Robotica: (2021) volume 39, pp. 1575–1587. © The Author(s), 2021. Published by Cambridge University Press.
doi:10.1017/S0263574720001368

A PD Computed Torque Control Method with Online Self-gain Tuning for a 3UPS-PS Parallel Robot

Xiaogang Song†, Yongjie Zhao†*, Chengwei Chen†, Liang'an Zhang‡ and Xinjian Lu¶

†*Department of Mechatronics Engineering, Shantou University, Shantou City, Guangdong 515063, P. R. China*
E-mails: 16xgsong@gmail.com, chengwei0101@outlook.com
‡*School of Mechanical Engineering, Anhui University of Technology, Maanshan 243000, P. R. China*
E-mail: robotlab@ahut.edu.cn
¶*Guangdong Goldenwork Robot Technology Ltd, Foshan City, Guangdong 528226, P. R. China*
E-mail: tigerw813@126.com

(Accepted December 9, 2020. First published online: January 20, 2021)

SUMMARY
In this paper, an online self-gain tuning method of a PD computed torque control (CTC) is used for a 3UPS-PS parallel robot. The CTC is applied to the 3UPS-PS parallel robot based on the robot dynamic model which is established via a virtual work principle. The control system of the robot comprises a nonlinear feed-forward loop and a PD control feedback loop. To implement real-time online self-gain tuning, an adjustment method based on the genetic algorithm (GA) is proposed. Compared with the traditional CTC, the simulation results indicate that the control algorithm proposed in this study can not only enhance the anti-interference ability of the system but also improve the trajectory tracking speed and the accuracy of the 3UPS-PS parallel robot.

KEYWORDS: 3UPS-PS parallel robot; Dynamics; Computed torque control; PD control; Genetic algorithm.

1. Introduction

Parallel robots have effective stiffness and a high load-to-weight ratio.[1–3] These advantages explain why they are widely used in industrial fields, such as machine tools,[4] high-speed sorting,[5,6] and motion simulators.[7,8] With an improvement in the industrial level, the requirements of high accuracy and speed tasks for parallel robots have improved considerably. In recent years, some scholars have carried out several studies on the control algorithm design of parallel robots to promote the performance of the trajectory tracking control of parallel robots to fulfill higher demands.

The controller of parallel robots can be divided into two categories: kinematic controllers and dynamic controllers.[9] Kinematic controllers are based on the kinematic model of parallel robots. Apart from the advantage of having simple structures, kinematic controllers are easy to implement.[10] Unlike the kinematic controllers, the control law of dynamic controllers is designed based on the dynamic model. Moreover, dynamic controllers have better control performance when high speed and greater accuracy are required as a result of the strong coupling, nonlinear dynamic characteristics of parallel robots.[11–13] Belonging to the dynamic controller, the computed torque control (CTC)

* Corresponding author. E-mail: meyjzhao@stu.edu.cn

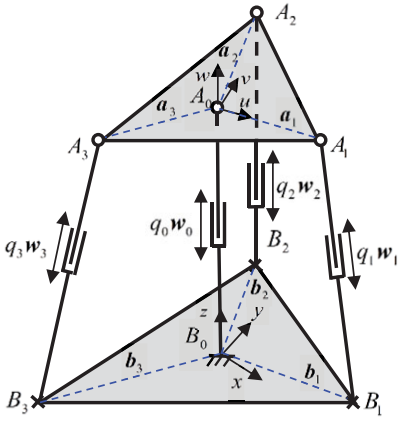

Fig. 1. 3UPS-PS parallel robot structure diagram.

has the advantages of a high tracking accuracy, simplicity, and low energy consumption,[14] and it is widely used in the field of the parallel-robot control.[15–17] However, this controller is sensitive to uncertainties such as modeling errors, frictional losses, and external disturbances because the traditional CTC depends on the accurate dynamic model.[18] To eliminate the effects of uncertainties in the control system, some extensive adaptive and robust control methods have been incorporated with the conventional CTC. Chen et al.[19] combine the conventional CTC and the adaptive control to eliminate the effects of uncertainties in the control system of a 3-DOF serial manipulator. Findings indicate that the proposed control scheme has higher convergence and stability than the traditional CTC. Zhu et al.[20] aim to reduce the effects of the uncertainties from the friction force and unknown disturbance of 3-DOF pneumatic muscles driven parallel manipulator with the adaptive robust control approach based on the traditional CTC. The simulation and experimental results reveal that this control method can guarantee transient performance and final tracking accuracy. Otherwise, the traditional CTC is based on the PD control; simply put, it is difficult for this controller to have a better performance with the fixed gain.[21] To solve this problem, the GA is used because it is easy to implement and because it is unnecessary to provide a set of valid input target pairs in the training set.[22] The study aims to propose a GA-based CTC controller which can be implemented by incorporating a conventional CTC controller and a self-gain tuning method based on the GA. In this paper, the PD gains of the controller proposed are implemented with online auto tuning by using the GA to promote the robustness and accuracy of the control system of the 3UPS-PS parallel robot.

This paper is arranged as follows: In Section 2, the dynamic model of the 3UPS-PS parallel robot is established by using the virtual work principle, and the seventh-order polynomial is used for trajectory planning. In Section 3, the CTC for the 3UPS-PS parallel manipulator with online self-gain tuning based on the GA is proposed. In Section 4, based on theoretical analysis, the simulation of the control system via MATLAB® is provided. Finally, a conclusion is drawn in Section 5.

2. 3UPS-PS Dynamic Analysis

2.1. Kinematics analysis

2.1.1. Position analysis. The 3UPS-PS parallel robot consists of a base platform, a moving platform, and four kinematic limbs which are connected with the base platform and the moving platform. The central limb is the PS limb, and three external limbs are the UPS limb. Here, the U, P, and S denote universal, prismatic, and spherical joints, respectively. The moving platform can implement a one-dimensional movement and a three-dimensional rotation in the workspace by adjusting the length of the four limbs. This kind robot has advantages of high load-to-weight ratio, so it can be adopted to develop a rehabilitation appliance, motion simulator, and so on. Figure 1 depicts the 3UPS-PS parallel robot. The coordinate system is established as follows: The reference coordinate system $B_0 - xyz$ and the moving coordinate system $A_0 - uvw$ are located at the center of mass of the base platform and

https://doi.org/10.1017/S0263574720001368 Published online by Cambridge University Press

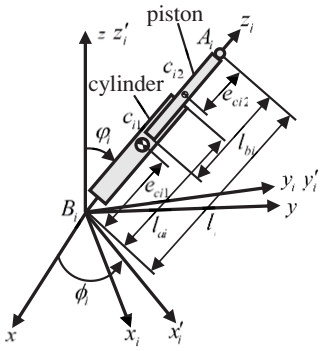

Fig. 2. The local coordinate system of the *i*th limb.

the moving platform, respectively. The orientation of the moving platform can be described by the rotation matrix ${}^{B_0}\boldsymbol{R}_{A_0}$ as shown in Eq. (1)

$$ {}^{B_0}\boldsymbol{R}_{A_0} = \mathrm{Rot}(z, \phi_z)\mathrm{Rot}(y, \phi_y)\mathrm{Rot}(x, \phi_x) \tag{1} $$

where ϕ_x, ϕ_y, and ϕ_z are the Euler angles of rotation of the moving platform on the fixed x-axis, y-axis, and z-axis, respectively.

