Bộ điều khiển pid online auto - tuning sử dụng mạng nơ ron cho hệ xe hai bánh tự cân bằng

Hoàng Anh Vũ*, Đào Tăng Tín
Trường Đại học Quốc tế Hồng Bàng

TÓM TẮT

Trong bài báo này, chúng ta tiếp cận một phương pháp điều khiển khác cho hệ thống phi tuyến với bộ điều khiển PID (Proportional - Integral - Derivative) online auto - tuning sử dụng mạng nơ ron. Sự phát triển của khoa học kỹ thuật đã tạo tiền đề cho ngành công nghệ robot người đạt những thành tựu to lớn như: ASIMO, Avatar… chúng cân bằng tốt và di chuyển linh hoạt. Để hiểu được thuật toán cân bằng của nó, ta phải nghiên cứu những vấn đề cân bằng cơ bản thông qua xe hai bánh tự cân bằng. Nó hoạt động dựa trên mô hình hệ con lắc ngược phi tuyến gắn trên trục có hai bánh xe, do động học phi tuyến của hệ thống nên cần phải có một bộ điều khiển thích nghi để robot cân bằng và di chuyển linh hoạt. Hiệu suất của bộ điều khiển PID online auto - tuning sử dụng mạng nơ ron là đối tượng để nghiên cứu trong bài báo này.

Từ khóa: *phương pháp điều khiển, PID tuning, mạng nơ ron, bộ điều khiển PID, hệ thống phi tuyến*

1. ĐẶT VẤN ĐỀ

Ngày nay, bộ điều khiển PID trong công nghiệp chiếm một tỉ lệ khá lớn vì tính lợi ích của bộ điều khiển này là: chi phí thấp, dễ ứng dụng, dễ điều khiển và tương đối ổn định. Các hệ thống có tính phi tuyến đòi hỏi tính thích nghi bộ điều khiển phải tốt hơn, do đó các nhà nghiên cứu đã tìm ra phương pháp cải tiến bộ điều khiển PID như: Neural - PID, GA - PID, Fuzzy - PID,… với mục tiêu: nâng cao chất lượng của bộ điều khiển PID, cải thiện tính thích nghi của bộ điều khiển, ứng dụng cho nhiều loại đối tượng với độ ổn định cao.

Bộ điều khiển PID auto - tuning update online đã ra đời giúp cho việc điều khiển các đối tượng trở nên dễ dàng hơn, thích nghi tốt hơn cho các hệ thống phi tuyến và làm giảm thời gian thiết kế bộ điều khiển nên làm giảm chi phí so với các phương pháp khác. Bộ điều khiển PID auto - tuning là thuật toán mà trong đó trọng số Kp, Ki, Kd thay đổi theo thời gian tùy theo đặc tính của đối tượng để đáp ứng của hệ thống thích nghi tốt hơn giống như dựa trên nguyên lý điều khiển trượt [1 - 2] hay điều khiển mờ [3]. Điều này rất thích hợp đối với hệ thống động và làm cho hệ thống có tính thích nghi cao [4]. PID auto - tuning sử dụng mạng nơ ron nhân tạo [5] với phương pháp Gradient (steepest descent) để nâng cao chất lượng điều khiển của hệ thống theo thời gian thực và giúp việc hội tụ một cách dễ dàng hơn bộ điều khiển PID cổ điển [6].

Hầu hết các ứng dụng công nghiệp hiện đại không còn điều chỉnh vòng điều khiển sử dụng các phương pháp tính toán thủ công như PID Ziegler - Nichols tuning. Một phương pháp hiệu chỉnh khác là dùng giải thuật di truyền GA (Genetic Algorithm) - PID,

Tác giả liên hệ: ThS. Hoàng Anh Vũ
Email: vuha@hiu.vn

để tìm ra 3 hệ số Kp, Ki, Kd tối ưu thì phải lai gen của 3 hệ số qua nhiều thế hệ và thực hiện offline. Một phương pháp hiệu chỉnh nữa là mờ Fuzzy - PID [7] nhưng ba hệ số chỉ thay đổi trong khoảng cho trước, luật mờ và phụ thuộc nhiều vào kinh nghiệm của người thiết kế. Thay vào đó, phần mềm điều chỉnh PID online auto - tuning và tối ưu hóa vòng lặp được dùng để đảm báo kết quả đáp ứng hơn. Những gói phần mềm này sẽ tập hợp dữ liệu, phát triển các mô hình xử lý, và đề xuất phương pháp điều chỉnh tối ưu cho từng thời điểm và nó có thể tự cập nhật, điều chỉnh các thông số bằng cách thu thập dữ liệu từ các thay đổi của hệ thống.

Tính thích nghi online và hiệu suất của bộ điều khiển PID online auto - tuning sử dụng mạng nơ ron là cơ sở để đánh giá hiệu quả của phương pháp.

2. MÔ TẢ TOÁN HỌC CỦA VẤN ĐỀ CẦN GIẢI QUYẾT

Tùy vào đối tượng điều khiển trong miền tần số hay miền thời gian rời rạc, ta chọn bộ điều khiển PID tương ứng ở dạng liên tục hoặc rời rạc.

Ta có bộ điều khiển PID ở dạng liên tục:

$$u(t)= K_P \left[e(t) + \frac{1}{T_i} \int_0^t e(t)dt + T_d \frac{d\,e(t)}{dt} \right] \quad (1)$$

K_P : hệ số tỉ lệ; e: sai số đầu vào; u: ngõ ra;

T_i: thời gian tích phân; T_d : thời gian đạo hàm

Với thời gian lấy mẫu T, bộ điều khiển PID dạng rời rạc được viết lại như sau:

$$u(k) = u(k-1) + K_P[e(k) - e(k-1)] + + K_i e(k) + + K_d[e(k) - 2e(k-1) + e(k-2)] \quad (2)$$

$K_i = K_p T T_i^{-1}$: hệ số tích phân

$K_d = K_p T_d T^{-1}$: hệ số đạo hàm

Ta có thể viết lại dạng bộ điều khiển PID dạng rời rạc viết lại như sau:

$$u(k) = K_P e(k) + K_i e_{sum} + K_d[e(k) - e(k-1)] \quad (3)$$

2.1. Nơ ron tự hiệu chỉnh dạng một SNN (single neural network)

2.1.1. Cấu trúc cơ bản của một nơ ron tự hiệu chỉnh

$$net = I - f \quad (4)$$

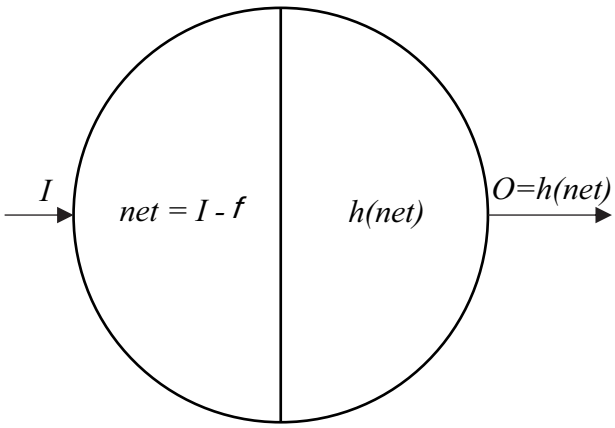
$$O = h(net) = \frac{a[1 - \exp(-bnet)]}{[1 + \exp(-bnet)]} \quad (5)$$

