SCIENCE CHINA
Technological Sciences

•Article•

June 2021 Vol.64 No.6: 1177–1188
https://doi.org/10.1007/s11431-020-1738-5

An iterative learning method for realizing accurate dynamic feedforward control of an industrial hybrid robot

WU Jun[1,2*], ZHANG BinBin[1,2], WANG LiPing[1,2] & YU Guang[1,2]

[1] *State Key Laboratory of Tribology and Institute of Manufacturing Engineering, Department of Mechanical Engineering, Tsinghua University, Beijing 100084, China;*
[2] *Beijing Key Lab of Precision/Ultra-precision Manufacturing Equipment and Control, Beijing 100084, China*

Received August 12, 2020; accepted October 16, 2020; published online March 23, 2021

Feedforward control based on an accurate dynamic model is an effective approach to reduce the dynamic effect of the robot and improve its performance. However, due to the complicated work environment with considerable uncertainty, it is difficult to obtain a high-precision dynamic model of the robot, which severely deteriorates the achievable control performance. This paper proposes an iterative learning method to accurately design the industrial feedforward controller and compensate for the external uncertain dynamic load of the robot. Based on a standard dynamic model, a complete linear feedforward controller is presented. An iterative design strategy is given to iteratively update the feedforward controller by combining the Moore-Penrose Inverse and the PID learning rate. Experiments are carried out on a 5-DOF industrial hybrid robot to validate the effectiveness of the proposed iterative learning method. The experiment results illustrate that the industrial feedforward controller can rapidly converge to the optimal controller and significantly improve the servo performance by using the proposed method. This paper provides an effective method for applying iterative learning control to an unopened industrial control system. It is very useful for the practical control of hybrid robots in industrial field.

hybrid robot, feedforward control, iterative learning control, parameters design, industrial control system

Citation: Wu J, Zhang B B, Wang L P, et al. An iterative learning method for realizing accurate dynamic feedforward control of an industrial hybrid robot. Sci China Tech Sci, 2021, 64: 1177–1188, https://doi.org/10.1007/s11431-020-1738-5

1 Introduction

As the counterpart of serial robots, parallel robots have some advantages such as high acceleration, high stiffness-to-weight ratio, and low manufacturing cost [1–3]. But the small workspace limits their wide applications. Hybrid robots combining the advantages of series robots and parallel robots have been proposed and increasingly used in the fields of machine tools, pick-and-place manipulation, and walking robots [4–6]. However, hybrid robots belong to strong coupling and nonlinear mechanisms whose dynamics vary with the configuration of the robot in its large workspace [7]. As a result, it is difficult to guarantee the high motion performance of the hybrid robot in the whole workspace [8]. Thus, it is necessary to consider the robot dynamics to design the control system. Many control approaches based on the dynamic model are proposed to consider the dynamic effect and realize high-precision motion control.

The dynamic feedforward control plus feedback control, as a widely used model-based control approach, is an effective method to improve the motion accuracy of multi-degree-of-freedom (DOF) systems such as multi-DOF robots and multi-axis CNC machine tools [9–11]. Veronesi and Visioli [12] utilized an identified feedback and feedforward control system to control a quadruple tanks apparatus and the motion error of the device is reduced. Lipiński [13] proposed a

* Corresponding author (email: jhwu@mail.tsinghua.edu.cn)

model-based feedforward controller, and used it to control a redundantly actuated 3RRR parallel manipulator with a perfectly identified dynamic model. However, in industrial applications, it is difficult to achieve the same ideal performance as theoretical results. The main reason is that the high-precision control system model and the robot dynamic model cannot be achieved even if system identification technology and other modeling technology are used. Therefore, due to the inaccurate control system model and dynamic model, the designed control parameters which directly relate to the feedforward control effective are not optimal for the feedforward control to compensate the dynamic effect [14–16].

Iterative learning control (ILC) is a typical learning method that can effectively improve the transient response and tracking performance of uncertain dynamic systems [17,18]. The current control signals are iteratively updated based on the previous motion error signals and the last input [19–21]. By using ILC, the robot performance can be improved through repeated iterations until the motion error reaches a minimum [22,23]. A distinct advantage of ILC over model-based control method is that it does not need to establish an accurate robot dynamic model. However, in industrial environment, the typical industrial servo system does not have an ILC, and thus the ILC cannot be directly realized in a robot. Taking both advantages of dynamic model-based feedforward and ILC, it is a good approach that uses basis functions to reflect dynamic behaviors and an ILC algorithm to design the control parameters [20].

In this paper, an iterative learning method for realizing accurate dynamic feedforward control of an industrial hybrid robot is proposed. According to the standard dynamic model of a hybrid robot, a complete feedforward controller is designed which includes all the feedforward compensation signals that the industrial system may provide. Then the iteration equation is derived and the PID learning rate is used to optimize the iterative speed. Based on these works, the ILC is realized offline by using a post-processing program in industrial environment. Finally, experiments are performed to validate the effectiveness of the integrated iterative learning method, and the method is applied to a 5-DOF industrial hybrid robot.

The rest of the paper is organized as follows. The complete feedforward control system is analyzed based on the robot dynamic model in Section 2. Section 3 investigates the iterative learning method and iteration strategy. Section 4 deals with the experiments, and some concluding remarks follow in Section 5.

2 Dynamic model and control system analysis

2.1 Dynamic model analysis

For a multi-DOF hybrid robot shown in Figure 1, its dynamic model can be obtained by using the Lagrangian dynamic equation. Through the equivalent transformation, the dynamic model can be written in the following compact form:

$$\boldsymbol{\tau} = \mathbf{M}\ddot{\boldsymbol{\theta}} + \begin{bmatrix} \dot{\boldsymbol{\theta}}^{\mathrm{T}}\mathbf{H}_1 \\ \vdots \\ \dot{\boldsymbol{\theta}}^{\mathrm{T}}\mathbf{H}_n \end{bmatrix} \dot{\boldsymbol{\theta}} + \mathbf{G} + \mathbf{f}, \tag{1}$$

where n is the joint number of the robot, $\boldsymbol{\tau}$ is the driving force, $\mathbf{M}$ is the inertial matrix of the robot, $\mathbf{H}_1$–$\mathbf{H}_n$ are the Hessian matrices of joint 1 to n which are the velocity matrices of Coriolis and centrifugal forces, $\mathbf{G}$ is the gravitational term, $\mathbf{f}$ is the friction vector, and $\ddot{\boldsymbol{\theta}}$ and $\dot{\boldsymbol{\theta}}$ are the acceleration and velocity vectors of n joints, respectively.

It is assumed that the friction consists of Coulomb friction and viscous friction. The friction can be expressed as

$$\mathbf{f} = \mathbf{f}_{\mathrm{v}}\dot{\boldsymbol{\theta}} + \mathbf{f}_{\mathrm{c}}\mathrm{sgn}(\dot{\boldsymbol{\theta}}), \tag{2}$$

where $\mathbf{f}_{\mathrm{v}}$ is the vector of viscous friction coefficient, and $\mathbf{f}_{\mathrm{c}}$ is the vector of the Coulomb friction coefficient.

Combining eqs. (1) and (2), the dynamic model can be rewritten as

$$\boldsymbol{\tau} = \mathbf{D}_1\ddot{\boldsymbol{\theta}} + \mathbf{D}_2\dot{\boldsymbol{\theta}} + \mathbf{D}_3\boldsymbol{\theta} + \mathbf{D}_4, \tag{3}$$

where $\mathbf{D}_1$, $\mathbf{D}_2$, $\mathbf{D}_3$, and $\mathbf{D}_4$ are acceleration, velocity, position, and constant correlation matrices, respectively.

The multi-DOF robot is a complex nonlinear system. For nonlinear system, the high-speed and high-precision control is still difficult. Considering the application convenience and reliability, the dynamic feedforward control plus feedback control is still the popular control method in industrial robots. In general, the control parameters for the feedback controller and feedforward controllers are tuned on a given configuration of the robot, and then the control parameters are used for the whole workspace of the robot. It is equivalent that the robot is regarded as a linear system to study its control problem. Since the control parameters keep unchanged, the dynamic characteristics in the given configuration are used to represent the dynamic characteristics of the robot in the whole workspace. However, the dynamic characteristics maybe have a great change in the whole workspace. In this paper, $\mathbf{D}_1$, $\mathbf{D}_2$, $\mathbf{D}_3$, and $\mathbf{D}_4$ in the dynamic model are calculated by taking the corresponding average value of these matrices within the whole workspace or on a given trajectory.

