SLIDING MODE CONTROL TO TRACK TRAJECTORIES FOR TWO-LINKED ROBOT ARM

Huu-Huy Nguyen*, Van-Dong-Hai Nguyen, Trong-Nam Doan, Thanh-Truc Nguyen, Thi-Diem-Quynh Huynh, Dac-Huy Huynh, Dang-Khoa Vo, Trong-Hien Huynh

Ho Chi Minh city University of Technology and Education (HCMUTE)
01, Vo Van Ngan street, Thu Duc district, Ho Chi Minh city, Vietnam
***** Corresponding author. E-mail: 19124110@student.hcmute.edu.vn

Abstract: Two-linked robot arm is a popular model in both laboratory and actual industry. This is a MIMO system which has the same number of inputs and outputs. Because of the interactions between variables, normal SISO control algorithms, such as PID algorithm, do not prove their effectiveness in process of operation. In this paper, we propose using a method of sliding algorithm to control two-linked robot tracking expected trajectories. Results of controlling are proved to be successful through Matlab/Simulink.
Keywords: Two-linked robot arm, MIMO system, sliding control, SISO control, trajectories tracking control.

1. Introduction

The robotic arm is a highly applicable object, often used in laboratories as well as in industrial practice [1]. The PID algorithm controlled the above system [2]. However, in this MIMO system, the interaction between state variables is significant. Therefore, decomposing this MIMO object into the form of separate SISO systems and applying the PID to each separated approximated system is mathematically unstable. Studies which upgrade PID control also did not solve the problem of a mathematical guarantee for the system [3]. In [5], the authors present a sliding method to control this system. However, the survey of the trajectory of each joint has not been mentioned in above study; only the beginning and end of the impact have been investigated. In addition, the system response is not mentioned. In the study [6], the authors mentioned a new way to design the sliding surface. However, the above authors stopped at the control at the specified set value. Based on this paper, we present the extended development of the upper sliding control algorithm to successfully control the robot arm following the trajectory as well as stabilizing in the desired state.

The robotic arm system applied in this paper is a two-order system. Control results are examined on Matlab/ Simulink.

2. Dynamic Equations

According to [4], mathematical model of robot arm has the form shown in Figure 1, and the system of dynamic equations of the system has the form shown in expressions (1) and (2).

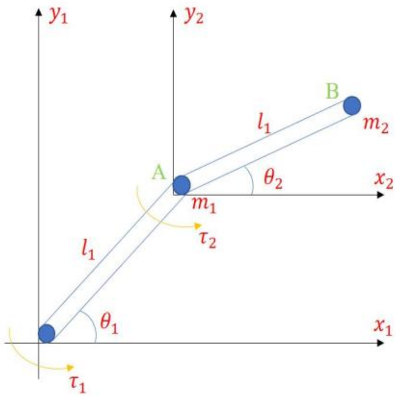

Fig. 1. Mathematical model of a two-linked robot arm.

$$\tau_1 = [(m_1+m_2)l_1^2 + m_2 l_2^2 + 2m_2 l_1 l_2 c_2]\ddot{\theta}_1 + [m_2 l_2^2 + m_2 l_1 l_2 c_2]\ddot{\theta}_2 + \\ -2m_2 l_1 l_2 s_2 \dot{\theta}_1 \dot{\theta}_2 - m_2 l_1 l_2 s_2 \dot{\theta}_2^2 + (m_1+m_2) g l_1 s_1 + m_2 g l_2 s_{12} \quad (1)$$

$$\tau_2 = (m_2 l_2^2 + m_2 l_1 l_2 c_2)\ddot{\theta}_1 + m_2 l_2^2 \ddot{\theta}_2 + m_2 l_1 l_2 s_2 \dot{\theta}_1^2 + m_2 g l_2 s_{12} \quad (2)$$

where, m_1 (kg), m_2 (kg) are the weights of links 1 and 2, respectively; l_1 (m) and l_2 (m) are the lengths of links 1 and 2, respectively; the angles θ_1 (rad) and θ_2 (rad) are angles between link 1 and 2 with horizontal direction, respectively; τ_1 (Nm) and τ_2 (Nm) are the torques acting on the arm, respectively, caused by the DC motor; $s_i=\sin\theta_i$ (i=1,2); $c_i=\cos\theta_i$ (i=1,2); $s_{12}=\sin(\theta_1+\theta_2)$

