

Thuật toán học tăng cường ứng dụng trong bài toán điều khiển thích nghi thông minh

Nguyễn Văn Đức, Sái Văn Cường*

Viện Tự động hóa Kỹ thuật quân sự, Viện Khoa học và Công nghệ quân sự, 89B Lý Nam Đế, Cửa Đông, Hoàn Kiếm, Hà Nội, Việt Nam.

*Email: svcuonghvtqs@gmail.com

Nhận bài: 15/01/2024; Hoàn thiện: 08/3/2024; Chấp nhận đăng: 14/3/2024; Xuất bản: 01/4/2024.

DOI: <https://doi.org/10.54939/1859-1043.j.mst.CAPITI.2024.62-68>

TÓM TẮT

Ngày nay, cùng với sự phát triển của khoa học và công nghệ, các đối tượng điều khiển ngày càng phức tạp, có độ phi tuyến cao và độ bất định lớn, khiến cho các giải thuật điều khiển kinh điển truyền thống không còn hiệu quả. Điều đó dẫn đến việc phải xây dựng các cấu trúc và tham số không xác định và yêu cầu phải có kỹ thuật điều khiển tiên tiến. Để giải quyết các bài toán điều khiển có yếu tố không xác định trong các mô hình động học của đối tượng điều khiển, phương pháp điều khiển thích nghi thông minh dựa trên thuật toán học tăng cường có khả năng điều chỉnh tự động và liên tục các tham số của bộ điều khiển được nhóm tác giả đề xuất trong bài báo này. Tính hiệu quả và khả thi của phương pháp đề xuất được kiểm chứng thông qua mô phỏng thực tế. Các kết quả mô phỏng so sánh đạt được đã xác nhận rằng bộ điều khiển được đề xuất là bộ điều khiển bền vững, thích nghi và có hiệu suất điều khiển cao.

Từ khóa: AI; Học tăng cường (Reinforcement learning); Điều khiển thông minh; Điều khiển dựa trên dữ liệu; Điều khiển thích nghi; PID.

