

## Designing sliding mode control using exponential functions applied for the 5-DOF VNR T1 robot

Le Ngoc Truc<sup>1\*</sup>, Nguyen Tung Lam<sup>2</sup>

<sup>1</sup>Hung Yen University of Technology and Education

<sup>2</sup>Hanoi University of Science and Technology

\*Corresponding author E-mail: lengoctruc@gmail.com, lam.nguyentung@hust.edu.vn

### Abstract

Sliding mode control (SMC) has been studied and applied in various industrial systems due to its robustness and wide applicability. The drawback of conventional SMC is that the chattering phenomenon exists in the system caused by the control signal having discontinuous component. Therefore we have to accept a long system's transient time to receive a small chattering level. Commonly, the chattering problem can be overcome by using a saturation function or some other functions instead of the signum function. In order to create a large control signal during reaching phase and maintain a small control value during the sliding phase, the paper presents a SMC which uses an exponential function in the switching law. Hence, the system can catch up the sliding surface quickly and to reduce the total settling time. While the system going on the sliding manifold, the reaching part of the control law preserves a small enough value such that it almost does not make the chattering degree in the sliding phase. The proposed SMC is applied to the 5-DOF VNR T1 robot. Simulation results performed on the quasi-physical robot model using MATLAB/Simscap Multibody reflect the effectiveness of this approach.

**Keywords:** Sliding mode control; Switching control; 5-DOF robot; Manipulator; Exponential function

### Abbreviations

SMC	Sliding Mode Control
VSC	Variable Structure Controller

### Tóm tắt

Điều khiển trượt đã được nghiên cứu và áp dụng trong nhiều hệ thống công nghiệp do có khả năng bền vững và lớp đối tượng rộng. Điểm hạn chế ở điều khiển trượt truyền thống là tồn tại hiện tượng rung trong hệ, do tín hiệu điều khiển có thành phần không liên tục. Nên khi thiết kế phải chấp nhận thời gian quá độ dài để có độ rung ở mức phù hợp hoặc ngược lại. Thông thường, độ rung có thể được khắc phục bằng việc sử dụng hàm tỉ lệ bão hòa hoặc một số hàm khác thay thế cho hàm xác định dấu. Để tạo ra tín hiệu điều khiển lớn trong giai đoạn tiến tới mặt trượt và duy trì tín hiệu điều khiển nhỏ trong giai đoạn trượt về gốc hệ tọa độ, bài báo đưa ra cách sử dụng hàm mũ theo biến trượt trong luật chuyển mạch. Nhờ đó, hệ thống nhanh chóng bắt kịp mặt trượt và làm giảm tổng thời gian quá độ. Khi hệ nằm trên mặt trượt, thì thành phần chuyển mạch trong tín hiệu điều khiển trở về duy trì ở giá trị đủ nhỏ để gần như không gây ra độ rung trong giai đoạn trượt. Bộ điều khiển sử dụng hàm mũ được áp dụng cho robot VNR T1 5 bậc tự do. Các kết quả mô phỏng được thực hiện trên mô hình vật lý ảo MATLAB/Simscap Multibody của robot đã phản ánh tính hiệu quả của cách tiếp cận này.

### 1. Phần mở đầu

Điều khiển trượt là một bộ điều khiển phi tuyến thuộc nhóm các bộ điều khiển cấu trúc thay đổi VSC (Variable Structure

Controller). Điều khiển VSC và điều khiển chuyển mạch (switching control) đã được áp dụng thành công trong nhiều lĩnh vực công nghiệp như: điều khiển động cơ điện, robot, tàu thủy, hàng không, và các hệ thống cơ khí khác. Bằng cách kết hợp với các phương pháp điều khiển khác, điều khiển VSC nói chung và điều khiển trượt nói riêng, ngày càng được nghiên cứu và sử dụng rộng rãi. Hiện nay đã có nhiều thuật toán điều khiển với các phương pháp khác nhau áp dụng cho tay máy robot dạng chuỗi cũng như robot song song được nghiên cứu và công bố. Trong đó, điều khiển trượt với khả năng bền vững và tính chất linh hoạt trong kỹ thuật chọn mặt trượt (sliding surface), điều kiện trượt (sliding condition) vẫn luôn thu hút nhiều hướng nghiên cứu khác nhau. Lý thuyết điều khiển trượt cơ bản cho hệ thống phi tuyến đã được trình bày trong nhiều tài liệu điển hình như [1]–[3]. Một nghiên cứu tổng hợp về ứng dụng kỹ thuật điều khiển trượt vào lĩnh vực tay máy robot đã được giới thiệu trong bài báo [4]. Từ đó đến nay, cách tiếp cận điều khiển trượt cho hệ thống phi tuyến (trong đó bao gồm cả robot) ngày càng được phát triển và mở rộng kết hợp với các kỹ thuật khác như: điều khiển mờ [5], mạng nơron [6], [7], thích nghi [8], mờ-thích nghi [9], nơron-thích nghi [10]. Trong điều khiển trượt thì việc chọn mặt trượt và điều kiện trượt để đưa hệ thống tiến tới mặt trượt và ổn định ở đó là rất quan trọng. Cách chọn mặt trượt, điều kiện trượt truyền thống đã được trình bày ở [1], [3] cũng như nhiều tài liệu khác. Dựa vào đó, đã có những nghiên cứu tìm cách đề xuất đơn giản hóa mặt trượt như [11], hay sử dụng mặt trượt có dạng mũ phân thức [12]. Nhằm hạn chế hiện tượng rung xuất hiện trong hệ thống điều khiển trượt, cách thường dùng là

thay thế hàm xác định dấu bằng hàm tỉ lệ bão hòa cho biến mặt trượt. Bên cạnh đó cũng có một số đề xuất khác như: sử dụng hàm atan thay thế cho hàm xác định dấu [13], dùng cấu trúc điều khiển trượt bậc cao nhằm hạn chế rung [14], hay mô hình bậc hai để loại bỏ thành phần không liên tục trong tín hiệu điều khiển [15], [16], hoặc xem xét hệ số khuếch đại trong điều kiện trượt là một hàm của biến mặt trượt [17].

Mục tiêu của nghiên cứu được trình bày trong bài báo này là cố gắng làm giảm độ rung mà vẫn giữ nguyên thời gian quá độ mong muốn của hệ thống. Do đó, bài báo đề xuất sử dụng một điều kiện trượt có hệ số khuếch đại dạng hàm mũ. Hệ số khuếch đại của bộ điều khiển trượt sẽ tăng-giảm theo độ lớn của biến mặt trượt. Trong giai đoạn tiến tới mặt trượt thì hệ số tỉ lệ sẽ có giá trị lớn để tăng cường tín hiệu điều khiển nhằm đưa hệ thống nhanh chóng đạt tới mặt trượt. Trong giai đoạn trên mặt trượt thì hệ số tỉ lệ sẽ được giảm xuống một cách phù hợp chỉ để duy trì hệ thống ở trên mặt trượt nhằm hạn chế hiện tượng rung.

