MOBILE ROBOT CONTROL USING DEEP REINFORCEMENT LEARNING ALGORITHM

Pham Thi Thu Ha
University of Economics - Technology for Industries

ARTICLE INFO

Received: 08/6/2025
Revised: 26/11/2025
Published: 26/11/2025

KEYWORDS

Mobile robot
ROS
SLAM
Gazebo
Artificial intelligence

ABSTRACT

This study presents the problem of controlling mobile robots used in many fields: industry, medicine, transportation, civil, etc. Mobile robots meet the need for intelligent control for intelligent navigation in flat environments, nonlinear environments applying deep learning algorithms. The article uses the programming research method with the ROS robot operating system, combined with the implementation of automatic intelligent navigation for robots in the process of positioning robots in flat environments and uncertain - nonlinear environments. On that basis, this study applies to establish simultaneous mapping - SLAM. The research results using the ROS programming tool, in the Gazebo environment, have confirmed that the control algorithm is always updated from the map, operating environment, robot control position. All obstacles have been calculated trajectories for robots in automatic intelligent navigation, avoiding obstacles safely without encountering any obstacles in the moving journey. This study is meaningful in contributing to improving the automation efficiency and applicability of mobile robots in complex moving environments.

ĐIỀU KHIỂN ROBOT DI ĐỘNG ỨNG DỤNG THUẬT TOÁN HỌC SÂU TĂNG CƯỜNG

Phạm Thị Thu Hà
Trường Đại học Kinh tế - Kỹ thuật Công nghiệp

THÔNG TIN BÀI BÁO

Ngày nhận bài: 08/6/2025
Ngày hoàn thiện: 26/11/2025
Ngày đăng: 26/11/2025

TỪ KHÓA

Robot di động
Hệ điều hành ROS
Bản đồ hóa đồng thời
Mô phỏng môi trường ảo
Trí tuệ nhân tạo

TÓM TẮT

Nghiên cứu này trình bày về vấn đề điều khiển robot di động sử dụng trong nhiều lĩnh vực: công nghiệp, y tế, giao thông, dân dụng, v.v. Robot di động đáp ứng nhu cầu điều khiển thông minh để điều hướng thông minh trong môi trường phẳng, môi trường phi tuyến ứng dụng thuật toán học sâu tăng cường. Bài báo sử dụng phương pháp nghiên cứu lập trình với hệ điều hành robot ROS, kết hợp với việc thực hiện điều hướng thông minh tự động cho robot trong quá trình định vị robot trong môi trường phẳng, môi trường không xác định - phi tuyến. Trên cơ sở đó, nghiên cứu này ứng dụng thiết lập bản đồ hóa đồng thời - SLAM. Kết quả nghiên cứu sử dụng công cụ lập trình ROS, trong môi trường Gazebo đã khẳng định được thuật toán điều khiển luôn cập nhật từ bản đồ, môi trường hoạt động, vị trí điều khiển robot. Mọi vật cản đã được tính toán quỹ đạo cho robot trong việc điều hướng thông minh tự động, tránh vật cản một cách an toàn mà không gặp bất kỳ trở ngại nào trong hành trình di chuyển. Nghiên cứu này có ý nghĩa góp phần nâng cao hiệu quả tự động hóa và khả năng ứng dụng của robot di động trong các môi trường di chuyển phức tạp.

DOI: https://doi.org/10.34238/tnu-jst.13008

Email: pttha@uneti.edu.vn

1. Mở đầu

Ngày nay, robot thông minh đang được quan tâm nghiên cứu, ứng dụng, phát triển trong nhiều lĩnh vực như quân sự, công nghiệp, y tế, giao thông và ngay cả trong dân dụng. Robot di động đang được sử dụng nhiều trong cuộc sống hàng ngày của con người như robot phục vụ (robot hút bụi, robot lau nhà, robot đưa hàng), bên cạnh đó, robot còn được ứng dụng rộng rãi trong công nghiệp (robot tự hành trong các nhà máy sản xuất), robot hỗ trợ trong bệnh viện, robot trong lĩnh vực dò mìn, trong chiến trường, trong giao thông vận tải, v.v. Công nghệ robot nói chung, robot di động nói riêng là một lĩnh vực đa ngành gồm: cơ khí, điện - điện tử, công nghệ thông tin, người máy [1] – [3]. Robot thực hiện hành động, nhiệm vụ, khả năng điều khiển thông minh tự động ở môi trường phi tuyến bất định [4] – [6]. Việc làm này phụ thuộc vào các thiết bị và môi trường làm việc. Bản đồ hóa đồng thời diễn ra khi điều khiển robot với một số vị trí khác nhau; môi trường không gian hành động khác nhau. Nghiên cứu này thực hiện thuật toán học sâu tăng cường (Deep Reinforcement Learning - DRN) để thực thi, ứng dụng trí tuệ nhân tạo trong việc xây dựng thuật toán mới trong điều khiển robot di động.