2.2 Control system analysis

For a multi-DOF hybrid robot shown in Figure 1, it is usually controlled by a feedback controller and a feedforward controller. Since the control of each joint is similar, a typical industrial control system of one active joint is given in Figure 2. The control system includes the control plant, a feedback controller, a feedforward controller, and an iterative learning

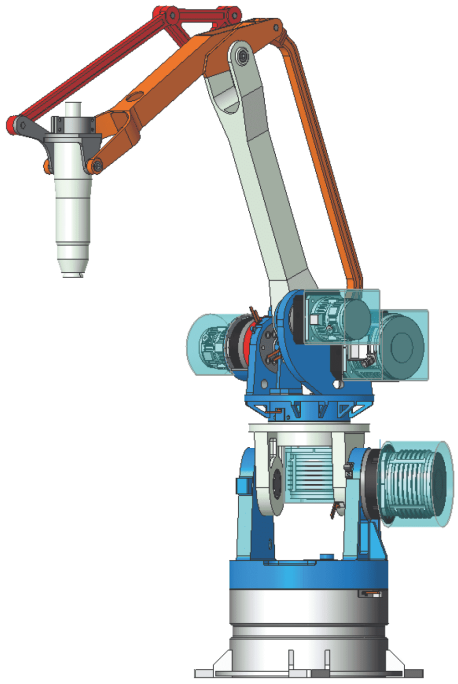

Figure 1 (Color online) A multi-DOF hybrid robot.

controller.

In Figure 2, θ_{ci} and θ_{ai} are the command position signal and the actual position signal of the joint i, respectively. $F_i(s)$ is the feedback controller. $P_{ik}(s)$ is feedforward controller transfer functions of the kth trial. The subscript k denotes the current number of the trial. $Q_i(s)$ is the dynamic transfer function based on the dynamic model eq. (3).

It should be noticed that because the industrial control system is not open and cannot be changed, the ILC cannot be realized directly. In Figure 2, the ILC is implemented by using a post-processing program. Based on the error signals and iteration equations, the feedforward control parameters are updated offline by using a post-processing program.

The complete iterative learning flow is shown in Figure 3. After the motion error signal is collected and stored in error memory, a new feedforward controller is obtained by adding the previous feedforward control signal and the iterative learning signal learned from the stored error signal. Next, the new feedforward controller is updated to the control system and stored in memory. Finally, the robot is controlled to generate new motion errors. Repeat the iteration until the error is minimal.

2.3 Feedforward controller design

It is well accepted that the hybrid robot cannot obtain good tracking performance with only kinematic-model based control. The control based on dynamic model is necessary for the robot to reduce the external dynamic disturbance when the robot works at high speed and high acceleration. Thus, the dynamic feedforward compensation would be designed based on the dynamic model shown in eq. (3).

For a n-DOF robot, its dynamic model is related to its n active links and gravity. According to the standard dynamic model eq. (3), the dynamic model $Q_i(s)$ shown in Figure 2 is divided into n+1 parts ($Q_{i1}(s)$–$Q_{in}(s)$ and C_{ai}). $Q_{i1}(s)$–$Q_{in}(s)$ and C_{ai} are the dynamic transfer functions based on the dynamic model eq. (3). Specifically, $Q_{ij}(s)$ is a transfer function from the motion of joint j to the coupling effect of joint i. C_{ai} is a constant external load on joint i. In order to compensate the dynamic effect, the feedforward controller $P(s)$ is also divided into n+1 parts ($P_{i1k}(s)$–$P_{ink}(s)$ and C_{cik}). $P_{i1k}(s)$–$P_{ink}(s)$ and C_{cik} are feedforward compensation transfer functions of the kth trial. Specifically, $P_{ijk}(s)$ is a transfer function from the motion of joint j to the compensation signal of joint i. C_{cik} is a constant compensation signal of joint i. Therefore, the control system can be updated, as shown in Figure 4.

Therefore, the transfer function $P_{ijk}(s)$ should be designed according to the dynamic model $Q_{ij}(s)$. From Figure 4, the output signal of the system can be expressed as

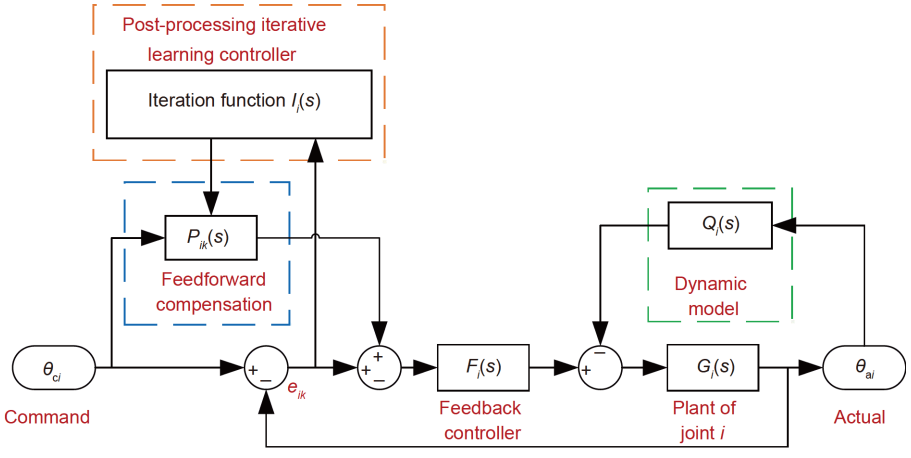

Figure 2 (Color online) Typical industrial control system with iterative learning controller.

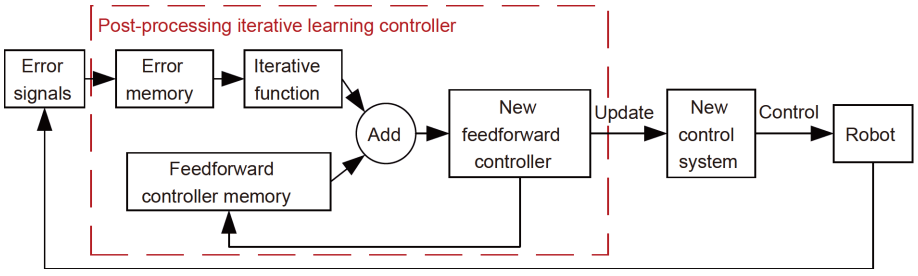

Figure 3 (Color online) Complete iterative learning flow.

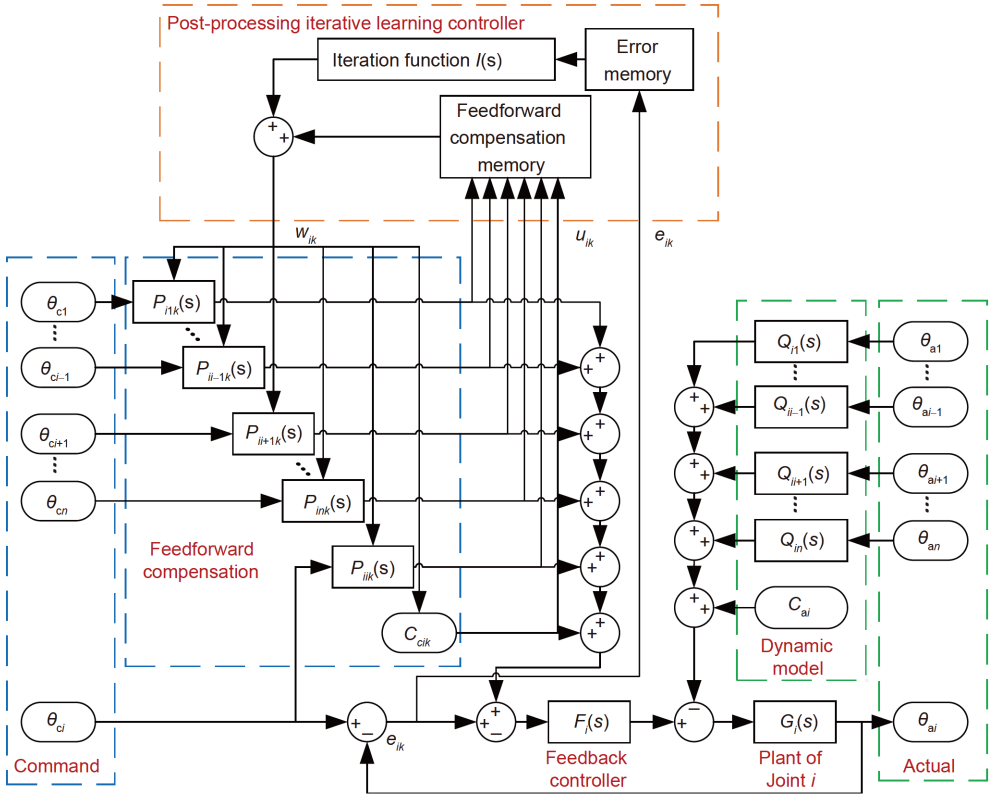

Figure 4 (Color online) Updated typical industrial control system.