Ngõ ra của nơ ron là một hàm tiếp tuyến tiệm cận dạng hyperbol

a: mức bão hòa

b: độ dốc của hàm ngõ ra

Đặt $q = [f \quad a \quad b]^T$ là bộ trọng số cần hiệu chỉnh đối với một nơ ron.

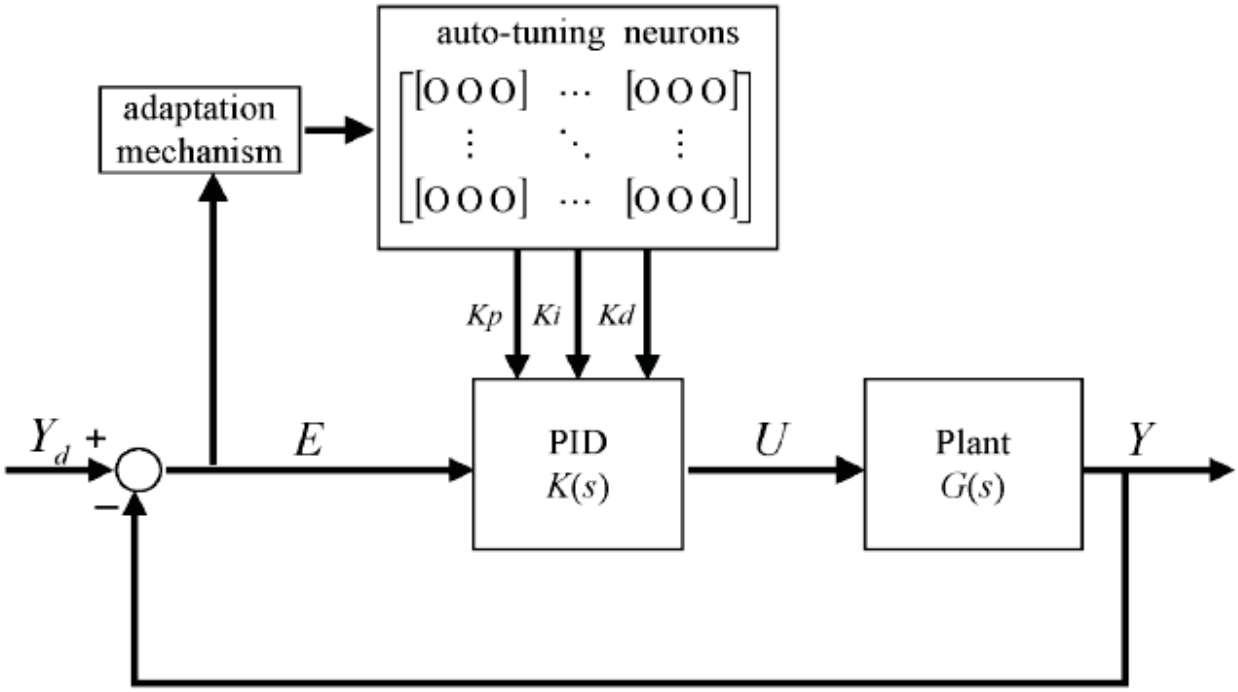


Hình 1. *Cấu trúc cơ bản của một nơ ron tự hiệu chỉnh*

2.1.2. Cấu trúc bộ điều khiển đa biến PID như hình sau

Cấu trúc nơ ron tự hiệu chỉnh dạng một cho ba hệ số K_p, K_i, K_d được thiết kế như sau:

- Mỗi hệ số sẽ được hiệu chỉnh bằng một nơ ron tương ứng với các giá trị đầu.
- Phương pháp gradient (steepest descent) sẽ tính toán và cập nhật lại các hệ số.
- Ba hệ số sau khi được cập nhật sẽ tiếp tục cập nhật cho luật điều khiển u.



Hình 2. *Cấu trúc bộ điều khiển đa biến PID với các nơ ron tự động hiệu chỉnh*

Vector ngõ ra mong muốn: $Y_d = [y_{d1} \quad y_{d2} \quad \dots \quad y_{dn}]^T$ (6)

Vector ngõ ra thực tế: $Y = [y_1 \quad y_2 \quad \dots \quad y_n]^T$ (7)

Vector error: $E = Y_d - Y = [e_1 \quad e_2 \quad \dots \quad e_n]^T$ (8)

Vector điều khiển ngõ vào: $U = [u_1 \quad u_2 \quad \dots \quad u_n]^T$ (9)

nxn đối tượng đa biến:

$$G(s) = \begin{bmatrix} g_{11}(s) & \cdots & g_{1n}(s) \\ \vdots & \ddots & \vdots \\ g_{n1}(s) & \dots & g_{nn}(s) \end{bmatrix} \quad (10)$$

nxn bộ điều khiển PID đa biến:

$$K(s) = \begin{bmatrix} k_{11}(s) & \cdots & k_{1n}(s) \\ \vdots & \ddots & \vdots \\ k_{n1}(s) & \dots & k_{nn}(s) \end{bmatrix} \quad (11)$$

Định nghĩa 1: đối tượng điều khiển phải là đáp ứng dương hoặc đáp ứng âm tương ứng với u và y. Nếu một ngõ ra y tăng hoặc giảm khi đầu vào u cũng tăng hoặc giảm. Cả đáp ứng dương hoặc đáp ứng âm phải là đáp ứng một cách đơn điệu.