Điều khiển tự động robot di động trên cơ sở các luật học sâu tăng cường DRN là một lĩnh vực con của học máy, nghiên cứu môi trường làm việc, thuật toán DRN nhằm thực thi các hành động, để đem đến những lợi ích về kinh tế cũng như trong điều khiển [7], [8]. Vấn đề thực thi trạng thái chuyển động ở thuật toán này được coi là ngẫu nhiên để thực hiện bài toán điều khiển robot [9] – [12]. Hai cách thức để ứng dụng giải quyết bài toán không gian hàm giá trị là "phép lặp chiến lược" và "phép lặp giá trị". Hai cách này chính là một số giải thuật học tăng cường được coi là đặc trưng, DRN được ứng dụng rộng phổ biến [7], [10], [13] – [15]. Quy trình DRN là một phương pháp học máy kết hợp giữa học tăng cường và mạng nơ-ron sâu. DRL sử dụng mạng nơ-ron sâu để xấp xỉ hàm giá trị hành động (Q-value), giúp tác nhân có thể đưa ra quyết định tối ưu trong môi trường. Deep Q-Learning (DQN) là một ví dụ điển hình, sử dụng mạng nơ ron để ước tính giá trị Q cho các hành động khác nhau trong một trạng thái. Quá trình này bao gồm việc quan sát phần thưởng và điều chỉnh dự đoán để cải thiện hiệu suất của tác nhân. Nghiên cứu [6] tìm hiểu về SLAM (Simultaneous Localization and Mapping), ROS (Robot Operating System), mới chỉ dừng lại với robot di động để thử nghiệm vấn đề ứng dụng thuật toán tìm đường đi tối ưu, mà vẫn không đi sâu vào tính toán thiết lập những vấn đề thực tế có nhiều yếu tố bất định. Các nghiên cứu [4], [11] sử dụng luật học tăng cường, chưa làm rõ những nhược điểm khi điều khiển robot di động. Nghiên cứu [5] thiết lập thuật toán trên hệ điều hành ROS, triển khai ứng dụng việc bản đồ hóa SLAM cho robot.

Bài báo này ứng dụng DRN để làm rõ vấn đề tính toán đường đi và tránh chướng ngại vật (môi trường phẳng, môi trường động, có nhiễu) cho robot. Hơn nữa, thuật toán này làm rõ nhiều vấn đề mà những nghiên cứu trước đây chưa thực hiện với thực tại của robot di động đang làm việc ở công nghiệp dân dụng: nhà máy, y tế, giao thông, môi trường không xác định mà robot vẫn thực hiện một cách an toàn trong cả hành trình di chuyển đến đích trọn vẹn.

2. Phương pháp nghiên cứu

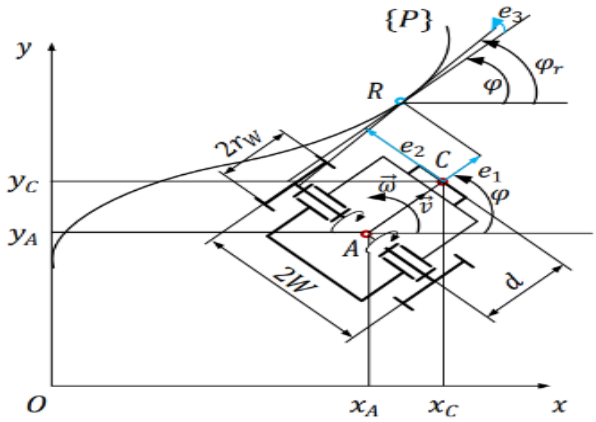
2.1. Xây dựng mô hình điều khiển cho robot di động

Từ mô hình, các thiết bị được cấu trúc như Hình 1, gồm: khung robot được thiết kế theo kiểu hình chữ nhật, hai bánh xe cách nhau là 0,35 m, bán kính là 0,065 m. Phương trình động học của robot di động:

$$v = v_A = \frac{(\omega_R - \omega_L) r_W}{2} \quad (1) \qquad \omega = \frac{(\omega_R - \omega_L) r_W}{2W} \quad (2)$$

Từ các biểu thức (1) và (2), ta có biểu thức chuyển động robot được viết ở hệ tọa độ tại *A* và tại *C* như sau:

$$\begin{bmatrix} \dot{x}_A \\ \dot{y}_A \\ \dot{\varphi} \end{bmatrix} = \begin{bmatrix} cos\varphi & 0 \\ sin\varphi & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (3)$$



