Acta Astronautica 176 (2020) 156–172

Contents lists available at ScienceDirect

Acta Astronautica

journal homepage: www.elsevier.com/locate/actaastro

Motion control of a space manipulator using fuzzy sliding mode control with reinforcement learning

Zhicheng Xie [a], Tao Sun [a], Trevor Kwan [b], Xiaofeng Wu [a,*]

[a] School of Aerospace, Mechanical and Mechatronic Engineering, The University of Sydney, NSW, 2006, Australia
[b] Department of Thermal Science and Energy Engineering, University of Science and Technology of China, Heifei, China

ARTICLE INFO

Keywords:
CubeSat
Fuzzy logic inference
Reinforcement learning
Sliding mode control
Space manipulator

ABSTRACT

The free-flying space manipulators present challenges in controlling the motions of both the spacecraft bus and the manipulator, because of the highly-coupling system dynamics and the unknown space environment disturbances. Although the sliding mode controllers are robust to the unknown disturbances and system uncertainties, the chattering effect could affect the pointing accuracy and the lifetime of the actuators. This paper first introduces the dynamics of a CuBot, which is a 3-rigid-link manipulator based on the CubeSat platform. To maintain the robustness while decreasing the chattering effect, an innovative reinforcement learning based fuzzy adaptive sliding mode controller is proposed. To maintain the robustness while reducing the chattering effect, an innovative reinforcement learning based fuzzy adaptive sliding mode controller is proposed. The switching gain is updated to estimate the lumped upper bound of the system uncertainties and the unknown disturbances, and then a new fuzzy logic adaptive law is applied on the switching gain to decrease the chattering effects. On top of that, the fuzzy logic rules are tuned by an innovative modified reinforcement learning mechanism to achieve the better tracking performance. The uniformly ultimately bounded tracking errors are guaranteed by the proposed control scheme, and the effectiveness is validated by the simulation results.

1. Introduction

The space manipulators have played an increasingly crucial role in various recent complex space missions [1]. The first remotely operated space manipulator was successfully demonstrated in 1986 [2]. There has been a growing popularity to develop the researches and projects. Since then a number of space manipulators have been developed. The ETS-VII was developed by JAXA for testing the technologies of rendezvous docking in 1997 [3]. The European Robotic Arm (ERA) was designed to attach to the Russian segment of the International Space Station (ISS) [4]. The humanoid robot ROBONAUT was proposed to replace the astronauts from high risk extravehicular activities [5]. However, for a space manipulator attached to the body of the target spacecraft, its workspace typically is constrained around a small area. To achieve more workspace the manipulator can be mounted on a small sized spacecraft that can free fly around the target spacecraft [6]. In this paper a CuBot is proposed, where an 8U CubeSat platform, 20 cm × 20 cm x 20 cm, is employed as the spacecraft bus. where The manipulator is mounted on the CubeSat as shown in Fig. 1. The motion of the CubeSat is controlled by thrusters when the manipulator is in operations.

Compared to terrestrial robot manipulators, the free-flying space manipulators have more complicated dynamics due to the external forces and reaction torques from the motion of the manipulator. The existing control strategies of handling the free flying space robots can be classified into 2 types. One type called pose-fixed mode is to treat the bus and the manipulator separately: firstly the position and attitude of the spacecraft bus are fixed using the thruster and reaction wheels to compensate any forces or torques resulted from the motions of the manipulator, and then applying the controllers designed for the fixed-base manipulators [7]. The other is to control the motions of both the spacecraft bus and the manipulator simultaneously by considering the dynamic coupling effects [8,9]. Fonseca et al. provided the approximate linear dynamic equations of a free-flying spacecraft equipped with a manipulator in the scenario of the close proximity phase of rendezvous docking operation, and then applied the LQR and PID controllers in both the approximate linear model and the non-linear model, demonstrating that the spacecraft's attitude can be stabilized by proper control parameters [10]. Liu et al. reformulated the dynamics equations of a space robot with different models of joint friction, and applied PD controller with the aid of state feedback decoupling method, which requires an accurate dynamics model, to achieve the exponential stability of the spacecraft attitude [11]. Yang et al. modelled the dynamics of a rigid

* Corresponding author.
E-mail address: xiaofeng.wu@sydney.edu.au (X. Wu).

https://doi.org/10.1016/j.actaastro.2020.06.028
Received 12 January 2019; Received in revised form 13 April 2020; Accepted 15 June 2020
Available online 18 June 2020

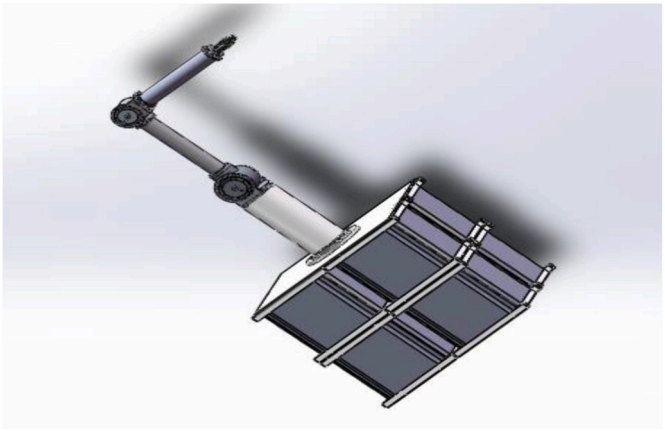

Fig. 1. The manipulator mounted on an 8U CubeSat.

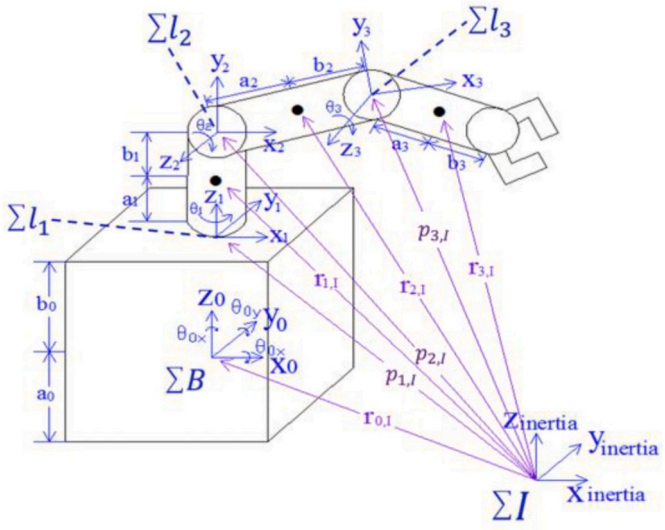

Fig. 2. The free-flying CubeSat with a mounted manipulator.

free-flying spacecraft equipped with the manipulator, consisting of the rigid links and flexible joints, by the assumed mode approach and the Lagrangian method, and then they applied a sliding mode controller (SMC) and a LQR controller to achieve the desired trajectory tracking [12]. However, most of these controllers require the known upper bound of the disturbances and system uncertainties that are hard to know in practice.

The unknown disturbances in space and the system uncertainties may result in the failure to control the space manipulators. The issue of the system uncertainties and disturbances can be solved by applying the adaptive algorithms such as neural networks to estimate the unknown disturbances and system uncertainties [13–16] or their upper bounds [17,18]. Yoo et al. proposed a dynamic surface based adaptive controller with the aid of the self-recurrent wavelet neural network (SRWNN) to control the 3-linkage-flexible-joint manipulator on the ground. The SRWNN was used to estimate the unknown system uncertainties and disturbances, and the control algorithm was verified by both the simulation results and mathematic stability analysis [19]. However, the performance of neural networks to approximate the system dynamics is highly dependent on the structure of neural network such as the number of hidden neurons and type of activation function. The methods of determining the optimal structure of the neural networks are still empirical [20]. The other method of handling unknown system uncertainties and disturbances is to apply the sliding mode controller with an upper bound of the unknown disturbances and uncertainties [21]. Ref [22]. modelled the dynamics equations of a free-flying spacecraft bus equipped with 2 manipulators, and then a robust sliding mode controller (SMC) requiring a known upper bound of system uncertainties and a model predictive controller (MPC) based on the nominal system parameters are applied in motion control. Shi et al. derived the dynamics model of a free-flying space robot considering with zero momentum control, and a robust sliding mode control (SMC) with the known upper bound of the system uncertainties and disturbances was proposed to stabilize both the attitude of the spacecraft and the motion of the manipulator [23]. The conventional sliding mode controllers invariably lead to the tradeoff between the robustness and the chattering effects that can decrease the tracking performance [24]. The big value of switching gain would result in the significant chattering effect while the small value would result in lower robustness.

This paper presents a new reinforcement learning based adaptive sliding mode controller (RLFSMC) for the CuBot motion control. The RLFSMC is capable of mitigating the chattering effects without sacrificing the robustness of the sliding mode controllers. It can achieve the uniformly ultimately bounded (UUB) tracking errors. The proposed method applies an innovative adaptive law governed by the fuzzy logic interference in SMC to provide adaptive switching gains, which can effectively reduce chattering effects without sacrificing the robustness. Unlike Ref [25] and Ref [26] where the fuzzy rules are given by the designer, a modified reinforcement learning mechanism is adopted to tune the fuzzy rules online to continuously search the optimal solutions in this paper. Moreover, unlike the conventional reinforcement Q learning mechanism in Ref. [27] where each pair of an action and a reward only lead to the update on 1 element in the Q matrix, the modified learning mechanism can utilize a pair of an action and a reward to update the Q values of multiple actions, showing more efficient learning.

The remaining paper is organized as follows. Section 2 presents the dynamics and kinematics of an 8U CubeSat equipped with a 3-rigid-link-age manipulator; In Section 3, we present the adaptive sliding mode controller with the aid of a fuzzy logic inference, and the reinforcement Q learning mechanism tuning the fuzzy rules is also detailed in this section; Section 4 presents simulation results and the comparisons between the proposed controller and a conventional sliding mode controller; Section 5 concludes.

2. Dynamics of the CuBot

The following assumptions for the presented model are imposed to maintain model simplicity:

Assumption1. The CubeSat body and manipulator links are rigid.

Assumption2. The effects of microgravity and orbital mechanics on the system dynamics are ignored, because they are insignificant compared to the control forces and torques during the typical maneuvers of short length for a free flying space manipulator.

The space manipulator system is constituted by an 8U CubeSat and a n-links. The structure of the space manipulator is shown in Fig. 2. $\sum B$ denotes the CubeSat (base) coordinate, $\sum I$ denotes the inertia coordinate. $\sum l_i$ $(i = 1, 2, 3)$ is the coordinate of the i^{th} link. $r_{i,I}$ $(i = 1, 2, 3)$ is the position vector of center of mass (CM) of the i^{th} link. $p_{i,I}$ $(i = 1, 2, 3)$ is the position vector of the i^{th} joint. a_i $(i = 1, 2, 3)$ is the position vector from the i^{th} joint to the CM of the i^{th} link. b_i $(i = 1, 2, 3)$ is the position vector from the CM of the i^{th} link to the $(i+1)^{th}$ joint.

The dynamics equation can be derived by the general Lagrangian formulation [28]:

$$\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{q}_i}\right) - \left(\frac{\partial T}{\partial q_i}\right) = Q_i, \quad i = 1, \ldots, N, \tag{1}$$

where T refers to the kinematic energy of the system. N means the degrees of freedom (DOF) of the system, which is equal to 6+ n consisting of the 6 variables for the spacecraft bus attitudes and positions as well as the n variables describing the angular movements of the n manipulator joints. Q_i and q_i are the i^{th} elements in the $(n+6) \times 1$ vector of generalized control forces/torques and the $(n+6) \times 1$ vector of generalized coordinates respectively.

$$q = [\delta_b, \theta_b, \theta_m]^T \tag{2}$$

$$Q = [F_b, \tau_b, \tau_m]^T \tag{3}$$

where $\delta_b = [x, y, z]^T$ are the positions of the CubeSat with respect to the inertia frame, and $\theta_b = [\theta_{0x}, \theta_{0y}, \theta_{0z}]^T$ are the angular positions of the CubeSat in the body frame respectively and $\theta_m = [\theta_1, \theta_2, \ldots, \theta_n]^T$ are the angular movements of the n manipulator joints. $F_b = [F_{bx}, F_{by}, F_{bz}]^T$ and $\tau_b = [\tau_{bx}, \tau_{by}, \tau_{bz}]^T$ are the control forces and torques applied on the CubeSat respectively. $\tau_m = [\tau_1, \tau_2, \ldots, \tau_n]^T$ are the control torques on the manipulator joints. T can be described in the discrete form by assuming the mass of each rigid body mainly concentrates on the mass center [29]:

$$T = \frac{1}{2}\sum_{i=0}^{n}\left(m_i v_{i,I}^T v_{i,I} + \omega_{i,I}^T I_i \omega_{i,I}\right) \tag{4}$$

where $v_{i,I}$, $\omega_{i,I}$, m_i, I_i, are the linear velocity vector, angular velocity vector, mass, and the inertial matrix of the i^{th} rigid link respectively. To describe the $\omega_{i,I}$ and $v_{i,I}$ in terms of q and $\dot{q}$, the following kinematics equations can be applied [30]:

$$\begin{bmatrix} v_{i,I} \\ \omega_{i,I} \end{bmatrix} = J_{b,i}\begin{bmatrix} v_0 \\ \omega_0 \end{bmatrix} + J_{m,i}\dot{\theta}_m \tag{5}$$

The v_0 and ω_0 are the vectors of the linear velocity and angular velocity of the CubeSat respectively. The $J_{b,i}$ is the matrix that maps the velocity vector of the CubeSat to the velocity vector of the i^{th} rigid body, and the $J_{m,i}$ is the Jacobian matrix that maps the joint vector of the manipulator to the velocity vector of the i^{th} rigid body [28,29].

$$J_{b,i} = \begin{bmatrix} E & -p_{0i}^{\times} \\ 0 & E \end{bmatrix} \in R^{6\times 6}, \quad p_{0i} = r_{i,I} - r_{0,I} \tag{6}$$

$$J_{m,i} = \begin{bmatrix} k_1 \times (r_{i,I} - p_{1,I}) & k_2 \times (r_{i,I} - p_{2,I}) \ldots k_i \times (r_{i,I} - p_{i,I}) \\ k_1 & k_2 \quad \ldots \quad k_i \end{bmatrix} \in R^{6\times i} \tag{7}$$

where the k_i is the unit vector along the rotational direction of the i^{th} joint of the manipulator. Parameters $r_{i,I}$ and $p_{i,I}$ are respectively the position vectors of the i^{th} joint and the mass center of the i^{th} rigid body with respect to the inertia frame. The E is the 3×3 identity matrix and the 0 is the 3×3 matrix in which all elements are zero. The $p_{0i}^{\times}$ is the 3×3 matrix determined by the 3×1 vector $p_{0i} = [p_{0i}^{(x)}, p_{0i}^{(y)}, p_{0i}^{(z)}]^T$, which results in the following equation:

$$p_{0i}^{\times} = \begin{bmatrix} 0 & -p_{0i}^{(z)} & p_{0i}^{(y)} \\ p_{0i}^{(z)} 0 - p_{0i}^{(x)} - p_{0i}^{(y)} & & \\ & p_{0i}^{(x)} & 0 \end{bmatrix} \tag{8}$$