$$\boldsymbol{\theta}_{\mathrm{a}} = \mathbf{G}(\mathbf{F}(\mathbf{P}_k\boldsymbol{\theta}_{\mathrm{c}} + \mathbf{C}_{ck} + \mathbf{E}_k) - (\mathbf{Q}\boldsymbol{\theta}_{\mathrm{a}} + \mathbf{C}_{\mathrm{a}})), \tag{4}$$

where $\boldsymbol{\theta}_{\mathrm{a}} = \left[\boldsymbol{\theta}_{\mathrm{a1}} \ \cdots \ \boldsymbol{\theta}_{\mathrm{an}}\right]^{\mathrm{T}}$,

$\mathbf{G} = \mathrm{diag}(G_1(s), \ldots, G_n(s))$,

$\mathbf{F} = \mathrm{diag}(F_1(s), \ldots, F_n(s))$,

$$\mathbf{P}_k = \begin{bmatrix} P_{11k}(s) & \cdots & P_{1ik}(s) & \cdots & P_{1nk}(s) \\ \vdots & \ddots & \vdots & & \vdots \\ P_{i1k}(s) & \cdots & P_{iik}(s) & \cdots & P_{ink}(s) \\ \vdots & & \vdots & \ddots & \vdots \\ P_{n1k}(s) & \cdots & P_{nik}(s) & \cdots & P_{nnk}(s) \end{bmatrix},$$

$\boldsymbol{\theta}_{\mathrm{c}} = \left[\theta_{\mathrm{c1}} \ \cdots \ \theta_{\mathrm{cn}}\right]^{\mathrm{T}}$,

$\mathbf{C}_{ck} = \left[C_{\mathrm{c}1k} \ \cdots \ C_{\mathrm{c}nk}\right]^{\mathrm{T}}$,

$\mathbf{E}_k = [e_{1k} \ \cdots \ e_{nk}]^{\mathrm{T}}$,

$$\mathbf{Q} = \begin{bmatrix} 0 & \cdots & Q_{1i}(s) & \cdots & Q_{1n}(s) \\ \vdots & \ddots & \vdots & & \vdots \\ Q_{i1}(s) & \cdots & 0 & \cdots & Q_{in}(s) \\ \vdots & & \vdots & \ddots & \vdots \\ Q_{n1}(s) & \cdots & Q_{ni}(s) & \cdots & 0 \end{bmatrix},$$

$\mathbf{C}_{\mathrm{a}} = \left[C_{\mathrm{a}1} \ \cdots \ C_{\mathrm{a}n}\right]^{\mathrm{T}}$, the subscript k denotes the number of the trial, and s represents the Laplace variable.

Thus, the error of the control system can be expressed as

$$\mathbf{E}_k = -(\mathbf{P}_k\boldsymbol{\theta}_{\mathrm{c}} + \mathbf{C}_{ck}) + \mathbf{F}^{-1}(\mathbf{G}^{-1}\boldsymbol{\theta}_{\mathrm{a}} + \mathbf{Q}\boldsymbol{\theta}_{\mathrm{a}} + \mathbf{C}_{\mathrm{a}}), \tag{5}$$

where $\mathbf{G}^{-1}\boldsymbol{\theta}_{\mathrm{a}} + \mathbf{Q}\boldsymbol{\theta}_{\mathrm{a}} + \mathbf{C}_{\mathrm{a}}$ is the external dynamic load.

In eq. (3), since $\mathbf{D}_1$, $\mathbf{D}_2$, $\mathbf{D}_3$, and $\mathbf{D}_4$ are calculated by taking

the corresponding average value of these matrices within the whole workspace or on a given trajectory, $\mathbf{D}_1$, $\mathbf{D}_2$, $\mathbf{D}_3$, and $\mathbf{D}_4$ are constant. Thus, the dynamic system shown in eq. (3) is linear. Applying the Laplace transform on the dynamic model eq. (3) yields

$$\boldsymbol{\tau}(s) = [\mathbf{D}_1 s^2 + \mathbf{D}_2 s + \mathbf{D}_3 \quad \mathbf{D}_4]\begin{bmatrix}\boldsymbol{\theta}_a \\ 1\end{bmatrix}. \tag{6}$$

Substituting the dynamic model eq. (6) into eq. (5) leads to

$$\mathbf{E}_k = -\begin{bmatrix}\mathbf{P}_k & \mathbf{C}_{ck}\end{bmatrix}\begin{bmatrix}\boldsymbol{\theta}_c \\ 1\end{bmatrix} + \boldsymbol{\Omega}, \tag{7}$$

where

$$\boldsymbol{\Omega} = \mathbf{F}^{-1}\left(\begin{bmatrix}\mathbf{D}_1 s^2 + \mathbf{D}_2 s + \mathbf{D}_3 & \mathbf{D}_4\end{bmatrix}\begin{bmatrix}\boldsymbol{\theta}_a \\ 1\end{bmatrix}\right).$$

For the feedback controller, the transfer function $\mathbf{F}$ is a continuous function on the interval from 0 to its upper bound. Therefore, according to Weierstrass's first approximation theorem [24], $\mathbf{F}$ can be approximately expressed as

$$\mathbf{F}^{-1} \approx \sum_{l=0}^{r} f_l s^l, \tag{8}$$

where f_l is the coefficient of the lth order system.

Thus, substituting eq. (8) into $\boldsymbol{\Omega}$ leads to

$$\boldsymbol{\Omega} = \begin{bmatrix} \sum_{l=0}^{m}\xi_{11l}s^l & \cdots & \sum_{l=0}^{m}\xi_{1il}s^l & \cdots & \sum_{l=0}^{m}\xi_{1nl}s^l & \xi_{1(n+1)} \\ \vdots & \ddots & \vdots & & \vdots & \vdots \\ \sum_{l=0}^{m}\xi_{i1l}s^l & \cdots & \sum_{l=0}^{m}\xi_{iil}s^l & \cdots & \sum_{l=0}^{m}\xi_{inl}s^l & \xi_{i(n+1)} \\ \vdots & & \vdots & \ddots & \vdots & \vdots \\ \sum_{l=0}^{m}\xi_{n1l}s^l & \cdots & \sum_{l=0}^{m}\xi_{nil}s^l & \cdots & \sum_{l=0}^{m}\xi_{nnl}s^l & \xi_{n(n+1)} \end{bmatrix}\begin{bmatrix}\boldsymbol{\theta}_a \\ 1\end{bmatrix}, \tag{9}$$

where m=r+2.

If the error $\mathbf{E}_k$ is equal to zero, the command position signal is equal to the actual position. The following equation can be obtained:

$$\boldsymbol{\theta}_c - \boldsymbol{\theta}_a = 0. \tag{10}$$

Namely, if the error $\mathbf{E}_k$ is equal to zero, the compensation signal $\mathbf{P}_k\boldsymbol{\theta}_c$+$\mathbf{C}_c$ would be equal to the external load $\boldsymbol{\Omega}$ according to eq. (7). Combining eqs. (7), (9), and (10) leads to

$$\begin{bmatrix} P_{11k}(s) & \cdots & P_{1ik}(s) & \cdots & P_{1nk}(s) & C_{c1k} \\ \vdots & \ddots & \vdots & & \vdots & \vdots \\ P_{i1k}(s) & \cdots & P_{iik}(s) & \cdots & P_{ink}(s) & C_{cik} \\ \vdots & & \vdots & \ddots & \vdots & \vdots \\ P_{n1k}(s) & \cdots & P_{nik}(s) & \cdots & P_{nnk}(s) & C_{cnk} \end{bmatrix} = \begin{bmatrix} \sum_{l=0}^{m}\xi_{11l}s^l & \cdots & \sum_{l=0}^{m}\xi_{1il}s^l & \cdots & \sum_{l=0}^{m}\xi_{1nl}s^l & \xi_{1(n+1)} \\ \vdots & \ddots & \vdots & & \vdots & \vdots \\ \sum_{l=0}^{m}\xi_{i1l}s^l & \cdots & \sum_{l=0}^{m}\xi_{iil}s^l & \cdots & \sum_{l=0}^{m}\xi_{inl}s^l & \xi_{i(n+1)} \\ \vdots & & \vdots & \ddots & \vdots & \vdots \\ \sum_{l=0}^{m}\xi_{n1l}s^l & \cdots & \sum_{l=0}^{m}\xi_{nil}s^l & \cdots & \sum_{l=0}^{m}\xi_{nnl}s^l & \xi_{n(n+1)} \end{bmatrix}. \tag{11}$$

Simplifying eq. (11) yields

$$\begin{cases} P_{ijk}(s) = \sum_{l=0}^{m}\xi_{ijl}s^l, \\ C_{cik} = \xi_{i(n+1)}. \end{cases} \tag{12}$$

Thus, when the transfer function of the feedforward controller satisfies eq. (12), the external load disturbance can be eliminated and the control error can be reduced. From eq. (12), one may see that the feedforward compensation is a complete feedforward compensation including a constant compensation signal, low order compensation signals, and high order compensation signals.