1. ĐẶT VẤN ĐỀ

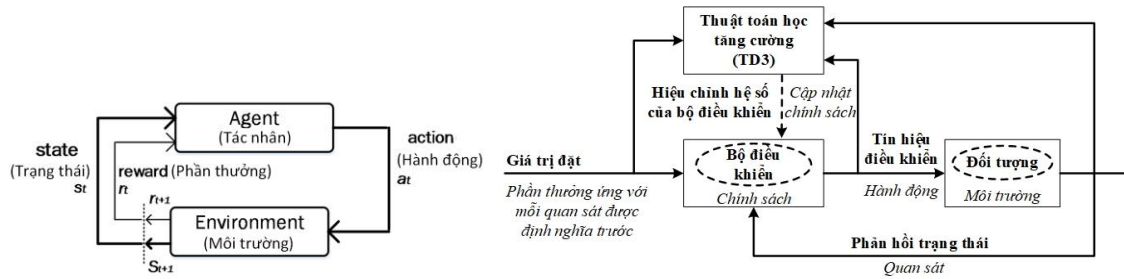
Lĩnh vực điều khiển và tự động hóa ngày càng phát triển, đặc biệt là điều khiển chính xác, đã trở thành một phần không thể thiếu của nền công nghiệp hiện đại. Với cấu trúc đơn giản và tính bền vững cao của phương pháp điều khiển tỷ lệ - vi phân - tích phân (PID) đã giúp nó trở thành một trong những bộ điều khiển điển hình của kỹ thuật điều khiển tự động, được sử dụng hầu hết trong các ứng dụng điều khiển quá trình tự động trong công nghiệp hiện nay. Khoảng 95% hệ thống vòng kín áp dụng kỹ thuật điều khiển này, trong đó chủ yếu là bộ điều khiển PI [1]. PID có nhiều ưu điểm như có khả năng điều chỉnh sai số thấp nhất có thể, tăng tốc độ đáp ứng, giảm độ vọt lố, hạn chế sự dao động. Tuy nhiên, thiết kế bộ điều khiển PID đòi hỏi phải biết chính xác về đối tượng, hoặc mô hình hóa tương đối chi tiết đối tượng. Việc lựa chọn các hệ số K_p, K_i, K_d của bộ điều khiển phải qua rất nhiều khâu tính toán thủ công, nhưng các thông số K_p, K_i, K_d tìm được này chưa phải là các thông số tối ưu. Bên cạnh đó, trong nhiều trường hợp, khả năng nhận dạng đối tượng qua mô hình toán là cực kỳ khó khăn và đôi khi là không thể, đặc biệt là các đối tượng phức tạp với các đặc tính động học mang tính phi tuyến cao và chịu ảnh hưởng của nhiều yếu tố nhiễu loạn bên ngoài. Trong trường hợp này, việc sử dụng bộ điều khiển PID truyền thống không đảm bảo được các chỉ tiêu về chất lượng điều khiển. Chính vì vậy, việc cải tiến chất lượng bộ điều khiển PID vẫn luôn là vấn đề thời sự cho các nhà nghiên cứu. Hiện nay, trên thế giới, các phương pháp điều khiển thích nghi hiện đại dựa trên các kỹ thuật tiên tiến trên cơ sở trí tuệ nhân tạo như mạng nơ ron nhân tạo, hệ chuyên gia, hệ logic mờ, mô hình định tính, thuật toán di truyền và nhiều tổ hợp từ các phương pháp này,... được đề xuất như một giải pháp thích hợp đối với bài toán điều khiển thích nghi cho các đối tượng phi tuyến phức tạp. Nhiều phương pháp dựa trên logic mờ để tự điều chỉnh thông số của bộ điều khiển đã được đề xuất [2], trong đó, các quy tắc mờ được áp dụng để thu được các tham số cho bộ điều khiển PID nhằm điều khiển các đối tượng phi tuyến khác nhau [3]. Theo cách tương tự, các thuật toán di truyền cũng được đề xuất sử dụng trong các bài toán điều chỉnh bộ điều khiển PID thích nghi [4, 5]. Gần đây, Zeng và cộng sự [6] đã phát triển

một phương pháp PID thích nghi dựa trên mạng nơ-ron nhân tạo đa biến để điều khiển các hệ thống phi tuyến đa biến, bộ điều này đã thể hiện tích khả thi thông qua các thí nghiệm mô phỏng.

Các phương pháp điều khiển thích nghi kể trên cho phép thiết kế bộ điều khiển với đối tượng bất định, dựa trên các luật thích nghi cho bộ điều khiển, có thể là gián tiếp thông qua cơ cấu nhận dạng đối tượng hay trực tiếp chỉnh định tham số bộ điều khiển, tuy nhiên, điều khiển thích nghi chưa xét đến yếu tố tối ưu chất lượng của luật điều khiển. Trong nghiên cứu này, mục tiêu nhóm tác giả hướng đến là xây dựng phương pháp điều khiển thích nghi thông minh dựa trên thuật toán học tăng cường (Reinforcement learning, RL). Ưu điểm chính của việc phương pháp học tăng cường là khả năng học hỏi từ sự tương tác với môi trường và cung cấp một chiến lược điều khiển tối ưu, cho phép điều khiển hệ thống mà không cần biết trước về mô hình động học của đối tượng.

2. THUẬT TOÁN HỌC TĂNG CƯỜNG

Học tăng cường (Reinforcement Learning, RL) [7] là một kỹ thuật trong trí tuệ nhân tạo (Artificial Intelligence, AI), trong đó, các tác nhân (agent) tương tác với môi trường (environment) thông qua các hành động (action) như thể hiện ở hình 1a. Một trạng thái (state) được cung cấp cho môi trường và tác nhân chọn một hành động dựa trên trạng thái đó để tối đa hóa phần thưởng (reward) [8]. Bản chất của RL là *trial-and-error*, nghĩa là thử đi thử lại và rút ra kinh nghiệm sau mỗi lần thử như vậy. Trong đó, hai thành phần cốt lõi là tác nhân và môi trường 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 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 (policy) tối đa hoá giá trị phần thưởng tích lũy trong thời gian dài.



a) Tương tác "Tác nhân - Môi trường" trong RL b) Bộ điều khiển thích nghi tối ưu trên cơ sở thuật toán học tăng cường.