Using the (5)–(8) by applying the general Lagrangian Formulation of Eq (1), the dynamic equations of the space manipulator can be obtained in the typical form [31,32]:

$$\begin{bmatrix} H_b & H_{bm} \\ H_{bm}^T & H_m \end{bmatrix}\begin{bmatrix} \ddot{x}_b \\ \ddot{\theta} \end{bmatrix} + \begin{bmatrix} C_b \\ C_m \end{bmatrix} = \begin{bmatrix} \sigma_b \\ \tau_m \end{bmatrix} + \begin{bmatrix} J_b^T \\ J_m^T \end{bmatrix}\sigma_e \tag{9}$$

where H_b, H_m and H_{bm} are the inertial matrix of the CubeSat, the robotic manipulator and the coupling dynamics between the CubeSat and the manipulator respectively. $x_b = [\delta_b, \theta_b]^T$ means the motions of the CubeSat. C_b and C_m are the velocity dependent nonlinear matrixes of the CubeSat and the manipulator respectively. $\sigma_b = [F_b, \tau_b]^T$ are the control forces and torques on the CubeSat. J_b and J_m are the Jacobian matrixes of the CubeSat and manipulator respectively, and σ_e is the unknown disturbance forces and torques. For the simple representation, Eq. (9) is rewritten as:

$$H(q)\ddot{q} + C(q, \dot{q}) = Q + D \tag{10}$$

where $H(q)$ is a $(n+6) \times (n+6)$ matrix and the $(n+6) \times 1$ vector $C(q, \dot{q})$ contains all the non-linear velocity term in a microgravity environment. The $(n+6) \times 1$ vector Q contains the control forces and torques on both the CubeSat and the manipulator, the $(n+6) \times 1$ vector D refers to the disturbances. For the convenience to design the controller, Eq. (10) can be rewritten to:

$$\ddot{q} = -H^{-1}C + H^{-1}D + H^{-1}Q \tag{11}$$

The system error and its derivative state vector are respectively defined as $e_1 = q - q_d$ and $e_2 = \dot{e}_1 = \dot{q} - \dot{q}_d$, where $q_d = [x_d, y_d, z_d, \theta_{0x,d}, \theta_{0y,d}, \theta_{0z,d}, \theta_{1,d}, \theta_{2,d}, \theta_{3,d}]^T$ is the vector containing the desired values of the CubeSat attitude and position as well as the joint angles of the mounted manipulator. Now Let the $A = -H^{-1}C$, $B = H^{-1}$, $\overline{D} = H^{-1}D$ and $u = Q$. As a result, the dynamic error is calculated as follows:

$$\dot{e}_1 = e_2 \tag{12}$$

$$\dot{e}_2 = A - \ddot{q}_d + Bu + \overline{D} \tag{13}$$

To handle the scenario in which the system parameters cannot be exactly known, we assume the $\Delta A = A - \widehat{A}$ and $\Delta B = B - \widehat{B}$ as the difference between the actual value and the nominal value of the system parameters, where $\widehat{A}$ and $\widehat{B}$ are exactly known. Accordingly, Eq. (13) can be rewritten as:

$$\dot{e}_2 = \widehat{A} + \widehat{B}u - \ddot{q}_d + \Delta A + \quad \Delta Bu + \overline{D} \tag{14}$$

Assumption 3. similar to Ref. [6,22], we assume the unknown bound of the system uncertainties and disturbances exists so that $\Gamma = (\Delta A + \Delta Bu + \overline{D})$, $|\Gamma_i| \le \Gamma_i^*$ for all $i = 1, 2, \ldots, N$. Where Γ_i is the i^{th} element of the $N \times 1$ vector Γ.

Definition 1. [33]: Consider the nonlinear dynamical system $\dot{x}(t) = f(x(t))$, $x(0) = x_0$. Uniformly ultimately bounded with ultimate bound B if there exist positive constants B and C, as well as $T = T(A, B)$ independent of $t_0 \ge 0$, for every A$\in$(0,C), such that $x(t_0) \le A \Rightarrow x(t) \le B$, $\forall\, t \ge t_0 + T$.

3. Control strategy

The controller is designed to control the attitude and position of the CubeSat and space manipulator's joints. The sliding mode controller (SMC) with the aid of the state feedback decoupling method is used, which can allow the independent selections of the switching gains for each system variable [22]. The error state of the system can move to the designed sliding manifold initially by the positive designed derivative of the switching gain. Once the error state enters into the described vicinity of the manifold, the fuzzy logic inference will govern the adaption law and adjust the switching gain in order to mitigate the chattering effects. Instead of setting up the exact fuzzy rules manually, the modified reinforcement Q learning mechanism is applied to search the optimal set of fuzzy rules from the given rule candidates in a fast way. The entire control scheme is shown in Fig. 3.

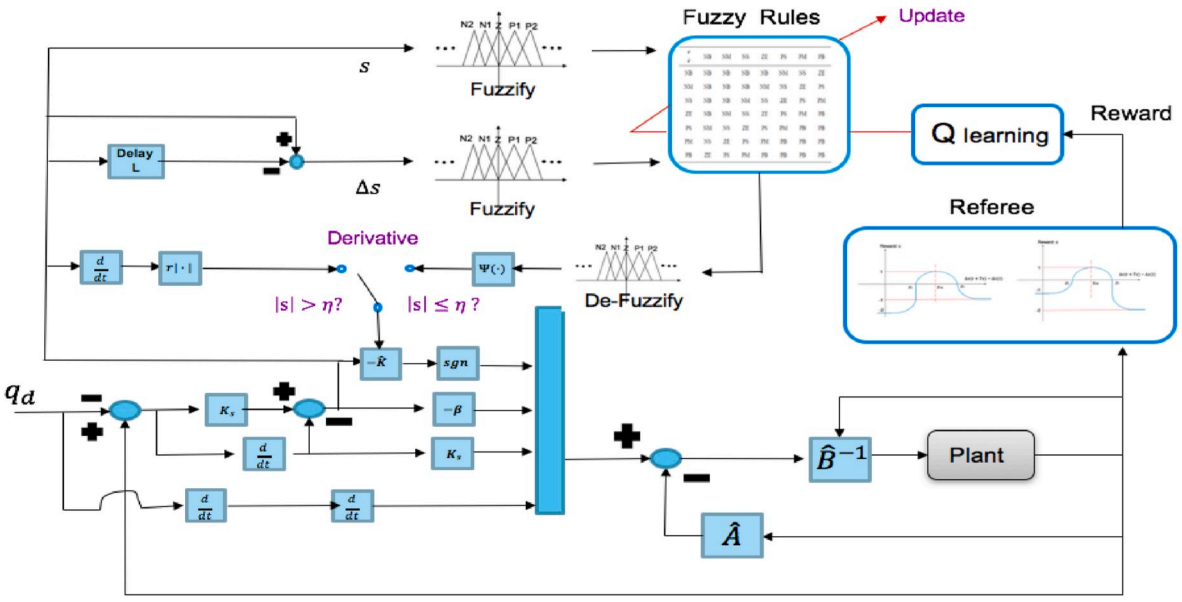

Fig. 3. The proposed RLFASMC scheme.

3.1. Sliding mode controller and fuzzy adaption law

The sliding variable is defined as:

$$s = \dot{e}_1 + K_s e_1 \tag{15}$$

where $K_s = diag([k_{s1}, k_{s1}, \ldots, k_{sN}])$ is the $(N \times N)$ positive constant diagonal matrix and $s = [s_1, s_1, \ldots, s_N]^T$ is the $(N \times 1)$ vector of sliding variables . From (12)-(15), the derivative of (15) can be written as:

$$\dot{s} = \widehat{A} + \widehat{B}u + K_s e_2 - \ddot{q}_d + \Delta A + \quad \Delta B u + \overline{D} \tag{16}$$

The control law is:

$$u = \widehat{B}^{-1}[-\widehat{A} - K_s e_2 + \ddot{q}_d - \beta s - \widehat{K} sgn(s)] \tag{17}$$

where the $(N \times N)$ diagonal matrixes $\beta = diag(\beta_1, \beta_2, \ldots, \beta_N)$ and $\widehat{K} = diag(\widehat{k}_1, \widehat{k}_2, \ldots, \widehat{k}_N)$ in which all the elements β_i and $\widehat{k}_1$ are positive to stabilize the system. The system can be decoupled by the u into multiple SISO system:

$$\dot{s}_i = \Gamma_i - \beta_i s_i - \widehat{k}_i \cdot sgn(s_i), \quad i = 1, 2, \ldots, N \tag{18}$$

The $(N \times 1)$ vector $sgn(s) = [sgn(s_1), sgn(s_2), \ldots, sign(s_N)]^T$ are defined by:

$$sgn(s_i) = \begin{cases} 1, & if\ s_i \geq 0 \\ -1, & if\ s_i < 0 \end{cases} \tag{19}$$

The adaption law of the switched switching gains $\widehat{K}$ are defined as:

$$\dot{\widehat{k}}_i = \begin{cases} \gamma_i \cdot |s_i|, & if\ |s|_i \geq \eta_i \\ \Psi_i, & else \end{cases} \tag{20}$$

η is a positive constant meaning the vicinity of the manifold. The γ_j is a positive constant. The Ψ_j is defined as:

$$\Psi_i = \begin{cases} \gamma_i \cdot |s_i|, & if\ \widehat{k}_i < \sigma_i \\ \gamma_i \cdot \delta_1^{h_i \rho_i} \cdot |s_i|, & if\ \rho_i \geq 0\ and\ \widehat{k}_i > \sigma_i \\ \delta_2 \cdot \rho_i \cdot |s_i|^{\rho_i}, & if\ \rho_i < 0\ and\ \widehat{k}_i > \sigma_i \end{cases} \tag{21}$$

where $\sigma_i < 0$ is the lower bound of $\widehat{k}_i$, and h_i is a positive constant to adjust the output. δ_1 is a positive number larger than 1, and δ_2 is a positive number smaller than 1. The ρ_i is the fuzzy output that is designed to be bounded $\rho_i \in [-\overline{\rho}_i, \rho_i^*]$. $-\overline{\rho}_i < 0$ and $\rho_i^* > 0$ are the minimum and maximum outputs of fuzzy logic inference, which can be set by the designer.

Theorem 1. *For a space manipulator mounted on a CubeSat* (10) *controlled by* (17)-(21) *with the Assumption* 1, *the sliding variables enter into their vicinities* ($|s|_i < \eta_i$) *of the sliding manifold within the finite time* t_η, *and then the sliding variables are guaranteed to be UUB*:

$$||s||_2 \leq \sqrt{\sum_{i=1}^{N} \left(\eta_i^2 + \delta_1^{h_i \rho_i^*} \cdot \max\left\{ V_{2,i}^*, \mathbb{V}_{2,i}^*, \cdot_{2,i}^* \right\}, \ t_\eta = \max\{t_{\eta_i}\}, \quad i = 1, 2, \ldots N \right.} \tag{22}$$

where $V_{2,i}^*$, $\mathbb{V}_{2,i}^*$ and $\upsilon_{2,i}^*$ are positive constants. $\eta_i > 0$ is the designed vicinity of sliding manifold. $\delta_1 > 1$ and $h_i > 0$ are parameters of controller decided by the designer. $\rho_i^* > 0$ is the maximum output of fuzzy logic inference decided by the designer. $\max\{\cdot\}$ means the maximum element.

Proof. *choose the candidate of Lyapnuov function as*:

$$V_{1,i}(t) = \frac{1}{2} s_i^2(t) + \frac{1}{2} \frac{\tilde{k}_i^2(t)}{\gamma_i}, \quad i = 1, 2, \ldots, N \tag{23}$$

where $\tilde{k}_i = \Gamma_i^* - \widehat{k}_i$; $\widehat{k}_i$ is the estimates of Γ_i^*. Differentiating (23) and use (16)-(21), resulting in:

$$\begin{aligned} \dot{V}_{1,i} &= s_i \dot{s}_i - \frac{\tilde{k}_i \dot{\widehat{k}}_i}{\gamma_i} = s_i[\Gamma_i - \beta_i s_i - \widehat{k}_i \cdot sgn(s_i)] - \frac{\tilde{k}_i \dot{\widehat{k}}_i}{\gamma_i} \leq -\beta_i s_i^2 + |s_i|(\Gamma_i^* - \widehat{k}_i) \\ &\quad - \frac{\tilde{k}_i \dot{\widehat{k}}_i}{\gamma_i} = -\beta_i s_i^2 + (\Gamma_i^* - \widehat{k}_i)\left(|s_i| - \frac{\dot{\widehat{k}}_i}{\gamma_i} \right) \\ &= -\beta_i s_i^2 \leq -\beta_i \eta_i^2 < 0 \left(\text{when } |s|_i \geq \eta_i > 0 \right) \end{aligned} \tag{24}$$

(24) implies that $V_{1,i}$ initially decreases and then enters into the vicinity of the sliding manifold ($|s|_i < \eta_i$) within a finite time [34]. After the sliding variables enter into the vicinity of the manifold, it is possible for the sliding variables to move out from the vicinity because the

Table 1
The derivative of the switching gain

The derivative of $\hat{k}_i$	Situation
$\dot{\hat{k}}_i = \gamma_i \cdot \lvert s_i \rvert$	when the sliding variables move out the vicinity
$\dot{\hat{k}}_i = \gamma_i \cdot \delta_1^{\rho_i} \cdot \lvert s_i \rvert$, $\rho_i \geq 0$	when the sliding variables stay inside the vicinity and the fuzzy output ρ_i is non-negative
$\dot{\hat{k}}_i = \delta_2 \cdot \rho_i \cdot \lvert s_j \rvert^{\rho_i}$, $\rho_i < 0$	when the sliding variables stay inside the vicinity and the fuzzy output is negative

derivative of $V_{1,i}$ is not guaranteed to be negative. If the sliding variables move out the vicinity ($|s|_i < \eta_i$), the $\dot{V}_{1,i}$ will become negative and consequently the sliding variables will go back towards the vicinity of manifold. As a result, the following inequality is obtained:

$$|s_i|^2 \leq V_{1,i} \leq \frac{1}{2}\eta_i^2 + \frac{1}{2}\frac{\tilde{k}_i^2}{\gamma_i}, \quad i = 1, 2, \ldots, N \tag{25}$$

Subsequently, we should prove the sliding variables are bounded after firstly entering into the vicinity ($|s|_i < \eta_i$), which can be achieved by proving the bound of $\tilde{k}_i^2$ exists so that $\tilde{k}_i^2 \leq \overline{K}_{M,i}$ for $i = 1, 2, \ldots, N$.

There are 3 situations of $\dot{\hat{k}}_i$ after the sliding variables firstly entering into the vicinity:

The following Lyapnuov function is chosen:

$$V_{2,i}(t) = \frac{1}{2}s_i^2(t) + \frac{\tilde{k}_i^2(t)}{2\gamma_i \cdot \delta_1^{h_i\rho_i^*}} \quad i = 1, 2, \ldots, N \tag{26}$$

The Proof of bounded $\tilde{k}_i^2$ can be achieved by proving that the $V_{2,i}$ is bounded after the first time of sliding variables entering into the vicinity of the manifold ($t > t_{\eta_i}$) for all $i = 1, 2, \ldots N$. Where $\rho_i^* > 0$ is the maximum fuzzy output, which will be used in the following proof. Differentiating (26) and using (16)-(21), resulting in:

$$\begin{aligned} \dot{V}_{2,i} &= s_i\dot{s}_i - \frac{\tilde{k}_i\dot{\hat{k}}_i}{\gamma_i \cdot \delta_1^{h_i\rho_i^*}} = s_i[\Gamma_i - \beta_i s_i - \hat{k}_i \cdot sgn(s_i)] - \frac{\tilde{k}_i\dot{\hat{k}}_i}{\gamma_i \cdot \delta_1^{h_i\rho_i^*}} \\ &\leq -\beta_i s_i^2 + (\Gamma_i^* - \hat{k}_i)\left(|s_i| - \frac{\dot{\hat{k}}_i}{\gamma_i \cdot \delta_1^{h_i\rho_i^*}}\right) \end{aligned} \tag{27}$$

To the 1st situation in Table 1, where $\dot{\hat{k}}_i = \gamma_i \cdot \delta_1^{h_i\rho_i} \cdot |s_i|$, $\rho_i \geq 0$, according to (27):

$$\dot{V}_{2,i} \leq -\beta_i s_i^2 + |s_i|(\Gamma_i^* - \hat{k}_i)\left(1 - \frac{\delta_1^{h_i\rho_i}}{\delta_1^{h_i\rho_i^*}}\right) = |s_i| \cdot \left[-\beta_i |s_i| + (\Gamma_i^* - \hat{k}_i)\left(1 - \frac{\delta_1^{h_i\rho_i}}{\delta_1^{h_i\rho_i^*}}\right)\right] \tag{28}$$

where the current fuzzy output $0 \leq \rho_i \leq \rho_i^*$, $\delta_1 > 1$, $\beta_i > 0$, $0 < \delta_1^{h_i\rho_i} < \delta_1^{h_i\rho_i^*}$ and $\hat{k}_i \geq \sigma_i$ and Γ_i^* is a constant. We assume a sufficiently large value $V_{2,i}^*$ of Lyapunov function $V_{2,i}$ (26) that requires at least one of the terms $\frac{1}{2}s_i^2$ and $\frac{\tilde{k}_i^2}{2\gamma_i \cdot \delta_1^{\rho_i^*}}$ should be sufficiently large. If the $\frac{1}{2}s_i^2$ is sufficiently large, the $|s_i|$ will be sufficiently large. $-\beta_i|s_i| + (\Gamma_i^* - \hat{k}_i)\left(1 - \frac{\delta_1^{h_i\rho_i}}{\delta_1^{h_i\rho_i^*}}\right) < -\beta_i|s_i| + (\Gamma_i^* + |\sigma_i|) < 0$ holds as long as $|s_i| > \frac{(\Gamma_i^* + |\sigma_i|)\left(1 - \frac{\delta_1^{h_i\rho_i}}{\delta_1^{h_i\rho_i^*}}\right)}{\beta_i} \geq \frac{\Gamma_i^* + |\sigma_i|}{\beta_i}$, which means $\dot{V}_{2,i} < 0$ resulted by a sufficiently large $|s_i| > \frac{\Gamma_i^* + |\sigma_i|}{\beta_i}$.

