

	VIETTEL AI RACE	Public 108
	<b>NGHIÊN CỨU TỐI ƯU BÀI TOÁN ĐỊNH VỊ BẢN ĐỒ CHO ROBOT DI ĐỘNG TRONG MÔI TRƯỜNG KHÔNG XÁC ĐỊNH SỬ DỤNG PHƯƠNG PHÁP HỌC TĂNG CƯỜNG</b>	Lần ban hành: 1

## 1. Giới thiệu

Bài báo này đề xuất phương pháp định vị và tránh vật cản cho robot di động hoạt động trong môi trường đa vật thể dựa trên thuật toán học tăng cường. Mô hình robot di động gồm đầy đủ các thông số hình học và thông số vật lý được xây dựng trên nền tảng phần mềm Gazebo. Các hoạt động huấn luyện cho mô hình để robot tự tìm đường di chuyển được thực hiện cho cả thuật toán Q-Learning và thuật toán SARSA. Kết quả thử nghiệm được so sánh giữa hai thuật toán để đánh giá hiệu quả và chất lượng của các hoạt động huấn luyện.

## 2. Cơ sở lý thuyết

Phương pháp học tăng cường tập trung vào việc học hướng tới mục tiêu từ sự tương tác khác nhau. Thực thể thực hiện quá trình học tập sẽ không biết trước hành động cần phải thực hiện, thay vào đó phải tự khám phá ra hành động nào mang lại phần thưởng lớn nhất bằng cách kiểm tra các hành động này thông qua phương pháp thử sai. Các thành phần cơ bản trong học tăng cường bao gồm:

- Tác nhân (Agent): đóng vai trò trong việc giải quyết các vấn đề ra quyết định, tác động dưới sự không chắc chắn.
- Môi trường (Environment): là những gì tồn tại bên ngoài tác nhân, tiếp nhận các tác động từ tác nhân và tạo ra phần thưởng và những quan sát.
- Hành động (Actions): tập hợp các phương thức hành động mà tác nhân tác động đến môi trường.
- Trạng thái (State): trạng thái của tác nhân sau khi tác động qua lại với môi trường.
- Phần thưởng (Reward): là giá trị thu được tương ứng với mỗi cặp Trạng thái - Hành động của tác nhân nhận được khi thực hiện tương tác với 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): là hàm biểu diễn sự tương quan giữa những quan sát thu được từ môi trường và hành động cần thực hiện.

Trong đó, tác nhân và môi trường là hai thành phần cốt lõi của một mô hình học tăng cường. Hai thành phần này tương tác liên tục với nhau theo trình tự: Tác nhân thực hiện các tương tác tới môi trường thông qua các hành động, từ đó môi

	VIETTEL AI RACE	Public 108
	<b>NGHIÊN CỨU TỐI ƯU BÀI TOÁN ĐỊNH VI BẢN ĐỒ CHO ROBOT DI ĐỘNG TRONG MÔI TRƯỜNG KHÔNG XÁC ĐỊNH SỬ DỤNG PHƯƠNG PHÁP HỌC TĂNG CƯỜNG</b>	Lần ban hành: 1

trường tác động lại các hành động của tác nhân. Môi trường lưu trữ các luồng thông tin khác nhau và phản hồi cho tác nhân một “giá trị khen thưởng” sau mỗi hành động của tác nhân. Giá trị này biểu hiện mức độ hiệu quả từng hành động của tác nhân trong quá trình hoàn thành nhiệm vụ. Mục đích của phương pháp học tăng cường là tác nhân tìm ra được chính sách tối đa hoá giá trị phần thưởng tích lũy trong thời gian dài. Trong hướng tiếp cận của bài báo, tác giả chỉ ra tính hiệu quả của phương thức triển khai mô hình đề xuất dựa trên hai thuật toán học tăng cường Q-Learning và SARSA.

## 2.1. Thuật toán Q-Learning

Q-Learning là một thuật toán học tăng cường thực hiện phương thức cập nhật giá trị (values-based) dựa trên cập nhật hàm giá trị từ phương trình Bellman [14]. Phương trình Bellman tính toán giá trị kỳ vọng của trạng thái như sau:

$$V^*(s_t, a_t) = \max_a Q^\pi(s_t, a_t)$$

(1)

- Trong đó:  $V^*(s_t)$  là giá trị tối ưu trả về từ giá trị kỳ vọng theo trạng thái  $s_t$  theo chính sách thực hiện  $\pi$ ;  $\max_a Q^\pi$  là giá trị Q lớn nhất thể hiện hành động  $a_t$  tại trạng thái  $s_t$  theo chính sách  $\pi$ .