Hình 1. Thuật toán học tăng cường ứng dụng trong bài toán điều khiển.

Bài toán RL được tổng quát hóa như một quá trình điều khiển ngẫu nhiên theo thời gian rời rạc, trong đó, một tác nhân tương tác với môi trường của nó theo cách sau: tác nhân bắt đầu, ở một trạng thái nhất định trong môi trường của nó $s_0 \in S$, bằng cách thu thập quan sát ban đầu $w_0 \in \Omega$. Tại mỗi bước thời gian t , tác nhân thực hiện hành động tại $a_t \in A$. Như được minh họa trong hình 1-a, nó tuân theo ba hệ quả: (1) tác nhân nhận được phần thưởng $r_t \in R$, (2) trạng thái chuyển sang $s_{t+1} \in S$ và (3) tác nhân nhận được quan sát $w_{t+1} \in \Omega$.

Gần đây, sự kết hợp RL với các kỹ thuật học sâu (Deep learning) được gọi là Deep RL đã nổi lên như một kỹ thuật tiên tiến để giải quyết các vấn đề phức tạp trong nhiều lĩnh vực khác nhau [9], từ những ứng dụng trong xe tự hành để giải quyết các bài toán tối ưu hóa quỹ đạo, lập kế hoạch chuyển động, dẫn động, tối ưu hóa bộ điều khiển,... cho đến các ứng dụng trong giao dịch và tài chính, hay là các ứng dụng của Deep RL trong việc chinh phục thành công các trò chơi, ví dụ các thuật toán của Google DeepMind như AlphaGo, AlphaZero, AlphaStar,... đã chiến thắng áp đảo

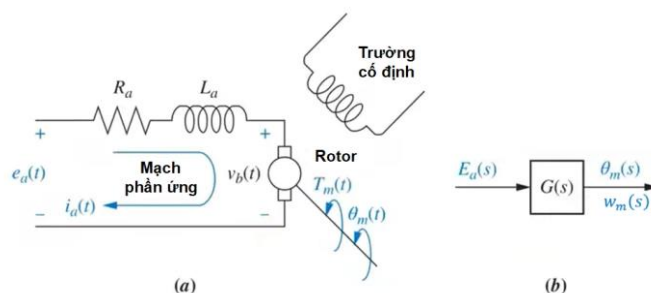
các tuyến thủ thế giới trong những trò chơi mà con người đã từng nghĩ rằng máy móc sẽ không bao giờ có thể vượt qua con người như cờ vây hay StarCraft. Trước hết, những thành tựu đạt được này thông qua những tiến bộ của các thuật toán nổi tiếng như là Deep Q-learning Network (DQN) [10], Actor – Critic (AC) [11], Deep Deterministic Policy Gradient (DDPG) [12] and Twin Delayed DDPG (TD3) [13]. Trong nghiên cứu này, nhóm tác giả nghiên cứu ứng dụng thuật toán TD3 trong bài toán điều khiển tốc độ động cơ DC, với nguyên lý được mô tả như trong hình 1-b. Ưu điểm của Deep RL trong bài toán này là cho phép điều khiển không cần mô hình, giúp giải quyết các vấn đề điều khiển ngay cả khi không có đầy đủ thông tin của mô hình hệ thống.

3. XÂY DỰNG BỘ ĐIỀU KHIỂN THÍCH NGHI THÔNG MINH TRÊN CƠ SỞ THUẬT TOÁN HỌC TĂNG CƯỜNG

Động cơ điện một chiều (DC) là một trong các thiết bị được sử dụng rộng rãi trong sản xuất và đời sống do sử dụng và điều khiển đơn giản, momen khởi động lớn. Hiện nay, bộ điều khiển tích phân tỉ lệ, gọi tắt là PI (Proportional-Integral controller) được biết đến như là kỹ thuật phổ biến nhất được sử dụng trong điều khiển động cơ DC. Ở phần này, nhóm tác giả mô tả việc xây dựng bộ điều khiển thích nghi thông minh trên cơ sở thuật toán học tăng cường TD3 thay thế bộ điều khiển PI truyền thống nhằm cải thiện chất lượng điều khiển.