Equations (1) and (2) are calculated and converted to the form:

$$\dot{x} = F + Gu_1 + Hu_2 \tag{3}$$

Where, $x=\begin{bmatrix} x_1 & x_2 & x_3 & x_4 \end{bmatrix}^T=\begin{bmatrix} \theta_1 & \dot{\theta}_1 & \theta_2 & \dot{\theta}_2 \end{bmatrix}^T$; $u=\begin{bmatrix} \tau_1 & \tau_2 \end{bmatrix}^T$; $F=\begin{bmatrix} x_2 & F_1 & x_4 & F_2 \end{bmatrix}^T=\begin{bmatrix} x_2 & F_1(x) & x_4 & F_2(x) \end{bmatrix}^T$;
$G=\begin{bmatrix} x_2 & G_1 & x_4 & G_2 \end{bmatrix}^T=\begin{bmatrix} x_2 & G_1(x) & x_4 & G_2(x) \end{bmatrix}^T$; $H=\begin{bmatrix} x_2 & H_1 & x_4 & H_2 \end{bmatrix}^T=\begin{bmatrix} x_2 & H_1(x) & x_4 & H_2(x) \end{bmatrix}^T$

Through Matlab computation, we have:

$$g_1 = \frac{1}{l_1^2\left(m_1 + m_2 - m_2 * \cos x_3^2\right)} \tag{4}$$

$$h_1 = \frac{-(l_2 + l_1 \cos x_3)}{l_1^2 l_2\left(m_1 + m_2 - m_2 \cos x_3^2\right)} \tag{5}$$

$$f_1 = \frac{\begin{pmatrix} g m_1 \sin(x_1 + 2x_3) - g m_2 \sin x_1 - 2 g m_1 \sin x_1 + l_1 m_2 x_2^2 \sin 2x_3 \\ + 2 l_1 m_2 x_2^2 \sin x_3 + 2 l_2 m_2 x_4^2 \sin x_3 + 4 l_2 m_2 x_2 x_4 \sin x_3 \end{pmatrix}}{l_1\left(2m_1 + m_2 - m_2 \cos(2x_3)\right)} \tag{6}$$

$$h_2 = \frac{l_1^2 m_1 + l_1^2 m_2 + {l_2}^2 m_2 + 2 l_1 l_2 m_2 \cos x_3}{{l_1}^2 l_2 m_2\left(m_1 + m_2 - m_2 \cos {x_3}^2\right)} \tag{7}$$

$$g_2 = \frac{-(l_2 + l_1 \cos x_3)}{l_1^2 l_2 m_2\left(m_1 + m_2 - m_2 \cos x_3^2\right)} \tag{8}$$

$$f_2 = \frac{-\begin{pmatrix} l_1^2 m_1 x_2^2 \sin x_3 + l_1^2 m_2 x_2^2 \sin x_3 + l_2^2 m_2 x_2^2 \sin x_3 + {l_2}^2 m_2 x_4^2 \sin x_3 \\ - g l_2 m_1 \sin x_1 - g l_2 m_2 \sin x_1 + g l_1 m_1 \cos x_1 \sin x_3 + g l_1 m_2 \cos x_1 \sin x_3 \\ + 2 l_2^2 m_2 x_2 x_4 \sin x_3 + g l_2 m_2 \cos x_3^2 \sin x_1 + l_1 l_2 m_2 x_2^2 \sin 2x_3 \\ + \frac{l_1 l_2 m_2 x_4^2 \sin 2x_3}{2} + g l_2 m_2 \cos x_1 \cos x_3 \sin x_3 + l_1 l_2 m_2 x_2 x_4 \sin 2x_3 \end{pmatrix}}{l_1 l_2\left(m_1 + m_2 - m_2 \cos {x_3}^2\right)} \tag{9}$$

3. Sliding Control

The structure of the traction tracking controller is shown in

Fig. 2.

