Robotica (2023), **41**, pp. 1159–1178
doi:10.1017/S0263574722001606

CAMBRIDGE UNIVERSITY PRESS

RESEARCH ARTICLE

Robust PID optimal tuning of a Delta parallel robot based on a hybrid optimization algorithm of particle swarm optimization and differential evolution

Yong-Ju Pak, Yong-Su Kong* and Jin-Song Ri

Faculty of Electronics and Automation, Kim Il Sung University, Pyongyang, Democratic People's Republic of Korea
*Corresponding author. E-mail: ys.kong0428@ryongnamsan.edu.kp

Received: 23 February 2022; **Revised:** 12 September 2022; **Accepted:** 5 October 2022;
First published online: 12 December 2022

Keywords: Delta parallel robot, particle swarm optimization, differential evolution, robust PID control, computed torque control

Abstract

In this paper, we propose an approach to tune optimal parameters of a robust PID controller by means of computed torque control (CTC) strategy for trajectory tracking of a Delta parallel robot, using a hybrid optimization algorithm of Particle Swarm Optimization (PSO) and differential evolution (DE). It differs from previous works that they propose robust PID controller parameters tuning based on conventional gradient-based optimization algorithms and apply them to process control. First, we reduce the tuning problem of a robust PID controller with CTC strategy satisfying requirements including robustness and disturbance attenuation to an optimization problem with nonlinear constraints by considering the nonlinear dynamic model of a Delta parallel robot. Second, we set up the design characteristics for the trajectory tracking of a Delta parallel robot. Then, we propose a robust PID controller in a way of obtaining the global optimization solution of the nonlinear optimization problem by running a PSO-DE hybrid optimization algorithm of finding the global optimal solution by maintaining the diversity of swarm during evolution based on the evolution of cognitive experience. Simulation and experimental results demonstrate that the proposed controller outperforms previous works with respect to robust performance and active disturbance attenuation when it is applied to tracking control of a Delta parallel robot.

1. Introduction

Recent years have witnessed ever-increasing applications of high-speed pick-and-place parallel robots such as Delta parallel robot in fields of electronics, food, pharmaceutical, packaging and other light industries [1–3]. For the parallel robots, it is difficult to design the control system with high control performance due to the presence of nonlinearity in the dynamic model, the high interaction and the kinematical constraints between the robot links [1].

To enhance the trajectory tracking performance of a robot manipulator, some traditional control strategies have been employed such as the independent PID control on each axis, the PID control with an acceleration feedforward component, the PID control with gravity compensation and nonlinear PID-based control [4, 5]. Since these strategies never or partially use the dynamic model of the Delta parallel robot, dynamic interaction between robot links could not be considered. Hence, one cannot expect the high control performance when high speed and accuracy of a parallel robot are of concern.

Otherwise, the computed torque control (CTC) strategy has been proposed which employs all the dynamic model of the manipulator [6]. However, the CTC strategy provides a good control performance only when the obtained model is accurate enough, and it does not consider model uncertainties and external disturbances that affect the control performances of the robot manipulator.

Robust control strategies such as H infinity control theory have been suggested to achieve the good control performances despite the presence of model uncertainties and external disturbances. But the controller design and implementation based on the approach is complex [7, 8].

Therefore, some researchers have investigated the robust control techniques using PID controller with simple and easy-to-implement structure. These techniques mainly focus on how to tune the parameters of the PID controller so that the closed-loop system guarantees robustness and disturbance attenuation. In refs. [9–12], the authors discussed the controller parameter tuning by using system identification method. In ref. [13], PID controller tuning for a planar parallel robot is stated as a nonlinear optimization problem which is solved by differential evolution (DE) algorithm. However, these techniques consider only one or two factors such as the stability and tracking performance to tune the controller parameters.

Recently, active researches on the fractional order PID control have been conducted. The fractional order PID controller provides the improvement of stability and performance of the system although model uncertainties and external disturbances appear. A typical method of the fractional order PID controller tuning is Monje's method [14, 15]. It has an advantage of tuning a controller so that the closed-loop system meets several conditions regarding robustness to plant uncertainties, load disturbances and high-frequency noise. In Monje's method, tuning is reduced to a nonlinear optimization problem with nonlinear constraints. However, it is unlikely to obtain the accurate global minimum. Normally, the optimization problem was solved out by using the function FMINCON in MATLAB Optimization Toolbox, which employs the traditional gradient-based paradigm [2, 14, 15].

For a nonlinear optimization problem, however, the gradient-based approaches reveal the drawbacks of the premature convergence to local minimum, high sensitivity to selection of the initial condition and the failure of convergence when discontinuities exist. Therefore, the obtained solution may not be a global optimal solution. It implies that it is necessary to overcome these issues. This motivates the study of this paper.

To the best of the authors' knowledge, the robust PID controller design approaches based on Monje's method are mainly applied to process control [16, 17]. In this paper, to enhance the convergence in Monje's method, we propose a method based on a hybrid algorithm of particle swarm optimization (PSO) and DE rather than FMINCON used in previous works. This PSO-DE hybrid optimization algorithm has advantage of global search capability which was shown in refs. [7, 8].

The main contribution of this paper is to design a robust PID controller that meets some requirements such as robustness, high-frequency noise attenuation and disturbance rejection by using PSO-DE and to apply it to the trajectory tracking control of Delta parallel robot. In this paper, the design problem of the robust PID controller for the parallel robot has been investigated in the following stages:

- The design specifications are set concretely.
- The design problem is referred to an optimization problem with several constraints regarding design specifications using Monje's method.
- To obtain global solution, this optimization problem is solved using the PSO-DE.
- For comparison, another PID controller is also tuned using the method in ref. [2].

Simulation and Experimental results show that the designed robust PID controller has a robust performance and active disturbance attenuation when it is applied to tracking control of the Delta parallel robot. The rest of this paper is organized as follows. Section 2 shows the structure, the kinematic and dynamic models of the developed Delta parallel robot. Section 3 presents the control structure of robust PID controller with CTC strategy, design specifications and PSO-DE-based tuning algorithm of the robust PID controller. Simulation and experimental results are discussed in Section 4. Finally, conclusions are given in Section 5.

2. Modeling of the Delta parallel robot

The Delta parallel robot is a 3-degree-of-freedom robot, which consists of a fixed platform, a mobile platform and three identical RUU (Revolution, Universal, Universal) legs between the fixed platform

https://doi.org/10.1017/S0263574722001606 Published online by Cambridge University Press

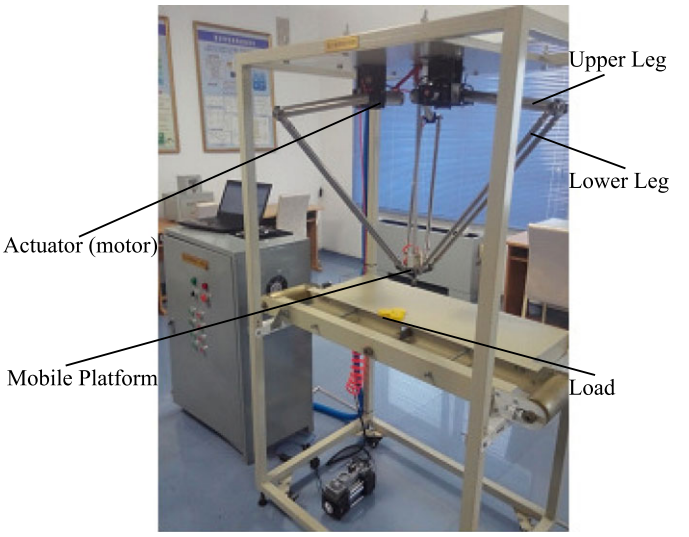

Figure 1. *Prototype of the Delta parallel robot.*

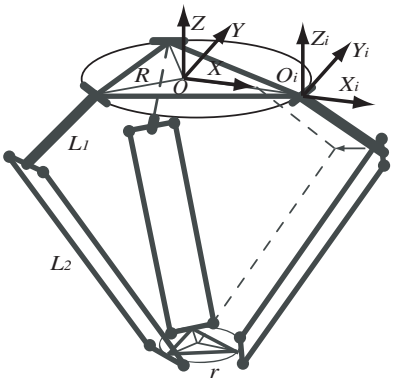

Figure 2. *The base coordinate systems and physical parameters.*

and the mobile one. The revolute joints are actuated by fixed rotational actuators. Since actuators are placed in the fixed platform, the Delta parallel robot can achieve high speed and acceleration [2, 18].