Hình 1. *Cấu trúc mô hình động học robot di động ba bánh*

Ta lại có:

$$\begin{bmatrix}\dot{x}_C\\ \dot{y}_C\\ \dot{\varphi}\end{bmatrix}=\begin{bmatrix}\dot{x}_A-\dot{\varphi}dsin\varphi\\ \dot{y}_A-\dot{\varphi}dcos\varphi\\ \omega\end{bmatrix} \quad (4)$$

Việc thực hiện bám quỹ đạo chuyển động cho robot theo lý thuyết ổn định, hội tụ Lyapunov [1], [2], [4]. Khi đó phương trình sai số như sau:

$$\begin{bmatrix}\dot{e}_1\\ \dot{e}_2\\ \dot{e}_3\end{bmatrix}=\begin{bmatrix}v_r cos\, e_3\\ v_r sin\, e_3\\ \omega_r\end{bmatrix}+\begin{bmatrix}-1 & e_2\\ 0 & -d-e_1\\ 0 & -1\end{bmatrix}\begin{bmatrix}v\\ \omega\end{bmatrix} \quad (5)$$

Khi robot bám theo đường quỹ đạo chuyển động của robot di động; từ (5), điểm cân bằng của hệ phi tuyến chính là $X_0=[e_{10}\quad e_{20}\quad e_{30}]^T=[000]^T$ với đầu vào khởi tạo là $u_{sk0}=[v_r\quad \omega_r]^T$. Tuyến tính hóa (5) ta được:

$$\begin{bmatrix}\dot{e}_1\\ \dot{e}_2\\ \dot{e}_3\end{bmatrix}=\begin{bmatrix}0 & -\omega_r & -v_r\\ -\omega_r & 0 & v_r\\ 0 & 0 & 0\end{bmatrix}+\begin{bmatrix}e_1\\ e_2\\ e_3\end{bmatrix}+\begin{bmatrix}-1 & 0\\ 0 & -d\\ 0 & -1\end{bmatrix}\begin{bmatrix}v\\ \omega\end{bmatrix} \quad (6)$$

Trên thực tế, vận tốc chuyển động robot là không thay đổi $e_1\approx 0$. Lúc này, (6) viết là:

$$\begin{bmatrix}\dot{e}_1\\ \dot{e}_3\end{bmatrix}=\begin{bmatrix}0 & v_r\\ 0 & 0\end{bmatrix}\begin{bmatrix}e_1\\ e_3\end{bmatrix}+\begin{bmatrix}-d\\ -1\end{bmatrix}u, u=\omega \quad (7)$$

Biến đổi (7) và thực hiện đạo hàm, tính toán ta được:

$$\ddot{e}_2=-d.\dot{u}-v_r.u \quad (8)$$

Ta đặt: $x_1=e_2, x_2=\dot{x}_1-\beta.u$, với $\beta=-d$, ta có:

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\end{bmatrix}=\begin{bmatrix}0 & 1\\ 0 & 0\end{bmatrix}\begin{bmatrix}x_1\\ x_3\end{bmatrix}+\begin{bmatrix}-d\\ -v_r\end{bmatrix}u \quad (9)$$

Biểu thức (9) ở dạng $\dot{X}=A.X+B.U$ và ma trận $M=[B\ AB]$ có dạng $\det(M)\neq 0\ \forall\ v_r\neq 0$. Thành phần điều khiển $u=-K.X$ với $K=[k_1\ k_2]$ có hồi tiếp. Khai triển luật điều khiển ta có:

$$u=\frac{-k_1}{1+k_2 d}e_2+\frac{-k_2}{1+k_2 d}\dot{e}_2 \quad (10)$$

Luật (10) ở dạng điều khiển PD; ta có: $u=K_p e_2+K_d\dot{e}_2$ với $K_p=\frac{k_1}{1+k_2 d}, K_d=\frac{-k_2}{1+k_2 d}$. Hệ số k_1, k_2 xác định được dựa trên cơ sở phương pháp tọa độ điểm cực (-) vị trí hay điều khiển tối ưu bền vững. Tại một thời điểm vận tốc thiết bị dẫn động của robot di động tại (i) lần lượt là $C_A C_{(i)}$ và $C_A O_{(i)}$. Góc lệch giữa thiết bị với robot ở thời điểm (i) là:

$$\theta_{(i)}=\varphi_{(i)}-\varphi_{A_{(i-1)}} \quad (11)$$

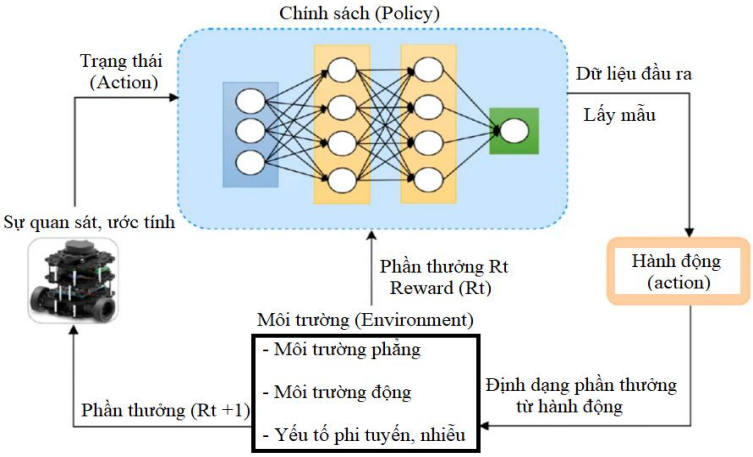
Vì thiết bị truyền động của hai bánh xe và thân robot có gắn kết truyền động tại O, do đó $v_D=v_A=v_0=v$. Từ đó ta có:

$$C_A O_{(i)} = \frac{L}{\sin(\theta_{(i)})} C_A C_{(i)} \quad (12) \qquad \omega_{A(i)} = \frac{v_{(i)}}{C_A O_{(i)}} = \frac{v_{(i)} sin(\theta_{(i)})}{L} \quad (13)$$

2.2. Nghiên cứu thuật toán DRN khiển robot di động

2.2.1. Thuật toán DRN

Học sâu tăng cường là một phần của thuật toán học máy, nhằm tiến bộ hơn về tính toán tối ưu cho robot di động. Thuật toán DRN với "trạng thái - s_t", "hành động - a_t" thời gian hiển thị bởi $(t = 0, 1, 2, v.v)$. Quá trình điều khiển luôn được cập nhật $S = \{S_1, S_1, \dots S_m\}$ cùng với lượng tử hóa được hành động. Khi đó, ta thiết lập $A = \{a_1, a_1, \dots a_n\}$, môi trường $r_t = r(s_t, a) \in R$. Từ đó, ta đưa ra cấu trúc huấn luyện như Hình 2.



Hình 2. *Sơ đồ cấu trúc về tổng quan học sâu tăng cường trong nghiên cứu này*

Trong DRN, thành phần định dạng phần thưởng là việc xác định cách thức mà một hệ thống hoặc mô hình học tập huấn luyện sẽ nhận được phản hồi về hành động. Vấn đề này liên quan việc đưa ra yếu tố, tiêu chí, và cách thức đánh giá một tác nhân - Agent trong một môi trường nhất định. Từ đó xác định phần thưởng - reward hoặc hình phạt - Penalty tương ứng [3].

Ở đây, ta có tính toán R_t đem lại là:

$$R_t = E[r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \cdots] = E[\sum_{k+1}^{\infty} \gamma^k r_{t+k}] \quad (14)$$

Với $0 \leq \gamma < 1$ là điều kiện của phần thưởng. Khi γ nhỏ thì phần thưởng sẽ được cập nhật. Lúc này, Q được tính toán:

$$Q^2(s, a) = E_\pi\{R_t | s_t = s, a_t = a\} = E_\pi[\sum_{k=0}^{\infty} \gamma^k r_{t+k} | s_t = s, a_t = a] \quad (15)$$

Với $E_\pi\{\dots\}$ là kỳ vọng chính sách ngẫu nhiên. Thành phần $Q^n(s, a)$ chọn hành động a, trạng thái s, hàm π. Với Q được viết ở dạng biểu thức đệ quy:

$$\begin{aligned} Q_{i+1}^{\pi}(s, a) &= E_\pi[r_t + \gamma \sum_{k=0}^{\infty} \gamma^k r_{t+k} | s_t = s, a_t = a] \\ &= E_\pi[r_t + \gamma Q_i^{\pi}(s_{t+1} = s', a_{t+1} = a') | s_t = s, a_t = a] \end{aligned} \quad (16)$$

Với $s, s' \in S$ trạng thái và $a, a' \in A$ tập hành động. Từ đó, ta có thể xác định rằng hàm Q ∈ π^*, tức là hàm Q tối ưu, thỏa mãn phương trình tối ưu Bellman:

$$Q_{i+1}^*(s, a) = E_\pi\{r_t + \gamma max_{a'} Q_i^* | s', a'\} \quad (17)$$

Thuật toán DRN, hàm Q được tính toán từ biểu thức (10), hàm Q hội tụ ngẫu nhiên thành $Q_i^*(s, a)$ và $Q_i^*: a^* = argmax_{a'}(s_t, a_t)$.