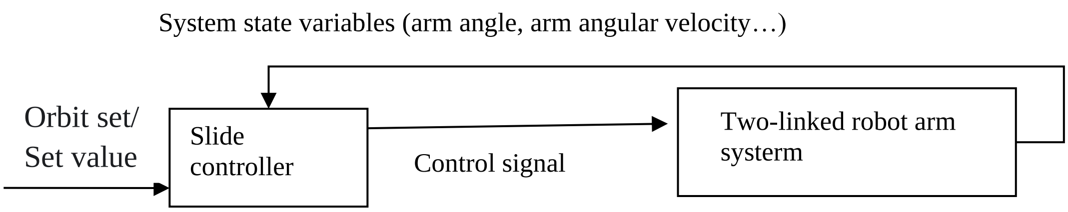

Fig. 2. Traction control structure of the sliding controller

We define the errors that need to be controlled to 0 as follows:

$$e_1 = x_1 - x_{1d};\ e_3 = x_3 - x_{3d} \tag{10}$$

where, x_{1d} , x_{3d} are the reference signals that link 1 and link 2 must track. In the case of tracking control, x_{1d} and x_{3d} are functions of time, respectively. In the case of a fixed-position control, x_{1d} and x_{3d} are constants. The control so that x_1 and x_3 follow x_{1d} and x_{3d} successfully, corresponding to the errors in (10) to 0, is to ensure that the arm system works to follow an expected trajectory or position.

Derivative of both sides of the expressions in (10), we obtain

$$\dot{e}_1 = \dot{x}_1 - \dot{x}_{1d} = x_2 - \dot{x}_{1d} \tag{11}$$

$$\dot{e}_3 = \dot{x}_3 - \dot{x}_{3d} = x_4 - \dot{x}_{3d} \tag{12}$$

The sliding surfaces are defined as follows:

$$S = \begin{bmatrix} S_1 \\ S_2 \end{bmatrix} = \begin{bmatrix} s_1 \\ s_3 \end{bmatrix} = \begin{bmatrix} c_1 e_1 + c_2 \dot{e}_1 \\ c_3 e_3 + c_4 \dot{e}_3 \end{bmatrix} \tag{13}$$

For values c_i (with i=1, 2, 3, 4) that are positive constants that need to be selected first. The control of e_1, e_3 to zero will be through the control.

Derivative of both sides of (13), we get:

$$\dot{S} = \begin{bmatrix} \dot{S}_1 \\ \dot{S}_2 \end{bmatrix} = \begin{bmatrix} \dot{s}_1 \\ \dot{s}_3 \end{bmatrix} = \begin{bmatrix} c_1 \dot{e}_1 + c_2 \ddot{e}_1 \\ c_3 \dot{e}_3 + c_4 \ddot{e}_3 \end{bmatrix} \tag{14}$$

We substitute (11), (12) and the corresponding derivative on both sides of (11), (12) into (4). Then, we substitute (3) in the result we just found. We obtain:

$$\dot{S} = \begin{bmatrix} \dot{S}_1 \\ \dot{S}_2 \end{bmatrix} = \begin{bmatrix} c_1 x_2 - c_1 \dot{x}_{1d} + \\ +c_2 \begin{bmatrix} f_1(x) + g_1(x) u_1 + \\ +h_1(x) u_2 \end{bmatrix} - c_2 \ddot{x}_{1d} \\ c_3 x_4 - c_3 \dot{x}_{3d} + \\ +c_4 \begin{bmatrix} f_2(x) + g_2(x) u_1 + \\ +h_2(x) u_2 \end{bmatrix} - c_4 \ddot{x}_{3d} \end{bmatrix} \tag{15}$$

We proceed to choose u_1, u_2 such that $\dot{S}_i$ has the opposite sign of S_i. The selection of the control signal is

$$\begin{bmatrix} \dot{S}_1 \\ \dot{S}_2 \end{bmatrix} = \begin{bmatrix} -\eta_1 sign(S_1) \\ -\eta_2 sign(S_2) \end{bmatrix} \tag{16}$$

With u_1, u_2 satisfying expression (16), we see:

- When the sliding surface $S_i > 0$, we have $\dot{S}_i < 0$. The sliding surface is pulled around the value 0
- When the sliding surface $S_i < 0$, we have $\dot{S}_i > 0$. The sliding surface is pulled around the value 0

Thus, the value of the sliding surface S_i will be made by the controller to oscillate only around the 0 point, but not too far away. From there, the system will be stable. However, due to the reciprocal oscillation, which has no decision component, the output will get closer and closer to 0, so chattering will occur in this control method

Identifying (16) and (15), we get:

$$\begin{aligned} & c_1 x_2 - c_1 \dot{x}_{1d} + \\ & +c_2 \begin{bmatrix} f_1(x) + g_1(x) u_1 + \\ +h_1(x) u_2 \end{bmatrix} - c_2 \ddot{x}_{1d} \\ & = -\eta_1 sign(S_1) \end{aligned} \tag{17}$$

$$\begin{aligned} & c_3 x_4 - c_3 \dot{x}_{3d} + \\ & +c_4 \begin{bmatrix} f_2(x) + g_2(x) u_1 + \\ +h_2(x) u_2 \end{bmatrix} - c_4 \ddot{x}_{3d} \\ & = -\eta_2 sign(S_2) \end{aligned} \tag{18}$$

Solving two equations (17), (18) according to two unknowns u_1 and u_2, we get the control signal by sliding method as:

$$u_1 = N_1(x, \dot{x}, x_d, \dot{x}_d, \ddot{x}_d) \tag{19}$$

$$u_1 = N_1(x, \dot{x}, x_d, \dot{x}_d, \ddot{x}_d) \tag{20}$$

where, $x_d = \begin{bmatrix} x_{1d} & x_{3d} \end{bmatrix}^T$ and x are defined in (3).

Applying Matlab calculation, the control signal in(19), (20) is calculated as follows:

```
u1 = (c1*c4*l1^2*m1*x1d_dot - c4*l1^2*m2*nuy1*sign(c1*x1 + c2*x2 - c1*x1d -
c2*x1d_dot) - c4*l2^2*m2*nuy1*sign(c1*x1 + c2*x2 - c1*x1d - c2*x1d_dot) -
```
(21)

```
c2*l2^2*m2*nuy2*sign(c3*x3 + c4*x4 - c3*x3d - c4*x3d_dot) - c1*c4*l1^2*m1*x2 -
c1*c4*l1^2*m2*x2 - c1*c4*l2^2*m2*x2 - c2*c3*l2^2*m2*x4 -
c4*l1^2*m1*nuy1*sign(c1*x1 + c2*x2 - c1*x1d - c2*x1d_dot) +
c1*c4*l1^2*m2*x1d_dot + c1*c4*l2^2*m2*x1d_dot + c2*c3*l2^2*m2*x3d_dot +
c2*c4*l1^2*m1*x1d_2dot + c2*c4*l1^2*m2*x1d_2dot + c2*c4*l2^2*m2*x1d_2dot +
c2*c4*l2^2*m2*x3d_2dot + c2*c4*g*l2*m2*sin(x1 + x3) + c2*c4*g*l1*m1*sin(x1) +
c2*c4*g*l1*m2*sin(x1) - 2*c4*l1*l2*m2*nuy1*sign(c1*x1 + c2*x2 - c1*x1d -
c2*x1d_dot)*cos(x3) - c2*l1*l2*m2*nuy2*sign(c3*x3 + c4*x4 - c3*x3d -
c4*x3d_dot)*cos(x3) - c2*c4*l1*l2*m2*x4^2*sin(x3) -
2*c1*c4*l1*l2*m2*x2*cos(x3) - c2*c3*l1*l2*m2*x4*cos(x3) +
2*c1*c4*l1*l2*m2*x1d_dot*cos(x3) + c2*c3*l1*l2*m2*x3d_dot*cos(x3) +
2*c2*c4*l1*l2*m2*x1d_2dot*cos(x3) + c2*c4*l1*l2*m2*x3d_2dot*cos(x3) -
2*c2*c4*l1*l2*m2*x2*x4*sin(x3))/(c2*c4)

u2 = (l2*m2*(c1*c4*l2*x1d_dot - c2*c3*l2*x4 - c1*c4*l2*x2 + c2*c3*l2*x3d_dot +
c2*c4*l2*x1d_2dot + c2*c4*l2*x3d_2dot + c2*c4*g*sin(x1 + x3) -
c4*l2*nuy1*sign(c1*x1 + c2*x2 - c1*x1d - c2*x1d_dot) - c2*l2*nuy2*sign(c3*x3 +
c4*x4 - c3*x3d - c4*x3d_dot) - c4*l1*nuy1*sign(c1*x1 + c2*x2 - c1*x1d -
c2*x1d_dot)*cos(x3) + c2*c4*l1*x2^2*sin(x3) - c1*c4*l1*x2*cos(x3) +
c1*c4*l1*x1d_dot*cos(x3) + c2*c4*l1*x1d_2dot*cos(x3)))/(c2*c4)
```
(22)