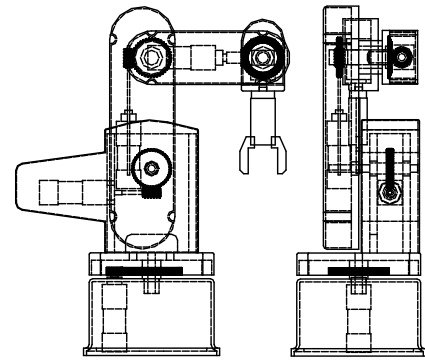
Chất lượng của bộ điều khiển trượt sử dụng hàm mũ sẽ được kiểm chứng thông qua việc áp dụng cho robot VNR T1 – một sản phẩm của công ty AKB Việt Nam thuộc chương trình nghiên cứu đào tạo và xây dựng hạ tầng kỹ thuật công nghệ cao (Hình 1). Bài báo có bố cục như sau: Mục 2 trình bày về cấu trúc và thiết lập mô hình động lực học của robot AKB VNR T1. Mục 3 xây dựng mô hình vật lý ảo cho robot sử dụng MATLAB/Simscap Multibody. Một số bộ điều khiển trượt cơ bản và bộ điều khiển trượt sử dụng hàm mũ sẽ được viết chi tiết ở Mục 4. Sơ đồ mô phỏng, các kết quả sẽ được đưa ra ở Mục 5. Các kết luận đánh giá sẽ được nêu trong Mục 6.



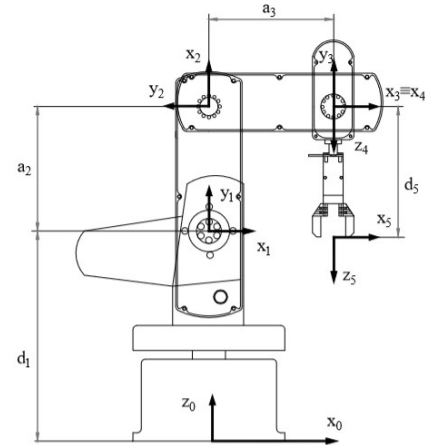
Hình 1: Robot VNR T1 của AKB Việt Nam.

## 2. Mô hình động lực học của robot AKB VNR T1

Robot VNR T1 có 5 bậc tự do, khối lượng 15kg, khả năng mang tải 1kg, không gian làm việc có tầm với tối đa 610mm, hoạt động với tốc độ 600mm/s và có độ chính xác lặp 0.8mm. Các khớp quay của robot được dẫn động bằng các động cơ điện servo 24VDC. Cấu trúc của robot được thể hiện trong Hình 2. Dựa vào quy tắc gắn các hệ tọa độ lên các khâu, sơ đồ mô tả các hệ tọa độ này được trình bày ở Hình 3, và lập bảng tham số D-H cho robot trong Bảng 1.



Hình 2: Cấu trúc của robot VNR T1.



Hình 3: Sơ đồ gắn các hệ tọa độ khớp cho robot VNR T1.

Bảng 1: Bảng tham số D-H của robot VNR T1

Joint $i$	$\theta_i$ (rad)	$d_i$ (m)	$a_i$ (m)	$\alpha_i$ (rad)
1	$q_1$	$d_1 = 0.253$	$a_1 = 0$	$\alpha_1 = \pi/2$
2	$q_2 + \pi/2$	$d_2 = 0$	$a_2 = 0.15$	$\alpha_2 = 0$
3	$q_2 - \pi/2$	$d_3 = 0$	$a_3 = 0.15$	$\alpha_3 = 0$
4	$q_4$	$d_4 = 0$	$a_4 = 0$	$\alpha_4 = \pi/2$
5	$q_5$	$d_5 = 0.157$	$a_5 = 0$	$\alpha_5 = 0$

Từ bảng tham số D-H (Bảng 1), ta lần lượt tính các ma trận biến đổi thuần nhất  $T_i^{i-1}$  giữa hệ tọa độ thứ  $i$  và thứ  $i - 1$  của robot ( $i = 1, \dots, 5$ ):

$$T_1^0 = \begin{bmatrix} c_1 & 0 & s_1 & 0 \\ s_1 & 0 & -c_1 & 0 \\ 0 & 1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}, T_2^1 = \begin{bmatrix} -s_2 & -c_2 & 0 & -a_2 s_2 \\ c_2 & -s_2 & 0 & a_2 c_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

$$T_3^2 = \begin{bmatrix} s_3 & c_3 & 0 & a_3 s_3 \\ -c_3 & s_3 & 0 & -a_3 c_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, T_4^3 = \begin{bmatrix} c_4 & 0 & s_4 & 0 \\ s_4 & 0 & -c_4 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

$$T_5^4 = \begin{bmatrix} c_5 & -s_5 & 0 & 0 \\ s_5 & c_5 & 0 & 0 \\ 0 & 0 & 1 & d_5 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3)$$

trong đó  $\sin(q_i)$ ,  $\cos(q_i)$ ,  $\sin(q_i + q_j)$ , và  $\cos(q_i + q_j)$  lần lượt được ký hiệu bởi  $s_i$ ,  $c_i$ ,  $s_{ij}$ , và  $c_{ij}$  (với  $i, j = 1, \dots, n; n = 5$ ). Sau đó tiếp tục tính các ma trận biến đổi thuần nhất  $T_i^0$  giữa hệ tọa độ thứ  $i$  và hệ tọa độ gốc:

$$\mathbf{T}_1^0 = \begin{bmatrix} c_1 & 0 & s_1 & 0 \\ s_1 & 0 & -c_1 & 0 \\ 0 & 1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \mathbf{T}_2^0 = \begin{bmatrix} -c_1 s_2 & -c_1 c_2 & s_1 & -c_1 a_2 s_2 \\ -s_1 s_2 & -s_1 c_2 & -c_1 & -s_1 a_2 s_2 \\ c_2 & -s_2 & 0 & a_2 c_2 + d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4)$$

$$\mathbf{T}_3^0 = \begin{bmatrix} c_1 c_{23} & -c_1 s_{23} & s_1 & -c_1 (a_2 s_2 - c_{23} a_3) \\ s_1 c_{23} & -s_1 s_{23} & -c_1 & -s_1 (a_2 s_2 - c_{23} a_3) \\ s_{23} & c_{23} & 0 & a_2 c_2 + a_3 s_{23} + d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5)$$

$$\mathbf{T}_4^0 = \begin{bmatrix} c_1 c_{234} & s_1 & c_1 s_{234} & -(a_2 s_2 - a_3 c_{23}) c_1 \\ s_1 c_{234} & -c_1 & s_1 s_{234} & -(a_2 s_2 - a_3 c_{23}) s_1 \\ s_{234} & 0 & -c_{234} & a_2 c_2 + a_3 s_{23} + d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (6)$$

$$\mathbf{T}_5^0 = \begin{bmatrix} c_1 c_{234} c_5 + s_1 s_5 & -c_1 c_{234} s_5 + s_1 c_5 \\ s_1 c_{234} c_5 - c_1 s_5 & -s_1 c_{234} s_5 - c_1 c_5 \\ s_{234} c_5 & -s_{234} s_5 \\ 0 & 0 \\ c_1 s_{234} & c_1 (a_3 c_{23} - a_2 s_2 + s_{234} d_5) \\ s_1 s_{234} & s_1 (a_3 c_{23} - a_2 s_2 + s_{234} d_5) \\ -c_{234} & -c_{234} d_5 + a_3 s_{23} + a_2 c_2 + d_1 \\ 0 & 1 \end{bmatrix} \quad (7)$$