3 Iterative learning method

According to the analysis in Section 2, the formula to calculate the feedforward parameters is obtained. Ideally, when the feedforward parameters satisfy eq. (12), the control error can be reduced to zero. In fact, this rarely happens. On the one hand, the dynamic model of the robot is not completely accurate due to external uncertain disturbances, elastic deformation, transmission system clearance, and other factors. On the other hand, the actual control system cannot be accurately established and identified either. For example, the temperature drift, zero drift, and dead zones are difficult to be identified and modeled accurately. Therefore, if the feedforward parameters are calculated by eq. (12), the feedforward compensation can reduce certain errors, but it cannot completely eliminate all external load disturbances so that the control system can reach the optimal control performance. Considering the gap between the practice and the ideal, the iterative learning method is utilized to design the parameters. By continuously iteratively learning, the parameters of the control system can be updated gradually and the control system performance tends to be optimal.

3.1 Iterative equation

In this section, the iterative learning equation is established by analyzing the motion error signals. To reduce complexity, the iterative equations of each joint are analyzed and estab-

lished. Then, the iterative equations of the whole robot are obtained.

Assume that the feedforward compensation signal of the kth trial is expressed as

$$\begin{cases} P_{ijk}(s) = \sum_{l=0}^{m} \mu_{ijlk} s^{l}, \\ C_{cik}(s) = \mu_{i(n+1)k}, \end{cases} \tag{13}$$

where μ_{ijlk} and $\mu_{i(n+1)k}$ are the feedforward compensation parameters of the kth trial. Obviously, ξ_{ijl} and $\xi_{i(n+1)}$ in eq. (12) are the ideal feedforward parameters. The target of this section is to make μ_{ijlk} and $\mu_{i(n+1)k}$ gradually approach ξ_{ijl} and $\xi_{i(n+1)}$ by using iterative equations.

Based on eqs. (12) and (13), the error e_{ik} can be expressed as

$$\begin{aligned} e_{ik} &= \theta_{ci} - \theta_{ai} \\ &= \sum_{j=1}^{n}\left(\sum_{l=0}^{m} \xi_{ijl} s^{l} \theta_{cj}\right) + \xi_{i(n+1)} - \sum_{j=1}^{n}\left(\sum_{l=0}^{m} \mu_{ijlk} s^{l} \theta_{cj}\right) \\ &\quad - \mu_{i(n+1)k}. \end{aligned} \tag{14}$$

Taking an inverse Laplace transform on the error signal eq. (14) of joint i yields

$$\begin{aligned} \Gamma_i(\Upsilon_{ik}) &= \mathbb{L}^{-1}(e_{ik}) \\ &= \begin{bmatrix} \boldsymbol{\Theta}_{a1} & \cdots & \boldsymbol{\Theta}_{ai} & \cdots & \boldsymbol{\Theta}_{an} & 1 \end{bmatrix} \psi_i \\ &\quad - \begin{bmatrix} \boldsymbol{\Theta}_{c1} & \cdots & \boldsymbol{\Theta}_{ci} & \cdots & \boldsymbol{\Theta}_{cn} & 1 \end{bmatrix} \Upsilon_{ik}, \end{aligned} \tag{15}$$

where Υ_{ik} denotes the feedforward controller parameter vector of joint i of the kth trial:

$$\Upsilon_{ik} = \begin{bmatrix} \begin{bmatrix} \mu_{i10k} & \cdots & \mu_{i1mk} \end{bmatrix}^{\mathrm{T}} \\ \vdots \\ \begin{bmatrix} \mu_{ij0k} & \cdots & \mu_{ijmk} \end{bmatrix}^{\mathrm{T}} \\ \vdots \\ \begin{bmatrix} \mu_{in0k} & \cdots & \mu_{inmk} \end{bmatrix}^{\mathrm{T}} \\ \mu_{i(n+1)k} \end{bmatrix},$$

$\Gamma_i(\Upsilon_{ik})$ is the error signal of joint i in the time domain when the feedforward controller parameter vector is Υ_{ik} and $\boldsymbol{\psi}_i$ denotes the ideal feedforward parameter vector of joint i:

$$\boldsymbol{\psi}_i = \begin{bmatrix} \begin{bmatrix} \xi_{i10} & \cdots & \xi_{i1m} \end{bmatrix}^{\mathrm{T}} \\ \vdots \\ \begin{bmatrix} \xi_{ij0} & \cdots & \xi_{ijm} \end{bmatrix}^{\mathrm{T}} \\ \vdots \\ \begin{bmatrix} \xi_{in0} & \cdots & \xi_{inm} \end{bmatrix}^{\mathrm{T}} \\ \xi_{i(n+1)} \end{bmatrix},$$

$$\boldsymbol{\Theta}_{ci} = \begin{bmatrix} \theta_{ci} & \dfrac{\mathrm{d}\theta_{ci}}{\mathrm{d}t} & \dfrac{\mathrm{d}^2\theta_{ci}}{\mathrm{d}t^2} & \cdots & \dfrac{\mathrm{d}^m\theta_{ci}}{\mathrm{d}t^m} \end{bmatrix},$$

$$\boldsymbol{\Theta}_{ai} = \begin{bmatrix} \theta_{ai} & \dfrac{\mathrm{d}\theta_{ai}}{\mathrm{d}t} & \dfrac{\mathrm{d}^2\theta_{ai}}{\mathrm{d}t^2} & \cdots & \dfrac{\mathrm{d}^m\theta_{ai}}{\mathrm{d}t^m} \end{bmatrix}.$$

The error equation of joint i can be obtained by performing a Taylor first-order expansion on eq. (16). The error equation of joint i can be expressed as

$$\Gamma_i(\Upsilon_i) \approx \Gamma_{ik}(\Upsilon_{ik}) + \frac{\partial \Gamma_{ik}(\Upsilon_{ik})}{\partial \Upsilon_{ik}}(\Upsilon_i - \Upsilon_{ik}), \tag{16}$$

where $\dfrac{\partial \Gamma_{ik}(\Upsilon_{ik})}{\partial \Upsilon_{ik}}$ is the Jacobian matrix of the error equation of joint i. Namely,

$$\begin{aligned} \mathbf{J} = \frac{\partial \Gamma_{ik}(\Upsilon_{ik})}{\partial \Upsilon_{ik}} &= \begin{bmatrix} \begin{bmatrix} \dfrac{\partial \Gamma_{ik}(\Upsilon_{ik})}{\partial \mu_{i10}} & \cdots & \dfrac{\partial \Gamma_{ik}(\Upsilon_{ik})}{\partial \mu_{i1m}} \end{bmatrix}^{\mathrm{T}} \\ \vdots \\ \begin{bmatrix} \dfrac{\partial \Gamma_{ik}(\Upsilon_{ik})}{\partial \mu_{ij0}} & \cdots & \dfrac{\partial \Gamma_{ik}(\Upsilon_{ik})}{\partial \mu_{ijm}} \end{bmatrix}^{\mathrm{T}} \\ \vdots \\ \begin{bmatrix} \dfrac{\partial \Gamma_{ik}(\Upsilon_{ik})}{\partial \mu_{in0}} & \cdots & \dfrac{\partial \Gamma_{ik}(\Upsilon_{ik})}{\partial \mu_{inm}} \end{bmatrix}^{\mathrm{T}} \\ \dfrac{\partial \Gamma_{ik}(\Upsilon_{ik})}{\partial \mu_{i(n+1)}} \end{bmatrix}^{\mathrm{T}} \\ &= \begin{bmatrix} \boldsymbol{\Theta}_{c1} & \cdots & \boldsymbol{\Theta}_{cj} & \cdots & \boldsymbol{\Theta}_{cn} & 1 \end{bmatrix}. \end{aligned} \tag{17}$$

Thus, based on the Moore-Penrose inverse, the recursive equation can be expressed as

$$\Upsilon_{ik+1} = \Upsilon_{ik} - \mathbf{J}(\mathbf{J}\mathbf{J}^{\mathrm{T}})^{-1} \Gamma_{ik}(\Upsilon_{ik}). \tag{18}$$

According to eq. (18), the number of the iterative equation is only one, which is less than the number of parameters that need to be identified. Based on the linear algebra knowledge, this type of equation has infinite solutions. Thus, we need to collect system signals when the robot is in different position and posture, and combine them to form new equations, so that the number of equations is greater than the number of parameters. At last, the least-squares method is used to obtain the optimal identification parameters.