The linear velocity, angular velocity, and angular acceleration of the moving platform can be expressed as follows:[23,24]

$$ v = \dot{q}_0 \tag{2a} $$

$$ \boldsymbol{\omega} = \begin{bmatrix} \dot{\phi}_x & \dot{\phi}_y & \dot{\phi}_z \end{bmatrix}^T \tag{2b} $$

$$ \dot{\boldsymbol{\omega}} = \begin{bmatrix} \ddot{\phi}_x & \ddot{\phi}_y & \ddot{\phi}_z \end{bmatrix}^T \tag{2c} $$

The generalized position, velocity, and acceleration of the moving platform can be defined as follows:

$$ \boldsymbol{x} = \begin{bmatrix} q_0 & \phi_x & \phi_y & \phi_z \end{bmatrix}^T \tag{3a} $$

$$ \dot{\boldsymbol{x}} = \begin{bmatrix} v & \boldsymbol{\omega} \end{bmatrix}^T \tag{3b} $$

$$ \ddot{\boldsymbol{x}} = \begin{bmatrix} \dot{v} & \dot{\boldsymbol{\omega}} \end{bmatrix}^T \tag{3c} $$

To describe the orientation of each limb, the coordinate system $B_i - x_i y_i z_i$ is established at point B_i as illustrated in Fig. 2. The rotation matrix ${}^{B_0}\boldsymbol{R}_{B_i}$, which describes the relationship between the coordinate system $B_i - x_i y_i z_i$ and the coordinate system $B_0 - xyz$, can be written as follows:

$$ {}^{B_0}\boldsymbol{R}_{B_i} = \mathrm{Rot}(z, \phi_i)\mathrm{Rot}(y', \varphi_i) \tag{4} $$

The position equation associated with the *i*th limb can be written as follows:

$$ q_0\boldsymbol{w}_0 + \boldsymbol{a}_i = \boldsymbol{b}_i + q_i\boldsymbol{w}_i \quad i = 1, 2, 3 \tag{5} $$

where q_0, q_i, $\boldsymbol{w}_0$, $\boldsymbol{w}_i$, $\boldsymbol{a}_i$, and $\boldsymbol{b}_\mathrm{i}$ represent the length of central limb, the length of the *i*th limb, the unit vector along the central limb, the vector along the *i*th limb, the vector A_0A_i, and the vector B_0B_i, respectively.

From Eq. (5), the following equation is obtained:

$$ q_\mathrm{i} = \sqrt{(q_0\boldsymbol{w}_0 + \boldsymbol{a}_\mathrm{i} - \boldsymbol{b}_\mathrm{i})^T(q_0\boldsymbol{w}_0 + \boldsymbol{a}_\mathrm{i} - \boldsymbol{b}_\mathrm{i})} \tag{6} $$

https://doi.org/10.1017/S0263574720001368 Published online by Cambridge University Press

2.1.2. Velocity analysis. Taking the time derivative and both sides dot product with $\boldsymbol{w}_{\mathrm{i}}$ of Eq. (5) and writing it in the matrix form, the linear velocity of the ith limb can be written as follows:

$$\dot{\boldsymbol{q}}_{\mathrm{i}} = \begin{bmatrix} \dot{q}_0 & \dot{q}_1 & \dot{q}_2 & \dot{q}_3 \end{bmatrix}^T = \boldsymbol{J} \begin{bmatrix} v \\ \boldsymbol{\omega} \end{bmatrix} = \boldsymbol{J}\dot{\boldsymbol{x}} \tag{7}$$

$$\boldsymbol{J} = \begin{bmatrix} 1 & \boldsymbol{0}_{3\times 1} \\ \boldsymbol{w}_1^T \boldsymbol{w}_0 & (\boldsymbol{a}_1 \times \boldsymbol{w}_1)^T \\ \boldsymbol{w}_2^T \boldsymbol{w}_0 & (\boldsymbol{a}_2 \times \boldsymbol{w}_2)^T \\ \boldsymbol{w}_3^T \boldsymbol{w}_0 & (\boldsymbol{a}_3 \times \boldsymbol{w}_3)^T \end{bmatrix} \tag{8}$$

Taking the time derivative of Eq. (5) and both sides cross product with $\boldsymbol{w}_{\mathrm{i}}$ of the equation and then simplifying it, the angular velocity in the coordinate system $B_0{-}xyz$ of the ith limb can be obtained as follows:

$$\boldsymbol{\omega}_{\mathrm{i}} = \frac{1}{q_i} S(\boldsymbol{w}_i)(\boldsymbol{v} + S(\boldsymbol{\omega})\boldsymbol{a}_i) \tag{9}$$

where $S(\boldsymbol{w}_i)$ and $S(\boldsymbol{\omega})$ are skew-symmetric matrices.

$$S(\boldsymbol{w}_i) = \begin{bmatrix} 0 & -\boldsymbol{w}_{iz} & \boldsymbol{w}_{iy} \\ \boldsymbol{w}_{iz} & 0 & -\boldsymbol{w}_{ix} \\ -\boldsymbol{w}_{iy} & \boldsymbol{w}_{ix} & 0 \end{bmatrix} \tag{10a}$$

$$S(\boldsymbol{\omega}) = \begin{bmatrix} 0 & -\boldsymbol{\omega}_z & \boldsymbol{\omega}_y \\ \boldsymbol{\omega}_z & 0 & -\boldsymbol{\omega}_x \\ -\boldsymbol{\omega}_y & \boldsymbol{\omega}_x & 0 \end{bmatrix} \tag{10b}$$

Taking the product of both sides of the Eq. (9) with ${}^{B_i}\boldsymbol{R}_{B_0}$ and writing it in the matrix form, the angular velocity of the coordinate system $B_i{-}x_i y_i z_i$ of the ith limb can be expressed as follows:

$${}^{B_i}\boldsymbol{\omega}_{\mathrm{i}} = \frac{1}{q_i} \left[S\left({}^{B_i}\boldsymbol{w}_i\right) {}^{B_i}\boldsymbol{R}_{B_0} - S\left({}^{B_i}\boldsymbol{w}_{\mathrm{i}}\right) S\left({}^{B_i}\boldsymbol{a}_i\right) {}^{B_i}\boldsymbol{R}_{B_0} \right] \begin{bmatrix} \boldsymbol{0}_{2\times 4} \\ \boldsymbol{E}_{4\times 4} \end{bmatrix} \begin{bmatrix} v \\ \boldsymbol{\omega} \end{bmatrix} \tag{11}$$

$$= \boldsymbol{J}_{i\omega} \begin{bmatrix} v \\ \boldsymbol{\omega} \end{bmatrix} \quad i = 1, 2, 3$$

$${}^{B_0}\boldsymbol{\omega}_0 = \boldsymbol{J}_{0\omega} \begin{bmatrix} v \\ \boldsymbol{\omega} \end{bmatrix} = 0_{3\times 1} \tag{12}$$

The linear velocity of the center of mass c_{i1} in the ith limb can be described in the coordinate system $B_i{-}x_i y_i z_i$ as follows:

$${}^{B_i}\boldsymbol{v}_{c_{i1}} = -{}^{B_i}\boldsymbol{e}_{i1} \times {}^{B_i}\boldsymbol{\omega}_i \tag{13a}$$

$${}^{B_i}\boldsymbol{v}_{c_{i1}} = -S\left({}^{B_i}\boldsymbol{e}_{c_{i1}}\right) \boldsymbol{J}_{i\omega} \begin{bmatrix} v \\ \boldsymbol{\omega} \end{bmatrix} = J_{vc_{i1}} \begin{bmatrix} v \\ \omega \end{bmatrix} i = 1, 2, 3 \tag{13b}$$

$${}^{B_0}\boldsymbol{v}_{c_{01}} = \boldsymbol{J}_{vc_{01}} \begin{bmatrix} v \\ \boldsymbol{\omega} \end{bmatrix} = 0_{3\times 1} \tag{14}$$

$$\boldsymbol{J}_{c_{i1}} = \begin{bmatrix} \boldsymbol{J}_{vc_{i1}} & \boldsymbol{J}_{i\omega} \end{bmatrix}^T \tag{15}$$

where ${}^{B_i}\boldsymbol{e}_{ci1} = \boldsymbol{e}_{ci1}{}^{Bi}\boldsymbol{w}_i$.

https://doi.org/10.1017/S0263574720001368 Published online by Cambridge University Press