Với $0 < \alpha \leq 1$ là điều kiện quá trình huấn luyện. Khi đó, tốc độ huấn luyện càng cao thì dữ liệu sẽ luôn được cập nhật mới hơn.

Trong phương pháp DRN, ta có $Q(s, a; \theta)$. Việc cập nhật tham số theo biểu thức được viết:

$$L(\theta) = E_\pi[r + \gamma max_{a'} Q(s', a', \theta) - Q(s', a', \theta))^2] \quad (18)$$

Với $r + \gamma max_{a'}Q(s', a', \theta)$ là thành phần mục tiêu phụ thuộc vào (10). Sự phân biệt của hàm mất mát đối với các tham số θ của nó dẫn đến độ dốc sau:

$$\frac{\partial L(\theta)}{\partial \theta} = E[(r + \gamma max_{a'}Q(s', a', \theta) - Q(s, a, \theta))]\frac{\partial (s,a,\theta)}{\partial \theta} \quad (19)$$

Thuật toán DRN cho phép tính toán, xấp xỉ ứng dụng mạng nơ ron ba lớp để đem lại dữ liệu đầu ra chính xác. Từ đó lấy thành phần tối ưu để đưa đến thành phần Actor. Đây là cách áp dụng các phương pháp dựa trên Phương pháp gradient để tối ưu hóa mạng nơ ron. Thông thường, phương pháp này trên được tối ưu hóa các thành phần ngẫu nhiên (là một thuật toán tối ưu hóa được sử dụng để tìm giá trị nhỏ nhất của một hàm số). Phương pháp này được sử dụng rộng rãi trong học máy và học sâu để đào tạo các mô hình bằng cách điều chỉnh lặp đi lặp lại các tham số theo hướng của độ dốc âm. Hàm xấp xỉ có thể là hàm tuyến tính hoặc hàm phi tuyến tính (ở đây đối với DRN là mạng nơ ron) của các tham số θ.

2.2.2. Điều khiển chuyển động robot di động ứng dụng DRN

Thuật toán DRN điều khiển robot di động theo cách thức vi phân; ở đây dùng sai lệch của một bước lặp để tính toán, ước tính hàm Q theo (17). Vấn đề điều khiển robot thực thi với các tính năng thay đổi liên tục khác nhau, với việc hoạt động (trái, phải), tránh chướng ngại vật (vật cản tĩnh, vật cản động), v.v. Khi đó, ta chọn tham số α = 0,1; γ = 0,98 cho việc xác định tính toán.

Thuật toán: Thuật toán học sâu tăng cường cho robot di động (dạng tiếng việt)

Initialize: Khởi tạo trọng số của mạng là θ, θ'
Đặt các giá trị bộ nhớ, kích thước bộ đệm, dữ liệu huấn luyện mạng,…
Đặt ngưỡng khoảng cách va chạm d (đến), d (va chạm) cho tập = 1, M để thực hiện hành động.
Đặt robot Turtlebot vào vị trí bắt đầu
Lấy cường độ tối thiểu của hình ảnh độ sâu dưới dạng d_t
While $d_t > d_{đến}$ hoặc $d_t > d_{va\ trạm}$
  Chụp ảnh độ sâu s_t; Với xác suất ε chọn một hành động ngẫu nhiên a_t.
  Nếu không, lựa chọn $a_t = argmax_a Q(s_t, a, \theta)$; Di chuyển bằng lệnh di chuyển đã chọn a_t
  Cập nhật d_t bằng thông tin độ sâu mới a_t ; If $d_t < d$ đến thì $r(s_t, a_t) = r_{đến}$
  else if $dt > d_{đến}$ hoặc $d_t < d_{va\ trạm}$ thì $r(s_t, a_t) = r_{va\ trạm}$
  else $r(s_t, a_t) = c(d_{t-1} - d_t)$
  Chụp ảnh độ sâu mới s_{t+1}
  end if
  Lưu trữ chuyển tiếp (s_t, a_t, r_t, s_{t+1}) trong D; Thay thế bộ dữ liệu cũ nhất nếu $|D| \geq Nr$
  Chọn ngẫu nhiên một loạt Nb chuyển tiếp (s_k, a_k, r_k, s_{k+1}) từ D
  Cho mỗi lần chuyển đổi: $a^{max}(s_{k+1}; \theta) = argmax_{a'} Q(s_{k+1}, a'; \theta)$
    Nếu $r_k = r_{đến}$ hoặc $r_k = r_{va\ chạm}$ hoặc $r_k = r_{hết\ thời\ gian}$ thì $y_k = r_k$
  else
    $y_k = r_k + \gamma Q(s_{k+1}, a^{max}(s_{k+1}; \theta); \theta')$
  end if
  Cập nhật thông qua một gradient với mất mát $(y_k - Q(s, a; \theta))^2$
  Thay thế các tham số mục tiêu θ′ bằng θ sau mỗi bước Nrf
 end while
end for