Phương trình tính toán giá trị Q kỳ vọng thực hiện một hành động  $a_t$  tại trạng thái  $s_t$  dựa trên phương trình Bellman:

$$Q^*(s_t, a_t) = r_t + \gamma \max_a Q^*(s_{t+1}, a)$$

(2)

- Trong đó:  $Q^*(s_t, a_t)$  là giá trị kỳ vọng của phần thưởng mà phương trình hướng đến nhằm tối ưu cho mỗi cặp trạng thái  $s_t$  và hành động  $a_t$  tại thời điểm  $t$ ;  $r_t$  là phần thưởng tức thời nhận lại được tại thời điểm  $t$ ;  $\gamma$  là hằng số chiết khấu xác định mức độ quan trọng được trao cho phần thưởng hiện tại và phần thưởng trong tương lai;  $\max_a Q^*(s_{t+1}, a)$  là giá trị kỳ vọng lớn nhất có thể xảy ra của Q tại trạng thái  $s_{t+1}$  với mọi hành động  $a$ .

Q-Learning là một thuật toán Off-policy, quá trình học của mô hình chủ yếu dựa trên giá trị của chính sách tối ưu và độc lập với các hành động của chủ thể. Off-policy được định nghĩa là tác nhân tuân theo một chính sách quyết định

	VIETTEL AI RACE	Public 108
	<b>NGHIÊN CỨU TỐI ƯU BÀI TOÁN ĐỊNH VI BẢN ĐỒ CHO ROBOT DI ĐỘNG TRONG MÔI TRƯỜNG KHÔNG XÁC ĐỊNH SỬ DỤNG PHƯƠNG PHÁP HỌC TĂNG CƯỜNG</b>	Lần ban hành: 1

cho việc lựa chọn hành động để đạt trạng thái  $s_{t+1}$  từ trạng thái  $s_t$ . Kể từ trạng thái  $s_{t+1}$ , tác nhân sử dụng một chính sách khác cho khâu quyết định này.

Phương trình của thuật toán Q-Learning được trình bày như sau:

$$Q_{s_t,a_t} = Q_{s_t,a_t} + \alpha * [r_t + \gamma * \max_a Q(s_{t+1}, a) - Q_{s_t,a_t}]$$

$Q^*(s, a)$  trong (3) là giá trị kỳ vọng (phần thưởng của chiết khấu tích lũy trong việc thực hiện hành động  $a$  ở trạng thái  $s$  và sau đó tuân theo chính sách tối ưu. Hành động từ mỗi trạng thái thu được của thuật toán Q-Learning được xác định bởi quy trình ra quyết định Markov (MDP) [15, 16]. Các bước triển khai thuật toán được trình bày như trong bảng 1.

	VIETTEL AI RACE	Public 108
	NGHIÊN CỨU TỐI ƯU BÀI TOÁN ĐỊNH VI BẢN ĐỒ CHO ROBOT DI ĐỘNG TRONG MÔI TRƯỜNG KHÔNG XÁC ĐỊNH SỬ DỤNG PHƯƠNG PHÁP HỌC TĂNG CƯỜNG	Lần ban hành: 1

Bảng 1. Thuật toán cập nhật Q-Learning.

<p><b>Đầu vào:</b></p> <p>Tập trạng thái <math>S = \{1, 2, \dots, s_n\}</math> ;</p> <p>Tập hành động <math>A = \{1, 2, \dots, a_n\}</math> ;</p> <p>Hàm phần thưởng: <math>S \times A \rightarrow [ ]</math></p> <p>Khởi tạo các siêu tham số của thuật toán: <math>\alpha, \gamma \in [0; 1]</math> ;</p>
<p><b>Phương thức:</b></p> <p>Khởi tạo Q: <math>S \times A \rightarrow [ ]</math> ngẫu nhiên;</p> <p><b>for</b> giá trị Q chưa hội tụ:</p> <p>    <b>do:</b> đặt trạng thái <math>s_t \in S</math> ;</p> <p>    <b>for</b> s không phải là trạng thái cuối:</p> <p>        <b>do:</b></p> <p>            Lựa chọn hành động a mới (dựa vào chính sách tối ưu);</p> <p>            Thực hiện hành động a;</p> <p>            Quan sát trạng thái s mới và thu nhận phần thưởng R;</p> <p>            Cập nhật;</p> <p>            <math>Q_{st,at} = Q_{st,at} + \alpha * [r_t + \gamma * \max Q(st+1, a) - Q_{st,at}]</math></p> <p>            Cập nhật trạng thái <math>s' \leftarrow s</math></p> <p>        <b>End for</b></p> <p>    <b>End for</b></p>
<p><b>Đầu ra:</b></p> <p>Hành động tốt nhất được lựa chọn <math>a'</math> .</p>

## 2.2. Thuật toán SARSA

Tương tự Q-Learning, SARSA là một thuật toán học tăng cường tuân thủ theo phương thức cập nhật Value-based và được tính toán dựa trên phương trình Bellman. Tuy nhiên, SARSA là một thuật toán On-policy. Thuật toán On-policy là thuật toán đánh giá và cải thiện cùng một chính sách  $\pi$ , hay nói cách khác tác nhân học và tuân theo một chính sách duy nhất xuyên suốt quá trình đào tạo.