The linear velocity of the center of mass c_{i2} in the ith limb can be described in the coordinate system $B_i - x_i y_i z_i$ as follows:

$$ {}^{B_i}\boldsymbol{v}_{c_{i2}} = {}^{B_i}\boldsymbol{v} + {}^{B_i}\boldsymbol{\omega} \times {}^{B_i}\boldsymbol{a}_i - {}^{B_i}\boldsymbol{\omega}_i \times {}^{B_i}\boldsymbol{e}_{c_{i2}} \tag{16a} $$

$$ {}^{B_i}\boldsymbol{v}_{c_{i2}} = \left(\left[{}^{B_i}\boldsymbol{R}_{B_0} \quad -S\left({}^{B_i}\boldsymbol{a}_i\right) {}^{B_i}\boldsymbol{R}_{B_0} \right] \begin{bmatrix} \boldsymbol{0}_{2\times 4} \\ \boldsymbol{E}_{4\times 4} \end{bmatrix} + S\left({}^{B_i}\boldsymbol{e}_{c_{i2}}\right) \boldsymbol{J}_{i\omega} \right) \begin{bmatrix} v \\ \boldsymbol{\omega} \end{bmatrix} \tag{16b} $$

$$ \boldsymbol{J}_{vc_{i2}} = \left[{}^{B_i}\boldsymbol{R}_{B_0} \quad -S\left({}^{B_i}\boldsymbol{a}_i\right) {}^{B_i}\boldsymbol{R}_{B_0} \right] \begin{bmatrix} \boldsymbol{0}_{2\times 4} \\ \boldsymbol{E}_{4\times 4} \end{bmatrix} + S\left({}^{B_i}\boldsymbol{e}_{c_{i2}}\right) \boldsymbol{J}_{i\omega} \quad i = 1, 2, 3 \tag{17} $$

$$ {}^{B_i}\boldsymbol{v}_{c_{02}} = \dot{q}\boldsymbol{w}_0 = \boldsymbol{J}_{vc_{02}} \begin{bmatrix} v \\ \boldsymbol{\omega} \end{bmatrix} \tag{18} $$

2.1.3. Acceleration analysis. Taking the second derivative of Eq. (5) with respect to time and both sides dot product with $\boldsymbol{w}_\mathrm{i}$ of the equation and rewriting it in the matrix form, the acceleration of the ith limb can be calculated as follows:

$$ \ddot{\boldsymbol{q}}_i = \begin{bmatrix} \ddot{q}_0 \\ \ddot{q}_1 \\ \ddot{q}_2 \\ \ddot{q}_3 \end{bmatrix} = \boldsymbol{J} \begin{bmatrix} \dot{v} \\ \dot{\boldsymbol{\omega}} \end{bmatrix} + \begin{bmatrix} 0 \\ q_1 \left(\boldsymbol{\omega}_1^T \boldsymbol{\omega}_1\right) + \left(\boldsymbol{w}_1^T \boldsymbol{\omega}\right) \left(\boldsymbol{\omega}^T \boldsymbol{a}_1\right) - \left(\boldsymbol{w}_1^T \boldsymbol{a}_1\right) \left(\boldsymbol{\omega}^T \boldsymbol{\omega}\right) \\ q_2 \left(\boldsymbol{\omega}_2^T \boldsymbol{\omega}_2\right) + \left(\boldsymbol{w}_2^T \boldsymbol{\omega}\right) \left(\boldsymbol{\omega}^T \boldsymbol{a}_2\right) - \left(\boldsymbol{w}_2^T \boldsymbol{a}_2\right) \left(\boldsymbol{\omega}^T \boldsymbol{\omega}\right) \\ q_3 \left(\boldsymbol{\omega}_3^T \boldsymbol{\omega}_3\right) + \left(\boldsymbol{w}_3^T \boldsymbol{\omega}\right) \left(\boldsymbol{\omega}^T \boldsymbol{a}_3\right) - \left(\boldsymbol{w}_3^T \boldsymbol{a}_3\right) \left(\boldsymbol{\omega}^T \boldsymbol{\omega}\right) \end{bmatrix} \tag{19} $$

Taking the second derivative of the right side of Eq. (5) with respect to time and taking the cross product of both sides of the equation with ${}^{\mathrm{B}_i}\boldsymbol{w}_\mathrm{i}$ and substituting ${}^{B_i}\boldsymbol{w}_i^{T\,B_i}\boldsymbol{\omega}_i = {}^{B_i}\dot{\boldsymbol{w}}_i^{T\,B_i}\boldsymbol{\omega}_i = 0$ into the function, the angular acceleration in the coordinate system $B_i - x_i y_i z_i$ of the ith limb can be computed as follows:

$$ \begin{aligned} {}^{B_i}\dot{\boldsymbol{\omega}}_\mathrm{i} &= \boldsymbol{J}_{i\omega} \begin{bmatrix} \dot{v} \\ \dot{\boldsymbol{\omega}} \end{bmatrix} \\ &\quad + \frac{1}{q_i} \begin{pmatrix} \left({}^{B_i}\omega^{T\,B_i}\boldsymbol{a}_i\right) \left({}^{B_i}\boldsymbol{w}_i \times {}^{B_i}\omega\right) \\ -\left({}^{B_i}\boldsymbol{\omega}^{T\,B_i}\boldsymbol{\omega}\right) \left({}^{B_i}\boldsymbol{w}_i \times {}^{B_i}\boldsymbol{a}_i\right) - 2\dot{q}_i\, {}^{B_i}\boldsymbol{\omega}_i \end{pmatrix} \end{aligned} \tag{20} $$

Since the central limb is fixed on the base platform, ${}^{B_0}\dot{\boldsymbol{\omega}}_0 = 0_{3\times 1}$.

Taking the derivative of Eq. (13a) with respect to time and substituting Eq. (20) into the equation, the linear acceleration of the center of mass c_{i1} in the ith limb described in the coordinate system $B_i - x_i y_i z_i$ can be expressed as follows:

$$ \begin{aligned} {}^{B_i}\dot{\boldsymbol{v}}_{c_{i1}} &= \boldsymbol{J}_{vc_{i1}} \begin{bmatrix} \dot{v} \\ \dot{\boldsymbol{\omega}} \end{bmatrix} - \frac{1}{q_i} \begin{pmatrix} \left(\left({}^{B_i}\boldsymbol{\omega}^{T\,B_i}\boldsymbol{a}_i\right) \left({}^{B_i}\boldsymbol{e}_{i1}^{T\,B_i}\boldsymbol{\omega}\right) - \left({}^{B_i}\boldsymbol{\omega}^{T\,B_i}\boldsymbol{\omega}\right) \left({}^{B_i}\boldsymbol{e}_{i1}^{T\,B_i}\boldsymbol{a}_i\right)\right) {}^{B_i}\boldsymbol{w}_i \\ +e_{i1} \left(-\left({}^{B_i}\boldsymbol{\omega}^{T\,B_i}\boldsymbol{a}_i\right) {}^{B_i}\boldsymbol{\omega} + \left({}^{B_i}\boldsymbol{\omega}^{T\,B_i}\boldsymbol{\omega}\right) {}^{B_i}\boldsymbol{a}_i\right) \\ +2\dot{q}_i \left({}^{B_i}\boldsymbol{\omega}_i \times {}^{B_i}\boldsymbol{e}_{i1}\right) \end{pmatrix} \\ &\quad - \left({}^{B_i}\boldsymbol{\omega}_i^{T\,B_i}\boldsymbol{\omega}_i\right) {}^{Bi}\boldsymbol{e}_{i1} \quad i = 1, 2, 3 \end{aligned} \tag{21} $$

