An Iterative Tuning Approach for Feedforward Control of Parallel Manipulators by Considering Joint Couplings

Qi Liu [a], Juliang Xiao [a], Xu Yang [a], Haitao Liu [a], Tian Huang [a,b,*] and D. G. Chetwynd [b]

[a]*Key laboratory of Modern Mechanisms and Equipment Design of Education Ministry, Tianjin University, Tianjin 300072, China*
[b]*School of Engineering, The University of Warwick, Coventry CV4 7AL, UK*

Abstract: Feedforward control is an effective way to improve the joint tracking accuracy of robotic systems. This paper presents an iterative approach for feedforward controller parameter tuning of parallel manipulators that considers joint couplings (cross-talk). Based upon a compound control strategy, increments of the feedforward tuning parameters are iteratively updated by minimizing the sum of squares of joint tracking errors at each step. A plant-free identification Jacobian is formulated using the measured data associated with a number of sequential statuses within each iteration cycle. Experiments on the 3-DOF parallel mechanism within a 5-DOF hybrid robot verify parameter convergence and the extrapolation capability of the proposed approach. Compared to otherwise similar feedforward control not considering joint couplings, the root mean square of joint tracking errors was reduced by up to 22% when the mechanism moved at high speed along a path in the neighborhood of the reference configuration.

Keywords: Feedforward controller tuning; Parallel manipulators.

1. Introduction

In the light of the invariant principle of compound control schemes, feedforward control is an effective way to improve the tracking accuracy of multi-axis servo systems of robotic manipulators, machine tools and many other high-performance mechatronic devices [1-3]. The principle behind the feedforward control is to use the predicted axis torques to resist against disturbances [4]. Parameter tuning is an important issue in the design of feedforward controllers.

Parameter tuning divides roughly into two categories: model-based methods and data-driven methods. In model-based feedforward, the plant is usually parameterized as a lower–order rational function [5-9] and the model parameters are estimated using curve fitting techniques. The feedforward control can then be implemented directly *via* model inversion. Although desirable performance and extrapolation capacity can theoretically be achieved, the realistic effectiveness is highly dependent upon the degree of dynamic complexity and practical accuracy of the parameter identification of the system being controlled. Therefore, tremendous efforts have been made towards the improvement of identification accuracy of dynamic parameters of robotic manipulators [10-15] though many others may not be included here. Iterative learning control (ILC) is a typical data-driven approach that draws upon measurements of tracking error signals to enable feedforward controller parameters to be updated *via* minimizing the gradient or increment of tracking errors in an iterative trial manner [16, 17]. An obvious advantage of ILC over model-based feedforward is that it does not need detailed knowledge of the system, although its extrapolation capacity needs to be improved. An elegant approach, combining the advantages of model-based feedforward and ILC, has been presented that uses basis functions to reflect dynamic behaviors and an ILC algorithm to tune model parameters [18, 19]. Satisfactory servo performance could be achieved in terms of extrapolation capacity as well as tracking accuracy. Improvements building on this idea have had the goals of describing higher order dynamics [20, 21], connecting with closed-loop system identification [22, 23], and improving computational accuracy in parameter optimization using the Gauss-Newton method [24, 25]. MIMO feedforward control methods [26, 27] have been applied to multiple-axis servo systems in which the couplings between motion axes may have significant influence on the servo performance. These methods use finite impulse response (FIR) filters as the basis functions of feedforward controllers. The FIR coefficients can be determined using a set of perturbed-parameter experiments in a single trial provided that the objective function being minimized is convex, and the tracking errors are affine in terms of the coefficients. However, these conditions may not be necessary if iterative (or multiple trials) schemes are performed.

Inspired by the method proposed in [19], this paper deals with iterative feedforward tuning of *n*-DOF (degrees of freedom) parallel manipulators by considering joint couplings, so seeking to achieve better joint tracking accuracy in the neighborhood of a given configuration. It is a prerequisite for automatic parameter tuning over the entire work envelop of robotic manipulators by means of polynomial interpolation or fuzzy logical/cluster algorithms. The paper is organized as follows. After this short review of feedforward tuning methods, Section 2 briefly recalls the invariant principle of compound control that arises from the rigid body dynamics of the system. Then Section 3 develops an algorithm that allows the gradient of feedforward parameters to be updated using a set of measured data. In order to overcome the difficulty to perform inverse Laplace transform and to solve the possible ill-conditioned problem encountered in the single-axis iterative tuning algorithm proposed in [19], particular attention is focused on the formulation of a plant-free

*The corresponding author
Email: tianhuang@tju.edu.cn

and full raw ranked identification Jacobian using the measured data associated with a number of perturbed-parameter experiments in each iteration cycle. Experiments on the 3-DOF parallel mechanism within a 5-DOF hybrid robot described in Section 4 verify convergence and the extrapolation capability of the proposed approach, before conclusions are drawn in Section 5.

2. Feedforward Control with Consideration of Joint Couplings

Without loss of generality, we consider a $n-\mathrm{DOF}$ parallel manipulator comprising essentially l limbs connecting a base with a platform. We assume each limb contains at most one actuated joint. Then either $l=n$ with actuated limbs numbered 1,L ,n, or if there exists a properly constrained passive limb, $l=n+1$ and that limb is numbered l. Neglecting the friction and input disturbances, the rigid body dynamics in the joint space [4] of such a manipulator can be represented by

$$\boldsymbol{\tau}=\boldsymbol{\tau}_a+\boldsymbol{\tau}_v+\boldsymbol{\tau}_g \tag{1}$$

$$\boldsymbol{\tau}_a=\boldsymbol{M}(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}},\ \boldsymbol{\tau}_v=\boldsymbol{H}(\boldsymbol{\theta},\dot{\boldsymbol{\theta}})\dot{\boldsymbol{\theta}},\ \boldsymbol{\tau}=(\tau_1\quad\tau_2\quad\mathrm{L}\quad\tau_n)^{\mathrm{T}},\ \boldsymbol{\theta}=(\theta_1\quad\theta_1\quad\mathrm{L}\quad\theta_n)^{\mathrm{T}}$$

where $\boldsymbol{\theta}$ and $\boldsymbol{\tau}$ are the motion and driving torque vectors of actuated joints, and the latter can be expressed as the sum of $\boldsymbol{\tau}_a$, $\boldsymbol{\tau}_v$ and $\boldsymbol{\tau}_g$ associated with the inertial, Coriolis/Centrifugal, gravitational effects. Since $\boldsymbol{\tau}_v$ is highly nonlinear in nature, it is a common practice to treat $\boldsymbol{\tau}_v$ as a viscous damping vector at the neighborhood of a given configuration [4]. In this way, Eq.(1) can be rewritten as

$$\tau_p=m_{pp}\ddot{\theta}_p+h_{pp}\dot{\theta}_p+d_{c,p}+d_{g,p},\ d_{c,p}=\sum_{q=1,q\neq p}^{n}\left(m_{pq}\ddot{\theta}_q+h_{pq}\dot{\theta}_q\right),\quad p=1,2,\mathrm{L}\ ,n \tag{2}$$

where θ_p and τ_p represent the motion and the driving torque of the pth actuated joint; $d_{g,p}$ and $d_{c,p}$ the static loading due to gravity, and dynamic loading due to motions of other joints; and m and h the various effective direct and cross inertial and drag (velocity dependent) effects. The parallel manipulator is then taken as a multiple-input multiple-output (MIMO) system in terms of dynamic couplings between forces and motions of the actuated joints.

