NGHIÊN CỨU, THIẾT KẾ BỘ ĐIỀU KHIỂN CÁNH TAY ROBOT 6 BẬC TỰ DO DỰA TRÊN PHƯƠNG PHÁP HỌC TĂNG CƯỜNG MỀM ACTOR-CRITIC

RESEARCH AND DESIGN OF 6-DOF ROBOT ARM CONTROLLER BASED ON SOFT ACTOR-CRITIC REINFORCEMENT LEARNING METHOD

Nguyễn Quý Dương, Đinh Sỹ Thông, Nguyễn Anh Dũng, ĐặngThái Việt*
Trường Cơ khí, Đại học Bách Khoa Hà Nội
**Email: viet.dangthai@hust.edu.vn*

TÓM TẮT

Điều khiển chính xác cánh tay robot trong môi trường động, bất định là thách thức lớn đối với các bộ điều khiển kinh điển và các phương pháp hiện đại như điều khiển thích nghi hay điều khiển tối ưu. Mặc dù học tăng cường sâu (DRL) đã mở ra hướng đi mới, nhưng vẫn bộc lộ hạn chế về tính hội tụ cục bộ và thiếu sự cân bằng trong chiến lược khám phá không gian. Để giải quyết triệt để vấn đề này, bài báo đề xuất một bộ điều khiển tiên tiến cho cánh tay robot 6 bậc tự do (6-DOF) dựa trên phương pháp học tăng cường mềm Actor-Critic (Soft Actor-Critic - SAC). Trong nghiên cứu này, hệ thống trạng thái trước tiên được thiết lập các phương trình động học, động lực học và mô hình hóa dưới dạng quá trình quyết định Markov (MDP). Không gian trạng thái, hàm phần thưởng đa mục tiêu và cấu trúc mạng nơ-ron Actor-Critic được thiết kế nhằm tối đa hóa phần thưởng tích lũy kết hợp với cực đại hóa entropy. Cơ chế này giúp robot khám phá môi trường hiệu quả, tránh rơi vào các điểm tối ưu cục bộ và bám quỹ đạo trơn tru. Để kiểm chứng, bộ điều khiển được áp dụng vào bài toán thực nghiệm động lực học phức tạp: điều khiển robot dự đoán và đánh trúng quả bóng bàn với các quỹ đạo rơi ngẫu nhiên. Kết quả kỳ vọng cho thấy thuật toán SAC vượt trội về tốc độ hội tụ, độ ổn định và khả năng phản xạ theo thời gian thực, khẳng định tiềm năng to lớn trong việc giải quyết các tác vụ đa nhiệm phức tạp.

Từ khóa: *Cánh tay robot; Mô hình học tăng cường, Điều khiển dự đoán; Phương pháp học tăng cường mềm Actor-Critic (SAC).*

ABSTRACT

Precisely controlling a robotic arm in a dynamic, uncertain environment is a major challenge for classical controllers and modern methods such as adaptive or optimal control. Although deep reinforcement learning (DRL) has opened new avenues, but still exhibited limitations in local convergence and a lack of balance in spatial exploration strategies. To thoroughly address this problem, this paper proposes an advanced controller for a 6-DOF robotic arm based on the Soft Actor-Critic (SAC) reinforcement learning method. In this study, the state system is first established with kinematic and dynamic equations and modeled as a Markov decision process (MDP). The state space, multi-objective reward function, and Actor-Critic neural network structure are designed to maximize cumulative reward combined with maximizing entropy. This mechanism helps the robot explore its environment efficiently, avoiding local optimal points and maintaining a smooth trajectory. To verify this, the controller was applied to a complex dynamics experiment: controlling the robot to predict and hit a table tennis ball with random falling trajectories. The expected results showed that the SAC algorithm excelled in convergence speed, stability, and real-time responsiveness, confirming its enormous potential in solving complex multitasking problems.

Keywords: *Robot Arms; Reinforcement learning model; Predictive control method; Soft Actor-Critic Method (SAC).*

1. ĐẶT VẤN ĐỀ

Trong bối cảnh công nghiệp 4.0 và sự phát triển mạnh mẽ của các hệ thống tự động hóa, trí tuệ nhân tạo (AI) và cụ thể là học máy (Machine Learning) đang đóng vai trò then chốt trong việc định hình lại các phương pháp dẫn đường và điều khiển thông minh [1]. Đối với các hệ thống robot và máy móc tự hành, khả năng thích ứng linh hoạt trong môi trường phức tạp là một yêu cầu cấp thiết mà các bộ điều khiển kinh điển khó có thể đáp ứng trọn vẹn. Việc tích hợp các nền tảng học máy cho phép robot không chỉ thực thi các quỹ đạo được lập trình sẵn mà còn có khả năng tự trích xuất đặc trưng, học hỏi từ dữ liệu tương tác và liên tục tối ưu hóa hành vi [2]. Sự chuyển dịch từ tự động hóa cơ học thuần túy sang tự chủ thông minh thông qua AI đã và đang mở ra những tiềm năng to lớn, nâng cao đáng kể độ chính xác, tính ổn định và khả năng phục hồi của hệ thống robot khi đối mặt với những nhiễu động không thể lường trước.