Định nghĩa 2: nếu đối tượng là đáp ứng dương hoặc âm tương ứng với u và y, ta định nghĩa

$$\operatorname{sgn}\left(\frac{\partial y}{\partial u}\right) = 1(-1).$$

2.1.3. Phương thức cập nhật trọng số

Hàm mục tiêu:

$$J=\frac{1}{2}\left(Y_d - Y\right)^2 \tag{12}$$

Phương pháp gradient (steepest descent)

$$q(k+1)=q(k)-h\nabla J \tag{13}$$

h >0: hằng số học

$$\nabla J=\left(\frac{\partial J}{\partial q}\right)^T \tag{14}$$

q_{ij} : là một vector cột gồm 3 thông số $q_{ij}=\left[f_{ij} \quad a_{ij} \quad b_{ij}\right]^T$ nên ta có luật cập nhật mỗi thông số cho PID như sau:

Hệ số $K_{p_{ij}}$

$$\frac{\partial J}{\partial a_{ij_p}}=\left\{\sum_{m=1}^{n}\frac{\partial J}{\partial y_m}\frac{\partial y_m}{\partial u_i}\right\}\frac{\partial u_i}{\partial K_{p_{ij}}}\frac{\partial K_{p_{ij}}}{\partial a_{ij_p}} \tag{15}$$

$$\frac{\partial J}{\partial b_{ij_p}}=\left\{\sum_{m=1}^{n}\frac{\partial J}{\partial y_m}\frac{\partial y_m}{\partial u_i}\right\}\frac{\partial u_i}{\partial K_{p_{ij}}}\frac{\partial K_{p_{ij}}}{\partial b_{ij_p}} \tag{16}$$

$$\frac{\partial J}{\partial f_{ij_p}}=\left\{\sum_{m=1}^{n}\frac{\partial J}{\partial y_m}\frac{\partial y_m}{\partial u_i}\right\}\frac{\partial u_i}{\partial K_{p_{ij}}}\frac{\partial K_{p_{ij}}}{\partial f_{ij_p}} \tag{17}$$

Hệ số $K_{i_{ij}}$

$$\frac{\partial J}{\partial a_{ij_i}}=\left\{\sum_{m=1}^{n}\frac{\partial J}{\partial y_m}\frac{\partial y_m}{\partial u_i}\right\}\frac{\partial u_i}{\partial K_{i_{ij}}}\frac{\partial K_{i_{ij}}}{\partial a_{ij_i}} \tag{18}$$

$$\frac{\partial J}{\partial b_{ij_i}}=\left\{\sum_{m=1}^{n}\frac{\partial J}{\partial y_m}\frac{\partial y_m}{\partial u_i}\right\}\frac{\partial u_i}{\partial K_{i_{ij}}}\frac{\partial K_{i_{ij}}}{\partial b_{ij_i}} \tag{19}$$

$$\frac{\partial J}{\partial f_{ij_i}}=\left\{\sum_{m=1}^{n}\frac{\partial J}{\partial y_m}\frac{\partial y_m}{\partial u_i}\right\}\frac{\partial u_i}{\partial K_{i_{ij}}}\frac{\partial K_{i_{ij}}}{\partial f_{ij_i}} \tag{20}$$

Hệ số $K_{d_{ij}}$

$$\frac{\partial J}{\partial a_{ij}}=\left\{\sum_{m=1}^{n}\frac{\partial J}{\partial y_m}\frac{\partial y_m}{\partial u_i}\right\}\frac{\partial u_i}{\partial K_{d_{ij}}}\frac{\partial K_{d_{ij}}}{\partial a_{ij}} \tag{21}$$

$$\frac{\partial J}{\partial b_{ij}}=\left\{\sum_{m=1}^{n}\frac{\partial J}{\partial y_m}\frac{\partial y_m}{\partial u_i}\right\}\frac{\partial u_i}{\partial K_{d_{ij}}}\frac{\partial K_{d_{ij}}}{\partial b_{ij}} \tag{22}$$

$$\frac{\partial J}{\partial f_{ij}}=\left\{\sum_{m=1}^{n}\frac{\partial J}{\partial y_m}\frac{\partial y_m}{\partial u_i}\right\}\frac{\partial u_i}{\partial K_{d_{ij}}}\frac{\partial K_{d_{ij}}}{\partial f_{ij}} \tag{23}$$

i, j, m Î n

Ta có:

$$\frac{\partial J}{\partial y_m}=-e_m \tag{24}$$

Hầu hết các hệ thống xử lý là đáp ứng dương hoặc đáp ứng âm tương ứng với u_i và y_m. Vì vậy ta có thể thay thế

$$\frac{\partial y_m}{\partial u_i} \text{ bằng } \operatorname{sgn}\left[\frac{\partial y_m}{\partial u_i}\right]=1(-1). \tag{25}$$

Nếu tính theo dạng rời rạc thứ nhất:

$$\frac{\partial u_i}{\partial K_{p_{ij}}}=e_j(k)-e_j(k-1) \tag{26}$$

$$\frac{\partial u_i}{\partial K_{i_{ij}}}=e_j(k) \tag{27}$$

$$\frac{\partial u_i}{\partial K_{d_{ij}}}=e_j(k)-2e_j(k-1)+e_j(k-2) \tag{28}$$

Nếu tính theo dạng rời rạc thứ hai:

$$\frac{\partial u_i}{\partial K_{p_{ij}}}=e_j(k) \tag{29}$$

$$\frac{\partial u_i}{\partial K_{i_{ij}}}=e_{j_{sum}}(k) \tag{30}$$

$$\frac{\partial u_i}{\partial K_{d_{ij}}}=e_j(k)-e_j(k-1) \tag{31}$$

Ta có:

$$\frac{\partial K_{p_{ij}}}{\partial a_{ij_p}} = \frac{K_{p_{ij}}}{a_{ij_p}} \quad (32)$$

$$\frac{\partial K_{d_{ij}}}{\partial b_{ij_d}} = \frac{a_{ij_d} net_{ij_d}}{2}\left(1 + \frac{K_{d_{ij}}}{a_{ij_d}}\right)\left(1 - \frac{K_{d_{ij}}}{a_{ij_d}}\right) \quad (37)$$

$$\frac{\partial K_{i_{ij}}}{\partial a_{ij_i}} = \frac{K_{i_{ij}}}{a_{ij_i}} \quad (33)$$

$$\frac{\partial K_{p_{ij}}}{\partial f_{ij_p}} = \frac{a_{ij_p} f_{ij_p}}{2}\left(1 + \frac{K_{p_{ij}}}{a_{ij_p}}\right)\left(1 - \frac{K_{p_{ij}}}{a_{ij_p}}\right) \quad (38)$$

$$\frac{\partial K_{d_{ij}}}{\partial a_{ij_d}} = \frac{K_{d_{ij}}}{a_{ij_d}} \quad (34)$$

$$\frac{\partial K_{i_{ij}}}{\partial f_{ij_i}} = \frac{a_{ij_i} f_{ij_i}}{2}\left(1 + \frac{K_{i_{ij}}}{a_{ij_i}}\right)\left(1 - \frac{K_{i_{ij}}}{a_{ij_i}}\right) \quad (39)$$

$$\frac{\partial K_{p_{ij}}}{\partial b_{ij_p}} = \frac{a_{ij_p} net_{ij_p}}{2}\left(1 + \frac{K_{p_{ij}}}{a_{ij_p}}\right)\left(1 - \frac{K_{p_{ij}}}{a_{ij_p}}\right) \quad (35)$$

$$\frac{\partial K_{d_{ij}}}{\partial f_{ij_d}} = \frac{a_{ij_d} f_{ij_d}}{2}\left(1 + \frac{K_{d_{ij}}}{a_{ij_d}}\right)\left(1 - \frac{K_{d_{ij}}}{a_{ij_d}}\right) \quad (40)$$

$$\frac{\partial K_{i_{ij}}}{\partial b_{ij_i}} = \frac{a_{ij_i} net_{ij_i}}{2}\left(1 + \frac{K_{i_{ij}}}{a_{ij_i}}\right)\left(1 - \frac{K_{i_{ij}}}{a_{ij_i}}\right) \quad (36)$$