Fig. 1 shows a block diagram for servo control of the pth actuated joint in the neighborhood of a given configuration, which takes into account the joint couplings mentioned above and expressed in the (Laplacian) frequency domain. Please note that this control scheme is adopted only for feedforward parameter tuning, while a decentralized proportional–integral–derivative (PID) controller is employed to ensure robust stability of the feedback loop of each actuated joint. In the diagram, $\Theta_{d,p}$ and $\Theta_{a,p}$ denote the desired and measured signals of joint p; C_p and P_p denote transfer functions of the feedback controller and the plant of the joint; $D_{g,p}$ denotes the torque disturbance due to gravity and other unmodeled factors; $D_{c,pq}$ ($q\neq p$) denotes the cross-talk transfer function between the motion of joint q and the induced dynamic loading it imposes upon joint p. F_{pq} is the transfer function of the specific feedforward controller designated to compensate the driving torques so as to reject dynamic loading applied to joint p

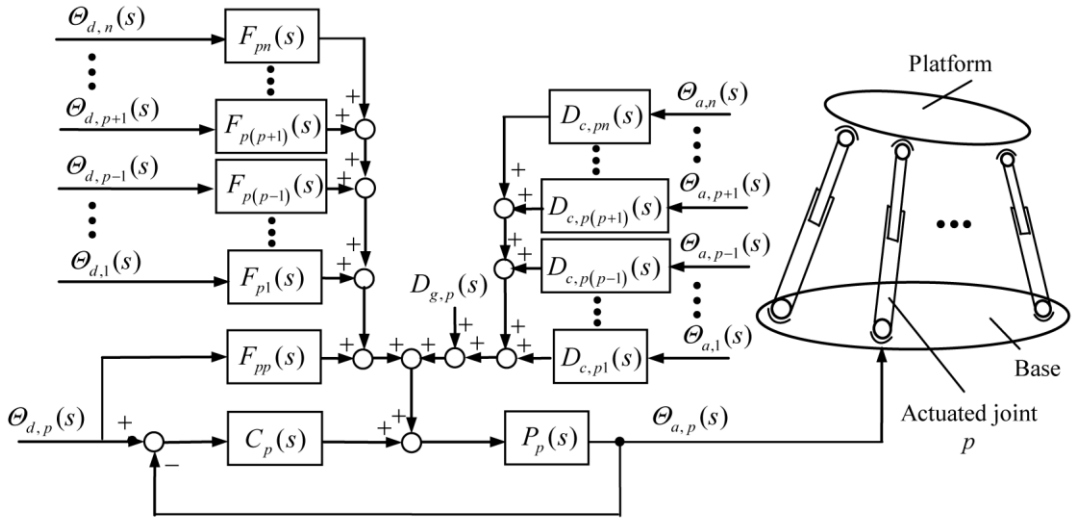

Fig.1 Block diagram of control scheme of a n-DOF parallel manipulator

by the motion of joint q.

The feedforward tuning under consideration is based upon two important arguments: (1) $D_{g,p} \approx 0$ as the steady-state tracking error caused by the gravity can be compensated, to a great extent, by integral parameter tuning *via* PID based feedback control and the PID parameters are well tuned prior to the feedforward control [4]; and (2) $\Theta_{a,q} \approx \Theta_{d,q}$ as $\left|\Theta_{a,q} - \Theta_{d,q}\right| = \left|\Theta_{a,q}\right|$ such that the disturbances being produced by $\Theta_{a,q}$ can be approximately predicated using $\Theta_{d,q}$ ($q = 1,2,\mathrm{K}\ ,n,\ q \neq p$) [28]. These considerations yield the approximated expression of tracking error vector of all actuated joints

$$\boldsymbol{E} = \boldsymbol{A}\left(\boldsymbol{I}_n - \boldsymbol{P}\left(\boldsymbol{F} + \boldsymbol{D}_c\right)\right)\boldsymbol{\Theta}_d \tag{3}$$

with

$$\boldsymbol{A} = \left(\boldsymbol{I}_n + \boldsymbol{PC}\right)^{-1},\ \ \boldsymbol{P} = \mathrm{diag}\left[P_1 \quad P_2 \quad \mathrm{L} \quad P_n\right],\ \ \boldsymbol{C} = \mathrm{diag}\left[C_1 \quad C_2 \quad \mathrm{L} \quad C_n\right]$$

$$\boldsymbol{E} = \left(E_1 \quad E_2 \quad \mathrm{L} \quad E_n\right)^{\mathrm{T}},\ \ \boldsymbol{\Theta}_d = \left(\Theta_{d,1} \quad \Theta_{d,2} \quad \mathrm{L} \quad \Theta_{d,n}\right)^{\mathrm{T}}$$

$$\boldsymbol{F} = \begin{bmatrix} F_{11} & F_{12} & \mathrm{L} & F_{1n} \\ F_{21} & F_{22} & \mathrm{L} & F_{2n} \\ \mathrm{M} & \mathrm{M} & \mathrm{O} & \mathrm{M} \\ F_{n1} & F_{n2} & \mathrm{L} & F_{nn} \end{bmatrix},\ \ \boldsymbol{D}_c = \begin{bmatrix} 0 & D_{12} & \mathrm{L} & D_{1n} \\ D_{21} & 0 & \mathrm{L} & D_{2n} \\ \mathrm{M} & \mathrm{M} & \mathrm{O} & \mathrm{M} \\ D_{n1} & D_{n2} & \mathrm{L} & 0 \end{bmatrix}$$

where $\boldsymbol{I}_n$ denotes a unit matrix of order n. The invariant principle of compound control gives the condition for $\boldsymbol{E} = \boldsymbol{0}$

$$\boldsymbol{I}_n - \boldsymbol{P}\left(\boldsymbol{F} + \boldsymbol{D}_c\right) = \boldsymbol{0} \tag{4}$$

or

$$\boldsymbol{F} = \boldsymbol{P}^{-1} - \boldsymbol{D}_c \tag{5}$$

with

$$F_{pq} = \begin{cases} P_p^{-1} & p = q \\ -D_{c,pq} & p \neq q \end{cases},\ \ p, q = 1,2,\mathrm{L}\ ,n \tag{6}$$

From the dynamics given in Eq.(1), P_p^{-1} and $D_{c,pq}$ can be expressed in the unified (polynomial) forms

$$P_p^{-1} = a_p s^2 + b_p s,\ \ D_{c,pq} = a_{pq} s^2 + b_{pq} s\ \ (p \neq q),\ \ p, q = 1,2,\mathrm{L}\ ,n \tag{7}$$

This allows the feedforward control law of F_{pq} to be parameterized by

$$F_{pq} = \lambda_{pq,1} s + \lambda_{pq,2} s^2 = \sum_{k=1}^{2} \lambda_{pq,k} s^k,\ \ p, q = 1,2,\mathrm{L}\ ,n \tag{8}$$

where $\lambda_{pq,1}$ and $\lambda_{pq,2}$ are referred to as the velocity and acceleration feedforward control parameters of joint q with respect to joint p. Therefore, under the conditions that the feedback controllers of all the actuated joints have been well tuned around the neighborhood of a given configuration, tuning the feedforward controller parameters becomes an issue of determining the full set of $\lambda_{pq,1}$ and $\lambda_{pq,2}$ ($p, q = 1,2,\mathrm{L}\ ,n$) such that the tracking errors of all actuated joints can be simultaneously minimized in the least squares sense.

3. Iterative Tuning Approach for Feedforward Control

By taking account of the joint couplings, this section presents an approach for updating the full set of feedforward parameters. A gradient-approximation based iterative tuning algorithm is proposed first by minimizing the sum of squares of joint tracking errors when the platform moves along a trajectory in the neighborhood of a given configuration. Then, the use of joint tracking error measurements allows a data-driven method to be developed for the formulation of a plant-free identification Jacobian.