Theo kết quả đã được công bố trong bài báo [18], dạng ma trận của phương trình động lực học cho robot (8) và các ma trận/vector tham số (9)-(11) sẽ được áp dụng và tính toán ra.

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \boldsymbol{\tau} \quad (8)$$

trong đó  $\mathbf{q} = [q_1, q_2, \dots, q_5]^T$  là vector biến khớp,  $\mathbf{M} \in \mathbb{R}^{5 \times 5}$  là ma trận quán tính tổng quát hóa,  $\mathbf{C} \in \mathbb{R}^{5 \times 5}$  là ma trận các thành phần ly tâm và Coriolis,  $\mathbf{g} \in \mathbb{R}^5$  là vector thành phần lực trọng trường,  $\boldsymbol{\tau} \in \mathbb{R}^5$  là vector mô men khớp.

$$\mathbf{M} = \sum_{i=1}^5 (m_i \mathbf{J}_{T_i}^0)^T \mathbf{J}_{T_i}^0 + \mathbf{J}_{R_i}^T \mathbf{I}_i \mathbf{J}_{R_i} \quad (9)$$

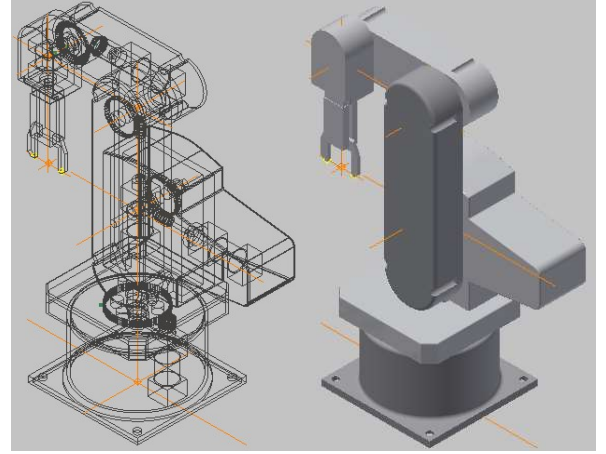
$$\mathbf{C} = \frac{1}{2} \left[ \frac{\partial \mathbf{M}}{\partial \mathbf{q}} (\mathbf{1}_5 \otimes \dot{\mathbf{q}}) + \frac{\partial \mathbf{M}}{\partial \dot{\mathbf{q}}} (\dot{\mathbf{q}} \otimes \mathbf{1}_5) - \left( \frac{\partial \mathbf{M}}{\partial \mathbf{q}} (\dot{\mathbf{q}} \otimes \mathbf{1}_5) \right)^T \right] \quad (10)$$

$$\mathbf{g} = \left( \frac{\partial P}{\partial \mathbf{q}} \right)^T \quad (11)$$

trong đó  $m_i$  là khối lượng của khâu  $i$ ,  $\mathbf{J}_{T_i}^0$  và  $\mathbf{J}_{R_i} \in \mathbb{R}^{3 \times 5}$  lần lượt là ma trận Jacobi tịnh tiến và Jacobi quay của khâu  $i$ ,  $\mathbf{I}_i \in \mathbb{R}^{3 \times 3}$  là ma trận inertia tensor của khâu  $i$ ,  $\mathbf{1}_5$  là ma trận đơn vị  $5 \times 5$ ,  $P$  là tổng thế năng của robot,  $\otimes$  là toán tử phép nhân Kronecker giữa hai ma trận.

### 3. Mô hình Simscape Multibody của robot VNR T1

Dựa trên các bản vẽ kỹ thuật của robot VNR T1 được cung cấp bởi công ty AKB Việt Nam, bằng các phần mềm thiết kế CAD 3D chuyên nghiệp như SolidWork, Onshape, hoặc Autodesk Inventor có thể xây dựng được mô hình 3D của robot VNR T1 (Hình 4) với các thông số quan trọng như khối lượng  $m_i$ , vị trí điểm trọng tâm  $\mathbf{r}_{ci}$ , ma trận inertia tensor  $\mathbf{I}_i$  của các khâu (biểu diễn trong hệ tọa độ thứ  $i$ ) xấp xỉ với robot thật.



Hình 4: Mô hình CAD 3D của robot VNR T1 xây dựng bằng Autodesk Inventor.

$$m_0 = 6.2$$

$$m_1 = 2.7, \mathbf{r}_{C_1} = \begin{bmatrix} -13.746 \\ -61.394 \\ 29.587 \end{bmatrix} \times 10^{-3}, \mathbf{I}_1 = \begin{bmatrix} 13.018 & 1.188 & 0 \\ 1.188 & 12.899 & -2.491 \\ 0 & -2.491 & 17.512 \end{bmatrix} \times 10^{-3}$$

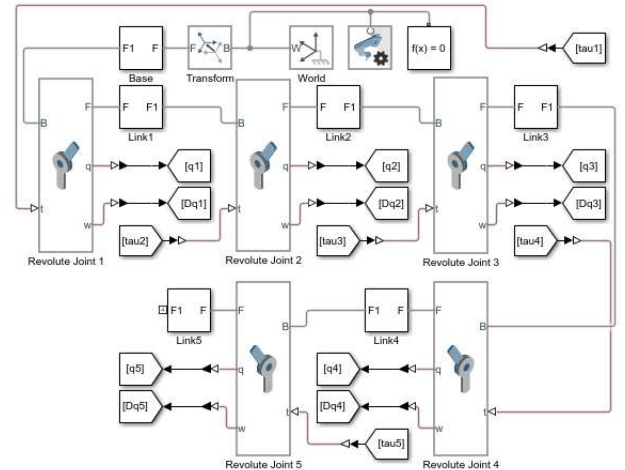
$$m_2 = 1.65, \mathbf{r}_{C_2} = \begin{bmatrix} -98.497 \\ 14.389 \\ -13.211 \end{bmatrix} \times 10^{-3}, \mathbf{I}_2 = \begin{bmatrix} 1.823 & 0 & 0 \\ 0 & 12.602 & 0 \\ 0 & 0 & 12.836 \end{bmatrix} \times 10^{-3}$$