Từ công thức (13) kết hợp với (15) đến (40), ta có công thức cập nhật thông số sau:

$$a_{ij_p}(k+1) = a_{ij_p}(k) + \eta_{ij_p}\left(\sum_{m=1}^{n} e_m \operatorname{sgn}\left[\frac{\partial y_m}{\partial u_i}\right]\frac{\partial u_i}{\partial K_{p_{ij}}}\right)\frac{K_{p_{ij}}}{a_{ij_p}} \quad (41)$$

$$a_{ij_i}(k+1) = a_{ij_i}(k) + \eta_{ij_i}\left(\sum_{m=1}^{n} e_m \operatorname{sgn}\left[\frac{\partial y_m}{\partial u_i}\right]\frac{\partial u_i}{\partial K_{i_{ij}}}\right)\frac{K_{i_{ij}}}{a_{ij_d}} \quad (42)$$

$$a_{ij_d}(k+1) = a_{ij_d}(k) + \eta_{ij_d}\left(\sum_{m=1}^{n} e_m \operatorname{sgn}\left[\frac{\partial y_m}{\partial u_i}\right]\frac{\partial u_i}{\partial K_{d_{ij}}}\right)\frac{K_{d_{ij}}}{a_{ij_d}} \quad (43)$$

$$b_{ij_d}(k+1) = b_{ij_d}(k) + \eta_{ij_d}\left(\sum_{m=1}^{n} e_m \operatorname{sgn}\left[\frac{\partial y_m}{\partial u_i}\right]\frac{\partial u_i}{\partial K_{p_{ij}}}\frac{a_{ij_p} net_{ij_p}}{2}\right)\left(1 + \frac{K_{p_{ij}}}{a_{ij_p}}\right)\left(1 - \frac{K_{p_{ij}}}{a_{ij_p}}\right) \quad (44)$$

$$b_{ij_i}(k+1) = b_{ij_i}(k) + \eta_{ij_i}\left(\sum_{m=1}^{n} e_m \operatorname{sgn}\left[\frac{\partial y_m}{\partial u_i}\right]\frac{\partial u_i}{\partial K_{i_{ij}}}\frac{a_{ij_i} net_{ij_i}}{2}\right)\left(1 + \frac{K_{i_{ij}}}{a_{ij_i}}\right)\left(1 - \frac{K_{i_{ij}}}{a_{ij_i}}\right) \quad (45)$$

$$b_{ij_d}(k+1) = b_{ij_d}(k) + \eta_{ij_d}\left(\sum_{m=1}^{n} e_m \operatorname{sgn}\left[\frac{\partial y_m}{\partial u_i}\right]\frac{\partial u_i}{\partial K_{d_{ij}}}\frac{a_{ij_i} net_{ij_i}}{2}\right)\left(1 + \frac{K_{d_{ij}}}{a_{ij_d}}\right)\left(1 - \frac{K_{d_{ij}}}{a_{ij_d}}\right) \quad (46)$$

$$f_{ij_d}(k+1)=f_{ij_d}(k)\quad h_{ij_d}\left[\sum_{m=1}^{n} e_m \operatorname{sgn}\left[\frac{\partial y_m}{\partial u_i}\right]\frac{\partial u_i}{\partial K_{p_{ij}}}\frac{a_{ij_p}b_{ij_p}}{2}\left(1+\frac{K_{p_{ij}}}{a_{ij_p}}\right)\frac{K_{p_{ij}}}{a_{ij_p}}\right] \tag{47}$$

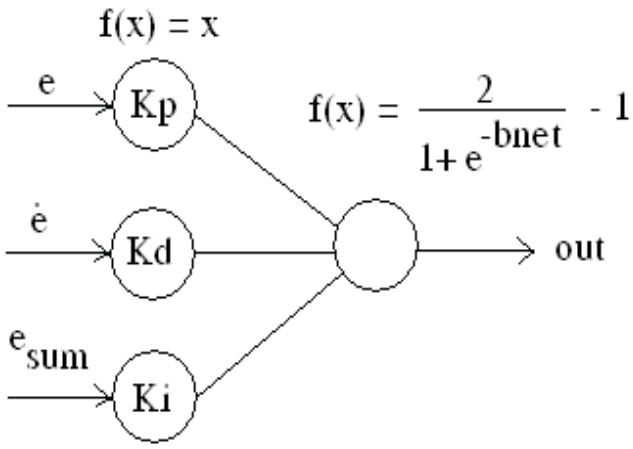
$$f_{ij_i}(k+1)=f_{ij_i}(k)\quad h_{ij_i}\left[\sum_{m=1}^{n} e_m \operatorname{sgn}\left[\frac{\partial y_m}{\partial u_i}\right]\frac{\partial u_i}{\partial K_{i_{ij}}}\frac{a_{ij_i}b_{ij_i}}{2}\left(1+\frac{K_{i_{ij}}}{a_{ij_i}}\right)\frac{K_{i_{ij}}}{a_{ij_i}}\right] \tag{48}$$

$$f_{ij_d}(k+1)=f_{ij_d}(k)+h_{ij_d}\left[\sum_{m=1}^{n} e_m \operatorname{sgn}\left[\frac{\partial y_m}{\partial u_i}\right]\frac{\partial u_i}{\partial K_{d_{ij}}}\frac{a_{ij_i}b_{ij_i}}{2}\left(1+\frac{K_{d_{ij}}}{a_{ij_d}}\right)\frac{K_{d_{ij}}}{a_{ij_d}}\right] \tag{49}$$

Các công thức từ (41) đến (49) thể hiện quy luật cập nhật các thông số của một nơ ron trong hệ đa biến. Sau đó, ta sử dụng công thức (5) tính toán ngõ ra của một nơ ron. Đây chính là phương thức các hệ số K_p, K_i, K_d được chỉnh định của bộ điều khiển đa biến PID.