Now, we input a set of control signals and collect the error signals. The time vector of the error signals can be assumed as

$$\mathbf{t} = \begin{bmatrix} t_0 & t_1 & \cdots & t_{s-1} & t_s \end{bmatrix}. \tag{19}$$

Next, by combining the iterative equations at the different time, a new iterative equation can be obtained as

$$\Upsilon_{ik+1} = \Upsilon_{ik} - \mathbf{J}^{(\mathbf{t})}\left(\mathbf{J}^{(\mathbf{t})}(\mathbf{J}^{(\mathbf{t})})^{\mathrm{T}}\right)^{-1} \boldsymbol{\Gamma}_{ik}{}^{(\mathbf{t})}, \tag{20}$$

where

$$\mathbf{J}^{\mathbf{t}} = \begin{bmatrix} \boldsymbol{\Theta}_{c1}^{(t_0)} & \cdots & \boldsymbol{\Theta}_{ci}^{(t_0)} & \cdots & \boldsymbol{\Theta}_{cn}^{(t_0)} & 1 \\ \vdots & \ddots & \vdots & & \vdots & \vdots \\ \boldsymbol{\Theta}_{c1}^{(t_i)} & \cdots & \boldsymbol{\Theta}_{ci}^{(t_i)} & \cdots & \boldsymbol{\Theta}_{cn}^{(t_i)} & 1 \\ \vdots & & \vdots & \ddots & \vdots & \vdots \\ \boldsymbol{\Theta}_{c1}^{(t_s)} & \cdots & \boldsymbol{\Theta}_{ci}^{(t_s)} & \cdots & \boldsymbol{\Theta}_{cn}^{(t_s)} & 1 \end{bmatrix},$$

$$\boldsymbol{\Gamma}_{ik}^{(t)}=\begin{bmatrix}\Gamma_{ik}^{(t_0)}\\ \vdots\\ \Gamma_{ik}^{(t_i)}\\ \vdots\\ \Gamma_{ik}^{(t_s)}\end{bmatrix}.$$

Note that the time in the upper right corner indicates that the signal is collected at that time. For example, $\Gamma_{ik}^{(t_0)}$ is the joint i error signal of the kth trial when the time is t_0.

3.2 Accelerated iteration

For ILC, iterative efficiency and speed are critical. If the speed is too slow, the iterative learning method may be worse than the empirical exhaustive method. At the same time, if the input and output of the system do not have an ideal linear relationship, it will also bring challenges to the application of iterative algorithms. Therefore, the PID learning rate is used to optimize the iterative speed [25]. The iterative equation can be expressed as

$$\begin{aligned}\boldsymbol{\Upsilon}_{ik+1}=\boldsymbol{\Upsilon}_{ik}-\mathbf{J}^{(\mathbf{t})}\left(\mathbf{J}^{(\mathbf{t})}(\mathbf{J}^{(\mathbf{t})})^{\mathrm{T}}\right)^{-1}\\ \times\left(\mathbf{K}_{\mathrm{p}}\boldsymbol{\Gamma}_{ik}^{(\mathbf{t})}+\mathbf{K}_{\mathrm{i}}\int\boldsymbol{\Gamma}_{ik}^{(\mathbf{t})}\mathrm{d}\tau+\mathbf{K}_{\mathrm{d}}\dot{\boldsymbol{\Gamma}}_{ik}^{(\mathbf{t})}\right),\end{aligned}\tag{21}$$

where $\mathbf{K}_{\mathrm{p}}$, $\mathbf{K}_{\mathrm{i}}$, and $\mathbf{K}_{\mathrm{d}}$ are $n\times n$ matrices of learning proportional, integral, and derivative gains, respectively.

When the iterative learning process is performed, the $\mathbf{K}_{\mathrm{p}}$, $\mathbf{K}_{\mathrm{i}}$, and $\mathbf{K}_{\mathrm{d}}$ parameters can be increased or decreased to accelerate the iterative efficiency according to the iterative result. The basic principle of adjusting $\mathbf{K}_{\mathrm{p}}$, $\mathbf{K}_{\mathrm{i}}$, and $\mathbf{K}_{\mathrm{d}}$ parameters is to increase $\mathbf{K}_{\mathrm{p}}$ and $\mathbf{K}_{\mathrm{i}}$ or decrease $\mathbf{K}_{\mathrm{d}}$ when the iterative speed is slow. In addition, due to the existence of noise signals, the signal-to-noise ratio of the error signal is relatively high at the beginning of the iteration. The iterative learning method can use PD or P learning laws to increase iterative speed. When the motion error gradually decreases, the signal-to-noise ratio is also reduced. Then, PI learning laws can be used to reduce the interference of noise signals, and improve the iterative precision and robustness.

3.3 Iteration strategy

Based on the previous work, the iteration strategy of designing the accurate feedforward controller by using the iteration eq. (21) is given in Figure 5.

(1) Preparation. According to the robot dynamic model and the empirical method, the initial feedforward parameters can be manually tuned and set as $\boldsymbol{\Upsilon}_0$. Meanwhile, $\mathbf{K}_{\mathrm{p}}$, $\mathbf{K}_{\mathrm{i}}$, and $\mathbf{K}_{\mathrm{d}}$ are set to all-one matrix, zero matrix, and zero matrix, respectively.

(2) Signal collection. The robot is controlled to move a typical trajectory and the error signals of all joints are collected and recorded as $\mathbf{e}_k^{(\mathbf{t})}$.

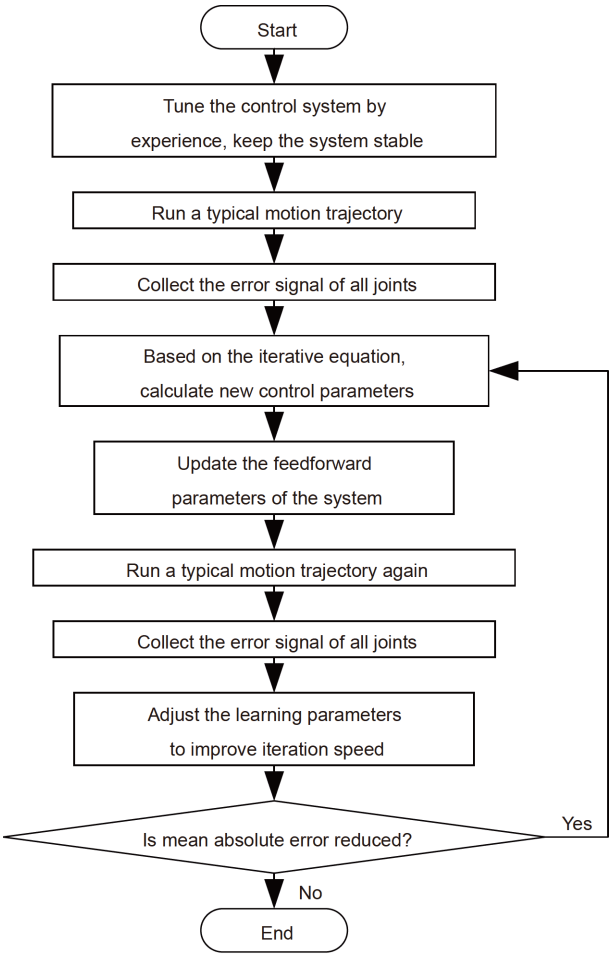

Figure 5 Iteration flow.

(3) Iteration. Based on eq. (21) and the error signal $\mathbf{e}_k^{(\mathbf{t})}$, the new feedforward parameters can be obtained as $\boldsymbol{\Upsilon}_{k+1}$.

(4) Update parameters. Update the control parameters $\boldsymbol{\Upsilon}_{k+1}$ of the control system.

(5) Signal repeated collection. The robot is controlled to repeat the motion trajectory in step (2), and the error signals of all joints are collected again. The error signal is recorded as $\mathbf{e}_{k+1}^{(\mathbf{t})}$.

