Task Space Contouring Error Estimation and Precision Iterative Control of Robotic Manipulators

Chuxiong Hu, *Senior Member, IEEE*, Shize Lin, Ze Wang, and Yu Zhu

***Abstract*—The task space contouring performance is significant for the machining accuracy of industrial robotic manipulators, but the contouring control of end-effector which is important in the industry has received scant attention. In this letter, a novel task space contouring error estimation and control scheme is developed to address the end-effector precision contouring problem of robotic manipulators. In order to quantify manipulator contouring performance, a 6-DOF synchronized contouring error is defined in the task space. Based on the definition, an optimization problem is constructed and solved to achieve high precision contouring error estimation. The proposed method shows excellent estimation accuracy even under extreme contouring conditions, which is the prerequisite for accurate contouring control. Then the solved contouring error is iteratively compensated into the end-effector reference in a feedforward way to achieve contouring error reduction. The effectiveness of the proposed method is derived theoretically under the assumption of small error and nonsingularity. Comparative experiments are conducted on an industrial manipulator. Quantitative results on different extreme trajectories demonstrate that the proposed control scheme not only has advantages in control stability and implementation simplicity, but also outperforms traditional joint space iterative learning control and cross-coupled iterative learning control on the end-effector contouring accuracy. The proposed method actually provides a practical solution to high-precision contouring applications for industrial manipulators.**

***Index Terms*—Contouring error, end-effector contouring control, manipulator, task space Newton-ILC.**

I. Introduction

In THE past decade, industrial manipulators have been widely applied in varieties of manufacturing fields, such as welding [1], precision assembly [2], additive manufacturing [3], etc. Compared to Computer Numerical Control (CNC) machining [4], manipulators with high machining flexibility and large workspace are rather competent for machining tasks. However, the coupled dynamic characteristics and terminal flexibility significantly influence the end-effector machining accuracy, hindering further application of industrial manipulators.

In order to achieve excellent motion control performance and to improve machining quality of industrial manipulators, various researches have been conducted on the enhancement of joint space tracking performance, especially under the influence of disturbances [5], external payloads [6] and uncertainties [7]. Even though, the motion control performance of end-effector cannot be effectively guaranteed through controllers based on joint space tracking error. Therefore, to improve control accuracy of end-effector directly, the exact kinematics information, i.e., Jacobian kinematics, of the manipulator is required to synthesize controllers in task space [8]. Further, to eliminate the effect of kinematic uncertainties and increase end-effector tracking accuracy, many improved versions of task space control methods have been proposed, such as adaptive Jacobian methods [9], [10]. From another perspective, for repetitive contouring tasks, iterative learning control (ILC) is an effective tool to significantly reduce tracking error, both in joint space and task space [11]. In [12], error reduction and time optimality are achieved simultaneously using ILC. However, ILC suffers from performance deterioration if the trajectory changes [13]. For better generalization ability to trajectory variation, various neural-network-based methods have been proposed to model tracking error and generalize to compensate unseen trajectories beyond training set [14].

The above control algorithms essentially realize tracking error reduction in the task space of robotic manipulators. However, in practical machining operation, contouring error, defined as the perpendicular distance between the actual position of the end-effector and the reference path, is the major indicator of the manipulator machining quality [15], whilst the contouring error reduction does not necessarily rely on the improvement of end-effector tracking accuracy. Therefore, advanced contouring control can be more efficient to improve the end-effector contouring quality. Researchers studied the end-effector error dynamics in the task coordinate frame [16], but therein the definition of the end-effector contouring error is unclear and the proposed method is basically a tracking error control scheme. In order to eliminate the end-effector contouring error, Ouyang *et al.* studied the manipulator dynamics in the position domain [17]. A master-slave control strategy was proposed to eliminate tracking error caused by the master motion, thereby reducing contouring error. Furthermore, in [18], the end-effector contouring error was estimated using straight-line approximation and real-time contouring control was conducted based on the Jacobian matrix. Apart from real-time methods, off-line precompensation to the end-effector trajectory is also important for contouring error reduction. Inspired by reference correction method in traditional CNC machining, researchers applied a data-driven way to precompensate the end-effector trajectory [19], in which the measured end-effector contouring error in the last iteration

Manuscript received January 23, 2022; accepted May 18, 2022. Date of publication June 8, 2022; date of current version July 7, 2022. This letter was recommended for publication by Associate Editor J. Zhang and Editor C. Gosselin upon evaluation of the reviewers' comments. This work was supported in part by the National Nature Science Foundation of China under Grants 51922059 and 51775305, in part by the Beijing Natural Science Foundation under Grant JQ19010, and in part by the China Postdoctoral Science Foundation under Grant 2021T140371. *(Corresponding author: Chuxiong Hu; Ze Wang.)*

The authors are with the State Key Laboratory of Tribology, Department of Mechanical Engineering, Tsinghua University, Beijing 100084, China (e-mail: cxhu@tsinghua.edu.cn; linsz20@mails.tsinghua.edu.cn; wangze20@mail.tsinghua.edu.cn; zhuyu@tsinghua.edu.cn).

