn hàm Q dưới dạng hàm tham số Q(s, a; θ) bằng cách sử dụng các tham số θ và sau đó cập nhật các thông số theo biểu thức gradient như sau:

$$\theta \leftarrow \theta + \alpha\left(target_Q - Q(s,a;\theta)\right)\nabla_\theta Q(s,a;\theta) \tag{12}$$

Ở đây, "$target_Q$" là giá trị mục tiêu dựa trên phương trình Bellman tối ưu (10) và nó được tính toán như sau:

$$target_Q\, r(s,a;s') + \gamma \max_{a'} Q(s',a';\theta) \tag{13}$$

Hàm Q được cập nhật theo một trình tự nhất quán của nó. Thuật toán Q-learning là một phương pháp dựa trên các hàm giá trị và từ hàm giá trị đưa ra chính sách tối ưu, trong đó, giá trị xấp xỉ hàm Q được hồi quy về giá trị mục tiêu, giá trị này phụ thuộc vào chính nó. Điều này ngụ ý rằng, giá trị đích thực thay đổi tự động khi luật học tập được cập nhật. Do đó, khi một hàm phi tuyến tính, chẳng hạn như mạng nơ-ron được sử dụng để xấp xỉ p, quá trình học tập này trở nên không ổn định do các thay đổi động học trong mục tiêu và trong trường hợp xấu nhất, hàm Q sẽ phân kỳ (theo Sutton và Barto, 1998).

2.2.2. Điều hướng robot di động sử dụng thuật toán Q-learning

Trong thuật toán Q-learning giá trị của các vị trí điều khiển điều hướng cho robot thường được cập nhật theo phương pháp vi phân tức thời, sử dụng sai lệch giữa một bước lặp để ước lượng, tính toán hàm giá trị Q theo các biểu thức (11) ở trên. Khi gặp các bài toán điều hướng cho robot di chuyển với nhiều trạng thái khác nhau và hành động dịch chuyển (sang trái, sang phải), tránh chướng ngại vật (vật cản di động, vật cản cố định),… ta chọn α = 0,1; γ = 0,95; Lúc này, robot di chuyển với nhiều tình huống khác nhau, quá trình cập nhật bảng Q được thực hiện.

Khi bắt đầu huấn luyện thuật toán, đối tượng sẽ đi một hoặc hai lần sang phải, nhưng ngay khi hành động sang trái được chọn, thì hành động này sẽ tiếp tục được chọn ở những lần di chuyển tiếp theo bởi vì luôn nhận được phần thưởng khi thực thi hành động sang trái này.

Mục tiêu của mô hình trong quá trình điều hướng cho robot, là giữ cho nó trong một giới hạn cho phép, tức là ± 5 độ. Lúc đầu, mô hình robot, ma trận Q, chính sách π sẽ được khởi tạo. Có một số điểm quan trọng để thực hiện điều hướng trong quá trình di chuyển, như các trạng thái không hữu hạn. Trong phạm vi giới hạn, có thể có hàng trăm và hàng nghìn góc cao độ, và có hàng nghìn cột là không thể xảy ra khi cập nhật thuật toán Q-learning. Vì vậy, ta đã sắp xếp các giá trị trạng thái thành 20 góc trạng thái từ -10 độ đến 10 độ. Đối với giá trị hành động, chúng ta đã chọn mười vận tốc khác nhau và chúng là [- 200, - 100, - 50, - 25, - 10, 10, 25, 50, 100, 200]

ms^{-1}. Ma trận Q có 20 cột, mỗi cột đại diện cho một trạng thái và mười hàng mỗi hàng đại diện cho mọi hành động. Ban đầu, các giá trị Q được giả định là 0 và một số hành động ngẫu nhiên được chỉ định cho mọi trạng thái trong chính sách π. Chúng ta đã huấn luyện trong 1600 tập, mỗi tập có 2000 lần lặp lại. Vào đầu mỗi tập dạy, thì đều được mô phỏng được làm mới. Bất cứ khi nào trạng thái của robot vượt quá giới hạn, nó sẽ bị phạt bằng cách gán phần thưởng cho -100. Bảng Q được cập nhật ở mỗi bước theo biểu thức (11). Từ đó, ta có thuật toán thiết lập quỹ đạo điều hướng tự động cho robot được hiển thị như sau:

Algorithm: Q Learning Algorithm as applied in the system

```
Initialize Robot;
Initialize Q Matrix Q;
Initialize Policy π;
Initialize Penalty Reward pen;
for number of episodes do
    Reset simulation ;
    Wait for 1-second ;
    Pause simulation ;
    Read the pitch angle φ of the robot ;
    state ← φ ;
    Unpause simulation ;
    for number of iterations do
        Generate a random number rand;
        if rand ≤ μ then
            take random action ;
        end
        else
            take action based on π ;
        end
        state_new ← φ;
        Pause simulation;
        if absolute value of state_new ≥ limit then
            if reward_total ≤ Target then
                reward ← pen;
                Update Q ;
                Update π;
            end
            Break ;
            else
                Print Passed ;
                Break ;
            end
        end
        else
            reward ← 1;
            Update Q;
            Update π state ← state_new
        end
    end
end
```

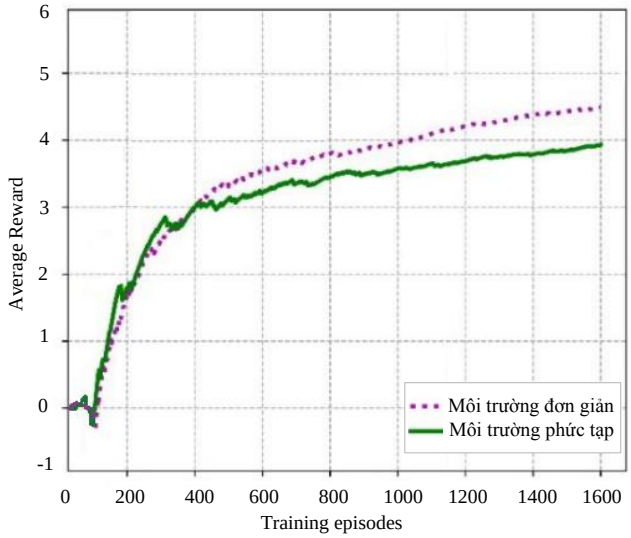
Thuật toán Q-learning thực hiện các tác nhân hành động cho việc điều hướng tự động thông minh cho robot di động nhằm thực hiện các quỹ đạo để tránh các vật cản động cũng như vật cản

tĩnh trong quá trình di chuyển của robot đồng thời tính ra quỹ đạo ngắn nhất cho robot di chuyển đến đích với một đường đi nhanh nhất.

3. KẾT QUẢ MÔ PHỎNG VÀ THỰC NGHIỆM

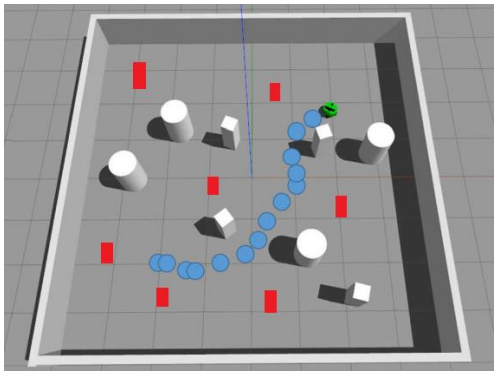
3.1. Nghiên cứu mô phỏng

Với thuật toán đã nghiên cứu và đề xuất ở phần hai như trên tiến hành nghiên cứu một số mô phỏng được tiến hành dựa trên công cụ môi trường Gazebo. Hình 3 trình bày kết quả học tập trong môi trường mô phỏng Gazebo. Như chúng ta có thể thấy, phần thưởng trung bình của robot trong mỗi tập dạy không ngừng tăng lên khi quá trình đào tạo tiếp tục. Robot sẽ học được kiến thức về môi trường thông qua việc tương tác với môi trường. Cuối cùng, robot có thể điều hướng đến đích một cách nhanh chóng và tự chủ trong cả môi trường đơn giản và phức tạp mà không có bất kỳ những va chạm nào với các chướng ngại vật. Các nghiên cứu thử nghiệm cho thấy tính hiệu quả của mô hình mà các tác giả đã đề xuất.



Hình 3. *Kết quả phần thưởng trung bình của quá trình học tập.*

Trong phần này, một số mô phỏng được thực hiện dựa trên công cụ mô phỏng trong môi trường Gazebo. Như trong hình 4 cho thấy, bản đồ được xây dựng trên Gazebo là bản đồ được tạo ra với các bức tường nghiêm ngặt và một robot di động màu xanh lá cây có thể được điều khiển để di chuyển xung quanh các chướng ngại vật cố định tạo bối cảnh cho các chuyển động của robot được sử dụng để xây dựng bản đồ hành động của robot di động.