Taking the time derivative of Eq. (16a) and substituting Eq. (20) into the equation, the linear acceleration of the center of mass c_{i2} in the ith limb described in the coordinate system $B_i - x_i y_i z_i$ is written as follows:

$$ \begin{aligned} {}^{B_i}\dot{\boldsymbol{v}}_{c_{i2}} &= \boldsymbol{J}_{vc_{i2}} \begin{bmatrix} \dot{v} \\ \dot{\boldsymbol{\omega}} \end{bmatrix} + {}^{B_i}\boldsymbol{\omega} \times \left({}^{B_i}\boldsymbol{\omega} \times {}^{B_i}\boldsymbol{a}_i\right) - {}^{B_i}\boldsymbol{\omega}_i \times \left({}^{B_i}\boldsymbol{\omega}_i \times {}^{B_i}\boldsymbol{e}_{i2}\right) \\ &\quad + \frac{1}{q_i} S\left({}^{B_i}\boldsymbol{e}_{i2}\right) \begin{pmatrix} \left({}^{B_i}\boldsymbol{\omega}^{T\,B_i}\boldsymbol{a}_i\right) \left({}^{B_i}\boldsymbol{w}_i \times {}^{B_i}\boldsymbol{\omega}\right) \\ -\left({}^{B_i}\boldsymbol{\omega}^{T\,B_i}\boldsymbol{\omega}\right) \left({}^{B_i}\boldsymbol{w}_i \times {}^{B_i}\boldsymbol{a}_i\right) - 2\dot{q}_i\, {}^{B_i}\boldsymbol{\omega}_i \end{pmatrix} \end{aligned} \tag{22} $$

https://doi.org/10.1017/S0263574720001368 Published online by Cambridge University Press

$$ {}^{B_i}\dot{\boldsymbol{v}}_{c_{i2}} = \begin{bmatrix} \boldsymbol{w}_{i0} & \boldsymbol{0}_{3\times 3} \end{bmatrix} \begin{bmatrix} \dot{v} \\ \dot{\boldsymbol{\omega}} \end{bmatrix} = \boldsymbol{J}_{vc_{02}} \begin{bmatrix} \dot{v} \\ \dot{\boldsymbol{\omega}} \end{bmatrix} \tag{23} $$

2.2. Dynamic analysis

According to the D'Alembert's principle, the following equation represents the resultant force/moment acting on the center of mass of the moving platform:

$$ \boldsymbol{F} = \begin{bmatrix} \boldsymbol{f} \\ \boldsymbol{n} \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}_e + m\boldsymbol{g} - m\dot{\boldsymbol{v}}_c \\ \boldsymbol{n}_e - {}^{B_0}\boldsymbol{I}\dot{\boldsymbol{\omega}} - \boldsymbol{\omega} \times ({}^{B_0}\boldsymbol{I}\boldsymbol{\omega}) \end{bmatrix} \tag{24} $$

where $\boldsymbol{f}$ and $\boldsymbol{n}$ denote the external force and moment exerted at the barycenter of the moving platform, respectively. ${}^{B_0}\boldsymbol{I} = {}^{B_0}\boldsymbol{R}_{A_0}\,{}^{A_0}\boldsymbol{I}\,{}^{B_0}\boldsymbol{R}_{A_0}^{-1}$ is the inertia matrix of the moving platform about the barycenter in the coordinate system $B_0 - xyz$, and $\dot{\boldsymbol{v}}_c$is the acceleration of the center of mass of the moving platform.

The following equation is the resultant force/moment exerted on the center of mass c_{i1} and c_{i2}:

$$ {}^{B_i}\boldsymbol{F}_{c_{ij}} = \begin{bmatrix} {}^{B_i}\boldsymbol{f}_{c_{ij}} \\ {}^{B_i}\boldsymbol{n}_{c_{ij}} \end{bmatrix} = \begin{bmatrix} m_{c_{ij}}\,{}^{B_i}\boldsymbol{R}_{B_0}\boldsymbol{g} - m_{c_{ij}}\dot{\boldsymbol{v}}_{c_{ij}} \\ -{}^{B_i}\boldsymbol{I}_{c_{ij}}\,{}^{B_i}\dot{\boldsymbol{\omega}}_i - {}^{B_i}\boldsymbol{\omega}_i \times \left({}^{B_0}\boldsymbol{I}_{c_{ij}}\,{}^{B_i}\boldsymbol{\omega}_i\right) \end{bmatrix} \quad i = 0, 1, 2, 3 \;\; j = 1, 2 \tag{25} $$

where m_{ci1}, m_{ci2} ${}^{B_i}\boldsymbol{I}_{ci1}$, and ${}^{B_i}\boldsymbol{I}_{ci2}$ denote the mass of the cylinder, the mass of the piston of the ith limb, the inertia matrix of the cylinder, and the inertia matrix of the piston of the ith limb in the coordinate system $B_i - x_i y_i z_i$, respectively.

The following equation is obtained according to the virtual work principle:

$$ \delta\boldsymbol{x}^T\boldsymbol{J}_c\boldsymbol{F} + \sum_{i=1}^{3}\left(\delta\boldsymbol{x}^T\boldsymbol{J}_{c_{i1}}^T\,{}^{B_i}\boldsymbol{F}_{c_{i1}}\right) + \sum_{i=0}^{3}\left(\delta\boldsymbol{x}^T\boldsymbol{J}_{c_{i2}}^T\,{}^{B_i}\boldsymbol{F}_{c_{i2}}\right) + \delta\boldsymbol{x}^T\boldsymbol{J}^{\mathrm{T}}\boldsymbol{\tau} = 0 \tag{26a} $$

where $\boldsymbol{\delta x}$ is the rotational virtual displacement of the moving platform and the virtual displacement of the point of the joint A_0. Because the Eq. (26a) is valid for any virtual displacement $\boldsymbol{\delta x}$. It must be as follows:

$$ \boldsymbol{J}_{\mathrm{c}}\boldsymbol{F} + \sum_{i=1}^{3}\left(\boldsymbol{J}_{c_{i1}}^T\,{}^{B_i}\boldsymbol{F}_{c_{i1}}\right) + \sum_{i=0}^{3}\left(\boldsymbol{J}_{c_{i2}}^T\,{}^{B_i}\boldsymbol{F}_{c_{i2}}\right) + \boldsymbol{J}^{\mathrm{T}}\boldsymbol{\tau} = 0 \tag{26b} $$

Substituting Eqs. (7), (11), (20)–(22), (24), (25) into the Eq. (26b) and rewriting the equation in the matrix form, the dynamic model of the 3UPS-PS parallel robot can be written as follows:

$$ \boldsymbol{J}^{\mathrm{T}}\boldsymbol{\tau} = \boldsymbol{M}(\boldsymbol{x})\,\ddot{\boldsymbol{x}} + \boldsymbol{C}(\boldsymbol{x}, \dot{\boldsymbol{x}}) + \boldsymbol{G}(\boldsymbol{x}) \tag{27} $$

where

$$ \boldsymbol{M}(\boldsymbol{x}) = \begin{pmatrix} \boldsymbol{J}_c^T \begin{bmatrix} m\boldsymbol{E}_{3\times 3} & \boldsymbol{0}_{3\times 3} \\ \boldsymbol{0}_{3\times 3} & {}^{B_0}\boldsymbol{I} \end{bmatrix} \boldsymbol{J}_c \\ + \sum_{i=1}^{3}\left(\boldsymbol{J}_{vci1}^T m_{ci1}\boldsymbol{J}_{vci1} + \boldsymbol{J}_{i\omega}^T\,{}^{B_i}\boldsymbol{I}_{ci1}\boldsymbol{J}_{i\omega}\right) \\ + \sum_{i=0}^{3}\left(\boldsymbol{J}_{vci2}^T m_{ci2}\boldsymbol{J}_{vci2} + \boldsymbol{J}_{i\omega}^T\,{}^{B_i}\boldsymbol{I}_{ci2}\boldsymbol{J}_{i\omega}\right) \end{pmatrix} \tag{28} $$

$$ \boldsymbol{J}_c = \begin{bmatrix} \boldsymbol{w}_0 & \boldsymbol{0}_{3\times 3} \\ \boldsymbol{0}_{3\times 1} & \boldsymbol{E}_{3\times 3} \end{bmatrix} \tag{29} $$

https://doi.org/10.1017/S0263574720001368 Published online by Cambridge University Press

$$\boldsymbol{G}(\boldsymbol{x}) = -\begin{pmatrix} \left(\boldsymbol{J}_c^T \begin{bmatrix} m\boldsymbol{g} \\ \boldsymbol{0}_{3\times 1} \end{bmatrix}\right) + \sum_{i=1}^{3} \left(\boldsymbol{J}_{ci1}^T \begin{bmatrix} m_{ci1}{}^{B_i}\boldsymbol{R}_{B_0}\boldsymbol{g} \\ \boldsymbol{0}_{3\times 1} \end{bmatrix}\right) \\ + \sum_{i=0}^{3} \left(\boldsymbol{J}_{ci2}^T \begin{bmatrix} m_{ci1}{}^{B_i}\boldsymbol{R}_{B_0}\boldsymbol{g} \\ \boldsymbol{0}_{3\times 1} \end{bmatrix}\right) \end{pmatrix} \tag{30}$$