Để giải quyết bài toán dẫn đường và điều khiển chuyển động cho cánh tay robot, nhiều phương pháp tiếp cận đã được đề xuất và phát triển. Ban đầu, các bộ điều khiển tuyến tính như PID hay điều khiển mô-men tính toán (Computed Torque Control) được áp dụng phổ biến nhờ cấu trúc đơn giản và khả năng bám quỹ đạo tốt khi cấu hình động lực học được nhận dạng chính xác [3]. Tuy nhiên, các phương pháp này bộc lộ sai số lớn khi hệ thống hoạt động trong môi trường động và chịu nhiều biến thiên phi tuyến. Để xử lý sự thay đổi tham số hệ thống như tải trọng động, điều khiển thích nghi (Adaptive Control) được ứng dụng nhằm tự động cập nhật các hệ số điều khiển; song, tốc độ hội tụ chậm trong điều kiện nhiễu đột ngột vẫn là một hạn chế [4]. Đồng thời, các kỹ thuật điều khiển tối ưu (Optimal Control) mang lại hiệu năng cao trong việc tối thiểu hóa năng lượng hoặc thời gian di chuyển dựa trên một hàm mục tiêu xác định, nhưng lại phụ thuộc quá mức vào độ chính xác của mô hình toán học và khó đáp ứng yêu cầu tính toán theo thời gian thực đối với hệ 6 bậc tự do phức tạp [5]. Nhằm khắc phục tính phụ thuộc vào mô hình, điều khiển trượt (SMC) được phát triển giúp duy trì ổn định dưới tác động của nhiễu ngoại vi, dù hiện tượng "chattering" vẫn chưa được loại bỏ triệt để. Để nâng cao tính linh hoạt [6], các phương pháp tích hợp logic mờ (Fuzzy Logic) và mạng nơ-ron nhân tạo (Neural Networks) đã được đề xuất để xấp xỉ các thành phần bất định và hiệu chỉnh tham số trực tuyến [7]; tuy nhiên, chúng thường đòi hỏi khối lượng lớn tri thức chuyên gia để xây dựng tập luật (đối với hệ mờ) hoặc tiêu tốn nhiều tài nguyên tính toán để xử lý dữ liệu (đối với mạng nơ-ron). Tương tự, điều khiển dự báo mô hình (MPC) dù xử lý tốt các đa ràng buộc của hệ thống nhưng lại vấp phải rào cản lớn về gánh nặng tính toán [8]. Khắc phục những hạn chế đó và hướng tới việc đảm bảo tính đa nhiệm, tối ưu hóa quỹ đạo trong môi trường động không biết trước, các phương pháp tiếp cận dựa trên học tăng cường (Reinforcement Learning - RL) đã tạo ra một bước đột phá [9]. Các thuật toán thế hệ đầu như Q-learning hay DQN mang lại ưu thế phi mô hình (model-free) nhưng lại bị giới hạn bởi hiện tượng bùng nổ chiều không gian do chỉ hoạt động trên không gian hành động rời rạc [10]. Để giải quyết rào cản này, cấu trúc học tăng cường sâu (Deep Reinforcement Learning) dạng Actor-Critic như DDPG (Deep Deterministic Policy Gradient) đã ra đời [11], cho phép tối ưu hóa trực tiếp trên không gian hành động liên tục. Dẫu vậy, DDPG vẫn tồn tại nhược điểm về tính hội tụ cục bộ, sự nhạy cảm với các siêu tham số và thiếu sự cân bằng trong chiến lược khám phá. Gần đây nhất, phương pháp học tăng cường mềm Actor-Critic (Soft Actor-Critic - SAC) nổi lên như một giải pháp toàn diện để giải quyết các khiếm khuyết trên. Bằng cách tích hợp cơ chế cực đại hóa entropy vào hàm mục tiêu, SAC không chỉ khuyến khích robot khám phá không gian trạng thái một cách hiệu quả, tránh rơi vào các điểm tối ưu cục bộ, mà còn gia tăng đáng kể tính ổn định và tốc độ hội tụ. Ưu thế vượt trội này khiến SAC trở thành một trong những kỹ thuật điều khiển hiện đại, đáp ứng xuất sắc yêu cầu thích ☞

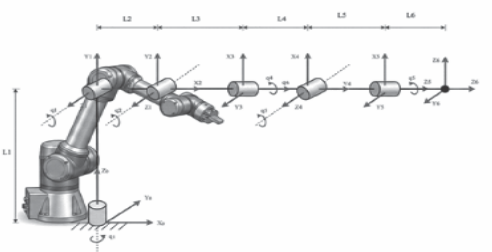
ứng và nội suy hành động trong các tác vụ động học phức tạp của robot.

Trong nghiên cứu này, chúng tôi đề xuất một quy trình thiết kế đồng bộ bộ điều khiển học tăng cường SAC áp dụng cho cánh tay robot 6 bậc tự do (6-DOF). Bước đầu tiên, nền tảng toán học của hệ thống được thiết lập thông qua việc xây dựng các phương trình động học và động lực học nghiêm ngặt. Dựa trên cơ sở đó, hệ thống được mô hình hóa dưới dạng một quá trình chính sách quyết định, bao gồm việc xây dựng không gian trạng thái, định nghĩa các biến trạng thái (như góc khớp, vận tốc khớp, tọa độ đối tượng), thiết lập hàm phần thưởng đa mục tiêu và tối ưu hóa các mạng nơ-ron chính sách (Actor) - giá trị (Critic) của thuật toán SAC nhằm đảm bảo điều khiển bám quỹ đạo trơn tru. Cuối cùng, độ tin cậy và khả năng phản xạ theo thời gian thực của bộ điều khiển được kiểm chứng khắt khe thông qua một bài toán thực nghiệm động lực học phức tạp: điều khiển cánh tay robot dự đoán và đánh trúng quả bóng với các quỹ đạo ném ngẫu nhiên và biến thiên liên tục khi rơi xuống bàn bóng bàn.