$$m_3 = 1.6, \mathbf{r}_{C_3} = \begin{bmatrix} -90.040 \\ 0 \\ 45.566 \end{bmatrix} \times 10^{-3}, \mathbf{I}_3 = \begin{bmatrix} 1.541 & 0 & 0 \\ 0 & 6.221 & 0 \\ 0 & 0 & 5.789 \end{bmatrix} \times 10^{-3}$$

$$m_4 = 0.6, \mathbf{r}_{C_4} = \begin{bmatrix} 0 \\ 7.648 \\ 3.651 \end{bmatrix} \times 10^{-3}, \mathbf{I}_4 = \begin{bmatrix} 548 & 0 & 0 \\ 0 & 521 & 0 \\ 0 & 0 & 302 \end{bmatrix} \times 10^{-6}$$

$$m_5 = 0.3, \mathbf{r}_{C_5} = \begin{bmatrix} 0 \\ -0.962 \\ -71.180 \end{bmatrix} \times 10^{-3}, \mathbf{I}_5 = \begin{bmatrix} 219 & 0 & 0 \\ 0 & 244 & 0 \\ 0 & 0 & 57 \end{bmatrix} \times 10^{-6}$$

trong đó các đơn vị của khối lượng, chiều dài, tensor quán tính lần lượt là kg, m và  $\text{kgm}^2$ . Các file CAD 3D của robot sẽ được liên kết với MATLAB Simscape Multibody để tạo ra mô hình vật lý ảo của robot VNR T1 với các đặc tính vật lý được bảo toàn tương tự nguyên mẫu (Hình 5).



Hình 5: Mô hình vật lý ảo Simscape Multibody của robot VNR T1.

#### 4. Thiết kế bộ điều khiển trượt sử dụng hàm mũ cho robot VNR T1

Từ phương trình (8), do tính chất khả đảo của ma trận quán tính tổng quát hóa  $\mathbf{M}$ , ta có thể chuyển thành phương trình:

$$\ddot{\mathbf{q}} = \mathbf{M}(\mathbf{q})^{-1}(-\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} - \mathbf{g}(\mathbf{q}) + \boldsymbol{\tau}) \quad (12)$$

Đặt lại ký hiệu cho các biến như sau:

$$\begin{cases} \mathbf{x}_1 = \mathbf{q} \\ \mathbf{x}_2 = \dot{\mathbf{x}}_1 = \dot{\mathbf{q}} \\ \mathbf{u} = \boldsymbol{\tau} \end{cases} \quad (13)$$

Thay thế cách đặt biến trạng thái ở (13) vào (12), thu được phương trình trạng thái cho robot là

$$\begin{cases} \dot{\mathbf{x}}_1 = \mathbf{x}_2 \\ \dot{\mathbf{x}}_2 = \mathbf{M}(\mathbf{x}_1)^{-1}(-\mathbf{C}(\mathbf{x}_1, \mathbf{x}_2)\mathbf{x}_2 - \mathbf{g}(\mathbf{x}_1)) + \mathbf{M}(\mathbf{x}_1)^{-1}\mathbf{u} \end{cases} \quad (14)$$

Mặt trượt  $\mathbf{s}$  được chọn theo [1] có dạng:

$$\mathbf{s} = \left(\frac{d}{dt} + \boldsymbol{\alpha}\right)\mathbf{e} = \dot{\mathbf{e}} + \boldsymbol{\alpha}\mathbf{e} \quad (15)$$

trong đó  $\boldsymbol{\alpha} = \text{diag}(\alpha_i) \in \mathbb{R}^{n \times n}$  là ma trận đường chéo với  $\alpha_i > 0$  ( $i = 1, \dots, n$ ;  $n = 5$ ),  $\mathbf{e} = \mathbf{x}_{1d} - \mathbf{x}_1$  là vector sai lệch giữa quỹ đạo mong muốn  $\mathbf{x}_{1d}$  và đáp ứng  $\mathbf{x}_1$ . Ký hiệu  $\text{diag}(\diamond)$  biểu diễn cho một ma trận đường chéo với các phần tử thuộc đường chéo chính là " $\diamond$ ". Hệ số dương  $\alpha_i$  ảnh hưởng trực tiếp đến tốc độ trượt về điểm cân bằng của hệ, nên cần chọn sao cho phù hợp với yêu cầu thiết kế. Xét điều kiện trượt  $\mathbf{S} \cdot \dot{\mathbf{s}} < 0$  ( $\forall t > 0$ ) trong đó ma trận  $\mathbf{S} = \text{diag}(s_i) \in \mathbb{R}^{n \times n}$  với  $s_i$  là phần tử thứ  $i$  của vector  $\mathbf{s}$ . Cách chọn cơ bản cho  $\dot{\mathbf{s}}$  để thỏa mãn điều kiện trượt là sử dụng hàm xác định dấu  $\text{sgn}(\cdot)$  như sau:

$$\dot{\mathbf{s}} = -\mathbf{K}\text{sgn}(\mathbf{s}), \quad \forall t > 0 \quad (16)$$

trong đó vector  $\text{sgn}(\mathbf{s}) \in \mathbb{R}^n$  được định nghĩa là  $\text{sgn}(\mathbf{s}) := [\text{sgn}(s_1), \text{sgn}(s_2), \dots, \text{sgn}(s_n)]^T$ , ma trận  $\mathbf{K} = \text{diag}(k_i) \in \mathbb{R}^{n \times n}$ ,  $k_i > 0$ ,  $i = 1, \dots, n$ . Hàm  $\text{sgn}(s_i)$  được mô tả bởi (17):

$$\text{sgn}(s_i) = \begin{cases} -1 & \text{khi } s_i < 0 \\ 0 & \text{khi } s_i = 0 \\ 1 & \text{khi } s_i > 0 \end{cases} \quad (17)$$

Theo cách đặt các biến trạng thái, từ các phương trình (14) và (15) ta có

$$\begin{aligned} \dot{\mathbf{s}} &= \boldsymbol{\alpha}\dot{\mathbf{e}} + \ddot{\mathbf{e}} \\ &= \boldsymbol{\alpha}\dot{\mathbf{e}} + (\ddot{\mathbf{x}}_{1d} - \ddot{\mathbf{x}}_1) \\ &= \boldsymbol{\alpha}\dot{\mathbf{e}} + (\ddot{\mathbf{x}}_{1d} - \dot{\mathbf{x}}_2) \\ &= \boldsymbol{\alpha}\dot{\mathbf{e}} + \ddot{\mathbf{x}}_{1d} - \mathbf{M}^{-1}(-\mathbf{C}\mathbf{x}_2 - \mathbf{g} + \mathbf{u}) \end{aligned} \quad (18)$$