2.3. Trajectory planning

The purpose of trajectory planning is to make the moving platform of the 3UPS-PS parallel robot reach the specified orientation along a smooth curve without a flexible and rigid impact. The seventh-order polynomial planning method is a common planning method, which can satisfy the condition that the initial/final speed, acceleration, and jerk of the moving platform of the robot are 0. The specific constraints are as follows:

$$\begin{cases} \phi_{x0} = c_{x0},\ \phi_{xf} = c_{xf},\ \dot{\phi}_{x0} = \dot{\phi}_{xf} = 0 \\ \phi_{y0} = c_{y0},\ \phi_{yf} = c_{yf}, \dot{\phi}_{y0} = \dot{\phi}_{yf} = 0 \\ \phi_{z0} = c_{z0},\ \phi_{zf} = c_{zf}, \dot{\phi}_{z0} = \dot{\phi}_{zf} = 0 \\ z_0 = c_{zt0}, z_f = c_{zf0}, \dot{z}_0 = \dot{z}_f = 0 \end{cases} \tag{31a}$$

$$\begin{cases} \ddot{\phi}_{x0} = \ddot{\phi}_{xf} = 0\,,\ \dddot{\phi}_{x0} = \dddot{\phi}_{xf} = 0 \\ \ddot{\phi}_{y0} = \ddot{\phi}_{yf} = 0\,,\ \dddot{\phi}_{y0} = \dddot{\phi}_{yf} = 0 \\ \ddot{\phi}_{z0} = \ddot{\phi}_{zf} = 0\,,\ \dddot{\phi}_{z0} = \dddot{\phi}_{zf} = 0 \\ \ddot{z}_0 = \ddot{z}_f = 0\,,\ \dddot{z}_0 = \dddot{z}_f = 0 \end{cases} \tag{31b}$$

where $\phi_{x0}, \phi_{xf}, \phi_{y0}, \phi_{yf}, \phi_{z0}, \phi_{zf}$ are the initial/final Euler angles of rotation of the moving platform on the fixed x-axis, y-axis, and z-axis, respectively. z_0, z_f represent the initial/final displacement of the moving platform along the z-axis. The moving platform moves from initial orientation to final orientation through the seventh-order polynomial interpolation which can be written as follows:

$$P(\mathrm{t}) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5 + a_6 t^6 + a_7 t^7 \tag{32}$$

Substituting Eq. (31) into Eq. (32), the trajectory of the 3UPS-PS parallel robot can be obtained as follows. The simulation results are shown in Fig. 3.

$$\begin{cases} \phi_{\mathrm{x}} = \left(0.035\mathrm{t}^4 - 0.0084\mathrm{t}^5 + 7\times 10^{-4}\mathrm{t}^6 - 2\times 10^{-5}\mathrm{t}^7\right)\times \pi/180 \\ \phi_{\mathrm{y}} = \left(0.035\mathrm{t}^4 - 0.0084\mathrm{t}^5 + 7\times 10^{-4}\mathrm{t}^6 - 2\times 10^{-5}\mathrm{t}^7\right)\times \pi/180 \\ \phi_{\mathrm{z}} = \left(-10 + 0.035\mathrm{t}^4 - 0.0042\mathrm{t}^5 - 3.5\times 10^{-4}\mathrm{t}^6 - 10^{-5}\mathrm{t}^7\right)\times \pi/180 \\ z = 0.313 + 3.5\times 10^{-5}\mathrm{t}^4 - 8.4\times 10^{-6}\mathrm{t}^5 - 7\times 10^{-7}\mathrm{t}^6 - 2\times 10^{-8}\mathrm{t}^7 \end{cases} \tag{33}$$

3. Control System Structure

The CTC is applied to control the 3UPS-PS parallel robot based on the dynamic model. The control system consists of a nonlinear feedforward loop and a PD control feedback loop with an online self-gain tuning approach based on the GA. Figure 4 illustrates the control system structure.

3.1. Computed torque control

3.1.1. Control law design. Given the disturbance in the control process, the dynamics model of the 3UPS-PS parallel robot can be expressed as follows:

$$\boldsymbol{J}^T\boldsymbol{\tau} = \boldsymbol{M}(\boldsymbol{x})\ddot{\boldsymbol{x}} + \boldsymbol{C}(\boldsymbol{x}, \dot{\boldsymbol{x}}) + \boldsymbol{G}(\boldsymbol{x}) + \boldsymbol{\tau}_B \tag{34}$$

where $\boldsymbol{\tau}_B = \boldsymbol{J}_c\boldsymbol{\tau}_b$, $\boldsymbol{\tau}_b$ is the disturbance force.

https://doi.org/10.1017/S0263574720001368 Published online by Cambridge University Press

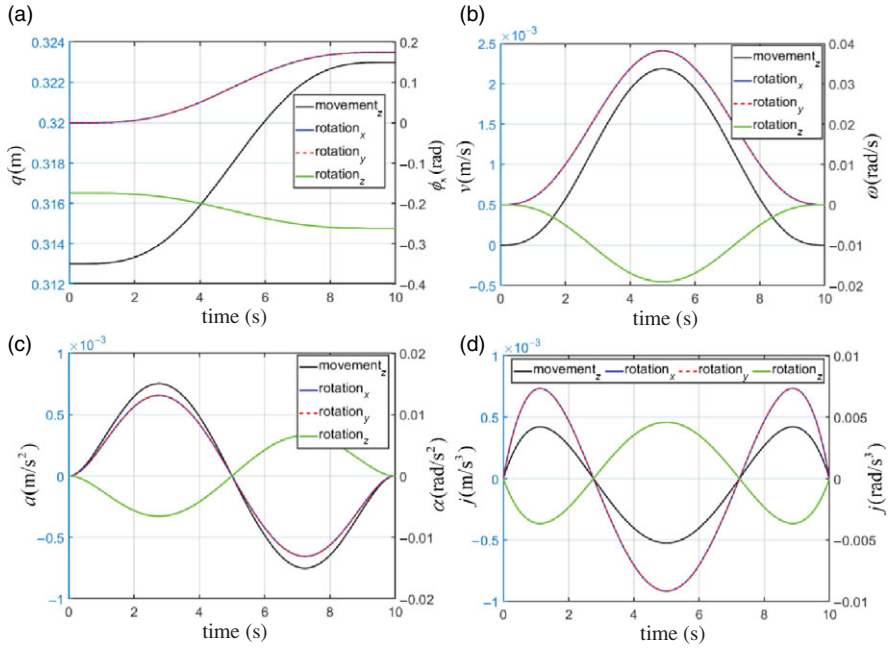

Fig. 3. Trajectory planning for 3UPS-PS robot.

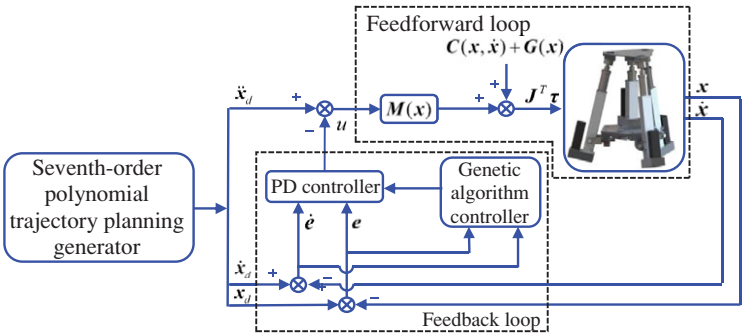

Fig. 4. 3UPS-PS control system structure diagram.