3.1. Động lực học của động cơ điện một chiều (DC)

Mô hình vật lý của động cơ điện DC thể hiện ở hình 2a. Hình 2b mô tả tóm tắt sơ đồ hình 2a với ngõ vào là điện áp phản ứng $E_a(s)$ và ngõ ra là vị trí $\theta_m(s)$ hoặc vận tốc $w_m(s)$.



Hình 2. Mô hình động cơ DC.

Hàm truyền của động cơ DC trên hình 2 với ngõ vào là điện áp phản ứng và ngõ ra là vận tốc như sau:

$$G_v(s) = \frac{w_m(s)}{E_a(s)} = \frac{K_t}{L_a J_m s^2 + (R_a J_m + L_a D_m) s + (R_a D_m + K_b K_t)} \quad (1)$$

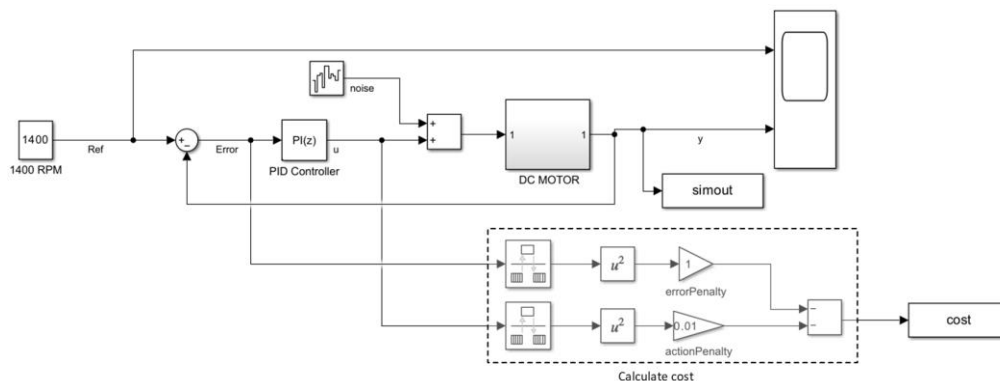
Ký hiệu và giá trị một số thông số động học của động cơ DC được nhóm tác giả sử dụng trong nghiên cứu này để mô phỏng thể hiện trong bảng 1.

Bảng 1. Các thông số động học của động cơ DC.

Ký hiệu	Ý nghĩa	Đơn vị	Giá trị
R_a	Điện trở phản ứng	Ohm	4.08
L_a	Điện cảm phản ứng	H	0.011307
J_m	Momen quán tính của rotor	Kgm ²	0.00048115
D_m	Hằng số ma sát dính	Nm/rad	0.0026829
K_t	Hằng số momen	Nm/A	0.22076
K_b	Hằng số điện áp cảm ứng	Vs/rad	0.22076

3.2. Xây dựng bộ điều khiển động cơ DC – PI

Mô hình điều khiển ổn định tốc độ động cơ DC với bộ điều khiển PI được thiết kế và mô phỏng trong Simulink dựa trên nguyên lý được biểu diễn như hình 3. Đồng thời đây cũng là mô hình môi trường để sử dụng thuật toán học tăng cường đề xuất. Mục tiêu của hệ thống điều khiển là duy trì tốc độ động cơ DC phù hợp với giá trị tốc độ đặt (tham chiếu).



Hình 3. Sơ đồ điều khiển tốc độ động cơ DC bằng bộ điều khiển PI.