Mặc dù các phương pháp học tăng cường sâu đã đạt được nhiều thành công trong điều khiển robot, việc học trực tiếp trên toàn bộ không gian hành động vẫn gặp khó khăn do không gian tìm kiếm lớn và đặc tính phần thưởng thưa của bài toán đánh bóng bàn. Hơn nữa, việc điều khiển chính xác quỹ đạo bóng sau va chạm đòi hỏi sự phối hợp chặt chẽ giữa động học robot và khả năng thích nghi của học tăng cường. Xuất phát từ những vấn đề đó, nghiên cứu này đề xuất một kiến trúc điều khiển lai kết hợp bộ lập kế hoạch quỹ đạo dựa trên động học ngược với thuật toán Soft Actor-Critic theo cơ chế Residual Policy Learning và Curriculum Learning. Đồng thời, quỹ đạo bóng được dự đoán bằng phương pháp Runge–Kutta bậc 4 có xét đến lực cản không khí và hiệu ứng Magnus, kết hợp với hàm thưởng phân tầng nhằm nâng cao khả năng đánh bóng thành công. Các đóng góp chính của nghiên cứu bao gồm: đề xuất kiến trúc điều khiển IK + Residual SAC + Curriculum Learning cho robot 6 bậc tự do; xây dựng bộ lập kế hoạch vung vợt dựa trên dự đoán quỹ đạo bóng; và đánh giá hiệu năng hệ thống thông qua so sánh với các thuật toán học tăng cường phổ biến và nghiên cứu trong cùng lĩnh vực.

2. PHƯƠNG PHÁP NGHIÊN CỨU

2.1. Mô hình động học cánh tay robot 6 bậc tự do



Hình 1. Biểu diễn hệ quy chiếu cánh tay robot 6 bậc tự do

Cánh tay robot sử dụng trong mô hình thực nghiệm là dạng cánh tay robot 6 bậc tự do (6-DOF) trong Hình 1, cho phép điều khiển đầy đủ cả vị trí và hướng của đầu công tác trong không gian 3D. Động học thuận của robot được mô tả bằng quy ước Denavit–Hartenberg (D–H) trong Bảng 1.

Bảng 1. Bảng thông số D-H của robot 6 bậc tự do

Khâu	θ_i	d_i	a_i	α_i
1	q_1	L_1	0	90°
2	q_2	0	L_2	0
3	q_3	0	0	90°
4	q_4	L_4	0	-90°
5	q_5	0	0	90°
6	q_6	L_6	0	0

Ma trận biến đổi thuần nhất từ khâu i-1 sang khâu i được định nghĩa như sau:

$$ {}^{i-1}T_i = \begin{bmatrix} cos\theta_i & -sin\theta_i cos\alpha_i & sin\theta_i sin\alpha_i & a_i cos\theta_i \\ sin\theta_i & cos\theta_i cos\alpha_i & -cos\theta_i sin\alpha_i & a_i sin\theta_i \\ 0 & sin\alpha_i & cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1) $$

ISSN 2615 - 9910 (bản in), ISSN 2815 - 5505 (online)
TẠP CHÍ CƠ KHÍ VIỆT NAM, Số 342, tháng 6 năm 2026
cokhivietnam.vn

Ma trận biến đổi tổng thể từ hệ tọa độ gốc đến điểm cuối công tác được tính bằng phương trình (2):

$$^{0}T_{6} = \prod_{i=1}^{6} {}^{i-1}T_{i} = f(q) \quad (2)$$

Cho trước một tư thế mong muốn của đầu công tác:

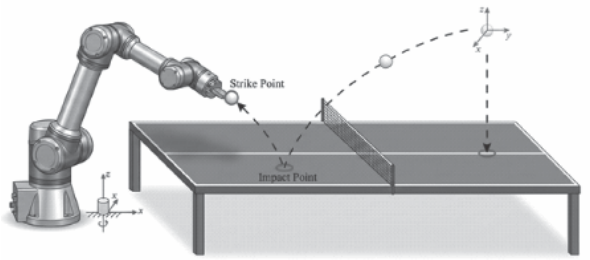
$$x = (p,R) \quad (3)$$

Trong đó, $p \in \mathbb{R}^3$ là vị trí (x, y, z) và $R \in SO(3)$ là ma trận quay.

Phương trình động học ngược được biểu diễn như sau:

$$q^* = \underset{q}{\operatorname{argmin}} \| f(q) - x \|^2 \quad (4)$$

Để đảm bảo tính đồng bộ hoàn toàn với engine vật lý của môi trường mô phỏng và xử lý triệt để các ràng buộc về giới hạn góc khớp, nghiên cứu này sử dụng phương pháp giải tích số học thông qua bộ giải IK tích hợp sẵn của PyBullet.



Hình 2. Môi trường làm việc của cánh tay robot

Dựa trên quỹ đạo bay của bóng được dự đoán bằng Runge–Kutta bậc 4 (RK4) trong Hình 2, cánh tay robot xác định được điểm đánh p_{hit} và hướng tiếp cận bóng. Thay vì giải IK cho duy nhất điểm đánh, để tạo ra cú đánh mượt mà, tại đầu mỗi episode robot chọn ra 3 điểm dựa trên điểm va chạm p_{hit} và hướng vận tốc bóng tại thời điểm va chạm:

$$d_{ball} = \frac{v_{ball}}{\|v_{ball}\|} \quad (5)$$

Điểm chuẩn bị :

$$p_{pre} = p_{hit} + 0.12 \cdot d_{ball} \quad (6)$$

Điểm vung vợt :

$$p_{follow} = p_{hit} - 0.1 \cdot d_{ball} \quad (7)$$

Từ 3 điểm này, robot giải động học ngược ra 3 cấu hình khớp tương ứng q_{pre}, q_{hit}, q_{follow}.

Quỹ đạo khớp cơ sở theo thời gian q_{ik} được tính toán dựa trên thời gian còn lại đến lúc va chạm t_{hit}.

$$q_{ik} = \begin{cases} q_{pre}, & \text{nếu } t_{hit} > t_{swing} \\ q_{ik} = (1-t)\, q_{hit} + t\, q_{follow}, & \text{nếu } t_{hit} \le t_{swing} \end{cases} \quad (8)$$

Trong đó:

$$t_{swing} = 0.35s \text{ và } t = \text{clip}\left(1 - \frac{t_{hit}}{t_{swing}},\ 0.0,\ 1.0\right)$$

Sau đó, quỹ đạo khớp được nội suy theo thời gian để tạo ra chuyển động liên tục:

$$q_{target} = \text{clip}(q_{ik} + (1-\alpha) \cdot \text{Action} \cdot \delta,\ q_{min},\ q_{max}) \quad (9)$$

Trong đó: $\alpha \in [0,1]$ là hệ số curriculum động, $\delta = 0.8$ rad là hệ số bù trừ tối đa cho phép.

