

# Nghiên cứu bộ điều khiển trượt chống rung và mô phỏng PIL cho tay máy robot VNR-T1 5 bậc tự do

## Processor in the loop simulation based anti chattering sliding mode control for 5-d of robot VNR-T1

Lê Ngọc Trúc<sup>1,2</sup>, Trần Văn Chi<sup>1,2</sup>, Nguyễn Hữu Hải<sup>3</sup>, Nguyễn Danh Huy<sup>2</sup>,  
Nguyễn Trọng Các<sup>4</sup>, Nguyễn Tùng Lâm<sup>2\*</sup>

\*Email: lam.nguyentung@hust.edu.vn

<sup>1</sup>Trường Đại học Sư phạm kỹ thuật Hưng Yên

<sup>2</sup>Trường Đại học Bách khoa Hà Nội

<sup>3</sup>Trường Đại học Công nghiệp Hà Nội

<sup>4</sup>Trường Đại học Sao Đỏ

Ngày nhận bài: 25/3/2021

Ngày nhận bài sửa sau phản biện: 02/6/2021

Ngày chấp nhận đăng: 20/6/2021

### 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. Hạn chế điều khiển trượt cơ bản là xuất hiện rung trong hệ thống nên người thiết kế phải cân bằng giữa hai tiêu chí là thời gian quá độ và mức độ rung. Bài báo này là một bước cải tiến mới khi bộ điều khiển trượt chống rung kết hợp với mô phỏng PIL ứng dụng cho robot 5 bậc tự do VNR-T1. Bài báo sử dụng hệ số khuếch đại biến thiên theo hàm mũ trong luật điều khiển trượt để có thể hạn chế độ rung và giảm thời gian quá độ của toàn hệ một cách tốt hơn so với bộ điều khiển trượt cơ bản sử dụng hệ số khuếch đại là hằng số. Bộ điều khiển trượt sử dụng hệ số khuếch đại dạng hàm mũ sẽ tạo ra tín hiệu điều khiển lớn trong giai đoạn tiến tới mặt trượt làm tăng tốc độ đạt tới mặt trượt, và duy trì tín hiệu điều khiển nhỏ trong giai đoạn trượt để giảm độ rung. Các kết quả mô phỏng PIL được trình bày cho thấy tính khả dụng của bộ điều khiển trong ứng dụng robot VNR-T1.

*Từ khóa: Điều khiển trượt; robot công nghiệp; xử lý vòng lặp; chống rung.*

### Abstract

Sliding mode control is vastly used in industrial applications due to its robustness and applicability. In sliding mode control, chattering is unavoidable and it is a trade off between chattering level and dynamical responses of the system. The paper introduces a chattering suppression solution and processor in the loop simulation for 5 - d of VNR-T1 robot. Control gain is varying according to exponential function to limit chattering phenomenon in comparison to conventional sliding mode control. Due to the presence of the exponential varying control gain, the chattering is reduced considerably. The effectiveness of the control is tested via processor in the loop simulation for VNR-T1 robot.

*Keywords: Sliding mode control; industrial robot; processor in the loop; anti chattering.*

### 1. ĐẶT VẤN ĐỀ

Kể từ khi công trình nghiên cứu [1] viết về mặt trượt, giáo trình giảng dạy robot công nghiệp [8], điều khiển truyền động điện thông minh [9] được viết ra để nhằm nghiên cứu các phương pháp điều khiển cho tay máy

robot. Việc thiết kế và mô phỏng cho tay máy đa phần chỉ được áp dụng khi đối tượng và bộ điều khiển đều nằm trên phần mềm chuyên dụng của PC. Cũng rất nhiều tác giả đã đề cập đến mô phỏng PIL áp dụng với kiểm nghiệm kiểm soát vệ tinh [5], đối tượng động cơ, servo, các bài toán truyền động điện [6]. Tuy nhiên, nghiên cứu này PIL được áp dụng mô phỏng trên đối tượng là tay máy VNR-T1 (5DOF) kết hợp với vi điều khiển STM32F407.

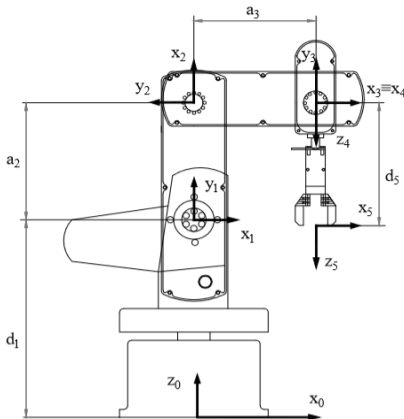
Người phản biện: 1. PGS. TS. Bùi Đăng Thành

2. PGS. TS. Nguyễn Văn Liễn

Discovery [7] với Matlab Simulink trên máy tính để kiểm nghiệm kết quả trước khi tiến đến thực nghiệm. Để phát triển hơn nữa từ các nghiên cứu trước bộ điều khiển và đối tượng khi tiến hành mô phỏng ở bài báo này đã được tách riêng kết hợp với vi điều STM32F407 Discovery với Matlab Simulink trên máy tính để kiểm nghiệm kết quả. Quá trình trên là một bước quan trọng không thể thiếu trong quá trình phát triển để đưa các bộ điều khiển áp vào các đối tượng thực. Từ quỹ đạo đặt của khâu chấp hành cuối, qua việc giải bài toán động học ngược, ta tìm ra được vector góc đặt của các khớp. Cùng với mô hình toán học được xây dựng, ta tiến hành thiết kế bộ điều khiển. Đầu ra của bộ điều khiển là tín hiệu điều khiển, cụ thể là vector momen. Tín hiệu điều khiển này được đưa vào đối tượng điều khiển là mô hình vật lý ảo mô tả robot VNR-T1 được xây dựng bằng Simscape Multibody, tiếp theo kết hợp với vi điều khiển và matlab chuyển mô phỏng sang PIL [10] bước đầu cho việc nhúng thuật toán vào thực tế.

