Original Research Article

SCIENCE PROGRESS

Science Progress
2024, Vol. 107(4) 1–21

DOI: 10.1177/00368504241286381
journals.sagepub.com/home/sci

Sage

Sliding mode control with self-adaptive parameters of a 5-DOF hybrid robot

Yanqin Zhao[1], Mingkun Wu[2], Jiangping Mei[3], Wen Zhao[1] and Yan Jin[4]

[1]School of Mechanical Engineering, Yangzhou University, Yangzhou, China
[2]Institute of Mechanism Theory, Machine Dynamics and Robotics, RWTH Aachen University, Aachen, Germany
[3]Key Laboratory of Mechanism Theory and Equipment Design of Ministry of Education, Tianjin University, Tianjin, China
[4]School of Mechanical and Aerospace Engineering, Queen's University, Belfast, UK

Abstract

Due to the advantages of high stiffness, high precision, high load capacity and large workspace, hybrid robots are applicable to drilling and milling of complicated components with large sizes, for instance car panels. However, the difficulty in establishing an exact dynamic model and external disturbances affect the high accuracy control directly, which will decrease the machining accuracy and thereby affect the machining quality and efficiency of the system. Sliding mode control is an effective approach for high-order nonlinear dynamic systems since that it is very insensitive to disturbances and parameter variations. However, chattering may exist in traditional sliding mode control with fixed parameters, which results from a constant approaching speed. Besides, the approaching speed will affect the chattering strength directly. To solve these problems, a modified sliding mode controller with self-adaptive parameters is proposed to enhance the trajectory-tracking performance of a 5-degree-of-freedom hybrid robot. Firstly, the kinematic model of the robot is established. Then adopting the principle of virtual work, a rigid dynamic model of the robot is built. Based on the built dynamic model, a modified sliding mode control method is developed, of which the approaching speed is dependent on the system state. Finally, the sliding mode controller with self-adaptive parameters is created for a hybrid robot. The proposed sliding mode controller can achieve a rapid approaching speed and suppress chattering simultaneously. Simulation results demonstrate that the proposed modified sliding mode controller can achieve a comparatively accurate and smooth trajectory, which owns good robustness to external disturbances.

Corresponding author:
Yanqin Zhao, School of Mechanical Engineering, Yangzhou University, Yangzhou, 225009, China.
Email: zhaoyanqin_91@163.com

Keywords
Hybrid robot, modified sliding mode control, dynamics, trajectory tracking.

Introduction

Compared to serial kinematic machines (SKMs), parallel kinematic machines (PKMs) can achieve higher stiffness, higher load capacity and higher accuracy in theory. However, SKMs own higher flexibility. Therefore, hybrid robots, which are usually constructed by connecting a 2R manipulator to a PKM serially, have attracted much attention in recent years. For instance, the hybrid 5-degree-of-freedom (5-DOF) robots Tricept and Exechon have been applied in the drilling and milling of large-scale complex components successfully.[1,2] Considering the structure features of the two hybrid robots, Huang et al.[3,4] proposed a novel hybrid robot which is known as TriMule and the robot could be regarded as a plug-and-play functional module to construct different workstations. Some research related to performance evaluation, optimal design has been conducted in literature.[5–7] Despite of the mechanical structure with good performance, the control of a robot is crucial to the high efficiency and high precision machining.

In general, the control methods of a robot fall into kinematic control methods and dynamic control methods. Usually, the kinematic control methods are simple in design and fast in calculation, which have been widely applied in industry. The PID control and PD control are commonly used kinematic control methods.[8,9] Based on a simplified rigid dynamic model and in order to achieve good trajectory tracking performance for the high-speed parallel robot Delta, Wu et al.[10] established a nonlinear PD controller with nonlinear disturbance observer. Meanwhile, in order to enhance a controller's dynamic response and the anti-disturbance capacity, some intelligent algorithms have been added to the kinematic control methods. For example, Han et al.[11] presented a PID controller with fuzzy gain scheduling for a hybrid robot, which can suppress the control performance varying with robots' configurations effectively. In order to isolate vibration, Taghizadeh et al.[12] presented self-tuning PID controller based on neural network for the Stewart platform, which also showed good robustness against external disturbances. Lee et al.[13] proposed a hybrid PID-sliding mode control method, which was proved to own good transient response and trajectory tracking performance simultaneously.

The dynamic control methods are mainly divided into dynamic feedforward control methods[14,15] and computed torque control methods.[16,17] Usually, the dynamic feedforward control mainly consists of a closed-loop kinematic control system and a dynamic control system. The closed-loop kinematic control system takes control of the servo system of kinematic links directly, which will suppress the effects of dynamic factors that cannot be compensated by the dynamic control system simultaneously. Due to the good stability, high speed and good accuracy, the PID controller is usually applied to design the closed-loop kinematic control system. The dynamic control system helps reduce effects of nonlinear dynamic characteristics on a controller, which will improve a system's dynamic response. Therefore, the dynamic feedforward control owns the merits of the kinematic control methods and can improve dynamic response of a controller simultaneously. Furthermore, some intelligent algorithms have been introduced to the closed-loop kinematic control system to reduce effects of dynamic parameters on the control performance.[18–20]

However, the dynamic feedforward control methods with intelligent algorithms are difficult to be realized in practice for the instability of intelligent algorithms, large calculation load and high requirement for a controller.

The computed torque control methods usually rely on a system's rigid dynamic model, based on which the required driving force/torque for a desired motion trajectory is obtained and is used to control each kinematic link subsequently. Therefore, the computed torque control can achieve comparatively high control accuracy. It can be known that the control performance is strongly dependent on the rigid dynamic model's modeling accuracy. In addition, results show that the control performance of a computed torque method will also be influenced by nonlinear factors during the motion process.[21,22] Therefore, it is difficult for a simple computed torque method to obtain ideal control performance. Considering these issues, researchers improved the computed torque control method by introducing fuzzy control, adaptive control, and sliding mode control.[23–25] For example, adopting a fuzzy adaptive reaching law, Zhao et al.[26] proposed a discrete sliding mode controller for a 6-PRRS parallel robot, which can achieve good performance with smooth control actions and minimum reaching time. By mixing and extending adaptive fuzzy sliding mode control and adaptive fuzzy sliding mode observer methods, Navvabi et al.[27] designed a controller for the 6-DOF Stewart manipulator, which was composed of a fuzzy controller and a robust switching controller in parallel.

It is noted that parallel/hybrid mechanisms are typical nonlinear time-varying and strong coupling systems with multiple inputs and outputs, of which dynamic characteristics, the gravity and the equivalent inertia for instance, are strongly configuration-dependent. For these complex systems, it is hard to build exact dynamic models due to the following two main reasons. One is that it is hard to obtain the real inertia parameters and dimensional parameters of a parallel/hybrid mechanism after parts are fabricated and assembled. The other is that some terms are usually neglected for the convenience of derivation, such as friction. Besides, as machining tools, there are some external disturbances, such as cutting force. The mentioned issues above will decrease the control behavior of a hybrid robot working as a machine tool, which will thereby influence the working accuracy and working efficiency of the system. Sliding mode control is an effective approach which can deal with control of high-order nonlinear dynamic systems under various uncertainty conditions. Furthermore, sliding mode control is very insensitive to disturbances and parameter variations, thus an exact model is not required.[28,29]

Based on the above discussions, a sliding mode controller will be developed for the TriMule robot. It is noted that if a large model uncertain happens, large switching control is required to ensure a stable reaching phase, which will result in chattering and thereby affect the system's machining accuracy.[30] In order to avoid the undesirable system chattering and achieve a rapid approaching speed, a modified sliding mode controller with self-adaptive parameters dependent on the system state is presented in the paper. The advantage of this controller is parameters dependent on the system state, which will lead to the approaching speed adaptive to the system state thereby. To be specific, the presented sliding mode control method can ensure that if the system state gets far away from sliding surfaces, the system will approach to sliding surfaces at an exponential rate. Contrarily, when the system state gets near sliding surfaces, the system state's speed approaching to sliding surfaces will become smaller and smaller until 0, which thus can

suppress system chattering efficiently. Compared to traditional PID control, the proposed sliding mode control can achieve higher control behaviors for the system's dynamics is considered and it is not sensitive to disturbances. Compare to traditional sliding mode control with fixed parameters, chattering will be suppressed, thus can achieve a much smooth trajectory.

The remainder of this paper is constructed as follows. In Section 2, mechanical structure and kinematic model of the TriMule robot are introduced. Based on the principle of virtual work, the rigid dynamic model is built in Section 3, which will be used in the subsequent controller design. A modified sliding mode controller is presented in Section 4, of which the stability is proved by the Lyapunov functions. Based on a typical reference trajectory, the control performance of the presented sliding mode control is evaluated in Section 5. Finally, some conclusions are summarized.