Chính định tham số bộ điều khiển PI:

Bộ điều khiển PI được định nghĩa bởi (Johnson M. A. & M. H. Moradi, 2005) [14]:

$$u(t) = K_p e(t) + K_i \int_0^t e(t) dt \quad (2)$$

trong đó, u là tín hiệu điều khiển và e là sai lệch giữa tốc độ đặt (tham chiếu) và tốc độ thực ($e = v_{ref} - v$). K_p và K_i lần lượt là độ lợi tỉ lệ và độ lợi tích phân của bộ điều khiển. Nhiệm vụ của người thiết kế bộ điều khiển PI, là chọn lựa bộ giá trị $\{K_p, K_i\}$ thỏa mãn các yêu cầu về chất lượng điều khiển.

Trong lý thuyết điều khiển, có rất nhiều phương pháp để hiệu chỉnh thông số của bộ điều khiển PI, phổ biến nhất là phương pháp Ziegler-Nichols [15]. Tuy nhiên, việc hiệu chỉnh bộ điều khiển PI bằng phương pháp này đòi hỏi một quá trình thực nghiệm dựa trên kết quả đo đạc được từ đáp ứng của hệ thống khá mất thời gian [16]. Vì vậy, để có được sự điều chỉnh tối ưu cho bộ điều khiển PI, trong phạm vi nghiên cứu này nhóm tác giả sử dụng bộ điều chỉnh PID Tuner có sẵn trên MATLAB/Simulink. Kết quả thu được hệ số $K_p = 0.0034$ và $K_i = 0.022$.

3.3. Xây dựng bộ điều khiển động cơ DC – RL

Xây dựng môi trường huấn luyện tác nhân RL:

Để xây dựng môi trường huấn luyện cho tác nhân RL, mô hình Simulink trong bộ điều khiển PI hình 2 được sửa đổi bằng cách xóa bộ điều khiển PI và thay thế bằng khối tác nhân RL, đồng thời tạo vector quan sát $[\int edt \quad e]^2$, trong đó, e biểu thị sai số giữa tốc độ đặt và tốc độ thực của động cơ Dc. Kết quả thu được mô hình điều khiển ổn định tốc độ động cơ DC với bộ điều khiển thích nghi thông minh trên cơ sở sử dụng thuật toán học tăng cường như hình 4.

Thiết kế cấu trúc và tham số của tác nhân TD3:

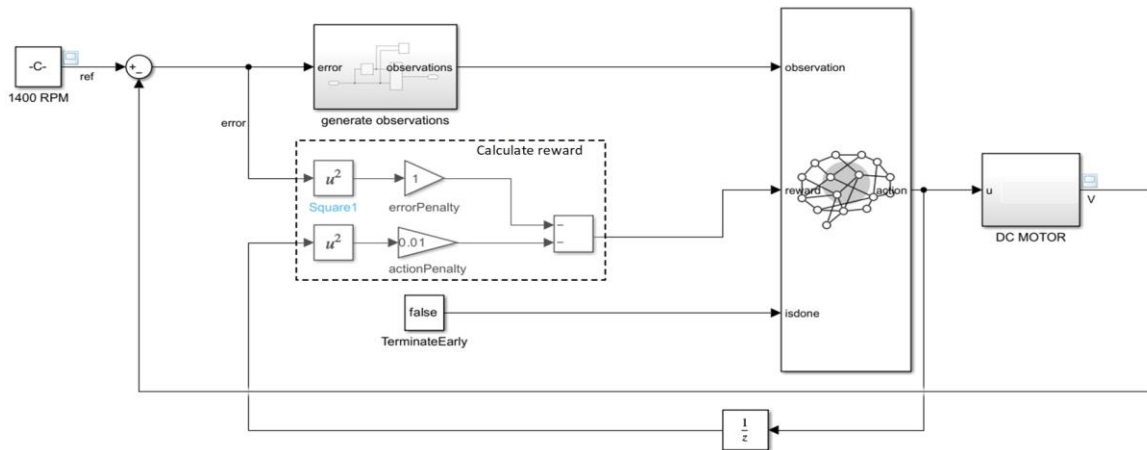
Tác nhân TD3 được xây dựng với hai mạng nơ-ron Critic với tham số độc lập để mỗi Critic có thể tìm hiểu về độ lợi tỉ lệ và độ lợi tích phân một cách riêng lẻ, vì chúng có giá trị rất khác nhau trong quá trình chỉnh định tham số của hệ thống. Mạng Critic được xây dựng với cấu trúc hai đầu vào và một đầu ra. Đầu vào là trạng thái quan sát và hành động, đầu ra là giá trị ước lượng.