## 2. MÔ HÌNH TOÁN HỌC CỦA ROBOT

Robot VNR T1 là một loại robot công nghiệp 5 - DOF do công ty AKB Việt Nam nghiên cứu và chế tạo, nó có cấu hình và tham số D-H được mô tả như sau:



Hình 1. Mô hình robot và cách gắn hệ tọa độ các khớp

Bảng 1. Bảng tham số D-H

Khớp $i$	$\theta_i$ (rad)	$d_i$ (m)	$a_i$ (m)	$\alpha_i$ (rad)
1	$q_1$	$d_1 = 0,253$	$a_1 = 0$	$\alpha_1 = \frac{\pi}{2}$
2	$q_2 + \frac{\pi}{2}$	$d_2 = 0$	$a_2 = 0,15$	$\alpha_2 = 0$
3	$q_3 - \frac{\pi}{2}$	$d_3 = 0$	$a_3 = 0,15$	$\alpha_3 = 0$
4	$q_4$	$d_4 = 0$	$a_4 = 0$	$\alpha_4 = \frac{\pi}{2}$
5	$q_5$	$d_5 = 0,157$	$a_5 = 0$	$\alpha_5 = 0$

Dạng ma trận của phương trình động lực học robot mà không xét đến ma sát là:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) = \tau \quad (1)$$

Trong đó:

$M = [m_{ki}] \in \mathbb{R}^{n \times n}$  là ma trận quán tính suy rộng;

$C = [c_{ki}] \in \mathbb{R}^{n \times n}$  là ma trận ly tâm và coriolis;

$g = [g_k] \in \mathbb{R}^{n \times 1}$  là vector thành phần lực trọng trường;

$\tau = [\tau_k] \in \mathbb{R}^{n \times 1}$  là vector momen (lực).

Các tham số vector/ma trận trong (1) được tính bằng các công thức sau:

$$M = \sum_{i=1}^5 \left( m_i (J_{T_i}^0)^T J_{T_i}^0 + J_{R_i}^T I_i J_{R_i} \right)$$

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

$$g = \left( \frac{\partial P}{\partial q} \right)^T \quad (2)$$

## 3. MÔ HÌNH HÓA BỞI SIMSCAPE MULTIBODY

Simscape Multibody cung cấp mô phỏng 3D cho các hệ thống cơ khí đa thành phần chẳng hạn như robot, hệ thống đa thân, hệ thống xe hơi. Simscape Multibody cung cấp các khối liên kết, khớp, cảm biến và các thành phần momen xoắn/lực. Simscape Multibody cho phép người dùng vẽ với các khối đơn giản. Tuy nhiên, hoàn toàn có thể sử dụng CAD để thiết kế (ví dụ như: Autodesk Inventor, SolidWorks,...) và MATLAB/Simscape Multibody có thể sử dụng các tệp tin CAD để tạo ra hệ thống mô hình.

Dựa trên mô hình CAD của robot VNR T1, chúng tôi khai thác Autodesk Inventor để khám phá mô hình robot và xây dựng mô hình 3D của robot VNR T1. Các giá trị xấp xỉ của khối lượng  $m_i$ , vị trí điểm trọng tâm  $r_{ci}$  và ma trận momen quán tính  $I_i$  của từng khâu (biểu diễn trong hệ tọa độ thứ  $i$ ) lần lượt thể hiện như Bảng 2:

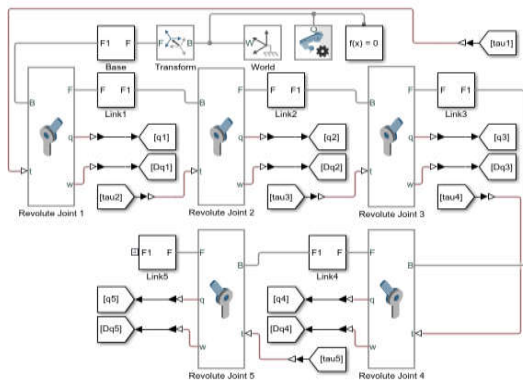
Bảng 2. Thông số của robot VNR T1

Base	$a_i = 6,2$	
Link 1	$m_1 = 2.7, r_{c1} = \begin{bmatrix} -13.746 \\ -61.394 \\ 29.587 \end{bmatrix} 10^{-3}$	$I_1 = \begin{bmatrix} 13.018 & 1.188 & 0 \\ 1.188 & 12.899 & -2.491 \\ 0 & -2.491 & 17.512 \end{bmatrix} 10^{-3}$
Link 2	$m_2 = 1.65, r_{c2} = \begin{bmatrix} -98.497 \\ 14.389 \\ -13.211 \end{bmatrix} 10^{-3}$	$I_2 = \begin{bmatrix} 1.823 & 0 & 0 \\ 0 & 12.602 & 0 \\ 0 & 0 & 12.836 \end{bmatrix} 10^{-3}$
Link 3	$m_3 = 1.6, r_{c3} = \begin{bmatrix} -90.040 \\ 0 \\ 45.566 \end{bmatrix} 10^{-3}$	$I_3 = \begin{bmatrix} 1.541 & 0 & 0 \\ 0 & 6.221 & 0 \\ 0 & 0 & 5.789 \end{bmatrix} 10^{-3}$
Link 4	$m_4 = 0.6, r_{c4} = \begin{bmatrix} 0 \\ 7.648 \\ 3.651 \end{bmatrix} 10^{-3}$	$I_4 = \begin{bmatrix} 548 & 0 & 0 \\ 0 & 521 & 0 \\ 0 & 0 & 302 \end{bmatrix} 10^{-6}$
Link 5	$m_5 = 0.3, r_{c5} = \begin{bmatrix} 0 \\ -0.962 \\ -71.180 \end{bmatrix} 10^{-3}$	$I_5 = \begin{bmatrix} 219 & 0 & 0 \\ 0 & 244 & 0 \\ 0 & 0 & 57 \end{bmatrix} 10^{-6}$

Trong đó, các đơn vị khối lượng, chiều dài, tenxơ quán tính được tính theo kg, m và kg, m<sup>2</sup>. Autodesk Inventor có thể xuất file CAD của robot và tập hợp các file được đưa vào MATLAB Simscape Multibody để tự động tạo mô hình Simscape Multibody của robot VNR T1 với cùng mô tả vật lý.