(6) Improve iteration speed. To improve the iteration speed, the learning parameters, $\mathbf{K}_{\mathrm{p}}$, $\mathbf{K}_{\mathrm{i}}$, and $\mathbf{K}_{\mathrm{d}}$, should be adjusted. $\mathbf{K}_{\mathrm{p}}$ can be updated by $\mathbf{K}'_{\mathrm{p}}=\left(\overline{\mathbf{e}_{k+1}^{(\mathbf{t})}}-\overline{\mathbf{e}_k^{(\mathbf{t})}}\right)\cdot(\boldsymbol{\Upsilon}_{k+1}-\boldsymbol{\Upsilon}_k)^{+}\mathbf{K}_{\mathrm{p}}$, where $\mathbf{K}'_{\mathrm{p}}$ represents the new $\mathbf{K}_{\mathrm{p}}$. The parameters $\mathbf{K}_{\mathrm{i}}$ and $\mathbf{K}_{\mathrm{d}}$ can be slightly increased or decreased based on the principle proposed in Section 3.2.

(7) Error analysis. The error signals $\mathbf{e}_{k+1}^{(\mathbf{t})}$ and $\mathbf{e}_k^{(\mathbf{t})}$ are compared and analyzed. If the mean absolute error is no longer reduced or even increased, stop the iteration; otherwise, to step (3).

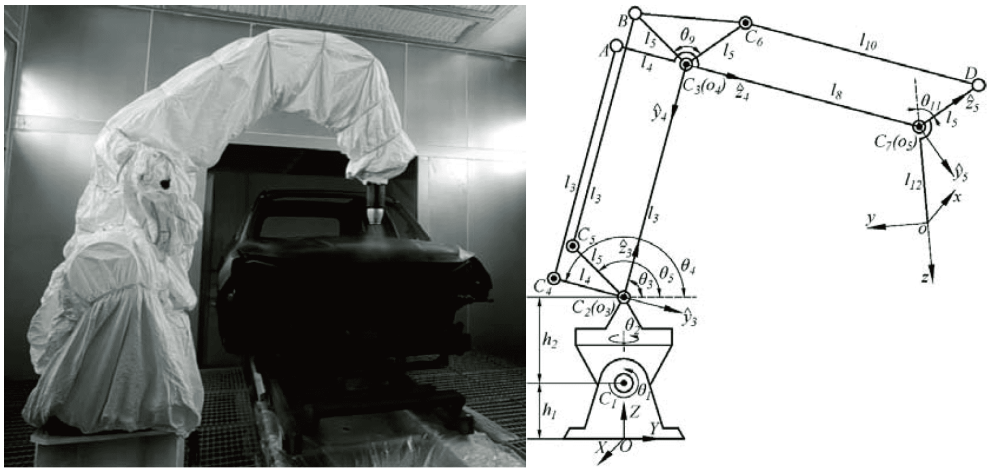

Figure 6 (Color online) Five-DOF hybrid spray-painting robot. (a) The robot is painting a car; (b) kinematic model.

4 Experiments

4.1 Robot prototype and control system

To demonstrate the feasibility and effectiveness of the proposed iterative learning method, the method is implemented on the prototype of a 5-DOF hybrid spray-painting robot. As shown in Figure 6, the robot is designed and manufactured by Tsinghua University and Jiangsu Changhong Intelligent Equipment Company, China. The dynamic model is derived in ref. [26] and the geometrical and inertial parameters are given in ref. [26]. The robot is controlled by the Turbo UMAC controller made by DELTA TAU. The encoder is an absolute position encoder with a resolution of 2500 increments per revolution. In the experiments, the current and velocity loop controllers are realized in servo drivers. The position loop controller is implemented in the Turbo UMAC controller. The Turbo UMAC controller gives the analog output with range −10–10 V and quadruples the encoder's angle measurement. The block diagram of the control system is shown in Figure 7.

Based on the D-H method and the Lagrangian dynamic formulation, the kinematic and dynamic models of the hybrid robot can be derived. The kinematic model of the robot is expressed as

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} c(\theta_5-\theta_9+\theta_{11})s\theta_2 \\ s(\theta_5-\theta_9)(c\theta_1c\theta_2s\theta_{11}+s\theta_1c\theta_{11})+c(\theta_5-\theta_9)(s\theta_1s\theta_{11}-c\theta_1c\theta_2c\theta_{11}) \\ s(\theta_5-\theta_9)(s\theta_1c\theta_2s\theta_{11}-c\theta_1c\theta_{11})-c(\theta_5-\theta_9)(c\theta_1s\theta_{11}+s\theta_1c\theta_2c\theta_{11}) \end{bmatrix}, \tag{22}$$

$$\begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix} = \begin{bmatrix} s\theta_2(l_{12}c(\theta_5-\theta_9+\theta_{11})-l_3c\theta_3+(l_8-l_4)c\theta_4) \\ -h_2s\theta_1+l_3(c\theta_1c\theta_2c\theta_3-s\theta_1s\theta_3)+(l_8-l_4)(s\theta_1s\theta_4-c\theta_1c\theta_2c\theta_4)+l_{12}v \\ h_1+h_2c\theta_1+l_3(s\theta_1c\theta_2c\theta_3+c\theta_1s\theta_3)-(l_8-l_4)(c\theta_1s\theta_4+s\theta_1c\theta_2c\theta_4)+l_{12}w \end{bmatrix}, \tag{23}$$

where $[u\ v\ w]^{\mathrm{T}}$ is the direction vector of the sprayer, $[x_0\ y_0\ z_0]^{\mathrm{T}}$ is the position of the end of the sprayer. θ_1–θ_{11}, l_1–l_{11}, and

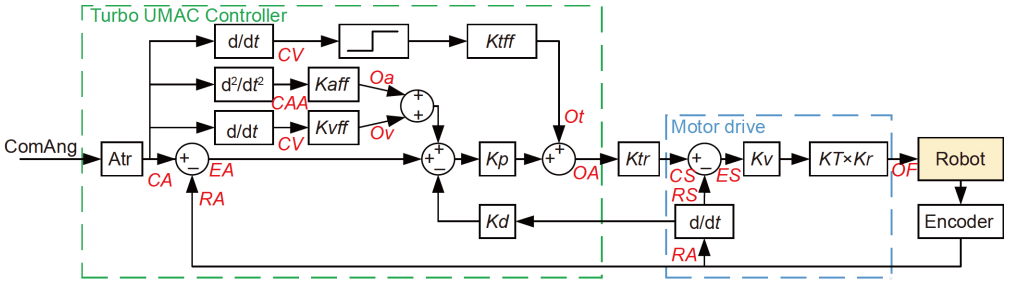

Figure 7 (Color online) Block diagram of the control system.

h_1–h_2 respectively indicate the rotation angles of robot links, link length, and platform height, as shown in Figure 6(b). Notice that $s\theta_1$ and $c\theta_1$ denote the sine and cosine of angle θ_1 respectively, and a similar convention will be adopted throughout the following formulations.

The dynamic equation of the robot [26] can be expressed as

$$\boldsymbol{\tau}=\mathbf{M}(\boldsymbol{\Theta})\ddot{\boldsymbol{\Theta}}+\mathbf{V}(\boldsymbol{\Theta},\dot{\boldsymbol{\Theta}})+\mathbf{G}(\boldsymbol{\Theta})+\boldsymbol{\tau}_{\mathrm{f}}, \tag{24}$$

where $\mathbf{M}(\boldsymbol{\Theta})$ is the mass matrix of the robot, $\mathbf{V}(\boldsymbol{\Theta},\dot{\boldsymbol{\Theta}})$ is the coefficient matrix of Coriolis and centrifugal forces, $\mathbf{G}(\boldsymbol{\Theta})$ is the gravitational term, and $\boldsymbol{\tau}_{\mathrm{f}}$ is the friction torque.

The control system of each active joint is shown in Figure 7. *Kaff*, *Kvff*, and *Ktff* are the acceleration feedforward parameter, the velocity feedforward parameter, and the constant feedforward parameter, respectively. It can be seen that the feedforward control system is simpler than the system shown in Figure 4. Due to high reliability and ease of operation, the actual industrial control system generally does not contain complete feedforward parameters. Since the complete iterative learning equation (eq. (20)) has been derived, we only need to delete the equation corresponding to the missing parameters, recombine the remaining equations, and iteratively calculate. The final feedforward parameters can be obtained.

4.2 Iterative experiment

To verify the effectiveness of the iterative learning method, a complex spray-painting trajectory is planned for designing the feedforward parameters. The trajectory is shown in Figure 8.

