



Nghiên cứu tối ưu hóa các tham số bộ điều khiển PID trong hệ truyền động động cơ một chiều trên cơ sở ứng dụng thuật toán di truyền tự thích nghi

Nguyễn Thế Hồng^{1*}, Phạm Trung Dũng²

¹Viện Tên lửa, Viện Khoa học và Công nghệ quân sự

²Viện Tên lửa và Kỹ thuật điều khiển, Học viện Kỹ thuật Quân sự

*Email: thehong.mta@gmail.com

Tóm tắt

Bộ điều khiển PID là bộ điều khiển được sử dụng rộng rãi trong các hệ truyền động điện, tuy nhiên hiệu năng phụ thuộc mạnh vào việc lựa chọn các tham số K_p , K_i , K_d . Các phương pháp điều chỉnh kinh điển thường khó đạt chất lượng tối ưu khi hệ vận hành trong điều kiện tải biến đổi hoặc phi tuyến. Việc áp dụng thuật toán di truyền giúp cải thiện đáng kể thời gian đáp ứng và giảm vọt lố so với cách tinh chỉnh kinh điển. Tuy nhiên, khi sử dụng thuật toán di truyền truyền thống, thuật toán dễ bị hội tụ sớm vào cực trị cục bộ. Để khắc phục điều này, bài báo đề xuất áp dụng thuật toán di truyền tự thích nghi để tối ưu bộ điều khiển PID cho hệ truyền động động cơ một chiều. Kết quả mô phỏng cho thấy PID tối ưu bằng thuật toán đề xuất có chất lượng tổng thể tốt hơn và ổn định hơn so với các bộ điều khiển PID truyền thống và PID điều chỉnh bằng GA thông thường.

Từ khóa: Động cơ DC, PID, Tự thích nghi, Thuật toán di truyền.

Abstract

Proportional–Integral–Derivative (PID) controllers are extensively applied in electric drive systems due to their simple structure and ease of implementation. However, the control performance strongly depends on the appropriate tuning of the controller parameters K_p , K_i , and K_d . Classical tuning techniques often fail to provide satisfactory performance when the system operates under varying load conditions or exhibits nonlinear characteristics. In recent years, genetic algorithms (GAs) have been employed to improve the transient response and reduce overshoot compared to conventional tuning methods. Nevertheless, standard GAs suffer from premature convergence, which may lead to suboptimal solutions. To overcome this drawback, this paper proposes an adaptive genetic algorithm for the optimal tuning of a PID controller applied to a DC motor drive system. The adaptive mechanism dynamically adjusts the genetic operators to enhance population diversity and avoid convergence to local optima. Simulation results validate the effectiveness of the proposed approach, demonstrating improved transient performance, reduced overshoot, and enhanced robustness compared with both conventionally tuned PID controllers and PID controllers optimized using a standard genetic algorithm.

Keywords: DC motor, PID controller, self-adaptive, genetic algorithm.



1. ĐẶT VẤN ĐỀ

Động cơ điện một chiều (DC) đóng vai trò quan trọng trong nhiều hệ thống công nghiệp hiện đại nhờ khả năng điều khiển tốc độ và mô-men linh hoạt, hiệu suất cao, chi phí hợp lý và khả năng thích nghi tốt với tải khác nhau [1]. Với đặc tính phản hồi nhanh, mô-men khởi động lớn và khả năng điều chỉnh tốc độ chính xác bằng các cấu trúc điều khiển đơn giản, động cơ DC được ứng dụng rộng rãi trong robot công nghiệp, băng chuyền sản xuất và các hệ truyền động yêu cầu độ chính xác cao. Do đó, động cơ DC vẫn giữ vị trí thiết yếu trong điều khiển chuyển động và tự động hóa hiện đại.

Trong các hệ servo công nghiệp, sự thay đổi tham số, ma sát và nhiễu ngoại cảnh có thể gây ảnh hưởng đáng kể đến chất lượng điều khiển tốc độ [2]. Vì vậy, bài toán điều khiển tốc độ chính xác và ổn định của động cơ DC khi tải thay đổi luôn là mục tiêu quan trọng nhằm đảm bảo tính an toàn và độ tin cậy của hệ thống [3]. Trong số các chiến lược điều khiển hiện có, bộ điều khiển PID vẫn là lựa chọn phổ biến nhất trong công nghiệp; hơn 90% các vòng điều khiển sử dụng PID nhờ cấu trúc đơn giản, độ tin cậy cao và khả năng triển khai dễ dàng [4], [5]. Mặc dù vậy, chất lượng điều khiển phụ thuộc mạnh vào việc lựa chọn các tham số PID, và các phương pháp cổ điển thường chỉ cho kết quả gần đúng, dễ gây dao động và vọt lố [6].

Nhiều phương pháp hiệu chỉnh PID đã được phát triển, bao gồm các phương pháp cổ điển, điều khiển dựa trên mô hình, các phương pháp phi mô hình như logic mờ và mạng nơ-ron, cũng như các thuật toán metaheuristic [6], [7], [8]. Trong đó, các thuật toán metaheuristic như giải thuật di truyền (GA), tối ưu bầy đàn (PSO) hay tiến hóa vi phân (DE) cho thấy tiềm năng lớn nhờ khả năng tìm kiếm toàn cục và không yêu cầu mô hình chính xác. Tuy nhiên, thuật toán di truyền cổ điển vẫn tồn tại nhược điểm hội tụ sớm do sử dụng các tham số lai ghép và đột biến cố định, làm giảm tính đa dạng quần thể và dẫn đến nghiệm tối ưu cục bộ [9].