Figure 1 shows prototype of the Delta parallel robot on which experiments were carried out. Figures 2 and 3 show physical parameters, a base coordinate system ($O - xyz$) and chain coordinate systems ($O_i - x_iy_iz_i$, $i = 1, 2, 3$) established in the Delta parallel robot. In addition, the physical parameters of the Delta parallel robot are listed in Table I.

As shown in Fig. 2, the robot has a base coordinate system $O - xyz$ located at the center of the fixed platform, in which z-axis is in the reverse direction of the gravity, x-axis is directed to the upper leg of the first kinematic chain and y-axis is chosen to be the right-handed coordinated system. Over each kinematic chain, a chain coordinate system $O_i - x_iy_iz_i$, $i = 1, 2, 3$) is located at distance R from $O - xyz$ and rotated an angle $\phi_i(0°, 120°, 240°)$, respectively.

For simplicity, upper legs and lower legs of the three kinematic chains are translated to the origin of the base coordinate system by radius r of the mobile platform, as shown in Fig. 3.

https://doi.org/10.1017/S0263574722001606 Published online by Cambridge University Press

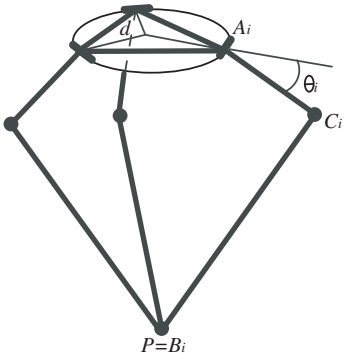

Figure 3. *The structure after simplification.*

2.1. Kinematic model

In general, the kinematic model of the robot consists of the inverse kinematic model and the direct kinematic model. The kinematic model of delta robot has been widely researched [2, 18–20].

Given the coordinates $P = [x_p,\ y_p,\ z_p]^{\mathrm{T}}$ of the mobile platform in the base Cartesian coordinate system, the inverse kinematic model computes actuated revolute joint angles $\theta = [\theta_1, \theta_2, \theta_3]^{\mathrm{T}}$ of upper legs. The inverse kinematic model can be summarized as follows [2]. Given the coordinates $P = [x_p,\ y_p,\ z_p]^{\mathrm{T}}$ of the mobile platform on the $O - xyz$, its position in each chain coordinate system $O_i - x_i y_i z_i$ $(i = 1, 2, 3)$ is converted using a rotation transformation matrix as follows:

$$\begin{bmatrix} x_{p_i} \\ y_{p_i} \\ z_{p_i} \end{bmatrix} = \begin{bmatrix} \cos\phi_i & -\sin\phi_i & 0 \\ \sin\phi_i & \cos\phi_i & 0 \\ 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} x_p \\ y_p \\ z_p \end{bmatrix}, \quad \left(\phi_i = 0, \frac{2\pi}{3}, \frac{4\pi}{3};\ i = 1, 2, 3\right) \tag{1}$$

Since the upper leg link L_1 can rotate only on the plane $x_i O_i z_i$, the joint J_i is located on the circular trajectory with the center at A_i and a L_1 radius.

$$(x_i - d)^2 + z_i^2 = L_1^2, \quad (i = 1, 2, 3) \tag{2}$$

Moreover, the joint J_i is also located on a spherical surface centered in B_i with radius L_2.

$$(x_i - x_{p_i})^2 + (y_i - y_{p_i})^2 + (z_i - z_{p_i})^2 = L_2^2, \quad (i = 1, 2, 3) \tag{3}$$

Then, J_i is located on the intersection between (2) and (3) in the following plane

$$y_i = 0, \quad (i = 1, 2, 3). \tag{4}$$

From (2) to (4), the position of joint J_i in each chain coordinate system can be stated as

$$x_{J_i} = \frac{(d - A_i B_i) \pm \sqrt{(1 + A_i^2)L_1^2 - (A_i d + B_i)^2}}{1 + A_i^2},\ y_{J_i} = 0,\ z_{J_i} = A_i x_{J_i} + B_i, \tag{5}$$

where

$$A_i = -\frac{d - x_{p_i}}{z_{p_i}}, B_i = -\frac{x_{p_i}^2 + y_{p_i}^2 + z_{p_i}^2 + L_1^2 - L_2^2 - d^2}{2z_{p_i}}$$

Considering [21, 22], we have the revolute joint angle of the upper leg as below:

$$\theta_i = \arctan 2\left(z_{J_i},\ x_{J_i} - d\right), i = 1, 2, 3. \tag{6}$$

In the Delta parallel robot, the direct kinematic model is the reverse process of the inverse kinematic model, which calculates the Cartesian position $P = [x_p, y_p, z_p]^{\mathrm{T}}$ of the mobile platform in the base coordinate system, given the revolute joint angles $\theta = [\theta_1, \theta_2, \theta_3]^{\mathrm{T}}$ of the upper legs.

https://doi.org/10.1017/S0263574722001606 Published online by Cambridge University Press

Table I. *Physical parameters for the Delta parallel robot.*

Symbol	Parameter	Value
R	Radius of the fixed platform (m)	0.155
r	Radius of the mobile platform (m)	0.045
d	$R-r$	0.11
L_1	Length of the upper leg (m)	0.3
L_2	Length of the lower leg (m)	0.65
θ	Joint position angle of the upper legs	$\theta=[\theta_1,\ \theta_2,\ \theta_3]$
I_m	Inertia moment of motor shaft (kg · m^2)	1.8×10^{-4}
m_1	Mass of the upper leg (kg)	0.457
m_{1j}	Mass of the upper leg joint (kg)	0.05
m_2	Mass of the lower leg (kg)	0.313
m_3	Mass of the mobile platform(kg)	0.406
m_4	Mass of the load (kg)	0.1

The position of joint J_i in each chain coordinate system is obtained as follows:

$$x'_{J_i}=d+L_1\cos\theta_i,\ y'_{J_i}=0,\ z'_{J_i}=-L_1\sin\theta_i. \tag{7}$$

Then, the position of J_i in base coordinate system is obtained using the matrix of revolution transformation as follows:

$$\begin{bmatrix}x_{J_i}\\ y_{J_i}\\ z_{J_i}\end{bmatrix}=\begin{bmatrix}\cos\phi_i & -\sin\phi_i & 0\\ \sin\phi_i & \cos\phi_i & 0\\ 0 & 0 & 1\end{bmatrix}^{-1}\begin{bmatrix}x'_{J_i}\\ y'_{J_i}\\ z'_{J_i}\end{bmatrix},\ \left(\phi_i=0,\ \frac{2\pi}{3},\ \frac{4\pi}{3};\ i=1,2,3\right). \tag{8}$$

Moreover, the joint J_i is also located on a spherical surface centered in p with radius L_2.

$$(x_{J_i}-x_p)^2+(y_{J_i}-y_p)^2+(z_{J_i}-z_p)^2=L_2^2,\ (i=1,2,3). \tag{9}$$

From (7) to (9),

$$\begin{cases}(x_{J_1}-x_p)^2+y_p^2+(z_{J_1}-z_p)^2=L_2^2\\ (x_{J_2}-x_{J_1})x_p+y_{J_2}y_p+(z_{J_2}-z_{J_1})z_p=\dfrac{1}{2}(\rho_2-\rho_1)\\ (x_{J_3}-x_{J_1})x_p+y_{J_3}y_p+(z_{J_3}-z_{J_1})z_p=\dfrac{1}{2}(\rho_3-\rho_1)\end{cases} \tag{10}$$

where $\rho_i=x_{J_i}^2+y_{J_i}^2+z_{J_i}^2,\ (i=1,2,3)$.