SARSA là một thuật toán chỉ định rằng tại trạng thái thời điểm  $s_t$ , thực hiện hành động  $a_t$ , tiếp đó phần thưởng  $r_t$  được nhận lại và kết thúc với trạng

	VIETTEL AI RACE	Public 108
	<b>NGHIÊN CỨU TỐI ƯU BÀI TOÁN ĐỊNH VI BẢN ĐỒ CHO ROBOT DI ĐỘNG TRONG MÔI TRƯỜNG KHÔNG XÁC ĐỊNH SỬ DỤNG PHƯƠNG PHÁP HỌC TĂNG CƯỜNG</b>	Lần ban hành: 1

thái  $s_{t+1}$ , đồng thời thực hiện hành động  $a_{t+1}$ . Do đó, chuỗi giá trị  $(s_t, a_t, r_t, s_{t+1}, a_{t+1})$  đại diện cho chính tên gọi của thuật toán. Điểm khác biệt duy nhất là thành phần  $Q(s_{t+1}, a_{t+1})$ , thay vì tối đa hoá cập nhật dựa trên giá trị  $Q$  kỳ vọng cao nhất  $\max Q(s_{t+1}, a)$  trong bảng giá trị kinh nghiệm như Q-Learning. SARSA được thiết lập thêm bước cập nhật hành động tại thời điểm kế tiếp. Phương trình cơ bản của thuật toán SARSA được trình bày như trong (4) và các bước triển khai thuật toán được mô tả trong bảng 2.

$$Q_{st,at} = Q_{st,at} + \alpha[r_t + \gamma Q(s_{t+1}, a_{t+1}) - Q_{st,at}]$$

	VIETTEL AI RACE	Public 108
	NGHIÊN CỨU TỐI ƯU BÀI TOÁN ĐỊNH VỊ BẢN ĐỒ CHO ROBOT DI ĐỘNG TRONG MÔI TRƯỜNG KHÔNG XÁC ĐỊNH SỬ DỤNG PHƯƠNG PHÁP HỌC TĂNG CƯỜNG	Lần ban hành: 1

Bảng 2. Thuật toán cập nhật SARSA

<p><b>Đầu vào:</b></p> <p>Tập trạng thái <math>S = \{1, 2, \dots, s_n\}</math>;</p> <p>Tập hành động <math>A = \{1, 2, \dots, a_n\}</math>;</p> <p>Hàm phần thưởng: <math>S \times A \rightarrow [ ]</math></p> <p>Khởi tạo các siêu tham số của thuật toán: <math>\alpha, \gamma \in [0; 1]</math></p>
<p><b>Phương thức:</b></p> <p>Khởi tạo <math>Q: S \times A \rightarrow [ ]</math> ngẫu nhiên;</p> <p><b>for</b> giá trị <math>Q</math> chưa hội tụ, <b>do</b>:</p> <p>    Đặt trạng thái <math>s_t \in S</math> ;</p> <p>    Thực hiện hành động <math>a_t \in A</math> (dựa vào chính sách tối ưu);</p> <p>    <b>for</b> <math>s</math> không phải là trạng thái cuối, <b>do</b>:</p> <p>        Lựa chọn hành động <math>a_{t+1} \in A</math> (dựa vào chính sách tối ưu);</p> <p>        Thực hiện hành động <math>a_{t+1}</math>;</p> <p>        Quan sát trạng thái <math>s</math> mới và thu nhận phần thưởng <math>R</math>;</p> <p>        Cập nhật;</p> <p>        <math display="block">Q_{st,at} = Q_{st,at} + \alpha[r_t + \gamma Q(s_{t+1}, a_{t+1}) - Q_{st,at}]</math></p> <p>        Cập nhật trạng thái <math>s_{t+1} \leftarrow s_t</math>; <math>a_{t+1} \leftarrow a_t</math> ;</p> <p>    <b>End for</b></p> <p><b>End for</b></p>
<p><b>Đầu ra:</b></p> <p>Hành động tốt nhất được lựa chọn <math>a_{t+1}</math>.</p>