Kinematic modeling of the hybrid robot

Structure of the TriMule robot

As can be seen from Figure 1, the hybrid robot is mainly constructed by a parallel kinematic machine module and a serial 2R kinematic machine module. There are four kinematic links in the parallel kinematic machine module, three of which are active while one is passive. Three active links are connected to the base through universal joints, while connected to the platform by spherical joints. The passive link is connected to the base by a universal joint while attached to the platform fixedly. The serial 2R kinematic machine module mainly consists of two perpendicular axes, i.e. A-axis and C-axis. Thus, the hybrid robot can achieve 5-DOF when driven by 5 servomotors independently,

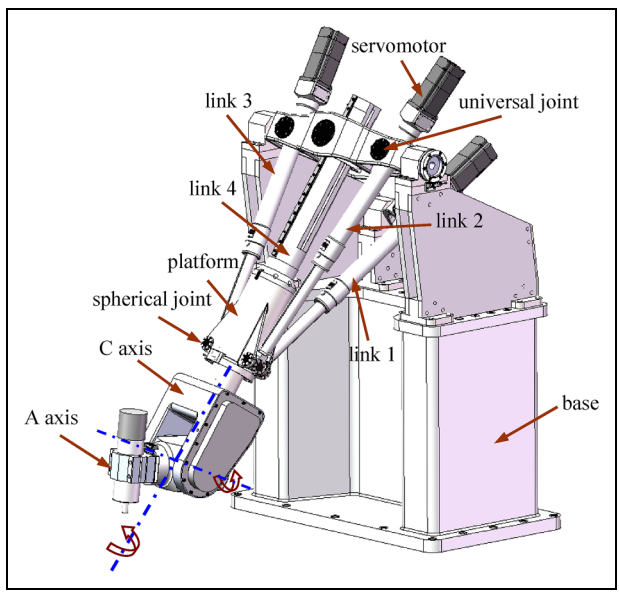

Figure 1. 3D model of the 5-DOF TriMule robot.

which can be used for drilling and milling of large-scale complex components, such as aircraft panels in the aerospace industry.

The schematic diagram of the robot is shown in Figure 2. For the convenience of derivation, the following coordinate frames are established. As shown, the geometrical center of the spherical joint in link j is denoted as A_j, while that of the universal joint in link j is denoted as B_j ($j = 1 \sim 4$). To be specific, A_4 is the attached point on the interface of link 4 and the platform. C_j denotes the end point of the jth link. The A axis and C axis intersect at point P. And the end point of the machine tool is expressed as C. The system's global coordinate frame B_4-xyz is built at the geometrical center of the universal joint B_4, with the x direction pointing from points B_4 to B_2 and the y direction vertically up to segment B_3B_2. Then the z direction could be decided definitely by the right-hand rule. The platform's coordinate frame A-$x_Ay_Az_A$ is built at the center of segment A_3A_2, of which the x_A direction is along the segment A_3A_2 and the z_A direction is vertical to the platform. Similarly, the y_A direction is determined through the right-hand rule. The body-fixed coordinate frame of the A axis is set at point C, where the x_c direction is along the rotation axis of the A axis while the z_c direction is along the machine tool's axis. Accordingly, the y_c axis can be determined through the right-hand rule. The coordinate frame of the ith link B_i-$x_iy_iz_i$ is established at point B_i, with the z_i direction along the ith link and the y_i direction along one of the rotation axes of the ith universal joint ($i = 1 \sim 3$). Accordingly, the x_i direction is decided through the right-hand rule.

The hybrid robot's kinematic model has been included in our previous work,[4] which will not be described hereby. Velocity and acceleration analysis will be built in this section, and will be applied in the subsequent dynamic modeling of the hybrid robot.

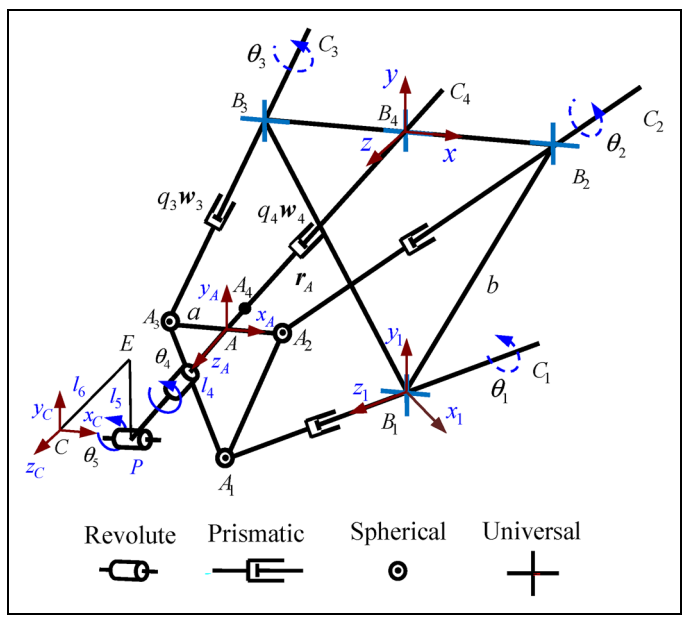

Figure 2. Schematic diagram of the TriMule robot.

Velocity analysis

In the system's global coordinate frame B_4-xyz, position of point A can be described as

$$\boldsymbol{r}_A = \boldsymbol{b}_i + q_i \boldsymbol{w}_i - \boldsymbol{a}_i, (i = 1 \sim 3) \tag{1}$$

$$\boldsymbol{r}_A = q_4 \boldsymbol{w}_4 \tag{2}$$

where $\boldsymbol{w}_i$ and q_i respectively represent the unit vector and length of the ith active link while $\boldsymbol{w}_4$ and q_4 express the unit vector and length of the passive link respectively. $\boldsymbol{a}_i$ and $\boldsymbol{b}_i$ are vectors in the system's global coordinate frame B_4-xyz pointing from A/B_4 to A_i/B_i, respectively.

Taking derivation of (1) and (2) leads to

$$\boldsymbol{v}_A = \dot{q}_i \boldsymbol{w}_i + q_i(\boldsymbol{\omega}_i \times \boldsymbol{w}_i) - \boldsymbol{\omega}_A \times \boldsymbol{a}_i, (i = 1 \sim 3) \tag{3}$$

$$\boldsymbol{v}_A = \dot{q}_4 \boldsymbol{w}_4 + q_4(\boldsymbol{\omega}_A \times \boldsymbol{w}_4) \tag{4}$$

where $\boldsymbol{v}_A$ and $\boldsymbol{\omega}_A$ express the linear velocity and angular velocity of the platform, respectively; $\dot{q}_i$ and $\boldsymbol{\omega}_i$ are the ith active link's feed speed and angular velocity, respectively.

In order to eliminate $\boldsymbol{\omega}_i$, one can take inner dot of (3) by $\boldsymbol{w}_i$ and obtain

$$\dot{q}_i = \boldsymbol{w}_i^T \boldsymbol{v}_A + (\boldsymbol{a}_i \times \boldsymbol{w}_i)^T \boldsymbol{\omega}_A, (i = 1 \sim 3) \tag{5}$$

Taking cross product of (4) by $\boldsymbol{w}_4$, the following relationship can be obtained

$$\boldsymbol{\omega}_A = \boldsymbol{J}_{\omega 4} \boldsymbol{v}_A \tag{6}$$

where $\boldsymbol{J}_{\omega 4} = \frac{1}{q_4}[\boldsymbol{w}_4\times]$, $[\boldsymbol{w}_4\times]$ denotes the skew matrix of $\boldsymbol{w}_4$.

Substituting (6) into (5), one can obtain

$$\dot{q}_i = \boldsymbol{w}_i^T \boldsymbol{v}_A - \frac{1}{q_4} \boldsymbol{w}_i^T \boldsymbol{w}_4 \boldsymbol{a}_i^T v_A, (i = 1 \sim 3) \tag{7}$$

One can rewrite (7) as

$$\boldsymbol{J} \boldsymbol{v}_A = \dot{\boldsymbol{q}} \tag{8}$$

where $\dot{\boldsymbol{q}} = (\dot{q}_1 \quad \dot{q}_2 \quad \dot{q}_3)^{\mathrm{T}}$, $\boldsymbol{J}$ denotes the Jacobian matrix of the system, which can be expressed as

$$\boldsymbol{J} = \begin{bmatrix} \boldsymbol{w}_1^{\mathrm{T}} \\ \boldsymbol{w}_2^{\mathrm{T}} \\ \boldsymbol{w}_3^{\mathrm{T}} \end{bmatrix} - \frac{1}{q_4} \begin{bmatrix} \boldsymbol{w}_1^{\mathrm{T}} \boldsymbol{w}_4 \boldsymbol{a}_1^{\mathrm{T}} \\ \boldsymbol{w}_2^{\mathrm{T}} \boldsymbol{w}_4 \boldsymbol{a}_2^{\mathrm{T}} \\ \boldsymbol{w}_3^{\mathrm{T}} \boldsymbol{w}_4 \boldsymbol{a}_3^{\mathrm{T}} \end{bmatrix}$$

.

One also can take inner dot of (4) by $\boldsymbol{w}_4$ and derive

$$\dot{q}_4 = \boldsymbol{J}_4 \boldsymbol{v}_A \tag{9}$$

where $\boldsymbol{J}_4 = \boldsymbol{w}_4^{\mathrm{T}}$.

To obtain the ith active link's angular velocity, one can take cross product of (3) by $\boldsymbol{w}_i$ as

$$\boldsymbol{\omega}_i = \frac{1}{q_i}(\boldsymbol{w}_i \times \boldsymbol{v}_A + (\boldsymbol{w}_i^{\mathrm{T}}\boldsymbol{a}_i)\boldsymbol{\omega}_A - \boldsymbol{a}_i(\boldsymbol{w}_i^{\mathrm{T}}\boldsymbol{\omega}_A)), (i = 1 \sim 3) \tag{10}$$

Substituting (6) into (10), the ith active link's angular velocity can be rewritten as

$$\boldsymbol{\omega}_i = \boldsymbol{J}_{\omega i}\boldsymbol{v}_A, (i = 1 \sim 3) \tag{11}$$

where $\boldsymbol{J}_{\omega i} = \frac{1}{q_i}([\boldsymbol{w}_i\times] + (\boldsymbol{w}_i^{\mathrm{T}}\boldsymbol{a}_i\boldsymbol{E}_{3\times3} - \boldsymbol{a}_i\boldsymbol{w}_i^{\mathrm{T}})\boldsymbol{J}_{\omega 4})$, and $[\boldsymbol{w}_i\times]$ denotes the skew matrix of $\boldsymbol{w}_i$.