2.2. Nơ ron tự hiệu chỉnh dạng hai MLFNN (multi - layer feed - forward neural network)

2.2.1. Cấu trúc cơ bản của mạng nơ ron tự hiệu chỉnh



Hình 3. *Cấu trúc cơ bản của mạng nơ ron tự hiệu chỉnh*

$$u=K_p*e+K_i*e_{sum}+K_d*\dot{e} \tag{50}$$

$$O = h(net) = \frac{[1-\exp(-bnet)]}{[1+\exp(-bnet)]} \tag{51}$$

Ngõ ra của nơ ron là một hàm tansig (khả vi).

2.2.2. Phương thức cập nhật trọng số

Luật cập nhật trọng số bộ điều khiển PID auto - tuning:

Hàm mục tiêu:

$$J=\frac{1}{2}(Y_d - Y)^2 \tag{52}$$

Phương pháp gradient (steepest descent)

$$q(k+1)=q(k)-h\nabla J \tag{53}$$

h >0: hằng số học

Ta có:

$$K_p=K_p-h\frac{\partial E}{\partial K_p} \tag{54}$$

$$K_i=K_i-h\frac{\partial E}{\partial K_i} \tag{55}$$

$$K_d=K_d-h\frac{\partial E}{\partial K_d} \tag{56}$$

$$\frac{\partial E}{\partial K_p}=\frac{\partial E}{\partial y}\frac{\partial y}{\partial u}\frac{\partial u}{\partial K_p} \tag{57}$$

$$\frac{\partial E}{\partial K_i}=\frac{\partial E}{\partial y}\frac{\partial y}{\partial u}\frac{\partial u}{\partial K_i} \tag{58}$$

$$\frac{\partial E}{\partial K_d}=\frac{\partial E}{\partial y}\frac{\partial y}{\partial u}\frac{\partial u}{\partial K_d} \tag{59}$$

$$\frac{\partial E}{\partial y}=-e \tag{60}$$

$$\frac{\P y}{\P u} = f^{'}(u) \quad (61)$$

$$\frac{\P u}{\P K_p} = e \quad (62)$$

$$\frac{\P u}{\P K_i} = e_{sum} \quad (63)$$

$$\frac{\P u}{\P K_d} = e \quad (64)$$

Từ công thức (53) kết hợp với (54) đến (64), ta có công thức cập nhật thông số sau:

$$K_p = K_p - h.e.f^{'}(u).e \quad (65)$$

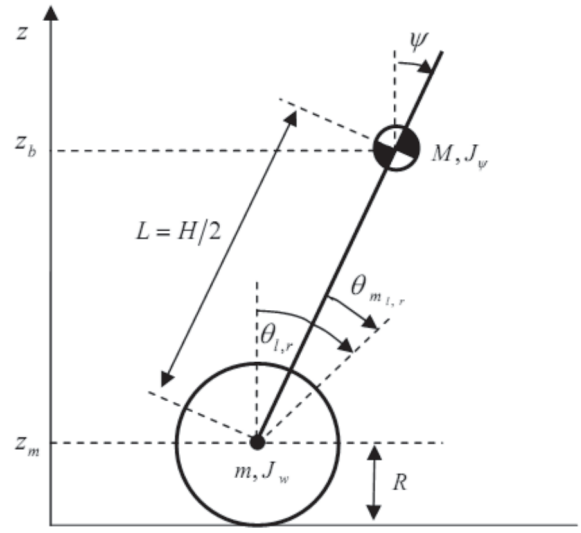
$$K_i = K_i - h.e.f^{'}(u).e_{sum} \quad (66)$$

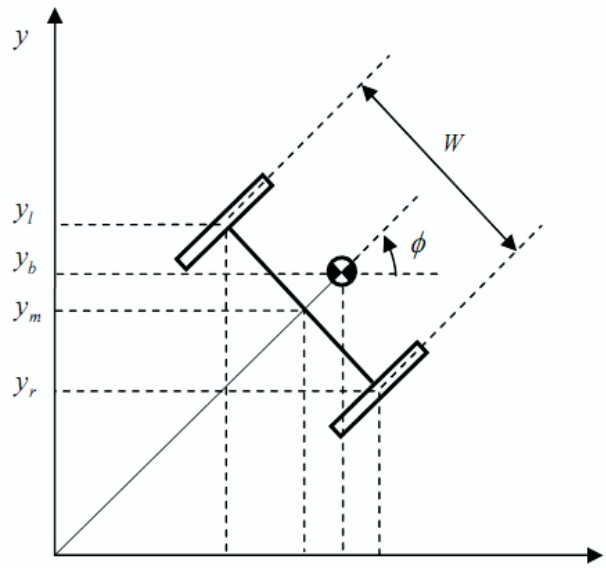
$$K_d = K_d - h.e.f^{'}(u).e \quad (67)$$

2.3. Mô hình hóa robot hai bánh tự cân bằng

Robot hai bánh tự cân bằng dựa trên mô hình hệ con lắc ngược phi tuyến tương tự pendubot [8] nhưng ít hơn một khâu. Việc điều khiển mô hình robot hai bánh tự cân bằng giống như ý tưởng điều khiển hệ thanh trượt và banh [1] nhưng khó hơn vì tính phi tuyến và quán tính của hệ thống [9]. Hơn nữa, ta có thể điều khiển vị trí xe, góc pitch, góc yaw nên hệ thống là đa biến ngõ ra [10] nên gặp khó khăn trong điều khiển hệ thống.

2.3.1. Mô tả phần cứng





Hình 4. *Hình chiếu cạnh và chiếu bằng của xe con lắc ngược 2 bánh*

Ψ: góc pitch của con lắc

Φ: góc yaw của con lắc

$\theta_{l,r}$: góc trái và phải của bánh

$q_{m_{l,r}}$: góc động cơ DC

Các thông số của con lắc ngược 2 bánh

g = 9.81[m/s^2]: gia tốc trọng trường

m = 0.05[kg]: khối lượng bánh

R = 0.051[m]: bán kính bánh

J_w= mR^2/2[kgm^2]: mômen quán tính bánh

M = 0.6[kg]: khối lượng con lắc

W = 0.227[m]: bề rộng con lắc

D = 0.178[m]: bề ngang con lắc

H = 0.144[m]: chiều cao con lắc

L = H/2[m]: khoảng cách từ trọng tâm con lắc đến tâm bánh

J_ψ=ML^2/3[kgm^2]: mômen quán tính pitch của con lắc

J_Φ=M(W^2 + D^2)/12: mômen quán tính yaw của con lắc

Các thông số của động cơ DC

J_m= 1x10^{-5}[kgm^2]: mômen quán tính của động cơ DC

R_m= 6.69[Ω]: điện trở của động cơ DC

K_b= 0.468[Vs/rad]: hệ số suất điện động hồi tiếp (BEMF) của động cơ DC

K_t= 0.317[Nm/A]: hệ số mômen của động cơ DC

n = 1: tỉ số truyền của hộp giảm tốc

f_m= 0.002: hệ số ma sát giữa con lắc và động cơ DC

f_w= 0.18: hệ số ma sát giữa bánh và sàn

2.3.2. Mô hình toán học

Giả sử hướng của con lắc ngược 2 bánh là hướng dương của trục x vào thời điểm t = 0, theo hệ tọa độ ta có các phương trình sau:

Ψ: góc pitch của con lắc

Φ: góc yaw của con lắc

θ: góc trung bình của bánh trái và bánh phải

Phương trình Lagrange của hệ:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}}\right) - \frac{\partial L}{\partial q} = F_q \qquad \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{y}}\right) - \frac{\partial L}{\partial y} = F_y \qquad \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{f}}\right) - \frac{\partial L}{\partial f} = F_f \tag{68}$$

$$\left[(2m+M)R^2 + 2J_w + 2n^2 J_m\right]\ddot{q} + (MLR\cos y - 2n^2 J_m)\ddot{y} - MLR\dot{y}^2 \sin y = F_q \tag{69}$$

$$(MLR\cos y - 2n^2 J_m)\ddot{q} + (ML^2 + J_y + 2n^2 J_m)\ddot{y} - MgL\sin y - ML^2 \dot{f}^2 \sin y \cos y = F_y \tag{70}$$

$$\left[\frac{1}{2} mW^2 + J_f + \frac{W^2}{2R^2}(J_w + 2n^2 J_m) + ML^2 \sin^2 y\right]\ddot{f} + 2ML^2 \dot{y}\dot{f} \sin y \cos y = F_f \tag{71}$$

$$F_q = F_l + F_r \quad (72), \qquad F_y = F_y \quad (73), \qquad F_f = \frac{W}{2R}(F_r - F_l) \tag{74}$$

$$F_l = nK_t i_l + f_m(\dot{y} - \dot{q}_l) - f_w \dot{q}_l \tag{75}$$

$$F_r = nK_t i_r + f_m(\dot{y} - \dot{q}_r) - f_w \dot{q}_r \tag{76}$$

$$F_y = -nK_t i_l - nK_t i_r + f_m(\dot{y} - \dot{q}_l) - f_m(\dot{y} - \dot{q}_r) \tag{77}$$

$$L_m \dot{i}_{l,r} = V_{l,r} + K_b(\dot{y} - \dot{q}_{l,r}) - R_m i_{l,r} \tag{78}$$

Giả sử L_m =0, từ (78) ta có:

$$i_{l,r} = \frac{V_{l,r} + K_b(\dot{y} - \dot{q}_{l,r})}{R_m} \tag{79}$$

$$F_q = \frac{nK_t}{R_m}(V_l + V_r) - 2\left(\frac{nK_t K_b}{R_m} + f_m + f_w\right)\dot{q} + 2\left(\frac{nK_t K_b}{R_m} + f_m\right)\dot{y} \tag{80}$$

$$F_y = -\frac{nK_t}{R_m}(V_l + V_r) + 2\left(\frac{nK_t K_b}{R_m} + f_m\right)\dot{q} - 2\left(\frac{nK_t K_b}{R_m} + f_m\right)\dot{y} \tag{81}$$

$$F_f = \frac{W}{2R}\frac{nK_t}{R_m}(V_r - V_l) - \frac{W^2}{2R^2}\left(\frac{nK_t K_b}{R_m} + f_m + f_w\right)\dot{f} \tag{82}$$

$$a = \frac{nK_t}{R_m} \qquad b = \frac{nK_t K_b}{R_m} + f_m \qquad g = \frac{W}{R} \tag{83}$$

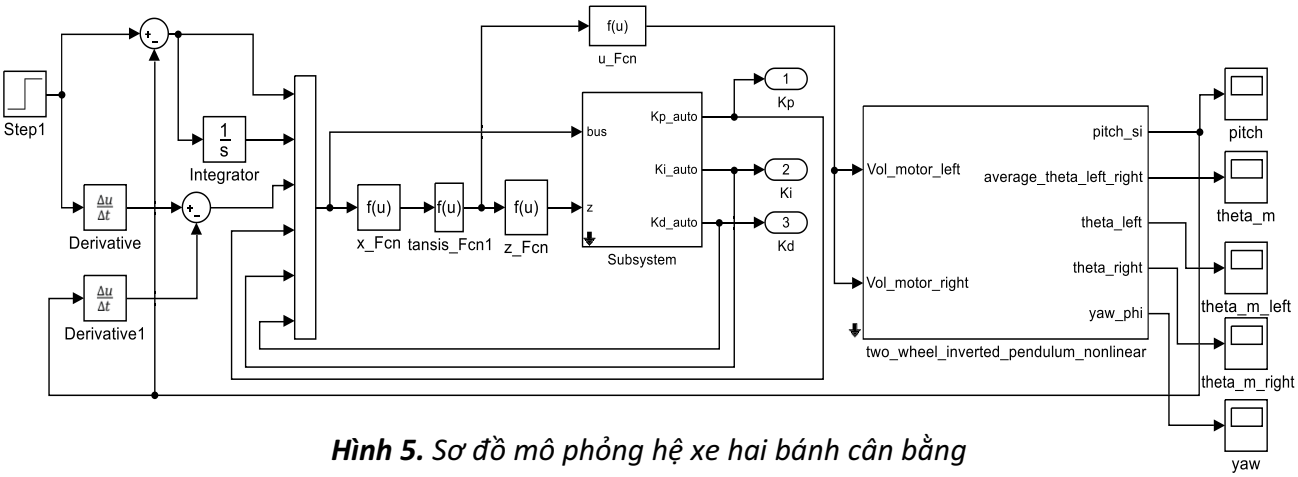
$$F_q = a(V_l + V_r) - 2(b + f_w)\dot{q} + 2b\dot{y} \quad (84)$$

$$F_y = -a(V_l + V_r) + 2b\dot{q} - 2b\dot{y} \quad (85)$$

$$F_f = \frac{1}{2}ga(V_r - V_l) - \frac{1}{2}g^2(b + f_w)\dot{f} \quad (86)$$

Ta không tuyến tính hóa hệ thống quanh điểm cân bằng để giữ tính phi tuyến của hệ thống trong việc mô hình hóa đối tượng để kiểm tra tính thích nghi của bộ điều khiển PID online auto-tuning.