Để khắc phục những hạn chế này, một số nghiên cứu đã đề xuất thuật toán di truyền thích nghi (AGA), trong đó xác suất lai ghép và đột biến được điều chỉnh thích nghi trong quá trình tiến hóa của quần thể [9]. Dưới đây là bảng tổng hợp các phương pháp hiệu chỉnh thích nghi tham số của GA:

TT	AGA điển hình	Ưu điểm	Nhược điểm
1.	Xác suất lai ghép/đột biến được điều chỉnh đồng thời dựa trên sự tiến hóa	Cải thiện tính linh hoạt thuật toán, có thể tích hợp kiến thức chuyên gia vào bài toán thông qua việc thiết kế quy luật biến	Việc lựa chọn quy luật biến thiên của xác suất lai ghép và xác suất đột biến mang tính chủ quan, phụ thuộc vào



TT	AGA điển hình	Ưu điểm	Nhược điểm
	của quần thể hoặc hàm đặt trước [10], [11].	đổi các tham số của GA; dễ dàng tích hợp vào bài toán vì quy luật thích nghi đã biết trước nên có thể dễ dàng hiện thực hóa.	từng bài toán cụ thể.
2.	Xác suất lai ghép/đột biến được thích nghi theo entropy của quần thể [12], [13], [14].	Chiến lược thích nghi có cơ sở vững chắc (entropy của quần thể). Do đó việc cân bằng giữa khả năng khai thác và khám phá của thuật toán được thực hiện một cách có hệ thống.	Chi phí tính toán lớn; entropy đôi khi không phản ánh hoàn toàn chính xác trạng thái hội tụ của quần thể; các nghiên cứu theo hướng này thường không áp dụng cho bài toán điều khiển.
3.	Xác suất lai ghép/đột biến được thích nghi theo luật mờ [15], [16].	Xử lý phi tuyến tốt, thích nghi linh hoạt.	Thiết kế hệ mờ để phụ thuộc hoàn toàn vào kinh nghiệm chuyên gia; tăng chi phí tính toán và độ phức tạp cài đặt.
4.	Kết hợp các thuật toán metaheuristic [17], [18].	Tận dụng được thế mạnh bổ trợ giữa hai cơ chế tối ưu: GA có khả năng khai phá không gian nghiệm toàn cục tốt, trong khi các thuật toán metaheuristic khác như PSO hội tụ nhanh về vùng nghiệm tối ưu.	Tăng độ phức tạp tính toán; cấu trúc lai thường được xây dựng theo từng bài toán cụ thể nên khó chuẩn hóa thành một mô hình tổng quát.
5.	Xác suất đột biến/lai ghép thích nghi theo độ dốc của hàm thích nghi [19], [20].	Thuật toán dễ hiểu, dễ cài đặt, bám sát tiến trình hội tụ, thường cải thiện được độ tin cậy và chất lượng nghiệm so với GA tham số cố định.	Giai đoạn đầu có gradient nhỏ, dẫn đến khó thoát khỏi cực trị địa phương; chất lượng thuật toán phụ thuộc mạnh vào độ nhiễu của hàm thích nghi.
6.	Các tham số của GA tự thích nghi bằng cách đưa chúng vào cấu trúc gene của quần thể [21], [22].	Các tham số tự tiến hóa mà không cần luật thủ công; tính linh hoạt cao và phù hợp với bài toán không biết rõ tham số tối ưu; khả năng thoát tối ưu cục bộ tốt hơn GA chuẩn.	Tăng khối lượng tính toán do tăng kích thước gene; hiện nay, có rất ít nghiên cứu áp dụng cho bộ điều khiển PID của động cơ DC.

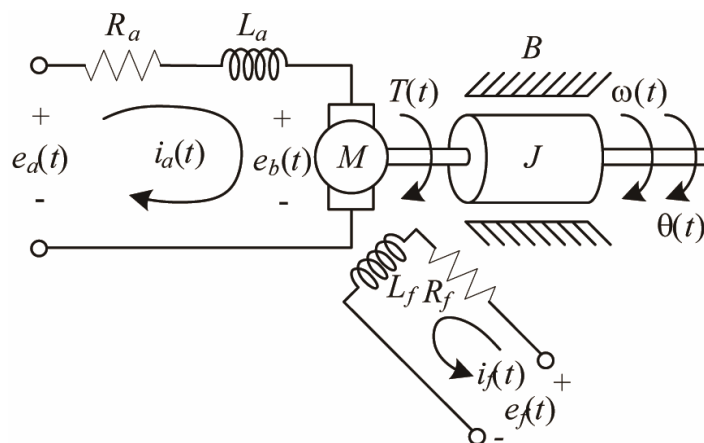
Bảng 1. Tổng hợp các hướng thực hiện thuật toán di truyền thích nghi điển hình

Từ bảng trên có thể thấy phần lớn các AGA hiện tại được điều chỉnh bằng luật bên ngoài và không tiến hóa với vai trò như một gene. Trong khi đó hướng tiếp cận AGA theo mô hình tự thích nghi có tiềm năng cao trong duy trì đa dạng di truyền và khả năng tối ưu toàn cục mới chủ yếu dừng lại ở việc thử nghiệm đối với các hàm số học điển hình; vẫn rất hiếm được áp dụng trong tối ưu PID cho động cơ DC, đặc biệt dưới điều kiện tải biến thiên. Do đó, bài báo đề xuất thuật toán di truyền tự thích nghi SAGA (Self-Adaptive Genetic Algorithm) để giải quyết hiệu quả bài toán điều khiển tốc độ động cơ DC trong môi trường công nghiệp thông qua tối ưu bộ hệ số của bộ điều khiển PID. Kết quả bài báo sẽ được kiểm chứng thông qua mô phỏng trên phần mềm MATLAB.

2. GIẢI QUYẾT VẤN ĐỀ

2.1. Mô hình động cơ DC

Động cơ điện một chiều (DC) có thể được phân loại theo mô hình toán học (dựa trên phương thức kích từ trường) thành bốn loại chính: động cơ DC kích từ độc lập, động cơ DC kích từ song song, động cơ DC kích từ nối tiếp, và động cơ DC nam châm vĩnh cửu [23]. Trong số các loại trên, động cơ DC kích từ độc lập được xem là loại được sử dụng phổ biến nhất trong các hệ truyền động công nghiệp hiện nay nhờ ưu điểm có thể điều khiển độc lập điện áp phản ứng và dòng kích từ để đáp ứng yêu cầu điều khiển chính xác trong nhiều ứng dụng công nghiệp [24]. Sơ đồ cấu trúc của một động cơ điện DC kích từ độc lập được cho trên Hình 1 [25], [26].



Hình 1. Sơ đồ cấu trúc động cơ DC kích từ độc lập

Trong đó:

$e_a(t)$ là điện áp cấp vào phần ứng; $i_a(t)$ là dòng điện phần ứng; R_a là điện trở phần ứng;
 L_a là điện cảm phần ứng; $e_b(t)$ là sức điện động phản kháng;



M là phần điện-cơ của động cơ; $T(t)$ là mô-men điện từ sinh ra; J là mô-men quán tính; B là hệ số ma sát nhớt; $\omega(t)$ là tốc độ góc trục động cơ; $\theta(t)$ là vị trí góc trục động cơ; $e_f(t)$ là điện áp kích từ cấp cho cuộn kích từ; $i_f(t)$ là dòng kích từ; R_f là điện trở cuộn kích từ; L_f là điện cảm cuộn kích từ.