If the $\frac{\tilde{k}_i^2}{2\gamma_i \cdot \delta_1^{\rho_i^*}}$ is sufficiently large, the $\dot{V}_{2,i} \leq 0$ will be also achieved by considering the following optimization problem similar to Ref. [34]:

$$\max\left\{ \frac{1}{\gamma_i \cdot \delta_1^{h_i\rho_i^*}}(\Gamma_i^* - \hat{k}_i) \right\} \tag{29}$$

subject to:

$$\frac{1}{\gamma_i \cdot \delta_1^{h_i\rho_i^*}}(\Gamma_i^* - \hat{k}_i)^2 = R_i \leq V_{2,i}^*, \quad \hat{k}_i \geq \sigma_i, \quad \Gamma_i^* \geq 0 \tag{30}$$

where the R_i is the sufficiently large number smaller than or equal to $V_{2,i}^*$, the optimization problem results in a sufficiently large negative value of $(\Gamma_i^* - \hat{k}_i) < 0$ because $\Gamma_i^* > 0$ is a positive constant:

$$(\Gamma_i^* - \hat{k}_i) < 0 \tag{31a}$$

The semi-negative derivative $\dot{V}_{2,i} \leq 0$ can hold by (31a) because $1 - \frac{\delta_1^{h_i\rho_i}}{\delta_1^{h_i\rho_i^*}} \geq 0$ holds by $0 \leq \rho_i \leq \rho_i^*$, $\delta_1 > 1$ and $h_i > 0$, shown as the (31b).

$$\dot{V}_{2,i} \leq -\beta_i s_i^2 + |s_i|(\Gamma_i^* - \hat{k}_i)\left(1 - \frac{\delta_1^{h_i\rho_i}}{\delta_1^{h_i\rho_i^*}}\right) \leq 0 \tag{31b}$$

As a result, according to (28)-(31), either the sufficiently large $\frac{1}{2}s_i^2$ or $\frac{\tilde{k}_i^2}{2\gamma_i \cdot \delta_1^{h_i\rho_i^*}}$ can result in the semi-negative derivative of $V_{2,i}$, $\dot{V}_{2,i} \leq 0$, and consequently the $V_{2,i}$ is bounded by a sufficiently large $V_{2,i}^*$ when the $\dot{\hat{k}}_i = \gamma_i \cdot \delta_1^{h_i\rho_i} \cdot |s_i|$ and $0 \leq \rho_i < \rho_i^*$:

$$V_{2,i} \leq \frac{(\Gamma_i^* + |\sigma_i|)^2}{2\beta_i^2} + R_i \leq V_{2,i}^* \quad \left(\text{when } \dot{\hat{k}}_i = \gamma_i \cdot \delta_1^{h_i\rho_i} \cdot |s_i|\right) \tag{32}$$

To the 2nd situation in Table 1, where $\dot{\hat{k}}_i = \gamma_i|s_i|$, according to (27):

$$\dot{V}_{2,i} \leq -\beta_i s_i^2 + |s_i|(\Gamma_i^* - \hat{k}_i)\left(1 - \frac{1}{\delta_1^{h_i\rho_i^*}}\right) = |s_i| \cdot \left[-\beta_i|s_i| + (\Gamma_i^* - \hat{k}_i)\left(1 - \frac{1}{\delta_1^{h_i\rho_i^*}}\right)\right] \tag{33}$$

Similarly, we assume a sufficiently large value $\mathbb{V}_{2,i}^*$ of Lyapunov function $V_{2,i}$ (26) that also requires at least one of the 2 terms $\frac{1}{2}s_i^2$ and $\frac{\tilde{k}_i^2}{2\gamma_i \cdot \delta_1^{h_i\rho_i^*}}$ should be sufficiently large. If the $\frac{1}{2}s_i^2$ is sufficiently large, the $|s_i|$ will be sufficiently large, so that $-\beta_i|s_i| + (\Gamma_i^* - \hat{k}_i)\left(1 - \frac{1}{\delta_1^{h_i\rho_i^*}}\right) < 0$ holds as long as $|s_i| > \frac{(\Gamma_i^* + |\sigma_i|)\left(1 - \frac{1}{\delta_1^{h_i\rho_i^*}}\right)}{\beta_i} \geq \frac{\Gamma_i^* + |\sigma_i|}{\beta_i}$, which means $\dot{V}_{2,i} < 0$ resulted by a sufficiently large $|s_i| > \frac{\Gamma_i^* + |\sigma_i|}{\beta_i}$.

If the $\frac{\tilde{k}_i^2}{2\gamma_i \cdot \delta_1^{h_i\rho_i^*}}$ is sufficiently large, the $(\Gamma_i^* - \hat{k}_i) < 0$ will be achieved by considering the optimization problem same as (29)-(31).

$$\frac{1}{\gamma_i \cdot \delta_1^{h_i\rho_i^*}}(\Gamma_i^* - \hat{k}_i)^2 = \mathbb{R}_i \leq \mathbb{V}_{2,i}^*, \quad \hat{k}_i \geq \sigma_i, \quad \Gamma_i^* \geq 0 \tag{34}$$

where the $\mathbb{R}_i$ is the sufficiently large number smaller than or equal to $\mathbb{V}_{2,i}^*$, the optimization problem results in a sufficiently large negative value of $(\Gamma_i^* - \hat{k}_i) < 0$ because $\Gamma_i^* > 0$ is a positive constant. The semi-negative derivative $\dot{V}_{2,i} \leq 0$ can hold by (31a) because $1 - \frac{1}{\delta_1^{h_i\rho_i^*}} \geq 0$ holds by $\rho_i^* > 0$, $\delta_1 > 1$ and $h_i > 0$, shown as the (31b).

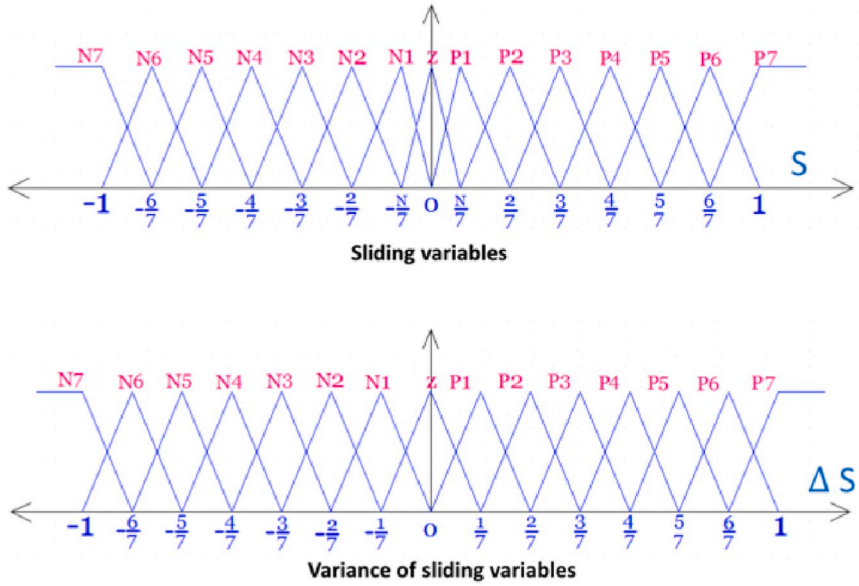

Fig. 4. Membership function of inputs: s and Δs.

$$\dot{V}_{2,i} \le -\beta_i s_i^2 + |s_i|\left(\Gamma_i^* - \widehat{k}_i\right)\left(1 - \frac{1}{\delta_1^{h_i\rho_i^*}}\right) \le 0 \tag{35}$$

As a result, according to (33)-(35), either sufficiently large $\frac{1}{2}s_i^2$ or $\frac{\tilde{k}_i^2}{2\gamma_i\cdot\delta_1^{h_i\rho_i^*}}$ results in the semi-negative derivative of $V_{2,i}$, $\dot{V}_{2,i} \le 0$, and the $V_{2,i}$ is bounded by a sufficiently large $\mathbb{V}_{2,i}^*$ when the $\dot{\widehat{k}}_i = \gamma_i\cdot|s_i|$:

$$V_{2,i} \le \frac{\left(\Gamma_i^* + |\sigma_i|\right)^2}{2\beta_i^2} + \mathbb{R}_i \le \mathbb{V}_{2,i}^* \ \left(\text{when } \dot{\widehat{k}}_i = \gamma_i\cdot|s_i|\right) \tag{36}$$

To the 3rd situation in Table 1, where $\dot{\widehat{k}}_i = \delta_2\cdot\rho_i\cdot|s_i|^{\rho_i}$, $\rho_i < 0$, according to (27):

$$\dot{V}_{2,i} \le -\beta_i|s_i|^2 + \left(\Gamma_i^* - \widehat{k}_i\right)\left(|s_i| + \frac{\delta_2\cdot|\rho_i|\cdot|s_i|^{\rho_i}}{\gamma_i\cdot\delta_1^{h_i\rho_i^*}}\right) \tag{37}$$

Similarly, we still assume a sufficiently large value $v_{2,i}^*$ of Lyapunov function $V_{2,i}$ (26) that also requires at least one of the 2 terms $\frac{1}{2}s_i^2$ and $\frac{\tilde{k}_i^2}{2\gamma_i\cdot\delta_1^{h_i\rho_i^*}}$ should be sufficiently large. If the $\frac{1}{2}s_i^2$ is sufficiently large, the $|s_i|$ will be sufficiently large so that $|s_i|>1$. The (37) can be further written as (38) because $|s_i|^{\rho_i} < 1$ holds by $\rho_i < 0$ and $|s_i| > 1$ now:

$$\dot{V}_{2,i} < -\beta_i|s_i|^2 + \left(\Gamma_i^* + |\sigma_i|\right)\left(|s_i| + \frac{\delta_2\cdot\rho_i^*}{\gamma_i\cdot\delta_1^{h_i\rho_i^*}}\right) < -\beta_i\left(|s_i| - \frac{\Gamma_i^* + |\sigma_i|}{2\beta_i}\right)^2 + \Lambda_i \tag{38}$$

where$\Lambda_i = \frac{\left(\Gamma_i^*+|\sigma_i|\right)^2}{4\beta_i^2} + \frac{\delta_2\cdot\rho_i^*}{\gamma_i\cdot\delta_1^{h_i\rho_i^*}}\left(\Gamma_i^* + |\sigma_i|\right)$ is a positive constant, and $\dot{V}_{2,i} < 0$ holds as long as the sufficiently large $|s_i| > \frac{\Gamma_i^*+|\sigma_i|}{2\beta_i} + \sqrt{\frac{\Lambda_i}{\beta_i}}$ according to (34), which means $\dot{V}_{2,i} < 0$ resulted by a sufficiently large $|s_i| > \max\left\{1, \frac{\Gamma_i^*+|\sigma_i|}{2\beta_i} + \sqrt{\frac{\Lambda_i}{\beta_i}}\right\}$.

If $\frac{\tilde{k}_i^2}{2\gamma_i\cdot\delta_1^{h_i\rho_i^*}}$ is sufficiently large, the $\left(\Gamma_i^* - \widehat{k}_i\right) < 0$ will be achieved by the optimization problem in (29)-(31).

$$\frac{1}{\gamma_i\cdot\delta_1^{h_i\rho_i^*}}\left(\Gamma_i^* - \widehat{k}_i\right)^2 = R_i \le \ V_{2,i}^*, \ \ \widehat{k}_i \ge \sigma_i, \ \ \Gamma_i^* \ge 0 \tag{39}$$

where the R_i is the sufficiently large number smaller than or equal to $V_{2,i}^*$, the optimization problem results in a sufficiently large negative value of $\left(\Gamma_i^* - \widehat{k}_i\right) < 0$ because $\Gamma_i^* > 0$ is a positive constant. The semi-negative derivative $\dot{V}_{2,i} \le 0$ can hold by (31a) because $|s_i| + \frac{\delta_2\cdot|\rho_i|\cdot|s_i|^{\rho_i}}{\gamma_i\cdot\delta_1^{h_i\rho_i^*}} \ge 0$ holds, shown as the (40).

$$\dot{V}_{2,i} \le -\beta_i s_i^2 + |s_i|\left(\Gamma_i^* - \widehat{k}_i\right)\left(|s_i| + \frac{\delta_2\cdot|\rho_i|\cdot|s_i|^{\rho_i}}{\gamma_i\cdot\delta_1^{h_i\rho_i^*}}\right) \le 0 \tag{40}$$

As a result, either sufficiently large $\frac{1}{2}s_i^2$ or $\frac{\tilde{k}_i^2}{2\gamma_i\cdot\delta_1^{h_i\rho_i^*}}$ results in the semi-negative derivative of $V_{2,i}$, $\dot{V}_{2,i} \le 0$, and then the $V_{2,i}$ is bounded by a sufficiently large value $v_{2,i}^*$ when the $\dot{\widehat{k}}_i = \delta_2\cdot\rho_i\cdot|s_i|^{\rho_i}$, $\rho_i < 0$.

$$V_{2,i} \le V_{2,i} \le \frac{1}{2}\Pi_i^2 + r_i \le v_{2,i}^* \ \left(\text{when } \dot{\widehat{k}}_i = \delta_2\cdot\rho_i\cdot\ s_i|^{\rho_i}\right) \tag{41}$$

where $\Pi_i = \max\left\{1, \frac{\Gamma_i^*+|\sigma_i|}{2\beta_i} + \sqrt{\frac{\Lambda_i}{\beta_i}}\right\}$.

Consequently, according to (32), (36) and (41), the $\dot{V}_{2,i} \le 0$ holds by the sufficiently large value $\max\{V_{2,i}^*, \mathbb{V}_{2,i}^*, v_{2,i}^*\}$, for all $i = 1, 2, \ldots N$ under all 3 sorts of adaptive law. In other words, the $V_{2,i}$ cannot exceed the sufficiently large value $\max\{V_{2,i}^*, \mathbb{V}_{2,i}^*, v_{2,i}^*\}$, which means all terms of $V_{2,i}$ including $\tilde{k}_i$ are bounded.

$$\frac{\tilde{k}_i^2}{2\gamma_i\cdot\delta_1^{h_i\rho_i^*}} \le V_{2,i} \le \max\left\{V_{2,i}^*, \mathbb{V}_{2,i}^*, v_{2,i}^*\right\} \quad i = 1, 2, \ldots, N \tag{42}$$

$$\tilde{k}_i^2 \le \overline{K}_{m,i}, \ \ \overline{K}_{m,i} = 2\gamma_i\cdot\delta_1^{h_i\rho_i^*}\cdot\max\left\{V_{2,i}^*, \mathbb{V}_{2,i}^*, v_{2,i}^*\right\} \tag{43}$$

According to (24), (25) and (43), the sliding variables are also bounded:

Table 2
Fuzzy rules.