Hình 3. *Thuật toán học sâu tăng cường cho robot di động*

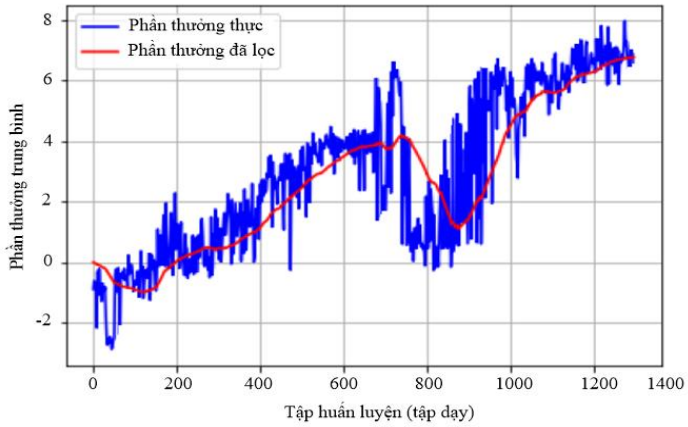
Mục tiêu việc điều hướng cho robot là di chuyển đi lại trong một thời gian cho phép, tức là ± 5 độ. Ban đầu, mô hình robot, ma trận Q, với π sẽ được khởi tạo vào lập trình điều khiển.

Thực hiện thuật toán DRN, việc sắp xếp các giá trị trạng thái thành 20 góc trạng thái từ -10 độ đến 10 độ. Đối với giá trị hành động, ta đã chọn mười giá trị vận tốc là [- 300, -200, - 100, - 50, - 25, - 10, 10, 25, 50, 100, 200, 300] ms^{-1}. Ma trận Q có 20 cột, mỗi cột đại diện cho một trạng thái và mười hàng, mỗi hàng đại diện cho một hành động. Ban đầu, các giá trị Q được giả định là 0, với một việc làm, hay hoạt động để thực hiện cho trạng thái trong chính sách π. Ở đây, nghiên cứu huấn luyện trong 1400 tập, mỗi tập có 2500 lần lặp lại. Với một trạng thái đầu vào, mỗi tập

dạy sẽ 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 (17), (18). Khi đó, thuật toán DRN được thực thi quá trình di chuyển quỹ đạo theo việc điều hướng thông minh cho robot di động như Hình 3. Với DRN, việc thực hiện những tác nhân, hành động, cho việc điều hướng robot nhằm bám quỹ đạo chuyển động là tối ưu sao cho robot làm việc tin cậy an toàn trong suốt quá trình dịch chuyển.

3. Kết quả nghiên cứu

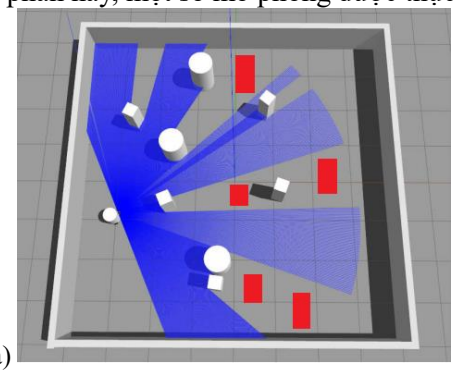
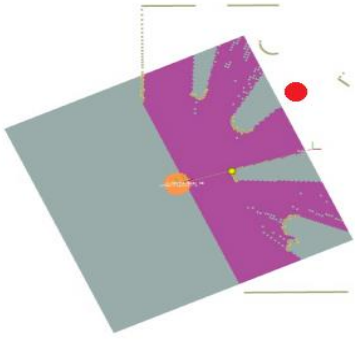
Từ những nghiên cứu thuật toán ở trên tiến hành thực hiện mô phỏng trên Matlab, ta có một số các kết quả như sau:



Hình 4. *Hình ảnh phần thưởng trung bình của quá huấn luyện*

Hình 4 mô tả việc thực thi cập nhật luật học ở môi trường mô phỏng Gazebo. Khi đó, phần thưởng thực không ngừng tăng lên theo quá trình đào tạo huấn luyện, phần thưởng đã lọc luôn hội tụ bám sát theo phần thưởng thực. Robot luôn được huấn luyện kiến thức về môi trường thông qua việc dạy trước đó với môi trường phẳng và môi trường phức tạp. Nghiên cứu thử nghiệm cho thấy tính hiệu quả của mô hình mà tác giả đã đề xuất đạt chất lượng bám cao hơn so với thuật toán thông thường khác.