Theo định lý Krichhoff, ta có phương trình cho điện áp mạch phần cứng [27]:

$$e_a(t) = R_a i_a(t) + L_a \frac{di_a(t)}{dt} + e_b(t) \quad (1)$$

trong đó sức điện động cảm ứng $e_b(t) = K_e \Phi(t) \omega(t) = K_e k_f i_f(t) \omega(t)$ với $\Phi(t)$ là từ thông kích từ.

Ta có phương trình điện cho mạch kích từ [28]:

$$e_f(t) = R_f i_f(t) + L_f \frac{di_f(t)}{dt} \quad (2)$$

Phương trình cơ học của động cơ DC xuất phát từ phương trình động lực quay của Newton. Theo [29] ta có phương trình mô tả cân bằng mô-men:

$$J \frac{d\omega(t)}{dt} + B\omega(t) = T(t) \quad (3)$$

trong đó $T(t)$ là mô-men điện từ sinh ra và được xác định bởi:

$$T(t) = K_t \Phi(t) i_a(t) = K_t k_f i_f(t) i_a(t) \quad (4)$$

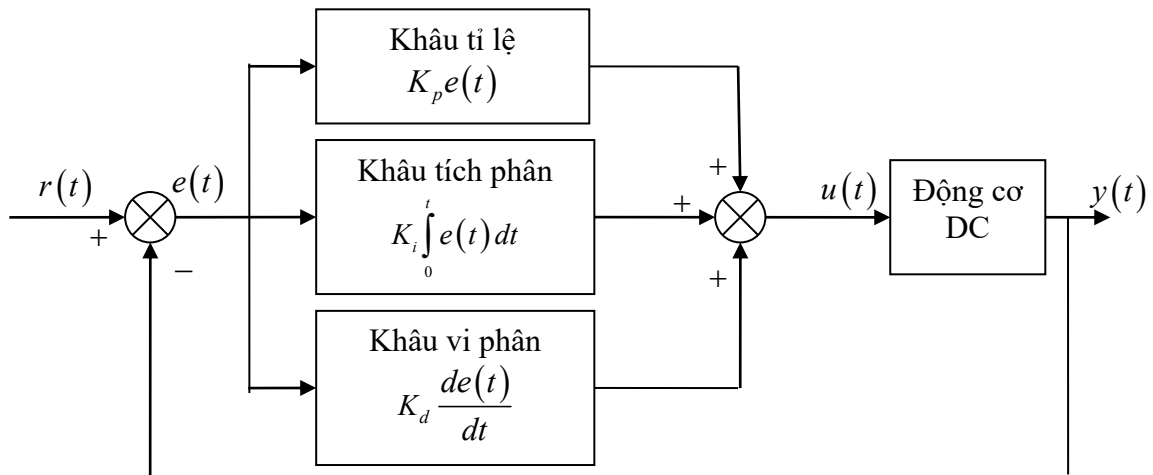
Thực hiện biến đổi Laplace các phương trình trên ta nhận được hàm truyền vòng hở của động cơ DC như sau:

$$G_m(s) = \frac{K_t}{(L_a s + R_a)(J s + B) + K_e K_t} \quad (5)$$

2.2. Thiết kế bộ điều khiển PID bởi thuật toán SAGA

Bộ điều khiển PID có ba thành phần: tác động tỉ lệ (P) để giảm nhanh sai lệch tức thời, tác động tích phân (I) để loại bỏ sai lệch tĩnh ở trạng thái xác lập và thành phần đạo hàm (D) phản ứng với tốc độ biến thiên của sai lệch, giúp cải thiện ổn định của hệ thống [5].

Dưới đây là sơ đồ cấu trúc của vòng điều khiển động cơ DC sử dụng bộ điều khiển PID [5]:



Hình 2. Sơ đồ cấu trúc của vòng điều khiển động cơ DC sử dụng bộ điều khiển PID

Để thiết kế các hệ số K_p , K_i , K_d của bộ điều khiển PID có nhiều phương pháp khác nhau [5]. Tuy nhiên, như đã trình bày ở trên, bài báo sẽ sử dụng thuật toán SAGA để tối ưu bộ hệ số K_p , K_i , K_d . Trong đó, tham số được thích nghi là xác suất đột biến của thuật toán SAGA. GA cổ điển có rất nhiều tham số khác nhau ảnh hưởng tới chất lượng tổng thể của thuật toán. Với mục đích là khắc phục nhược điểm hội tụ sớm và cực trị cục bộ của GA cổ điển thì nhiều nghiên cứu kinh điển về GA đã chỉ ra rằng việc điều chỉnh động tham số đột biến giúp duy trì đa dạng di truyền và giảm nguy cơ hội tụ sớm, đồng thời cải thiện khả năng tiếp cận nghiệm toàn cục [30], [31]. Tuy nhiên, thay vì coi xác suất đột biến là một tham số ngoại lai đặt trước, nó sẽ được đưa vào cấu trúc gene và tiến hóa cùng quần thể để tạo ra một cơ chế thích nghi mang tính bản chất: mỗi cá thể không chỉ mang thông tin về lời giải (ở đây là bộ tham số PID) mà còn mang chiến lược đột biến phù hợp nhất với bản thân nó. Nếu một giá trị xác suất đột biến P_m giúp cá thể tạo ra thế hệ con chất lượng tốt hơn, chiến lược đột biến đó sẽ được bảo tồn và lan truyền. Khi đó, quần thể tự hình thành sự cân bằng nội tại giữa khai thác (duy trì cấu trúc lời giải tốt) và khám phá (mở rộng không gian tìm kiếm), mà không cần một quy tắc điều chỉnh thủ công. Dưới đây là các bước thực hiện thuật toán SAGA có xác suất đột biến tự thích nghi để tối ưu bộ ba hệ số K_p , K_i , K_d :

i) Mô tả nhiệm vụ sắc thể và hàm thích nghi

Giả sử mỗi cá thể trong quần thể được mã hóa dạng $X = [K_p, K_i, K_d, P_m]$, trong đó K_p , K_i , K_d là tham số bộ điều khiển PID; P_m là xác suất đột biến riêng của từng cá thể (gene tự thích nghi).