Bộ điều khiển PI lúc này được mô hình hóa như một mạng nơ-ron, trong đó, lớp *fully connected layer* với các quan sát sai lệch điều khiển e và tích phân của e :

$$u = \left[\int edt \quad e \right] * [K_i \quad K_p]^T \quad (3)$$

trong đó:

- u : Đầu ra của mạng nơ-ron tác nhân;
- K_p và K_i : Giá trị tuyệt đối của các trọng số trong mạng nơron;
- e : Sai lệch giữa tốc độ tham chiếu và tốc độ thực tế của động cơ.



Hình 4. Mô hình môi trường huấn luyện tác nhân RL.

Hàm reward và các tham số huấn luyện:

Quá trình học của tác nhân được mô phỏng thành từng tập, mỗi tập bao gồm các bước lặp. Ở mỗi bước lặp, môi trường sẽ gửi về tác nhân điểm thưởng r_t được xác định như sau:

$$r(t) = -((v_{ref} - v)^2(t) + 0.01u^2(t)) \quad (4)$$

Mỗi tập sẽ diễn ra đến khi có tín hiệu hủy (vận tốc của động cơ lớn hơn 1400 RPM). Sau khi kết thúc một lần huấn luyện thuật toán, tổng phần thưởng tích lũy từ trạng thái đầu tiên đến trạng thái cuối cùng. Quá trình huấn luyện thuật toán TD3 trong nghiên cứu này được nhóm tác giả thực hiện với các tham số chính như sau:

Tham số	Giá trị
Số tập tối đa	1000
Số bước lặp tối đa trong mỗi tập	100
Tiêu chí hủy tập	Vận tốc của động cơ lớn hơn 1400 RPM

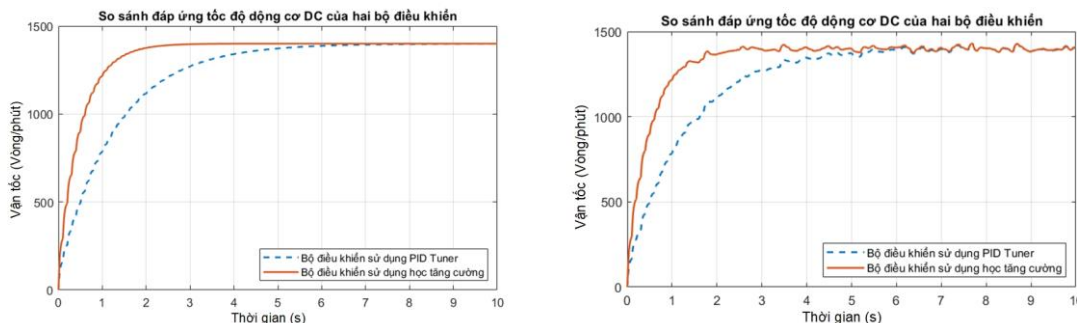
Tác nhân được huấn luyện bằng cách sử dụng hàm *train* trong thư viện **Reinforcement Learning Toolbox**.

4. KẾT QUẢ NGHIÊN CỨU

Trong phần này, nhóm tác giả thực hiện mô phỏng trên Matlab/Simulink để đánh giá chất lượng của bộ điều khiển thích nghi thông minh trên cơ sở thuật toán học tăng cường so với bộ điều khiển PI truyền thống ứng dụng trong bài toán điều khiển động cơ DC.