$L(\Delta s_i)/L(s_i)$	$N7$	$N6$	…	$P7$
$N7$	$R_1(N7,N7)$	$R_2(N7,N6)$	…	$R_{15}(N7,P7)$
$N6$	$R_8(N6,N7)$	$R_9(N6,N6)$	…	$R_{30}(N6,P7)$
⋮	⋮	⋮	⋱	⋮
$P6$	$R_{196}(P6,N7)$	$R_{197}(P6,N6)$	…	$R_{210}(P6,P7)$
$P7$	$R_{211}(P7,N7)$	$R_{212}(P7,N6)$	…	$R_{225}(P7,P7)$

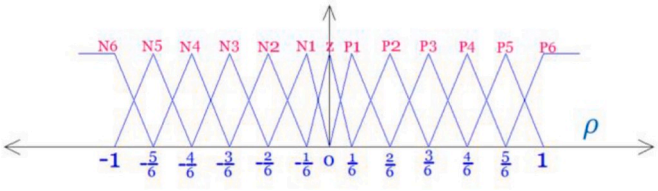

Fig. 5. Membership function of output ρ_i.

$$\left|s_i\left(t\right)\right|^2 \le 2V_{1,i}\left(t\right) \le \eta_i^2 + \delta_1^{h_i\rho_i^*}\cdot\max\left\{V_{2,i}^*, \mathbb{V}_{2,i}^*, v_{2,i}^*\right\},\quad t > t_{\eta_i}, i = 1,2,\ldots,N \tag{44a}$$

We now consider all subsystems shown as (18), the bounded sliding variables of system can be achieved according to (43) and (44a)

$$||s(t)|| = \sqrt{\sum_{i=1}^{N} s_i^2\left(t\right)} \le \sqrt{\sum_{i=1}^{N}\left(\eta_i^2 + \delta_1^{h_i\rho_i^*}\cdot\max\left\{V_{2,i}^*, \mathbb{V}_{2,i}^*, v_{2,i}^*\right\}\right)},\quad t > \max\{t_{\eta_i}\},\quad i = 1,2\ldots N \tag{44b}$$

Proof complete

3.2. Fuzzy logic inference

The bigger value of sliding switching gain $\widehat{k}_i$ can result in the more robust controller in that the negative derivative of s_i^2 can be achieved as long as $\widehat{k}_i > \Gamma_i$ according to Eq. (18). However, the bigger $\widehat{k}_i$ will bring significant chattering effects. As a result, this paper applies a fuzzy logic inference to adjust the derivative of $\widehat{k}_i$ to mitigate the chattering effects.

$$\dot{\widehat{k}}_i = \begin{cases} \gamma_i\cdot\delta_1^{h_i\rho_i}\cdot|s_i| \ , \ if\ \rho_i \ge 0\ and\ \widehat{k}_i > \sigma_i \\ \delta_2\cdot\rho_i\cdot|\ s_i|^{\rho_i} \ , \ if\ \rho_i < 0\ and\ \widehat{k}_i > \sigma_i \end{cases} \tag{45}$$

According to (45), it is clear that the smaller ρ_i (eg. $\rho_i = -1$) can result in the smaller switching gain to alleviate the chattering effects even if the sliding variables are near the manifold because of its negative exponent form, while the bigger ρ_i (eg. $\rho_i = 1$) can also result in the bigger switching gain to avoid the variables moving away from the manifold. The ρ_i is the fuzzy output from the fuzzy logic inference taking the sliding variables $s_i(t)$ and their variance $\Delta s_i(t) = s_i(t) - s_i(t - T_s)$ during a sampling period T_s as the inputs.

$$\rho_i(t) = FI[s_i(t),\ \Delta s_i(t)] \tag{46}$$

where $FI(\cdot,\cdot)$ is the aggregated fuzzy output based on the rules fired by the input: $s_i(t)$ and $\Delta s_i(t)$. The step of fuzzification is to map the range of numerical values of the inputs to the linguistic variables, triangular type input and output membership functions shown in Fig. 4 are used with the linguistic fuzzy sets:

$$Lin(s_i) \in \{N7, N6, N5, N4, N3, N2, N1, Z, P1, P2, P3, P4, P5, P6, P7\} \tag{47}$$

$$Lin(\Delta s_i) \in \{N7, N6, N5, N4, N3, N2, N1, Z, P1, P2, P3, P4, P5, P6, P7\} \tag{48}$$

$$Lin(\rho_i) \in \{N6, N5, N4, N3, N2, N1, Z, P1, P2, P3, P4, P5, P6\} \tag{49}$$

where N means negative, P means positive, Z means zero and the following numbers indicate the intensity (eg. $N7$ means the very negative big). The associated fuzzy domains, where the universal of discourse (UoD) of the fuzzy values of s_i and Δs_i as well as ρ_i are normalized from -1 to 1, are shown in Eqs. (43) and (44). As a result, the nonfuzzy values of s_i and Δs_i as well as ρ_i should be scaled to fit the UoD of the fuzzified variables with the scaling factors $K_{fd}(s_i)$, $K_{fd}(\Delta s_i)$ and $K_{fd}(\rho_i)$ that can also be regarded as the defined ranges of s_i, Δs_i and ρ_i.

$$fd(s_i) = \left\{-1, -\frac{6}{7}, -\frac{5}{7}, -\frac{4}{7}, -\frac{3}{7}, -\frac{2}{7}, -\frac{\aleph_i}{7}, 0, \frac{\aleph_i}{7}, \frac{2}{7}, \frac{3}{7}, \frac{4}{7}, \frac{5}{7}, \frac{6}{7}, 1\right\} \tag{50}$$

$$fd(\Delta s_i) = \left\{-1, -\frac{6}{7}, -\frac{5}{7}, -\frac{4}{7}, -\frac{3}{7}, -\frac{2}{7}, -\frac{1}{7}, 0, \frac{1}{7}, \frac{2}{7}, \frac{3}{7}, \frac{4}{7}, \frac{5}{7}, \frac{6}{7}, 1\right\} \tag{51}$$

$$fd(\rho_i) = \left\{-1, -\frac{5}{6}, -\frac{4}{6}, -\frac{3}{6}, -\frac{2}{6}, -\frac{1}{6}, 0, \frac{1}{6}, \frac{2}{6}, \frac{3}{6}, \frac{4}{6}, \frac{5}{6}, 1\right\} \tag{52}$$

where $\aleph_i$ is the positive constant that is important to the following reinforcement learning and will be detailed later. The mostly used way of acquiring the fuzzy rules is to analyze the behavior of the controlled system, where the rules are derived in the way that the desired state can be achieved [35]. However, the exact rules are hard to acquire because the perfect expertise of designer's knowledge is expensive in the practice. As a result, instead of manually setting up the fuzzy rules based on the expertise of designers, we initially set up the membership functions and the scaling factors as well as the group of rule candidates, and then applies the reinforcement Q learning to search the optimal rules (shown in Table 2) from the rule candidates.

The firing strengths of each fuzzy rule in Table 2 can be calculated by using the given membership functions:

$$\begin{aligned} &\mu_i^k[R_k(n,m)] = \min[\mu_n(s_i), \mu_m(\Delta s_i)], \quad n = -7,-6,\ldots,6,7 \\ &\qquad\qquad\qquad\qquad\qquad\qquad\quad m = -7,-6,\ldots,6,7 \\ &\qquad\qquad\qquad\qquad\qquad\qquad\quad i = 1,2,\ldots,N \\ &\qquad\qquad\qquad\qquad\qquad\qquad\quad k = 1,2,\ldots,15\times 15 \end{aligned} \tag{53}$$

where the operator $\min(\cdot)$ selects the minimum value among the membership $\mu_n(s_i)$ of s_i and membership $\mu_m(\Delta s_i)$ of Δs_i. After calculating the firing strength of each rule, the numerical output can be achieved [36]:

$$\rho_i = FI(s_i,\ \Delta s_i) = \frac{\sum_{k=1}^{km}\mu_i^k\Phi_i^k K_{fd}(\rho_i)}{\sum_{k=1}^{km}\mu_i^k},\quad i = 1,2,..,\ N \tag{54}$$

where Φ_i^k is the fuzzy values corresponding to the linguistic value of the k^{th} rule in the i^{th} subsystem, shown in Fig. 5, corresponding to the i^{th} subsystem shown in (18). km is the number of rules.

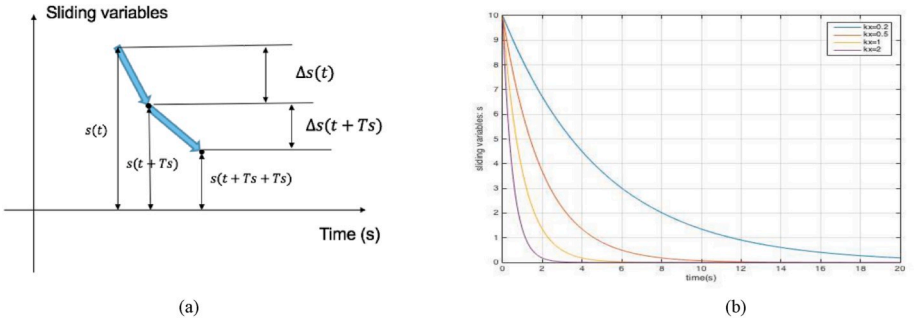

Fig. 6. The sliding variables and their variance (a) and the chattering-free dynamics $\dot{s} = -k_x s$ (b).

3.3. Modified fuzzy reinforcement Q learning

Q learning has the ability to optimize the unknown system by modelling the system into the Markov decision process, where an agent (the fuzzy logic inference in this paper) applies an action $u \in U(w)$ at time instant t on the state $w \in W$ to make the system move to the next state $w' \in W$ with a resulted reward/penalty c. The agent is supposed to find the policy $\pi(w) \rightarrow u$ that can generate the actions, which result inthe Q-value of either the biggest reward or the lowest penalty:

$$Q(w,u) = c(w,u) + \lambda \sum_{(w' \in W)} [p(w,w',u) \cdot Q^*(w',u)] \tag{55}$$

where $p(w,w',u)$ is the possibility of moving from w to w', Q^* is the maximum Q value. The update law of Q value can be generated from (55):

$$Q(w^t,u^t) \leftarrow Q(w^t,u^t) + \alpha\left[c(w^t,u^t) + \lambda \cdot Q^*\left(w^{t+1},u^t\right) - Q(w,u)\right] \tag{56}$$

$$Q^*\left(w^{t+1},u^t\right) \triangleq \arg\max_{u \in U(w^{t+1})} \left[Q\left(w^{t+1},u^t\right)\right] \tag{57}$$

where w^t and w^{t+1} refer to the state on t^{th} and $(t+1)^{th}$ instant, and u^t is the action taken on the t^{th} instant. $\alpha \in [0,1]$ is the learning rate and $\lambda \in [0,1]$ is the discount factor. The optimal Q-values can be achieved by visiting each pair of state and action [27].

In this paper, the Q learning will be used to determine the Q values for each pair of visited state (the linguistic variables of s_i and Δs_i) and the taken action (the fuzzy rules taken).

$$w_i(t) \triangleq (L[s_i(t)], L[\Delta s_i(t)]) \ , \ i = 1,2,..,N \tag{58}$$

$$u_i(t) \triangleq R_k(L[s_i(t)],\ L[\Delta s_i(t)]) = L[\rho_i(t)] \ , \ i = 1,2,\ldots,N \tag{59}$$

For each state $w_i(t)$ shown in Tables 2 and 3 different groups of rule candidates are given, they are the sliding variables approximating the manifold, drifting away the manifold and keeping static around the manifold.

R_k: If $L[s_i(t)] \in \{P7,P6,\ldots,P1,Z\}$ and $L[\Delta s_i(t)] \in \{N7,N6,\ldots,N1\}$, or if $L[s_i(t)] \in \{N7,N6,\ldots,N1,Z\}$ and $L[\Delta s_i(t)] \in \{P7,P6,\ldots,P1\}$, then $U_k^i = \{N6,N5,\ldots,N1,Z\}$ and:

$$u_k^i(t) = u_{k,1}^i(t) = N6 \text{ with } q\left(u_{k,1}^i,k\right)$$

$$u_k^i(t) = u_{k,2}^i(t) = N5 \text{ with } q\left(u_{k,2}^i,k\right)$$

$$\vdots$$

$$u_k^i(t) = u_{k,7}^i(t) = Z \text{ with } q\left(u_{k,7}^i,k\right)$$

R_k: If $L[s_i(t)] \in \{P7,P6,\ldots,P1\}$ and $L[\Delta s_i(t)] \in \{P7,P6,\ldots,P1,Z\}$, or if $L[s_i(t)] \in \{N7,N6,\ldots,N2,N1\}$ and $L[\Delta s_i(t)] \in \{N7,N6,\ldots,Z\}$, then $U_k^i = \{Z,P1,\ldots,P5,P6\}$ and:

$$u_k^i(t) = u_{k,1}^i(t) = Z \text{ with } q\left(u_{k,1}^i,k\right)$$

$$u_k^i(t) = u_{k,2}^i(t) = P1 \text{ with } q\left(u_{k,2}^i,k\right)$$

$$\vdots$$

$$u_k^i(t) = u_{k,7}^i(t) = P6 \text{ with } q\left(u_{k,7}^i,k\right)$$

R_k: If $L[s_i(t)] = Z$ and $L[\Delta s_i(t)] = Z$, then $U_k^i = \{N3,N2,\ldots,P2,P3\}$ and:

$$u_k^i(t) = u_{k,1}^i(t) = N3 \text{ with } q\left(u_{k,1}^i,k\right)$$

$$u_k^i(t) = u_{k,2}^i(t) = N2 \text{ with } q\left(u_{k,2}^i,k\right)$$

$$\vdots$$

$$u_k^i(t) = u_{k,7}^i(t) = P3 \text{ with } q\left(u_{k,7}^i,k\right)$$

where $(u_{k,1}^i(t), u_{k,2}^i(t), \ldots, u_{k,7}^i(t))$ are the possible actions corresponding to the rule $R_k$and state $w_i(t)$, those actions are all from the group candidates U_k^i. It indicates that the states of approximating the manifold will be given the actions that could result in $\rho_i < 0$ to decrease chattering effects, while the states of drifting away the manifold will be given the actions that could bring up $\rho_i > 0$ to steer the sliding variables back.

For an input state $w_i(t)$, the firing strength associated with each rule is generated as $\mu_i(t) = [\mu_{i,1}(t), \mu_{i,2}(t), \ldots, \mu_{i,k}(t), \ldots \mu_{i,km}(t)]$, where k_m is the number of rules. Greedy search is taken to acquire the maximum Q-value:

$$u^{*i}_k = \arg\max_{u_{k,l}^i \in U_k^i} q\left(u_{k,l}^i,k\right) \ , l = 1,2,..,7 \tag{60}$$

where u^{*i}_k is selected from the group of candidates U_k^i, the exploration exploitation policy (EEP) is applied to find the better solutions by randomly searching:

$$\widehat{u}_k^i = \begin{cases} \overline{u}_k^i \ , & \textit{with probability } \zeta \\ u_k^{*i}, & \textit{with probability } 1 - \ \zeta \end{cases} \tag{61}$$

$\widehat{u}_k^i$ is the last selected action and $\overline{u}_k^i$ is a random selection from the group of candidates U_k^i. The ζ is declined from 1 to 0 over the time to search the optimal action by the end of exploration. And then, the numerical output can be achieved by (54).