Hàm thích nghi được xây dựng căn cứ vào các mục tiêu cần tối ưu. Trong nhiều bài toán điều khiển nói chung, tổng tích lũy của tích giữa thời gian với sai số giữa đầu ra mong muốn với đầu ra thực tế của hệ thống (ITAE) và tổng tích lũy bình phương tín hiệu điều khiển thường được chọn làm chỉ tiêu chất lượng của hệ thống [6]. Ngoài ra, đối với đáp ứng bước trong miền thời gian của động cơ DC, ta còn quan tâm tới độ vọt lố và thời gian xác lập bởi vì nó ảnh hưởng trực tiếp tới độ an toàn và tính đáp ứng nhanh của hệ thống. Bởi vì ta đều mong muốn các tham số đó càng nhỏ càng tốt; trong khi đó, mục tiêu của SAGA là khiến hàm thích nghi càng lớn càng tốt. Do đó, hàm thích nghi được chọn dưới dạng sau:

$$Fitness = \frac{1}{w_1 \cdot \int t \cdot |e(t)| dt + w_2 \int u^2(t) dt + w_3 \cdot M_p + w_4 \cdot t_s} \quad (6)$$

trong đó $e(t)$ là sai số giữa đầu ra hệ thống và đầu ra mong muốn, $u(t)$ là tín hiệu điều khiển động cơ DC (đầu ra bộ điều khiển PID), M_p là độ vọt lố, t_s là thời gian xác lập. Các trọng số $w_1 = 1$, $w_2 = 0.01$, $w_3 = 1$, $w_4 = 1$ quyết định vai trò của từng thành phần trong quá trình tối ưu.

ii) Thuật toán SAGA

Bước 1: Khởi tạo

Khởi tạo ngẫu nhiên các bộ hệ số $X^k = [K_p^k, K_i^k, K_d^k, P_m^k]$ nằm trong dải cho phép, trong đó $k = 1..N$ (N là kích thước quần thể). Ứng với một bộ hệ số đó là một cá thể.

Bước 2: Đánh giá quần thể

Đối với mỗi cá thể X^k ta sẽ gán bộ 3 tham số $K_p^k, K_i^k, K_d^k, P_m^k$ cho bộ điều khiển PID. Sau đó tiến hành mô phỏng đáp ứng của hệ với tín hiệu đặt để từ đó tính giá trị của hàm thích nghi.

Bước 3: Thực hiện vòng lặp tiến hóa

Bước 3.1: Chọn lọc

Chọn lọc theo phương pháp sắp hạng tuyến tính. Phương pháp này chọn lọc theo thứ hạng của từng cá thể chứ không theo giá trị của hàm thích nghi nên ổn định hơn.

Bước 3.2: Lai ghép

Bài báo chọn phương pháp lai ghép $BLX - \alpha$ vì nó cân bằng tốt giữa khai thác giá trị tốt và khám phá vùng tiềm năng mới, rất phù hợp khi các tham số PID cần điều chỉnh chính xác nhưng vẫn phải vượt qua bẫy cực trị cục bộ. Để thực hiện lai ghép theo phương pháp này, ta chọn ngẫu nhiên hai cá thể bố mẹ lần lượt là $X_p^1 = [K_p^1, K_i^1, K_d^1, P_m^1]$ và



$X_p^2 = [K_p^2, K_i^2, K_d^2, P_m^2]$. Với mỗi gene $j \in [K_p, K_i, K_d, P_m]$ ta tính $I_j = C_{j_{\max}} - C_{j_{\min}}$, trong đó $C_{j_{\min}} = \min(j_{X^1}, j_{X^2})$ và $C_{j_{\max}} = \max(j_{X^1}, j_{X^2})$.

Khi đó miền $BLX - \alpha$ mở rộng được xác định như sau: $L_j = C_{j_{\min}} - \alpha \cdot I_j$ và $U_j = C_{j_{\max}} + \alpha \cdot I_j$. Hai cá thể con sẽ được tạo ngẫu nhiên trong khoảng (L_j, U_j) theo phân bố đều.

Bước 3.3: Đột biến thích nghi

Trong các phương pháp đột biến thì đột biến đa thức (Polynomial) là toán tử đột biến được thiết kế riêng cho phương pháp mã hóa gene bằng số thực, mô phỏng sự phân bố biến động theo “độ cong” điều khiển bằng tham số đột biến η_m (quy định mức độ đột biến). Đột biến đa thức đảm bảo tinh chỉnh tốt vùng nghiệm tiềm năng và đồng thời duy trì khả năng thoát khỏi cực trị cục bộ. Giả sử một gene $j \in [j_{\min}, j_{\max}]$, ta có hệ số biến đổi δ_q như sau:

$$\delta_1 = \frac{j - j_{\min}}{j_{\max} - j_{\min}}; \quad \delta_2 = \frac{j_{\max} - j}{j_{\max} - j_{\min}} \quad (7)$$

Khởi tạo một giá trị ngẫu nhiên đều $u \in (0,1)$. Nếu $u \leq 0,5$ thì:

$$\delta_q = \left(2u + (1-2u)(1-\delta_1)^{\eta_m+1}\right)^{\frac{1}{\eta_m+1}} - 1 \quad (8)$$

Nếu $u > 0,5$ thì:

$$\delta_q = 1 - \left(2(1-u) + 2(u-0,5)(1-\delta_2)^{\eta_m+1}\right)^{\frac{1}{\eta_m+1}} \quad (9)$$

Gene mới sau đột biến được xác định như sau:

$$j_{\text{new}} = j + \delta_q (j_{\max} - j_{\min}) \quad (10)$$

Tính thích nghi được thể hiện ở cấu trúc nhiễm sắc thể, tức là mỗi thể j có một xác suất đột biến P_{m_j} khác nhau. Các P_{m_j} này được sử dụng để đột biến chính ba gene còn lại của cá thể j đó. Do đó, mỗi cá thể có khả năng khám phá không gian nghiệm khác nhau. Qua các thể hệ, xác suất đột biến tốt kết hợp với các hệ số K_p, K_i, K_d tốt sẽ được chọn lọc và nhân bản. Ngoài ra, bản thân các xác suất đột biến cũng bị đột biến đa thức với một xác suất $P_{m_{\text{base}}} = 0,01$ đặt trước; đây chính là bước tạo ra các biến dị và tìm các xác suất đột biến P_{m_j} tốt hơn.