3.1 The iterative tuning algorithm

To facilitate the feedforward tuning process, let the platform move along a specific path with a given motion profile repeatedly in each iteration cycle. We assume that $\boldsymbol{P}$, $\boldsymbol{C}$ and $\boldsymbol{D}_c$ given in Eq.(3) are invariant in the neighborhood of a given configuration around which the conditions for the repetition of the initial settings are satisfied. Then, the tracking errors and transfer function matrices, $\boldsymbol{F}_i$ and $\boldsymbol{F}_{i+1}$, of the feedforward controllers associated with the ith and

$(i+1)$th iteration cycles are related by

$$\boldsymbol{E}_{i+1} = \boldsymbol{E}_i - \boldsymbol{B}\left(\boldsymbol{F}_{i+1} - \boldsymbol{F}_i\right)\boldsymbol{\Theta}_d \tag{9}$$

where

$$\boldsymbol{B} = \boldsymbol{AP} = \mathrm{diag}\left[B_1 \quad B_2 \quad \mathrm{L} \quad B_n\right] \text{ with } B_p = \frac{P_p}{1+P_pC_p},\ p = 1,2,\mathrm{L}\ ,n$$

$$\boldsymbol{F}_{i+1} - \boldsymbol{F}_i = \begin{bmatrix} \sum_{k=1}^{2}\Delta\lambda_{11,k,i}s^k & \sum_{k=1}^{2}\Delta\lambda_{12,k,i}s^k & \mathrm{L} & \sum_{k=1}^{2}\Delta\lambda_{1n,k,i}s^k \\ \sum_{k=1}^{2}\Delta\lambda_{21,k,i}s^k & \sum_{k=1}^{2}\Delta\lambda_{22,k,i}s^k & \mathrm{L} & \sum_{k=1}^{2}\Delta\lambda_{2n,k,i}s^k \\ \mathrm{M} & \mathrm{M} & \mathrm{O} & \mathrm{M} \\ \sum_{k=1}^{2}\Delta\lambda_{n1,k,i}s^k & \sum_{k=1}^{2}\Delta\lambda_{n2,k,i}s^k & \mathrm{L} & \sum_{k=1}^{2}\Delta\lambda_{nn,k,i}s^k \end{bmatrix}$$

with $\Delta\lambda_{pq,k,i} = \lambda_{pq,k,i+1} - \lambda_{pq,k,i}$ ($k = 1,2$) where $\lambda_{pq,k,i+1}$ and $\lambda_{pq,k,i}$ are the corresponding feedforward controller parameters determined in the ith and $(i+1)$th iteration cycles, respectively.

Expanding Eq. (9) leads to

$$E_{i+1,p} = E_{i,p} - \boldsymbol{\varphi}_p^{\mathrm{T}}\Delta\boldsymbol{\lambda}_{i,p},\quad p = 1,2,\mathrm{L}\ ,n \tag{10}$$

$$\Delta\boldsymbol{\lambda}_{i,p} = \left(\Delta\lambda_{p1,1,i} \quad \Delta\lambda_{p1,2,i} \quad \mathrm{L} \quad \Delta\lambda_{pq,1,i} \quad \Delta\lambda_{pq,2,i} \quad \mathrm{L} \quad \Delta\lambda_{pn,1,i} \quad \Delta\lambda_{pn,2,i}\right)^{\mathrm{T}} \tag{11}$$

$$\boldsymbol{\varphi}_p = B_p\left(s\Theta_{d,1} \quad s^2\Theta_{d,1} \quad \mathrm{L} \quad s\Theta_{d,q} \quad s^2\Theta_{d,q} \quad \mathrm{L} \quad s\Theta_{d,n} \quad s^2\Theta_{d,n}\right)^{\mathrm{T}} \tag{12}$$

Clearly, $\boldsymbol{\varphi}_p(s)$ is related not only to B_p of the actuated joint p itself, but also to the desired motions of all actuated joints because of the joint couplings. Then, the inverse Laplace transform of Eq. (10) can identify $\Delta\boldsymbol{\lambda}_{i,p}$, i.e.,

$$\mathrm{L}^{-1}\left(E_{i+1,p}\right) = \mathrm{L}^{-1}\left(E_{i,p}\right) - \mathrm{L}^{-1}\left(\boldsymbol{\varphi}_p^{\mathrm{T}}\right)\Delta\boldsymbol{\lambda}_{i,p} \tag{13}$$

results in

$$e_{i+1,p}(t) = e_{i,p}(t) - \boldsymbol{\phi}_p^{\mathrm{T}}(t)\Delta\boldsymbol{\lambda}_{i,p},\quad t \in [0, T] \tag{14}$$

where T denotes the total time interval required for the platform to complete its movement along the path. For numerical implementation, we divide T into N segments with evenly spaced sampling period $T_s = T/N$, leading to $N+1$ sampling nodes and allowing Eq. (14) to be rewritten in a matrix notation as

$$\boldsymbol{e}_{i+1,p} = \boldsymbol{e}_{i,p} - \boldsymbol{\Phi}_p^{\mathrm{T}}\Delta\boldsymbol{\lambda}_{i,p} \tag{15}$$

$$\boldsymbol{\Phi}_p = \left[\boldsymbol{\phi}_p(0) \quad \boldsymbol{\phi}_p(T_s) \quad \mathrm{L} \quad \boldsymbol{\phi}_p(NT_s)\right],\ \boldsymbol{e}_{i,p} = \left(e_{i,p}(0) \quad e_{i,p}(T_s) \quad \mathrm{L} \quad e_{i,p}(NT_s)\right)^{\mathrm{T}} \tag{16}$$

where $\boldsymbol{\Phi}_p$ is the identification Jacobian of parameter set $\Delta\boldsymbol{\lambda}_{i,p}$; $\boldsymbol{e}_{i,p}$ and $\boldsymbol{e}_{i+1,p}$ are the vectors formed by the measured tracking errors of joint p in the ith and $(i+1)$th iteration cycles. Hence, $\Delta\boldsymbol{\lambda}_{i,p}$ can be estimated by minimizing $J = \boldsymbol{e}_{i+1.p}^{\mathrm{T}}\boldsymbol{e}_{i+1.p}$ using ordinary least squares provided that $N+1 \geq 2n$ and $\boldsymbol{\Phi}_p$ is full raw ranked, giving

$$\Delta\hat{\boldsymbol{\lambda}}_{i,p} = \left(\boldsymbol{\Phi}_p\boldsymbol{\Phi}_p^{\mathrm{T}}\right)^{-1}\boldsymbol{\Phi}_p\boldsymbol{e}_{i,p} \tag{17}$$

Consequently, $\boldsymbol{\lambda}_{i,p}$ can be iteratively updated by

$$\boldsymbol{\lambda}_{i+1,p} = \boldsymbol{\lambda}_{i,p} + \Delta\hat{\boldsymbol{\lambda}}_{i,p} \tag{18}$$

until satisfying the convergence criterion

$$\varepsilon_{i,p} \times 100\% \leq [\varepsilon],\quad \varepsilon_{i,p} = \left|\frac{\mathrm{RMS}_{i,p} - \mathrm{RMS}_{i-1,p}}{\mathrm{RMS}_{i-1,p}}\right|,\ \mathrm{RMS}_{i,p} = \sqrt{\sum_{j=0}^{N} e_{i,p}^2(jT)\Big/(N+1)} \tag{19}$$

where $[\varepsilon]$ denotes the threshold of relative convergence. Note that non-zero initial values, i.e. $\lambda_{pq,1,1}$ and $\lambda_{pq,2,1}$ in

$\boldsymbol{\lambda}_{1,p}$, must be assigned for the first iteration such that

$$\boldsymbol{\lambda}_{1,p} = \begin{pmatrix} \lambda_{p1,1,1} & \lambda_{p1,2,1} & \mathrm{L} & \lambda_{pq,1,1} & \lambda_{pq,2,1} & \mathrm{L} & \lambda_{pn,1,1} & \lambda_{pn,2,1} \end{pmatrix}^{\mathrm{T}}$$

3.2 Formulation of the identification Jacobian

As seen from Eq. (12) to Eq. (15), $\boldsymbol{\Phi}_p$ is closely related to B_p, which reflects the dynamics of the plant and the feedback controller of the pth actuated joint. The formal identification of B_p is a tedious and complicated procedure. It is worthwhile pointing out that there are two shortcomings in the algorithm proposed in [19] where a single gradient parameter experiment was implemented for identifying two parameter gradients. The first is the difficulty to achieve explicit expression of $\boldsymbol{\phi}_p^{\mathrm{T}}(t) = \mathrm{L}^{-1}\left(\boldsymbol{\varphi}_p^{\mathrm{T}}(s)\right)$, and the second is the ill-conditioned $\boldsymbol{\Phi}_p$ due to multicollinearity. In order to overcome the problems mentioned above, we propose an improved data-driven method to formulate $\boldsymbol{\Phi}_p$ of an n-DOF parallel manipulator by considering joint coupling effects. In this method, the ith iteration cycle is extended into $2n+1$ physical perturbed-parameter experiments, run sequentially and labeled as status 0 to status $2n$, with the corresponding tracking errors of all actuated joints recorded each time. In status 0 neither velocity nor acceleration feedforward are applied. In status $2q$ both of them are applied only to the actuated joints from 1 to q using $\boldsymbol{\lambda}_{i,p}$ estimated by the $(i-1)$th iteration cycle. Status $2q-1$ is the same as status $2q$ except that acceleration feedforward is not applied to joint q. Following exactly the logic of Eq. (10) for these sub-steps of the ith iteration, this specific arrangement leads to