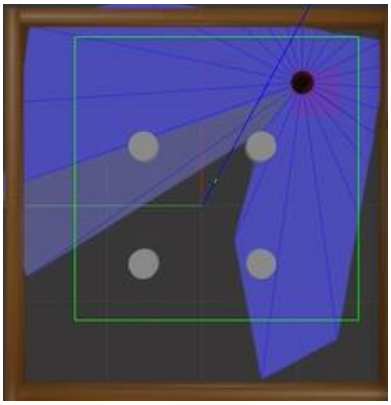
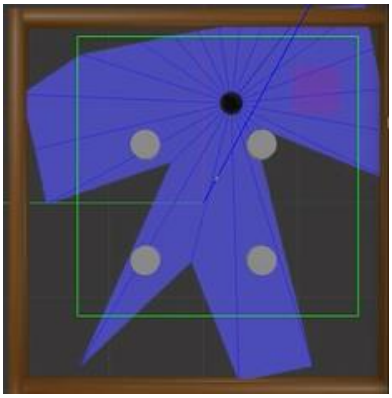
Trong phần này, một số mô phỏng được thực hiện với môi trường trong Gazebo.

a)  b) 

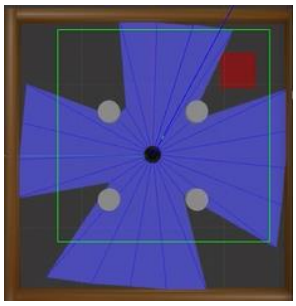
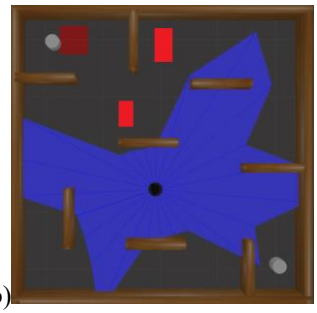
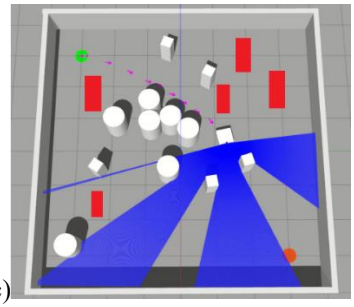
Hình 5. *a) Môi trường huấn luyện ở Gazebo; b) Bản đồ hóa tỷ lệ tương ứng thực hiện ở Rviz*

Ở Hình 5, bản đồ hóa đồng thời được xây dựng trên Gazebo và trong Rviz là bản đồ được tạo ra với phần khung chia rõ ràng riêng biệt.

Hình 6a cho thấy robot đang di chuyển từ vị trí ban đầu đến đích, trong khi Hình 6b mô tả quá trình robot di chuyển để tránh chướng ngại vật. Hình 7a thể hiện robot di động với quá trình huấn luyện trong môi trường ảo Gazebo. Hình 7b mô tả robot tìm đường đi tại vị trí trung tâm, có vật cản.

a)  b) 

Hình 6. *Trình tự hình ảnh trong môi trường đầu tiên thực hiện thuật toán DRN: a) Khi robot di chuyển từ vị trí xuất phát; b) Khi robot di chuyển tránh chướng ngại vật*

a)  b)  c) 

Hình 7. *a) Thực hiện hành động huấn luyện ở Gazebo, b) Bản đồ trực quan và đường đi cho robot di động trong Gazebo, c) đường đi của robot di động*

Ở Hình 7c, đường di chuyển của robot được biểu diễn bằng nét đứt, thể hiện đường đi của robot vượt qua vật cản được tạo bởi cảm biến thông minh. Vị trí chuyển động của robot được thực thi (màu xanh lá cây) ở đây sử dụng kích thước đo hình học, môi trường Gazebo. Đây là một công cụ trực quan nhằm cung cấp, cập nhật online với thiết bị máy tính và robot, bản đồ hóa thực hiện từ thuật toán SLAM để điều khiển robot. Việc bám quỹ đạo của robot để điều hướng và tránh vật cản ở Hình 5, Hình 6 và Hình 7 đã thể hiện robot xác định chính xác quỹ đạo và di chuyển đến đích một cách an toàn và chính xác. Những phát hiện này mang lại giá trị thực tiễn lớn, có thể áp dụng được một số ứng dụng như robot trong nhà xưởng, robot trong giao thông, y tế, cũng như trong các nhà máy, xí nghiệp công nghiệp và các ứng dụng dân dụng như robot hút bụi, robot lau nhà.