By allowing $\boldsymbol{e}$ to be the deviation between the actual orientation and the desired orientation of the moving platform, the following equation is obtained:

$$\boldsymbol{e} = \boldsymbol{x}_d(t) - \boldsymbol{x}(t) \tag{35}$$

Taking the derivative of the Eq. (34) with respect to time, the equation is as follows:

$$\dot{\boldsymbol{e}} = \dot{\boldsymbol{x}}_d(t) - \dot{\boldsymbol{x}}(t) \tag{36}$$

$$\ddot{\boldsymbol{e}} = \ddot{\boldsymbol{x}}_d(t) - \ddot{\boldsymbol{x}}(t) \tag{37}$$

Substituting Eq. (34) into Eq. (37), the equation is as follows:

$$\ddot{\boldsymbol{e}} = \ddot{\boldsymbol{x}}_d + \boldsymbol{M}(\boldsymbol{x})^{-1}\left(\boldsymbol{C}(\boldsymbol{x},\dot{\boldsymbol{x}}) + \boldsymbol{G}(\boldsymbol{x}) + \boldsymbol{\tau}_B - \boldsymbol{J}^T\boldsymbol{\tau}\right) \tag{38}$$

The control input function is defined as follows:

$$\boldsymbol{u} = \ddot{\boldsymbol{x}}_d + \boldsymbol{M}(\boldsymbol{x})^{-1}\left(\boldsymbol{C}(\boldsymbol{x},\dot{\boldsymbol{x}}) + \boldsymbol{G}(\boldsymbol{x}) - \boldsymbol{J}^T\boldsymbol{\tau}\right) \tag{39}$$

The control law of the system can be obtained using Eq. (39).

$$\boldsymbol{J}^T\boldsymbol{\tau} = \boldsymbol{M}(\boldsymbol{x})(\ddot{\boldsymbol{x}}_d - \boldsymbol{u}) + \boldsymbol{C}(\boldsymbol{x},\dot{\boldsymbol{x}}) + \boldsymbol{G}(\boldsymbol{x}) \tag{40}$$

https://doi.org/10.1017/S0263574720001368 Published online by Cambridge University Press

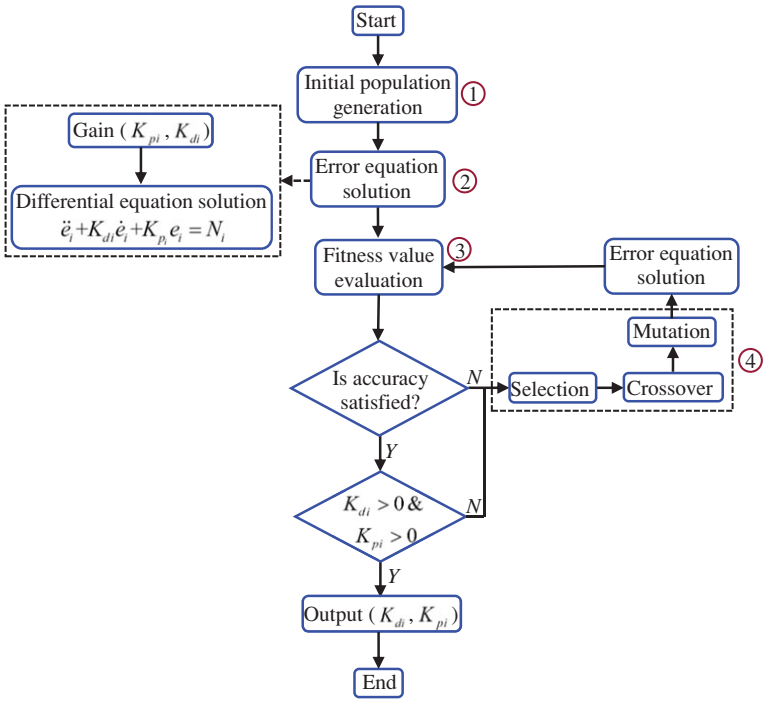

Fig. 5. Self-gain tuning algorithm.

Substituting Eq. (34) into Eq. (40), the equation is as follows:

$$\ddot{\boldsymbol{e}} = \boldsymbol{u} + \boldsymbol{M}(\boldsymbol{x})^{-1}\boldsymbol{\tau}_B \tag{41}$$

3.1.2. Feedback loop design. The feedback loop adopts the mature theory of the PD control, and the feedback law is defined as follows:

$$\boldsymbol{u} = -\boldsymbol{K}_d\dot{\boldsymbol{e}} - \boldsymbol{K}_p\boldsymbol{e} \tag{42}$$

where $\boldsymbol{K}_d = \mathrm{diag}(K_{d1} \quad K_{d2} \quad K_{d3} \quad K_{d4})$ and $\boldsymbol{K}_p = \mathrm{diag}(K_{p1} \quad K_{p2} \quad K_{p3} \quad K_{p4})$ denote the differential coefficient matrix and the proportional coefficient matrix, respectively, and both of $\boldsymbol{K}_d$ and $\boldsymbol{K}_p$ are diagonal matrices.

Substituting Eq. (42) into Eq. (40), the control input of the moving platform of the robot is written as follows:

$$\boldsymbol{J}^T\boldsymbol{\tau} = \boldsymbol{M}(\boldsymbol{x})(\ddot{\boldsymbol{x}}_d + \boldsymbol{K}_d\dot{\boldsymbol{e}} + \boldsymbol{K}_p\boldsymbol{e}) + \boldsymbol{C}(\boldsymbol{x}, \dot{\boldsymbol{x}}) + \boldsymbol{G}(\boldsymbol{x}) \tag{43}$$

Substituting Eq. (42) into Eq. (41), the closed-loop system error equation can be expressed as follows:

$$\ddot{\boldsymbol{e}} + \boldsymbol{K}_d\dot{\boldsymbol{e}} + \boldsymbol{K}_p\boldsymbol{e} = \boldsymbol{M}(\boldsymbol{x})^{-1}\boldsymbol{\tau}_B = \boldsymbol{N} \tag{44}$$

Defining the state variable of the control system as $\boldsymbol{Q} = \left[\boldsymbol{e}\ \dot{\boldsymbol{e}}\right]^T$, the error equation can be expressed in form of the state equation.

$$\frac{d}{d_t}\begin{bmatrix}\boldsymbol{e}\\ \dot{\boldsymbol{e}}\end{bmatrix} = \begin{bmatrix}0_{4\times4} & \boldsymbol{E}_{4\times4}\\ -\boldsymbol{K}_p & -\boldsymbol{K}_d\end{bmatrix}\begin{bmatrix}\boldsymbol{e}\\ \dot{\boldsymbol{e}}\end{bmatrix} + \begin{bmatrix}0_{4\times4}\\ \boldsymbol{E}_{4\times4}\end{bmatrix}\left(\boldsymbol{M}(\boldsymbol{x})^{-1}\boldsymbol{\tau}_B\right) \tag{45}$$

3.2. PD gain self-tuning algorithm

The 3UPS-PS parallel robot is a highly nonlinear coupled time-varying system. A lot of uncertainties can affect the stability and accuracy of the control system, such as model errors and frictional losses. To improve the performance of the traditional CTC, the GA is used to implement the online self-gain tuning of proportional parameters and differential parameters of PD controllers. The computation process is divided into four steps, and the algorithm flowchart is shown in Fig. 5.

https://doi.org/10.1017/S0263574720001368 Published online by Cambridge University Press

Table I. The parameters of the moving platform (m).

Parameters	**0**	**1**	**2**	**3**
x_{Bi}	0	0.1040	−0.0520	−0.0520
y_{Bi}	0	0	0.0901	−0.0901
z_{Bi}	0	0	0	0

Table II. The parameters of the based platform (m).

Parameters	**0**	**1**	**2**	**3**
x_{Ai}	0	0.1813	−0.0907	−0.0907
y_{Ai}	0	0	0.1570	−0.1570
z_{Ai}	0	0	0	0

Table III. The parameters of the control system.