Cân bằng hai phương trình (16) và (18) thu được

$$\boldsymbol{\alpha}\dot{\mathbf{e}} + \ddot{\mathbf{x}}_{1d} - \mathbf{M}^{-1}(-\mathbf{C}\mathbf{x}_2 - \mathbf{g} + \mathbf{u}) = -\mathbf{K}\text{sgn}(\mathbf{s}) \quad (19)$$

Suy ra bộ điều khiển trượt cơ bản là

$$\mathbf{u} = \mathbf{M}(\boldsymbol{\alpha}\dot{\mathbf{e}} + \ddot{\mathbf{x}}_{1d} + \mathbf{K}\text{sgn}(\mathbf{s})) + \mathbf{C}\mathbf{x}_2 + \mathbf{g} \quad (20)$$

Luật điều khiển này có chứa thành phần không liên tục  $\text{sgn}(s_i)$  nên khi hệ thống đang ở trên mặt trượt thì hàm  $\text{sgn}(s_i)$  sẽ làm cho tín hiệu điều khiển bị thăng giáng một cách đột ngột gây ra hiện tượng rung trong hệ. Một cách thông dụng để làm giảm độ rung là thay thế hàm xác định dấu  $\text{sgn}(\cdot)$  bằng hàm tỉ lệ bão hòa  $\text{sat}(\cdot)$ . Khi đó luật chuyển mạch là

$$\dot{\mathbf{s}} = -\mathbf{K}\text{sat}(\mathbf{s}, \mathbf{a}), \quad \forall t > 0 \quad (21)$$

trong đó vector  $\text{sat}(\mathbf{s}, \mathbf{a}) \in \mathbb{R}^n$  được định nghĩa là  $\text{sat}(\mathbf{s}, \mathbf{a}) := [\text{sat}(s_1/a_1), \text{sat}(s_2/a_2), \dots, \text{sat}(s_n/a_n)]^T$ ,  $a_i$  là hằng số dương, và hàm  $\text{sat}(s_i/a_i)$  được mô tả theo (22).

$$\text{sat}(s_i/a_i) = \begin{cases} -1 & \text{khi } s_i \leq -a_i \\ \frac{s_i}{a_i} & \text{khi } -a_i < s_i < a_i \\ 1 & \text{khi } s_i \geq a_i \end{cases} \quad (22)$$

Cân bằng hai phương trình (18) và (21) suy ra bộ điều khiển trượt sử dụng hàm tỉ lệ bão hòa:

$$\mathbf{u} = \mathbf{M}(\boldsymbol{\alpha}\dot{\mathbf{e}} + \ddot{\mathbf{x}}_{1d} + \mathbf{K}\text{sat}(\mathbf{s}, \mathbf{a})) + \mathbf{C}\mathbf{x}_2 + \mathbf{g} \quad (23)$$

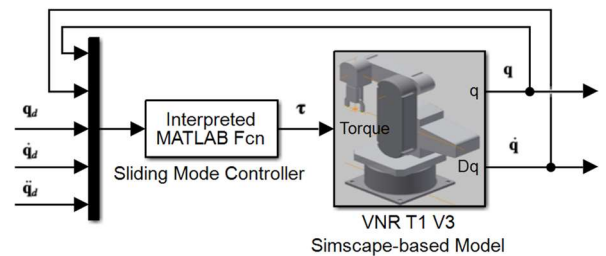
Hàm tỉ lệ bão hòa đã làm tốt vai trò hạn chế hiện tượng rung trong tín hiệu điều khiển, nhưng chỉ làm cho hệ thống hội tụ về một vùng lân cận gốc hệ tọa độ, còn hàm  $\text{sgn}$  sẽ đảm bảo hệ thống hội tụ về không. Ta đều biết rằng, tốc độ đạt tới mặt trượt phụ thuộc vào hệ số  $k_i$ . Khi  $k_i$  tăng thì tốc độ đạt tới mặt trượt tăng, nhưng đồng thời cũng làm cho mức độ rung của hệ thống tăng. Ứng dụng kết quả nghiên cứu ở tài liệu [19], trong bài báo này các tác giả sử dụng bộ điều khiển trượt dùng hàm mũ có khả năng làm giảm thời gian tiến tới mặt trượt và hạn chế độ rung cho robot VNR T1. Luật chuyển mạch sử dụng hàm mũ được đưa ra như sau

$$\dot{\mathbf{s}} = -\mathbf{N}^{-1}\bar{\mathbf{K}}\text{sgn}(\mathbf{s}), \quad \forall t > 0 \quad (24)$$

trong đó  $\bar{\mathbf{K}} = \text{diag}(\bar{k}_i) \in \mathbb{R}^{n \times n}$ ,  $\bar{k}_i > 0$ ,  $\mathbf{N} = \text{diag}(n_i) \in \mathbb{R}^{n \times n}$  với  $n_i = \beta_i + (1 - \beta_i)e^{-\rho|s_i|^\rho}$ ,  $0 < \beta_i < 1$ ,  $0 < \rho$ , và  $\rho$  là một số nguyên dương. Bộ điều khiển trượt sử dụng hàm mũ thu được có dạng