2.2. Kiến trúc mô hình

Để giải quyết bài toán điều khiển liên tục trong không gian trạng thái đa chiều, nghiên cứu áp dụng thuật toán SAC với cơ chế tối đa hóa Entropy (Maximum Entropy RL). Nghiên cứu này sử dụng kiến trúc mạng nơ-ron truyền thẳng đa tầng (Multi-Layer Perceptron - MLP). Hệ thống SAC bao gồm hai thành phần mạng song song: Actor Network và Critic Network được biểu diễn như trong Hình 3.

Actor Network được thiết kế với 2 lớp ẩn, mỗi lớp gồm 256 nơ-ron sử dụng hàm kích hoạt ReLU. Actor Network biểu diễn chính sách ngẫu nhiên $\pi_\theta(a \mid s)$ dưới dạng phân phối ☞

Gaussian nhiều biến với trung bình μ(s) và độ lệch chuẩn σ(s) phụ thuộc vào trạng thái. Hành động thực tế được lấy mẫu thông qua Reparameterization Trick kết hợp với hàm giới hạn để đưa tín hiệu về [-1,1]:

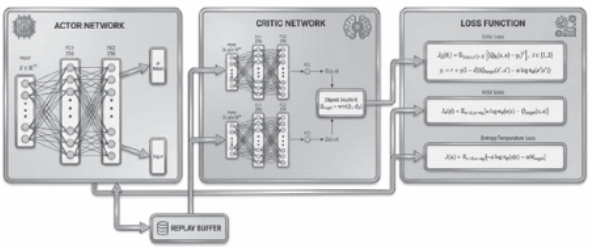
$$\varepsilon \sim N(0, I_6),\ x = \mu(s) + \sigma(s) \odot \varepsilon\ ,\ a = \tanh(x) \in [-1, 1]^6 \quad (10)$$

Hàm mất mát của mạng Actor được thiết kế để tối đa hóa giá trị Q dự kiến đồng tối đa hóa entropy:

$$J_\pi(\phi) = E_{s\sim D}\left[E_{a\sim\pi_\phi(\cdot\vee s)}\left(\alpha\log\pi_\phi(a\vee s) - Q_{min}(s,a)\right)\right] \quad (11)$$

Trong đó:

$$Q_{min}(s,a) = \min_{i\in\{1,2\}} Q_{\theta_i}(s,a)$$



Hình 3. Kiến trúc mạng học tăng cường mềm Actor - Critic (SAC)

Để tránh hiện tượng overestimation của giá trị Q, vốn là nguyên nhân gây bất ổn định trong các thuật toán actor-critic, hệ thống sử dụng hai mạng Critic độc lập. Hai mạng nơ-ron song song $Q_{\theta 1}$ (s,a), $Q_{\theta 2}$ (s,a) có cùng cấu trúc: Nhận đầu vào là vector nối giữa S_t và A_t với kích thước là 28 chiều; đi qua 2 lớp ẩn (256 nơ-ron, ReLU) và xuất ra một giá trị Q vô hướng. Giá trị nhỏ nhất giữa 2 mạng này được sử dụng để cập nhật cho Actor Network:

$$Q_{min}(s_t,a_t) = \min\left(Q_{\theta_1}(s_t,a_t), Q_{\theta_2}(s_t,a_t)\right) \quad (12)$$

Hàm mất mát của Critic được định nghĩa bằng sai số bình phương trung bình (MSE) giữa dự đoán hiện tại và mục tiêu Bellman:

$$J_Q(\theta_i) = E_{(s,a,r,s\prime,d)\sim D}\left[\frac{1}{2}\left(Q_{\theta_i}(s,a) - y\right)^2\right], i \in \{1,2\} \quad (13)$$

Trong đó, mục tiêu y được tính bằng:

$$y = r + \gamma(1-d)\left[Q_{min}(s',a') - \alpha\log\pi_\phi(a'\vee s')\right]$$
$$\text{với } a' \sim \pi_\phi(\cdot\vee s') \quad (14)$$

Hệ số α kiểm soát sự cân bằng giữa khai thác và khám phá. Thay vì gán một giá trị tĩnh, hệ thống triển khai cơ chế điều chỉnh tự động để duy trì một mức Entropy mục tiêu. Loss function để cập nhật α được thiết lập:

$$J(\alpha) = E_{s\sim D, a\sim\pi_\phi(\cdot\vee s)}\left[-\alpha\left(\log\pi_\phi(a\vee s) + H_{target}\right)\right] \quad (15)$$

2.3. RL State

Trạng thái bao gồm thông tin về động học của cánh tay robot, chuyển động của bóng, cũng như thông tin thời gian tương tác cần thiết cho nhiệm vụ đánh bóng. Tại mỗi thời điểm t, trạng thái $S_t \in R$ được định nghĩa như sau:

$$S_t = \{q, \dot{q}, p_{ball}, v_{ball}, p_{hit}, t_{hit}\} \in \mathbb{R}^{22} \quad (16)$$

Trong đó :

$q = [q_1, q_2, q_3, q_4, q_5, q_6]^T$ là vị trí các khớp.

$\dot{q} = [\dot{q}_1, \dot{q}_2, \dot{q}_3, \dot{q}_4, \dot{q}_5, \dot{q}_6]^T$ là vận tốc các khớp. Các biến này cho biết trạng thái hiện tại của tay máy.

$p_{ball} = \{x_b, y_b, z_b\}$ là vị trí của bóng tại thời điểm t.