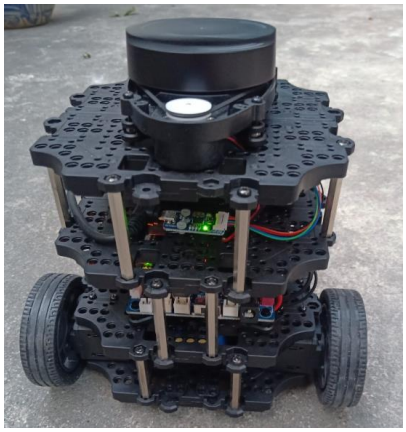
Hình 4. *Xây dựng bản đồ trực quan và mô hình mô phỏng robot trong Gazebo.*

Các đường màu xanh sẫm thể hiện đường đi của robot khi tránh chướng ngại vật được tạo bởi cảm biến thông minh, camera thông minh và vị trí hiện tại của robot được cập nhật (robot di động được ký hiệu bằng màu xanh lá cây) bằng cách sử dụng kích thước đo hình học. Đây là một công cụ trực quan có thể cung cấp cập nhật trực tiếp các bản đồ được tạo từ thuật toán SLAM để điều khiển robot di động. Hơn nữa, quỹ đạo chuyển động của robot di động đa hướng trong bản đồ cũng có thể tự động điều hướng và trong môi trường này luôn có thể tạo ra các chướng ngại vật, như trong hình 4 để cho robot di chuyển đến đích an toàn.

3.2. Nghiên cứu thực nghiệm

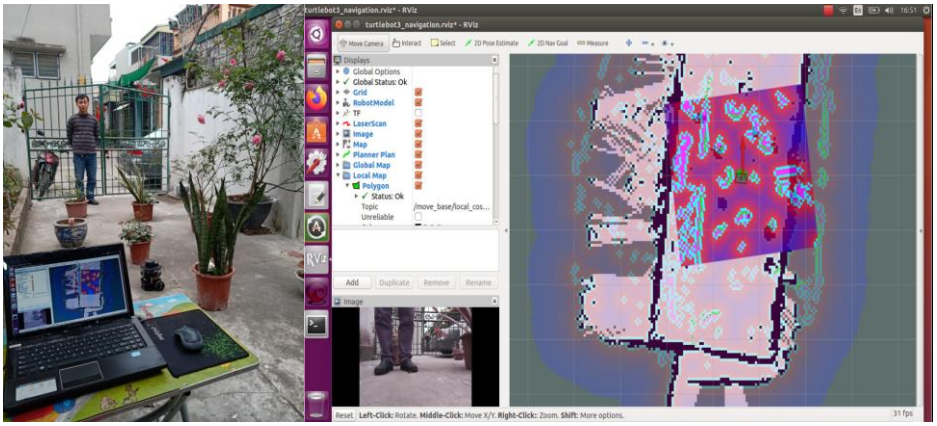
Sau đây, tác giả đi vào nghiên cứu mô hình robot di động đa hướng: với kiến trúc phần cứng thực tế của robot này được thể hiện như hình 5, trong đó, mỗi module phần cứng sẽ thực hiện một số nhiệm vụ trong chuỗi các hoạt động của robot di động đa hướng này: chẳng hạn như tìm đường, vượt chướng ngại vật,...

Trong các thử nghiệm, các tác giả thực hiện một số nghiên cứu trong hoạt động của robot di động đa hướng, như sử dụng máy tính nhúng Raspberry Pi 3 Model B+ hỗ trợ Ubuntu. Việc làm này nhằm trực tiếp xử lý thông tin từ một loạt các cảm biến thông minh, camera thông minh Astra và cảm biến thông minh, sau đó truyền lệnh đến một bộ vi điều khiển thông minh. Để ghi lại hình ảnh từ môi trường cũng như đo khoảng cách giữa robot di động đa hướng và các vật cản không xác định, robot di động được trang bị Rplidar được đặt ở trên cùng để quét 360 độ các giá trị đến vật cản môi trường xung quanh. Và phạm vi quét laser trong vòng 15 m tạo ra dữ liệu bản đồ được sử dụng cho quá trình lập bản đồ. Mạch điều khiển vi xử lý thông minh sẽ là bộ phận nhận tín hiệu điều khiển từ Raspberry Pi 3 Model B+, sau đó đưa tín hiệu đến mạch cầu MOSFET để vận hành hai động cơ.