It should be noted that it is a 40-second complex trajectory which is carefully and well designed. During the 40 s motion, the trajectory includes most of the robot workspace and all work states of the robot. Thus, based on this trajectory, we can optimize the robot to reach optimal performance within the whole workspace. For the sake of convenience, we divide the trajectory into several figures as shown in Figures 9. In Figure 9(a) and (b), the robot is in an initial and ready state. In this state, the robot slowly moves to the starting point and waits for the start signal to start painting. In Figure 9(c) and (d), the robot rotates 90° and is in a "lying down" work state. In this state, the robot is painting the side of the car, such as the door, the A-pillar, and the B-pillar. In Figure 9(e) and (f), the robot is in an "upright" normal work state. In this state, the robot is painting the upside of the car, such as the bonnet and the roof panel.

Next, the initial feedforward parameters are designed based on the dynamic model (eq. (24)) and engineer experience. Then, the iterative learning method is utilized to tuning the feedforward parameters by following the iteration flow shown in Figure 5. The parameters of the feedforward parameters are shown in Figure 10 as the trial increases. From Figure 10, the acceleration and constant feedforward parameters converge rapidly, which illustrates that the iterative learning method has high convergence speed and high efficiency.

Figure 11 shows the tracking performance of each active joint of the robot. One may see that the root-mean-square (RMS) values of the errors are decreased greatly because the iterative learning method makes the acceleration and constant feedforward parameters tend to be accurate, and the feedforward controller can accurately compensate the external uncertain dynamic characteristic of the robot.

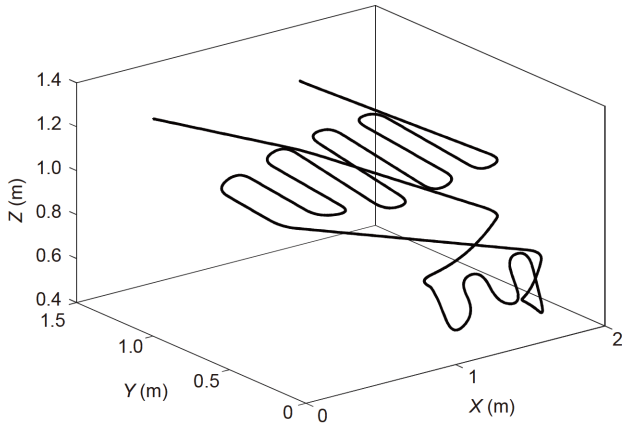

Figure 8 Typical spray-painting trajectory.

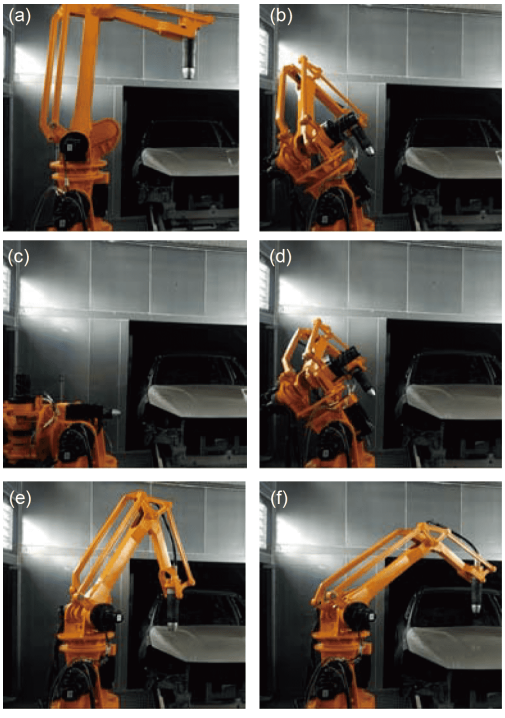

Figure 9 (Color online) A 40-second complex trajectory of robot. (a), (b) Robot is in an initial and ready state. (a) 1 s; (b) 8 s. (c), (d) Robot rotates 90° and is in a "lying down" working state. (c) 12 s; (d) 18 s. (e), (f) Robot is in an "upright" normal work state. (e) 23 s; (f) 33 s.

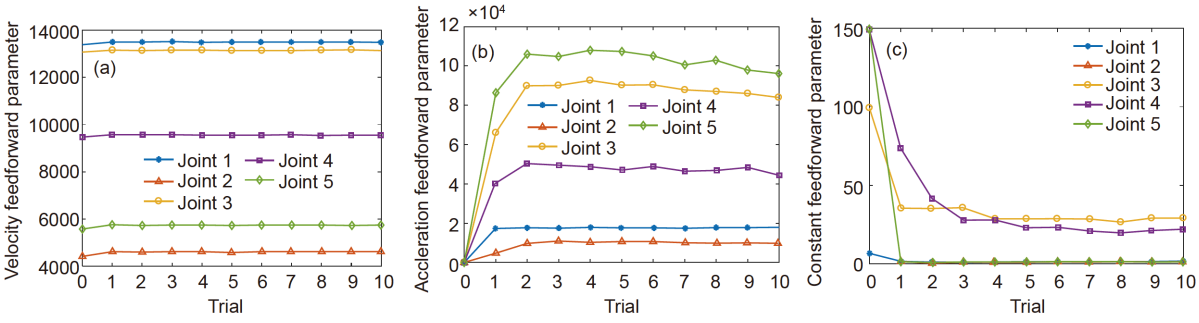

Figure 10 (Color online) Feedforward parameter values as the trial increases. (a) Velocity feedforward parameter values; (b) acceleration feedforward parameter values; (c) constant feedforward parameter values.

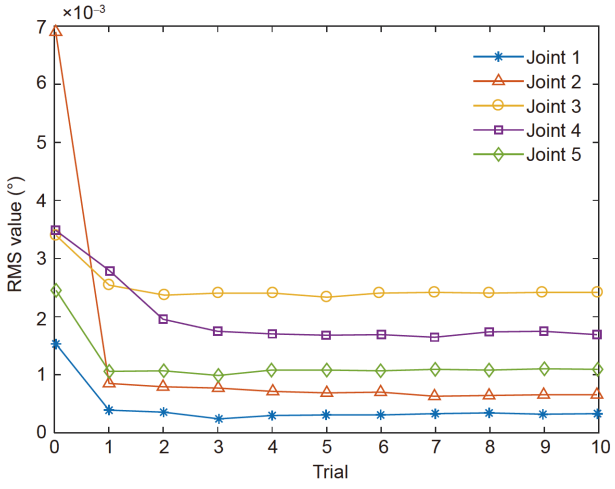

Figure 11 (Color online) RMS values of the errors in all joints.

For better observation, the trajectory of the end of the sprayer is obtained by using the kinematic model (eq. (22)). The tracking error of the robot can be obtained by subtracting the actual trajectory from the command trajectory. The values of the tracking errors of the robot are shown in Figure 12 as the trial increases.

From Figure 12, after ten trials, the average position errors are reduced from 0.18 to 0.06 mm, which is reduced by approximately 66%. The result verifies the effectiveness of the proposed iterative learning method.

4.3 Discussion

From Figure 10, it can be seen that the velocity feedforward parameter changes a little, while the acceleration and constant feedforward parameters vary greatly. It is because the velocity feedforward parameter is only relevant to the internal control system itself and is independent of the external robot. Since a feedback control system is an error-driven system, the control system inevitably has errors. Coincidentally, the control system errors can be compensated by the velocity feedforward compensator.

Let us review the control system again, as shown in Figure 7. If the position and velocity errors of the control system are zero, namely, the command angle is equal to the actual angle, and the command angular velocity is equal to the actual angular velocity. Then the equations can be written as

$$\begin{cases} EA = CA - RA = 0, \\ ES = CS - RS = 0, \end{cases} \tag{25}$$

where EA and ES are the angle error and the angular velocity error, CA and CS are the angle command and the angular velocity command, and RA and RS are the actual angle and

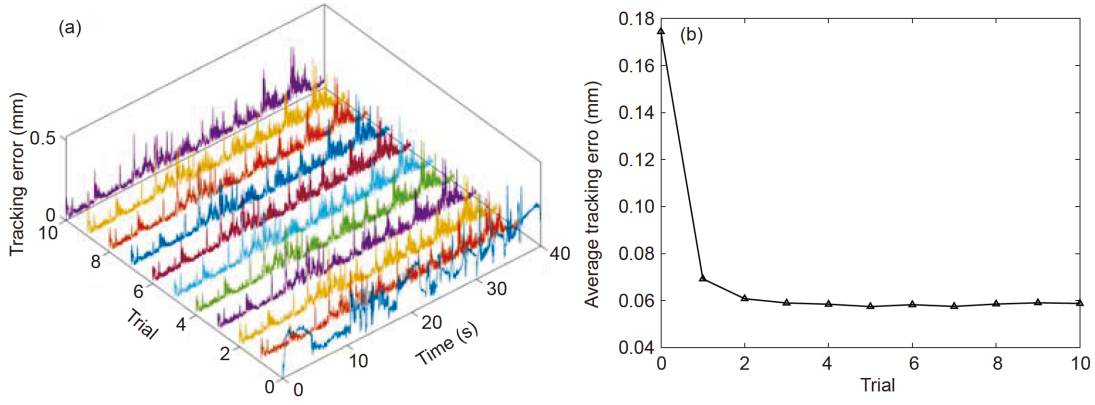

Figure 12 (Color online) Tracking errors of the end of the sprayer. (a) Values of the tracking errors of the end of the sprayer; (b) average tracking errors.

the actual angular velocity, respectively.