$v_{ball} = \{v_{bx}, v_{by}, v_{bz}\}$ là vận tốc của bóng tại thời điểm t. Các thành phần này cho biết trạng thái hiện tại của quả bóng, trong việc dự đoán quỹ đạo và xác định thời điểm đánh bóng và vị trí đánh bóng.

$p_{hit} = \{x_h, y_h, z_h\}$ là điểm đánh của cánh tay robot được tính toán thông qua việc dự đoán quỹ đạo dựa trên vị trí và vận tốc hiện tại của bóng. Điểm này biểu diễn vị trí tối ưu để cánh tay robot thực hiện tiếp xúc với bóng.

t_{hit} là đại lượng vô hướng này biểu thị thời gian còn lại trước khi bóng đạt tới điểm va

ISSN 2615 - 9910 (bản in), ISSN 2815 - 5505 (online)
TẠP CHÍ CƠ KHÍ VIỆT NAM, Số 342, tháng 6 năm 2026
cokhivietnam.vn

chạm dự đoán. Thông tin này giúp chính sách điều khiển thích nghi theo các pha chuyển động khác nhau.

Tổng số chiều vectơ trạng thái S_t là 22. Từ phương trình gia tốc của bóng:

$$m\frac{d\vec{v}}{dt} = \vec{F}_g + \vec{F}_d + \vec{F}_m \quad (17)$$

Chúng ta xác định quỹ đạo bóng bằng phương pháp Runge Kutta bậc 4. Để xác định p_{hit} cánh tay robot chọn điểm đánh bóng thuộc quỹ đạo của bóng và thuộc workspace của robot, các điểm sao cho $t_{dự} = t_{robot} - t_{ball} \geq 0.02s$, điểm gần đế robot, điểm gần tâm bàn và điểm ở độ cao phù hợp.

2.4. RL Action

Không gian hành động được thiết kế dưới dạng vector dư liên tục sáu chiều a $\in$ $[-1,1]^6$, trong đó mỗi chiều tương ứng với một khớp quay của cánh tay robot. Thay vì điều khiển trực tiếp vị trí khớp tuyệt đối, mạng chính sách đầu ra residual so với quỹ đạo thạm chiếu tính bằng động học ngược, theo mô hình residual policy learning:

$$q_{target} = clip(q_{ik} + (1-\alpha)\cdot Action\cdot\delta,\ q_{min},\ q_{max}) \quad (18)$$

Trong đó, Action là đầu ra của mạng chính sách ngẫu nhiên Actor π_θ (a | s), được sinh ra thông qua kỹ thuật tái tham số hóa:

$$\begin{aligned} \epsilon &\sim \mathcal{N}(0, I_6) \\ x &= \mu_\theta(s) + \sigma_\theta(s) \odot \epsilon \\ a_t &= \tanh(x) \in [-1,1]^6 \end{aligned} \quad (19)$$

Trong đó, $\mu_\theta(s) \in \mathbb{R}^6$ và $\sigma_\theta(s) \in \mathbb{R}^6$ lần lượt là vector trung bình và độ lệch chuẩn được mạng Actor suy ra từ trạng thái s, ϵ là nhiễu Gaussian chuẩn. Hàm kích hoạt tanh(·) được sử dụng để giới hạn không gian hành động trong miền $[-1,1]^6$.

2.5. RL Reward

Phần thưởng tổng R_t là tổng có trọng số của hai pha chính (đánh trúng và rơi đúng bàn) cùng với các hình phạt an toàn:

$$R_t = w_{hit}R_{hit} + w_{land}R_{land} + R_{pen} \quad (20)$$

Trong đó: $w_{hit} = 0.7$ và $w_{land} = 0.3$. Đánh trúng một vật thể di chuyển ở tốc độ cao là phần khó hơn việc gán trọng số áp đảo (0.7) ép mạng nơ-ron phải ưu tiên giải quyết bài toán đánh trúng bóng trước khi tối ưu đường bay của bóng.

- Phần thưởng pha đánh trúng R_{hit}.

Trong đó:

Định vị: $R_{pos} = \frac{w_{pos}}{\|p_{tcp} - p_{hit}\| + \epsilon}$ (21)

sẽ khuyến khích TCP tiến sát điểm chạm bóng dự đoán. Sau đó, vận tốc tiếp cận:

$$R_{approach} = w_{approach}\cdot\max(0, v_{tcp}\cdot\hat{d}_{ball}) \quad (22)$$

thúc đẩy TCP chủ động lao tới theo vector hướng d_{ball} về phía bóng.

Trong giai đoạn đánh bóng:

$$R_{swing} = w_{swing}\cdot\|v_{tcp}\| \quad (23)$$

tiếp tục khuyến khích vận tốc cao tại thời điểm đánh.

Chạm bóng: Điểm thưởng tuyệt đối +1.0 được cấp ngay khi xảy ra va chạm vật lý giữa vợt và bóng.

- Phần thưởng pha bóng rơi R_{land}

Thành phần này chỉ được tính khi đã đánh bóng thành công:

$$R_{land} = R_{dir} + R_{land}^{sparse} \quad (24)$$

Trong đó:

$R_{dir} = w_{dir} \cdot \max(0, v_{ball,x})$: thưởng khi bóng bay về phía đối phương.

$R_{land}^{sparse} = \begin{cases} +1, & \text{bóng này đúng bàn đối phương} \\ 0, & \text{ngược lại} \end{cases}$: thưởng khi bóng tiếp đất hợp lệ.

- Phần thưởng phạt R_{pen}

$R_{smooth} = -\lambda_1 \cdot \| a_t \|^2$: Giảm các hành động lớn, giúp chuyển động mượt hơn.

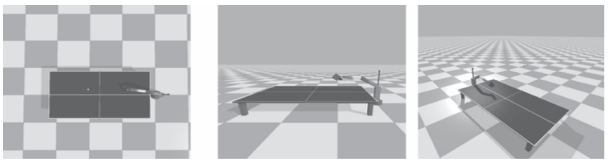
$R_{joint} = -\lambda_2 \cdot N_{limit}$: Tránh cấu hình nguy hiểm hoặc không khả thi.