Hình 5. *Mô hình robot Turtlebot di động đa hướng.*

Tiến hành thực nghiệm với robot Turtlebot di động đa hướng thực tế vào môi trường thực, đó là môi trường được sử dụng để thử nghiệm thực tế. Môi trường ở đây bao gồm các chướng ngại vật là các chậu hoa được đặt tại các vị trí khác nhau, có khoảng cách bất kỳ để tạo ra moi trường thật để robot di chuyển. Một người di chuyển làm vật cản di động bất kỳ tại vị trí nào trên mặt phẳng sân, robot được đặt trong mặt phẳng như hình 5, hình 6. Đường đi của robot sẽ được dạy trước thông qua máy tính, mạng Wifi và robot Turtlebot thực tế dựa trên thuật toán Q-learning và thuật toán SLAM. Ở đây chủ yếu là một công cụ trực quan hóa có thể cung cấp các bản cập nhật trực tiếp của các bản đồ được tạo từ các thuật toán SLAM và Q-learning. Hơn nữa, quỹ đạo của robot trong bản đồ cũng có thể được hiển thị trong môi trường thực tế mà quá trình đào tạo và dạy, nhận dạng để robot biết được trong quá trình tránh chướng ngại vật một cách thông minh và hoàn hảo.



Hình 6. *Xây dựng bản đồ trực quan và mô hình mô phỏng robot trong Rviz.*

Kết quả thu được như hình 6, quỹ đạo chuyển động toàn cục của robot được thể hiện trên bản đồ hóa trong môi cả môi trường Rviz (ở hình này, phía bên phải là giao diện theo dõi chuyển động của robot trên Rviz và thực tế). Kết quả cho thấy, robot đã lập quỹ đạo bản đồ đường đi một cách tối ưu, và ở đây, quá trình di chuyển hoàn toàn tránh được các chướng ngại vật cố định dựa trên bản đồ đã thiết lập. Khi gặp chướng ngại vật (không xác định) không có trong quá trình tạo của bản đồ hóa, thì robot sẽ lập quỹ đạo chuyển động một cách cục bộ và điều hướng thông minh. Khi đó, quỹ đạo di chuyển là đường màu đỏ thay đổi để robot có thể di chuyển vòng ra sau người đang đi tới, đảm bảo việc robot có thể tránh vật cản (động) và vẫn giữ nguyên xu hướng chuyển động đi tới vị trí đích như hình 6. Kết quả hình 6 cho thấy, robot tránh được cả chướng ngại vật di động (là người khi đang đến gần robot).

Những kết quả này mang lại lợi ích thực tiễn cao, có thể thực hiện hầu hết trên các robot trong công nghiệp, robot di động trong giao thông, trong y tế và robot tự hành trong các nhà máy xí nghiệp công nghiệp.

4. KẾT LUẬN

Trong bài báo này, các tác giả đã xây dựng hệ thống điều hướng thông minh cho robot di động đa hướng trong môi trường phẳng không xác định, sử dụng công cụ là ROS để lập trình điều khiển. Từ các thông tin về bản đồ, môi trường, vị trí hoạt động của robot, và vật cản (SLAM) kết hợp thuật toán Q-learning để tính toán quỹ đạo chuyển động cho robot trong hệ thống điều hướng thông minh này. Các kết quả mô phỏng và thực nghiệm trên robot Turtlebot đã cho thấy tính hiệu quả thực tế của hệ thống điều hướng tự động cho robot mà các tác giả đã xây dựng. Hướng phát triển của vấn đề nghiên cứu là mong muốn, sẽ được thực hiện áp dụng trên một số loại robot thực tế trong lĩnh vực công nghiệp và đời sống.

Lời cảm ơn: *Nghiên cứu này được hỗ trợ bởi nhóm nghiên cứu AI "CHĐ" của Khoa Điện, Trường Đại học Kinh tế - Kỹ thuật Công nghiệp; Số 353 Đường Trần Hưng Đạo, Phường Bà Triệu, Thành phố Nam Định. http://www.uneti.edu.vn/.*

TÀI LIỆU THAM KHẢO

[1]. A. Folkers, M. Rick, and C. Buskens, "*Controlling an autonomous vehicle with deep reinforcement learning*," IEEE Intell. Veh. Symp. Proc., vol. 2019-June, (2019), pp. 2025–2031.