3. Simulation Results

3.1. Simulation Conditions

We select the initial parameters as follows:

```
m1=5; m2=2; l1=0.34; l2=0.34;
g=9.81;
```
(23)

Initial state values of the system:

```
x1_init=0.01; x2_init=0.02;
x3_init=0; x4_init=0;
```
(24)

The control parameters (from the sliding surfaces in (13) and the η values at (16) are selected as follows:

```
c_1=1;c_2=1;c_3=1;c_4=1;
```
(25)

3.2. Sine Trajectories

We choose sine trajectories for two arm. These trajectories are described as:

$$x_{1d} = 0.2\sin(0.3t)(rad) \quad (26)$$

$$x_{3d} = 0.3\sin(0.3t)(rad) \quad (27)$$

The control results are shown from Fig. 3 to Fig. 6 corresponding to η1=η2=1, from Figure 7 to Fig. 10 corresponding to η1=η2=0.1. We see that, with the basic sliding surface values as in (25) but not yet optimized, the sliding controller has successfully controlled the two-step machine arm. When the value of η1, η2 increases, the system responds faster (link 1 angle is established after 5s in Fig. 3, link 2 angle is established after 8s in Fig. 4. Comparison between angle of link 2 and reference trajectory (rad**)** compared to link 1 angle is established after 15s in Fig. 7, link 2 is established after 23 seconds in Fig. 8. However, with increasing values of η1, η2, chattering is also bigger with control signal fluctuations. The chattering in Fig. 5 and Fig. 6 is more severe than in Fig. 9 and Fig. 10.

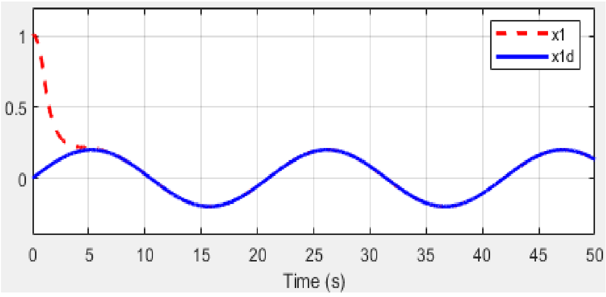

Fig. 3. Comparison between angle of link 1 and reference trajectory (rad)

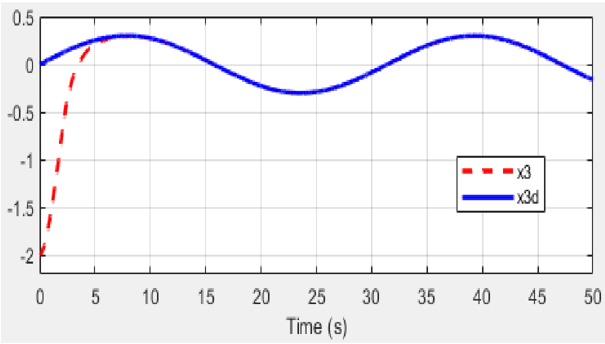

Fig. 4. Comparison between angle of link 2 and reference trajectory (rad)