Đáp ứng tốc độ của động cơ DC của hai bộ điều khiển với tín hiệu vào là hằng số thể hiện trong hình 5. Khi hệ thống chứa nhiễu trắng (hình 5-b) (Để mô phỏng nhiễu trắng nhóm tác giả sử dụng khối *Band-Limited White Noise* trong Matlab/Simulink với *Noise power = 0.01*), đáp ứng của hệ thống thay đổi liên tục trong quá trình ổn định tốc độ so với giá trị đặt là 1400 vòng/phút (RPM). Kết quả cho thấy, cả 2 bộ điều khiển đều đạt chất lượng điều khiển rất tốt. Tuy nhiên, bộ điều khiển dựa trên thuật toán học tăng cường có phản hồi tốt hơn so với bộ điều khiển PI được chỉnh định tham số

thủ công sử dụng công cụ PID Tuner, vì nó có thời gian đáp ứng và thời gian xác lập ngắn hơn. Cụ thể các đặc tính của hệ thống thu được từ hai bộ điều khiển thể hiện trong bảng 2 và bảng 3.



a) Đáp ứng của hệ thống khi không có nhiễu trắng (white noise). b) Đáp ứng của hệ thống khi có nhiễu trắng (white noise).

Hình 5. Đáp ứng tốc độ động cơ DC của hai bộ điều khiển.

Bảng 2. Đáp ứng của hệ thống khi không có nhiễu trắng.

	Thời gian tăng	Thời gian tắt dần	Thời gian xác lập	Độ vọt lố	Đỉnh
Bộ điều khiển PI sử dụng Tuner	2.8475	4.9343	4.9343	0.00031487	1399.4
Bộ điều khiển sử dụng học tăng cường	1.0974	1.9246	1.9246	0.00017112	1400

Bảng 3. Đáp ứng của hệ thống khi có nhiễu trắng.

	Thời gian tăng	Thời gian tắt dần	Thời gian xác lập	Độ vọt lố	Đỉnh
Bộ điều khiển PI sử dụng Tuner	2.8244	6.2541	6.2541	1.5535	1428.8
Bộ điều khiển sử dụng học tăng cường	1.1094	2.1712	2.1712	1.319	1431

Kết quả bảng 2 và 3 cho thấy, khi có nhiễu trắng các đặc tính phản hồi của bộ điều khiển PID truyền thống bị thay đổi rất nhiều so với khi không có nhiễu, còn bộ điều khiển dựa trên thuật toán RL vẫn tương đối ổn định. Do đó, có thể kết luận rằng, giải pháp dựa trên thuật toán học tăng cường cho phép duy trì độ tin cậy và độ ổn định cao.

5. KẾT LUẬN

Trong bài báo này, nhóm tác giả đề xuất một cách tiếp cận mới trong bài toán điều khiển. Bằng cách sử dụng bộ điều khiển thông minh thích nghi dựa trên thuật toán học tăng cường, nhóm tác giả cung cấp khả năng điều chỉnh tự động và liên tục các tham số của bộ điều khiển. Để chứng minh tính đúng đắn của phương pháp tiếp cận, nhóm tác giả đã thực hiện mô phỏng hệ điều khiển tốc độ động cơ DC thích nghi thông minh bằng công cụ Matlab - Simulink. Các kết quả thu được cho thấy, bộ điều khiển trên cơ sở thuật toán học tăng cường có hiệu suất điều khiển cao, đã đáp ứng tốt với các tính năng như: ổn định với sự thay đổi thông số của hệ thống khi có nhiễu. Kết quả thu được trong nghiên cứu này đặt nền tảng cho các nghiên cứu trong tương lai trong lĩnh vực điều khiển thích nghi và tối ưu hệ thống điều khiển. Trong những nghiên cứu tiếp theo, nhóm tác giả sẽ hướng tới tối ưu thuật toán và hướng tới nghiên cứu triển khai trên phần cứng để có thể áp dụng cho các bài toán điều khiển hệ thống có sử dụng động cơ DC hoặc các đối tượng khác.