From last two equations of (10),

$$x_p=K_1z_p+H_1,\ y_p=K_2z_p+H_2, \tag{11}$$

where

$$K_1=\frac{y_{J_3}(z_{J_2}-z_{J_1})-y_{J_2}(z_{J_3}-z_{J_1})}{(x_{J_3}-x_{J_1})y_{J_2}-(x_{J_2}-x_{J_1})y_{J_3}},\ H_1=\frac{1}{2}\cdot\frac{y_{J_2}(\rho_3-\rho_1)-y_{J_3}(\rho_2-\rho_1)}{(x_{J_3}-x_{J_1})y_{J_2}-(x_{J_2}-x_{J_1})y_{J_3}}$$

$$K_2=\frac{(x_{J_2}-x_{J_1})(z_{J_3}-z_{J_1})-(x_{J_3}-x_{J_1})(z_{J_2}-z_{J_1})}{(x_{J_3}-x_{J_1})y_{J_2}-(x_{J_2}-x_{J_1})y_{J_3}},\ H_2=\frac{1}{2}\cdot\frac{(x_{J_3}-x_{J_1})(\rho_2-\rho_1)-(x_{J_2}-x_{J_1})(\rho_3-\rho_1)}{(x_{J_3}-x_{J_1})y_{J_2}-(x_{J_2}-x_{J_1})y_{J_3}}$$

From first equation of (10) and (11),

$$(1+K_1^2+K_2^2)z_p^2+2(K_1H_1+K_2H_2-K_1x_{J_1}-z_{J_1})z_p+(H_1-x_{J_1})^2+H_2^2+z_{J_1}^2-L_2^2=0. \tag{12}$$

https://doi.org/10.1017/S0263574722001606 Published online by Cambridge University Press

Then, the position of P is obtained as follows:

$$x_p = K_1 z_p + H_1,\ y_p = K_2 z_p + H_2,\ z_p = \frac{-b + \sqrt{b^2 - 4ac}}{2a}, \tag{13}$$

where

$$a = (1 + K_1^2 + K_2^2),\ b = 2(K_1 H_1 + K_2 H_2 - K_1 x_{J_1} - z_{J_1}),\ c = (H_1 - x_{J_1})^2 + H_2^2 + z_{J_1}^2 - L_2^2.$$

2.2. Dynamic model

The dynamic equation can be expressed as follows:

$$M(\theta)\ddot{\theta} + C(\theta, \dot{\theta})\dot{\theta} + N(\theta) = \tau, \tag{14}$$

where $M(\theta)$ is the inertia matrix, $C(\theta, \dot{\theta})$ is the Coriolis matrix, $N(\theta)$ is the gravity vector, and $\tau = [\tau_1, \tau_2, \tau_3]^{\mathrm{T}}$ is the torque vector applied at the upper leg joints.

The inertia matrix $M(\theta)$ in (14) is defined as

$$M(\theta) = \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix} = I_{bt} + (m_3 + m_4 + \frac{1}{3}m_2)J^T J, \tag{15}$$

where I_{bt} is the inertia matrix of the driving axis and is defined as follows:

$$I_{bt} = \begin{bmatrix} I_{bti} & 0 & 0 \\ 0 & I_{bti} & 0 \\ 0 & 0 & I_{bti} \end{bmatrix}, \quad I_{bti} = I_m + L_1^2 \left(\frac{m_1}{3} + m_{1j} + \frac{2}{3}m_2 \right).$$

And J in (15) is Jacobian matrix defined as follows:

$$J = -\begin{bmatrix} s_1^T \\ s_2^T \\ s_3^T \end{bmatrix}^{-1} \begin{bmatrix} s_1^T b_1 & 0 & 0 \\ 0 & s_2^T b_2 & 0 \\ 0 & 0 & s_3^T b_3 \end{bmatrix}, s_i = \begin{bmatrix} x \\ y \\ z \end{bmatrix} - T_i \left(\begin{bmatrix} d \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} L_1 \cos\theta_i \\ 0 \\ -L_1 \sin\theta_i \end{bmatrix} \right),$$

$$b_i = T_i \begin{bmatrix} L_1 \sin\theta_i \\ 0 \\ L_1 \cos\theta_i \end{bmatrix},\ T_i = \begin{bmatrix} \cos\phi_i & -\sin\phi_i & 0 \\ \sin\phi_i & \cos\phi_i & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (\phi_i = 0, \frac{2\pi}{3}, \frac{4\pi}{3};\ i = 1, 2, 3).$$

$X = [x,\ y,\ z]^T = f(\theta)$: Position of the mobile platform center

The matrix $C(\theta, \dot{\theta})$ in (14) is calculated as follows:

$$C(\theta, \dot{\theta}) = -J^T(m_3 + m_4 + \frac{1}{3}m_2) \begin{bmatrix} s_1^T \\ s_2^T \\ s_3^T \end{bmatrix}^{-1} \left(\begin{bmatrix} \dot{s}_1^T \\ \dot{s}_2^T \\ \dot{s}_3^T \end{bmatrix} J + M_{sb} \right), \tag{16}$$

$$M_{sb} = \begin{bmatrix} \dot{s}_1^T b_1 + s_1^T \dot{b}_1 & 0 & 0 \\ 0 & \dot{s}_2^T b_2 + s_2^T \dot{b}_2 & 0 \\ 0 & 0 & \dot{s}_3^T b_3 + s_3^T \dot{b}_3 \end{bmatrix}, \dot{s}_i = \dot{X} + b_i \dot{\theta}_i = J\dot{\theta} + b_i \dot{\theta}_i,$$

$$\dot{b}_i = T_i \begin{bmatrix} L_1 \cos\theta_i \\ 0 \\ -L_1 \sin\theta_i \end{bmatrix} \dot{\theta}_i,\ (i = 1, 2, 3).$$

https://doi.org/10.1017/S0263574722001606 Published online by Cambridge University Press

In addition, $N(\theta)$ in (14) is calculated as follows:

$$N(\theta) = J^T\left(m_3 + m_4 + \frac{3}{2}m_2\right)\begin{bmatrix}0\\0\\g\end{bmatrix} - L_1\left(\frac{1}{2}m_1 + m_{1j} + \frac{1}{2}m_2\right)g\begin{bmatrix}\cos\theta_1\\\cos\theta_2\\\cos\theta_3\end{bmatrix}. \tag{17}$$

3. Robust PID control based on a PSO-DE hybrid optimization algorithm

3.1. Control structure for robust PID controller with CTC strategy

The CTC is a control strategy, which obtains a linearized and decoupled model of robot manipulator using a feedback linearization approach. Applying this strategy, the robot system can be controlled by the linear control techniques.

Consider the following linearization law:

$$\tau = \hat{M}(\theta)u + \hat{C}(\theta,\ \dot{\theta})\dot{\theta} + \hat{N}(\theta), \tag{18}$$

where u is an input of new system and $\hat{M}(\theta)$, $\hat{C}(\theta,\ \dot{\theta})$, $\hat{N}(\theta)$ are estimations of $M(\theta)$, $C(\theta,\ \dot{\theta})$, $N(\theta)$, respectively.

Assuming that Eq. (18) is exact enough to describe a dynamic model of the robotic system

$$M(\theta) = \hat{M}(\theta),\ C(\theta,\dot{\theta}) = \hat{C}(\theta,\dot{\theta}),\ N(\theta) = \hat{N}(\theta).$$

Then, we get the Eq. (19) from (14) and (18).

$$\hat{M}(\theta)u = M(\theta)\ddot{\theta}. \tag{19}$$

If the inertia matrix $M(\theta)$ is invertible, the dynamic model (14) is linearized and decoupled.

$$\ddot{\theta}(t) = u(t). \tag{20}$$

The obtained system is a linearized and decoupled system for joint variables $\theta(t)$.

In practice, however, model uncertainties and external disturbances should not be ignored, which affect the control performances of robotic system. These influences and errors can be attenuated by the feedback control for linear system (20).

For the linearized system (20), the following PID control strategy is proposed:

$$u(t) = \ddot{\theta}_d(t) + k_p e(t) + k_i\int e(t)dt + k_d\dot{e}(t), \tag{21}$$

where $e(t) = \theta_d(t) - \theta(t)$ is the joint error, $\dot{e}(t)$ is the joint velocity error, $\ddot{\theta}_d(t)$ is the desired joint acceleration, and k_p, k_i and k_d are the controller parameters to be determined.

If we replace the ideal derivative controller in (21) with the real derivative controller, then the resulting PID controller can be expressed as follows:

$$C(s) = K_p\left(1 + \frac{1}{T_i s} + T_d s\right)\frac{1}{T_f s + 1}, \tag{22}$$

where K_p is the proportional coefficient, T_i and T_d are the integral and derivative time constants, respectively. In addition, an output first-order filter with the time constant T_f makes the PID controller proper, and it filters the high-frequency noise. T_f is determined as in ref. [23].

$$T_f = \min\{T_i/10,\ T_d/10\}. \tag{23}$$

Figure 4 shows the control structure of the robust PID controller with CTC strategy for the Delta parallel robot. Most controllers for Delta parallel robot use joint sensors that allow us to easily measure joint angles. The trajectory is planned in Cartesian space, and the trajectory in joint space is obtained using inverse kinematic model. The controller uses the trajectory in joint space and measured joint position. Then, the position in Cartesian space is obtained using direct kinematic model.

https://doi.org/10.1017/S0263574722001606 Published online by Cambridge University Press

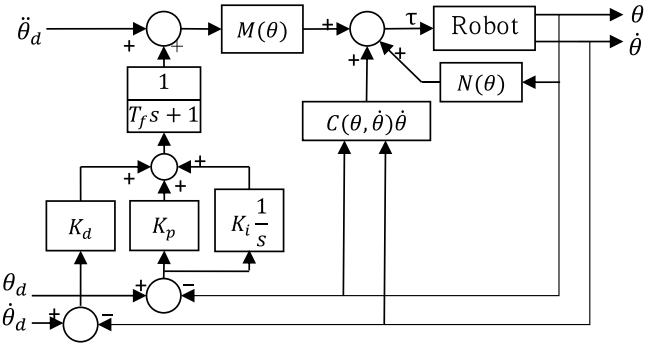

Figure 4. *Structure for the robust PID controller with CTC strategy for Delta parallel robot.*

In Fig 4, θ is measured joint position and θ_d is desired joint position. The error in joint position is reduced by individual PID controller with first-order filter. And the torque vector τ is calculated using (18). This calculation is performed in low-level controller equipped with ARM processor.