Parameters	z	$\boldsymbol{\phi}_x$	$\boldsymbol{\phi}_y$	$\boldsymbol{\phi}_z$
K_p	5000	5000	5000	5000
K_d	2000	2000	2000	2000
e_0	0.001	0.001	0.001	0.001

(1) **Initialization**: The evolutionary algebraic counter is set to 0, and 50 individuals associated with k_{di} and k_{pi} are randomly generated as the initial population.
(2) **Error equationsolution**: The individuals that involve k_{di} and k_{di} in the last step are substituted into the differential equations and then the error equation and the fitness value are solved.
(3) **Fitness value evaluation**: The fitness value is assessed by the following fitness value evaluation function:

$$\min\ Z = |x_{di} - x_i| = |e_i| \tag{46}$$

(4) **Generating a new population:** If the condition ($K_{di} > 0,\ K_{pi} > 0$) is satisfied (when the $\boldsymbol{K}_d$ and $\boldsymbol{K}_p$ are positive definite matrices, the error system of the 3UPS-PS parallel robot is asymptotically stable[25]), the fitness value meets the accuracy requirement, and the most optimal individuals are outputted. If not, the new populations are generated by selection, crossover, and mutation and then step 2 and step 3 are repeated.

4. Simulation

According to the theoretical analysis, the performance of the CTC based on the GA is compared with the traditional CTC by simulation. The structural parameters of the 3UPS-PS parallel robot are shown in Tables I and II. The parameters of the traditional CTC are shown in Table III. The mass of the moving platform, the mass of the piston of the *i*th limb, and the mass of cylinder of the *i*th limb are 5.5, 0.15, and 0.15 kg, respectively. The initial error is e_0. The trajectory of moving platform is given in Eq. (33). The trajectory tracking abilities of moving platform of traditional CTC and the control algorithm proposed in this paper along x-axis, y-axis, and z-axis, respectively, are compared and analyzed under uncertain disturbance. The simulation is lasting 10 s. The simulation results are shown in Figs. 6–9.

By analyzing the trajectory tracking image (a) in Figs. 6–9, the CTC based on the GA has a higher trajectory tracking accuracy and speed than the traditional CTC. When the CTC based on the GA is applied, the tracking speed and the anti-jamming ability of the robot system can be improved, the tracking error is less than 10^{-4}, and the error fluctuation is reduced (see image b in Figs. 6–9). The optimal value of K_d is between 0 and 300, and the optimal value K_p is between 0 and 18,000

https://doi.org/10.1017/S0263574720001368 Published online by Cambridge University Press

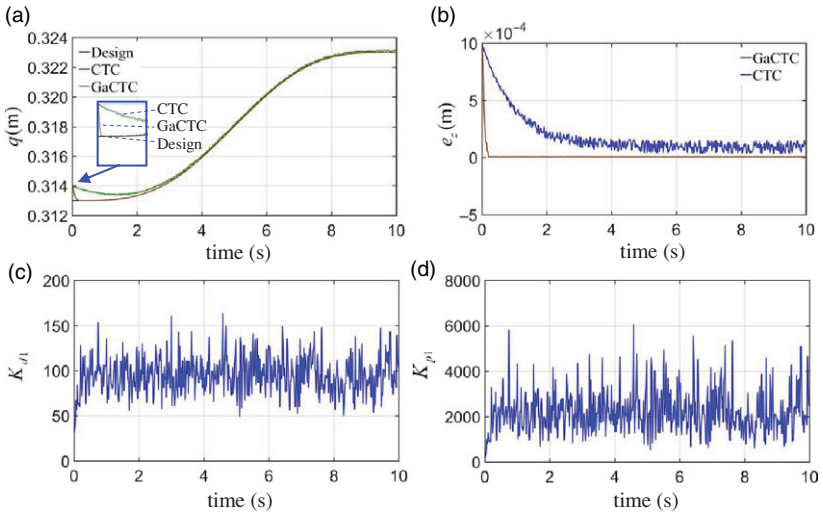

Fig. 6. Tracking error analysis along axis z (a) trajectory tracking, (b) tracking error, (c)K_{d1}, (d)K_{p1}.

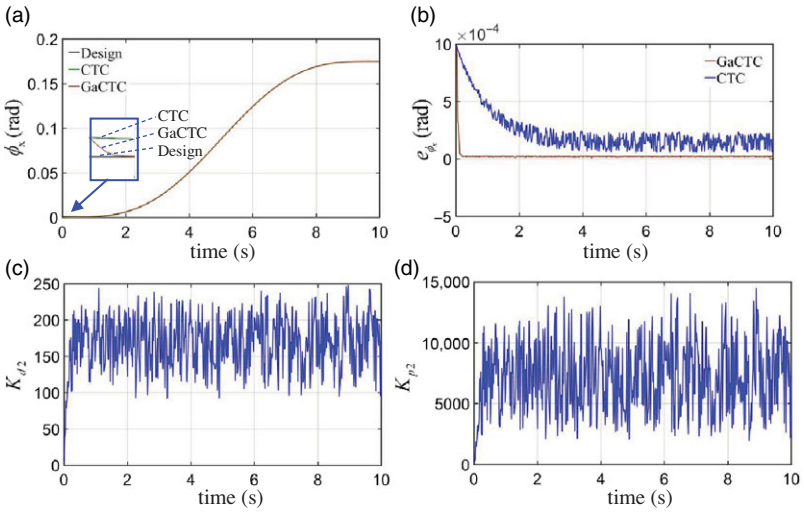

Fig. 7. Tracking error analysis along axis x rotation, (a) trajectory tracking, (b) tracking error, (c)K_{d2}, (d)K_{p2}.

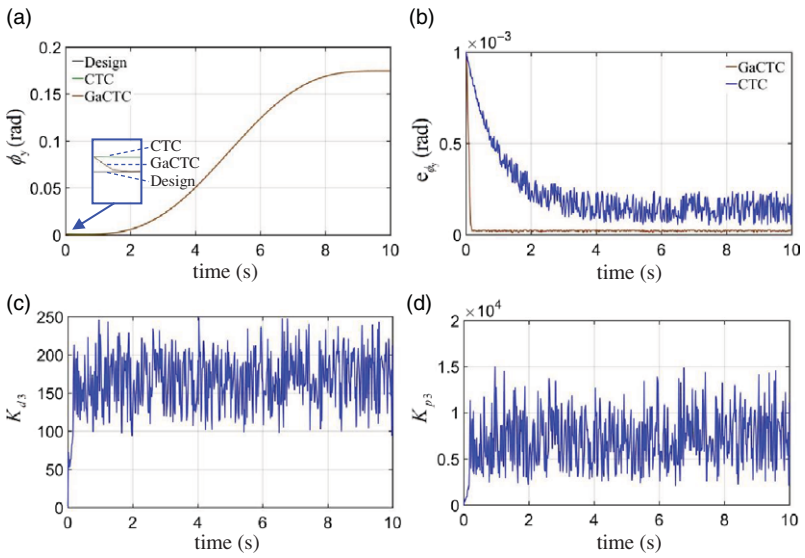

Fig. 8. Tracking error analysis along axis y rotation, (a) trajectory tracking, (b) tracking error, (c)K_{d3}, (d)K_{p3}.

https://doi.org/10.1017/S0263574720001368 Published online by Cambridge University Press

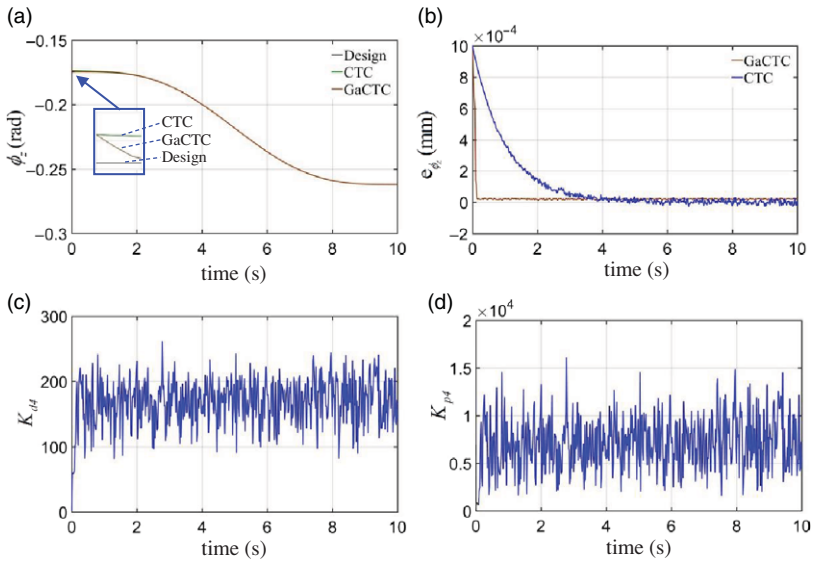

Fig. 9. Tracking error analysis along axis z rotation, (a) trajectory tracking, (b) tracking error, (c)K_{d4}, (d)K_{p4}.