4. Kết luận

Kết quả nghiên cứu của bài báo đã chỉ ra việc điều khiển robot di động có xét đến yếu tố phi tuyến trong môi trường động phức tạp với một robot Turtlebot và hệ điều hành ROS là công cụ đắc lực để lập trình điều khiển. Việc mô phỏng trên Gazebo đã minh chứng khả năng làm việc của robot di động; có thể đi đến các vị trí mục tiêu bất kỳ và tránh được các vật cản tĩnh và vật cản động trong suốt quá trình di chuyển trong môi trường phẳng và môi trường động. Nghiên cứu này cho thấy robot di động luôn bám sát được quỹ đạo chuyển động, di chuyển đến đúng mục tiêu, vượt qua được hoàn toàn chướng ngại vật xuất hiện trên đường đi một cách an toàn. Trong hướng phát triển tiếp theo, nhóm nghiên cứu sẽ triển khai thử nghiệm trên robot thật, đồng thời tích hợp các thuật toán AI nâng cao nhằm tối ưu khả năng học tập và tự thích nghi của hệ thống.

TÀI LIỆU THAM KHẢO/ REFERENCES

[1] P. D. Nguyen, *Advanced Control Theory*, Science and Engineering Publishing House, (in Vietnamese), Hanoi, Vietnam, 2018.

[2] C. Q. Hoang, V. H. Dao, V. A. Nguyen, and C. B. Le, *Electric drive systems in Robots*. People's Army Publishing House, (in Vietnamese), Hanoi, Vietnam, 2020.

[3] T. T. N. Vu, X. L. Ong, and H. N. Tran, *Enhanced learning in automatic control with Matlab simulink*, Hanoi Polytechnic Publishing House, (in Vietnamese), Hanoi, Vietnam, 2020.

[4] S. H. Le, D. C. Le, and H. V. Nguyen, *Industrial Robots Syllabus*, Ho Chi Minh City National University Publishing House, (in Vietnamese), Ho Chi Minh City, Vietnam, 2017.

[5] L. Joseph and J. Cacace, *Mastering ROS for Robotics Programming, vol. 2: Design, build, and simulate complex robots using the Robot Operating System,* Packt Publishing Ltd., UK., 2018.

[6] F. Guo, H. Yang, and X. Wu, "Model-based deep learning for low-cost IMU dead reckoning of wheeled mobile robot," *IEEE Trans. Ind. Electron.*, no. 1, pp. 7531–7541, 2023, doi: 10.1109/TIE.2023.3301531.

[7] Y. Li, "Deep reinforcement learning," in *ICASSP 2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, Calgary, AB, Canada, 2018, pp. 15-20.

[8] S. P. Thale, "ROS based SLAM implementation for Autonomous navigation using Turtlebot," *ITM Web of Conferences*, vol. 32, no. 5, 2020, Art. no. 01011, doi: 10.1051/itmconf/20203201011.

[9] H. X. Dong, C. Y. Weng, C. Q. Guo, and H. Y. Yu, "Real-time avoidance strategy of dynamic obstacles via half model-free detection and tracking with 2D Lidar for mobile robots," *IEEE/ASME Trans. on Mechatronics*, vol. 26, no. 4, pp. 2215 – 2225, August 2021.

[10] R. K. E. A. Megalingam, "ROS based autonomous indoor navigation simulation using SLAM algorithm," *Int. J. Pure Appl*., vol. 7, pp. 199-205, 2018.

[11] D. Kozlov, "Comparison of Reinforcement Learning Algorithms for Motion Control of an Autonomous Robot in Gazebo Simulator," *International Conference on Information Technology and Nanotechnology (ITNT*), IEEE Explore, 2021, pp. 1-5, doi: 10.1109/ITNT52450.2021.9649145.

[12] H. T. Tran and T. T. H. Pham, "Controlling mobile robot in flat environment taking into account nonlinear factors applying artificial intelligence," *Bulletin of Electrical Engineering and Informatics*, vol. 13, no. 5, pp. 3737-3745, October 2024, doi: 10.11591/eei.v13i5.7818.

[13] H. Lee, J. Kim, and J. Lee, "Resource Allocation in Wireless Networks with Deep Reinforcement Learning: A Circumstance-Independent Approach," *IEEE Syst. J.*, vol. 14, pp. 2589–2592, 2020.

[14] K. M. Othman and A. B. Rad, "A Doorway Detection and Direction (3Ds) System for Social Robots via a Monocular Camera," *Sensors*, vol. 20, pp. 2477-2489, 2020.

[15] K. Nolan, "Optitrack," 2025. [Online]. Available: https://www.optitrack.com/. [Accessed June 15, 2025].