Hình 2. Mô hình 3D tay máy robot



Hình 3. Mô hình Simscape Multibody

## 4. ĐIỀU KHIỂN TRƯỢT

### 4.1. Tổng quan

Điều khiển theo nguyên lý trượt, hay còn gọi là điều khiển trong chế độ trượt là một phương pháp điều khiển bền vững cho hệ phi tuyến.

Đã có rất nhiều nghiên cứu và rất nhiều các thuật toán để điều khiển hệ thống này như kỹ thuật điều khiển hybrid, kỹ thuật điều khiển tuyến tính, PID, điều khiển mờ... những thuật toán trên, mỗi thuật toán đều có ưu điểm và nhược điểm riêng của mình nhằm giải quyết được từng khía cạnh của bài toán điều khiển. Tuy nhiên, vấn đề mấu chốt ở đây là phải xây dựng một thuật toán điều khiển bền vững trước các nhiễu từ bên ngoài, nhiễu tải và bất định thông số mô hình.

Để đảm bảo các yêu cầu như đã nêu trên trong bài báo sẽ sử dụng thuật toán điều khiển trượt với luật tiếp cận theo hàm mũ (sliding mode control with exponential reaching law). Bộ điều khiển trượt đảm bảo được yêu cầu có thể điều khiển bền vững, đảm bảo hệ thống vẫn có thể điều khiển được trước những thay đổi của tải hay các nhiễu và bất định của hệ thống trong điều kiện cho phép.

Xét hệ không dừng có tín hiệu vào  $u = [u_1, \dots, u_m]^T$ , chứa thành phần bất định  $d = (x, u, t)$  mô tả bởi:

$$\dot{x} = f(x, u, d, t) \quad (3)$$

Trong đó:

$x \in R^n$  là vector trạng thái;

$f(x, u, d, t)$  là các vector hàm liên tục và một mặt cong trơn n chiều, thường được gọi là mặt trượt, được mô tả bởi một vector  $m$  hàm trơn chứa tất cả các quỹ đạo trạng thái mong muốn  $x(t)$  của hệ (theo một chỉ tiêu chất lượng cho trước).

$$s(x, t) = [s_1(x, t), s_2(x, t), \dots, s_m(x, t)]^T = 0 \quad (4)$$

Mặt trượt (4) trên thường gặp ở dạng tổng quát, vì nó có dạng không dừng (cấu trúc mặt trượt bị thay đổi theo thời gian).

Nhiều trường hợp, để đơn giản trong điều khiển sau này và khi điều kiện cho phép, người ta chỉ cần sử dụng mặt trượt dừng (có cấu trúc không biến đổi theo thời gian).

Nhiệm vụ của điều khiển trượt là phải xác định tín hiệu điều khiển  $u$  để đưa hệ (3) tiến về mặt trượt (4) và giữ nó lại trên đó. Ta sẽ ký hiệu tín hiệu điều khiển cần tìm  $u$  đó là:

$$\begin{cases} u = u_{eq}, & s(x, t) = 0 \\ u = u_N + u_{eq}, & s(x, t) \neq 0 \end{cases} \quad (5)$$

Trong đó:

$u_{eq}$  là thành phần tín hiệu giữ  $x(t)$  ở lại trên mặt trượt, tức là nếu đã có:  $s(x_o, t_o) = 0$  với  $x_o = x(t_o)$ . Thì  $u_{eq}$  sẽ phải tạo ra được:  $s(x, t) = 0$  khi  $t > t_o$ ;

$u_N$  là thành phần tín hiệu làm cho  $x(t)$  tiến về mặt trượt. Như vậy, ở trường hợp mặt trượt dừng, khi sử dụng hàm Lyapunov xác định dương:  $V(s) = \frac{1}{2}s^T s$  thì điều kiện đủ để  $x(t)$  tiến về mặt trượt là tín hiệu điều khiển  $u_N$  phải tạo ra được:

$$\dot{V}(s) = s^T \dot{s} < 0 \text{ khi } s(x) \neq 0 \quad (6)$$

Điều kiện này được gọi là điều kiện trượt và sử dụng với mặt trượt dừng. Khi đó các thành phần  $u_{eq}$  và  $u_N$  sẽ được xác định như sau:

Khi hệ (3) là hệ rõ và có cấu trúc:

$$\dot{x} = f(x, t) + B(x, t)u$$

Trong đó:

$$B(x, t) = (b_1(x, t), \dots, b_m(x, t)) \quad (7)$$

Từ điều kiện (6) ta có:

$$\dot{s}(x, t) = \frac{\partial s}{\partial x} \frac{dx}{dt} = \frac{\partial s}{\partial x} [f(x, t) + B(x, t)u_{eq}] = 0 \quad (8)$$

Vậy nếu ma trận:  $\frac{\partial s}{\partial x} B(x, t) \in R^{m \times m}$  không suy biến thì ta có:

$$u_{eq} = - \left( \frac{\partial s}{\partial x} B(x, t) \right)^{-1} \frac{\partial s}{\partial x} f(x, t) \quad (9)$$

Từ điều kiện (7) cho mặt trượt (4). Tín hiệu  $u = u_{eq} + \dots$

$$\begin{aligned} \dot{s}(x,t) &= \frac{\partial s}{\partial x} \frac{\partial x}{\partial t} \\ &= \frac{\partial s}{\partial x} [f(x,t) + B(x,t)(u_{eq} + \Delta)] \\ &= \frac{\partial s}{\partial x} B(x,t)\Delta \end{aligned} \quad (10)$$

Như vậy, ta chỉ cần chọn sao cho thỏa mãn điều kiện (7) là có thể xác định được tín hiệu điều khiển cho hệ.

$$\frac{\partial s}{\partial x} B(x,t)\Delta = -K \text{sign}(s) \quad (11)$$

Trong đó:

$K$  là ma trận hằng số dương và hàm  $\text{sign}(s)$  được định nghĩa như sau:

$$\text{sign}(s) = \begin{cases} 1 & \text{khi } s > 0 \\ -1 & \text{khi } s < 0 \\ 0 & \text{khi } s = 0 \end{cases} \quad (12)$$