Cuối cùng, sau khi hoàn thành các bước như trên, bộ điều khiển cánh tay robot 6 bậc đánh trúng bóng được biểu diễn như trong Hình 4.

Hình 4. Cấu trúc tổng thể bộ điều khiển cánh tay robot đánh trúng bóng dựa trên phương pháp học tăng cường mềm Actor-Critic

3. KẾT QUẢ

3.1. Môi trường mô phỏng



a) b) c)

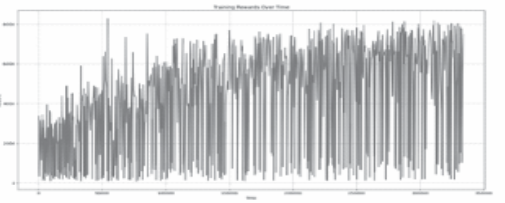
Hình 5. Môi trường thực tế ảo Pybullet mô phỏng cánh tay robot làm việc với (a): góc nhìn trên cao, (b) góc nhìn từ hình chiếu cạnh, và (c) góc nhìn chéo 45°

Quá trình mô phỏng được thực hiện trên máy tính với cấu hình: CPU Intel Core i7-11800H (Gen 11), RAM:16GB, GPU: NVIDIA GeForce RTX 3050. Môi trường mô phỏng được xây dựng trên PyBullet, trong đó robot là một tay máy 6 bậc tự do thực hiện nhiệm vụ đánh bóng bàn. Đầu công tác được thiết kế tương tự mặt vợt, tạo tiếp xúc điểm với quả bóng, làm tăng độ khó của bài toán do yêu cầu chính xác cao về vị trí và thời điểm va chạm. Bàn bóng bàn được mô phỏng với kích thước 2.74 × 1.525 m, mặt bàn ở độ cao 0.38 m. Vị trí ban đầu của Robot là [0.0 ; 0.0 ; 0.0], bóng được phóng ngẫu nhiên từ phía đối phương với vị trí x ∈ [2.0 ; 2.8] m, y ∈ [−0.4 ; 0.4] m, z ∈ [1.0 ; 1.5] m. Mô hình khí động học bao gồm lực cản không khí và lực Magnus được tích hợp để tạo chuyển động giống thực tế của bóng bàn như trong Hình 5.

Dựa trên môi trường Pybullet, mô hình được huấn luyện tổng cộng 2000000 steps, với mỗi episode kéo dài tối đa 500 steps.

3.2. Kết quả thực nghiệm

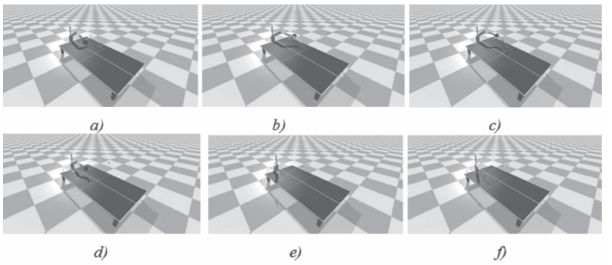
Ở giai đoạn đầu huấn luyện , mặc dù có những episode thành công nhưng phần lớn là do hệ số curriculum α còn cao, cánh tay robot được hướng dẫn bởi bộ lập kế hoạch IK. Theo thời gian, các kết quả thu được trở nên ổn định hơn vì cánh tay robot đã học được cách đánh bóng tối ưu và lựa chọn quỹ đạo vung vợt để đạt phần thưởng lớn nhất.



Hình 6. Kết quả huấn luyện với chỉ số phần thưởng (reward)

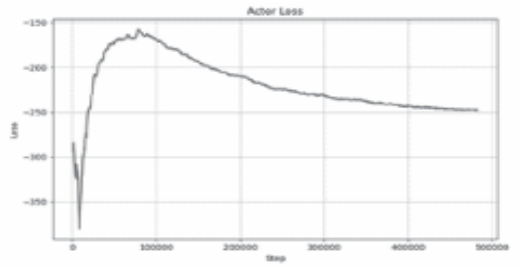
Biểu đồ Training Reward trong Hình 6 thể hiện phần thưởng trung bình theo số bước huấn luyện. Có thể thấy phần thưởng duy trì ở mức cao ngay từ giai đoạn đầu (2000 - 4000 điểm) do cơ chế Curriculum Residual Policy. Khi α giảm dần, phần thưởng tiếp tục tăng lên mức 6.000–8.000 điểm, chứng tỏ agent SAC

đã học được các hiệu chỉnh vượt ra ngoài khả năng của bộ lập kế hoạch động học. Điều này chứng minh rằng mô hình đề xuất hoạt động thành công và khám phá hiệu quả không gian hành động của môi trường (xem Hình 7).



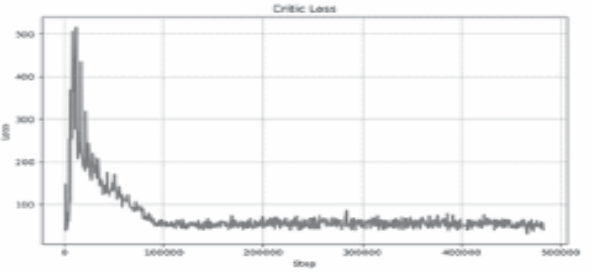
Hình 7. Kết quả mô phỏng cánh tay robot đánh bóng thành công với từ (a) đến (c): đánh thành công bóng bay từ phía nửa bàn bên phải và từ (d) đến (f) đánh bóng bay từ phía nửa bàn bên trái

Kết quả mô phỏng cho thấy robot có khả năng học và thực hiện cú đánh ổn định trong nhiều kịch bản khác nhau. Ở trường hợp (a) đến (c), bóng được đánh thành công từ phía phải, trong khi (d) đến (f), thể hiện khả năng xử lý tương tự từ phía trái. Quỹ đạo chuyển động của tay máy mượt và phù hợp với hướng bóng chứng tỏ chính sách học được hội tụ tốt, đồng thời cho thấy hàm phần thưởng và cơ chế Curriculum Learning đã hoạt động đúng như thiết kế.