Acceleration analysis

Taking derivation of (8) yields

$$\boldsymbol{J}\dot{\boldsymbol{v}}_A + \dot{\boldsymbol{J}}\boldsymbol{v}_A = \ddot{\boldsymbol{q}} \tag{12}$$

where $\ddot{\boldsymbol{q}} = (\ddot{q}_1 \quad \ddot{q}_2 \quad \ddot{q}_3)^{\mathrm{T}}$,

$$\dot{\boldsymbol{J}} = \begin{bmatrix} (\boldsymbol{\omega}_1 \times \boldsymbol{w}_1)^{\mathrm{T}} \\ (\boldsymbol{\omega}_2 \times \boldsymbol{w}_2)^{\mathrm{T}} \\ (\boldsymbol{\omega}_3 \times \boldsymbol{w}_3)^{\mathrm{T}} \end{bmatrix} + \frac{\dot{q}_4}{q_4^2}\begin{bmatrix} \boldsymbol{w}_1^{\mathrm{T}}\boldsymbol{w}_4\boldsymbol{a}_1^{\mathrm{T}} \\ \boldsymbol{w}_2^{\mathrm{T}}\boldsymbol{w}_4\boldsymbol{a}_2^{\mathrm{T}} \\ \boldsymbol{w}_3^{\mathrm{T}}\boldsymbol{w}_4\boldsymbol{a}_3^{\mathrm{T}} \end{bmatrix}$$
$$- \frac{1}{q_4}\begin{bmatrix} (\boldsymbol{\omega}_1 \times \boldsymbol{w}_1)^{\mathrm{T}}\boldsymbol{w}_4\boldsymbol{a}_1^{\mathrm{T}} + \boldsymbol{w}_1^{\mathrm{T}}(\boldsymbol{\omega}_{\mathrm{A}} \times \boldsymbol{w}_4)\boldsymbol{a}_1^{\mathrm{T}} + \boldsymbol{w}_1^{\mathrm{T}}\boldsymbol{w}_4(\boldsymbol{\omega}_{\mathrm{A}} \times \boldsymbol{a}_1)^{\mathrm{T}} \\ (\boldsymbol{\omega}_2 \times \boldsymbol{w}_2)^{\mathrm{T}}\boldsymbol{w}_4\boldsymbol{a}_2^{\mathrm{T}} + \boldsymbol{w}_2^{\mathrm{T}}(\boldsymbol{\omega}_{\mathrm{A}} \times \boldsymbol{w}_4)\boldsymbol{a}_2^{\mathrm{T}} + \boldsymbol{w}_2^{\mathrm{T}}\boldsymbol{w}_4(\boldsymbol{\omega}_{\mathrm{A}} \times \boldsymbol{a}_2)^{\mathrm{T}} \\ (\boldsymbol{\omega}_3 \times \boldsymbol{w}_3)^{\mathrm{T}}\boldsymbol{w}_4\boldsymbol{a}_3^{\mathrm{T}} + \boldsymbol{w}_3^{\mathrm{T}}(\boldsymbol{\omega}_{\mathrm{A}} \times \boldsymbol{w}_4)\boldsymbol{a}_3^{\mathrm{T}} + \boldsymbol{w}_3^{\mathrm{T}}\boldsymbol{w}_4(\boldsymbol{w}_{\mathrm{A}} \times \boldsymbol{a}_3)^{\mathrm{T}} \end{bmatrix}.$$

Taking derivation of (9), one can obtain

$$\ddot{\boldsymbol{q}}_4 = \boldsymbol{J}_4\dot{\boldsymbol{v}}_A + \dot{\boldsymbol{J}}_4\boldsymbol{v}_A \tag{13}$$

where $\dot{\boldsymbol{J}}_4 = (\boldsymbol{\omega}_A \times \boldsymbol{w}_4)^{\mathrm{T}}$.

Taking derivation of (6), the platform's angular acceleration can be obtained as

$$\dot{\boldsymbol{\omega}}_A = \boldsymbol{J}_{\omega 4}\dot{\boldsymbol{v}}_A + \dot{\boldsymbol{J}}_{\omega 4}\boldsymbol{v}_A \tag{14}$$

where $\dot{\boldsymbol{J}}_{\omega 4} = -\frac{\dot{q}_4}{q_4^2}[w_4\times] + \frac{1}{q_4}[(\boldsymbol{\omega}_A \times \boldsymbol{w}_4)\times]$, $[(\boldsymbol{\omega}_A \times \boldsymbol{w}_4)\times]$ expresses the skew matrix of $\boldsymbol{\omega}_A \times \boldsymbol{w}_4$.

Taking derivation of (11), the ith active link's angular acceleration can be expressed as

$$\dot{\boldsymbol{\omega}}_i = \boldsymbol{J}_{\omega i}\dot{\boldsymbol{v}}_A + \dot{\boldsymbol{J}}_{\omega i}\boldsymbol{v}_A, (i = 1 \sim 3) \tag{15}$$

where

$$\dot{\boldsymbol{J}}_{\omega i} = -\frac{\dot{q}_i}{q_i^2}([\boldsymbol{w}_i\times] + (\boldsymbol{w}_i^{\mathrm{T}}\boldsymbol{a}_i\boldsymbol{E}_{3\times3} - \boldsymbol{a}_i\boldsymbol{w}_i^{\mathrm{T}})\boldsymbol{J}_{\omega 4})$$
$$+ \frac{1}{q_i}([(\boldsymbol{\omega}_i \times \boldsymbol{w}_i)\times] + (\boldsymbol{w}_i^{\mathrm{T}}\boldsymbol{a}_i\boldsymbol{E}_{3\times3} - \boldsymbol{a}_i\boldsymbol{w}_i^{\mathrm{T}})\dot{\boldsymbol{J}}_{\omega 4} + ((\boldsymbol{\omega}_i \times \boldsymbol{w}_i)^{\mathrm{T}}\boldsymbol{a}_i\boldsymbol{E}_{3\times3}$$
$$+ \boldsymbol{w}_i^{\mathrm{T}}(\boldsymbol{\omega}_A \times \boldsymbol{a}_i)\boldsymbol{E}_{3\times3} - (\boldsymbol{\omega}_A \times \boldsymbol{a}_i)\boldsymbol{w}_i^{\mathrm{T}} - \boldsymbol{a}_i(\boldsymbol{\omega}_i \times \boldsymbol{w}_i)^{\mathrm{T}})\boldsymbol{J}_{\omega 4})$$

Dynamic modeling of the hybrid robot

In this section, the rigid dynamic model of the hybrid robot will be built according to the principle of virtual work. For the convenience of derivation, the following are assumed.

1. All the joints are assumed to be ideal.
2. Inertia of the universal joints connected to the base is neglected.
3. All the limbs are axisymmetric rigid bodies, of which inertia matrices in their body-fixed coordinate systems are diagonal.
4. The 2R serial mechanism module is regarded as a lumped mass model attached to the platform. However, inertia of the lumped mass model is configuration-dependent.

System component decomposition and kinematic description

The structure diagram of passive and active links is depicted in Figure 3. According to the above assumptions and motion characteristics of the four links, one can divide the passive and active links into different motion parts, respectively. The passive link can be regarded as a component in general rigid body motion, of which the mass center is D_4. As shown, the passive link is divided into three components named S_i, S'_{i1} and S'_{i2}. The component S_i is composed of a stretchable limb and a ball nut, which is in general rigid body motion and its mass center is D_i. The component S'_{i1} mainly consists of a servomotor and an oscillating link, which is in fixed point rotation motion about the two axes of the universal joint. The component S'_{i2} mainly consists of a servomotor motor, a coupling and a ball screw. In addition to the fixed-point rotation motion together with component S'_{i1}, the

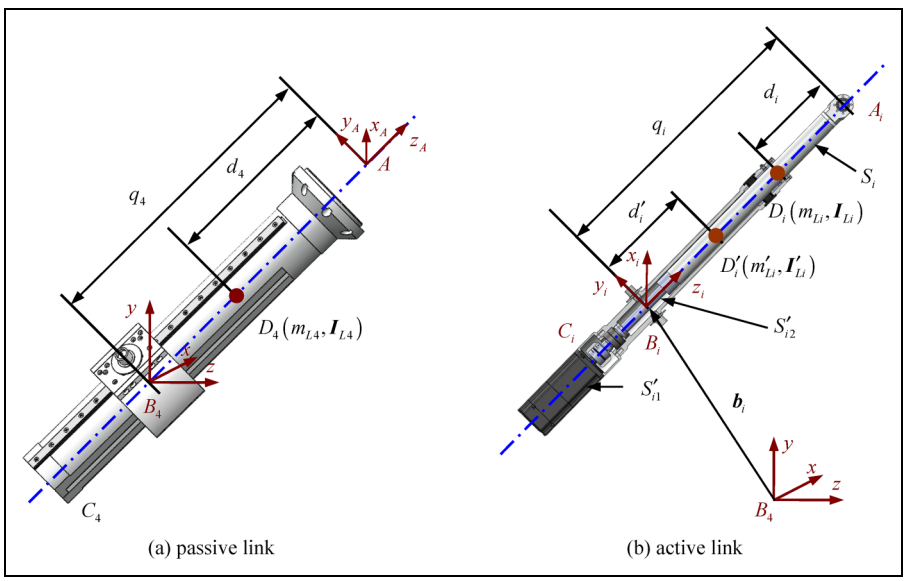

Figure 3. Structure diagram of passive and active links in TriMule.

component S'_{i2} also rotates about its axis. As there is no relative motion of component S'_{i1} with respect to component S'_{i2}, when the two components are considered as a component S'_i, its mass center is unchangeable and defined as D'_i.