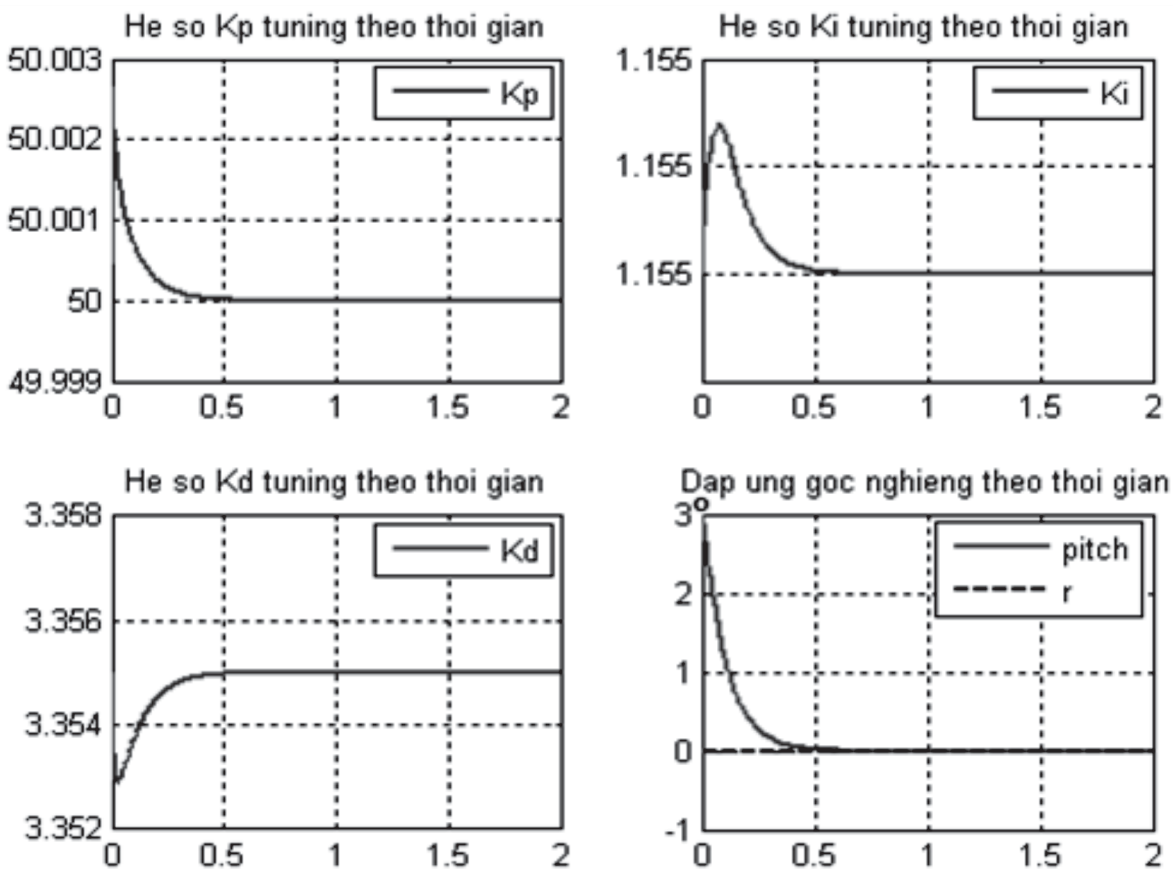
3. KẾT QUẢ MÔ PHỎNG



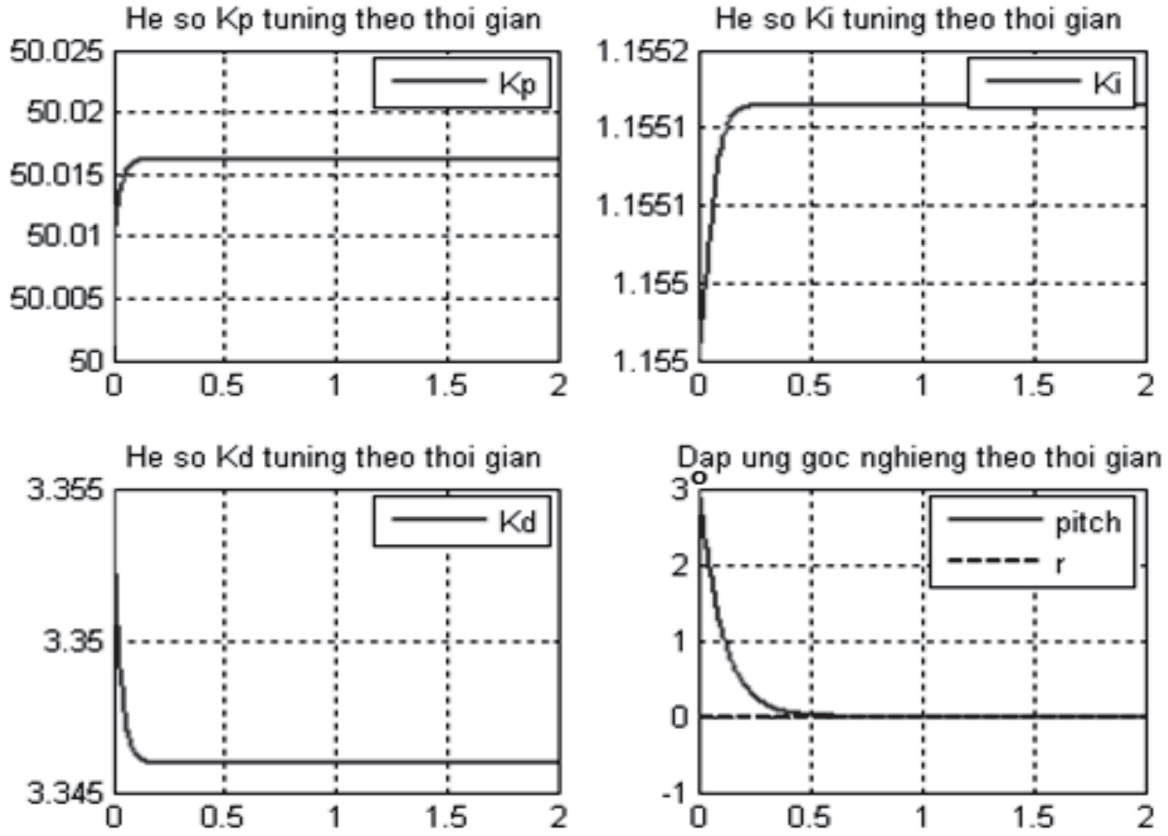
Hình 5. *Sơ đồ mô phỏng hệ xe hai bánh cân bằng*

Kết quả mô phỏng dạng một SNN và dạng hai MLFNN

Tại vị trí cân bằng góc pitch = 0^0. Góc vị trí ban đầu là 3^0. Ta chọn định các hệ số Kp = 50.00, Ki = 1.155, Kd = 3.355 tùy ý.