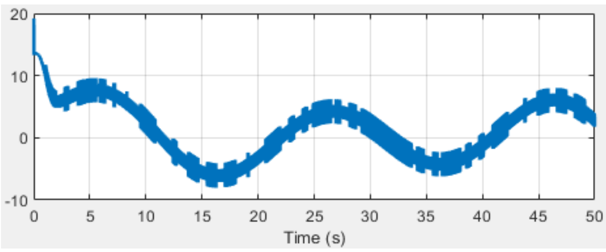

Fig. 5. Control signal u1 (Nm)

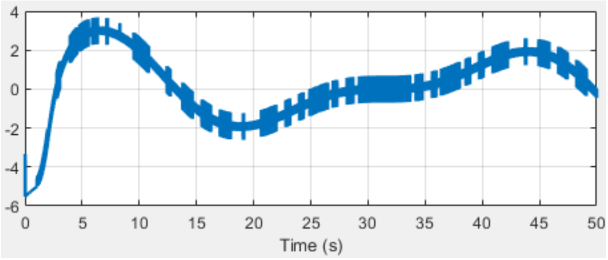

Fig. 6. Control signal u_2 (Nm)

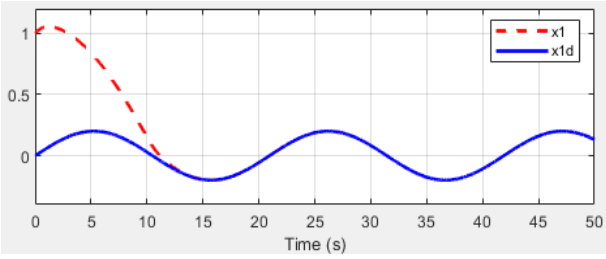

Fig. 7. Comparison between angle of link 1 and reference trajectory (rad)

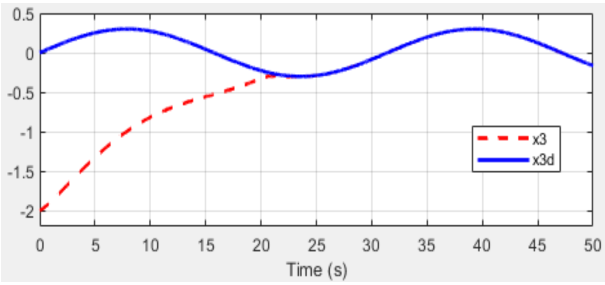

Fig. 8. Comparison between angle of link 2 and reference trajectory (rad)

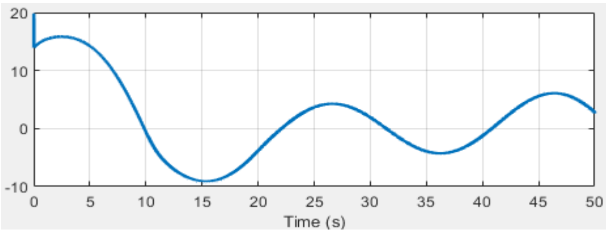

Fig. 9. Control signal u1(Nm)

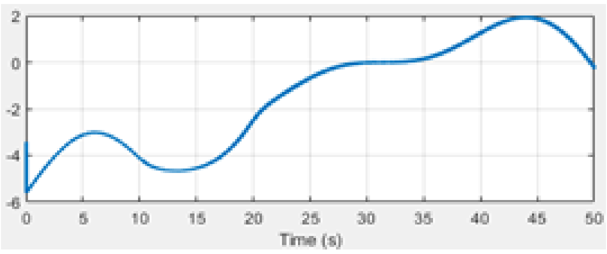

Fig. 10. Control signal u2 (Nm)

3.3. Stabilization at a Specified Location

With $\eta 1=\eta 2=1$, we choose $_{1d}=x_{3d}=0.2$ (rad), the simulation results are presented from Fig. 11 to Fig. 14. We see, with the stable case in place - the special case of tracking control. trajectory, the slide controller shows successful stabilization results. The establishment time of the two joints is 6s (Fig. 11 and Fig. 12). However, chattering still exists (Fig. 13 and Fig. 14). The solution to reducing chattering is to reduce the nu value. However, this will also correspondingly increase the setup time. The above results strengthen the conclusion and stability of the system at the working position determined by the sliding controller.