In the global coordinate frame B_4-xyz, position vector of the mass center D_4 can be denoted as

$$\boldsymbol{d}_4 = (q_4 - d_4)\boldsymbol{w}_4 \tag{16}$$

where d_4 is the distance between the mass center D_4 and point A.

Taking derivation of (16), one can obtain the velocity of the mass center D_4 as

$$\boldsymbol{v}_{D4} = \boldsymbol{J}_{v4}\boldsymbol{v}_A \tag{17}$$

where $\boldsymbol{J}_{v4} = \boldsymbol{w}_4\boldsymbol{J}_4 + (d_4 - q_4)[\boldsymbol{w}_4\times]\boldsymbol{J}_{\omega 4}$.

Accordingly, in the global coordinate frame B_4-xyz, position vector of the mass center D_i can be expressed as

$$\boldsymbol{d}_i = \boldsymbol{b}_i + (q_i - d_i)\boldsymbol{w}_i, (i = 1 \sim 3) \tag{18}$$

where d_i denotes the distance between point A_i and the mass center D_i.

Taking derivation of (18), one can obtain the velocity of the mass center D_i as

$$\boldsymbol{v}_{Di} = \boldsymbol{J}_{vi}\boldsymbol{v}_A, (i = 1 \sim 3) \tag{19}$$

where $\boldsymbol{J}_{vi} = \boldsymbol{w}_i\boldsymbol{J}_i + (d_i - q_i)[\boldsymbol{w}_i\times]\boldsymbol{J}_{\omega i}$; $\boldsymbol{J}_i$ denotes the ith row of the Jacobian matrix $\boldsymbol{J}$.

Taking derivation of (17) and (19), one can obtain the acceleration of the mass center D_j as

$$\dot{\boldsymbol{v}}_{Dj} = \boldsymbol{J}_{vj}\dot{\boldsymbol{v}}_A + \dot{\boldsymbol{J}}_{vj}\boldsymbol{v}_A, (j = 1 \sim 4) \tag{20}$$

where

$\boldsymbol{J}_{vj} = (\boldsymbol{\omega}_j \times \boldsymbol{w}_j)\boldsymbol{J}_j + \boldsymbol{w}_j\dot{\boldsymbol{J}}_j - \dot{q}_j[\boldsymbol{w}_j\times]\boldsymbol{J}_{\omega j} + (d_j - q_j)[(\boldsymbol{\omega}_j \times \boldsymbol{w}_j)\times]\boldsymbol{J}_{\omega j} + (d_j - q_j)[\boldsymbol{w}_j\times]\dot{\boldsymbol{J}}_{\omega j}$.

Based on the velocity superposition principle, the absolute angular velocity of component S'_{i2} can be obtained in the global coordinate frame B_4-xyz as

$$\boldsymbol{\omega}_{mi} = \boldsymbol{\omega}_i + \frac{2\pi\dot{q}_i}{p}\boldsymbol{w}_i = \boldsymbol{J}_{\omega mi}\boldsymbol{v}_A, (i = 1 \sim 3) \tag{21}$$

where $\boldsymbol{J}_{\omega mi} = \boldsymbol{J}_{\omega i} + \frac{2\pi w_i}{p}\boldsymbol{J}_i$; p denotes the pitch of the ball screw.

Taking derivation of (21), one can obtain the angular acceleration of component S'_{i2} as

$$\dot{\boldsymbol{\omega}}_{mi} = \boldsymbol{J}_{\omega mi}\dot{\boldsymbol{v}}_A + \dot{\boldsymbol{J}}_{\omega mi}\boldsymbol{v}_A, (i = 1 \sim 3) \tag{22}$$

where $\dot{\boldsymbol{J}}_{\omega mi} = \dot{\boldsymbol{J}}_{\omega i} + \frac{2\pi w_i}{p}\dot{\boldsymbol{J}}_i + \frac{2\pi(\omega_{Li}\times w_i)}{p}\boldsymbol{J}_i$.

Virtual work of the active link

The gravity and inertial force applying at the mass center D_i of component S_i is denoted as

$$\boldsymbol{F}_i = m_{Li}\boldsymbol{g} - m_{Li}\dot{\boldsymbol{v}}_{Di}, (i = 1 \sim 3) \tag{23}$$

where m_{Li} expresses the mass of component S_i; $\boldsymbol{g}$ is the gravitational acceleration vector.

The inertial moment of component S_i can be expressed as

$$\boldsymbol{T}_i = -\boldsymbol{I}_{Li}\dot{\boldsymbol{\omega}}_i - \boldsymbol{\omega}_i \times \boldsymbol{I}_{Li}\boldsymbol{\omega}_i = -\boldsymbol{I}_{Li}\dot{\boldsymbol{\omega}}_i, (i = 1 \sim 3) \tag{24}$$

where $\boldsymbol{I}_{Li}$ expresses the inertia matrix of the component S_i in the global coordinate frame.

The moment of the gravity about point B_i of component S_i' can be denoted as

$$\boldsymbol{T}'_i = d_i'\boldsymbol{w}_i \times m_{Li}'\boldsymbol{g}, (i = 1 \sim 3) \tag{25}$$

where d_i' denotes the distance between point B_i and the mass center D_i'; m_{Li}' expresses the mass of component S_i'.

The inertial moment of component S_{i1}' measured about point B_i can be expressed as

$$\boldsymbol{T}_{i1}' = -\boldsymbol{I}_{L1i}'\dot{\boldsymbol{\omega}}_i - \boldsymbol{\omega}_i \times \boldsymbol{I}_{L1i}'\boldsymbol{\omega}_i = -\boldsymbol{I}_{L1i}'\dot{\boldsymbol{\omega}}_i, (i = 1 \sim 3) \tag{26}$$

where $\boldsymbol{I}'_{L1i}$ denotes the inertia matrix of the component S_{i1}' about point B_i measured in the global coordinate system B_4-xyz.

The inertial moment of component S_{i2}' measured about point B_i can be expressed as

$$\boldsymbol{T}_{i2}' = -\boldsymbol{I}_{mi}\dot{\boldsymbol{\omega}}_{mi} - \boldsymbol{\omega}_{mi} \times \boldsymbol{I}_{mi}\boldsymbol{\omega}_{mi}, (i = 1 \sim 3) \tag{27}$$

where $\boldsymbol{I}_{mi}$ expresses the inertia matrix of the component S_{i2}' about point B_i measured in the coordinate system B_4-xyz.

The virtual work conducted by the inertial moment of component S_{i2}' can be expressed as

$$\delta\boldsymbol{\omega}_{mi}^T\boldsymbol{T}_{i2}' = -\delta\boldsymbol{\omega}_{Li}^T\boldsymbol{I}_{mi}\dot{\boldsymbol{\omega}}_i - (\frac{2\pi}{p})^2\boldsymbol{\omega}_{mi}\ddot{\boldsymbol{q}}_i\delta\dot{q}_i, (i = 1 \sim 3) \tag{28}$$

where I_m denotes the moment of inertia about its axis.

Thus, the virtual work conducted by the active limb can be derived as

$$\begin{aligned}\delta W_i = {} & \delta\boldsymbol{v}_{Di}^T(m_{Li}\boldsymbol{g} - m_{Li}\dot{\boldsymbol{v}}_{Di}) - \delta\boldsymbol{\omega}_i^T(\boldsymbol{I}_{Li} + \boldsymbol{I}_{Li}')\dot{\omega}_i - \left(\frac{2\pi}{p}\right)^2\boldsymbol{I}_m\ddot{\boldsymbol{q}}_i\delta\dot{q}_i \\ & + \boldsymbol{\omega}_i^T(d_i'\boldsymbol{w}_i \times m_i'\boldsymbol{g}), (i = 1 \sim 3)\end{aligned} \tag{29}$$

where $\boldsymbol{I}'_{Li}$ denotes the inertia matrix of the component S_i' about point B_i.

Virtual work of the passive link

The gravity and inertial force applying at the passive link's mass center D_4 can be expressed as

$$\boldsymbol{F}_4 = m_{L4}\boldsymbol{g} - m_{L4}\dot{\boldsymbol{v}}_{D4} \tag{30}$$

where m_{L4} denotes the passive link's mass.

The inertial moment of the passive link can be expressed as

$$\boldsymbol{T}_4 = -I_{L4}\dot{\boldsymbol{\omega}}_A - \boldsymbol{\omega}_A \times \boldsymbol{I}_{L4}\boldsymbol{\omega}_A = -\boldsymbol{I}_{L4}\dot{\boldsymbol{\omega}}_A \tag{31}$$

where $\boldsymbol{I}_{L4}$ expresses the passive link's inertia matrix measured in the global coordinate system B_4-xyz.

Thus, the virtual work conducted by the passive link can be expressed as

$$\delta W_4 = \delta \boldsymbol{v}_{D4}^T (m_{L4}\boldsymbol{g} - m_{L4}\dot{\boldsymbol{v}}_{D4}) - \delta \boldsymbol{\omega}_A^T \boldsymbol{I}_{L4}\dot{\boldsymbol{\omega}}_A \tag{32}$$

Virtual work of the platform

The gravity and inertial force applying at the platform's mass center A can be denoted as

$$\boldsymbol{F}_{\mathrm{P}} = m_{\mathrm{P}}\boldsymbol{g} - m_P\dot{\boldsymbol{v}}_A \tag{33}$$

where m_{P} denotes the platform's mass.