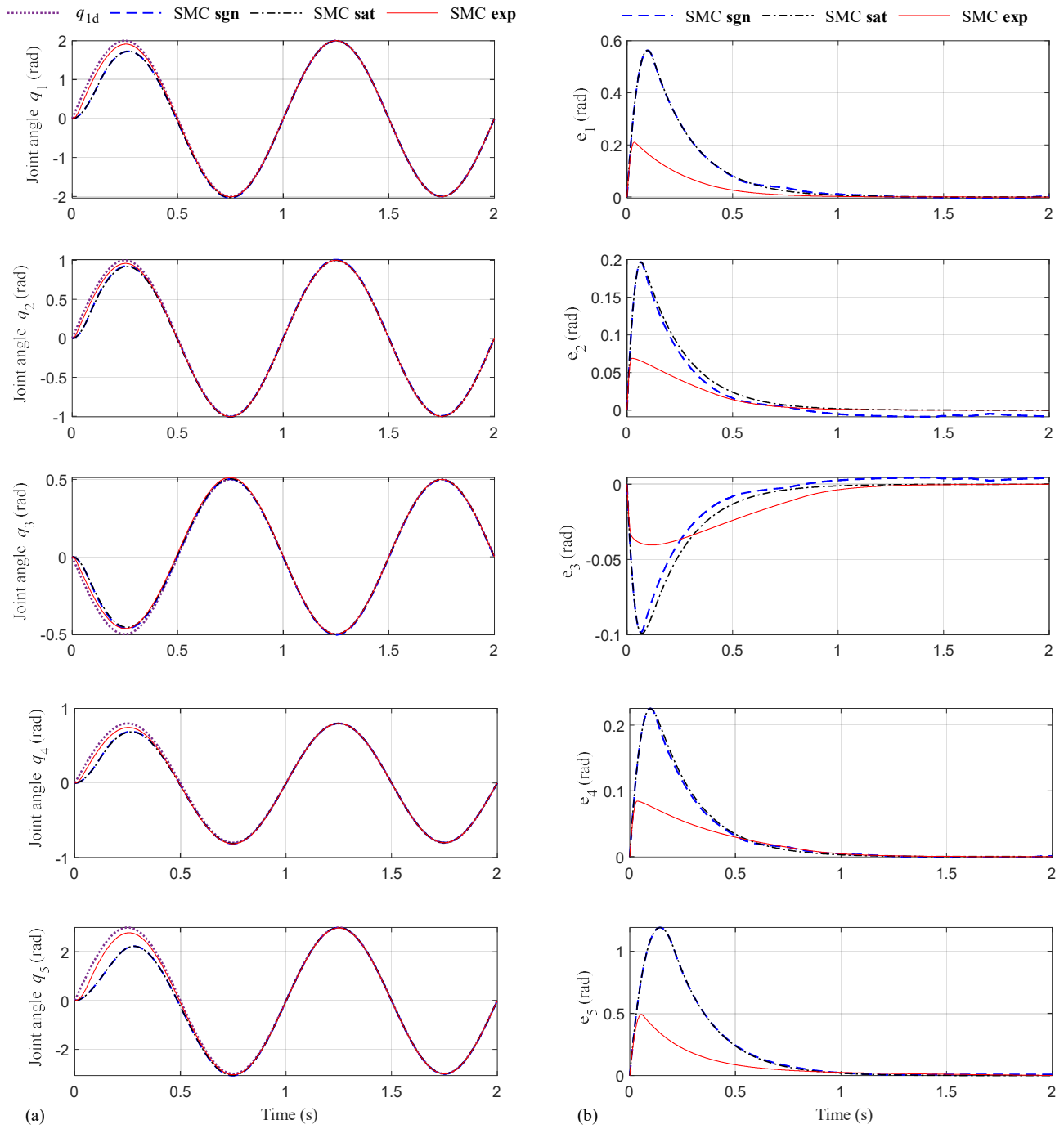
$$\mathbf{u} = \mathbf{M}(\boldsymbol{\alpha}\dot{\mathbf{e}} + \ddot{\mathbf{x}}_{1d} + \mathbf{N}^{-1}\bar{\mathbf{K}}\text{sgn}(\mathbf{s})) + \mathbf{C}\mathbf{x}_2 + \mathbf{g} \quad (25)$$

Do  $n_i$  luôn dương nên vẫn đảm bảo tính ổn định của hệ. Thêm vào đó, sự thay đổi giá trị của  $n_i$  theo độ lớn của  $s_i$  sẽ làm thay đổi tốc độ tiến về mặt trượt. Khi còn tồn tại sai lệch lớn, tức là  $|s_i|$  có giá trị lớn thì  $e^{-\rho|s_i|^\rho}$  xấp xỉ bằng 0, nên  $(1 - \beta_i)e^{-\rho|s_i|^\rho}$  cũng xấp xỉ bằng 0 và dẫn đến  $n_i \approx \beta_i$ . Kết quả là  $\bar{k}_i/n_i$  xấp xỉ bằng  $\bar{k}_i/\beta_i > \bar{k}_i$ . Nếu hệ số  $\beta_i$  càng nhỏ thì tốc độ tiến về mặt trượt càng nhanh. Khi hệ thống đã tiến gần mặt trượt và/hoặc nằm trên mặt trượt, chỉ còn tồn tại sai lệch nhỏ, tức là  $|s_i|$  nhỏ xấp xỉ 0 thì  $e^{-\rho|s_i|^\rho}$  xấp xỉ  $e^0 = 1$ , nên  $(1 - \beta_i)e^{-\rho|s_i|^\rho} \approx (1 - \beta_i)$  và dẫn đến  $n_i$  hội tụ về 1. Kéo theo kết quả là  $\bar{k}_i/n_i \approx \bar{k}_i$ . Với  $\bar{k}_i$  được chọn trước đủ nhỏ để giảm thiểu độ rung. Việc sử dụng hàm mũ trong hệ số tỉ lệ của luật chuyển mạch đã làm tăng chất lượng của hệ thống hơn so với hệ thống sử dụng bộ điều khiển trượt cơ bản. Thật vậy, khi chọn  $\bar{k}_i = k_i$  thì hai bộ điều khiển có cùng mức độ rung như nhau, nhưng tốc độ đạt tới mặt trượt của bộ điều khiển trượt sử dụng hàm mũ sẽ nhanh hơn. Nếu chọn  $\bar{k}_i < k_i$  và  $\bar{k}_i/\beta_i = k_i$  thì hai bộ điều khiển có cùng tốc độ đạt tới mặt trượt, nhưng mức độ rung của bộ điều khiển sử dụng hàm mũ sẽ ít hơn. Còn nếu chọn  $\bar{k}_i < k_i$  và  $\bar{k}_i/\beta_i > k_i$  thì bộ điều khiển trượt sử dụng hàm mũ vừa có tốc độ đạt tới mặt trượt nhanh hơn vừa đảm bảo mức độ rung nhỏ hơn.