Bước 3.4: Kiểm tra điều kiện dừng

Ở bước này tiến hành kiểm tra điều kiện dừng của thuật toán. Nếu thỏa mãn, kết thúc thuật toán. Nếu không thỏa mãn, quay lại Bước 2 để đánh giá quần thể để vào Bước 3. Điều kiện dừng thường là khi quần thể tiến hóa đạt tới số thế hệ cực đại hoặc khi hàm thích nghi biến đổi nhỏ hơn một ngưỡng nào đó liên tục trong một số thế hệ đặt trước.



Bước 4: Kết quả

Chọn cá thể tốt nhất $X^* = [K_p^*, K_i^*, K_d^*, P_m^*]$. Dùng K_p^*, K_i^*, K_d^* làm tham số bộ điều khiển PID; P_m^* là xác suất đột biến tối ưu mà SAGA tự học được.

2.3. Mô phỏng, kiểm chứng

2.3.1. Khởi tạo các điều kiện đầu

Để có thể mô phỏng, ta cần khởi tạo các giá trị cho các hệ số có trong các phương trình mô tả mô hình động cơ DC cũng như các tham số của thuật toán SAGA.

Đối với động cơ DC kích từ độc lập, ta coi là dòng kích từ không đổi nên từ thông Φ không đổi và mạch kích từ có thể được loại khỏi mô hình. Ta chỉ xét tới tốc độ ω . Khi đó hệ phương trình (1) và (3) là đủ để mô tả mô hình của động cơ DC. Ta viết lại hệ (1), (3) như sau:

$$\begin{cases} e_a(t) = R_a i_a(t) + L_a \frac{di_a}{dt} + K_e \omega(t) \\ J \frac{d\omega}{dt} + B\omega(t) = K_t i_a(t) \end{cases} \quad (11)$$

Trong đó các tham được chọn theo các tham số đã được sử dụng rộng rãi trong các nghiên cứu điều khiển tốc độ động cơ DC. Cụ thể, các tham số gồm: $R_a = 1\Omega$, $L_a = 0,5H$, $J = 0,01kg.m^2$, $B = 0,1N.m.s / rad$, $K_t = 0,01N.m / A$, $K_e = 0,01V.s / rad$. Việc bộ tham số này trùng khớp với tham số được sử dụng trong nghiên cứu của Khine và đồng nghiệp [32]. Thay các hệ số này vào (5) ta nhận được hàm truyền hở từ $e_a(t)$ đến $\omega(t)$ như sau:

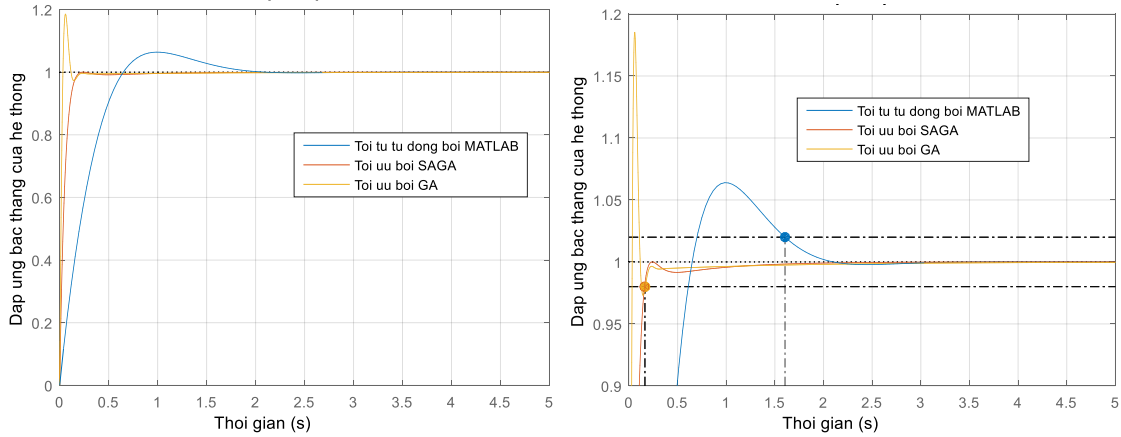
$$G_m(s) = \frac{0,01}{(0,5s + 1)(0,01s + 0,1) + 0,1^2} \quad (12)$$

Đối với thuật toán SAGA, kích thước quần thể được chọn là 10; số thế hệ tối đa là 50; chọn lọc sắp hạng tuyến tính với hệ số áp lực chọn lọc $\eta = 0,5$; lai ghép $BLX - \alpha$ với xác suất lai ghép là 0,85; đột biến đa thức với xác suất đột biến biên thiên trong đoạn từ 0,01 đến 0,3; các hệ số của bộ điều khiển PID biên thiên từ 0,01 đến 1000.

Kết quả khảo sát sẽ được so sánh với bộ điều khiển được tối ưu bằng GA cổ điển (xác suất đột biến cố định là 0,01) và công cụ tối ưu tự động trong MATLAB (hàm pidtune).

2.3.2. Kết quả mô phỏng

Ta có đáp ứng của động cơ với ba bộ điều khiển PID được thiết kế theo ba phương pháp khác nhau: Tối ưu tự động bởi MATLAB; tối ưu bởi GA cổ điển; tối ưu bởi SAGA đề xuất. Kết quả được mô tả dưới đây:



Hình 3. Đáp ứng của động cơ DC ứng ba bộ điều khiển PID

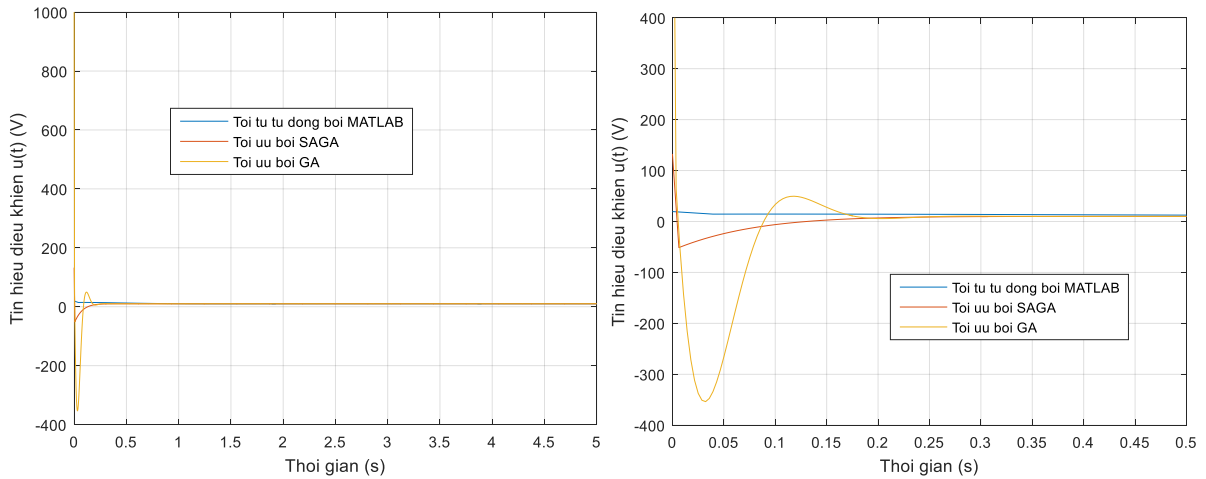
	Tối ưu tự động	GA	SAGA
Thời gian xác lập (s)	1,600	0,168	0,160
Độ quá chỉnh (%)	6,390	18,500	0,000
Thời gian tăng (s)	0,461	0,025	0,101

Bảng 2. Bảng tổng hợp giá trị một số tham số chất lượng của hệ thống trong miền thời gian

Từ Hình 3 và Bảng 2 ta nhận thấy đáp ứng của động cơ DC khi sử dụng bộ điều khiển PID được tối ưu bởi GA và SAGA có tính tác động nhanh cũng như thời gian xác lập đều tốt hơn khi sử dụng bộ điều khiển PID được tối ưu bởi tự động bởi MATLAB.

Khi xét riêng tham số là thời gian tăng thì bộ điều khiển PID tối ưu bởi GA thì tốt hơn bộ điều khiển PID tối ưu bởi SAGA (không đáng kể). Tuy nhiên, nó lại phải chịu một mức vọt lố rất lớn (lên tới 18,5%); trong khi đó bộ điều khiển PID tối ưu bởi SAGA giúp hệ thống đạt trạng thái xác lập nhanh hơn và không xuất hiện vọt lố.

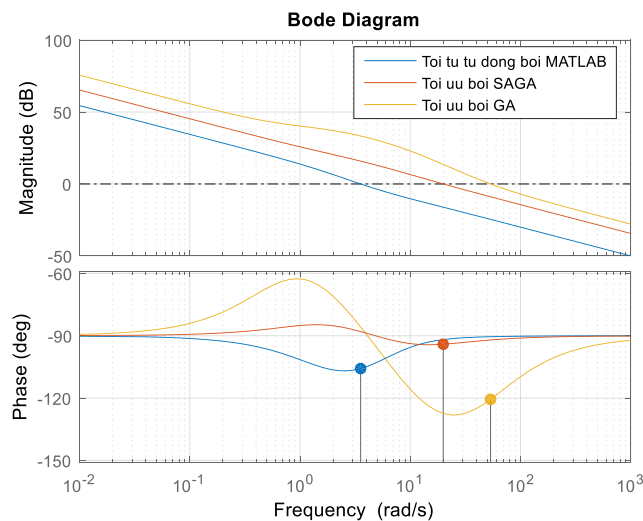
Để có thể đánh giá chính xác hơn, ta sẽ đánh giá tín hiệu điều khiển (đầu ra của bộ điều khiển PID) theo ba phương pháp. Dưới đây là kết quả tín hiệu điều khiển nhận được:



Hình 4. Tín hiệu điều khiển khi sử dụng ba bộ điều khiển PID

Từ Hình 4 ta có thể nhận thấy rằng tín hiệu điều khiển khi sử dụng bộ điều khiển PID tối ưu bởi GA biến thiên và dao động rất lớn ở giai đoạn bắt đầu điều khiển (biên độ dao động khoảng 400V trong khoảng 0,12s đầu). Điều này là điều không mong muốn bởi vì khi tín hiệu điều khiển lớn và dao động có thể gây ra hư hại đối với cơ cấu chấp hành, gây ra rung động cơ học, làm giảm chất lượng điều khiển nói chung. Trong khi đó tín hiệu điều khiển tạo bởi bộ điều khiển PID tối ưu bởi MATLAB (giảm gần tuyến tính từ 20V về giá trị điện áp điều khiển ổn định là 10V) và bộ điều khiển PID tối ưu bởi SAGA có mức độ biến thiên ít hơn (tăng từ -50V về giá trị điện áp điều khiển ổn định là 10V), dễ điều khiển hơn.

Để đánh giá tác động của bộ điều khiển PID tối tính ổn định của vòng điều khiển kín, ta có đồ thị Bode của hệ thống khi điều khiển bởi bộ điều khiển PID tối ưu theo ba phương pháp khác nhau như sau:



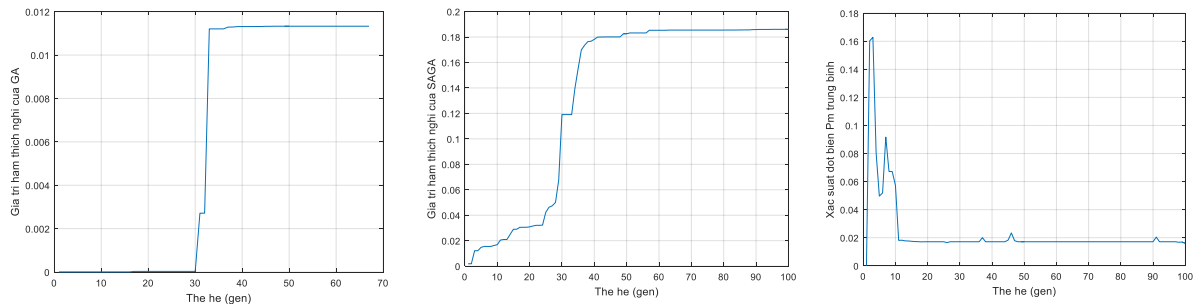
Hình 5. Đồ thị Bode của hệ thống trong miền tần số

	Tối ưu tự động	GA	SAGA
Độ dự trữ pha (deg)	74,2	59,4	85,9
Tần số cắt (rad/s)	3,55	53,5	19,9
Thời gian tăng (s)	0,461	0,025	0,101

Bảng 3. Bảng tổng hợp giá trị một số tham số chất lượng của vòng điều khiển kín trong miền tần số