3.2. Design specifications for the robust PID controller

In this section, we discuss how to design the robust PID controller with CTC strategy to simultaneously meet several specifications regarding frequency domain.

Robustness to gain variations of the plant is one of the most common specifications that attribute performance of the robust PID controller [24]. The robustness forces the phase of the open-loop system to be flat at the gain crossover frequency ω_{cg} and to be almost constant within an interval around ω_{cg}. Then, the system will be more robust to the gain variations and the overshoot of the transient response is almost constant within the range of gain variations.

Therefore, the robust PID controller design can be referred to solve an optimization problem, which minimizes the following cost function as follows for each possible solution [14].

$$J(K_p, T_i, T_d) = \int_{\omega_i}^{\omega_f} e_{FD}^2(\omega)d\omega = \int_{\omega_i}^{\omega_f} (\varphi(\omega) - (\varphi_m - \pi))^2 d\omega, \tag{24}$$

where error $e_{FD}(\omega)$ is calculated in a predefined interval between ω_i and ω_f around ω_{cg}. $\varphi(\omega)$ is the phase of the open-loop system at frequency ω and φ_m is the phase margin.

The advantage of this approach is that the frequency band $[\omega_i, \omega_f]$ in which the closed-loop system is robust to the gain variations can be selected according to user requirements. More specifications can be included in the cost function (24) with the following constraints [14, 15].

- The gain crossover frequency ω_{cg} must be within frequency interval $[\omega_{cgl}, \omega_{cgh}]$, and the phase margin must be at least φ_m. Then, these constraints can be described as follows:

$$\omega_{cgl} \le \omega_{cg} \le \omega_{cgh}, \tag{25}$$

$$\arg\left(C(j\omega_{cg})G(j\omega_{cg})\right) \ge -\pi + \varphi_m, \tag{26}$$

  where $C(j\omega_{cg})$ and $G(j\omega_{cg})$ are transfer functions of controller (22) and plant (20), respectively.

- To attenuate the high-frequency noise, the complementary sensitivity function needs to fulfill the following condition:

$$\left| T(j\omega) = \frac{C(j\omega)G(j\omega)}{1 + C(j\omega)G(j\omega)} \right|_{\text{dB}} \le M_t \text{ dB}, \ \forall\omega \ge \omega_t \text{ rad/s}, \ |T(j\omega_t)|_{\text{dB}} = M_t \text{ dB}, \tag{27}$$

  where M_t means the desired noise attenuation for frequencies $\omega \ge \omega_t$ rad/s.

https://doi.org/10.1017/S0263574722001606 Published online by Cambridge University Press

- To ensure a good output disturbance rejection, the sensitivity function $S(j\omega)$ is constrained by the following condition:

$$\left|S(j\omega) = \frac{1}{1 + C(j\omega)G(j\omega)}\right|_{\text{dB}} \leq M_s \text{ dB}, \forall\omega \leq \omega_s \text{rad/s}, |S(j\omega_s)|_{\text{dB}} = M_s \text{ dB}, \tag{28}$$

where M_s is the desired value of the sensitivity function for frequencies $\omega \leq \omega_s$.

Therefore, the design problem of the robust PID controller can be referred to a nonlinear optimization problem with nonlinear constraints as follows:

$$\min \ J(K_p, \ T_i, \ T_d) = \int_{\omega_i}^{\omega_f} (\varphi(\omega) - (\varphi_m - \pi))^2 d\omega. \tag{29}$$

Subject to (25)-(28).

In refs. [1, 15], the FOPID controller design is referred to as the similar nonlinear optimization problem, and it was solved using the function FMINCON of MATLAB. For the optimization problem (29), however, it is hard to find the global optimal solution with traditional approximation optimization methods such as gradient method since the cost function and constraints are complex nonlinear function on the design parameters (K_p, T_i, T_d).

Therefore, in the next section, the nonlinear optimization problem with nonlinear constraints is to be solved effectively by a PSO-DE hybrid optimization algorithm, which is one of the swarm intelligent optimization techniques [7, 8].

3.3. PSO-DE-based tuning algorithm of the robust PID controller

The feature of the PSO algorithm is to achieve the optimal solution by sharing of information among particles in the swarm. It is assumed that the swarm consists of NP particles in D-dimensional search space (i.e., solution space). Each particle is a D-dimensional vector and its position represents a candidate solution in the solution space. Since it needs to identify three parameters in consideration of (29) in our case, the dimension of the particle is $D = 3$.

If j is the current time step (generation), then the position of the i-th particle in the swarm can be represented as $\eta_j^i = (\eta_j^{i,1}, \ \eta_j^{i,2}, \ \ldots \ , \eta_j^{i,D})$, and it corresponds to

$$\eta_j^i = [K_p^i, \ T_i^i, \ T_d^i]$$

considering (22) and (24)-(29), where the subscript j is omitted for description brevity. The velocity of each particle is denoted by $V_j^i = (V_j^{i,1}, \ V_j^{i,2}, \ \ldots \ , V_j^{i,D})$. The best current position (i.e., the best personal position of the particle) of the i-th particle is denoted by $P_j^i = (p_j^{i,1}, \ p_j^{i,2}, \ \ldots \ , p_j^{i,D})$, and it is known as the cognitive experience. The best particle in the swarm (that is, in minimization problems, the particle with the smallest cost function value) is indicated by $P_j^{best} = (p_j^{best,1}, \ p_j^{best,2}, \ \ldots \ , p_j^{best,D})$, and it is known as the social experience.

We discuss the PSO version proposed by Clerc and Kennedy [25], which uses a new parameter χ, known as the constriction factor that improves the convergence. For each time step j (generation), the velocities and the positions of each particle in the swarm are represented as follows:

$$V_{j+1}^i = \chi(V_j^i + c_1 r_1(P_j^i - \eta_j^i) + c_2 r_2(P_j^{best} - \eta_j^i)), \tag{30}$$

$$\eta_{j+1}^i = \eta_j^i + V_{j+1}^i, \tag{31}$$

for $i = 1, \ 2, \ \ldots \ , NP$, where χ is the above-mentioned constriction factor parameter, c_1 and c_2 are positive constants which are referred to as cognitive and social parameters, respectively, r_1 and r_2 denote random numbers that are uniformly distributed in [0, 1] and are randomly chosen in each generation. This scheme is typically utilized for the constant $\chi = 0.72984$ and $c_1 = c_2 = 2.05$ [25, 26].

https://doi.org/10.1017/S0263574722001606 Published online by Cambridge University Press

The DE algorithm is a population-based stochastic parallel direct search method, and it requires few control parameters. However, it has better convergence behavior than other well-known algorithms such as Evolution Algorithm [26]. Like PSO, the population consists of NP individuals that are the D-dimensional vector and is initialized at random in the search space. The individuals are evolved over successive iterations (generations) to find the optimal solution of the cost function in the solution space. At each generation, the evolution of individuals is conducted through mutation, crossover and selection operations [27].

Nowadays, numerous PSO-DE hybrid approaches have been developed to improve the accuracy of the optimal solution and the convergence behavior of PSO and DE algorithms. The evolutionary algorithms are known to be still efficient, but it requires a great computational cost due to a large number of function evaluations. The incorporation of DE algorithm in each generation of PSO algorithm may cause an increase of the function evaluations. So, we adopt a PSO-DE hybrid algorithm based on the evolution of cognitive and social experiences in order to suggest an approach of the parameter tuning of the robust PID controller, since it costs smaller amount of computation along with better convergence behavior than other hybrid versions [7, 8].