Hình 6. *Kết quả mô phỏng dạng một SNN*



Hình 7. *Kết quả mô phỏng dạng hai MLFNN*

Nhận xét dạng 1: vì ta xác định các hệ số Kp, Ki, Kd lớn hơn nên tốc độ đáp ứng nhanh và bám sát tín hiệu đặt. Tại mỗi thời điểm, đáp ứng của hệ thống với một bộ các trọng số thay đổi khác nhau nên các trọng số Kp, Ki, Kd sẽ tự hiệu chỉnh qua lại để đáp ứng hệ thống là tốt nhất.

Nhận xét dạng 2: với các hệ số Kp, Ki, Kd lớn bất kỳ, ta thấy đáp ứng của hệ thống cần một bộ các trọng số khác nhau ở các thời điểm. Dạng cập nhật ở dạng hai cũng khác tương đối với dạng một nhưng thời gian đáp ứng của hai dạng gần như nhau. Như vậy, ta có thể thấy tính thích nghi rất cao của hệ điều khiển PID online auto - tuning sử dụng mạng nơ ron tại các thời điểm đáp ứng khác nhau.

4. KẾT LUẬN

PID auto - tuning sử dụng mạng nơ ron nhân tạo với phương pháp Gradient (steepest descent) đã nâng cao chất lượng điều khiển của hệ thống và tính thích nghi theo thời gian thực và giúp việc hội tụ một cách dễ dàng hơn bộ điều khiển PID.

Bộ điều khiển PID online auto - tuning cải tiến phù hợp với ứng dụng cho hệ xe hai bánh tự cân bằng theo thời gian thực. Bộ điều khiển này được thực hiện để kiểm soát tốt vị trí góc nghiêng cho hệ xe hai bánh tự cân bằng được thử nghiệm trong bài báo này. Kết quả thử nghiệm cho thấy bộ điều khiển PID online auto - tuning được đề xuất có thể học được các đặc tính động và phi tuyến của hệ thống nhanh chóng và do đó làm giảm sai số góc nghiêng và thời gian đáp ứng của hệ thống nhanh. Hiệu suất của bộ điều khiển PID online auto - tuning được nhận thấy là rất tốt và tính thích nghi cao khi có sự thay đổi bên ngoài. Hơn nữa, với đề xuất này, thuật toán PID online auto - tuning giá trị các thông số Kp, Ki và Kd có thể được sửa đổi trong thời gian thực và đáp ứng thực tế cũng có thể được giám sát. Điều này tạo điều kiện thuận lợi cho việc thử nghiệm trong các điều kiện đầu vào khác nhau và đảm bảo các ứng dụng trong tương lai của các hệ thống khác. Ta có thể tự tin rằng bộ điều khiển PID online auto - tuning được

đề xuất không chỉ chứng minh hiệu suất tuyệt vời trong việc điều khiển hệ xe hai bánh tự cân bằng với phi tuyến tính cao mà còn sẽ rất hiệu quả trong việc kiểm soát các ứng dụng công nghiệp và các hệ phi tuyến khác. Đây chính là hướng phát triển tiếp theo cho bộ điều khiển này và kiểm tra tính ổn định và thích nghi trên nhiều hệ thống khác.

TÀI LIỆU THAM KHẢO

[1] P. V. M. Maalini, G. Prabhakar and S. Selvaperumal, "*Modelling and Control of Ball and Beam System using PID Controller*" in Proc. of the Int. Conf. on Advanced Communication and Computing Technologies, pp. 322-326, 2016.

[2] Wang, W., Yi, J., Zhao, D. and Liu, D., "*Design of a stable sliding mode controller for a class of second-order underactuated systems*". IEE Proceedings - Control Theory and Applications 151, pp. 683–690, 2004.

[3] Chih-Min Lin, and Yi-Jen Mon, "*Decoupling Control by Hierarchical Fuzzy Sliding-Mode Controller*", IEEE Transactions on Control Systems Technology. Vol. 13.No. 4. July 2005, pp. 593-598, 2005.

[4] Luyang Li, Yun-Hui Liu, Tianjiao Jiang, Kai Wang, Mu Fang, "*Adaptive Trajectory Tracking of Nonholonomic Mobile Robots Using Vision-Based Position and Velocity Estimation*". IEEE Transactions on Cybernetics. Vol. 48.No. 2. Feb 2018, pp. 571–582, 2018.

[5] Hesselroth T, Sarkar K, Van der Smagt P, Schulten K, "*Neural network control of a pneumatic robot arm*", IEEE Trans. System Man Cybernetics 24(1), pp.28–38, 1994.

[6] Ahn K.K., Thanh T.D.C., "*Nonlinear PID control to improve the control performance of PAM manipulators using neural network*", In KSME, Int., Jour., 19(1):pp.106-115, 2005.

[7] K.S. Tang; Kim Fung Man; Guanrong Chen; S. Kwong, "*An optimal fuzzy PID controller*", IEEE Transactions on Industrial Electronics, Vol. 48, No. 4, Aug 2001, pp. 757-765, 2001.

[8] Fantoni, I., Lozano, R. and Spong, M.W., "*Energy based control of the pendubot*", IEEE Transactions on Automatic Control 45, pp. 725–729, 2000.

[9] Fang, Y., Dixon, W.E., Dawson, D.M. and Zergeroglu, E., "*Nonlinear coupling control laws for an underactuated overhead crane system*", IEEE/ASME Transactions on Mechatronics, Vol. 8, No. 3, Sept 2003, pp. 418–423, 2003.

[10] S. Laghrouche, F. Plestan, and A. Glumineau, "*Multivariable Paractical Higher Order Sliding Mode Control*", Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference 2005, pp. 1252-1257, 2005.

Online auto-tuning pid controller using neural network for two-wheeled self-balancing robot

Hoang Anh Vu*, Dao Tang Tin

ABSTRACT

In this paper, we approach a different control method for nonlinear systems with a PID (Proportional - Integral - Derivative) online auto - tuning (PID) controller using neural networks. The development of science and technology has created a premise for the human robot technology industry to achieve great achievements such as: ASIMO, Avatar ... They are well - balanced and flexible. To understand their balancing algorithm, we must study fundamental balance problems through two - wheeled self -

balancing robot. It works on the model of a nonlinear inverted pendulum system mounted on a two - wheeled axis, and because of the nonlinear dynamics of the system, an adaptive controller is required for the robot to balance and move flexibly. Performance of PID online auto - tuning controller using neural network is the main subject of study in this paper.

Key words: *control method, PID tuning, neural network, PID controller, nonlinear system*

Received: 29/09/2020
Revised: 13/11/2020
Accepted for publication: 20/11/2020