Từ đồ thị Bode (Hình 5) và Bảng 3 cho thấy ba phương pháp điều chỉnh PID tạo ra những đặc trưng khác nhau về chất lượng điều khiển. Bộ PID tối ưu tự động mang lại tính ổn định cao và an toàn, nhưng băng thông thấp khiến hệ thống phản ứng còn chậm và ít nhạy với tín hiệu đặt. Phương pháp tối ưu GA cho băng thông lớn nhất, thể hiện tốc độ đáp ứng cao; tuy nhiên, dự trữ pha nhỏ làm giảm độ bền vững, khiến hệ nhạy hơn với nhiễu và dễ xuất hiện dao động hoặc vọt lố trong miền thời gian (Hình 3). Ngược lại, SAGA đạt được sự cân bằng tối ưu giữa hai tiêu chí tốc độ và ổn định: tần số cắt vẫn lớn hơn PID MATLAB, trong khi dự trữ pha là cao nhất trong ba cấu hình. Nhờ đó, hệ thống sử dụng PID tối ưu bằng SAGA có tính đáp ứng nhanh nhưng vẫn ổn định, đồng thời giữ tín hiệu điều khiển ở mức biên độ thấp. Kết quả này cho thấy SAGA là phương pháp tối ưu hóa hiệu quả và bền vững hơn cho bài toán điều khiển tốc độ động cơ DC

Để thấy rõ hơn ưu điểm của SAGA so với GA theo quan điểm của di truyền ta có biểu đồ mô tả sự biến thiên giá trị của hàm thích nghi và xác suất lai ghép trung bình của quần thể theo thế hệ như sau:



a) Giá trị hàm thích nghi khi tối ưu bởi GA b) Giá trị hàm thích nghi khi tối ưu bởi SAGA c) Xác suất đột biến trung bình quần thể khi tối ưu bởi SAGA

Hình 6. Sự biến thiên của hàm thích nghi

Ta nhận thấy hàm thích nghi của quần thể khi thực hiện thuật toán GA dừng lại sau 68 thế hệ và biến đổi rất ít. Điều đó cho thấy quần thể mất đi sự đa dạng gene và hội tụ vào cực



trị cục bộ và không thoát ra được (Hình 6a). Trong khi đó, hàm thích nghi của thuật toán SAGA tăng ổn định và tiến dần tới cực trị toàn cục (bộ tham số của bộ điều khiển PID). Thuật toán SAGA đạt được điều này nhờ sự thay đổi thích nghi của tham số xác suất đột biến. Từ Hình 6c ta thấy trong những thế hệ đầu, xác suất đột biến trung bình của quần thể biến động lớn. Tuy nhiên, từ thế hệ 11 trở đi thì xác suất đột biến trung bình gần như không có biến động. Điều đó chứng tỏ SAGA đã tìm được giá trị xác suất đột biến phù hợp cho quần thể; do đó, nó được giữ lại và nhân bản trong các thế hệ tiếp theo.

3. TRAO ĐỔI VÀ KẾT LUẬN

Như vậy, thuật toán SAGA đề xuất đã tìm được bộ tham số tối ưu cho bộ điều khiển PID. Bằng việc đưa tham số xác suất đột biến P_m vào cấu trúc gene và để cho P_m tự tiến hóa cùng với các tham số hệ số của bộ điều khiển thì thuật toán SAGA đã tránh được hiện tượng hội tụ sớm vào cực trị cục bộ giống như GA cổ điển. Khi số lượng cá thể trong quần thể và số thế hệ bị hạn chế thì hiện tượng hội tụ sớm càng dễ xảy ra.

Kết quả mô phỏng khi sử dụng bộ điều khiển PID để điều khiển động cơ DC cho thấy chất lượng tổng thể của hệ thống khi sử dụng bộ điều khiển PID được tối ưu bởi SAGA vượt trội hơn hẳn khi sử dụng các phương pháp khác nhau tối ưu tự động trong MATLAB hoặc tối ưu bởi GA cổ điển.

Chất lượng của GA còn phụ thuộc những tham số khác như xác suất lai ghép, hệ số áp lực chọn lọc. Tuy nhiên, bài báo này chưa thực hiện thích nghi đồng thời cả 3 tham số này. Đồng thời, nếu được kiểm chứng trên thực tế với động cơ DC thật thì sẽ tăng tính thuyết phục và ý nghĩa thực tiễn của thuật toán SAGA đề xuất. Đây chính là hướng mở để thực hiện các nghiên cứu tiếp theo để có đánh giá tổng quát hơn nữa về chất lượng của thuật toán SAGA.

TÀI LIỆU THAM KHẢO

- [1] A. Top, "Optimization-Based Tuning of PI Controller Parameters for DC Motor Speed Control," (in en), *Hittite Journal of Science and Engineering*, vol. 12, no. 2, pp. 81–90, June 2025, doi: 10.17350/HJSE19030000354.
- [2] I. S. Okoro and C. O. Enwerem, "Robust control of a DC motor," *Heliyon*, vol. 6, no. 12, 2020/12 2020, doi: 10.1016/j.heliyon.2020.e05777.
- [3] J. A. Prakosa, A. Gusrialdi, E. Kurniawan, A. D. Stotckaia, H. Adinanta, and Suryadi, "Experimentally robustness improvement of DC motor speed control optimization by H-infinity of mixed-sensitivity synthesis," *International Journal of Dynamics and Control*, vol. 10, no. 6, pp. 1968–1980, 2022/12/01 2022, doi: 10.1007/s40435-022-00956-y.
- [4] S. Gulgonul, *PID Tuning via Desired Step Response Curve Fitting*. 2025.