The Q-value associated with the selected action will be also calculated:

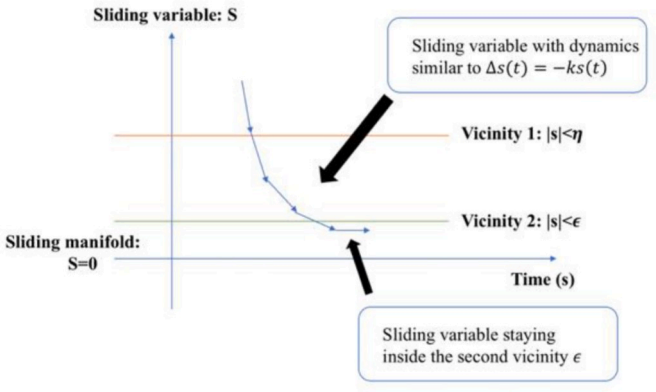

Fig. 7. The targets of the dynamics of the sliding variables.

$$Q_i\left(w_i(t),\widehat{u}^i(t)\right)=\frac{\sum_{k=1}^{km}\mu_{i,k}(t)q\left(\widehat{u}_k^i,k\right)}{\sum_{k=1}^{km}\mu_{i,k}(t)}\ ,\quad i=1,2,\ldots,N \tag{62}$$

$$Q_i^*\left(w_i(t),u^{*i}(t)\right)=\frac{\sum_{k=1}^{km}\mu_{i,k}(t)q\left(u^{*i}_k,\ k\right)}{\sum_{k=1}^{km}\mu_{i,k}(t)}\ ,\quad i=1,2,\ldots,N \tag{63}$$

We only consider the current reward so that discount factor $\lambda = 0$, according to (56) and (57), the update law of q-values can be achieved:

$$\Delta Q_i=c_i\left(w_i(t),\widehat{u}^i(t)\right)+\lambda\cdot Q_i^*\left(w_i(t+T_s),u^{*i}(t+T_s)\right)-Q_i\left(w_i(t),\widehat{u}^i(t)\right) \tag{64}$$

$$q\left(u_{k,l}^i,k\right)\leftarrow q\left(u_{k,l}^i,k\right)+\alpha\cdot\Delta Q_i\cdot\frac{\mu_{i,k}(t)}{\sum_{k=1}^{km}\mu_{i,k}(t)}\ ,\quad u_{k,l}^i=\widehat{u}_k^i,\ i=1,2,..,N \tag{65}$$

where c_i is the reward to evaluate the actions that result in the ρ_i. α is the learning efficiency of remembering new knowledge and forgetting old knowledge.

According to (18), the variance of $\widehat{k}_i$ can be considered as highly related to the variance of $\dot{s}_i$:

$$\Delta\dot{s}_i(t)=\Delta\Gamma_i(t)-\beta_i\Delta s_i(t)\pm\Delta\widehat{k}_i(t)\ ,i=1,2,\ldots,N \tag{66}$$

$$\Delta s_i(t+T_s)-\Delta s_i(t)\leftarrow\Delta\widehat{k}_i(t)\ ,i=1,2,\ldots,N \tag{67}$$

where $\Delta\dot{s}_i(t) = \dot{s}_i(t+T_s)-\dot{s}_i(t)$, $\Delta\Gamma_i(t)=\Gamma_i(t+T_s)-\Gamma_i(t)$ and $\Delta\widehat{k}_i(t) = \widehat{k}_i(t+T_s)-\widehat{k}_i(t)$. (66) holds when $s_i(t)s_i(t+T_s)>0$, which implies the sliding variable does not hit the manifold. The variance of $\dot{s}_i$ will be governed by the variance of $\widehat{k}_i$ if the $\Delta\Gamma_i$ and $\Delta s_i(t)$ are negligible compared to $\Delta\widehat{k}_i$, shown in (66). It is reasonable in this case because the $|s_i|$ is small in the designed vicinity of manifold and the small $\Delta\Gamma_i$ can be

Table 3
Function parameters of $f_r(\cdot)$ when ($\Delta s_i(t)>0$ and $s_i(t)>0$).

Situation:	When $\lvert s_i(t)\rvert\le\varepsilon_i$:	When $\lvert s_i(t)\rvert>\varepsilon_i$:
$\chi_{l,i}=$	$-\Delta s_i(t)-\Delta s_i(t)$	$-\Delta s_i(t)-\Delta s_i(t)-0.1s_i(t)$
$\chi_{m,i}=$	$-\Delta s_i(t)$	$-\Delta s_i(t)-\Delta s_i(t)-0.05s_i(t)$
$\chi_{r,i}=$	0	$-\Delta s_i(t)-\Delta s_i(t)$

Table 4
Parameters of $f_r(\cdot)$ when ($\Delta s_i(t)<0$ and $s_i(t)>0$).

Situation:	When $\lvert s_i(t)\rvert\le\varepsilon_i$:	When $\lvert s_i(t)\rvert>\varepsilon_i$:
$\chi_{l,i}=$	$-\Delta s_i(t)$	$-\Delta s_i(t)-0.06s_i(t+T_s)$
$\chi_{m,i}=$	$-\Delta s_i(t)-0.5\Delta s_i(t)$	$-\Delta s_i(t)-0.03s_i(t+T_s)$
$\chi_{r,i}=$	$-\Delta s_i(t)-\Delta s_i(t)$	$-\Delta s_i(t)$

Table 5
Parameters of $f_r(\cdot)$ when ($\Delta s_i(t)<0$ and $s_i(t)<0$).

Situation:	When $\lvert s_i(t)\rvert\le\varepsilon_i$:	When $\lvert s_i(t)\rvert>\varepsilon_i$:
$\chi_{l,i}=$	0	$-\Delta s_i(t)-\Delta s_i(t)$
$\chi_{m,i}=$	$-\Delta s_i(t)$	$-\Delta s_i(t)-\Delta s_i(t)-0.05s_i(t)$
$\chi_{r,i}=$	$-\Delta s_i(t)-\Delta s_i(t)$	$-\Delta s_i(t)-\Delta s_i(t)-0.1s_i(t)$

Table 6
Parameters of $f_r(\cdot)$ when ($\Delta s_i(t)>0$ and $s_i(t)<0$).

Situation:	When $\lvert s_i(t)\rvert\le\varepsilon_i$:	When $\lvert s_i(t)\rvert>\varepsilon_i$:
$\chi_{l,i}=$	$-\Delta s_i(t)-\Delta s_i(t)$	$-\Delta s_i(t)$
$\chi_{m,i}=$	$-\Delta s_i(t)-0.5\Delta s_i(t)$	$-\Delta s_i(t)-0.03s_i(t+T_s)$
$\chi_{r,i}=$	$-\Delta s_i(t)$	$-\Delta s_i(t)-0.06\ s_i(t+T_s)$

Table 7
CuBot parameters.

	Platform	Link1	Link2	Link3
Mass (kg)	8	0.5	0.5	0.5
Moment of inertia (I_x , $kg{\cdot}m^2$)	0.04	0.0004	0.00007	0.00007
Moment of inertia (I_y , $kg{\cdot}m^2$)	0.04	0.0004	0.00007	0.00007
Moment of inertia (I_z , $kg{\cdot}m^2$)	0.04	0.00007	0.0004	0.0004
Length (am)	0.1	0.05	0.05	0.05
Length (bm)	0.1	0.05	0.05	0.05

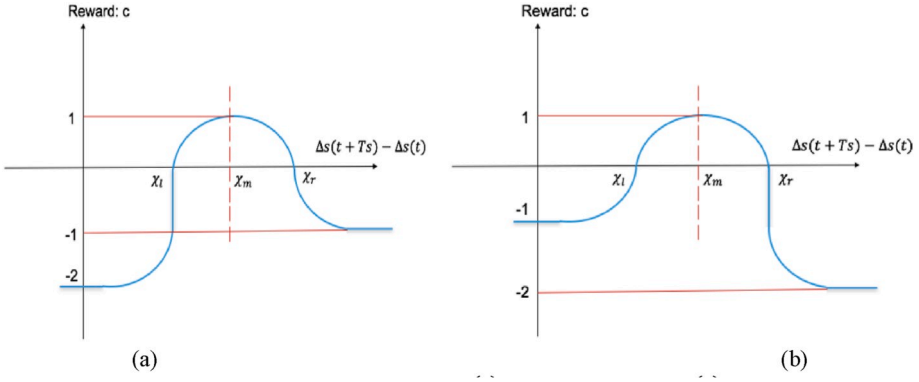

Fig. 8. The reward function when $s_i(t)>0$ (a) and when $s_i(t)>0$ (b).

Table 8
Scaling factors of the fuzzy logic inference and the 2 vicinities of each subsystem.

	$K_{fd}(s_i)$	$K_{fd}(\Delta s_i)$	$K_{fd}(\rho_i)$	η_i	ε_i
$i = 1$	0.0001	0.000001	0.8	0.0001	0.000001
$i = 2$	0.0001	0.000001	0.8	0.0001	0.000001
$i = 3$	0.0005	0.000001	0.8	0.0001	0.000001
$i = 4$	0.0005	0.000005	0.8	0.001	0.00001
$i = 5$	0.0005	0.000005	0.8	0.001	0.00001
$i = 6$	0.0005	0.000005	0.8	0.001	0.00001
$i = 7$	0.0005	0.000005	0.8	0.001	0.00005
$i = 8$	0.0005	0.000005	0.8	0.001	0.00005
$i = 9$	0.0005	0.000005	0.8	0.001	0.00005

achieved by small sampling period. As a result, we use the variance of $\dot{s}_i$ among 3 consecutive instants to indicate the effects from the variance of switching gain $\Delta\widehat{k}_i(t)$ that is determined by the output of fuzzy logic inference $\rho_i(t)$, shown in Fig. 6 a. And the mechanism of giving rewards to evaluate the actions is based on the chattering-free dynamic sliding variable $\dot{s} = -k_x s$ where s apprximates the manifold and never hit it so that chattering-free, shown in Fig. 6 b.

Considering the discrete form of $\dot{s} = -k_x s$ as the desired dynamics:

$$\Delta s_i^*(t) = -\mathscr{G}_i s_i(t), \quad i = 1, 2, \ldots, N \tag{68}$$

where $\mathscr{G}_i$ is a positive constant, and $\Delta s_i^*(t)$ is the desired variance of the sliding variable. We designed 2 different vicinities of sliding variables: the vicinity η_i and the second vicinity ε_i ($0 < \varepsilon_i < \eta_i$). We separate the dynamics of the sliding variables after entering into the vicinity η_i into 2 situations. The first situation is the sliding variable that is outside the designed second vicinity of manifold ε_i much smaller than the vicinity η_i so that $\eta_i > |s_i(t)| > \varepsilon_i$, where the sliding variable $s_i(t)$ is supposed to approximate to the manifold with the dynamics similar to Eq. (61) as much as possible. The second situation is the sliding variable that has been inside the second vicinity ε_i so that $|s_i(t)| \le \varepsilon_i < \eta_i$, where the sliding variable $s_i(t)$ is supposed to stay inside the vicinity ε_i and never hit the sliding manifold ($s_i = 0$) that results in the chattering effects. Clearly ε_i determines the errors of the sliding variable $s_i(t)$, which can be adjusted to the acceptable range by appropriately setting the parameters of the controller including δ_1, δ_2 and γ_i as well as the fuzzy domains of inputs and outputs shown in (47)-(49). The reinforcement learning can be concluded in Fig. 7.

Because the requirements on the dynamics of the sliding variable are different on the 2 vicinities ($\eta_i > |s_i(t)| > \varepsilon_i$ and $|s_i(t)| \le \varepsilon_i < \eta_i$), we let the non-fuzzy value of the sliding variable corresponding to the linguistic variables"$N1$" and "$P1$"in (50), be equal to the vicinity ε_i so that $\varepsilon_i = K_{fd}(s_i)\cdot\frac{N_1}{7}$. It is for the purpose that the actions on the linguistic states of "P1", "N1" and "Z" of the sliding variables ensure that the sliding variables stay inside the second vicinity ε_i.

The function of assigning the rewards to the corresponding actions is used in the reinforcement learning to determine the optimal rules among the candidates of the fuzzy rule. Eq. (61) takes the variance of $\Delta s_i(t)$ over a sampling period T_s as the input ($\chi_i = \Delta s_i(t + T_s) - \Delta s_i(t)$) and then outputs the reward to evaluate the fuzzy output $\rho_i(t)$. We designed 3 different variables $\chi_{l,i}$, $\chi_{m,i}$ and $\chi_{r,i}$ that will be used later.

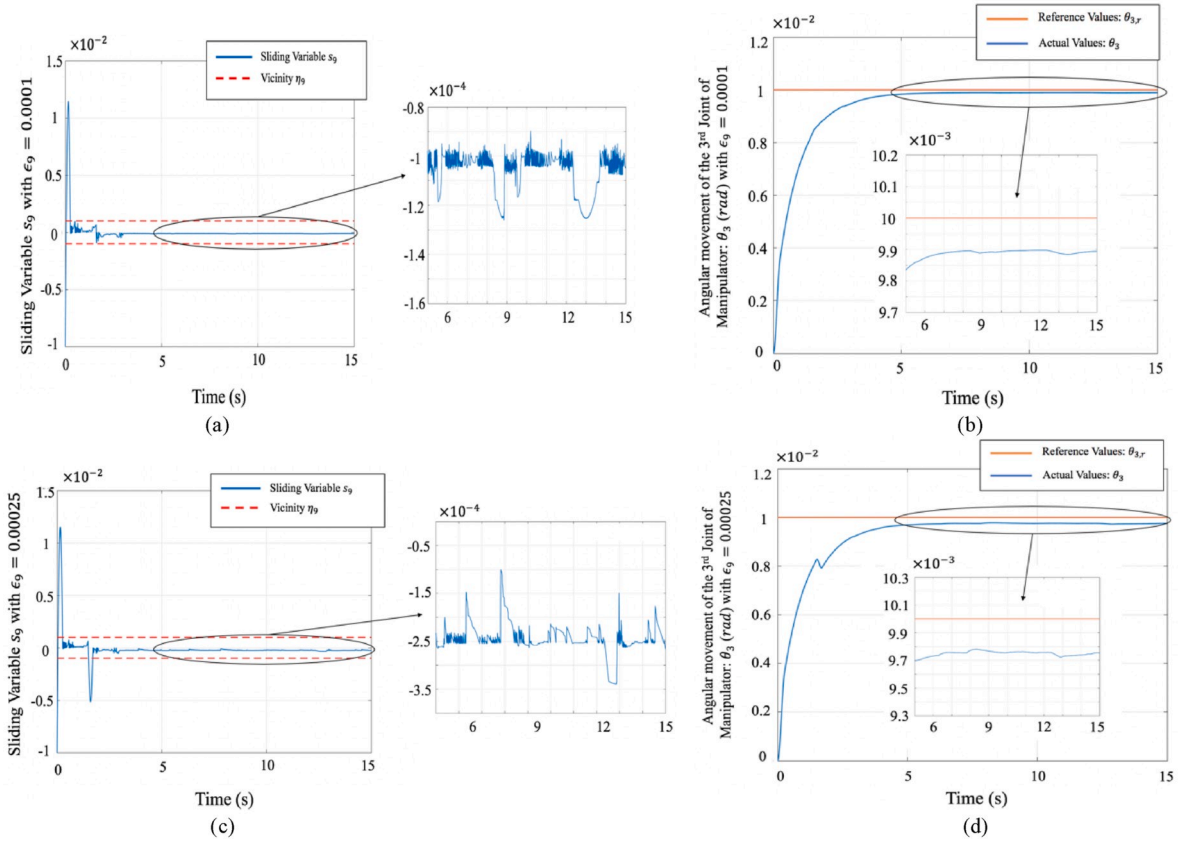

Fig. 9. The sliding variables and tracking performance of subsystem $i = 9$ when $\varepsilon_i = 0.0001$ *and* $\varepsilon_i = 0.00025$: (a). s_9 with $\varepsilon_i = 0.0001$. (b). θ_3 with $\varepsilon_i = 0.0001$. (c). s_9 with $\varepsilon_i = 0.00025$. (d). θ_3 with $\varepsilon_i = 0.00025$.