$$\Delta E_{i,\mathrm{p}}^{(2q-1)} = E_{i,p}^{(2q-1)} - E_{i,p}^{(2q-2)} = -B_p \lambda_{pq,1,i} s\Theta_{d,q}, \quad \Delta E_{i,\mathrm{p}}^{(2q)} = E_{i,p}^{(2q)} - E_{i,p}^{(2q-1)} = -B_p \lambda_{pq,2,i} s^2 \Theta_{d,q}, \quad p,q = 1,2,\mathrm{L}\ ,n \tag{20}$$

or

$$B_p s\Theta_{d,q} = -\frac{\Delta E_{i,\mathrm{p}}^{(2q-1)}}{\lambda_{pq,1,i}}, \; B_p s^2 \Theta_{d,q} = -\frac{\Delta E_{i,\mathrm{p}}^{(2q)}}{\lambda_{pq,2,i}} \tag{21}$$

Substituting Eq.(21) into Eq.(12), yields an alternative expression for $\boldsymbol{\varphi}_{i,p}$

$$\boldsymbol{\varphi}_{i,p} = -\begin{pmatrix} \dfrac{\Delta E_{i,p}^{(1)}}{\lambda_{p1,1,i}} & \dfrac{\Delta E_{i,p}^{(2)}}{\lambda_{p1,2,i}} & \mathrm{L} & \dfrac{\Delta E_{i,p}^{(2q-1)}}{\lambda_{pq,1,i}} & \dfrac{\Delta E_{i,p}^{(2q)}}{\lambda_{pq,2,i}} & \mathrm{L} & \dfrac{\Delta E_{i,p}^{(2n-1)}}{\lambda_{pn,1,i}} & \dfrac{\Delta E_{i,p}^{(2n)}}{\lambda_{pn,2,i}} \end{pmatrix}^{\mathrm{T}} \tag{22}$$

Executing the inverse transform $\mathrm{L}^{-1}\left(\boldsymbol{\varphi}_{i,p}\right)$ of this expression is quite easy, allowing $\boldsymbol{\phi}_{i,p}(jT_s)$ given in Eq. (16) to be obtained explicitly as

$$\boldsymbol{\phi}_{i,p}(jT_s) = -\begin{pmatrix} \dfrac{\Delta e_{i,p}^{(1)}(jT_s)}{\lambda_{p1,1,i}} & \dfrac{\Delta e_{i,p}^{(2)}(jT_s)}{\lambda_{p1,2,i}} & \mathrm{L} & \dfrac{\Delta e_{i,p}^{(2q-1)}(jT_s)}{\lambda_{pq,1,i}} & \dfrac{\Delta e_{i,p}^{(2q)}(jT_s)}{\lambda_{pq,2,i}} & \mathrm{L} & \dfrac{\Delta e_{i,p}^{(2n-1)}(jT_s)}{\lambda_{pn,1,i}} & \dfrac{\Delta e_{i,p}^{(2n)}(jT_s)}{\lambda_{pn,2,i}} \end{pmatrix}^{\mathrm{T}} \tag{23}$$

It is clear that $\boldsymbol{\Phi}_{i,p}$ formulated by this proposed method is independent of C_p and P_p, but is related to the tracking error discrepancies between two consecutive statuses and to the feedforward parameters estimated in the $(i-1)$th iteration cycle.

4. Verification

Experiments to verify the effectiveness of the proposed iterative tuning algorithm for feedforward control were carried out using the 3-DOF parallel mechanism within a newly invented 5-DOF hybrid robot named TriMule600 [29, 30] developed for high-speed machining. The experiments were designed to examine three important issues: (1) convergence of the iterative tuning algorithm; (2) stability of the tuned parameters against velocity changes; and (3) the extrapolation capability of the tuned parameters against trajectory changes. They involved comparative studies of performance with and without considering joint coupling effects (denoted FCwC and FCwoC, respectively). FCwC invokes the full system of Fig.1, whereas FCwoC disconnects the upper left-hand block (equivalent to setting to zero all F_{pq}, $q \neq p$).

Fig.2 and Fig.3 show the side and front views of the CAD model and the realised prototype of the TriMule 600 robot. It is essentially composed of a 1T2R (T-translation, R-rotation) parallel mechanism and an A/C wrist. The parallel mechanism comprises an actuated UPS limb plus a stand-alone 1T1R planar linkage containing two actuated RPS limbs with a properly constrained passive RP limb in between them. The base link of the 1T1R planar linkage is a single, elaborately designed 'three-in-one' part that locates the rear R joints of the two actuated RPS limbs, and the R and P joints of the RP limb, and is also connected with the machine frame by a pair of R joints. Here, R, P, U, and S denote

revolute, prismatic, universal, and spherical joints, respectively, and the underlined P denotes an actuated prismatic joint. For more information about this robot, please refer to [29, 30]. Equipped with an IPC + Turbo PMAC-PCI CNC system, the robot can move the platform reference point *Q* (see Fig.2) within a cylindrical task workspace at a maximum speed and acceleration of 60 m/min and 10 m/s^2, respectively. The dimensional parameters of task workspace are given in Fig.2.

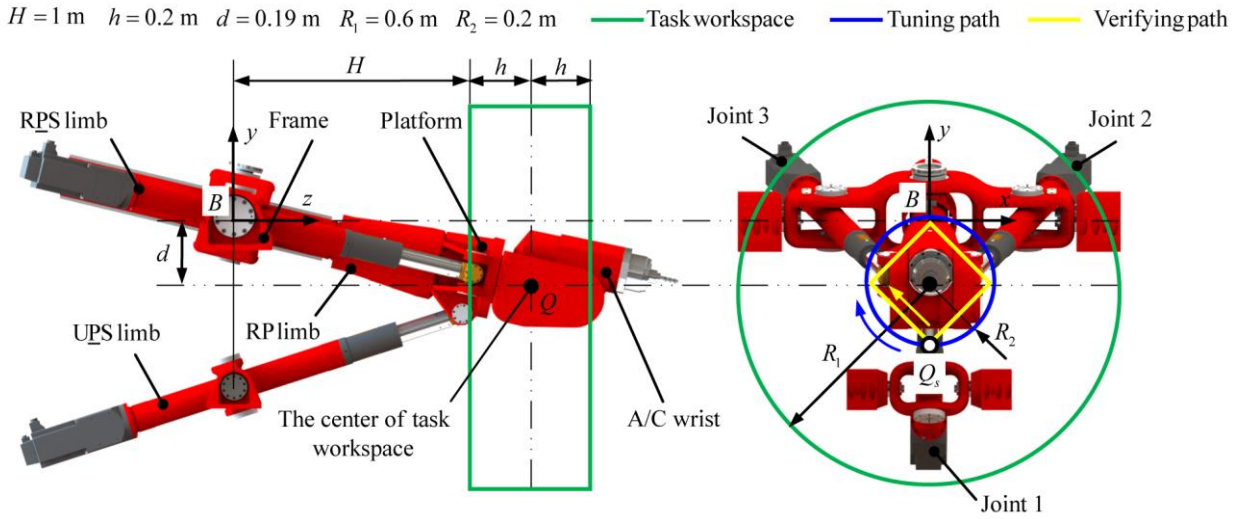

Fig.2 The side and front views of the TriMule600 robot

Fig.3 A prototype of the TriMule600

To verify the effectiveness of the new iterative algorithm, a circular path with radius R_2 was planned for tuning the feedforward parameters and a diamond path with diagonal length of $2R_2$ then used to verify the extrapolation capability, with $R_2 = 0.2$ m, $H = 1$ m, $h = 0.2$ m and $d = 0.19$ m as shown in Fig. 2. In the reference frame $B - xyz$ shown in Fig.2, the position vectors of *Q* associated with the tuning and verification paths are