According to Figure 12, the output signal of the Turbo UMAC controller (OA) can be written as

$$OA = (Atr \times EA - Kd \times RS + Oa + Ov)Kp + Ot, \tag{26}$$

where Atr is the scaling term between the command angle and the corresponding signal in the control system, Ov, Oa, and Ot are the feedforward compensation signals of velocity, acceleration, and constant, respectively. Ov, Oa, and Ot can be expressed as

$$Ov = Kvff \times CV, \tag{27}$$

$$Oa = Kaff \times CAA, \tag{28}$$

$$Ot = \begin{cases} Ktff & (CV > 0), \\ 0 & (CV = 0), \\ -Ktff & (CV < 0), \end{cases} \tag{29}$$

where CV and CAA are the first and second derivatives of the angle command, respectively.

Combining eqs. (25)–(29) leads to

$$\begin{cases} Kvff = \dfrac{1}{Ktr \times Kp} + Kd, \\ Kaff = 0, \\ Ktff = 0. \end{cases} \tag{30}$$

Therefore, when $Kvff$=1/(Ktr×Kp)+Kd, the error can be reduced to zero. It demonstrates that the velocity feedforward parameter only depends on the internal control system. In the actual tuning process, the empirical tuning method can only tune the robot when it stops at a certain position and posture. Thus, the method can accurately design the velocity feedforward parameters which are relevant to the internal control system and independent of the robot dynamic characteristics. But it is difficult to accurately design the acceleration and constant feedforward parameters which are strongly related to the robot dynamic characteristics. Thus, after iterative learning, the acceleration and constant feedforward parameters rapidly converge to the optimal parameters, and the iterative learned and accurate feedforward controller can compensate the external dynamics of the robot, which significantly improves the robot performance and reduces the motion error by 66%.

5 Conclusion

In this paper, an iterative learning method has been proposed to accurately design and iteratively updated feedforward controller parameters. Based on a standard dynamic model, a complete feedforward controller is designed. On the basis of such a control scheme, an iterative learning method is realized offline to optimize the feedforward control parameters by using a post-processing program in industrial environment. Then, experiments are carried out on the prototype of a 5-DOF industrial hybrid robot. After the proposed iterative learning method is applied, the average position errors are reduced from 0.18 to 0.06 mm, which is reduced by approximately 66%. The experiment results consistently validate that the proposed method is effective in improving the robot motion performance and can compensate unmodeled uncertainties to attenuate the tracking error. This work provides an effective method for applying ILC to an unopened industrial control system. It is very useful for the practical control of hybrid robots in industrial field.

Because the commercial control system is not open and cannot be changed, the ILC can only be realized offline by using a post-processing program. In our future research, we will focus on improving the efficiency of the proposed method.

This work was supported by the National Key Research and Development Program of China (Grant No. 2017YFE0111300), EU H2020-MSCA-RISE-ECSASDPE (Grant No. 734272), and the National Natural Science Foundation of China (Grant No. 51975321).

1 Huang T, Wang M, Yang S, et al. Force/motion transmissibility analysis of six degree of freedom parallel mechanisms. J Mech Robot, 2014, 6: 031010
2 Kang X, Dai J S. Relevance and transferability for parallel mechanisms with reconfigurable platforms. J Mech Robot, 2019, 11: 031012
3 Wu J, Wang J, Wang L, et al. Dynamics and control of a planar 3-DOF parallel manipulator with actuation redundancy. Mech Mach Theor, 2009, 44: 835–849
4 Li Z Y, Zhao D J, Zhao J S. Structure synthesis and workspace analysis of a telescopic spraying robot. Mech Mach Theor, 2019, 133: 295–310
5 Zhao Y, Gao F. Dynamic performance comparison of the 8PSS redundant parallel manipulator and its non-redundant counterpart—The 6PSS parallel manipulator. Mech Mach Theor, 2009, 44: 991–1008
6 Rushworth A, Axinte D, Raffles M, et al. A concept for actuating and controlling a leg of a novel walking parallel kinematic machine tool. Mechatronics, 2016, 40: 63–77
7 Lu Y, Dai Z. Dynamics model of redundant hybrid manipulators connected in series by three or more different parallel manipulators with linear active legs. Mech Mach Theor, 2016, 103: 222–235
8 Wu J, Yu G, Gao Y, et al. Mechatronics modeling and vibration analysis of a 2-DOF parallel manipulator in a 5-DOF hybrid machine tool. Mech Mach Theor, 2018, 121: 430–445
9 Shang W, Cong S. Nonlinear computed torque control for a high-speed planar parallel manipulator. Mechatronics, 2009, 19: 987–992
10 Yang X, Liu H, Xiao J, et al. Continuous friction feedforward sliding mode controller for a TriMule hybrid robot. IEEE/ASME Trans Mechatron, 2018, 23: 1673–1683
11 Matsubara A, Nagaoka K, Fujita T. Model-reference feedforward controller design for high-accuracy contouring control of machine tool axes. CIRP Ann, 2011, 60: 415–418
12 Veronesi M, Visioli A. Automatic tuning of feedforward controllers for disturbance rejection. Ind Eng Chem Res, 2014, 53: 2764–2770
13 Lipiński K. Modeling and control of a redundantly actuated variable mass 3RRR planar manipulator controlled by a model-based feedforward and a model-based-proportional-derivative feedforward-feedback controller. Mechatronics, 2016, 37: 42–53
14 Liu H, Huang T, Chetwynd D G, et al. Stiffness modeling of parallel mechanisms at limb and joint/link levels. IEEE Trans Robot, 2017, 33: 734–741
15 Wu J, Wang J, You Z. An overview of dynamic parameter identifi-

cation of robots. Robot Com-Int Manuf, 2010, 26: 414–419
16 Shang W, Cong S, Kong F. Identification of dynamic and friction parameters of a parallel manipulator with actuation redundancy. Mechatronics, 2010, 20: 192–200
17 Bukkems B, Kostic D, de Jager B, et al. Learning-based identification and iterative learning control of direct-drive robots. IEEE Trans Contr Syst Technol, 2005, 13: 537–549
18 Wu J, Han Y, Xiong Z, et al. Servo performance improvement through iterative tuning feedforward controller with disturbance compensator. Int J Mach Tools Manuf, 2017, 117: 1–10
19 Wang Z, Hu C, Zhu Y, et al. Newton-ILC contouring error estimation and coordinated motion control for precision multiaxis systems with comparative experiments. IEEE Trans Ind Electron, 2017, 65: 1470–1480
20 Liu Q, Xiao J, Yang X, et al. An iterative tuning approach for feedforward control of parallel manipulators by considering joint couplings. Mech Mach Theor, 2019, 140: 159–169
21 Li M, Zhu Y, Yang K, et al. Convergence rate oriented iterative feedback tuning with application to an ultraprecision wafer stage. IEEE Trans Ind Electron, 2018, 66: 1993–2003
22 Kober J, Bagnell J A, Peters J. Reinforcement learning in robotics: A survey. Int J Robot Res, 2013, 32: 1238–1274
23 Rani P, Liu C, Sarkar N, et al. An empirical study of machine learning techniques for affect recognition in human-robot interaction. Pattern Anal Applic, 2006, 9: 58–69
24 Pérez D, Quintana Y. A survey on the Weierstrass approximation theory. Divulgac Matemat, 2008, 16: 231–247
25 Park K H. A study on the robustness of a PID-type iterative learning controller against initial state error. Int J Syst Sci, 1999, 30: 49–59
26 Zhang B, Wu J, Wang L, et al. Accurate dynamic modeling and control parameters design of an industrial hybrid spray-painting robot. Robot Com-Int Manuf, 2020, 63: 101923