In the tuning algorithm, we define a set of the best personal experience (i.e., the cognitive experiences) of each particle at the j-th time step (generation) as $S_j^p = (P_j^1,\ P_j^2,\ \ldots\ , P_j^{NP})$. After each time step of PSO algorithm according to (30) and (31), DE procedure of one generation is applied to the elements $P_j^i,\ i = 1,\ \ldots\ , NP$, in the S_j^p set (i.e., the cognitive experience of each particle is evolved). It implies that the best personal positions $P_j^i,\ i = 1,\ \ldots\ , NP$ correspond to the individuals $x_j^i,\ i = 1,\ \ldots\ , NP$, of the DE algorithm. The DE mutation operation uses individuals from the S_j^p set. In particular, three evolution steps of DE algorithm (mutation, crossover and selection) are applied only to the 'promising' cognitive experiences which have been changed (i.e., improved) during the previous PSO step. From the viewpoint of the computational cost, this turns out to be more effective than the procedure of evolving all particles in the swarm.

The algorithmic scheme of the proposed approach to tune the parameters of the robust PID controller for the Delta parallel robot is as follows:

1: Individuals in the swarm (i.e., the controller parameters $\eta_{j0}^i = [K_{p0}^i,\ T_{i0}^i,\ T_{d0}^i], i = 1,\ \ldots\ , NP$) are randomly initialized in the given range. The population size NP should be chosen in the range from 5 to 10 times D. In order to improve the global convergence, it is set to $NP = 50$.

2: For each particle $\eta_{j0}^i = [K_{p0}^i,\ T_{i0}^i,\ T_{d0}^i](i = 1,\ \ldots\ , NP)$, if it agrees with inequalities of the stability and constraints (25)-(28), initialize the particle again.

3: For each particle $\eta_{j0}^i = [K_{p0}^i,\ T_{i0}^i,\ T_{d0}^i](i = 1,\ \ldots\ , NP)$, compute the cost function value $J(\eta_0^i)$ using (29) and let the best personal position P_0^i be the value. Then, let the social experience P_0^{best} be the minimum of $P_0^i,\ i = 1,\ 2,\ \ldots\ , NP$.

4:**for** each time step j **do**

5:**for** each particle i in the swarm **do**

6: **Update** the particle $\eta_j^i = [K_{pj}^i,\ T_{ij}^i,\ T_{dj}^i]$ (parameter vector of the controller) using (30) and (31). If the updated parameter vector η_j^i disagrees with the inequalities constraints (25)-(28), then updating process should be repeated until it does.

7: **Evaluate** particle $\eta_j^i = [K_{pj}^i,\ T_{ij}^i,\ T_{dj}^i]$, *i.e.*, compute the cost function value for the current parameter vector $\eta_j^i = [K_{pj}^i,\ T_{ij}^i,\ T_{dj}^i]$ using (29).

8: **If** $J(\eta_j^i) < J(P_{j-1}^i)$, **then** update the cognitive experience as $P_j^i = \eta_j^i$, **else** $P_j^i = P_{j-1}^i$ (i.e., do not update value of the previous generation). Also, **if** $J(P_j^i) < J(P_{j-1}^{best})$, **then** update the social experience as $P_j^{best} = P_j^i$, else $P_j^{best} = P_{j-1}^{best}$.

9: **If** P_j^i has been changed **then**

10: /* Evolve P_j^i utilizing one DE step in S_j^p */

11: **Mutate** P_j^i and generate the corresponding mutant vector v_j^i through the mutation strategy.

12: **Crossover** the mutant vector v_j^i and target vector P_j^i, and generate the corresponding trial vector u_j^i.

https://doi.org/10.1017/S0263574722001606 Published online by Cambridge University Press

Table II. *Parameter values of PSO-DE, PSO and DE algorithms.*

Parameter	Value	Parameter	Value
D	3	c_2	2.05
NP	50	F	0.5
χ	0.72984	CR	0.9
c_1	2.05	$g^*_{\max}$	50

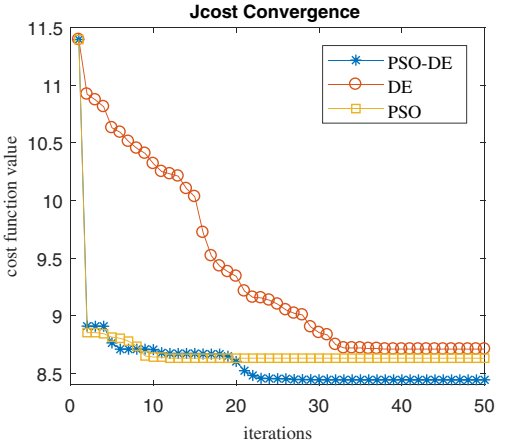

Figure 5. *Convergence behavior of the tuning algorithm based on PSO-DE.*

13: **Evaluate** the trial vector u_j^i, that is, under the trial vector u_j^i agrees with the inequalities constraints (25)-(28) (If it disagrees with the constraints, then mutation and crossover should be repeated until it does), compute the cost function value for u_j^i using (29).

14: **If** $J(u_j^i) < J(P_j^i)$, **then** update the best personal position P_j^i. Also, **if** $J(P_j^i) < J(P_j^{best})$, **then** update the best position P_j^{best}.

15: **end if**

16: **end for**

17: **end for**

4. Experimental results and analysis

4.1. Robust PID controller design for the Delta parallel robot

Considering the specifications of the Delta parallel robot and results in refs. [14, 23], the design specifications in the frequency domain mentioned in the inequalities constraints (25)-(28) of the optimization problem (29) are listed in (a)-(e):

(a) Minimum phase margin φ_m is equal to 45°.

(b) Gain crossover frequency ω_{cg} must be in the interval $[\omega_{cgl},\ \omega_{cgh}] = [50\,\text{rad/s}, 150\,\text{rad/s}]$.

(c) An interval $(\omega_f - \omega_i)$ where the phase is intended to be flat about the gain crossover frequency is equal to 0.25 dec.

(d) High-frequency noise attenuation: $\omega_t = 350\,\text{rad/s},\ M_t = -15\,\text{dB}$ that is, $\forall\omega \geq \omega_t,\ |T(j\omega)|_{\text{dB}} \leq -15\,\text{dB}$.

(e) Disturbance rejection: $\omega_s = 10\,\text{rad/s},\ M_s = -15\,\text{dB}$, that is, $\forall\omega \leq \omega_s,\ |S(j\omega)|_{\text{dB}} \leq -15\,\text{dB}$.

https://doi.org/10.1017/S0263574722001606 Published online by Cambridge University Press

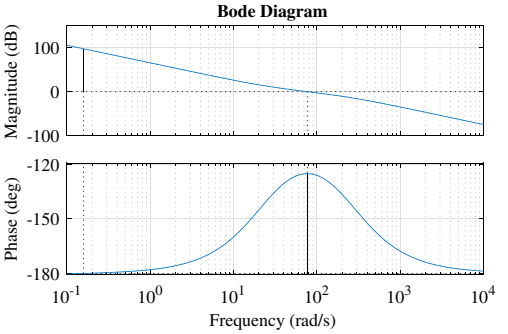

Figure 6. *Bode diagram of open-loop system ($PM = 54.9°$, $\omega_{cg} = 76.6\,\text{rad/s}$).*

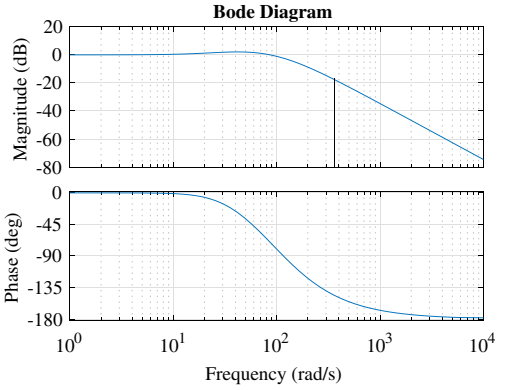

Figure 7. *Frequency specification of the complementary sensitivity function $T(j\omega)$ ($\omega_t = 350\,\text{rad/s}$, $|T(j\omega_t)|_{\text{dB}} = -17.1\,\text{dB}$).*

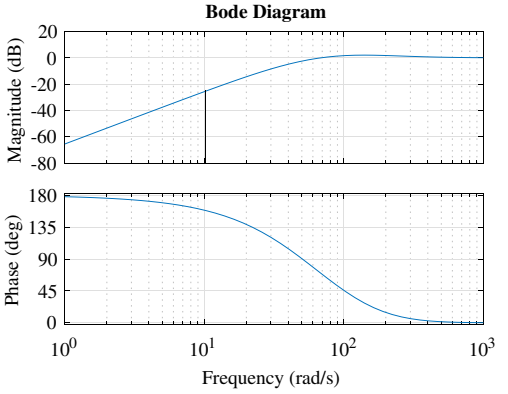

Figure 8. *Frequency specification of the sensitivity function $S(j\omega)$ ($\omega_s = 10\,\text{rad/s}$, $|S(j\omega_s)|_{\text{dB}} = -25.6\,\text{dB}$).*