The platform's inertial moment can be obtained as

$$\boldsymbol{T}_P = -\boldsymbol{I}_{\mathrm{P}}\dot{\boldsymbol{\omega}}_A - \boldsymbol{\omega}_A \times \boldsymbol{I}_{\mathrm{P}}\boldsymbol{\omega}_A = -\boldsymbol{I}_{\mathrm{P}}\dot{\boldsymbol{\omega}}_A \tag{34}$$

where $\boldsymbol{I}_{\mathrm{P}}$ expresses the platform's inertia matrix in the global coordinate system B_4-xyz.

Therefore, the platform's virtual work can be expressed as

$$\delta W_{\mathrm{P}} = \delta \boldsymbol{v}_A^T (m_{\mathrm{P}}\boldsymbol{g} - m_P\dot{\boldsymbol{v}}_A) - \delta \boldsymbol{\omega}_A^T \boldsymbol{I}_{\mathrm{P}}\dot{\boldsymbol{\omega}}_A \tag{35}$$

Rigid dynamic model of the system

Supposing that the driven force is $\boldsymbol{f} = \begin{pmatrix} f_1 & f_2 & f_3 \end{pmatrix}^{\mathrm{T}}$, there exists

$$\delta W_{\mathrm{P}} + \delta \dot{\boldsymbol{q}}^T \boldsymbol{f} + \sum\nolimits_{j=1}^{4} \delta W_j = 0 \tag{36}$$

Based on (6), (8), (11), (17) and (19), the following can be obtained

$$\delta \boldsymbol{\omega}_A = \boldsymbol{J}_{\omega 4}\delta \boldsymbol{v}_A,\ \delta \dot{\boldsymbol{q}} = \boldsymbol{J}\delta \boldsymbol{v}_A,\ \delta \boldsymbol{\omega}_{Lj} = \boldsymbol{J}_{\omega j}\delta \boldsymbol{v}_A,\ \delta \boldsymbol{v}_{Dj} = \boldsymbol{J}_{vj}\delta \boldsymbol{v}_A,\ (j = 1 \sim 4) \tag{37}$$

Substituting (37) into (36), one can derive the rigid dynamic model of the system as

$$\boldsymbol{f} = \boldsymbol{D}\dot{\boldsymbol{v}}_A + \boldsymbol{H}\boldsymbol{v}_A + \boldsymbol{G} \tag{38}$$

where $\boldsymbol{D} = \boldsymbol{M}\boldsymbol{J}$, $\boldsymbol{M} = \boldsymbol{J}^{-\mathrm{T}}\Big(m_{\mathrm{P}}\boldsymbol{E} + \boldsymbol{J}_{\omega 4}^{\mathrm{T}}\boldsymbol{I}_{\mathrm{P}}\boldsymbol{J}_{\omega 4} + \sum_{j=1}^{4}(\boldsymbol{J}_{vj}^{\mathrm{T}} m_{Lj}\boldsymbol{J}_{vj} + \boldsymbol{J}_{\omega j}^{\mathrm{T}}\boldsymbol{I}_{sj}\boldsymbol{J}_{\omega j})\Big)\boldsymbol{J}^{-1} + \boldsymbol{M}_m$,

$\boldsymbol{H} = \boldsymbol{J}^{-T}(\boldsymbol{J}_{\omega 4}^T\boldsymbol{I}_P\dot{\boldsymbol{J}}_{\omega 4} + \sum_{j=1}^{4}(\boldsymbol{J}_{vj}^T m_{Lj}\dot{\boldsymbol{J}}_{vj} + \boldsymbol{J}_{\omega j}^T\boldsymbol{I}_{sj}\dot{\boldsymbol{J}}_{\omega j}) + \boldsymbol{J}^T M_m\dot{\boldsymbol{J}})$, $\quad \boldsymbol{M}_m = \mathrm{diag}\Big[(\frac{2\pi}{p})^2 I_m\Big]$,

$\boldsymbol{G} = -\boldsymbol{J}^{-\mathrm{T}}\Big(m_{\mathrm{P}}\boldsymbol{g} + \sum_{j=1}^{4} m_{Lj}\boldsymbol{J}_{vj}^{\mathrm{T}}\boldsymbol{g} + \sum_{i=1}^{3}\boldsymbol{J}_{\omega i}^{\mathrm{T}}(d'_i\boldsymbol{w}_i \times m'_{Li}\boldsymbol{g})\Big)$,

$$\boldsymbol{I}_{sj} = \begin{cases} \boldsymbol{I}_{Lj} + \boldsymbol{I}'_{Lj}, j = 1 \sim 3 \\ \boldsymbol{I}_{L4}, j = 4 \end{cases}.$$

Therefore, the driven torque can be obtained as

$$\boldsymbol{\tau} = \frac{p}{2\pi}\boldsymbol{f} \tag{39}$$

Sliding mode control model

Traditional sliding mode controller

Considering uncertain factors, the driven torque can be expressed in the joint space as

$$\boldsymbol{\tau} = \boldsymbol{M}_J\ddot{\boldsymbol{\theta}} + \boldsymbol{C}_J\dot{\boldsymbol{\theta}} + \boldsymbol{G}_J + \boldsymbol{T}_D \tag{40}$$

where $\boldsymbol{M}_J$, $\boldsymbol{C}_J$ and $\boldsymbol{G}_J$ are inertia, Coriolis and centrifugal, and gravity matrices in the joint space respectively. $\boldsymbol{T}_D$ denotes the uncertain term of the system; $\dot{\boldsymbol{\theta}}$ and $\ddot{\boldsymbol{\theta}}$ express active joints' velocity vector and acceleration vector, respectively.

In order to facilitate the design of controller, the following assumption is made

$$\begin{pmatrix} -T_1 & -T_2 & -T_3 \end{pmatrix}^{\mathrm{T}} < \boldsymbol{M}_J^{-1}\boldsymbol{T}_D < \begin{pmatrix} T_1 & T_2 & T_3 \end{pmatrix}^{\mathrm{T}} \tag{41}$$

where T_1, T_2 and T_3 are positive constants.

Sliding mode control owns the advantages of fast response, strong robustness, and good adaptability especially for non-linear systems. A sliding surfaces vector is defined hereby as

$$\boldsymbol{s} = \boldsymbol{C}\boldsymbol{e} + \dot{\boldsymbol{e}} \tag{42}$$

$$\boldsymbol{e} = \begin{pmatrix} e_1 & e_2 & e_3 \end{pmatrix}^{\mathrm{T}} = \boldsymbol{\theta} - \boldsymbol{\theta}^r \tag{43}$$

where $\boldsymbol{C}$ denotes a positive constant diagonal matrix which is defined as $\boldsymbol{C} = \mathrm{diag}[\, C_1 \quad C_2 \quad C_3 \,]$; $\boldsymbol{\theta}$ and $\boldsymbol{\theta}^r$ denote the position vector and command position vector of active joints, respectively; $\boldsymbol{e}$ and $\dot{\boldsymbol{e}}$ denote the position error vector and velocity error vector of active joints, respectively.

Based on (42), the following exponential reaching law is adopted

$$\dot{\boldsymbol{s}} = -\boldsymbol{\varepsilon}\mathrm{sgn}(\boldsymbol{s}) - \boldsymbol{k}\boldsymbol{s} \tag{44}$$

where $\boldsymbol{\epsilon}$ and $\boldsymbol{k}$ are positive constant diagonal matrices which are defined by $\boldsymbol{\varepsilon} = \mathrm{diag}[\, \varepsilon_1 \quad \varepsilon_2 \quad \varepsilon_3 \,]$ and

$\boldsymbol{k} = \mathrm{diag}[\, k_1 \quad k_2 \quad k_3 \,]$, respectively .

Taking derivation of (42), one can obtain

$$\dot{\boldsymbol{s}} = \boldsymbol{C}\dot{\boldsymbol{e}} + \ddot{\boldsymbol{e}} \tag{45}$$

Combining (44) and (45), the following can be derived

$$\ddot{\boldsymbol{e}} = -\boldsymbol{\varepsilon}\mathrm{sgn}(\boldsymbol{s}) - \boldsymbol{k}\boldsymbol{s} - \boldsymbol{C}\dot{\boldsymbol{e}} \tag{46}$$

There exists

$$\ddot{\boldsymbol{e}} = \ddot{\boldsymbol{\theta}} - \ddot{\boldsymbol{\theta}}^r \tag{47}$$

Substituting (42) and (46) into (40) and neglecting the uncertain term, one can derive the control law of the system expressed as

$$\boldsymbol{\tau}_{RDC} = M_J\ddot{\boldsymbol{\theta}}^{\mathrm{r}} + M_J[-\boldsymbol{\varepsilon}\mathrm{sgn}(\boldsymbol{s}) - \boldsymbol{k}\boldsymbol{s} - \boldsymbol{C}\dot{\boldsymbol{e}}] + \boldsymbol{C}_J\dot{\boldsymbol{\theta}} + \boldsymbol{G}_J \tag{48}$$

Modified sliding mode controller

As can be seen from (44), when the system state gets far away from sliding surfaces, i.e. the sliding surfaces vector $\boldsymbol{s} \to \infty$, effects of parameters $\boldsymbol{\epsilon}$ on the above reaching law in (44) can be neglected. In this case, the reaching law in (44) can be simplified as

$$\dot{\boldsymbol{s}} = -\boldsymbol{k}\boldsymbol{s} \tag{49}$$

Based on (49), one can derive

$$\boldsymbol{s} = \mathrm{diag}[\, s_1(0) \quad s_2(0) \quad s_3(0)\,]\left(e^{-k_1 t} \quad e^{-k_2 t} \quad e^{-k_3 t}\right)^{\mathrm{T}} \tag{50}$$

As can be seen from the above equation, with the increment of time, the sliding surfaces vector $\boldsymbol{s}$ approaches to $\mathbf{0}$ exponentially. That is to say, the system state approaches to sliding surfaces at an exponential rate.