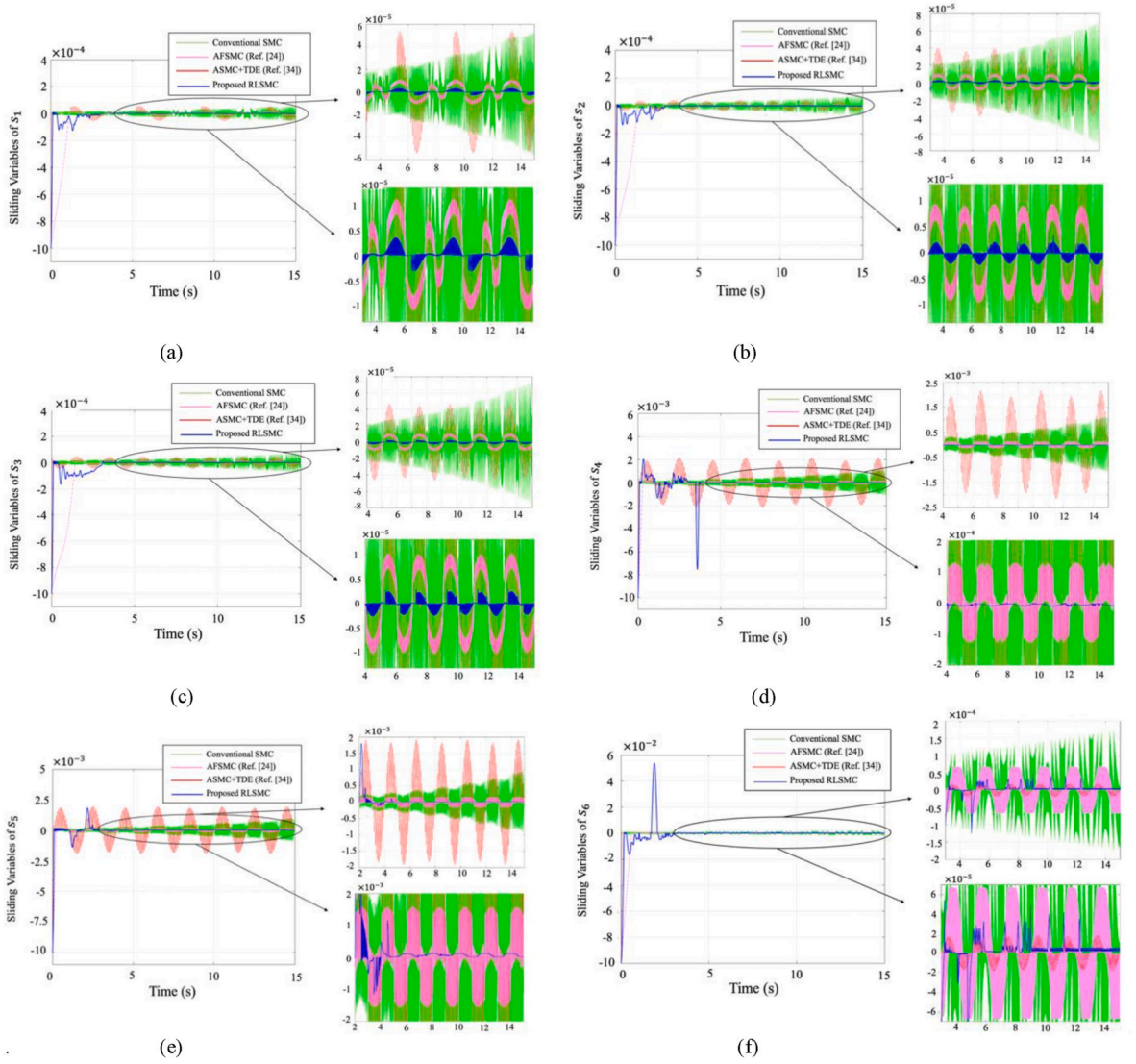

Fig. 10. The sliding variables of the subsystems of CubeSat (Subsystems $i = 1\tilde{}6$): (a) sliding variable s_1. (b) sliding variable s_2. (c) sliding variable s_3. (d) sliding variable s_4. (e) sliding variable s_5. (f) sliding variable s_6.

$$c_i(t) = f_r(\chi_i), \quad i = 1, 2, \ldots, N \tag{69}$$

where the function $f_r(\cdot)$ is the law of giving reward c_i. The details of (69) are given corresponding to Fig. 8. When $s_i(t) > 0$, the action $\rho_i(t)$ bringing up the χ_i smaller than χ_l will be given the negative reward smaller than -1, meaning either insufficient decline or excessive increase of $\widehat{k}_i(t)$ (sliding variable moving toward to the manifold too fast). And the action $\rho_i(t)$ resulting in the χ_i bigger than χ_r will be given the negative reward bigger than -1, meaning either excessive decline or insufficient increase of $\widehat{k}_i(t)$ (sliding variable drifting away from the manifold). When $s_i(t) < 0$, the action $\rho_i(t)$ resulting in the χ_i smaller than χ_l will be given the negative reward bigger than -1, meaning either insufficient increase or excessive decline of $\widehat{k}_i(t)$. And the action $\rho_i(t)$ resulting in the χ_i bigger than χ_r will be given the negative reward smaller than -1, meaning either excessive increase or insufficient decline of $\widehat{k}_i(t)$. For both of the $s_i(t) > 0$ and $s_i(t) < 0$, χ_m that will result in the maximum reward is the ideal value of $\Delta s_i(t+T_s) - \Delta s_i(t)$ corresponding to (68). The details of $\chi_{l,i}$, $\chi_{m,i}$ and $\chi_{r,i}$ are given as following:

If the sliding variable is diverging above the designed sliding manifold ($\Delta s_i(t) > 0$ and $s_i(t) > 0$), the $\chi_{l,i}$, $\chi_{m,i}$ and $\chi_{r,i}$ are shown in Table 3 corresponding to Fig. 8 (a).

If the sliding variable is converging above the designed sliding manifold (($\Delta s_i(t) < 0$ and $s_i(t) > 0$), the $\chi_{l,i}$, $\chi_{m,i}$ and $\chi_{r,i}$ are shown as following equations corresponding to Fig. 8 (a).

If the sliding variable is diverging under the designed sliding manifold ($\Delta s_i(t) < 0$ and $s_i(t) < 0$), the $\chi_{l,i}$, $\chi_{m,i}$ and $\chi_{r,i}$ are shown as following equations corresponding to Fig. 8 (b).

If the sliding variable is converging under the designed sliding manifold ($\Delta s_i(t) > 0$ and $s_i(t) < 0$), the $\chi_{l,i}$, $\chi_{m,i}$ and $\chi_{r,i}$ are shown as following equations corresponding to Fig. 8 (b).

The above definitions of $\chi_{l,i}$, $\chi_{m,i}$ and $\chi_{r,i}$ indicate that the fuzzy rules are learned in order to achieve the desired dynamics of the sliding variable similar to Eq. (61) shown in the 3rd column of Tables 3–6 when $|s_i(t)| > \varepsilon_i$. After the sliding variable enters into the vicinity $|s_i(t)| < \varepsilon_i$, the fuzzy rules are learned to make efforts to cage the variable $|s_i(t)|$ inside this vicinity without hitting the sliding manifold shown in the 2nd column of Tables 3–6.

The q values of the Q matrix in the reinforcement Q learning can be obtained by largely visiting the pair of the action (the candidates of each fuzzy rule) and the state (the firing strength of each fuzzy rule). However, the vast time used on the exploration of Q matrix will result in the inefficient learning and the bad performance on tracking signals. Intuitively, if an action of a state brings up the insufficient increase or

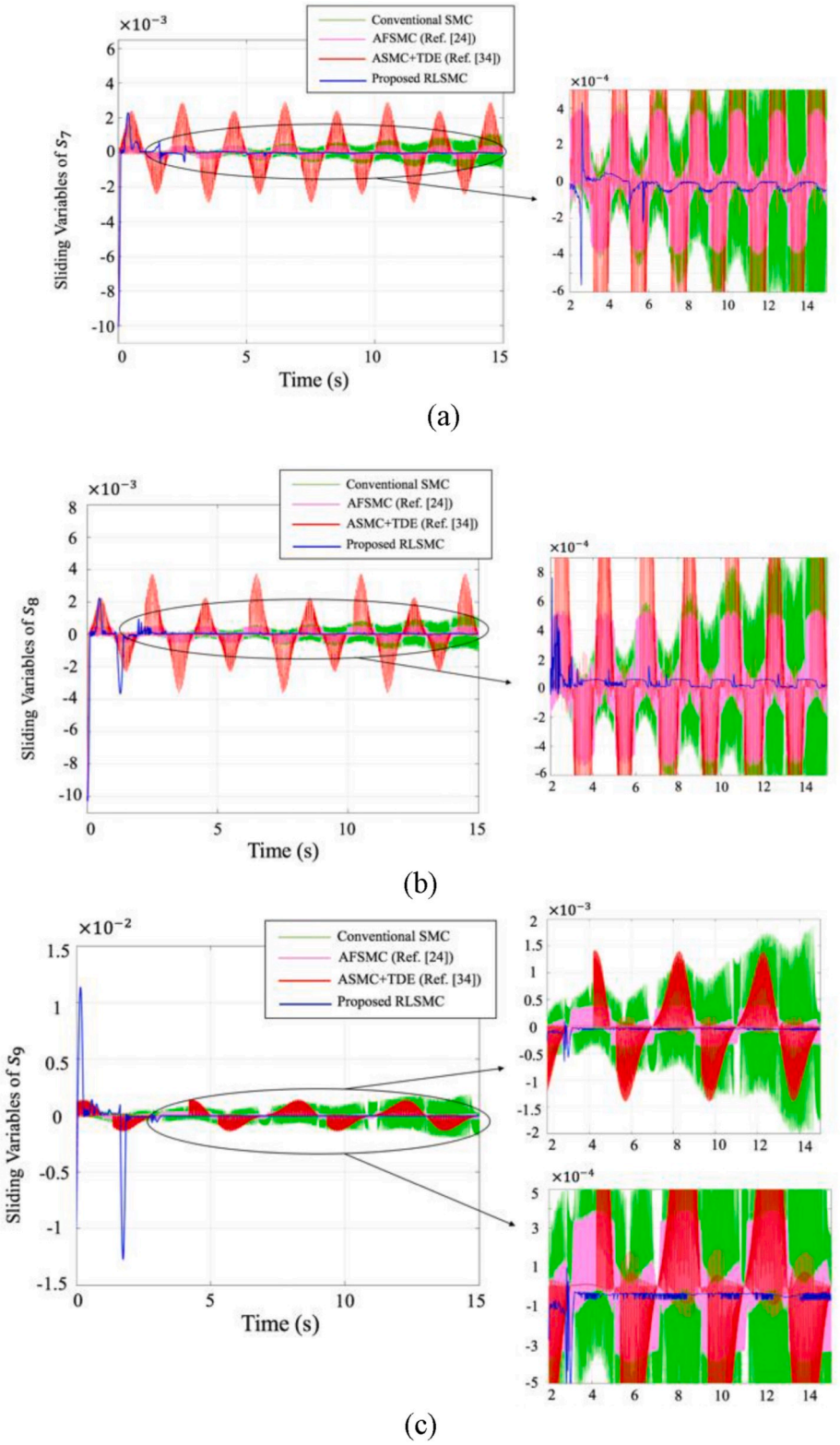

Fig. 11. The sliding variables of the mounted manipulator (Subsystems $i = 7$~9): (a) sliding variable s_7. (b) sliding variable s_8. (c) sliding variable s_9.

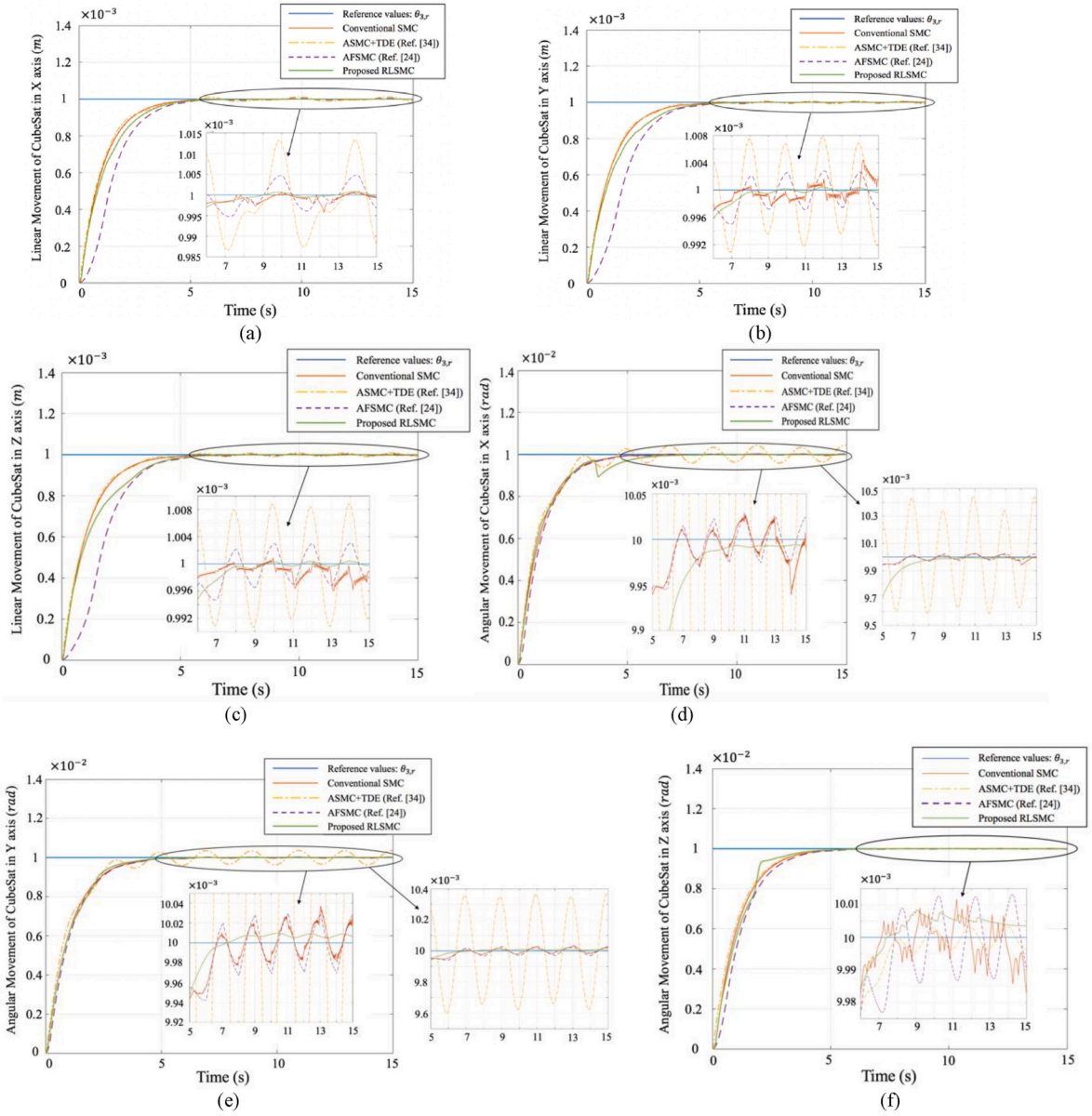

Fig. 12. The tracking performance of the CubeSat (subsystems $i = 1\tilde{}6$): (a) Linear movement of CubeSat in X axis. (b) Linear movement of CubeSat in Y axis. (c) Linear movement of CubeSat in Z axis. (d). Angular movement of CubeSat in X axis. (e) Angular movement of CubeSat in Y axis. (f) Angular movement of CubeSat in Z axis.

excessive decrease on the switching gain $\widehat{k}_i$(eg. The $\widehat{k}_i$ increases insufficiently so that the sliding variable drift away from the sliding manifold), the actions of this state that show smaller linguistic values could result in the more insufficient increase or the more excessive decrease. Similarly, if an action of a state results in the insufficient decrease or excessive increase on the switching gain $\widehat{k}_i$ (eg. The $\widehat{k}_i$ decreases insufficiently so that the sliding variable hits the sliding manifold, resulting in the chattering effects), the actions of this state that show the bigger linguistic values could bring up the more insufficient decrease or the more excessive increase.

As a result, the q-values of some actions can be updated even though these actions are not visited. The multiple q-values of the Q matrix can be updated by only a pair of action and state to improve the efficiency of learning. If the action of a state is regarded as leading to "the $\widehat{k}_i$ decreasing not enough" or "the $\widehat{k}_i$ increasing excessively", not only is this action given an update on q-values according to the acquired reward, but also the q-values of the other actions of this state that have bigger linguistic variables will be updated. Similarly, if the action of a state is regarded as leading to "the $\widehat{k}_i$ increasing not enough" or "the $\widehat{k}_i$ decreasing excessively", not only is this action given an update on q-values according to the acquired reward, but also the q-values of the other actions of this state that have smaller linguistic variables will be updated. For example, if an action $u^i_{k,l}(t)$ on the state $w_i(t)$ corresponding to the rule $R_k(L[s_i(t)],\ L[\Delta s_i(t)])$ results a reward smaller than -2, not only is the $q(u^i_{k,l}, k)$ given a update but also q-values $q(u^i_{k,L}, k)$, *for all* $L > l$ will be updated. In this paper, we only consider the current reward with the discount $\lambda = 0$. Consequently, the conventional law of updating the Q matrix of (64) and (65) can be modified to the innovative updating law in (70)-(72).