$$\text{Circular path:}\quad \boldsymbol{r}_Q = \begin{pmatrix} -R_2\sin\left(S_c(t)/R_2\right) & -R_2\cos\left(S_c(t)/R_2\right) - d & H + h \end{pmatrix}^{\mathrm{T}} \tag{24}$$

$$\text{Diamond Path:}\quad \boldsymbol{r}_Q = \begin{cases} \begin{pmatrix} -S_d(t)/\sqrt{2} & -R_2 - d + S_d(t)/\sqrt{2} & H + h \end{pmatrix}^{\mathrm{T}} & \text{Segment 1} \\ \begin{pmatrix} -R_2 + S_d(t)/\sqrt{2} & -d + S_d(t)/\sqrt{2} & H + h \end{pmatrix}^{\mathrm{T}} & \text{Segment 2} \\ \begin{pmatrix} S_d(t)/\sqrt{2} & R_2 - d - S_d(t)/\sqrt{2} & H + h \end{pmatrix}^{\mathrm{T}} & \text{Segment 3} \\ \begin{pmatrix} R_2 - S_d(t)/\sqrt{2} & -d - S_d(t)/\sqrt{2} & H + h \end{pmatrix}^{\mathrm{T}} & \text{Segment 4} \end{cases} \tag{25}$$

where $S_c(t)$ and $S_d(t)$ denote the circular and diamond paths parameterized with a trapezoidal acceleration profile.

In the experiments, $S_c(t)$ and $S_d(t)$ were interpolated with a rate of 10 ms to generate joint commands *via* inverse displacement analysis, and the feedback and feedforward signals updated at a rate of 2.26 kHz in accordance with the specification of the PMAC motion controller used.

Before beginning the feedforward parameters tuning, the feedback (PID) parameters were tuned using the method proposed in [31] to ensure the stability of the feedback loop of each actuated joint. Then, the feedforward tuning procedure was executed, taking the moderate values of 30 m/min and 5.0 m/s^2 for the maximum velocity and acceleration of *Q* along the path. In principle, a set of non-zero yet small values can be randomly selected as the initial parameters assigned for each actuated joint [26, 27]. But for illustration purpose, we select the initial values as

$$\boldsymbol{\lambda}_{1,p} = \left(0.5\times10^{-3} \quad 2\times10^{-3} \quad -0.1\times10^{-3} \quad -0.5\times10^{-3} \quad -0.1\times10^{-3} \quad -0.5\times10^{-3}\right)^{\mathrm{T}}, \quad p = 1,2,3$$

The relative convergent threshold was set to be $[\varepsilon] = 5\ \%$.

Fig. 4 shows the variations of the self-talk parameters $\lambda_{pp,1}$ and $\lambda_{pp,2}$ of the three actuated joints *vs.* iteration cycles for tuning processes with and without considering joint couplings, denoted by FCwC (solid line) and FCwoC (dashed line). Fig. 5 shows the variations of the cross-talk parameters $\lambda_{pq,1}$ and $\lambda_{pq,2}$ ($q \neq p$) *vs.* iteration cycles for FCwC. The feedforward controller parameters for both FCwC and FCwoC rapidly approach near constants after three iteration cycles, illustrating the satisfactory convergence rate of the proposed iterative tuning algorithm. The values for $\lambda_{pp,1}$ and $\lambda_{pp,2}$ identified by FCwoC are somewhat greater than those identified by FCwC because greater self-talk control effort is required in FCwoC to reject the additional dynamic loading disturbance $d_{c,p}$ shown in Eq.(1). Importantly, $\lambda_{pq,1}$ and $\lambda_{pq,2}$ ($q \neq p$) identified by FCwC have values approximately 20% those of $\lambda_{pp,1}$ and $\lambda_{pp,2}$, confirming the necessity to consider these cross-talk terms for rejecting the dynamic loading disturbance arising from joint couplings. From a viewpoint of parameter identification, the use of FCwC fulfills the completeness requirement of the linear regression problem so that the discrepancy between the measured and predicted joint tracking errors can be minimized in the least squares sense.

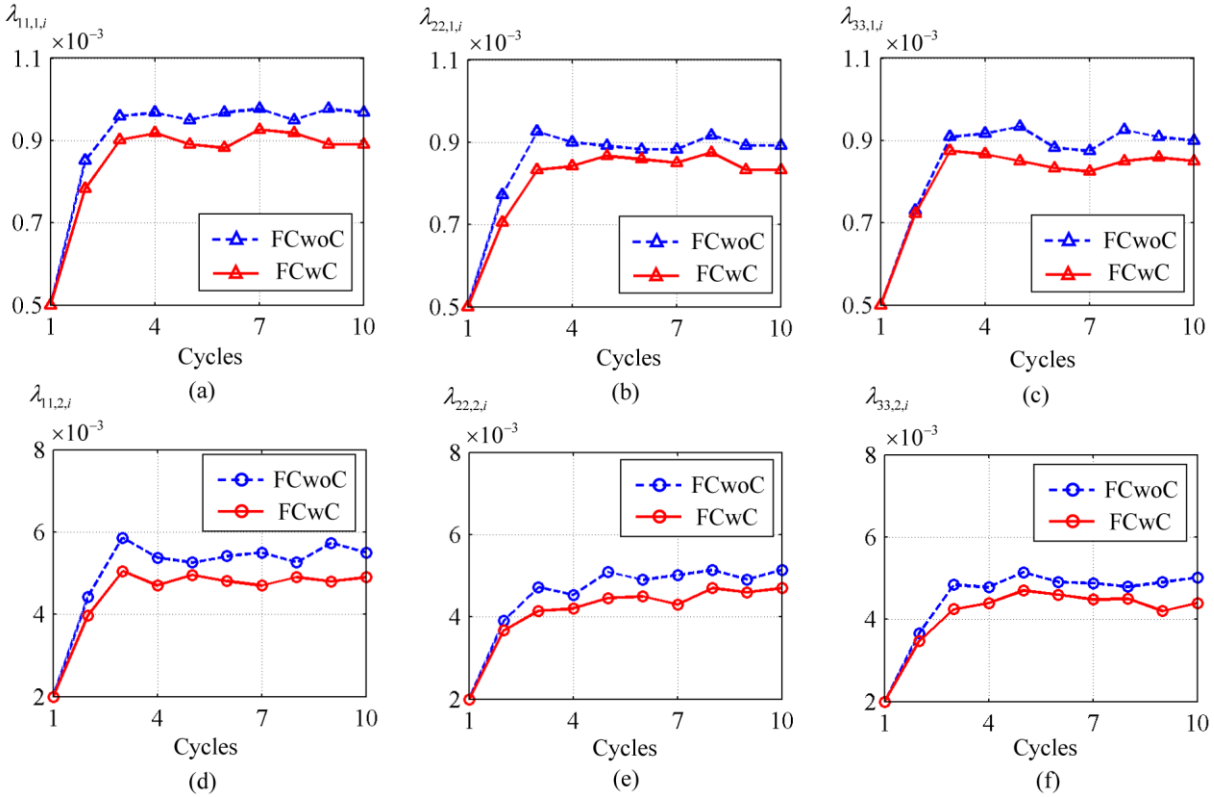

Fig.4. The self-talk feedforward parameters *vs* iterative cycles
(a)-(c) Velocity feedforward, (d)-(f) Acceleration feedforward

Equipped with the inverse kinematics developed in [30]， Fig. 6 and 7 show the desired displacements, velocities and accelerations of three actuated joints, which correspond to the circular and diamond paths planned in the workspace. Having tuned the feedforward controller parameters by FCwoC and FCwC at the moderate speed/acceleration mentioned