(see images c and d in Figs. 6–9). Sangdani et al.[22] applied similar methods to vision-based tracker robot and achieved good tracking results in the experiment.

5. Conclusion

This paper presents the CTC method based on the GA to improve the trajectory tracking accuracy and convergence speed of the 3UPS-PS parallel robot. The dynamic model of the 3UPS-PS parallel manipulator is established according to the virtual work principle. The seventh-order polynomial trajectory planning method is applied to make the moving platform achieve the specified orientation without a flexible and rigid impact along the smooth curve. Composed of a nonlinear feed-forward loop and a self-gain tuning PD control feedback loop based on the GA, the control system of the robot is constructed. After analyzing the performance of the proposed control algorithm and the traditional CTC, the simulation results reveal that the control method proposed in this paper can not only enhance the anti-jamming ability of the control system and improve the trajectory tracking speed and accuracy of the 3UPS-PS parallel robot but also reduce the error fluctuation of the control system. Meanwhile, this control method has a simple structure and is suitable for other kinds of parallel robots.

Acknowledgments

The research is jointly supported by the National Natural Science Foundation of China (Grant No. 51375288), the Science and Technology Program of Guangdong Province (Grant No. 2020ST004), the Department of Education of Guangdong Province (Grant No. 2017KZDXM036), and the Special Project for Science and Technology Innovation Team of Foshan City (Grant No. 2018IT100052). The authors would also like to thank the anonymous reviewers for their very useful comments.

References

1. W. Haouas and R. Dahmouche, "A new 7-dof parallel robot with a foldable platform," *J. Mech. Robot.* **10**(4), 1–15 (2018).
2. Y. J. Zhao and G. Cheng, "Dimensional synthesis of a 3UPS-PRU parallel robot," *Robotica* **35**(12), 2319–2329 (2017).
3. N**.** Zhang, "Dynamictrajectory planning of a 3-DOF under-constrainedcable-drivenparallel robot," *Mech. Mach. Theory* **98**, 21–35 (2016).
4. R. Maldonado-Echegoyen, E. Castillo-Castaneda and M. A. Garcia-Murillo, "Kinematic and deformation analyses of a translational parallel robot for drilling tasks," *J. Mech. Sci. Technol.* **29**(10), 4437–4443 (2015).
5. Q. Meng, F. Xie and X. J. Liu, "Conceptual design and kinematic analysis of a novel parallel robot for high-speed pick-and-place operations," *Front. Mech. Eng.* **13**(2), 211–224 (2018).

https://doi.org/10.1017/S0263574720001368 Published online by Cambridge University Press

6. L. A. Zhang, J. Wan and Y. L. Tan, "Dimensional synthesis of Ahut-Delta parallel mechanism based on improved chaotic particle swarm algorithm," *Trans. Chin. Soc. Agric. Mach.* **46**(8), 344–351 (2015) (in Chinese).
7. J. C. Shin and C. W. Lee, "Rider's net moment estimation using control force of motion system for bicycle simulator," *J. Robot. Syst*. **21**(11), 597–607 (2004).
8. W. Dong,Z. J. Du, Y. Q. Xiao and X. G. Chen, "Development of a parallel kinematic motion simulator platform," *Mechatronics* **23**(1), 154–161 (2013).
9. W. W. Shang and S. Cong, "Nonlinear computed torque control for a high-speed planar parallel manipulator," *Mechatronics* **19**(6), 987–992 (2009).
10. M. Lashin, M. Fanni, A. M. Mohamed and T. Miyashita, "Dynamic modeling and inverse optimal PID with feed-forward control in H_∞, framework for a novel 3D pantograph manipulator," *Int. J. Control* **16**(1), 1–16 (2018).
11. W. W. Shang, S. Cong and S. L. Jiang, "Dynamic model based nonlinear tracking control of a planar parallel manipulator," *Nonlinear Dyn*. **60**(4), 597–606 (2010).
12. J. Wu, D. Wang and L. P. Wang, "A control strategy of a two degree-of-freedom heavy duty parallel manipulator," *J. Dyn. Syst. T ASME* **6**(137), 1–10 (2015).
13. A. Agarwal, C. Nasa and S. Bandyopadhyay, "Dynamic singularity avoidance for parallel manipulators using a task-priority based control scheme," *Mech. Mach. Theory* **96**, 107–126 (2016).
14. W. Peng, Z. Lin and J. Su, "Computed torque control-based composite nonlinear feedback controller for robot manipulators with bounded torques," *IET Control Theory A* **3**(6), 701–711 (2009).
15. H. Cheng, Y. K. Yiu and Z. X. Li, "Dynamics and control of redundantly actuated parallel manipulators," *IEEE-ASME T Mech*. **8**(4), 483–491 (2003).
16. A. Codourey, "Dynamic modeling of parallel robots for computed-torque control implementation," *Int. J. Robot. Res*. **17**(12), 1325–1336 (1998).
17. I. Davliakos and E. Papadopoulos, "Model-based control of a 6-dof electrohydraulic Stewart-Gough platform," *Mech. Mach. Theory* **43**(11), 1385–1400 (2008).
18. A. Zubizarreta, M. Marcos, I. Cabanes and C. Pinto, "A procedure to evaluate extended computed torque control configurations in the Stewart-Gough platform," *Robot. Auton. Syst*. **59**(10), 770–781 (2011).
19. Y. Chen, G. Y. Ma, S. X. Lin and J. Gao, "Adaptive fuzzy computed-torquecontrol for robot manipulator with uncertain dynamics," *Int. J. Adv. Robot. Syst.* **9**(6), 1–9 (2012).
20. X. C. Zhu, G. L. Tao, B. Yao and J. Cao, "Adaptive robust posture control of a pneumatic muscles by driven parallel manipulator with redundancy," *Automatica* **13**(4), 2248–2257 (2008).
21. Z. Y. Yang, J. Wu and J. P. Mei, "Motor-mechanism dynamic model based neural network optimized computed torque control of a high speed parallel manipulator," *Mechatronics* **17**(7), 381–390 (2007).
22. M. H. Sangdani, A. R. Tavakolpour-Saleh and A. Lotfavar, "Genetic algorithm-based optimal computed torque control of a vision-based tracker robot: Simulation and experiment," *Eng. Appl. Artif. Intel*. **67**, 24–38 (2018).
23. L. W. Tsai, "Solving the inverse dynamics of a Stewart-Gough manipulator by the principle of virtual work," *J. Mech. Des.* **122**(1), 3–9 (2000).
24. Y. J. Zhao and F. Gao, "Inverse dynamics of the 6-dof out-parallel manipulator by means of the principle of virtual work," *Robotica* **27**(2), 259–268 (2009).
25. J. Wu, T. M. Li and L. W. Guan, "Computed-torque control for a moving flight simulator platform," *J. Tsinghua Univ.* (*Sci. Tech.*) **46**(8), 1405–1408+1413 (2006) (in Chinese).

https://doi.org/10.1017/S0263574720001368 Published online by Cambridge University Press