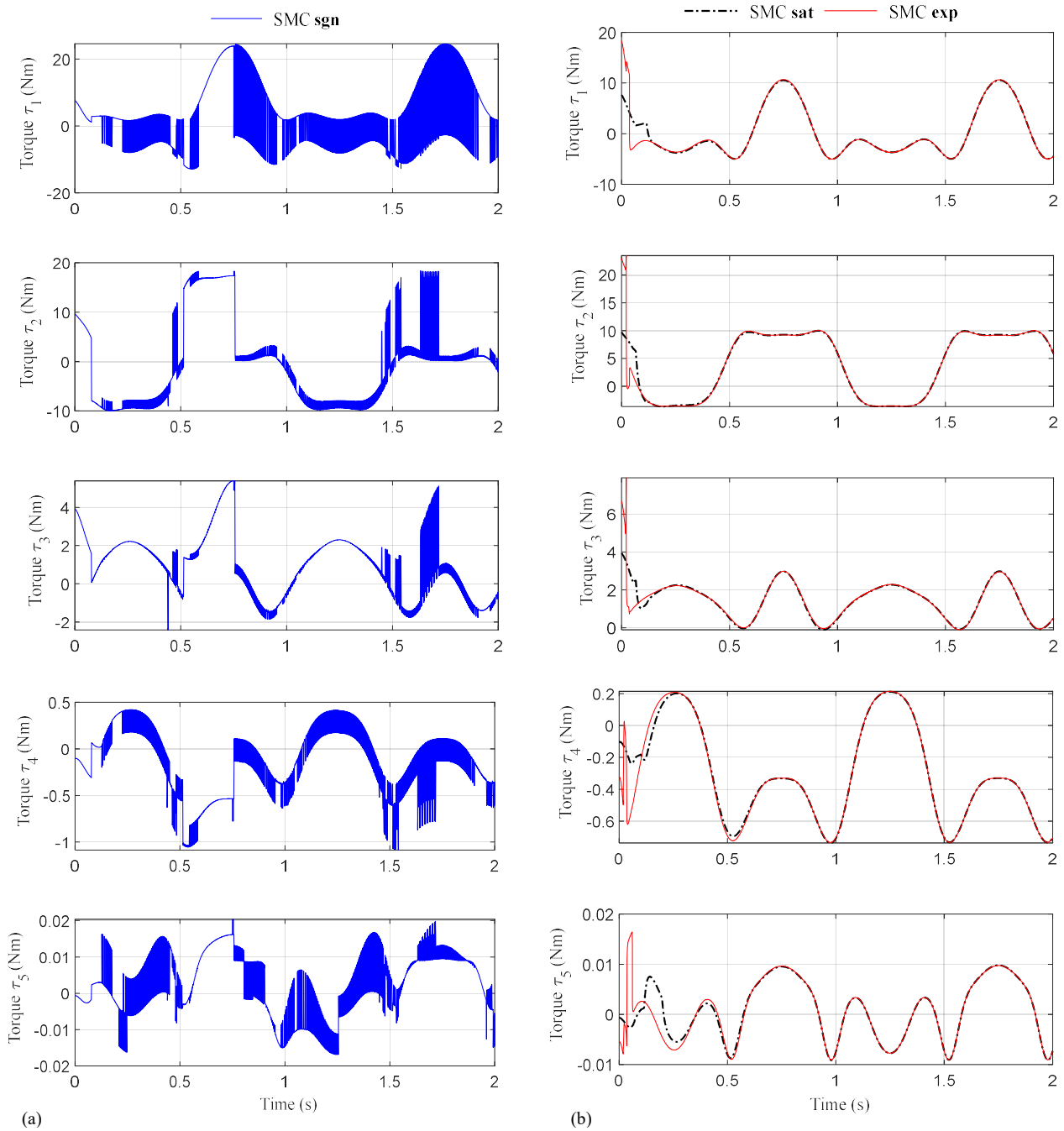


Hình 6: Sơ đồ mô phỏng điều khiển trượt cho robot VNR T1.





**Hình 7:** (a) Đáp ứng khớp  $q$  và (b) sai lệch bám  $e$  tương ứng với ba bộ điều khiển trượt sử dụng hàm  $sgn$ , hàm  $sat$ , và hàm mũ  $exp$ .



**Hình 8:** Tín hiệu mô men  $\tau$  tương ứng với ba bộ điều khiển trượt sử dụng (a) hàm sgn; (b) hàm sat, và hàm mũ exp.

## 5. Mô phỏng và các kết quả

Thực hiện mô phỏng hệ thống điều khiển trượt cho robot VNR T1 bằng phần mềm MATLAB/Simulink với sơ đồ mô phỏng được mô tả trong Hình 6. Ba bộ điều khiển trượt: sử dụng hàm xác định dấu sgn ở công thức (20), sử dụng hàm tỉ lệ bão hòa sat (23), và sử dụng hàm mũ (25), sẽ lần lượt được áp dụng cho robot VNR T1 để phân tích và so sánh. Quỹ đạo của các góc khớp đặt (theo radian) có dạng hình sin được chọn là

$$\begin{aligned}
 q_{1d} &= 2 \sin(2\pi t) & q_{4d} &= 0.8 \sin(2\pi t) \\
 q_{2d} &= \sin(2\pi t) & q_{5d} &= 3 \sin(2\pi t) \\
 q_{3d} &= -0.5 \sin(2\pi t)
 \end{aligned} \quad (26)$$

Để thuận lợi cho việc đánh giá chất lượng của các bộ điều khiển trượt, cả ba bộ điều khiển trượt đều có chung một mặt trượt (15) với  $\alpha_1 = \alpha_2 = \alpha_3 = \alpha_4 = \alpha_5 = 5$ . Thông số của bộ điều khiển trượt cơ bản sử dụng hàm sgn là  $k_1 = 100, k_2 = 80, k_3 = 40, k_4 = 40, k_5 = 90$ . Bên cạnh việc dùng chung tham số  $k_i$  với bộ điều khiển trượt cơ bản, bộ điều khiển trượt sử dụng hàm tỉ lệ bão hòa sẽ có thêm tham số  $a_i$  được chọn là  $a_1 = a_2 = a_3 = a_4 = a_5 = 1$ . Để làm nổi bật khả năng hạn chế độ rung mà vẫn tăng tốc độ hội tụ của bộ điều khiển trượt sử dụng hàm mũ, các tham số  $\bar{k}_i$  và  $\beta_i$  được chọn như sau:  $\bar{k}_1 = 0.5, \bar{k}_2 = 0.4, \bar{k}_3 = 0.2, \bar{k}_4 = 0.2, \bar{k}_5 = 0.3$ , và  $\beta_1 = \beta_2 = \beta_3 = \beta_4 = \beta_5 = 0.0015$ . Hai tham số còn lại lần lượt là  $\rho = 30$  và  $q = 2$ . Các kết quả mô

phòng cho đáp ứng góc khớp  $q$  và tín hiệu mô men  $\tau$  lần lượt được thể hiện trong Hình 7 và Hình 8.

Chất lượng đáp ứng  $q$  và sai lệch  $e$  dưới sự điều khiển của bộ điều khiển trượt cơ bản sử dụng hàm sgn và hàm sat gần như tương đương nhau (Hình 7). Điều này cũng dễ hiểu vì chúng có cùng hệ số khuếch đại  $k_i$  được chọn nên tốc độ ổn định là như nhau. Trong Hình 7 cũng đã thể hiện rõ khả năng tăng tốc độ ổn định cho hệ thống của bộ điều khiển trượt sử dụng hàm mũ. Đáp ứng khớp bám theo quỹ đạo đặt nhanh hơn và với sai lệch nhỏ hơn so với hai trường hợp sử dụng bộ điều khiển trượt còn lại. Khi sử dụng bộ điều khiển trượt truyền thống thì ở khớp 2 và khớp 3 của robot vẫn còn tồn tại sai lệch dù khá nhỏ (Hình 7b); nhưng với bộ điều khiển trượt dùng hàm mũ thì sai lệch này đã bị triệt tiêu.