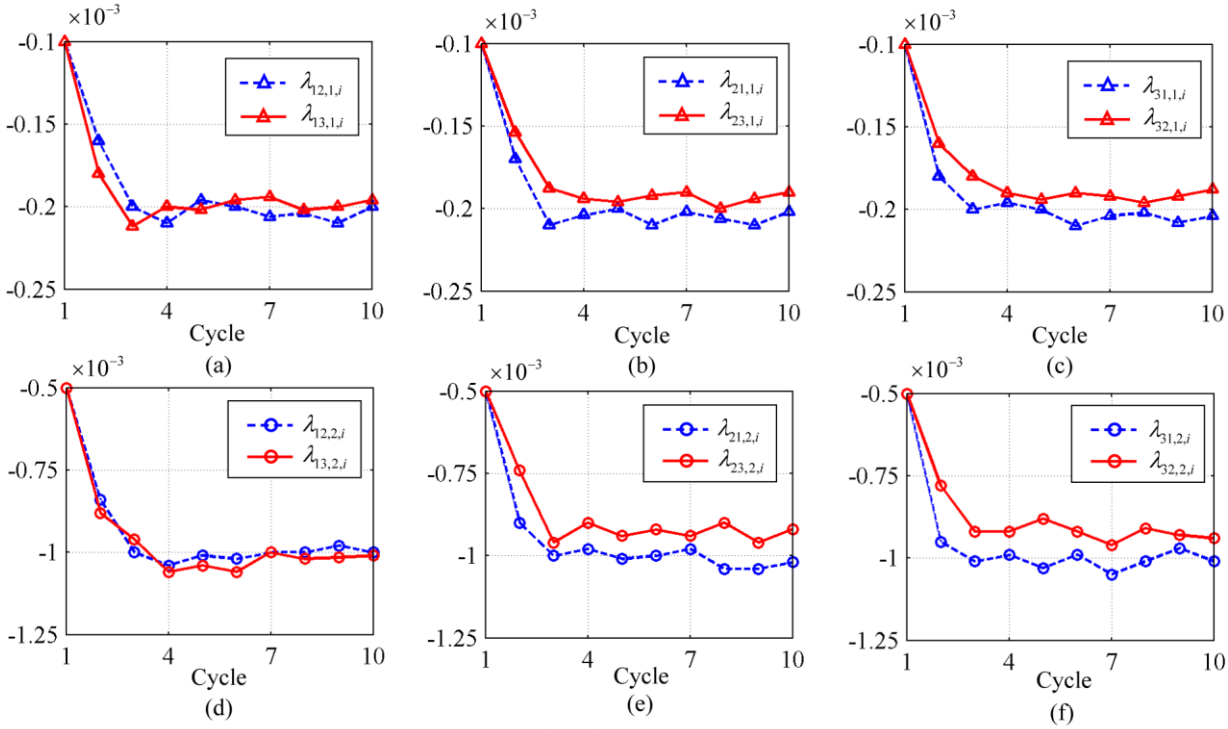

Fig.5 The cross-talk feedforward parameters *vs*. iteration cycles
(a)-(c) Velocity feedforward, (d)-(f) Acceleration feedforward

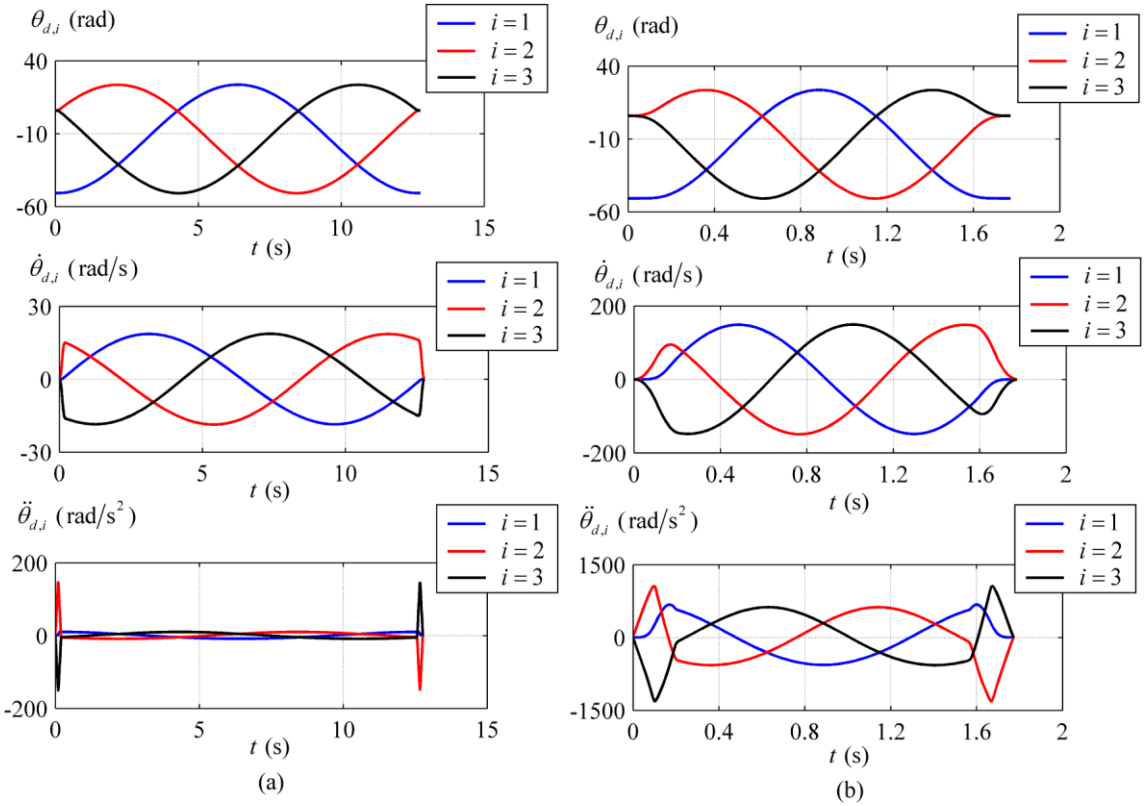

Fig.6 The desired displacement, velocity and acceleration of the actuated joints *vs*. time for the circular path planned in the operation space
(a) $v_{max} = 6$ m/min, $a_{max} = 1$ m/s^2, (b) $v_{max} = 48$ m/min, $a_{max} = 8$ m/s^2

above, the tracking errors of the three actuated joints of the parallel mechanism were recorded when point Q moved along the tuning path and the verifying path at speed/acceleration of $v_{\max}=6$ m/min / $a_{\max}=1$ m/s^2 and of $v_{\max}=48$ m/min / $a_{\max}=8$ m/s^2. The time histories are similar in general form, as shown Fig. 8, but their root mean squares (RMS) listed in Table 1 drop by 5.63%, 5.11% and 5.37% for joints 1, 2 and 3 at low speed, and by 21.24%, 22.22% and 22.61% at high speed for FCwC in comparison with FCwoC. This confirms the argument that considering the cross-talk feedforward parameters helps significantly to improve the joint tracking accuracy, especially when the parallel mechanism operates at high speed. Similar conclusions can be drawn from Fig. 9 and Table 2, in which point Q moves along the verifying path under the same experimental conditions. In this case, the relevant RMS values are 24.24%, 21.71% and 21.57% lower at high speed for FCwC compared to FCwoC. The low magnitudes of these tracking errors confirm the very good extrapolation capability of the tuned feedforward controller parameters against trajectory changes.