Digital Object Identifier 10.1109/LRA.2022.3180430

was flipped and added to the opposite side of the end-effector reference, then joint space trajectories were regenerated based on the inverse kinematics. It should be noted that the performance of trajectory precompensation depends largely on the accurate calculation of the contouring error, especially in extreme contouring conditions such as high feed rate, large contour curvature, and sharp corner. The above researches has explored end-effector contouring error control to a certain extent, but in general, the research on contouring control of industrial manipulator is insufficient, and there remains a paucity of systematic methodology.

The problem of contouring error estimation (CEE) and contouring error control (CEC) have been thoroughly studied in the field of multi-axis CNC [20], and some algorithms have reference value for the manipulator case. For a typical 5-axis machine tool including translational axes and rotary axes, there are two common definitions to depict the 5-dimension contouring error, individual definition [20] and synchronized definition [21]. The individual definition ignores the coupling error dynamics between position and orientation error, leading to extra difficulty for CEC. Therefore, more commonly used one is the synchronized definition, where the position contouring error is calculated based on the distance, and the orientation contouring error is synchronized to the same point on the reference path. According to the synchronized definition, various CEE and CEC methods have been proposed to eliminate the contouring error. In [22], CEE was conducted by line approximation, and the estimated contouring error was allocated to each axis and controlled using inverse Jacobian. If the contouring error can be analytically calculated or estimated before the first operation, CEC can be carried out in the form of feedforward precompensation [23]. Previous research in [24] analyzed the essential causes of contouring error theoretically, then implemented model-based feedforward compensation in a 3-axis CNC. However, if the accurate system model is unavailable, following the idea of ILC, trajectory precompensation can also be done in an off-line data-driven way [25]. Compared with real-time CEC, off-line methods are superior in stability and are easier to implement in engineering applications. More recently, some researchers followed the synchronized definition in 5-axis CNC and studied the contouring error suppression for robotic machining [26]. The task space contouring error was estimated online and compensated into the velocity command to achieve simultaneous reduction of 6-DOF end-effector contouring error, which demonstrates that precision contouring control of robotic manipulators can be fulfilled through proper feedback controller design. However, obtaining high-precision contouring for robotic manipulators through off-line compensation still remains a open issue.

In order to fill the lack of research on the contouring quality improvement of industrial robotic manipulators as mentioned above, a novel task space contouring error estimation and control scheme is proposed in this letter. First, a task space 6-DOF contouring error is defined for the end-effector contouring task. In terms of improving estimation accuracy, numerical Newton calculation method [25] is adopted for off-line CEE. Specifically, a quadratic distance function is defined as the optimization object, and the Newton algorithm is utilized to seek the extremum in the vicinity of the reference point. Afterwards, the estimated contouring error is implemented in CEC through iterative reference correction, i.e., the end-effector contouring error in each iteration is learned and compensated to refine the reference, then the joint space trajectories are regenerated accordingly. Finally, the proposed control scheme is tested on a 6-DOF industrial manipulator. The experimental results demonstrate that the proposed method can achieve excellent task space contouring performance and outperform traditional joint space ILC both in terms of stability and contouring quality. The proposed task space Newton-ILC method essentially provides a practical solution for industrial robotic manipulators to meet high-precision contouring requirements.

II. Manipulator Kinematics and Dynamics

The dynamic model of a typical 6-link robot manipulator can be described as

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{G}(\boldsymbol{q}) = \boldsymbol{u} \tag{1}$$

where $\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}} \in \mathbb{R}^{6\times 1}$ denote joint positions, velocities, accelerations, respectively. $\boldsymbol{M}(\boldsymbol{q})$ is the $\mathbb{R}^{6\times 6}$ inertia matrix, $\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} \in \mathbb{R}^{6\times 1}$ represents the Coriolis and centrifugal torques and $\boldsymbol{G}(\boldsymbol{q}) \in \mathbb{R}^{6\times 1}$ denotes the gravitational torques. To achieve asymptotic joint space tracking, joint space PID controller with gravitional compensation is adopted as feedback controller [27]. It must be noted that feedback controller design is not the focus of this letter, and PID can essentially be replaced by any joint or task space controller to stabilize the closed-loop system.

The forward kinematics manipulator is denoted as

$$\boldsymbol{X} = \boldsymbol{F}(\boldsymbol{q}), \quad \dot{\boldsymbol{X}} = \boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}} \tag{2}$$

where $\boldsymbol{X} \in \mathbb{R}^{6\times 1}$ is 6-DOF end-effector position and orientation, $\boldsymbol{F}$ and $\boldsymbol{J}$ denote forward kinematic chain and Jacobian.