- [5] R. Borase, D. Maghade, S. Sondkar, and S. Pawar, "A review of PID control, tuning methods and applications," *International Journal of Dynamics and Control*, vol. 9, 06/01 2021, doi: 10.1007/s40435-020-00665-4.
- [6] R. H. Subrata, F. Gozali, and E. Djuana, "Computational and intelligent optimization tuning method for Pid controller," *Journal of Theoretical and Applied Information Technology*, vol. 100, no. 7, pp. 2007–2017, 2022.
- [7] A. Varshney, D. Gupta, and B. Dwivedi, "Speed response of brushless DC motor using fuzzy PID controller under varying load condition," *Journal of Electrical Systems and Information Technology*, vol. 4, no. 2, pp. 310–321, 2017/09/01/ 2017, doi: <https://doi.org/10.1016/j.jesit.2016.12.014>.
- [8] L. Dinh Quy *et al.*, "Metaheuristic Algorithms for PID Controller Parameters Tuning: Approaches and Results," 2025, pp. 380–389.
- [9] S. Wenjuan, Q. Su, H. Yuan, and Y. Chen, "Optimization Performance Analysis for Adaptive Genetic Algorithm with Nonlinear Probabilities," *International Journal of Advanced Computer Science and Applications*, vol. 13, 01/01 2022, doi: 10.14569/IJACSA.2022.0130645.
- [10] J. Zhao and M. Xi, "Self-Tuning of PID Parameters Based on Adaptive Genetic Algorithm," *IOP Conference Series: Materials Science and Engineering*, vol. 782, no. 4, p. 042028, 2020/03/01 2020, doi: 10.1088/1757-899X/782/4/042028.
- [11] L.-Y. Chang and H.-C. Chen, "Tuning of fractional PID controllers using adaptive genetic algorithm for active magnetic bearing system," *WSEAS TRANSACTIONS on SYSTEMS*, vol. 8, pp. 158–167, 01/01 2009.
- [12] L. Yang, D. Widyantoro, T. Ioerger, and J. Yen, *An entropy-based adaptive genetic algorithm for learning classification rules*. 2001, pp. 790–796 vol. 2.
- [13] N. Stark, G. F. Minetti, and C. Salto, "A new strategy for adapting the mutation probability in genetic algorithms," 2012.
- [14] B. Zheng, X. Li, Z. Tian, and L. Meng, "Optimization Method for Distributed Database Query Based on an Adaptive Double Entropy Genetic Algorithm," *IEEE Access*, vol. 10, pp. 4640–4648, 2022.
- [15] J.-p. Cao, S.-K. Jeong, and Y.-M. Jung, "Fuzzy logic controller design with unevenly-distributed membership function for high performance chamber cooling system," *Journal of Central South University*, vol. 21, no. 7, pp. 2684–2692, 2014. [Online]. Available: <https://doi.org/10.1007/s11771-014-2230-y>.
- [16] M. Vannucci and V. Colla, "Fuzzy adaptation of crossover and mutation rates in genetic algorithms based on population performance," *J. Intell. Fuzzy Syst.*, vol. 28, no. 4, pp. 1805–1818, 2015.
- [17] Y.-T. Kao and E. Zahara, "A hybrid genetic algorithm and particle swarm optimization for multimodal functions," *Applied Soft Computing*, vol. 8, no. 2, pp. 849–857, 2008/03/01/ 2008, doi: <https://doi.org/10.1016/j.asoc.2007.07.002>.
- [18] E. M. El-Gendy, M. M. Saafan, M. S. Elksas, S. F. Saraya, and F. F. G. Areed, "Applying hybrid genetic–PSO technique for tuning an adaptive PID controller used in a chemical process," *Soft Computing*, vol. 24, no. 5, pp. 3455–3474, 2020/03/01 2020, doi: 10.1007/s00500-019-04106-z.
- [19] Z. Novikov and E. Sopov, "Development and analysis of adaptive mutation techniques in genetic algorithms," *ITM Web of Conferences*, vol. 72, 02/13 2025, doi: 10.1051/itmconf/20257205003.
- [20] J. Tang, D. Liu, Q. Wang, J. Li, and J. Sun, "Probabilistic Chain-Enhanced Parallel Genetic Algorithm for UAV Reconnaissance Task Assignment," *Drones*, vol. 8, no. 6, p. 213, 2024. [Online]. Available: <https://www.mdpi.com/2504-446X/8/6/213>.



- [21] S. Meyer-Nieberg and H.-G. Beyer, "Self-Adaptation in Evolutionary Algorithms," vol. 54, 2007, pp. 47–75.
- [22] J. M. Oliver, T. Kipouros, and A. M. Savill, "A Self-adaptive Genetic Algorithm Applied to Multi-Objective Optimization of an Airfoil," in *EVOLVE - A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation IV*, Heidelberg, M. Emmerich *et al.*, Eds., 2013// 2013: Springer International Publishing, pp. 261–276.
- [23] S. Sami, Z. Obaid, M. Muhssin, and A. Hussain, "Detailed modelling and simulation of different DC motor types for research and educational purposes," *International Journal of Power Electronics and Drive Systems (IJPEDS)*, vol. 12, p. 703, 06/01 2021, doi: 10.11591/ijpeds.v12.i2.pp703-714.
- [24] E. Arévalo, R. Herrera Hernández, D. Katselis, C. Reusser, and R. Carvajal, "On Modelling and State Estimation of DC Motors," *Actuators*, vol. 14, p. 160, 03/21 2025, doi: 10.3390/act14040160.
- [25] D. Puangdownreong, "Current Search: Performance Evaluation and Application to DC Motor Speed Control System Design," *Intelligent Control and Automation*, vol. 04, pp. 42–54, 01/01 2013, doi: 10.4236/ica.2013.41007.
- [26] M. Kuczmann, "Review of DC Motor Modeling and Linear Control: Theory with Laboratory Tests," *Electronics*, vol. 13, no. 11, p. 2225, 2024. [Online]. Available: <https://www.mdpi.com/2079-9292/13/11/2225>.
- [27] C. Reyes and G. Gerena, "Mathematical modeling of a DC motor with separate excitation," *Tekhnê*, vol. 15, no. 1, pp. 13–20, 2018.
- [28] F. Zaghrat, G. Brahim, B. Bouchiba, I. K. Bousserhane, and C. Imane, "Robust fuzzy sliding mode control implementation for DC motor," *PRZEGLĄD ELEKTROTECHNICZNY*, vol. 5/2022, p. 97, 04/02 2022, doi: 10.15199/46.2022.05.18.
- [29] B. C. Kuo, *Automatic Control Systems*. Prentice-Hall, 1982.
- [30] M. Srinivas and L. M. Patnaik, "Adaptive probabilities of crossover and mutation in genetic algorithms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 24, no. 4, pp. 656–667, 1994, doi: 10.1109/21.286385.
- [31] A. E. Eiben, R. Hinterding, and Z. Michalewicz, "Parameter control in evolutionary algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 2, pp. 124–141, 1999, doi: 10.1109/4235.771166.
- [32] M. Khine, M. Htwe, and M. Y. Y. Mon, "Simulation DC motor speed control system by using PID controller," *Int. J. Trend Sci. Res. Develop.*, vol. 3, no. 4, pp. 1418–1423, 2019.