Suy ra:  $\dot{V}(s) = s^T \dot{s} = -s^T K \text{sign}(s) < 0$

Ta xác định được:

$$\Delta = -\left(\frac{\partial s}{\partial x} B(x,t)\right)^{-1} K \text{sign}(s) \quad (13)$$

#### 4.2. Thiết kế bộ điều khiển cho VNR T1

Phương trình mô hình hóa của robot được xác định bởi công thức:

$$\begin{aligned} M(q)\ddot{q} + C(q,\dot{q})\dot{q} + g(q) &= u \\ \Rightarrow \ddot{q} &= M(q)^{-1}(-C(q,\dot{q})\dot{q} - g(q) + u) \end{aligned} \quad (14)$$

$$\text{Đặt } \begin{cases} x_1 = q \\ x_2 = \dot{x}_1 = \dot{q} \end{cases}$$

Từ phương trình (14) ta có:

$$x_2 = \dot{x}_1 = \dot{q} \Rightarrow \dot{x}_2 = \ddot{x}_1 = \ddot{q} \quad (15)$$

Từ (14) vào (15) ta thu được

$$\dot{x}_2 = M(q)^{-1}(-C\dot{q} - g + u) \quad (16)$$

Chọn mặt trượt có dạng:

$$s = \dot{z} + \lambda z \quad (17)$$

$$\text{Với } \begin{cases} z = x_1 - x_{1d} \\ \lambda > 0 \end{cases}$$

Từ (17) suy ra:

$$\dot{s} = \dot{z} + \lambda \dot{z} \quad (18)$$

Thay (17) vào (18) ta thu được:

$$\dot{s} = (\dot{x}_1 - \dot{x}_{1d}) + \lambda(x_1 - x_{1d}) \quad (19)$$

Thay (15) vào (19) ta thu được:

$$\dot{s} = (\dot{x}_2 - \dot{x}_{1d}) + \lambda(x_2 - \dot{x}_{1d}) \quad (20)$$

Trong đó:

$x_{1d}$  là tín hiệu đặt. Nhiệm vụ của điều khiển là tính giá trị momen điều khiển thích hợp sao cho giá trị góc quay

thực tế của từng khớp bám theo giá trị đặt. Như vậy có thể thấy rằng, việc duy trì quỹ đạo trạng thái trên mặt trượt đã chọn sẵn sẽ dẫn đến  $x_{1d} \rightarrow x_1$ . Khi  $s = 0$  thì phương trình (17) trở thành  $\dot{z} = -\lambda z$ . Phương trình này chỉ có một nghiệm duy nhất  $z = 0$ , hay nó đặc trưng cho hệ động lực học ổn định tiệm cận, từ đó điều kiện bám  $x_{1d} \rightarrow x_1$  được thỏa mãn. Do đó, vấn đề cần giải quyết của luật điều khiển là tìm momen tác động tại các khớp sao cho duy trì được quỹ đạo robot trên mặt trượt.

Tín hiệu điều khiển được thiết kế dựa trên sự tồn tại của một hàm Lyapunov xác định dương sau:

$$\dot{V} = s^T \cdot \dot{s} \quad (21)$$

Suy ra:

$$\dot{V} = s^T \cdot \dot{s}$$

Để hệ ổn định thì  $\dot{V} < 0$

Xét trường hợp  $s = 0$  ta có:

$$(\dot{x}_2 - \dot{x}_{1d}) + \lambda(x_2 - \dot{x}_{1d}) = 0 \quad (22)$$

Thay (16) vào (22) ta thu được:

$$M(q)^{-1}(-C(q,\dot{q})\dot{q} - g(q) + u) - \dot{x}_{1d} + \lambda(x_2 - \dot{x}_{1d}) = 0 \quad (23)$$

Như vậy ta tính được:

$$\begin{aligned} u_{eq} &= M[\dot{x}_{1d} - \lambda(x_2 - \dot{x}_{1d})] \\ &+ C\dot{q} + g \end{aligned} \quad (24)$$

Tín hiệu  $u_{eq}$  chỉ có tác dụng trên mặt trượt, nên cần phải tìm thêm thành phần giữ lại trạng thái của hệ trượt. Với thiết kế của bộ điều khiển trượt cơ bản ta có:

$$\begin{aligned} \dot{s} &= -K \text{sign}(s) \\ A &= -M(q) \cdot K \text{sign}(s) \end{aligned} \quad (25)$$

Thay (25) vào (21) sẽ cho hàm:

$$\dot{V} = s^T \cdot K \cdot \text{sign}(s) < 0 \quad (26)$$

Như vậy tín hiệu điều khiển của bộ điều khiển trượt cơ bản cần tìm:

$$\begin{aligned} u_N &= u_{eq} + D \\ &= M[-K \cdot \text{sign}(s) + \dot{x}_{1d} - \lambda(x_2 - \dot{x}_{1d})] + C\dot{q} + g \end{aligned} \quad (27)$$

#### 4.3. Điều khiển trượt chống rung

Như đã trình bày ở (4.2) về thiết kế bộ điều khiển trượt cơ bản. Luật điều khiển này cho thấy thành phần điều khiển có thành phần không liên tục  $\text{sign}(s)$ . Điều này tạo ra hiện tượng rung trong hệ. Có thể thấy độ rung của hệ phụ thuộc vào hệ số  $[K]$ . Để có thời gian ổn định nhanh hơn, độ ổn định bền vững, bám giá trị đặt thì giá trị  $[K]$  cần tăng lên. Tuy nhiên, việc tăng  $[K]$  sẽ ảnh hưởng trực tiếp đến độ rung. Để giải quyết vấn đề này chúng ta phải xóa bỏ sự phụ thuộc lẫn nhau giữa thời gian ổn định và độ rung. Để khắc phục hiện tượng này thì người ta sử dụng bộ điều khiển chống rung bằng cách chọn:

$$\dot{s} = \frac{K \cdot s}{N(s)} \quad (28)$$

Trong đó:

$$N = [n_{ij}] \in \sim mxm \quad (29)$$

$$n_i(s_i) = \delta_{0i} + (1 - \delta_{0i})e^{-\alpha_i |s_i|^p} \quad (30)$$

$\delta_{0i}$  là 1 giá trị xác định dương nhỏ hơn 1, p là 1 giá trị nguyên dương và  $\alpha$  là 1 giá trị dương. Ta có thể thấy công thức (28) không ảnh hưởng đến sự ổn định của hệ vì N(s) luôn dương. Từ (30) ta thấy khi tăng, N(s) tiến tới  $\delta_{0i}$ ,  $\frac{[K]}{N(s)}$  tiến tới  $\frac{[K]}{\delta_{0i}}$ , là 1 giá trị lớn hơn [K]. Điều này có ý nghĩa là  $\frac{[K]}{N(s)}$  tăng trong quá trình tiến về mặt trượt hay hệ tiến về mặt trượt nhanh hơn. Và ngược lại giảm, N(s) tiến tới 1,  $\frac{[K]}{N(s)}$  tiến tới [K]. Điều này có ý nghĩa khi hệ đã về mặt trượt thì  $\frac{[K]}{N(s)}$  giảm đồng nghĩa

độ rung giảm. Việc chọn s thay vì  $sign(s)$  trong (28) cũng nhằm giảm độ rung của hệ.

Từ (28) ta có:

$$\dot{s} = N^{-1} Ks \quad (31)$$

Ta cần giải phương trình:

$$f(x, t) + B(x, t)u_N = -N^{-1}Ks \quad (32)$$

Từ đó tính được:

Trên cơ sở xây dựng bộ điều khiển trượt cơ bản ta thu được:

$$u_N = (B(x, t))^{-1} (-N^{-1}Ks - f(x, t)) \quad (33)$$

$$u_N = M[\ddot{x}_{1d} - 1(x_2 - \dot{x}_{1d})] + C\dot{q} + g$$

Chọn:

$$\Delta = -MN^{-1}Ks \quad (34)$$

$$\dot{s} = N^{-1}Ks$$

$$\Rightarrow \dot{V} = -s^T N^{-1}Ks < 0$$

Như vậy, tín hiệu điều khiển của bộ điều khiển trượt chống rung cần tìm:

$$u = u_{eq} + \Delta = M[-N^{-1}Ks + \ddot{x}_{1d} - \lambda(x_2 - \dot{x}_{1d})] + C\dot{q} + g \quad (35)$$

### 5. MÔ PHỎNG PIL (PROCESSOR IN THE LOOP) SỬ DỤNG STM32DISCOVERY

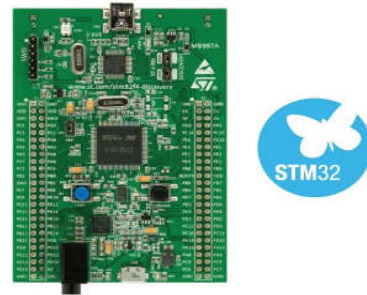
Bộ xử lý trong vòng lặp (PIL) là một kỹ thuật thử nghiệm cho phép các nhà thiết kế đánh giá bộ điều khiển, chạy trong bộ xử lý chuyên dụng, của một dự án trong nền tảng mô phỏng ngoại tuyến. PIL là 1 trong 4 bước kiểm tra và xác minh trong quá trình phát triển một hệ thống:

- Model in the loop (MIL);
- Software in the loop (SIL);
- Processor in the loop (PIL);
- Hardware in the loop (HIL);
- Processor in the loop (PIL).

Mô phỏng PIL: Trong giai đoạn này, mã điều khiển sẽ được nhúng xuống bộ vi xử lý và chạy mô phỏng vòng

lặp kín với mô hình đối tượng thông qua môi trường mô phỏng. Kiểm tra PIL sẽ giúp xác minh liệu bộ xử lý có khả năng chạy mã điều khiển hay không.

Code C được sinh tự động trên Matlab Simulink khi ta sử dụng thư viện nhúng (embedded coder) và liên kết matlab với phần mềm chuyên dụng với STM32F407 Discovery [7] là STM32CubeMX và Keil MDK-ARM.. Code sẽ được nạp vào vi điều khiển để thay thế cho mô hình bộ điều khiển (controller model). Hiện tại thì bộ điều khiển là code nằm trên Vi điều khiển, còn đối tượng cần điều khiển nằm trên Matlab của máy tính. Máy tính và Vi điều khiển sẽ giao tiếp và trao đổi dữ liệu với nhau qua cáp USB (ST-link).



Hình 5. Vi điều khiển STM32F407G-DISC

Các bước để lập trình cho STM32F407G-DISC1

- Xây dựng mô hình trên simulink;
- Mô phỏng;
- Build code (tạo khối PIL);
- Nạp chương trình cho STM32F407G-DISC1 thông qua giao tiếp ST-Link;
- Chạy chương trình và debug.

Sử dụng mô hình tay máy VNR T1 đã xây dựng được, ta tiến hành mô phỏng (PIL) bài toán điều khiển tay máy theo các luật điều khiển trượt. Để phù hợp với môi trường mô phỏng, ta viết lại mô hình động lực học tay máy VNR-T1 theo dạng:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) = u \quad (36)$$

$$P\ddot{q} = M(q)^{-1} (-C(q, \dot{q})\dot{q} - g(q) + u)$$

Các tính toán trên và tính chất đối xứng của  $\dot{M} - 2C$  được thực hiện và xác nhận một cách tương trưng bằng cách sử dụng phần mềm Maple.