Conversely, when the system state gets near sliding surfaces, i.e. the sliding surfaces vector $\boldsymbol{s} \to \mathbf{0}$, effects of parameters $\boldsymbol{k}$ on the reaching law in (44) becomes weak. In this case, the reaching law in (44) could be simplified into

$$\dot{\boldsymbol{s}} = -\boldsymbol{\epsilon}\mathrm{sgn}(\boldsymbol{s}) \tag{51}$$

As can be seen from (51), the system state approaches to sliding surfaces at a constant rate, which will cause the system chattering thereby. In addition, the approaching speed will affect the chattering strength directly.

Therefore, in order to achieve a rapid approaching speed and suppress chattering simultaneously, $\boldsymbol{\epsilon}$ can be designed as self-adaptive parameters dependent on the system state as

$$\boldsymbol{\epsilon} = \mathrm{diag}\left(\frac{h_1|e_1|}{1+|e_1|}e^{-\delta_1|s_1|} \quad \frac{h_2|e_2|}{1+|e_2|}e^{-\delta_2|s_2|} \quad \frac{h_3|e_3|}{1+|e_3|}e^{-\delta_3|s_3|}\right) \tag{52}$$

where $h_1 > 0,\ h_2 > 0,\ h_3 > 0,\ \delta_1 > 0,\ \delta_2 > 0,\ \delta_3 > 0$.

As can be seen from (52), when the system state is far away from sliding surfaces, i.e. $|s_i| \to \infty$, $\varepsilon_i \to 0$, the system will approach to sliding surfaces at an exponential rate as shown in (49). When the system state is near sliding surfaces, i.e. $|s_i| \to 0$, $\varepsilon_i \to \frac{h_i|e_i|}{1+|e_i|}$. With the decrement of position error e_i, $\dot{s}_i$ will approach to 0. That is, the system state's approaching speed to sliding surfaces will become smaller and smaller until 0, which thus can suppress the system chattering efficiently. Based on the above description, the modified sliding model controller's control law can be expressed as (48) shown.

A control framework is illustrated in Figure 4. Herein, x, y, z, α and β are configurations of the tool center point and tool orientation; θ_k^r, $\dot{\theta}_k^r$ and $\ddot{\theta}_k^r$ denote the command position, command velocity, and command acceleration of an active joint, respectively; $\theta_k, \dot{\theta}_k$ and denote the position, velocity and acceleration of an active joint, respectively; τ_k expresses the driving torque of an active joint. As shown, trajectory planning should be conducted first in the local coordinate frame, from which configurations of the tool center point and tool orientation with respect to time can be obtained. Then the command of five active joints, including the command position, command velocity and command acceleration, should be calculated through the inverse kinematics algorithm. Since structure of

the 2R serial mechanism module is comparatively simple, a PID controller is adopted. For the parallel mechanism module, the proposed modified sliding mode controller is designed for each active joint especially. The terminal device could be desired to move along its reference trajectory by driving all the active joints to follow their command. In order to simulate the real working environment, some disturbance could be added to the control system.

Stability analysis

In order to analyze the proposed modified sliding mode controller's stability, a Lyapunov function vector is defined

$$\boldsymbol{V} = \frac{1}{2}\mathrm{diag}[\, s_1 \quad s_2 \quad s_3 \,]\boldsymbol{s} \tag{53}$$

Taking derivation of (53) yields

$$\dot{\boldsymbol{V}} = \mathrm{diag}[\, s_1 \quad s_2 \quad s_3 \,]\dot{\boldsymbol{s}} = \mathrm{diag}[\, s_1 \quad s_2 \quad s_3 \,](\boldsymbol{C}\dot{\boldsymbol{e}} + \ddot{\boldsymbol{e}}) \tag{54}$$

Substituting (40) into (54) leads to

$$\dot{\boldsymbol{V}} = \mathrm{diag}[\, s_1 \quad s_2 \quad s_3 \,](\boldsymbol{C}\dot{\boldsymbol{e}} + \boldsymbol{M}_J^{-1}\boldsymbol{\tau} - \boldsymbol{M}_J^{-1}C_J\dot{\boldsymbol{\theta}} - \boldsymbol{M}_J^{-1}\boldsymbol{G}_J - \boldsymbol{M}_J^{-1}\boldsymbol{T}_D - \ddot{\boldsymbol{\theta}}^r) \tag{55}$$

Substituting (48) into (55), the first-order derivation of the above Lyapunov function vector can be expressed as

$$\dot{\boldsymbol{V}} = diag[\, s_1 \quad s_2 \quad s_3 \,](-\boldsymbol{M}_J^{-1}\boldsymbol{T}_D - \boldsymbol{\varepsilon}\mathrm{sgn}(\boldsymbol{s}) - \boldsymbol{k}\boldsymbol{s}) \tag{56}$$

Based on (56), the first-order derivation of the Lyapunov function in the 1st active link can be derived as

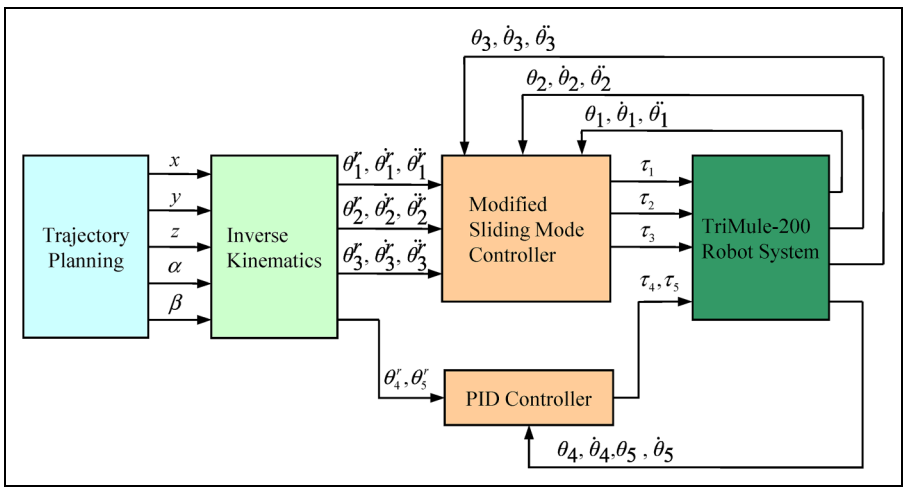

Figure 4. Block diagram of the modified sliding mode controller.

$$\dot{v}_1 = -s_1\Delta_1 - \frac{h_1|e_1|}{1+|e_1|}e^{-\delta_1|s_1|}\mathrm{sgn}(s_1)s_1 - k_1 s_1 s_1$$
$$\leq |s_1|T_1 - \frac{h_1|e_1|}{1+|e_1|}e^{-\delta_1|s_1|}|s_1| - k_1|s_1||s_1| \leq |s_1|T_1 - k_1|s_1||s_1| \quad (57)$$

As can be seen from (57), if $|s_1|T_1 - k_1|s_1||s_1| \leq 0$, i.e. $k_1|s_1| \geq T_1$, $\dot{v}_1 \leq 0$, which demonstrates that the proposed modified sliding model controller is stable. However, if $k_1|s_1| < T_1$, the sliding layer may appear due to the uncertainty of the model and the unconsidered modeling factors, which will cause the systems' steady-state error further. In this case, one can increase k_1 to narrow the width of sliding layer, which will decrease the system's steady-state error as much as possible. Stability of the 2nd and 3rd active links can be analyzed following the same procedure. Therefore, the control law in (48) can guarantee the closed-loop stability of the three motors.

Case study

The main geometrical parameters and mass parameters of the hybrid robot are listed respectively in Tables 1 and 2. It should be worth noting that inertia parameters listed in Table 2 are measured in each part's body-fixed coordinate system.

Description of a specified path

In order to facilitate analysis, a typical "S" path is shown in Figure 5. P_s and P_e denote the starting point and ending point of the path. P_2, P_3 and P_4 are tangent points of every two adjacent arcs. P_3-$x'z'$ denotes the path body-fixed coordinate frame, with the x' direction parallel to the x direction. R_1 and R_2 denote radii of arcs P_sP_2 and P_4P_e, and arcs P_2P_3 and P_3P_4 respectively. A machining angle is defined as γ, which describes the rotation angle of the local coordinate frame P_3-$x'z'$ about the x' axis with respect to the global coordinate frame B_4-xyz. It is noted that the tool axis should be kept perpendicular to the path plane.

As a result, position of a point in the specified "S" path can be described in B_4-xyz as

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\gamma & -\sin\gamma \\ 0 & \sin\gamma & \cos\gamma \end{bmatrix} \begin{pmatrix} x' \\ 0 \\ z' \end{pmatrix} + \begin{pmatrix} x_3 \\ y_3 \\ z_3 \end{pmatrix} \quad (58)$$

where x' and z' are coordinates of a point in the local coordinate frame P_3-$x'z'$ while x_3, y_3, and z_3 are those of the point P_3 expressed in the global coordinate frame B_4-xyz.

As could be known from the above analysis, position of the typical "S" path can be entirely decided by the machining angle γ and the position of point P_3. In order to keep the typical "S" path in the robot's workspace, point P_3 is fixed in the center of the robot's task workspace.