TÀI LIỆU THAM KHẢO

- [1]. Girirajkumar, S. M., Kumar, A. A., & Anantharaman, N. "Tuning of a PID controller for a real time industrial process using particle swarm optimization". International Journal of Computer Applications, 1(7), 35-40, (2010).
- [2]. Cominos P, Munro N. "Pid controllers: recent tuning methods and design to specification". IEEE Proc D; 149(1):46–53, (2002).
- [3]. Acosta GG, Mayosky MA, Catalfo JM. "An expert pid controller uses refined ziegler and nichols rules and fuzzy logic ideas". Appl Intell; 4(1): 53–66, (1994).
- [4]. Porter B, Jones AH. "Genetic tuning of digital pid controllers". Electron Lett; 28(9): 843–4, (1992).
- [5]. Acosta G, Todorovich E. "Genetic algorithms and fuzzy control: a practical synergism for industrial applications". Comput Ind;52(2): 183–95, (2003).
- [6]. Zeng G-Q, Xie X-Q, Chen M-R, Weng J. "Adaptive population extremal optimization-based pid neural network for multivariable nonlinear control systems". Swarm Evol Comput; 44: 320–34, (2019).
- [7]. Richard S. Sutton and Andrew G. Barto., "Reinforcement Learning: An Introduction", The MIT Press Cambridge, Massachusetts London, England, (2016).
- [8]. Li, Y. "Deep Reinforcement Learning: An Overview", (2017). ArXivabs/1701.07274.
- [9]. TOMIN, N., KURBATSKY, V., & GULIYEV, H. "Intelligent control of a wind turbine based on reinforcement learning". In 2019 16th Conference on Electrical Machines, Drives and Power Systems (ELMA) (pp. 1-6). IEEE, (2019).
- [10]. Fan J, Wang Z, Xie Y, Yang Z "A theoretical analysis of deep Q-learning. In: Learning for Dynamics and Control". PMLR, pp 486–489, (2020).
- [11]. Zheng, J., Kurt, M. N., & Wang, X. "Integrated actor-critic for deep reinforcement learning". In Artificial Neural Networks and Machine Learning–ICANN 2021: 30th International Conference on Artificial Neural Networks, Bratislava, Slovakia, September 14–17, 2021, Proceedings, Part IV 30 (pp. 505-518). Springer International Publishing, (2021).
- [12]. Tan, H. "Reinforcement Learning with Deep Deterministic Policy Gradient". In 2021 International Conference on Artificial Intelligence, Big Data and Algorithms (CAIBDA) (pp. 82-85). IEEE, (2021).
- [13]. Zhang F, Li J, Li Z "A td3-based multi-agent deep reinforcement learning method in mixed cooperation-competition environment". Neurocomputing 411: 206–215, (2020).
- [14]. Johnson M.A. and M.H. Moradi. "Chapter 8, in: PID Control - New Identification and Design Methods", pp. 297-337. Springer-Verlag London Limited. ISBN-10: 1-85233-702-8, (2005).
- [15]. Kwok, D.P. and P. Wang. "Fine-tuning of classical PID Controllers based on Genetic Algorithms". IEEE Inter. Workshop on Emerging Technologies and Factory Automation, pp. 37-43, (1992).
- [16]. Jones A.H. and P.B.M. Oliveira. "Genetic Auto-tuning of PID Controllers". IEEE Conf. Publ. No. 414, 12-14 Sep 1995, pp. 141-145, (1995).

ABSTRACT

Reinforcement learning algorithm for adaptive intelligent control

Nowadays, with the development of science and technology, control objects are increasingly complex, have high nonlinearities and large uncertainties, making traditional classic control algorithms no longer effective. That leads to the construction of unknown structures and parameters and requires advanced control techniques. To solve control problems with unknown elements in the dynamic models of the control object, the intelligent adaptive control method based on reinforcement learning algorithm is capable of automatically adjusting the parameters of the controller, proposed by the authors in this article. The effectiveness and feasibility of the proposed method are verified through practical simulation. The obtained comparative simulation results confirm that the proposed controller is robust, adaptive, and has high control performance.

Keywords: AI; Reinforcement learning; Intelligent control; Data-driven control; Adaptive control; PID.