Việc sử dụng hàm xác định dấu đã làm cho tín hiệu điều khiển bị rung với tần số cao (Hình 8a), và rõ ràng là hàm tỉ lệ bão hòa đã làm tốt nhiệm vụ của mình trong việc hạn chế độ rung ở tín hiệu mô men cấp cho robot (Hình 8b). Với tham số  $a_i$  đã chọn cho hàm sat, tín hiệu điều khiển gần như không còn rung. Trong khoảng thời gian ngắn kể từ lúc robot bắt đầu hoạt động thì sai lệch bám còn lớn (Hình 7b), bộ điều khiển trượt sử dụng hàm mũ đã thể hiện rõ ưu thế của nó khi xuất ra mô men lớn để nhanh chóng kéo trạng thái của biến sai lệch về mặt trượt, qua đó làm giảm tổng thời gian quá độ của hệ thống. Kết quả này là do hệ số khuếch đại của bộ điều khiển trượt sử dụng hàm mũ lúc đó xấp xỉ giá trị  $k_i/\beta_i > k_i$  nên tạo ra tín hiệu điều khiển lớn hơn so với tín hiệu điều khiển của bộ điều khiển trượt sử dụng hàm sgn cũng như hàm sat.

## 6. Kết luận

Bài báo trình bày về cách tiếp cận thiết kế bộ điều khiển trượt sử dụng hàm mũ của biến trượt trong điều kiện trượt. Luật điều khiển trượt cơ bản dựa trên cách lựa chọn điều kiện trượt truyền thống là chỉ sử dụng hệ số khuếch đại thuần túy đã làm cho yêu cầu về tốc độ ổn định và yêu cầu hạn chế hiện tượng rung bị mâu thuẫn với nhau. Việc sử dụng hàm tỉ lệ bão hòa thay thế cho hàm xác định dấu đã hạn chế tốt hiện tượng rung nhưng vẫn chưa hỗ trợ rút ngắn thời gian quá độ của hệ thống. Bằng cách sử dụng hàm mũ trong điều kiện chuyển mạch, tốc độ tiến tới mặt trượt đã được cải thiện đáng kể dẫn đến thời gian ổn định của toàn hệ được nhanh hơn. Thêm vào đó, tín hiệu điều khiển sẽ được duy trì ở giá trị đủ nhỏ trong quá trình trượt trên mặt trượt, nên hiện tượng rung đã được hạn chế đáng kể. Bộ điều khiển trượt sử dụng hàm mũ đã thể hiện chất lượng tốt khi đưa vào điều khiển mô hình vật lý ảo của robot VNR T1. Kết quả này góp phần làm tăng khả năng áp dụng bộ điều khiển trượt cho các đối tượng phi tuyến khác.

## Tài liệu tham khảo

- [1] J. E. Slotine and W. Li (1991) *Applied Nonlinear Control*. Prentice Hall, Englewood Cliffs, NJ, USA.
- [2] V. Utkin (1992) *Sliding Mode in Control and Optimization*. Springer-Verlag, Berlin, Germany.
- [3] N. D. Phước (2012) *Phân tích và điều khiển hệ phi tuyến*. NXB Bách Khoa Hà Nội.
- [4] R. Gorez (1999) *Conclusions of 5-year investigations in sliding mode control of manipulators*. European Control Conference (ECC), pp. 3546–3551, Karlsruhe, Germany.

- [5] Yew-Wen Liang, Sheng-Dong Xu, Der-Cherng Liaw, et al. (2008) *A Study of T-S Model-Based SMC Scheme With Application to Robot Control*. IEEE Transactions on Industrial Electronics, vol. 55, no. 11, pp. 3964–3971.
- [6] N. T. Hiệp and P. T. Cát (2008) *Điều khiển rôbot theo nguyên lý trượt sử dụng mạng nơron*. Tạp chí Tin học và điều khiển học, vol. 24, no. 3, pp. 236–246.
- [7] Liangyong Wang, Tianyou Chai, and Lianfei Zhai (2009) *Neural-Network-Based Terminal Sliding-Mode Control of Robotic Manipulators Including Actuator Dynamics*. IEEE Transactions on Industrial Electronics, vol. 56, no. 9, pp. 3296–3304.
- [8] J. Baek, M. Jin, and S. Han (2016) *A New Adaptive Sliding-Mode Control Scheme for Application to Robot Manipulators*. IEEE Transactions on Industrial Electronics, vol. 63, no. 6, pp. 3628–3637.
- [9] F. Baklouti, S. Aloui, and A. Chaari (2016) *Adaptive Fuzzy Sliding Mode Tracking Control of Uncertain Underactuated Nonlinear Systems: A Comparative Study*. Journal of Control Science and Engineering, vol. 2016, pp. 1–12.
- [10] T. H. Nguyễn, P. X. Minh, and N. C. Khoa (2014) *Điều khiển trượt nơron thích nghi bền vững cho robot 3 bậc tự do*. Tạp chí Khoa học và Công nghệ Việt Nam, vol. 52, no. 5, pp. 541–548.
- [11] Y. Xia, Z. Zhu, and M. Fu (2011) *Back-stepping sliding mode control for missile systems based on an extended state observer*. IET Control Theory & Applications, vol. 5, no. 1, pp. 93–102.
- [12] M. B. R. Neila and D. Tarak (2011) *Adaptive Terminal Sliding Mode Control for Rigid Robotic Manipulators*. International Journal of Automation and Computing, vol. 8, no. 2, pp. 215–220.
- [13] N. Q. Hoang and V. D. Vuong (2017) *Sliding mode control for a planar parallel robot driven by electric motors in a task space*. Journal of Computer Science and Cybernetics, vol. 33, no. 4, pp. 325–337.
- [14] T. Floquet, J.-P. Barbot, and W. Perruquetti (2003) *Higher-order sliding mode stabilization for a class of nonholonomic perturbed systems*. Automatica, vol. 39, no. 6, pp. 1077–1083.
- [15] G. Bartolini, A. Ferrara, and E. Usai (1998) *Chattering avoidance by second-order sliding mode control*. IEEE Transactions on Automatic Control, vol. 43, no. 2, pp. 241–246.
- [16] V. Parra-Vega and G. Hirzinger (2001) *Chattering-free sliding mode control for a class of nonlinear mechanical systems*. International Journal of Robust and Nonlinear Control, vol. 11, no. 12, pp. 1161–1178.
- [17] Weibing Gao and J. C. Hung (1993) *Variable structure control of nonlinear systems: a new approach*. IEEE Transactions on Industrial Electronics, vol. 40, no. 1, pp. 45–55.
- [18] L. N. Truc, N. V. Quyen, and N. P. Quang (2020) *Dynamic model with a new formulation of Coriolis/centrifugal matrix for robot manipulators*. Journal of Computer Science and Cybernetics, vol. 36, no. 1, pp. 89–104.
- [19] C. J. Fallaha, M. Saad, H. Y. Kanaan, et al. (2011) *Sliding-Mode Robot Control With Exponential Reaching Law*. IEEE Transactions on Industrial Electronics, vol. 58, no. 2, pp. 600–610.