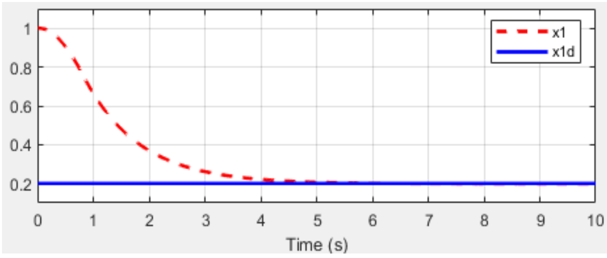

Fig. 11. Comparison between angle of link 1 and reference trajectory (rad)

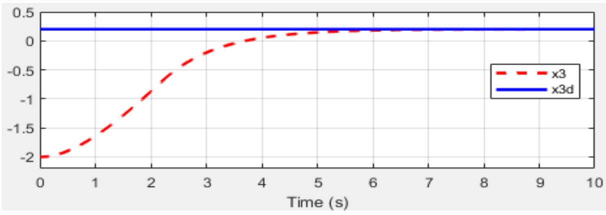

Fig. 12. Comparison between angle of link 2 and reference trajectory (rad)

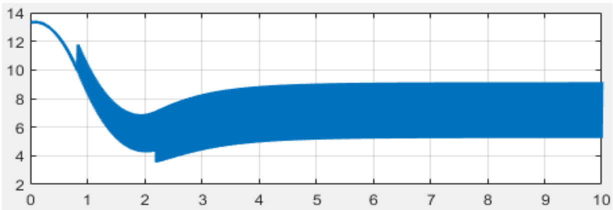

Fig. 13. Control signal u1 (Nm)

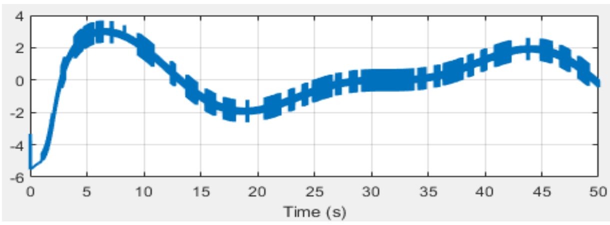

Fig. 14. Control signal u2 (Nm)

4. Conclusion

In the paper, the authors have extended the results from the article [6] to be able to apply to orbital tracking. The trajectory in this paper is a sinusoidal orbit. The verification of the nuy component to reduce chattering, as well as the disadvantages of making corrections to reduce chattering, were also re-examined in-situ stability control. Successful control results are proven through Matlab/Simulink testing.

5. References

[1] Radharamanan R., Jenkins H. E.: "Robot Applications in Laboratory-Learning Environment", 39th Southeastern Symposium on System Theory, pp. 80-84, 2007.

[2] Rocco P.: "Stability of PID control for industrial robot arms", in IEEE Transactions on Robotics and Automation, vol. 12, no. 4, pp. 606-614, 1996.

[3] Mohamed R.H., Bendary F., Elserafi K.: "Trajectory Tracking Control for Robot Manipulator using Fractional Order-Fuzzy-PID Controller", International Journal of Computer Applications (0975 – 8887), Volume 134, No.15, 2016.

[4] Okubanjo A., Martins O.: "Modelling of 2-DOF Robot Arm and Control", Futo Journal Series (FUTOJNLS) e-ISSN: 2476-8456 p-ISSN: 2467-8325 Volume-3, Issue-2, pp- 80 – 92, 2017.

[5] Hiếu P.Đ., Tuấn Đ.M., Duy L.N., Nghĩa L.V.: "Thiết kế bộ điều khiển trượt vị trí/ lực cánh tay robot tương tác với môi trường làm việc", Tạp chí Khoa học Công nghệ, p-ISSN: 1859-3585, e-ISSN: 2615-9619, số 55, trang 67-70, 2019.

[6] Duy L.T.: "Thiết kế bộ điều khiển trượt cho hệ tay máy robot", Tạp chí Khoa học và Công nghệ Đại học Đà Nẵng, số 4, 2003.