If the reward $c_i(t) \in (0, 1]$:

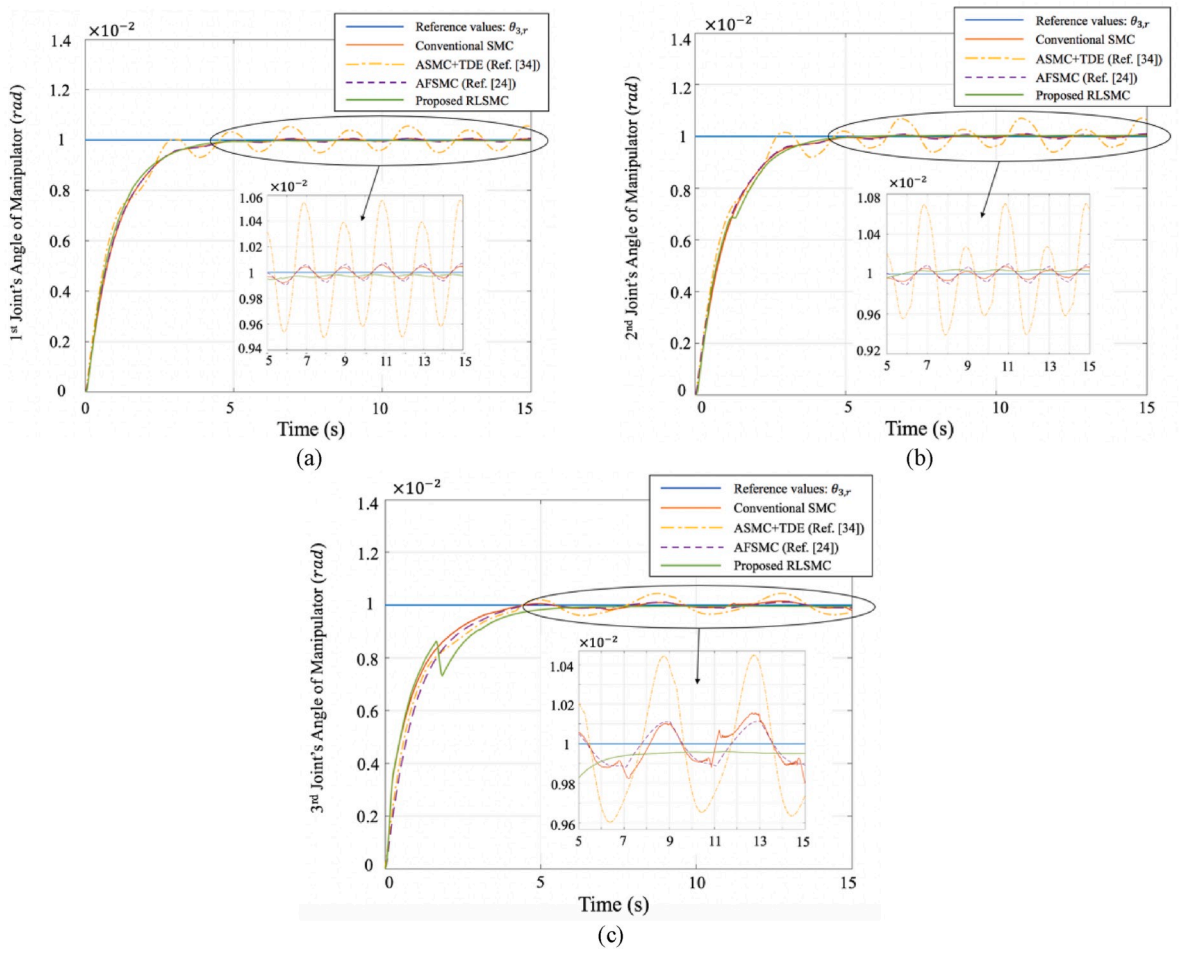

Fig. 13. The tracking performance of mounted manipulator (subsystems $i = 7$~9): (a) The 1st Joint's angle of Manipulator. (b) The 2nd Joint's angle of Manipulator. (c) The 3rd Joint's angle of Manipulator.

$$q\left(u_{k,L}^{i},k\right)\leftarrow q\left(u_{k,L}^{i},k\right)+\alpha\cdot\left[c_i\left(w_i(t),\widehat{u}^{i}(t)\right)-Q_i\left(w_i(t),\widehat{u}^{i}(t)\right)\right]\cdot \frac{\mu_{i,k}(t)}{\sum_{k=1}^{km}\mu_{i,k}(t)}\ ,\ u_{k,l}^{i}=\widehat{u}_{k}^{i},\ i=1,2,..,N \tag{70}$$

where $L = l$, only the q-value of the taken action is updated as a normal Q learning.

If the reward $c_i(t) \in [-1, 0)$:

$$q\left(u_{k,L}^{i},k\right)\leftarrow q\left(u_{k,L}^{i},k\right)+\alpha\cdot\left[c_i\left(w_i(t),\widehat{u}^{i}(t)\right)\cdot\varsigma_i^{l-L}-Q_i\left(w_i(t),\widehat{u}^{i}(t)\right)\right]\cdot \frac{\mu_{i,k}(t)}{\sum_{k=1}^{km}\mu_{i,k}(t)}\ ,\ u_{k,l}^{i}=\widehat{u}_{k}^{i},\ i=1,2,..,N \tag{71}$$

where $L \le l$, $\varsigma_i > 1$ is a positive constant bigger than 1. (71) means that ρ_i bringing up either insufficient increase of $\widehat{k}_i$ or excessive decline of $\widehat{k}_i$ allows the update of the q-values of multiple actions. The other actions of the current state will be given the update on the q-values with the reward modified by ς_i^{l-L}, indicating the actions that correspond to the smaller linguistic values will be given smaller rewards.

If the reward $c_i(t) \in [-2, -1)$:

$$q\left(u_{k,L}^{i},k\right)\leftarrow q\left(u_{k,L}^{i},k\right)+\alpha\cdot\left[c_i\left(w_i(t),\widehat{u}^{i}(t)\right)\cdot\varsigma_i^{L-l}-Q_i\left(w_i(t),\widehat{u}^{i}(t)\right)\right]\cdot \frac{\mu_{i,k}(t)}{\sum_{k=1}^{km}\mu_{i,k}(t)}\ ,\ u_{k,l}^{i}=\widehat{u}_{k}^{i},\ i=1,2,..,N \tag{72}$$

where $L \ge l$, $\varsigma_i > 1$ is a positive constant bigger than 1. The (72) means that ρ_i bringing up either insufficient decrease of $\widehat{k}_i$ or excessive increase of $\widehat{k}_i$ allows the update of the q-values of multiple actions. The other actions of the current state will be given the update on the q-values with the reward modified by ς_i^{L-l}, indicating the actions that correspond to the bigger linguistic values will be given smaller rewards.

It is noticeable that the taken action in the current state ($l = L$) will be updated as a normal Q learning because $\varsigma_i^0 = 1$.

4. Simulation results

The numerical simulations are carried to validate the proposed control strategy in this section. The conventional SMC, the time delay estimation (TDE) based adaptive sliding mode controller (ASMC) in Refs. [34] and the adaptive fuzzy sliding mode controller in Ref. [24] are used to compare with the proposed reinforcement learning based SMC. The CuBot shown in Fig. 2 has 9° of freedom (DOF), consisting of the angular movements of the 3 joints, the 3 angular movements and 3 linear movements of the CubeSat along the axes of roll, pitch and yaw respectively. The CuBot parameters are shown in Table 7.

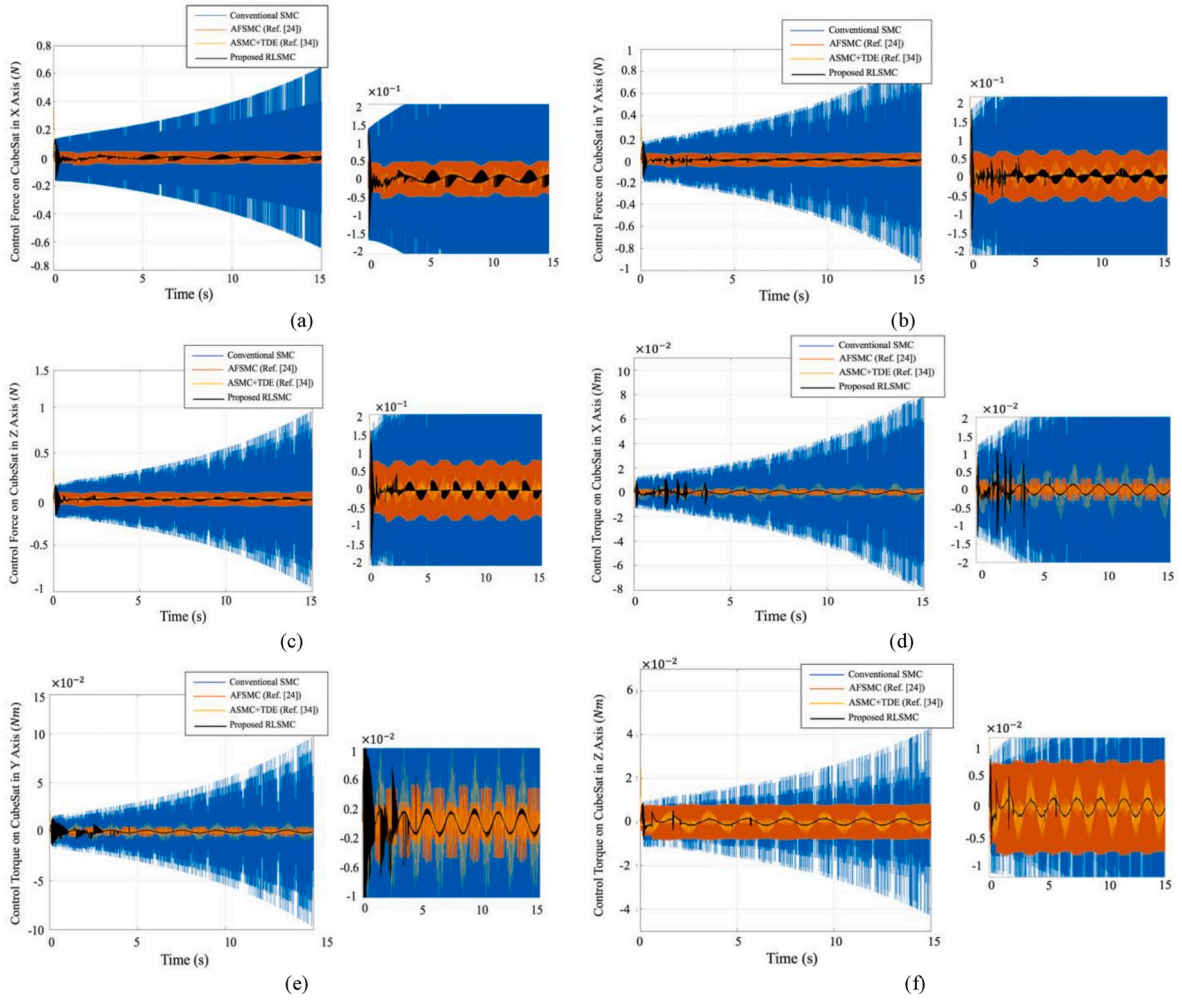

Fig. 14. The control torque and force applied on the CubeSat: (a) Control Force on CubeSat in X axis. (b) Control Force on CubeSat in Y axis. (c) Control Force on CubeSat in Z axis. (d) Control Torque on CubeSat in X axis. (e) Control Torque on CubeSat in Y axis. (f) Control Torque on CubeSat in Z axis.

We suppose that the initial states of the system are all zero so that the position of the CubeSat $\delta_{b0} = [0,0,0]^T$ m, the attitude of the CubeSat $\theta_{b0} = [0,0,0]^T$ rad and the angles of the manipulator joints $\theta_{m0} = [0,0,0]^T$ rad. The target position and target attitude of the CubeSat are set as $\delta_{b^*} = [0.001, 0.001, 0.001]^T$ m and $\theta_{b^*} = [0.01, 0.01, 0.01]^T$rad, while the target angles of the manipulator joints are set as $\theta_{m^*} = [0.01, 0.01, 0.01]^T$rad. The controller proposed to control the motions of both the CubeSat and the robotic manipulator needs to convert the initial state of δ_{b0}, θ_{b0} and θ_{m0} to the target state of δ_{b^*}, θ_{b^*} and θ_{m^*}. This process is equivalent to track the step signals.

The robust term $\beta_i = 0.0001$, for $i = 1, 2.., 9$. The small β_i means that the dynamics of the sliding variable s_i are mainly influenced by the $\widehat{k}_i$ that is governed by the designed Q learning based fuzzy logic inference. The $K_s = diag([1,1,1,1,1,1,1,1,1])$ determines the settling time of the system. The $\gamma_i = 150$, for $i = 1, 2.., 9$ are selected with the purpose that we select the big value for γ_i to achieve the robustness and short time of approximating the sliding manifold, and then the chattering effects are supposed to be mitigated by the adaptive law governed by the fuzzy logic inference. The $\delta_1 = 2.7$ and $\delta_2 = 0.001$ are selected, and $h_i = 10$ and $\varsigma_i = 2.7$, for all for $i = 1, 2.., 9$ are selected. With respect to the parameters of the modified Q learning based fuzzy logic inference, the scaling factors and vicinities are shown in Table 8. The learning efficiency is selected as $\alpha = 0.6$ for an efficient learning process. To carry the simulation with the scenario that the system uncertainties and disturbances exist, we use the actual system matrices A and B that are 40% and 30% respectively off from the nominal system matrices $\widehat{A}$ and $\widehat{B}$ that are used in the controller to decouple the system. The method of setting uncertainty is same as [22].

The disturbances used in simulation are of the combination of the sinusoid and cosine functions. In details, the disturbances shown in Eq. (11) are assumed $D = [D_c, D_m]^T$ that consist of the unknown disturbances applied on the mounted robotic manipulator $D_m = \left[0.002\sin(\pi t), 0.002\sin(\pi t), 0.001\sin(\pi t) + 0.001\cos\left(\frac{\pi}{2}t\right)\right]^T$ and the unknown disturbances applied on the Cube5Sat $D_c[= \left[0.01\sin\left(\frac{\pi}{2}t\right), 0.001\cos(\pi t), 0.0015\right.$

$\left.\sin(\pi t), 0.002\sin(\pi t), 0.002\sin(\pi t), 0.002\sin(\pi t)\right]^T$

.