[2]. Cuong Nguyen Manh, Tien Ngo Manh, Dung Pham Tien, Van Nguyen Thi Thanh, Manh Tran Van, Duyen Ha Thi Thanh and Duy Nguyen Duc "*Autonomous Navigation for Omnidirectional Robot Based on Deep Reinforcement Learning*," IJMERR, Vol. 9, No. 8, August (2020), pp: 1134-1139.

[3]. Do Quang Hiep, Ngo Manh Tien, Nguyen Manh Cuong, Pham Tien Dung, Tran Van Manh, Nguyen Tien Kiem, Nguyen Duc Duy, "*An Approach to Design Navigation System for Omnidirectional Mobile Robot Based on ROS*", IJMERR, Volume 11; Issue 9; (2020), pp: 1502-1508.

[4]. Roan Van Hoa, Tran Duc Chuyen, Nguyen Tung Lam, Nguyen Duc Dien, Tran Ngoc Son, Vu Thi To Linh, "*Reinforcement Learning based Method for Autonomous Navigation of Mobile Robots in Unknown Environments"*, Proceedings of the 2020 International Conference on Advanced Mechatronic Systems, Hanoi, Vietnam, December 10 - 13, (2020), pp. 266-269.
[5]. Hado van Hasselt, "*Double Q – learning*" Advances in neural information processing systems, (2010) pp. 1-9.
[6]. L. Xin, Q. Wang, J. She, and Y. Li, "*Robust adaptive tracking control of wheeled mobile robot*," Robotics and Autonomous Systems, vol. 78, (2016), pp. 36–48.
[7]. Shota Ohnishi, Eiji Uchibe, Yotaro Yamaguchi, Kosuke Nakanishi, Yuji Yasui and Shin Ishii, "*Constrained Deep Q-Learning Gradually Approaching Ordinary Q-Learning*", Vol 13, Publishing by Frontiers in Neurorobotics Journal, December (2019), pp 7-12.
[8]. Andrea Bacciotti**,** "*Stability and Control of Linear Systems"***,** Publishing Ltd; Springer Nature Switzerland AG, (2019).
[9]. Charu C. Aggarwal, "*Neural Networks and Deep Learning*", Springer International Publishing AG, part of Springer Nature, 2018.
[10]. Nguyen Thanh Tuan, "*Base Deep learning*", The Legrand Orange Book. Version 2, last update, August (2020).
[11]. Tran Hoai Linh, "*Neural network and its application in signal processing*", Hanoi Polytechnic Publishing House, (2015).
[12]. Nils J. Nilsson, "*The quest for artificial interlligence a history of ideas and achievements*", Web Version Print version published by Cambridge University Press, Publishing September 13, (2010).
[13]. Vu Thi Thuy Nga, Ong Xuan Loc, Trinh Hai Nam, "*Enhanced learning in automatic control with Matlab simulink",* Hanoi Polytechnic Publishing House, (2020).
[14]. V. Matt and N. Aran, "*Deep reinforcement learning approach to autonomous drivin*g," ed: arXiv, (2017).
[15]. https://*www.mathworks.com/help/reinfocerment-learning/ug/ddpg - Agent. html, (2020).*

ABSTRACT

AUTOMATIC NAVIGATION FOR MOBILE ROBOT TO AVOID OBSTACLES IN UNKNOWN ENVIRONMENT BASED ON Q-LEARNING ALGORITHM

This article presents a method to build an intelligent automatic navigation system for multi-directional mobile robots in an unknown flat environment. The studies using programming tools are the operating system for mobile robots (Robot Operating System - ROS). From updates on maps, operating environment, robot control position, and obstacles (Simultaneous Localization and Mapping (SLAM)) to calculating the trajectory of the robot in the navigation system automatically avoids the obstacles. The navigation system calculates the global and local orbit for the robot based on the application of Q-learning algorithm. The results of simulation studies in the Gazebo environment and especially the experimental run on the real Turtlebot mobile robot have shown the practical effectiveness of automatic navigation for this mobile robot.

Keywords: Artificial intelligence; Mobile robot control; Q-learning algorithm; Reinforcement learning.

Nhận bài ngày 15 tháng 4 năm 2021
Hoàn thiện ngày 13 tháng 8 năm 2021
Chấp nhận đăng ngày 05 tháng 9 năm 2021

Địa chỉ: [1]Khoa Điện, Trường Đại học Kinh tế - Kỹ thuật Công nghiệp/Bộ Công thương;
[2]Khoa Điện tử, Trường Đại học Kỹ thuật Công nghiệp Thái Nguyên.
[*]Email: tdchuyen@uneti.edu.vn; daohuydu@tnut.edu.vn.