To facilitate the following analysis, the motion time of arcs P_sP_2 and P_4P_e are set to be the same, which is denoted as T_1. Similarly, the motion time of arcs P_2P_3 and P_3P_4 are set to be the same and denoted as T_2. Therefore, the required total motion time is $2(T_1 + T_2)$. Coordinates of key points on the typical path are shown in Table 3. Herein, x_1 denotes the coordinate of point P_s in the x' direction; x_2 and z_2 are coordinates of point P_2 in x' and z'

Table 1. The hybrid robot's geometrical parameters (unit: mm).

radius of the platform a	75	distance between points A and P l_4	283.5
length of the base side b	334	distance between points P and E l_5	147
stroke of the platform ss	200	distance between points E and C l_6	99

Table 2. The hybrid robot's mass parameters (units: kg, kg·m^2).

	Mass	Inertia in the x' axis	Inertia in the y' axis	Inertia in the z' axis
oscillating link	4.8411	0.1366	0.1366	0.0021
stretchable link	1.2675	0.0181	0.0181	0.0005
passive link	11.3264	0.5234	0.5234	0.0242
platform	5.7433	0.0332	0.0332	0.0368
A axis	5.5657	0.0448	0.0103	0.0423
C axis	9.8016	0.3937	0.3937	0.0498

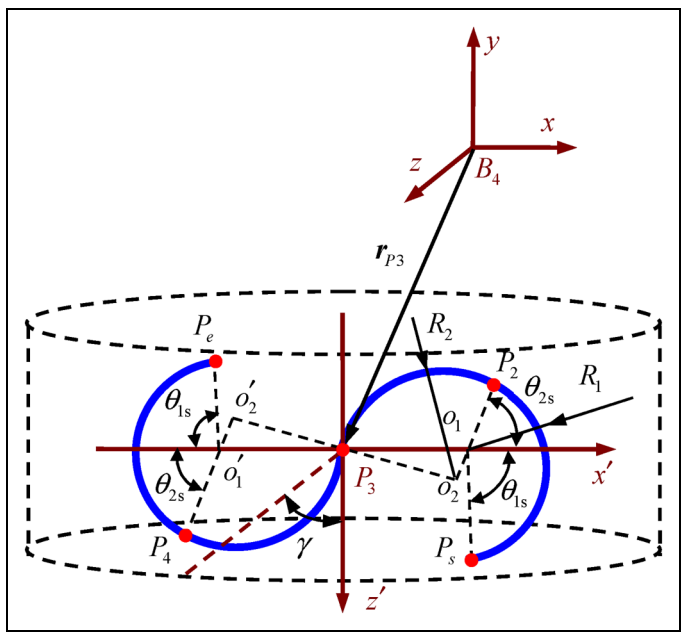

Figure 5. A typical path defined in a local coordinate frame.

directions respectively; θ_{1s} and θ_{2s} denote angles between o_1P_s, o_2P_2 and the x' direction, respectively.

For the above typical path, set $T_1 = 4.23$ s, $T_2 = 3.77$ s and $\gamma = -9.18°$. Meanwhile, the 7-order B-spline is adopted to conduct trajectory planning. This is the optimal result when adopting the 7-order B-spline by considering vibration errors and energy consumption simultaneously, which will not be analyzed in this paper.

Table 3. Parameters of the key points on trajectory in the local coordinate system.

Typical trajectory points	Coordinates expression	Coordinates (mm)	Time (s)
starting point P_s	$(x_1 + R_1\cos\theta_{1s},\ 0,\ R_1\sin\theta_{1s})^T$	$(116.5,0,62.9)^T$	0
tangent point P_2	$(x_2 + R_2\cos\theta_{2s},\ 0,\ z_2 - R_2\sin\theta_{2s})^T$	$(132.8,0,-56.1)^T$	T_1
tangent point P_3	$(00,0)^T$	$(00,0)^T$	$T_1 + T_2$
tangent point P_4	$(-x_2 - R_2\cos\theta_{2s},\ 0,\ -z_2 + R_2\sin\theta_{2s})^T$	$(-132.8,0,56.1)^T$	$T_1 + 2T_2$
ending point P_e	$(-x_1 - R_1\cos\theta_{1s},\ 0,\ -R_1\sin\theta_{1s})^T$	$(-116.5,0,-62.9)^T$	$2T_1 + 2T_2$

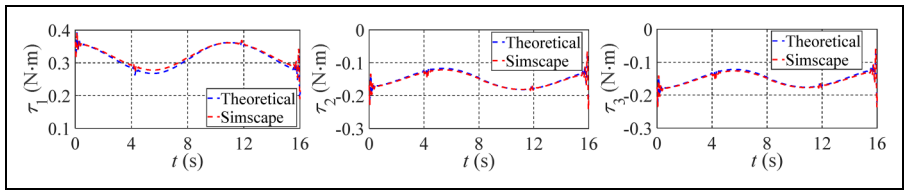

Figure 6. Comparison of driving torque between the theoretical model and Simscape.

Verification of the dynamic model

In order to verify the established dynamic model, one can build a simulation model in MATLAB/Simscape. Taking the typical "S" path addressed above as an example, trajectory planning can be conducted by adopting the 7-order B-spline, from which the command position, command velocity and command acceleration of servomotors can be obtained. Taking the command position, velocity and acceleration of servomotors into the simulation model, one can obtain the driving torque of three servomotors in the parallel mechanism module.

Comparison of driving torque between the theoretical model and Simscape model is shown in Figure 6. As can be seen, the driving torque curves of each servomotor based on the theoretical model and the Simscape model do not coincide completely since that limbs and the moving platform are axisymmetric in the theoretical model. However, the driving torque of the theoretical model is in conformity with that of the simulation model, between which the difference is relatively small. Therefore, the established dynamic model own good accuracy and can be used to the dynamic control of the hybrid robot.

Control behavior analysis

Based on the above parameters, the control algorithm is implemented by combining MATLAB/Simulink and MATLAB/Simscape. Meanwhile, to simulate as realistically as possible, some disturbance is added to the simulation model. The imposed external disturbance torque in the simulation model is white noise, which obeys normal distribution N (0,0.02) as shown in Figure 7.

For comparative analysis, the control behaviors of the PID control, the sliding mode control with fixed parameters (SMC) addressed in Section 4.1 and the proposed modified

sliding mode control (MSMC) are shown in Figure 8, simultaneously. Herein, e_1, e_2 and e_3 denote the tracking errors of three servomotors, respectively.

As can be seen, the three active servomotors own good tracking performance no matter the PID control, the proposed MSMC or the SMC with fixed parameters is adopted. In general, the control performance of the proposed MSMC and the SMC with fixed

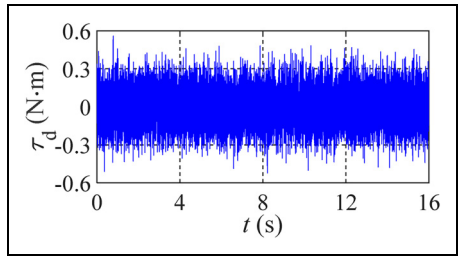

Figure 7. Imposed external disturbance torque.

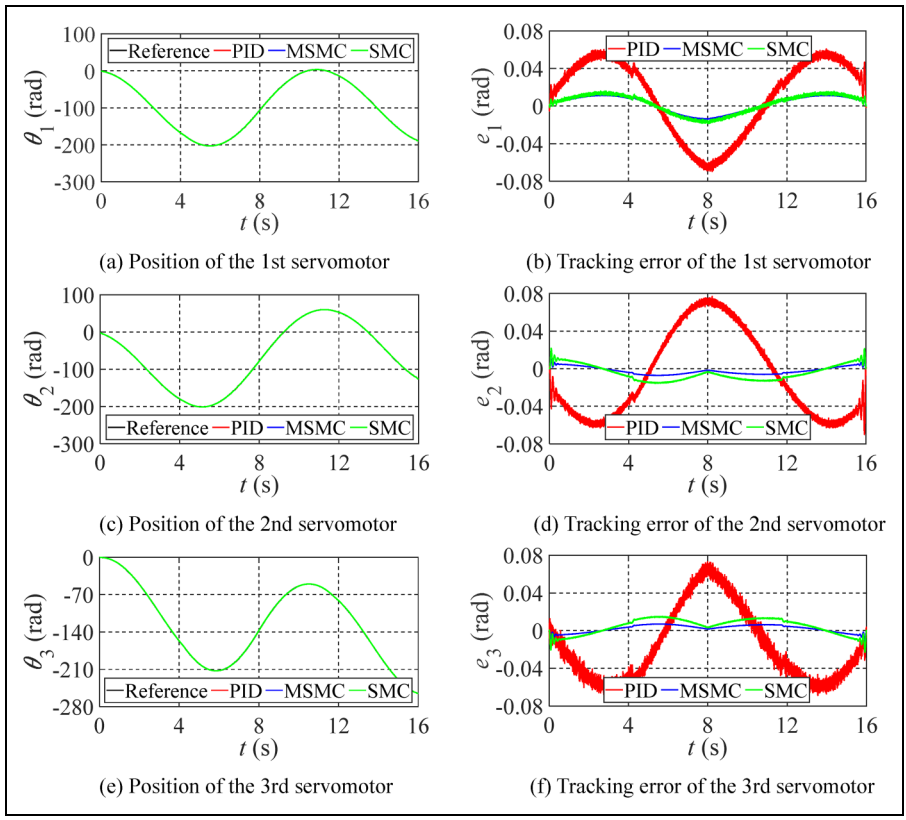

Figure 8. Tracking-trajectory results of three active links under the reference trajectory.