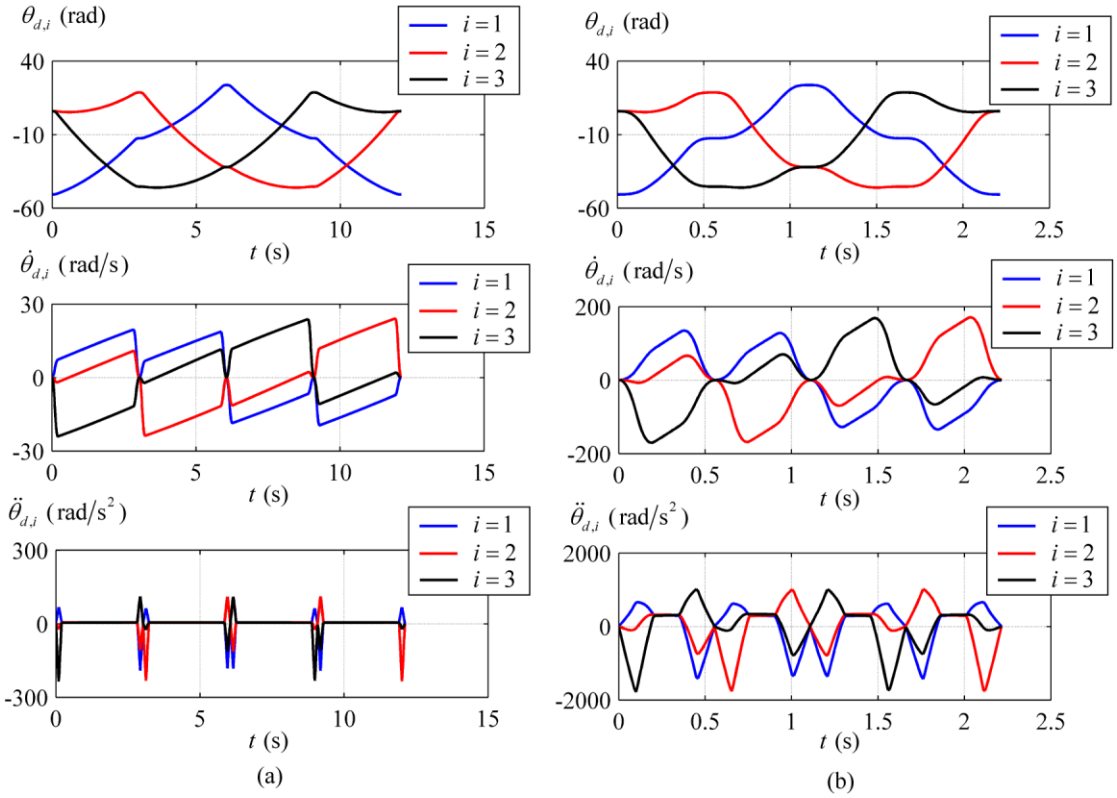

Fig.7 The desired displacement, velocity and acceleration of the actuated joints *vs*. time for the diamond path planned in the operation space

(a) $v_{\max}=6$ m/min, $a_{\max}=1$ m/s^2, (b) $v_{\max}=48$ m/min, $a_{\max}=8$ m/s^2

Table 1 The RMS of joint tracking errors along the tuning path ($\times 10^{-3}$ mm)

	$v_{\max}=6$ m/min, $a_{\max}=1$ m/s^2			$v_{\max}=48$ m/min, $a_{\max}=8$ m/s^2		
	$p=1$	$p=2$	$p=3$	$p=1$	$p=2$	$p=3$
FCwoC	2.13	2.35	2.42	9.04	9.45	9.51
FCwC	2.01	2.23	2.29	7.12	7.35	7.36

Table 2 The RMS of joint tracking errors along the verifying path ($\times 10^{-3}$ mm)

	$v_{\max}=6$ m/min, $a_{\max}=1$ m/s^2			$v_{\max}=48$ m/min, $a_{\max}=8$ m/s^2		
	$p=1$	$p=2$	$p=3$	$p=1$	$p=2$	$p=3$
FCwoC	3.23	2.51	2.51	11.18	9.72	9.69
FCwC	3.02	2.34	2.33	8.47	7.61	7.60

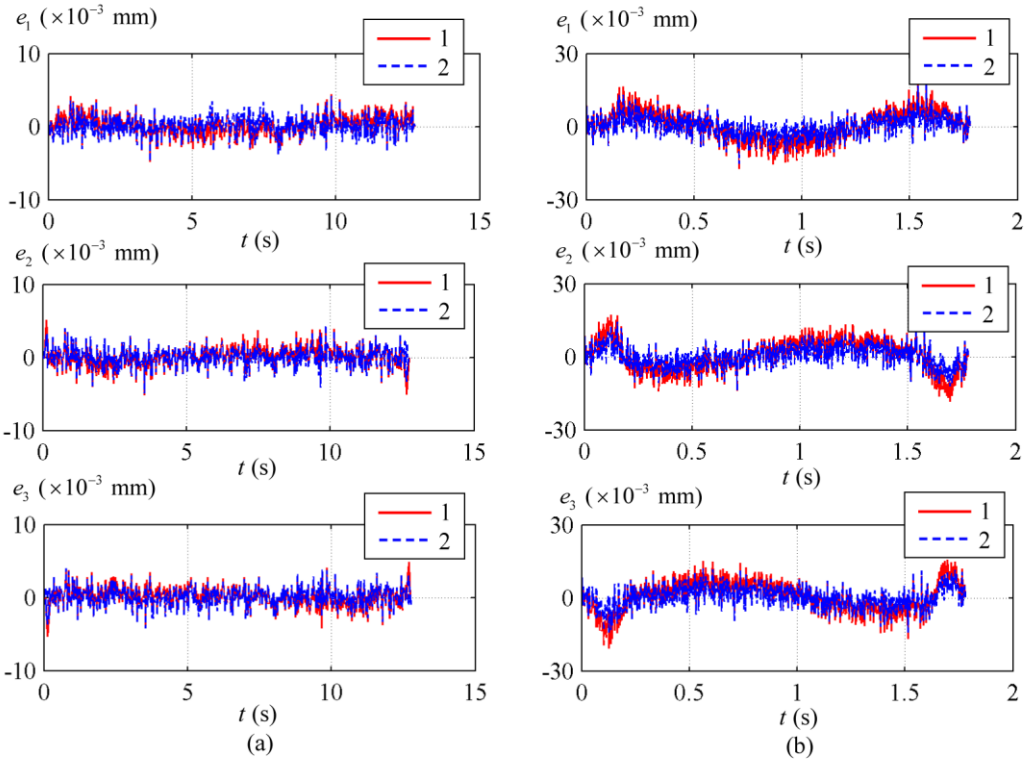

Fig.8 The joint tracking errors *vs* time along the circular path

(a) $v_{max} = 6$ m/min, $a_{max} = 1$ m/s^2, (b) $v_{max} = 48$ m/min, $a_{max} = 8$ m/s^2

1: FCwoC 2: FCwC

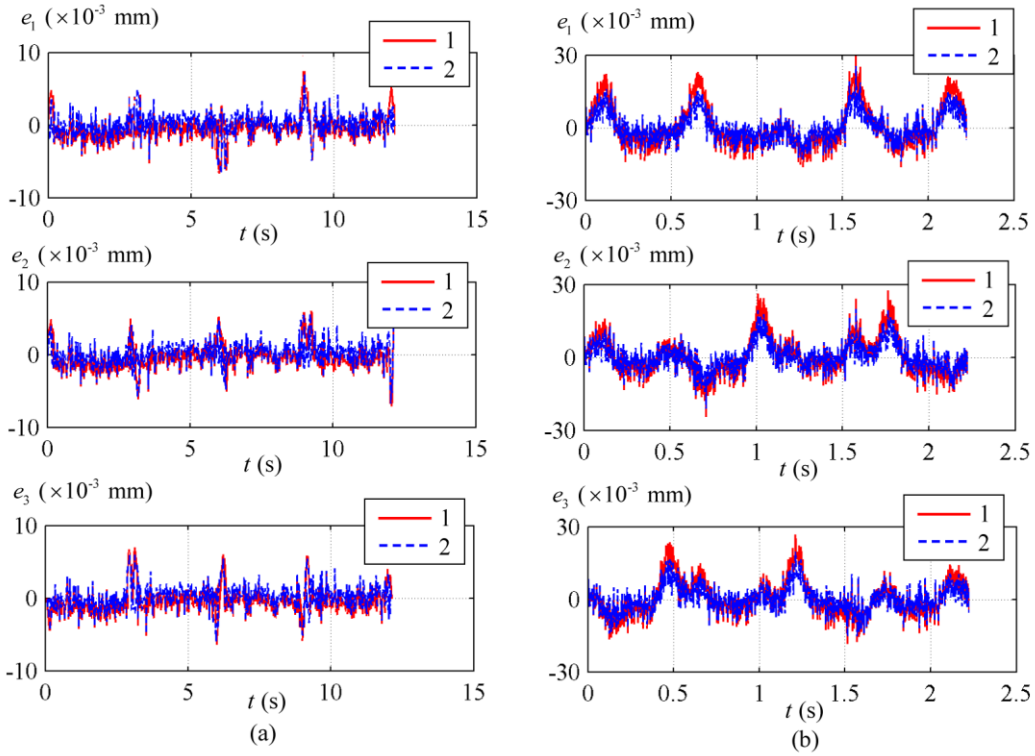

Fig.9 The joint tracking errors *vs* time along the diamond path

(a) $v_{max} = 6$ m/min, $a_{max} = 1$ m/s^2, (b) $v_{max} = 48$ m/min, $a_{max} = 8$ m/s^2

1: FCwoC 2: FCwC

5. Conclusions

This paper explores an improved iterative tuning approach for feedforward control of parallel manipulators that considers joint couplings (cross-talk). The conclusions are drawn as follows.