In the simulation of the proposed RLSMC, we set the probability of randomly searching the optimal solutions in the EEP as a decreasing values from 1 to 0 along the simulation time, indicating that the Q learning agent initially is given a high probability to randomly search the potentially better solution during the period when the q values of most actions are not updated and visited, and then the agent is inclined to take the actions that have the biggest q values (local optimal). The details of the probability are that $\zeta = 2.7^{-0.05t}$ when $t < 3s$, and $\zeta = 0$ when $t \geq 3s$. After the period of randomly searching, the sliding

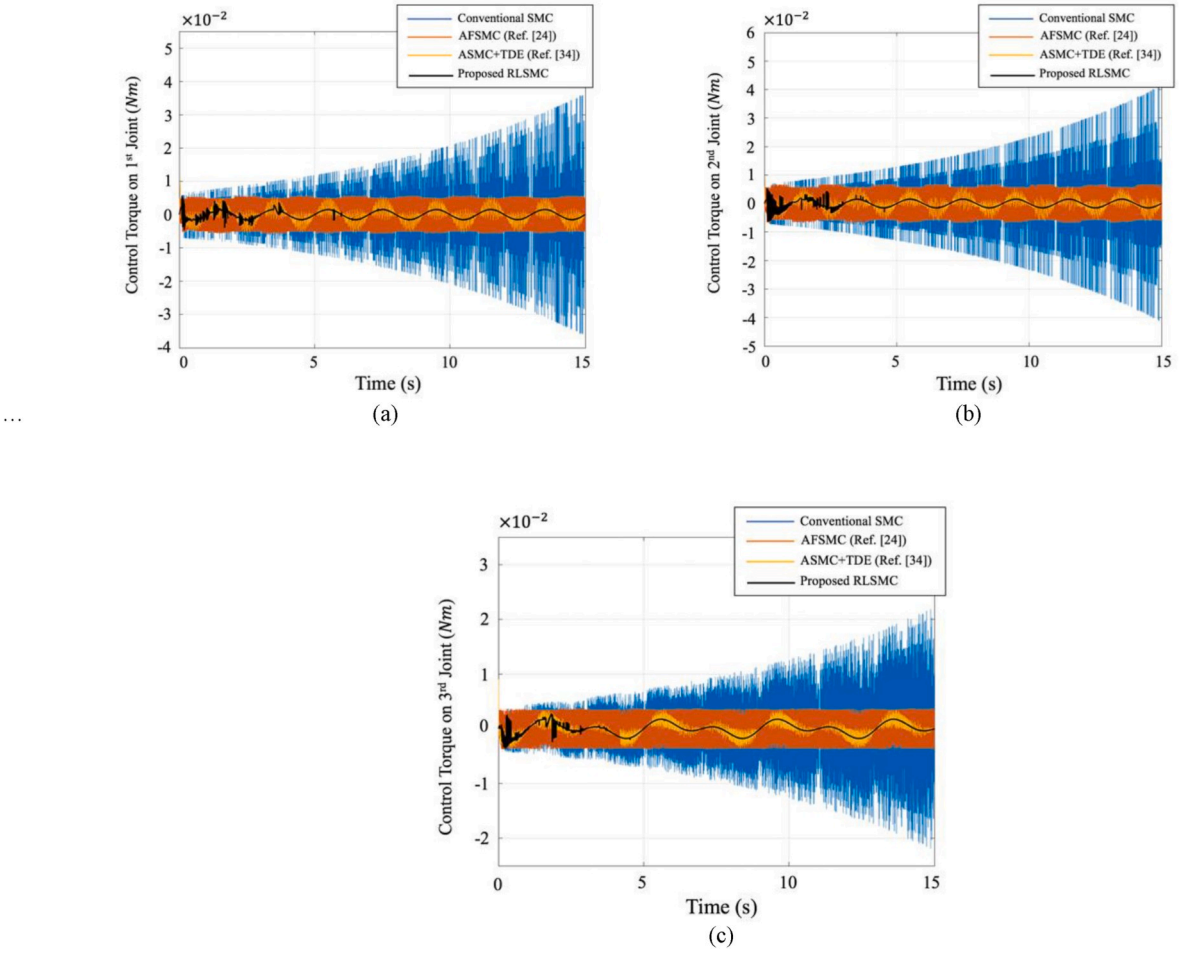

Fig. 15. The control torque applied on the joints of mounted manipulator: (a) Control Torque on the 1st Joint of Manipulator. (b) Control Torque on the 2nd Joint of Manipulator. (c) Control Torque on the 3rd Joint of Manipulator.

variables learn to stay inside the vicinity ($|s_i(t)| < \varepsilon_i$), and clearly the ε_i determines the errors of the tracking performance that the bigger values of ε_i mean the bigger errors of tracking performance. More precisely, we take the sliding variable of the 9^{th} subsystem (the angular movement of the 3rd joint of the manipulator) as the instance shown in Fig. 9, the sliding variable s_9 that is designed to stay inside the vicinity $\varepsilon_i = 0.0001$ shows the less steady state errors and better the tracking performance than the sliding variable s_9 that is designed to stay inside the vicinity $\varepsilon_i = 0.00025$. The value of ε_i should be selected carefully because it determines the steady state errors. In details, the too big value of ε_i will result in the sliding variable far away from the sliding manifold ($|s_i|$ is large), which could then bring up the large tracking error, while the too small value of ε_i will result in the sliding variables never enter into the second vicinity ε_i because of the limitation of capability of the controller (the sampling period cannot be the absolute zero).

The proposed control strategy is compared with the TDE-ASMC [34] and AFSMC [24] as well as a conventional sliding mode controller in which the adaptive law of switching gain is positively proportional to the absolute value of the sliding variables. Figs. 10 and 11 show the dynamics of the sliding variables of all subsystems $i = 1\tilde{}9$. Clearly, the conventional SMC has the worst chattering effects among 4 controllers because of the monotonously increasing switching gain, shown as the green curves in Figs. 10 and 11 that reflect the nature of discontinuous control inputs near the sliding manifold. The chattering can be much worse by increasing the values of the switching gain and the sampling interval. Although AFSMC [24] shows better chattering effect compared to the conventional SMC, the sliding variables drifting away the manifold (Fig 10a~c) and chattering (Fig 10d~f and Fig 11c) are observed, which results in the compromised tracking accuracy and chattering control signals. TDE-ASMC [34] yields the rebounding sliding variables that repeatedly approaches and leaves the manifold in all subsystems ($i = 1\tilde{}9$), shown as the red curves in Figs. 10 and 11. The tracking errors have a positive correlation to the distance between the manifold and the location to which the sliding variables travel, which means the tracking accuracy can be maintained by tuning the controller parameters (TDE-ASMC [34]) to make the sliding variable travelling within a small region near the manifold (Fig. 10f). Otherwise, the tracking accuracy could be decreased because of the sliding variables leaving the manifold (Fig 10a~e and Fig 11a~c). However, it is challenging and time-consuming for users to find the appropriate controller parameters manually. The proposed RLSMC shows the sliding variables that irregularly fluctuate around the sliding manifold during the initial stage of control for all subsystems (shown as the blue curves in Figs. 10 and 11 when $t \leq 4s$), which is caused by the reinforcement learning mechanism trying the actions randomly during the stage of policy exploration. However, the smallest chattering and smallest drifting sliding variables are achieved by the proposed RLSMC after the optimal policy is learnt ($t > 4s$).

The tracking performance is shown in Figs. 12 and 13. Clearly, the TDE-ASMC [34] and AFSMC [24] presents the significant tracking errors because of the rebounding sliding variables, while the conventional SMC shows the tracking errors caused by chattering effects of sliding variables. The TED-ASMC has the more significantly rebounding sliding variables than AFSMC, and therefore TED-ASMC shows the bigger

tracking errors than that of AFSMC. The smallest tracking errors during the later stage ($t > 4s$) are achieved by the proposed RLSMC in all subsystems. The fluctuation of system states under RLSMC during the initial seconds ($t < 4s$) is observed because of the reinforcement learning mechanism trying the actions randomly during the stage of exploration.

The control signals (torques and forces) are shown in Figs. 14 and 15. Clearly, the conventional SMC shows the most significantly chattering control signals (shown as the blue curves in Figs. 14 and 15) among 4 controllers, which is followed by AFSMC [24] (shown as the orange curves). TDE-ASMC [34] shows the much less chattering control signals than conventional SMC and AFSMC [24], shown as the yellow curves, while the proposed RLSMC shows the smallest chattering control signals among 4 controllers (shown as the black curves). The less chattering effects not only provide the smoother torques and forces to the system, but also it means the less energy consumption that is important to implement the missions in space especially for the un-rechargeable fuels such as the gas. By using the fuzzy logic inference with the sampling time smaller than the dynamics of the non-random disturbances, the proposed controller can be regarded as the controller equipped with a disturbance observer.

As a result, the proposed controller attenuates the chattering effects of both the sliding variables and the control torques/forces are significantly in comparison with TED-ASMC [34], AFSMC [24] and conventional SMC. Although the fluctuations of system sates and the sliding variables are observed on the initial period of learning, the chattering effects are significantly mitigated after learning the optimal actions among the given candidates, which results in the improvement of tracking accuracy and the smoothness of control inputs.

5. Conclusion

This paper modelled the dynamics of an 8U CubeSat equipped with a 3-rigid-link robotic manipulator. To control the angular movements of the manipulator joints and the attitude and position of the CubeSat simultaneously, the sliding mode controller with the aid of the decoupling method converting the MIMO system into the multiple SISO systems is used. For mitigating the chattering effects, we designed the reinforcement Q learning based fuzzy logic inference to tune the derivatives of the switching gain to avoid the sliding variables either hitting the sliding manifold or drifting away from the manifold, which can achieve the improvement of tracking accuracy and the smooth control inputs. The proposed reinforcement learning based fuzzy sliding mode controller has the 2 advantages: 1. It can yield the UUB system errors regardless of the failure of finding the optimal parameters of adaptive law of switching gain. The optimal fuzzy rules outputting the optimal parameters can yield small tracking errors while the inferior fuzzy rules can yield a bigger but still bounded tracking errors, which not only avoids the loss of stability resulted from negative derivative of switching gain but also can offer the reinforcement learning mechanism with a safe environment to learn optimal policies. 2. The modified Q learning mechanism in this paper can utilize a pair of action and state to update the q-values of multiple actions, which can improve the efficiency of learning. The simulation validates the effectiveness of the proposed control scheme.

Declaration of competing interest

None.

References

[1] A. Flores-Abad, O. Ma, K. Pham, S. Ulrich, A review of space robotics technologies for on-orbit servicing, Prog. Aero. Sci. 68 (2014) 1–26.

[2] G. Hirzinger, B. Brunner, J. Dietrich, J. Heindl, Rotex – the first remotely controlled robot in space, in: Proceedings of the IEEE International Conference of Robotics and Automation, SanDiego, 1994, pp. 2604–2611.

[3] M. Oda, K. Kibe, F. Yamagata, ETS-VII, space robot in-orbit experiment satellite, in: Robotics and Automation, 1996. Proceedings., 1996 IEEE International Conference on, vol. 1, IEEE, 1996, April, pp. 739–744.

[4] Didot G, Oort M, Kouwen J, Verzijden P. The ERA system: control architecture and performance results. In: International Symposium on Artificial Intelligence, Robotics and Automation in Space, Montreal, Canada.

[5] M.A. Diftler, R.O. Ambrose, Robonaut: a robotic astronaut assistant, in: ISAIRAS Conference, vol. 2001, 2001, June.

[6] H.S. Jayakody, L. Shi, J. Katupitiya, N. Kinkaid, Robust adaptive coordination controller for a spacecraft equipped with a robotic manipulator, J. Guid. Contr. Dynam. (2016) 2699–2711.

[7] S. Dubowsky, E. Papadopoulos, The kinematics, dynamics, and control of free-flying and free-floating space robotic systems, IEEE Trans. Robot. Autom. 9 (5) (1993) 531–543.

[8] X. Yu, L. Chen, Observer-based two-time scale robust control of free-flying flexible-joint space manipulators with external disturbances, Robotica 35 (11) (2017) 2201–2217.

[9] C. Jiang, K.L. Teo, H. Xu, L. Caccetta, G.R. Duan, Adaptive Jacobian force/position tracking for space free-flying robots with prescribed transient performance, Robot. Autonom. Syst. 72 (2015) 235–247.

[10] I.M. Da Fonseca, L.C. Goes, N. Seito, M.K. da Silva Duarte, É.J. de Oliveira, Attitude dynamics and control of a spacecraft like a robotic manipulator when implementing on-orbit servicing, Acta Astronaut. 137 (2017) 490–497.

[11] X.F. Liu, H.Q. Li, Y.J. Chen, G.P. Cai, Dynamics and control of space robot considering joint friction, Acta Astronaut. 111 (2015) 1–18.

[12] X. Yang, S.S. Ge, July). Dynamics and tracking control for a free-flying space robot with rigid-flexible links, in: Information and Automation (ICIA), 2014 IEEE International Conference on, IEEE, 2014, pp. 394–399.

[13] Z. Chu, J. Cui, F. Sun, Fuzzy adaptive disturbance-observer-based robust tracking control of electrically driven free-floating space manipulator, IEEE Syst. J. 8 (2) (2014) 343–352.

[14] W. He, Y. Chen, Z. Yin, Adaptive neural network control of an uncertain robot with full-state constraints, IEEE Trans. Cybernet. 46 (3) (2016) 620–629.

[15] Y. Li, S. Qiang, X. Zhuang, O. Kaynak, Robust and adaptive backstepping control for nonlinear systems using RBF neural networks, IEEE Trans. Neural Network. 15 (3) (2004) 693–701.

[16] J. Fei, H. Ding, Adaptive sliding mode control of dynamic system using RBF neural network, Nonlinear Dynam. 70 (2) (2012) 1563–1573.

[17] H.F. Ho, Y.K. Wong, A.B. Rad, Robust fuzzy tracking control for robotic manipulators, Simulat. Model. Pract. Theor. 15 (7) (2007) 801–816.

[18] C.Y. Su, T.P. Leung, A sliding mode controller with bound estimation for robot manipulators, IEEE Trans. Robot. Autom. 9 (2) (1993) 208–214.

[19] S.J. Yoo, J.B. Park, Y.H. Choi, Adaptive dynamic surface control of flexible-joint robots using self-recurrent wavelet neural networks, IEEE Trans. Syst. Man Cybernet., Part B (Cybernet.) 36 (6) (2006) 1342–1355.

[20] J. Qiao, X. Meng, W. Li, An incremental neuronal-activity-based RBF neural network for nonlinear system modeling, Neurocomputing 302 (2018) 1–11.

[21] Z. Zhu, Y. Xia, M. Fu, Adaptive sliding mode control for attitude stabilization with actuator saturation, IEEE Trans. Ind. Electron. 58 (10) (2011) 4898–4907.

[22] L. Shi, S. Kayastha, J. Katupitiya, Robust coordinated control of a dual-arm space robot, Acta Astronaut. 138 (2017) 475–489.

[23] L. Shi, J. Katupitiya, N. Kinkaid, July). A robust attitude controller for a spacecraft equipped with a robotic manipulator, in: American Control Conference (ACC), 2016, IEEE, 2016, pp. 4966–4971.

[24] A. Saghafinia, H.W. Ping, M.N. Uddin, K.S. Gaeid, Adaptive fuzzy sliding-mode control into chattering-free IM drive, IEEE Trans. Ind. Appl. 51 (1) (2015) 692–701.

[25] P.S. Londhe, M. Santhakumar, B.M. Patre, L.M. Waghmare, Task space control of an autonomous underwater vehicle manipulator system by robust single-input fuzzy logic control scheme, IEEE J. Ocean. Eng. 42 (1) (2017) 13–28.

[26] Q. Gao, J. Liu, T. Tian, Y. Li, Free-flying dynamics and control of an astronaut assistant robot based on fuzzy sliding mode algorithm, Acta Astronaut. 138 (2017) 462–474.

[27] A. Kumar, R. Sharma, Linguistic Lyapunov reinforcement learning control for robotic manipulators, Neurocomputing 272 (2018) 84–95.

[28] S.A.A. Moosavian, E. Papadopoulos, Explicit dynamics of space free-flyers with multiple manipulators via SPACEMAPLE, Adv. Robot. 18 (2) (2004) 223–244.

[29] B. Yang, Y. He, J. Han, G. Liu, Rotor-flying manipulator: modeling, analysis, and control, Math. Probl Eng. (2014), 2014.

[30] W. Xu, J. Peng, B. Liang, Z. Mu, Hybrid modeling and analysis method for dynamic coupling of space robots, IEEE Trans. Aero. Electron. Syst. 52 (1) (2016) 85–98.

[31] W. Xu, B. Liang, Y. Xu, Survey of modeling, planning, and ground verification of space robotic systems, Acta Astronaut. 68 (11–12) (2011) 1629–1649.

[32] P. Huang, J. Yuan, B. Liang, Adaptive sliding-mode control of space robot during manipulating unknown objects, in: Control and Automation, 2007. ICCA 2007. IEEE International Conference on, IEEE, 2007, May, pp. 2907–2912.

[33] A. Thowsen, Uniform ultimate boundedness of the solutions of uncertain dynamic delay systems with state-dependent and memoryless feedback control, Int. J. Contr. 37 (5) (1983) 1135–1143.

[34] J. Baek, M. Jin, S. Han, A new adaptive sliding-mode control scheme for application to robot manipulators, IEEE Trans. Ind. Electron. 63 (6) (2016) 3628–3637.

[35] C.C. Lee, Fuzzy logic in control systems: fuzzy logic controller, IEEE Trans. Syst. Man Cybernet. 20 (2) (1990) 404–418.

[36] J. Ragot, M. Lamotte, Fuzzy logic control, Int. J. Syst. Sci. 24 (10) (1993) 1825–1848.