Mô hình Matlab ở chế độ PIL:

Các thông số tay máy, quỹ đạo mong muốn được đưa ra như sau:

$$q_{1d} = 2 \sin(2\pi t)$$

$$q_{2d} = \sin(2\pi t)$$

$$q_{3d} = -0.5 \sin(2\pi t)$$

$$q_{4d} = 0.8 \sin(2\pi t)$$

$$q_{5d} = 3 \sin(2\pi t)$$

Thông số của bộ điều khiển trượt cơ bản:

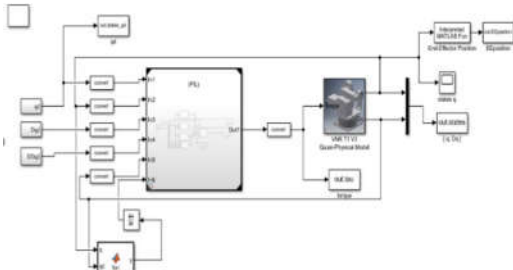
$$[K] = \begin{pmatrix} 100 & 0 & 0 & 0 & 0 \\ 0 & 100 & 0 & 0 & 0 \\ 0 & 0 & 50 & 0 & 0 \\ 0 & 0 & 0 & 20 & 0 \\ 0 & 0 & 0 & 0 & 20 \end{pmatrix}$$

$$[K] = \begin{pmatrix} 1000 & 0 & 0 & 0 & 0 \\ 0 & 2000 & 0 & 0 & 0 \\ 0 & 0 & 500 & 0 & 0 \\ 0 & 0 & 0 & 2000 & 0 \\ 0 & 0 & 0 & 0 & 1000 \end{pmatrix}$$

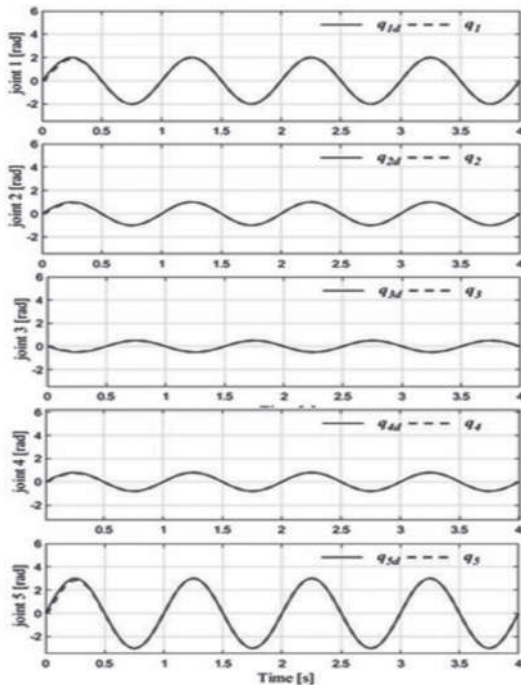
Thông số của bộ điều khiển trượt với luật tiếp cận theo hàm mũ:

$$N = \text{diag}(n) \text{ với } n = 0.6 + 0.6e^{(-20|s|)}$$

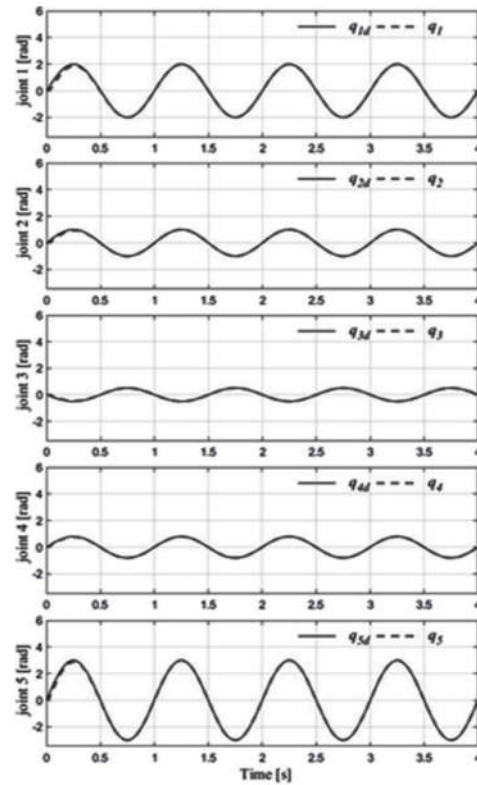
Với các thông số mô phỏng trình bày như ở trên, ta thu được kết quả mô phỏng của một số đại lượng ứng với hai trường hợp trượt cơ bản và trượt chống rung như dưới đây:



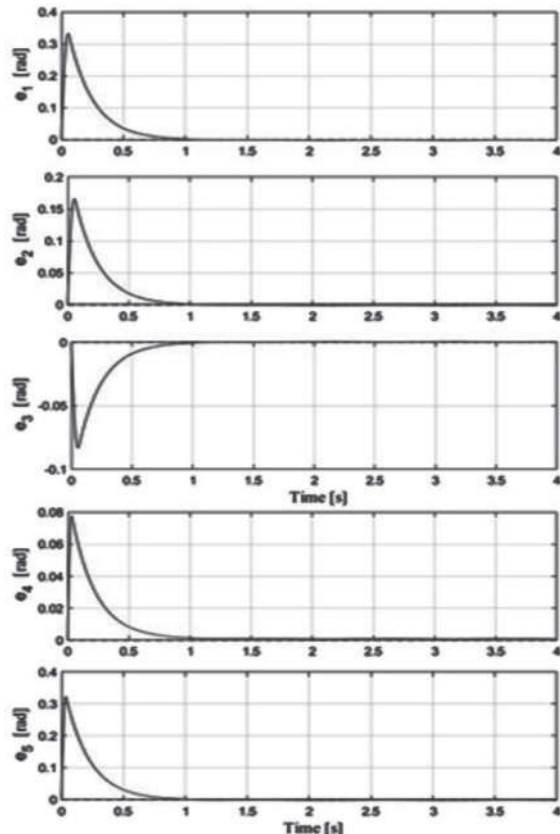
Hình 6. Mô hình PIL gắn với tay máy VNR-T1