Hình 8. Actor Loss

Từ biểu đồ Actor Loss (Hình 8) cho thấy trong giai đoạn đầu (0 -20000 steps), loss giảm mạnh xuống khoảng -380 do hệ số curriculum còn cao ($\alpha \approx 0.7$), trong khi SAC chỉ cần học phần residual nhỏ và Q-value vẫn chưa ổn định. Tiếp theo, Actor Loss tăng lên khoảng -155 tại step 80000. Giai đoạn này tương ứng với pha chuyển tiếp curriculum learning, khi SAC bắt đầu điều khiển tay máy và khám phá không gian hành động. Từ bước 100000 đến 500000, Actor Loss tiếp tục giảm dần từ -155 xuống khoảng -250, cho thấy policy ngày càng khai thác hiệu quả hơn.



Hình 9. Critic Loss

Từ biểu đồ Critic Loss (Hình 9) cho thấy Critic loss khởi đầu ở mức cao (500) trong khoảng 20000 bước đầu do mạng Q-network chưa xây dựng được ước lượng giá trị đáng tin cậy và replay buffer còn thưa dữ liệu. Sau đó, giá trị loss giảm nhanh và hội tụ về vùng ổn định khoảng 50-80 kể từ sau bước 100000 và duy trì ổn định. Xu hướng này cho thấy mạng Critic đã dần học được hàm giá trị Q(s,a) chính xác hơn, phản ánh tốt phần thưởng tích lũy thực tế của chính sách.

Sau khi đã huấn luyện thành công, robot thực hiện đánh bóng với 2000 episode, với các chỉ số kỹ thuật đạt được ấn tượng như Bảng 2.

Bảng 2. Các chỉ số hiệu suất chất lượng đạt được của cánh tay robot đánh bóng

Chỉ số (index)	**Giá trị (values)**	**Trạng thái (state)**
Hit rate	92%	Hit the ball
Land rate	25 %	Ball lands on opponent table
Inference time	0.060 ± 0.010 ms	Per action step
Control frequency	60 FPS	Control frequency
Alpha	0.096	Exploration coef

Bảng 2 trình bày các chỉ số hiệu năng chính của hệ thống đề xuất. Kết quả cho thấy tỷ lệ đánh trúng bóng (hit rate) đạt 92%, chứng tỏ agent đã học được chính sách điều khiển ổn định để tiếp cận và va chạm với bóng một cách hiệu quả. Tuy nhiên, tỷ lệ bóng sang bàn đối phương (land rate) chỉ đạt 25%, cho thấy nhiệm vụ điều khiển quỹ đạo sau va chạm vẫn còn nhiều thách thức, đặc biệt liên quan đến việc kiểm soát hướng và vận tốc của bóng sau khi đánh.

Về hiệu năng tính toán, thời gian suy luận trung bình chỉ 0.060 ms mỗi bước, tương ứng với tần số điều khiển 60 Hz, cho thấy hệ thống hoàn toàn đáp ứng yêu cầu thời gian thực. Ngoài ra, giá trị hệ số entropy $\alpha = 0.096$ phản ánh rằng chính sách đã hội tụ về mức khai thác cao, với mức độ khám phá thấp hơn so với giai đoạn đầu huấn luyện.Tổng thể, các kết quả này xác nhận rằng phương pháp nghiên cứu đạt hiệu quả cao trong nhiệm vụ đánh trúng bóng, đồng thời vẫn còn cải thiện đối với bài toán điều khiển quỹ đạo bóng sau va chạm.

3.3. Đánh giá ổn định của hệ thống

Bảng 3. Các chỉ số đánh giá sự ổn định của hệ thống

Chỉ số (index)	Giá trị	SD	95%CI
Hit Rate (%)	92	27.27	[86.59,97.41]
Land Rate (%)	25	43.28	[23.05,26.85]
Episode Length	134.5	72.5	[120.1,148.9]
Time to hit (s)	0.0384	0.05	[0.0281,0.0488]

Để đánh giá độ ổn định của hệ thống, mô hình được kiểm tra trên 100 episode độc lập với các điều kiện khởi tạo bóng ngẫu nhiên. Kết quả cho thấy tỷ lệ đánh trúng bóng đạt 92.0% (95% CI: [86.59, 97.41]), trong khi tỷ lệ bóng rơi sang phần sân đối phương đạt 25.0%, cho thấy hệ thống đã giải quyết tốt bài toán tiếp xúc cánh tay robot đánh bóng nhưng vẫn còn nhiều thách thức trong việc điều khiển chính xác quỹ đạo bóng sau va chạm.

3.4. So sánh với các thuật toán học tăng cường khác

Sau khi robot được huấn luyện với các thuật toán học tăng cường khác, cánh tay robot được chạy với 200 episodes.

Bảng 4. Các chỉ số hiệu suất chất lượng với các thuật toán khác

Phương pháp	Hit rate (%)	Land Rate (%)
PPO	1	0
DDPG	3	0
TD3	9	0.5
SAC	53	9.5
Đề xuất	90	25

Kết quả so sánh cho thấy phương pháp đề xuất đạt hiệu năng vượt trội so với các thuật toán PPO, DDPG, TD3 và SAC thuần túy. Cụ thể, tỷ lệ đánh trúng bóng đạt 90% và tỷ lệ bóng rơi sang phần sân đối phương đạt 25%, cao hơn đáng kể so với SAC thuần túy (53% và 9.5%). Trong khi đó, PPO, DDPG và TD3 hầu như không học được nhiệm vụ với tỷ lệ đánh trúng bóng rất thấp. Kết quả này cho thấy việc kết hợp bộ lập kế hoạch động học ngược với cơ chế Residual Policy Learning và Curriculum Learning giúp thu hẹp không gian tìm kiếm, cải thiện khả năng hội tụ và nâng cao đáng kể hiệu quả điều khiển trong bài toán cánh tay robot đánh bóng bàn.