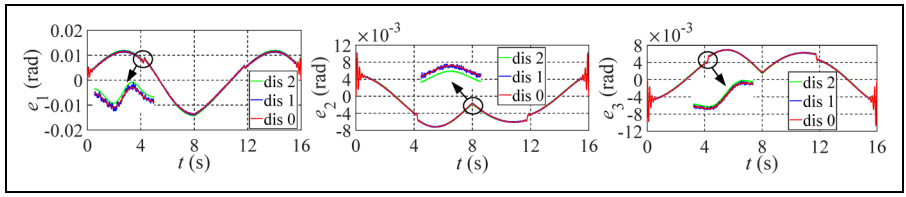

Figure 9. Tracking errors of the proposed MSMC under different disturbances.

parameters is much better than that of PID control. It is due to that the PID control do not consider the change in system's inertia and dynamics. Furthermore, compared to the SMC with fixed parameters, tracking errors of three active links are much smaller and much smoother when adopting the proposed MSMC. That is to say, the proposed MSMC can decrease tracking errors of active joints efficiently due to the self-adaptive parameters dependent on the system state. As can be seen, the three active joints' maximum tracking errors don't exceed 0.02 rad when adopting the proposed MSMC. Accordingly, the root-mean-square error is remarkably reduced from 0.039 rad to 0.011 rad when the proposed MSMC is applied to track the reference path.

Tracking errors of the proposed MSMC under three different working conditions are shown in Figure 9. Herein, dis 0 denotes the case where no disturbance is considered, dis 1 is the case where only external disturbance torque is considered while dis 3 is the one where external disturbance torque and the system's damping are considered simultaneously.

As can be seen from Figure 9, variations of position errors under three working conditions are consistent. Besides, the difference in each servomotor's position error among three working conditions is relatively small. This demonstrates that the proposed MSMC owns good robustness, which can assure the stability of the control system.

Conclusions

A method on motion control of a 5-DOF hybrid robot is proposed in this paper. Main conclusions are drawn as follows.

1. In order to achieve a rapid approaching speed and suppress chattering simultaneously, a modified sliding mode control method with self-adaptive parameters dependent on the system state is proposed.
2. The proposed MSMC could assure that the system will approach to sliding surfaces at an exponential rate when the system state gets far away from sliding surfaces and the speed of the system state approaching to sliding surfaces will become smaller and smaller until 0 when the system state gets near sliding surfaces.
3. Time derivation of the Lyapunov function confirms that the proposed MSMC owns good robustness to model uncertainty, which can assure the stability of the control system.
4. Compared to the traditional PID control and SMC with fixed parameters, the proposed MSMC can decrease tracking errors of the system efficiently. Furthermore, control performance of the proposed MSMC is much smooth.

5. The proposed MSMC is proved to be an effective and practical control method for the hybrid robot, which is possible to be applied to other robots with some modifications.

Declaration of conflicting interests

The authors declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The authors disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported by National Natural Science Foundation of China (52405036), the China Postdoctoral Science Foundation (2023M732984), EPSRC projects (EP/P026087/1, EP/P025447/1) and the European Union's Horizon 2020 research and innovation programme (No. 734272).

ORCID iDs

Yanqin Zhao https://orcid.org/0000-0002-7884-3994
Jiangping Mei https://orcid.org/0000-0001-5261-8759

Author statements

Yanqin Zhao: Supervision, Investigation, Writing – original draft, Funding acquisition.
Mingkun Wu: Investigation, Methodology, Validation.
Jiangping Mei: Investigation, Methodology, Final approval.
Wen Zhao: Investigation, Visualization, Validation.
Yan Jin: Validation, Review & editing, Funding acquisition.

References

1. Caccavale F, Siciliano B and Villani L. The Tricept robot: dynamics and impedance control. *IEEE/ASME Trans Mechatron* 2003; 8: 263–268.
2. Bi ZM and Jin Y. Kinematic modeling of Exechon parallel kinematic machine. *Robot Comput Integr Manuf* 2011; 27: 186–193.
3. Liu Q and Huang T. Inverse kinematics of a 5-axis hybrid robot with non-singular tool path generation. *Robot Comput Integr Manuf* 2019; 5: 140–148.
4. Wu L, Dong CL, Wang GF, et al. An approach to predict lower-order dynamic behaviors of a 5-DOF hybrid robot using a minimum set of generalized coordinates. *Robot Comput Integr Manuf* 2021; 67: 102024.
5. Zhang J, Zhao YQ and Jin Y. Kinetostatic-model-based stiffness analysis of Exechon PKM. *Robot Comput Integr Manuf* 2016; 37: 208–220.
6. Bi W, Xie F, Liu XJ, et al. Optimal design of a novel 4-degree-of-freedom parallel mechanism with flexible orientation capability. *Proc Inst Mech Eng B: J Eng Manuf* 2019; 233: 632–642.
7. Zhao YQ, Mei JP, Jin Y, et al. A new hierarchical approach for the optimal design of a 5-dof hybrid serial-parallel kinematic machine. *Mech Mach Theory* 2021; 156: 104160.
8. Zhen SC, Peng X, Liu XL, et al. A new PD based robust control method for the robot joint module. *Mech Syst Signal Process* 2021; 161: 107958.
9. Muller A and Hufnagel T. Model-based control of redundantly actuated parallel manipulators in redundant coordinates. *Rob Auton Syst* 2012; 60: 563–571.

10. Wu MK, Mei JP, Ni J, et al. Trajectory tracking control of delta parallel robot based on disturbance observer. *Proc Inst Mech Eng I: J Syst Control Eng* 2021; 235: 1193–1203.
11. Han J, Shan XL, Liu H, et al. Fuzzy gain scheduling PID control of a hybrid robot based on dynamic characteristics. *Mech Mach Theory* 2023; 184: 105283.
12. Taghizadeh M and Yarmohammadi MJ. Development of a self-tuning PID controller on hydraulically actuated stewart platform stabilizer with base excitation. *Int J Control Automat Syst* 2018; 16: 2990–2999.
13. Lee KJ, Choi JJ and Kim JS. A proportional-derivative-sliding mode hybrid control scheme for a robot manipulator. *Proc Inst Mech Eng I: J Syst Control Eng* 2004; 218: 667–674.
14. Wang J, Wu J, Wang L, et al. Dynamic feed-forward control of a parallel kinematic machine. *Mechatronics (Oxf)* 2009; 19: 313–324.
15. Song XG, Zhao YJ, Jin L, et al. Dynamic feedforward control in decoupling space for a four-degree-of-freedom parallel robot. *Int J Adv Rob Syst* 2019; 16: 1729881418820451.
16. Song Z, Yi J, Zhao D, et al. A computed torque controller for uncertain robotic manipulator systems: fuzzy approach. *Fuzzy Sets Syst* 2005; 154: 208–226.
17. Cheng H, Yiu YK and Li Z. Dynamics and control of redundantly actuated parallel manipulators. *IEEE/ASME Trans Mechatron* 2003; 8: 483–491.
18. Wu J, Zhang BB, Wang LP, et al. An iterative learning method for realizing accurate dynamic feedforward control of an industrial hybrid robot. *Sci China Technol Sci* 2021; 64: 1177–1188.
19. Zhang B, Wu J, Wang L, et al. A method to realize accurate dynamic feedforward control of a spray-painting robot for airplane wings. *IEEE/ASME Trans Mechatron* 2018; 23: 1182–1192.
20. Li G, Na J, Stoten DP, et al. Adaptive neural network feedforward control for dynamically substructured systems. *IEEE Trans Control Syst Technol* 2013; 22: 944–954.
21. Zhang H, Fang H, Zou Q, et al. Dynamic modeling and adaptive robust synchronous control of parallel robotic manipulator for industrial application. *Complexity* 2020; 1: 5640246.
22. Sancak KV and Bayraktaroglu ZY. Nonlinear computed torque control of 6-dof parallel manipulators. *Int J Control Automat Syst* 2022; 20: 2297–2311.
23. Llama MA, Kelly R and Santibañez V. Stable computed-torque control of robot manipulators via fuzzy self-tuning. *IEEE Trans Syst Man Cybernet B* 2000; 30: 143–150.
24. Chen Y, Ma G, Lin S, et al. Adaptive fuzzy computed-torque control for robot manipulator with uncertain dynamics. *Int J Adv Rob Syst* 2012; 9: 237.
25. Li J, Wang J, Peng H, et al. Fuzzy-torque approximation-enhanced sliding mode control for lateral stability of mobile robot. *IEEE Trans Syst Man Cybernet Syst* 2021; 52: 2491–2500.
26. Zhao J, Yang YG, Liu YB, et al. Discrete sliding mode control with fuzzy adaptive reaching law on 6-PRRS parallel robot. In *IEEE Sixth International Conference on Intelligent Systems Design and Applications*, Jinan, China, 16–18 October 2006, pp. 649–652. New York: IEEE.
27. Navvabi H and Markazi AHD. Position control of Stewart manipulator using a new extended adaptive fuzzy sliding mode controller and observer (E-AFSMCO). *J Franklin Inst* 2018; 355: 2583–2609.
28. Lafmejani AS, Masouleh MT and Kalhor A. Trajectory tracking control of a pneumatically actuated 6-DOF Gough–Stewart parallel robot using Backstepping-Sliding Mode controller and geometry-based quasi forward kinematic method. *Robot Comput Integr Manuf* 2018; 54: 96–114.
29. Zhang QY, Zhao XH, Liu L, et al. Adaptive sliding mode neural network control and flexible vibration suppression of a flexible spatial parallel robot. *Electronics (Basel)* 2021; 10: 212.
30. Lee H and Utkin VI. Chattering suppression methods in sliding mode control systems. *Annu Rev Control* 2007; 31: 179–188.