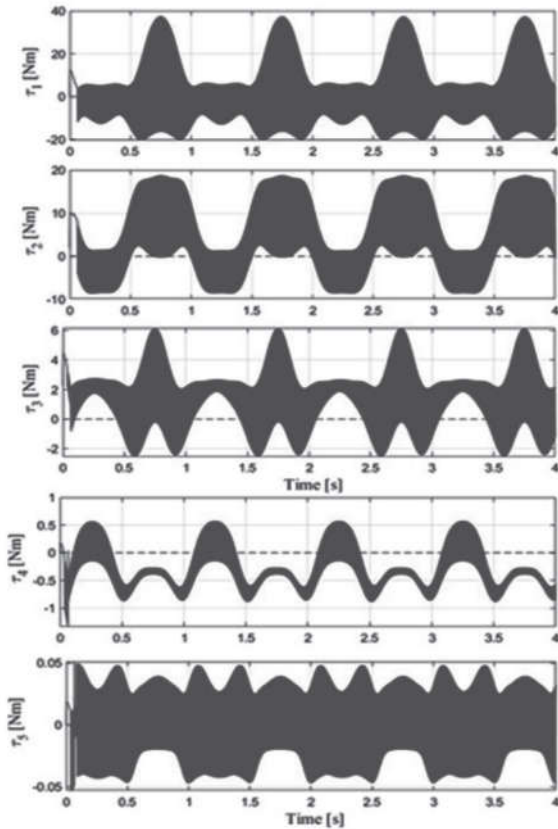
Hình 7. Góc quay của các khớp theo bộ điều khiển trượt cơ bản



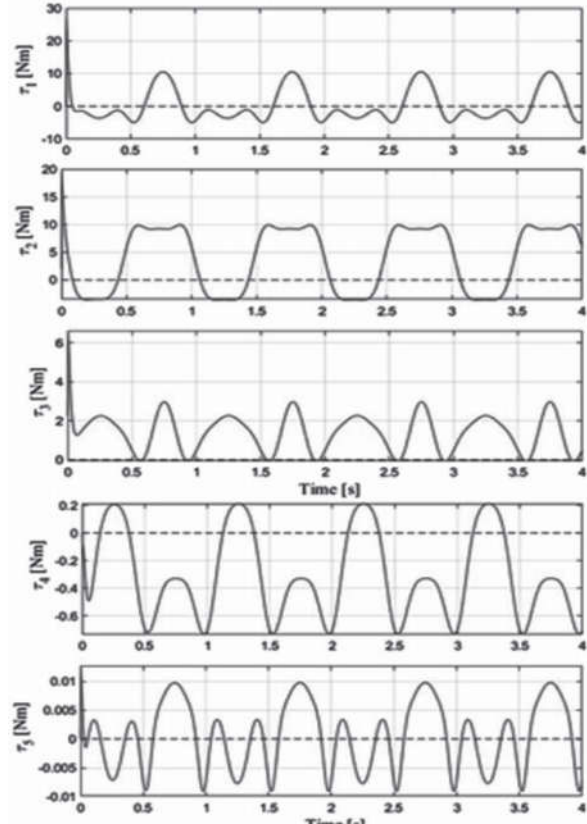
Hình 8. Góc quay của các khớp theo bộ điều khiển trượt chống rung



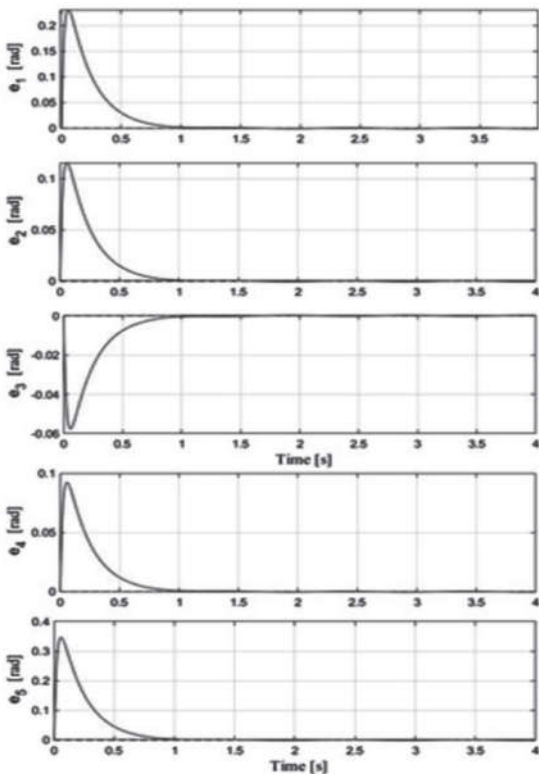
Hình 9. Sai lệch bám theo luật điều khiển trượt cơ bản



Hình 10. Sai lệch bám theo luật điều khiển trượt chống rung



Hình 12. Momen cung cấp cho các khớp theo bộ điều khiển trượt chống rung



Hình 11. Momen cung cấp cho các khớp theo bộ điều khiển trượt cơ bản

### Nhận xét

- Kết quả mô phỏng cho thấy bộ điều khiển trượt chống rung cho kết quả tối ưu hơn so với bộ điều khiển trượt cơ bản.
- Giao tiếp giữa vi điều khiển và matlab ổn định nhưng quá trình nạp bộ điều khiển qua giao tiếp ST-link còn chậm.
- Vi điều khiển có thể xử lý ổn định ở chế độ PIL nhưng để phát triển tiếp hệ thống đến HIL thì vẫn còn nhiều hạn chế về kết nối và các cổng giao tiếp I/O.

### 6. KẾT LUẬN

Các bộ điều khiển thiết kế được đều có khả năng điều khiển robot chính xác về mặt quỹ đạo. Tuy nhiên để điều khiển ổn định thì cần có bộ điều khiển trượt với luật tiếp cận hàm mũ hoặc các bộ điều khiển không gây rung, khử nhiễu tốt. Bộ điều khiển trượt chống rung cho kết quả tối ưu hơn so với bộ điều khiển trượt cơ bản. Vi điều khiển có thể xử lý ổn định ở chế độ PIL nhưng để phát triển tiếp hệ thống đến HIL thì vẫn còn nhiều hạn chế về kết nối và các cổng giao tiếp I/O.