3.5. Đánh giá so sánh

Bảng 5. Các chỉ số hiệu suất chất lượng khi thực hiện các phương pháp khác nhau

Method	Hit Rate (%)	Land Rate (%)
Full model	90	23
No Curriculum	60	20
No IK	58.5	10.5
No Residual	42	7

Kết quả Ablation Study cho thấy mô hình đầy đủ đạt hiệu năng cao nhất với Hit Rate 90.0% và Land Rate 23.0%. Khi loại bỏ Curriculum Learning, IK hoặc Residual Policy, hiệu năng đều suy giảm đáng kể. Điều này chứng tỏ sự kết hợp giữa bộ lập kế hoạch động học ngược, Residual Policy Learning và Curriculum Learning đóng vai trò quan trọng trong việc nâng cao khả năng đánh bóng thành công của hệ thống.

3.6. Hạn chế của hệ thống

Mặc dù hệ thống đạt Hit Rate cao, Land Rate chỉ đạt 25%, cho thấy việc điều khiển chính xác quỹ đạo bóng sau va chạm vẫn còn nhiều thách thức. Nguyên nhân chủ yếu là hàm thưởng ưu tiên nhiệm vụ đánh trúng bóng, trong khi bộ lập kế hoạch IK chưa trực tiếp tối ưu vận tốc và hướng của mặt vợt tại thời điểm va chạm. Ngoài ra, điều kiện phóng bóng ngẫu nhiên và đặc tính phần thưởng thưa của nhiệm vụ đưa bóng sang phần sân đối phương cũng làm tăng độ khó của quá trình học.

4. KẾT LUẬN

Bài báo nghiên cứu này đã đề xuất một bộ điều khiển cánh tay robot 6-DOF đánh bóng bàn kết hợp giữa bộ lập kế hoạch vung vợt dựa trên động học ngược (IK) và thuật toán Soft Actor-Critic (SAC) theo hướng residual policy, nhằm điều khiển cánh tay robot 6-DOF thực hiện nhiệm vụ đánh bóng bàn trong môi trường PyBullet. Kết quả thực nghiệm cho thấy hệ thống đạt hit rate 92%, chứng minh khả năng học hiệu quả trong điều kiện khởi tạo ngẫu nhiên. Đồng thời, mô hình Actor inference time đạt 0.060 ± 0.010 ms và độ trễ toàn bộ vòng điều khiển đạt 0.102 ms. và throughput 21,854 Hz trên CPU, đáp ứng yêu cầu điều khiển thời gian thực. Trong tương lai, nghiên cứu sẽ tập trung vào cải thiện land rate, thực hiện ablation study, đánh giá trên hệ thống robot thực, và mở rộng sang các kịch bản tương tác phức tạp hơn.❖

Ngày nhận bài: **10/6/2026**
Ngày phản biện: **25/6/2026**

Tài liệu tham khảo:

[1]. Tang Y, Zakaria MA, Younas M, "*Path Planning Trends for Autonomous Mobile Robot Navigation: A Review*". Sensors, vol. 25, no. 4, pp. 1206, 2025.

[2]. Hiffza S, "*Bio-Inspired Soft Robotics. AI-Driven Morphological Adaptation for Terrain Navigation*". International Journal of Advanced and Innovative Research (IJAIR), vol. 1, no. 3, pp. 17-20, 2025.

[3]. Vo DH et al, "*Position Control of 3-DOF Experimental Articulated Robot Arm using PID Controller*". Journal of Fuzzy Systems and Control, vol. 3, no. 1, pp. 73-80, 2025.

[4]. Du Z, Tang H, Gao S, Cai Y, "*Adaptive observed-based backstepping control for quantized robot arms*". Scientific Reports, vol. 16, no. 1, 2026

[5]. Xu G, Zhu Z, Lei Y, "*Privacy-Preservation-Based Adaptive Optimal Control for HiTL Dual-Arm Robot Systems with Actuator Faults*". Nonlinear Dynamics, vol. 113, no. 20, pp. 27785-27801, 2025.

[6]. Fu X, "*Design of Sliding Mode Control for Robot Arm Trajectory Tracking Based on Chattering Suppression*". Applied and Computational Engineering, vol. 146, no. 1, pp. 94-100, 2025.

[7]. Ersin C, Yaz M, "*Implementation and Comparison of Wearable Exoskeleton Arm Design with Fuzzy Logic and Machine Learning Control*". Journal of Sensors, vol. 2024, no. 1, pp. 1-17, 2024.

[8]. Gavgani BM et al, "*Real-Time Trajectory Adaptation for Tossing Robots Using Soft Switching Multiple Model Predictive Control*". Journal of Intelligent & Robotic Systems, vol. 111, no. 3, 2025.

[9]. Chen Y, Wang S, "*Optimization Control Method of Robot Arm Based on Multi-Agent Reinforcement Learning*". Journal of Circuits, Systems and Computers, 2025.

[10]. Gao T, "*Optimizing Robotic Arm Control Using Deep Q-Learning and Artificial Neural Networks Through Demonstration-Based Methodologies: A Case Study of Dynamic and Static Conditions*". Robotics and Autonomous Systems, vol. 181, 2024.

[11]. Zhang L et al, "*Morphologically Adaptive Compliance Control for Inchworm-Inspired Climbing Robot Through State-Dependent Switching DDPG Network*". IEEE/ASME Transactions on Mechatronics, vol. 31, no. 1, pp. 208-219, 2025.