In order to find the parameters (K_p, T_i, T_d, T_f) of the robust PID controller, this problem is converted into an optimization problem (29). And the controller is designed by the algorithm mentioned in the previous section. All the parameters used in our PSO-DE-based tuning algorithm are listed in Table II. PSO algorithm and DE algorithm also use the same parameters in Table II.

https://doi.org/10.1017/S0263574722001606 Published online by Cambridge University Press

Table III. *Path point of control simulation trajectory for Delta robot.*

Path point	Pass time(s)	Position (m)	Path point	Pass time(s)	Position (m)
Waypoint 1	0	(0, 0, −0.508)	Waypoint 10	6.3	(0, 0.2, −0.75)
Waypoint 2	0.7	(0, 0, −0.6)	Waypoint 11	7	(−0.2, 0, −0.6)
Waypoint 3	1.4	(0.2, 0, −0.75)	Waypoint 12	7.7	(0, −0.2, −0.75)
Waypoint 4	2.1	(0, 0, −0.6)	Waypoint 13	8.4	(−0.2, −0.2, −0.75)
Waypoint 5	2.8	(−0.2, 0, −0.75)	Waypoint 14	9.1	(0, 0, −0.6)
Waypoint 6	3.5	(0, 0.2, −0.75)	Waypoint 15	9.8	(0.2, 0.2, −0.75)
Waypoint 7	4.2	(0, 0, −0.6)	Waypoint 16	10.5	(−0.2, 0.2, −0.75)
Waypoint 8	4.9	(0, −0.2, −0.75)	Waypoint 17	11.2	(0, 0, −0.6)
Waypoint 9	5.6	(0.2, 0, −0.6)	Waypoint 18	12	(0.2, −0.2, −0.75)

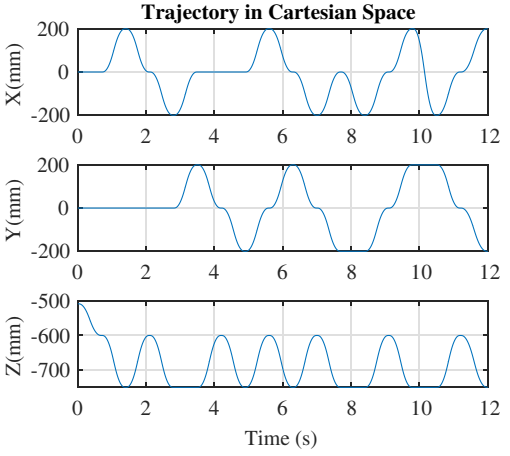

Figure 9. *Trajectories of x, y, z axes in the Cartesian space.*

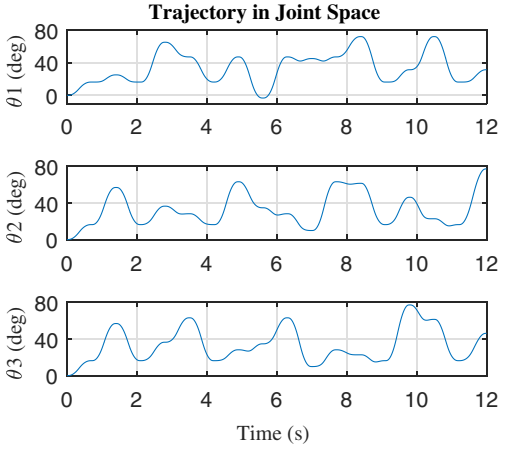

Figure 10. *Desired trajectory $\theta_d(t)$ in joint space.*

https://doi.org/10.1017/S0263574722001606 Published online by Cambridge University Press

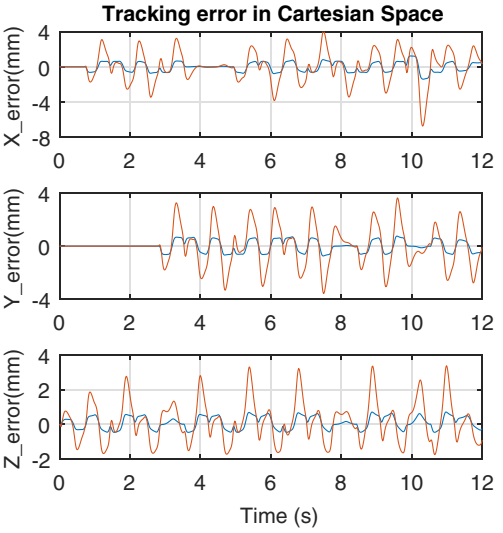

Figure 11. *Tracking error in Cartesian space (Orange: method in ref.* [2], *blue: robust PID).*

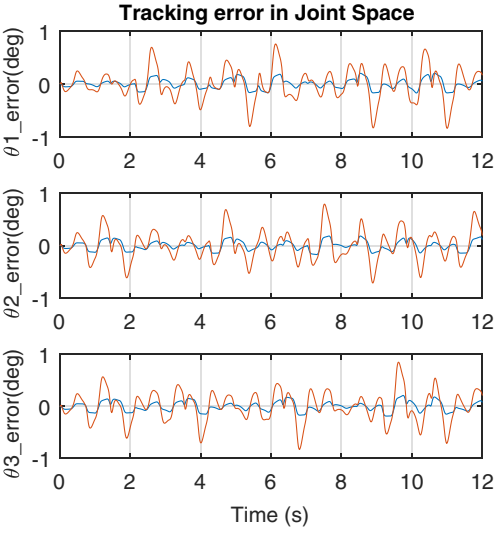

Figure 12. *Tracking error in joint space (Orange: method in* [2], *blue: robust PID).*

Figure 5 shows the convergence behaviors of the robust PID controller tuning algorithm for the Delta parallel robot based on the PSO-DE, DE and PSO. PSO-DE has better global search capability than PSO and DE.

After 50 generations, the parameters of the robust PID controller are obtained as follows:

$$K_p = 1853.4,\ T_i = 1073.7,\ T_d = 0.0413,\ T_f = 0.0041. \tag{32}$$

The bode diagram of the open-loop system of designed controller is shown in Fig. 6. Two diagrams demonstrate that the designed controller meets the design specifications (a), (b) and (c).

The frequency responses of the complementary sensitivity $T(j\omega)$ and the sensitivity function $S(j\omega)$ of the designed closed-loop system are shown in Figs. 7 and 8, respectively. As shown in these figures,

https://doi.org/10.1017/S0263574722001606 Published online by Cambridge University Press

Table IV. *Simulation results.*

	RMS position error			**RMS joint error**		
	X	***Y***	***Z***	***θ1***	***θ2***	***θ3***
Robust PID	0.539	0.423	0.370	0.095	0.090	0.092
Method in ref. [2]	1.662	1.388	1.118	0.283	0.268	0.280

Table V. *Real mass of the Delta robot.*

Symbol	**Parameter**	**Value**
m_{1r}	Real Mass of the upper leg (kg)	0.357
m_{1jr}	Real Mass of the upper leg joint (kg)	0.1
m_{2r}	Real Mass of the lower leg (kg)	0.353
m_{3r}	Real Mass of the mobile platform (kg)	0.456
m_{4r}	Real Mass of the load (kg)	0.3

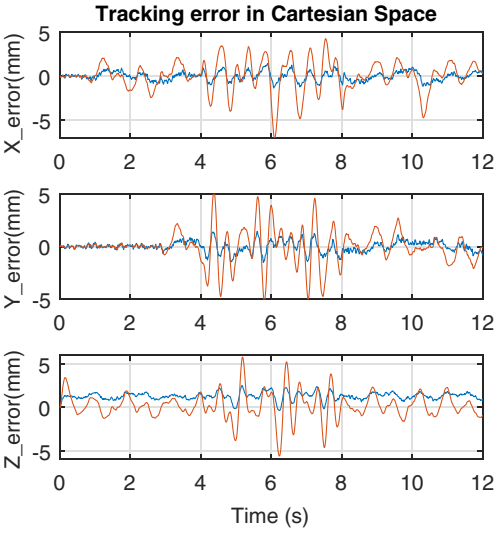

Figure 13. *Tracking error in Cartesian space in robustness analysis simulation (Orange: method in ref.* [2]*, blue: robust PID).*

the complementary sensitivity function meets the design specification (d) and the sensitivity function meets the design specification (e).

Consequently, we can make sure that the robust PID controller designed by the proposed approach satisfies the above-mentioned design specifications in frequency domain.