**TÀI LIỆU THAM KHẢO**

- [1]. Le Ngoc Truc, Nguyen Tung Lam (2020), *Designing sliding mode control using exponential functions applied for the 5-DOF VNR T1 robo.*, Measurement, Control, and Automation, vol.1, pp. 1-7.
- [2]. M. Chen, S. S. Ge, and B. Ren (2011), *Adaptive tracking control of uncertain MIMO nonlinear systems with input constraints*, Automatica, vol. 47, no. 3, pp. 452-465.
- [3]. D. Won, W. Kim, D. Shin and a. C. C. Chung (2015), *High-Gain Disturbance Observer-Based Backstepping Control With Output Tracking Error Constraint for Electro-Hydraulic Systems*, IEEE Transactions on control systems technology, vol.23, pp. 787-795.
- [4]. T. Le Ngoc, Q. van Nguyen, and P. Q. Nguyen (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.
- [5]. Santana A, Martins L, Duarte R, Arantes G (2014), *Processor-in-the-Loop Simulations Applied to the Design and Evaluation of a Satellite Attitude Control*, Computational and Numerical Simulations.
- [6]. B. Q. Khánh, N. V. Liễn, P. Q. Hải and D. V. Nghi (2004), *Điều chỉnh tự động truyền động điện*, NXB Khoa học và Kỹ thuật, 418 tr.
- [7]. Replace STM32F4DISCOVERY, Discovery kit with STM32F407VG MCU\* New order code STM32F407G-DISC1.
- [8]. Nguyễn Mạnh Tiến (2007), *Điều khiển robot công nghiệp*, NXB Khoa học và Kỹ thuật, 270 trang.
- [9]. Nguyễn Phùng Quang (2002), *Truyền động điện thông minh*, NXB Khoa học và Kỹ thuật, 285 trang.
- [10]. By Lars Rosqvist, Roger Aarenstrup, and Kristian Lindqvist, MathWorks, Processor-In-the-Loop Simulation on Embedded Linux Boards (<https://www.mathworks.com>).

**THÔNG TIN TÁC GIẢ**



**Lê Ngọc Trúc**

- Tóm tắt quá trình đào tạo, nghiên cứu (thời điểm tốt nghiệp và chương trình đào tạo, nghiên cứu):
- + Năm 2006: Tốt nghiệp Đại học ngành Điều khiển tự động, Trường Đại học Bách khoa Hà Nội.
- + Năm 2010: Tốt nghiệp Thạc sĩ ngành Điều khiển tự động, Trường Đại học Bách khoa Hà Nội.
- Tóm tắt công việc hiện tại: Giảng viên khoa Cơ khí động lực, Trường Đại học Sư phạm Kỹ thuật Hưng Yên.
- Lĩnh vực quan tâm: Điều khiển tự động, robotics, mô hình hóa hệ thống, động lực học hệ nhiều vật.
- Email: lengoctruc@gmail.com.
- Điện thoại: 0983 247 911.



**Trần Văn Chi**

- Từ 2016-2021: Sinh viên Trường Đại học Bách Khoa Hà Nội, chuyên ngành tự động hóa công nghiệp.
- Tóm tắt công việc hiện tại: Sinh viên Trường Đại học Bách khoa Hà Nội.
- Lĩnh vực quan tâm: Tự động hóa, robot.
- Email: tranvanchi98@gmail.com.
- Điện thoại: 0845 551 854.





**Nguyễn Hữu Hải**

- Tóm tắt quá trình đào tạo, nghiên cứu (thời điểm tốt nghiệp và chương trình đào tạo, nghiên cứu):
- + Năm 2002: Tốt nghiệp Đại học ngành Tự động hóa, Trường Đại học Bách khoa Hà Nội.
- + Năm 2006: Tốt nghiệp Thạc sĩ ngành Tự động hóa, Trường Đại học Bách khoa Hà Nội.
- Tóm tắt công việc hiện tại: Giảng viên khoa Điện, Trường Đại học Công nghiệp Hà Nội.
- Lĩnh vực quan tâm: Tự động hoá.
- Email: nguyenuu hai@hau.edu.vn.
- Điện thoại: 0936658826.



**Nguyễn Danh Huy**

- Tóm tắt quá trình đào tạo, nghiên cứu (thời điểm tốt nghiệp và chương trình đào tạo, nghiên cứu):
- + Năm 1997: Tốt nghiệp Đại học ngành Tự động hoá, Trường Đại học Bách khoa Hà Nội.
- + Năm 2000: Tốt nghiệp Thạc sĩ ngành Tự động hóa Trường Đại học Bách khoa Hà Nội.
- + Năm 2019: Tốt nghiệp Tiến sĩ ngành Tự động hóa Trường Đại học Bách khoa Hà Nội.
- Tóm tắt công việc hiện tại: Giảng viên Viện Điện, Trường Đại học Bách khoa Hà Nội.
- Lĩnh vực quan tâm: Tự động hóa.
- Email: huy.nguyendanh@hust.edu.vn.
- Điện thoại: 0913 557 899.



**Nguyễn Trọng Các**

- Tóm tắt quá trình đào tạo, nghiên cứu (thời điểm tốt nghiệp và chương trình đào tạo, nghiên cứu):
- + Năm 2002: Tốt nghiệp Đại học ngành Điện, chuyên ngành Điện nông nghiệp, Trường Đại học Nông nghiệp I Hà Nội (nay là Học viện Nông nghiệp Việt Nam).
- + Năm 2005: Tốt nghiệp Thạc sĩ ngành Kỹ thuật tự động hóa, chuyên ngành Tự động hóa, Trường Đại học Bách khoa Hà Nội.
- + Năm 2015: Tốt nghiệp Tiến sĩ ngành Kỹ thuật điện tử, chuyên ngành Kỹ thuật điện tử, Trường Đại học Bách khoa Hà Nội.
- Tóm tắt công việc hiện tại: Giảng viên khoa Điện, Trường Đại học Sao Đỏ.
- Lĩnh vực quan tâm: DCS, SCADA, NCS.
- Email: cacdhsd@gmail.com.
- Điện thoại: 0904 369 421.



**Nguyễn Tùng Lâm**

- Tóm tắt quá trình đào tạo, nghiên cứu (thời điểm tốt nghiệp và chương trình đào tạo, nghiên cứu):
- + Năm 2014: Tốt nghiệp Tiến sĩ tại Đại học Tây Úc.
- Tóm tắt công việc hiện tại: Giảng viên Viện Điện, Trường Đại học Bách khoa Hà Nội.
- Lĩnh vực nghiên cứu: Tự động hóa.
- Email: lam.nguyentung@hust.edu.vn.