(1) By combining the invariant principle of compound control and rigid body dynamics of parallel manipulators, we have proposed a MIMO feedforward control design that enables joint tracking accuracy to be improved by dealing with cross-talk torque disturbances.

(2) We have developed a data-driven approach to formulate the plant-free identification Jacobian such that it is only related to the feedforward controller parameters tuned in the previous iteration cycle and the joint tracking errors associated with 2*n* sequential perturbed statuses in the current iteration cycle of an *n*-DOF parallel manipulator.

(3) Experimental results on the 3-DOF parallel mechanism within a 5-DOF hybrid robot show that the parameters tuned by the algorithm converge at a satisfactory rate and provide good extrapolation capability. The results also show that the root mean square of joint tracking errors can be reduced by up to 22% compared with the case where no joint couplings are considered when the system operates at high speed. However, it is unnecessary to consider joint couplings if the system operates at relatively low speed.

(4) The proposed approach is potentially applicable to the feedforward controller parameter tuning of other types of parallel manipulators, whenever their joint couplings are not negligible.

(5) Although the effectiveness of the proposed approach is only demonstrated at the reference configuration, it is suitable for feedforward parameter tuning at any arbitrary configuration (or pose). Therefore, it indeed provides a basic prerequisite for automatic parameter tuning over the entire task workspace by means of polynomial interpolation or fuzzy logical/cluster algorithms. The relevant issue, however, will be reported in a separate article.

Acknowledgements

This work is partially supported by the National Key Research and Development Program (grant 2017YFE0111300), National Natural Science Foundation of China (grant 51420105007, 51622508) and EU H2020-RISE-ECSASDP (grant 734272).

References

[1] M. C. Tsai, I. F. Chiu, M. Y. Cheng, Design and implementation of command and friction feedforward control for CNC motion controllers, IEE Proceedings - Control Theory and Applications 151(1) (2004) 13-20.

[2] P. Lambrechts, M. Boerlage, M. Steinbuch, Trajectory planning and feedforward design for electromechanical motion systems, Control Engineering Practice 13(2) (2005) 145-157.

[3] JA. Butterworth, LY. Pao, DA. Abramovitch, Analysis and comparison of three discrete-time feedforward model-inverse control techniques for nonminimum-phase systems, Mechatronics 22 (2012) 577-87.

[4] B. Siciliano , O. Khatib, Springer handbook of robotics, Springer, 2016 .

[5] T. Hägglund, and K. J. Åström, Industrial adaptive controllers based on frequency response techniques, Automatica 27(4) (1991) 599-609.

[6] L. R. Hunt, G. Meyer, R. Su, Noncausal inverses for linear systems, IEEE Transactions on Automatic Control 41(4) (1996) 608-611.

[7] D. E. Torfs, R. Vuerinckx, J. Swevers, J. Schoukens, Comparison of two feedforward design methods aiming at accurate trajectory tracking of the end point of a flexible robot arm, IEEE Transactions on Control Systems Technology 6(1) (1998) 2-14.

[8] S. Devasia, Should model-based inverse inputs be used as feedforward under plant uncertainty, IEEE Transactions on Automatic Control 47(11) (2002) 1865-1871.

[9] M. Boerlage , M. Steinbuch, P. Lambrechts, M. Van de Wal, Model-based feedforward for motion systems, Proceedings of the 2003 IEEE Conference on Control Applications 2(2003) 1158-1163.

[10] N. Farhat , V. Mata , A. Page, F. Valero, Identification of dynamic parameters of a 3-DOF RPS parallel manipulator, Mechanism and Machine Theory 43(1) (2008) 1-17.

[11] J. Wu, J. Wang, L. Wang, T. Li. Dynamics and control of a planar 3-DOF parallel manipulator with actuation redundancy, Mechanism and Machine Theory 44(4) (2009) 835-849.

[12] Z. Qin, L. Baron, L. Birglen, A new approach to the dynamic parameter identification of robotic manipulators, Robotica 28(4) (2010) 539-547.

[13] J. Wu, J. Wang, Z. You, An overview of dynamic parameter identification of robots, Robotics and Computer Integrated Manufacturing 26(5) (2010) 414-419.

[14] M. Gautier, A. Janot, P. O. Vandanjon, A new closed-loop output error method for parameter identification of robot dynamics, IEEE Transactions on Control Systems Technology 21(2) (2013) 428-444.

[15] J. Wu, G. Yu, Y. Gao, L. Wang, Mechatronics modeling and vibration analysis of a 2-DOF parallel manipulator in a 5-DOF hybrid machine tool, Mechanism and Machine Theory 121 (2018) 430-445

[16] S. Zhao, K. K. Tan, Adaptive feedforward compensation of force ripples in linear motors, Control Engineering Practice 13(9) (2005) 1081-1092.

[17] Y. Wang, F. Gao, F.J. Doyle, Survey on iterative learning control, repetitive control, and run-to-run control, Journal of Process Control 19(10) (2009) 1589–1600.

[18] J. Van de Wijdeven, O.H. Bosgra, Using basis functions in iterative learning control: analysis and design theory, International Journal of Control 83(4) (2010) 661-675.

[19] J. Wu, Y. Han, Z. Xiong, H. Ding, Servo performance improvement through iterative tuning feedforward controller with disturbance compensator, International Journal of Machine Tools & Manufacture 117(2017) 1-10.
[20] F. Boeren, D. Bruijnen, N. V. Dijk, T. Oomen, Joint input shaping and feedforward for point-to-point motion: automated tuning for an industrial nanopositioning system, Mechatronics 24(6) (2014) 572-581.
[21] J. Bolder, T. Oomen, Rational basis functions in iterative learning control-with experimental verification on a motion system, IEEE Transactions on Control Systems Technology 23(2) (2015) 722-729.
[22] F. Boeren, T. Oomen, M. Steinbuch, Iterative motion feedforward tuning: a data-driven approach based on instrumental variable identification, Control Engineering Practice 37(2015) 11-19.
[23] F. Boeren, D. Bruijnen, T. Oomen, Enhancing feedforward controller tuning via instrumental variables: with application to nanopositioning, International Journal of Control 90(4) (2017) 746-764.
[24] S. van der Meulen, R. Tousain, O. Bosgra, Fixed structure feedforward controller design exploiting iterative trials: application to a wafer stage and a desktop printer. Journal of Dynamic Systems, Measurement, and Control 130(5) (2008) 0510061-05100616.
[25] M. Baggen, M. Heertjes, R. Kamidi, Data-based feed-forward control in MIMO motion systems, American Control Conference IEEE (2008) 3011-3016,.
[26] M. Heertjes, D. Hennekens, M. Steinbuch, MIMO feed-forward design in wafer scanners using a gradient approximation-based algorithm, Control Engineering Practice 18(5) (2010) 495-506.
[27] Y. Jiang, Y. Zhu, K. Yang, C. Hu, D. Yu, A data-driven iterative decoupling feedforward control strategy with application to an ultraprecision motion stage, IEEE Transactions on Industrial Electronics 62(1) (2015) 620-627.
[28] M. W. Spong, M. Vidyasagar, Robot dynamics and control, NewYork: Wiley. 1989.
[29] T. Huang, C. Dong, H, Liu, X. Qin, J. Mei, Q. Liu, Five-degree-of-freedom hybrid robot with rotational supports, US Patent 9,943,967, 2018.
[30] T. Huang, C. Dong, H. Liu, T. Sun, D.G. Chetwynd, A simple and visually orientated approach for type synthesis of overconstrained 1T2R parallel mechanisms, Robotica, available online, 2018.
[31] J. Wu, J. Hu, Z. Xiong, H. Ding, Cascaded proportional–integral–derivative controller parameters tuning for contour following improvement, Proc. Inst. Mech. Eng. Part I-J. Syst. Control Eng. 230(9) (2016) 892–904.