4.2. Simulation results

In order to analyze the control performance of the designed robust PID controller in the time domain, a trajectory planning was implemented using the modified trapezoidal acceleration curve method, in which acceleration and jerk is small for pick-and-place task [28, 29].

https://doi.org/10.1017/S0263574722001606 Published online by Cambridge University Press

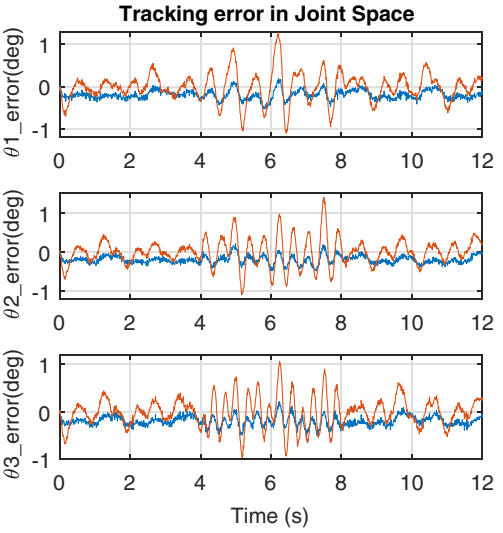

Figure 14. *Tracking error in joint space in robustness analysis simulation (Orange: method in ref.* [2]*, blue: robust PID).*

Table III shows the waypoints of the track for the control simulation of the Delta parallel robot. The desired trajectories on x, y, z axes and in joint space are shown in Figs. 9 and 10, respectively.

The desired trajectories on x, y, z axes are obtained from waypoints in Table III using modified trapezoidal acceleration curve method, and the trajectories in joint are obtained from trajectories on task space using inverse kinematic model.

The following waypoints are determined as considering pick-and-place tasks in workspace.

To analyze the performance of the robust PID controller designed in the previous section, the MATLAB/Simulink model is built using the physical parameters of the Delta parallel robot in Table I, the nonlinear dynamic model (14) and the control structure of Fig. 4.

Another robust PID controller with CTC [2] is implemented and used for comparative analysis. The design specifications of the closed-loop system are set as follows:

$$\zeta = 0.9,\ \ t_s = 0.3\,\text{s},\ \ \omega_n = 14.2\,\text{rad/s},\ \ \ \beta = 15.$$

Then, the obtained parameters are

$$K_p = 583.42, T_i = 3012.1, T_d = 40.5073, T_f = 0.0069.$$

Figures 11 and 12 show that the tracking errors of the controllers in this paper and [2] and Table IV show the RMS of tracking errors. As shown in these figures and table, the controller in this paper has a smaller tracking error for the desired trajectory.

To analyze the robustness of controllers, real mass of delta robot is setup in Table V, which is different from Table I. Moreover, we assume that random noises of [-0.1 deg, 0.1 deg] arise in feedback loop and disturbances are present in the torque driving each upper leg in interval [4 s, 8 s].

$$0.5\sin(10t),\ \ 0.5\sin(15t),\ \ 0.5\sin(20t).$$

Figures 13 and 14 show tracking errors of controllers in this paper and in ref. [2] for robustness analysis. Table VI shows RMS of tracking errors. As shown in these figures and table, the controller proposed in this paper has better robustness and good disturbance attenuation.

https://doi.org/10.1017/S0263574722001606 Published online by Cambridge University Press

Table VI. *Simulation results for robustness.*

	RMS position error			**RMS joint error**		
	X	***Y***	***Z***	***θ1***	***θ2***	***θ3***
Robust PID	0.518	0.502	0.449	0.225	0.227	0.226
Method in [2]	1.739	1.570	1.529	0.344	0.343	0.316

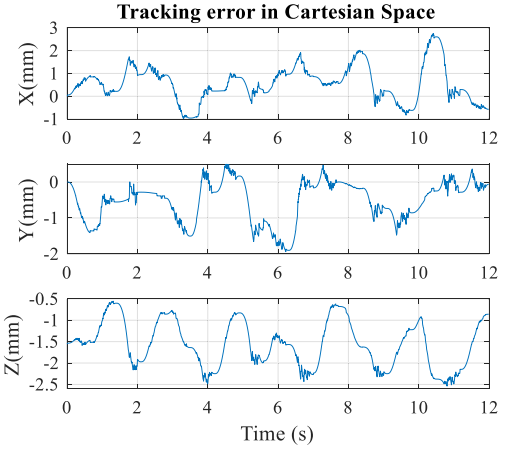

Figure 15. *Tracking error in Cartesian space when* $m_4 = 100$ g.

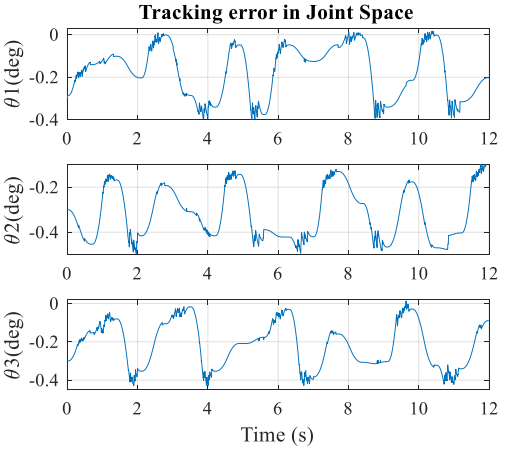

Figure 16. *Tracking error in the joint space when* $m_4 = 100$ g.

4.3. Experimental results

First, we built the Delta parallel robot to be controlled by the robust PID controller designed in Section 4.1. Then, the tracking performances for the desired trajectory in the Cartesian space and the joint space are shown in Figs. 15 and 16 when mass of load is 100 g. Figures 17 and 18 show tracking errors when mass of load is 300 g. Table VII shows RMS of tracking errors.

https://doi.org/10.1017/S0263574722001606 Published online by Cambridge University Press

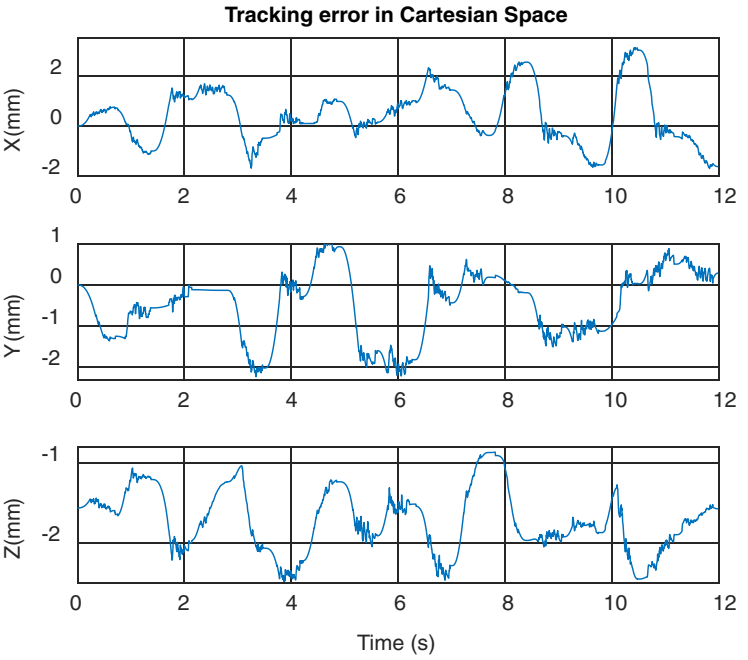

Figure 17. *Tracking error in Cartesian space when* $m_4 = 300$ g.

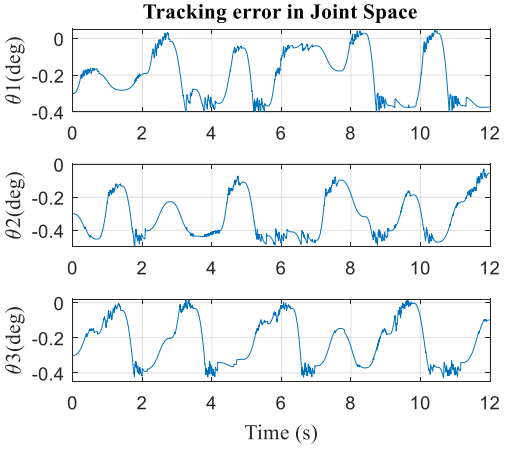

Figure 18. *Tracking error in the joint space when* $m_4 = 300$ g.

In conclusion, the robust PID controller designed in this paper tracks the desired trajectory with a small error, and a slight tracking error occurs when the desired value changes. At the same time, our experimental results show a slight increase of the tracking error than that of simulation results. It follows from the fact that the control performance is affected by nonlinearities as backlash of reduction gears, which is connected with the driving motors. If the load mass is less than 100g, the position tracking error is less than ± 3.5 mm.

https://doi.org/10.1017/S0263574722001606 Published online by Cambridge University Press

Table VII. *Experiment results.*

	RMS position error			RMS joint error		
	X	***Y***	***Z***	***θ1***	***θ2***	***θ3***
Mass of load = 0.1 kg	0.951	0.770	1.648	0.210	0.338	0.244
Mass of load = 0.3 kg	1.172	0.940	1.760	0.245	0.343	0.259

5. Conclusions

In this paper, we propose an approach to tune an optimal robust PID controller with CTC strategy for trajectory tracking of a Delta parallel robot by adopting a PSO-DE hybrid optimization algorithm. Considering the nonlinear dynamic model of the Delta parallel robot, the design problem of a robust PID controller with CTC strategy that guarantees robustness and disturbance attenuation has been formulated as a nonlinear optimization problem with nonlinear constraints. To obtain a more accurate global optimum of the nonlinear optimization problem, an algorithm based on a PSO-DE hybrid optimization algorithm has been newly proposed. Simulation and experimental results carried out on the Delta parallel robot show that the designed robust PID controller has robust performance and active disturbance attenuation despite the presence of model uncertainties and external disturbances.

The proposed method needs much calculations, but it is solved by hardware that has good computing power. FOPID has 5 parameters, so FOPID has better performances such as robustness than PID. In future works, the FOPID and ADRC (active disturbance rejection control) are proposed and applicated in other parallel robots.

Author contributions. Kong Yong-Su conceived and designed study. Ri Jin-Song and Pak Yong-Ju made the delta robot. Kong Yong-Su, Ri Jin-Song and Pak Yong-Ju designed and simulated control system. Ri Jin-Song and Pak Yong-Ju made experiments and verified effectiveness of new method approached in this paper. Kong Yong-Su and Pak Yong-Ju wrote the article.

Financial support. This research received no specific grant from any funding agency, commercial or not-for-profit sectors.

Conflicts of interest. The authors declare no conflicts of interest exist.

Ethical approval. Not applicable.

References

[1] R. Clavel, Device for the movement and positioning of an element in space, Google Patents (1990).
[2] L. Angel and J. Viola, "Fractional order PID for tracking control of a parallel robotic manipulator type delta," *ISA Trans.* **79**, 172–188 (2018).
[3] C. Liu, G.-H. Cao, Y.-Y. Qu and Y.-M. Cheng, "An improved PSO algorithm for time-optimal trajectory planning of delta robot in intelligent packaging," *Int. J. Adv. Manuf. Technol.* **107**(3), 1091–1099 (2020).
[4] Y. X. Su, B. Y. Duan, C. H. Zheng, Y. F. Zhang, G. D. Chen and G. D. Mi, "Disturbance rejection high-precision motion control of a Stewart platform," *IEEE Trans. Control Syst. Technol.* **12**(3), 364–374 (2004).
[5] H. D. Taghirad *Parallel Robots: Mechanics and Control* (CRC, Brighton, London, 2013).
[6] Z. Asier, M. Marga, C. Itziar and P. Charles, "A procedure to evaluate extended computed torque control configurations in the stewart? Gough platform," *Robot Auton. Syst.* **59**(10), 770–781 (2011).
[7] Y. S. Kong, D. Zhao, B. Yang, C. Han and K. Han, "Non-fragile multiobjective static output feedback control of vehicle active suspension with time-delay," *Veh. Syst. Dyn. Int. J. Veh. Mech. Mobility* **52**(7), 948–968 (2014).
[8] Y. Kong, D. Zhao, B. Yang, C. Han and K. Han, "Robust non-fragile H_∞/L_2-L_∞ control of uncertain linear system with time-delay and application to vehicle active suspension," *Int. J. Robust Nonlinear Control* **25**(13), 2122–2141 (2015).
[9] W. Shang, S. Cong and F. Kong, "Identification of dynamic and friction parameters of a parallel manipulator with actuation redundancy," *Mechatronics* **20**(2), 192–200 (2010).
[10] J. Wu, J. Wang, L. Wang and T. Li, "Dynamic formulation of redundant and nonredundant parallel manipulators for dynamic parameter identification," *Mechatronics* **19**(4), 586–590 (2009).
[11] R. Luiz, Q. Isabelle, G. Germain, P. Francois and T. Sophie, "Identification and vibration attenuation for the parallel robot Par2," *IEEE Trans. Control Syst. Technol.* **22**(1), 190–200 (2014).

https://doi.org/10.1017/S0263574722001606 Published online by Cambridge University Press

[12] Q. Zhao, P. Wang and J. Mei, "Controller parameter tuning of delta robot based on servo identification," *Chin. J. Mech. Eng.* **28**(2), 267–275 (2015).
[13] G. Miguel and A. Jaime, "Off-line PID control tuning for a planar parallel robot using DE variants," *Exp. Syst. Appl.* **64**, 444–454 (2016).
[14] C. A. Monje, Y. Q. Chen, D. Xue, B. M. Vinagre and V. Feliu, "Fractional-order Systems andControls," **In:** *Fundamentals and Applications*, Advances in Industrial Control (Springer-Verlag, London, 2010).
[15] C. A. Monje, B. M. Vinagre, V. Feliu and Y. Chen, "Tuning and auto-tuning of fractional order controllers for industry applications," *Control Eng. Pract.* **16**(7), 798–812 (2008).
[16] J. D. J. Rubio, G. Ochoa, R. Balcazar and J. Pacheco, "Disturbance rejection in two mechatronic systems," *IEEE Lat. Am. Trans.* **14**(2), 485–492 (2016).
[17] J. Viola and L. Angel, "Factorial design for robustness evaluation of fractional PID controllers," *Lat. Am. Trans. IEEE (Revista IEEE Am. Lat. May)* **13**(5), 1286–1293 (2015).
[18] N. Liu and J. Wu, "Kinematics and application of a hybrid industrial robot-Delta-RST," *Sens. Transducers* **169**(4), 186–192 (2014).
[19] V. Noppeney, T. Boaventura and A. Siqueira, "Task-space impedance control of a parallel delta robot using dual quaternions and a neural network," *J. Braz. Soc. Mech. Sci.* **43**(9), 440–447 (2021).
[20] C. Liu, G. Cao and Y. Qu, "Safety analysis via forward kinematics of delta parallel robot using machine learning," *Safety Sci.* **117**, 243–249 (2019).
[21] C. Ayiz and S. Kucuk, "The Kinematics of Industrial Robot Manipulators Based on the Exponential Rotational Matrices," **In:** *IEEE International Symposium on Industrial Electronics* (2009) pp. 977–982.
[22] S. Kucuk and B. D. Gungor, "Inverse Kinematics Solution of a New Hybrid Robot Manipulator Proposed for Medical Purposes," **In:** *Medical Technologies National Congress*, (Medical Technologies National Congress, 2016).
[23] P. Fabrizio and V. Antonio *Advances in Robust Fractional Control* (Springer, 2015) pp. 34–35.
[24] Y. Q. Chen and K. L. Moore, "Relay feedback tuning of robust PID controllers with iso-damping property," *IEEE Trans. Syst. Man Cybern Part B* **35**(1), 23–31 (2005).
[25] M. Clerc and J. Kennedy, "The particle swarm - explosion, stability, and convergence in a multidimensional complex space," *IEEE Trans. Evol. Comput.* **6**(1), 58–73 (2002).
[26] M. G. Epitropakis, V. P. Plagianakos and M. N. Vrahatis, "Evolving cognitive and social experience in particle swarm optimization through differential evolution: A hybrid approach," *Inf. Sci.* **216**, 50–92 (2012).
[27] D. Swagatam and N. S. Ponnuthurai, "Differential evolution: A survey of the state-of-the-art," *IEEE Trans. Evol. Comput.* **15**(1), 4–31 (2011).
[28] M. Afroun, T. Chettibi and S. Hanchi, "Planning Optimal Motions for a DELTA Parallel Robot," **In:** *14th Mediterranean Conference on Control and Automation* (2006) pp. 28–30.
[29] H. Xiuqing and S. Lei, "Trajectory Planning and Simulation of a New Symmetric Parallel Mechanism with Three Translational DOF," **In:** *International Conference on Measuring Technology and Mechatronics Automation* (2009) pp. 424–427.

Cite this article: Y.-J. Pak, Y.-S. Kong and J.-S. Ri (2023). "Robust PID optimal tuning of a Delta parallel robot based on a hybrid optimization algorithm of particle swarm optimization and differential evolution", Robotica **41**, 1159–1178. https://doi.org/10.1017/S0263574722001606