Based on the formulated closed-loop system dynamics, the remainder of this letter will develop a systematic CEE and CEC method in a feedforward compensation way to achieve high-precision task space contouring.

III. Task Space Newton-ILC Control Scheme

In this section, we will first give the definition of 6-DOF end-effector contouring error, and an advanced task space Newton-ILC control scheme will be presented. As depicted in Fig. 1, inside the blue frame is the joint space closed-loop plant with feedback controller, and input and output of the plant are associated with the end-effector contour through forward and inverse kinematics. Superscript j denotes iteration index. After each iteration, off-line CEE is conducted using Newton algorithm (purple frame). Then the calculated contouring error will be utilized to refine the reference for next iteration, until contouring error converges, which is shown in the red frame as task space ILC. Each part of the proposed control scheme will be detailed in the following parts.

A. Definition of 6-DOF Task Space Contouring Error

For a 6-DOF industrial manipulator, the task space reference is given and parameterized by time as

$$\boldsymbol{r}_d(t) = \ [\boldsymbol{r}_p(t); \boldsymbol{r}_o(t)] \tag{3}$$

where $\boldsymbol{r}_p(t) \in \mathbb{R}^{3\times 1}$ and $\boldsymbol{r}_o(t) \in \mathbb{R}^{3\times 1}$ are the desired end-effector position and orientation of tool coordinate frame relative to the fixed base coordinate frame, respectively. At time stamp t, the actual end-effector pose is $\boldsymbol{X}(t) = [\boldsymbol{p}(t); \boldsymbol{\vartheta}(t)] = [x(t), y(t), z(t), \varphi(t), \theta(t), \psi(t)]^T$. Inspired by the synchronized definition in 5-axis CNC, end-effector contouring error

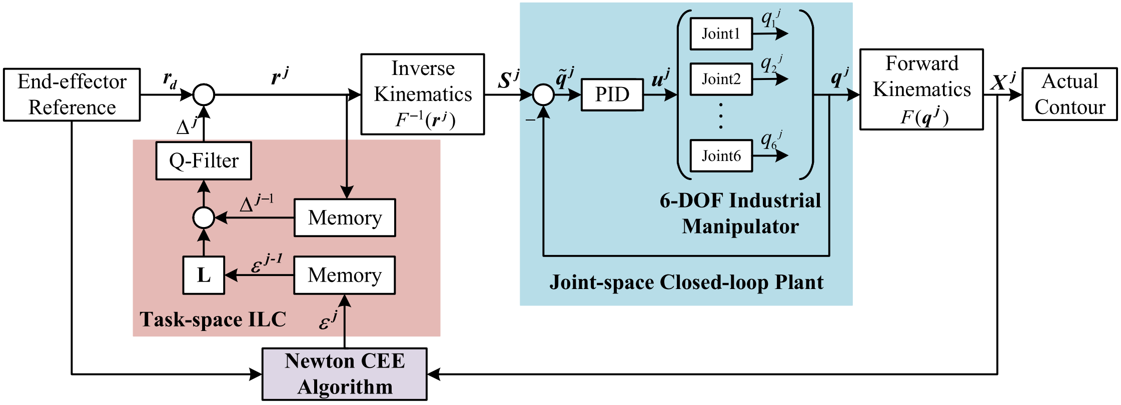

Fig. 1. Task space Newton-ILC control scheme.

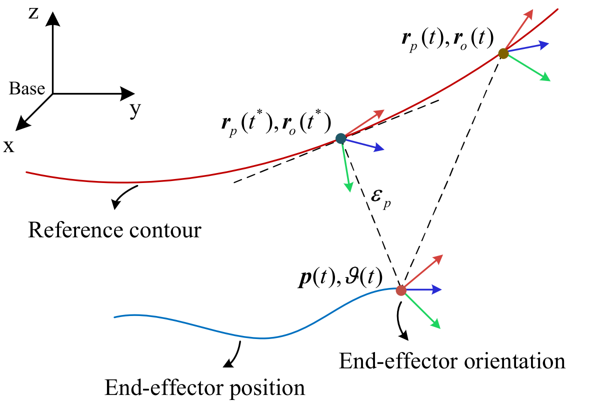

Fig. 2. Definition of end-effector contouring error. $\boldsymbol{r}_p(t)$ and $\boldsymbol{r}_o(t)$ are the reference position and orientation at t, while $\boldsymbol{r}_p(t^*)$ and $\boldsymbol{r}_o(t^*)$ denote reference position and orientation at contouring error point t^*.

is defined in task space for manipulators, as depicted in Fig. 2. Position contouring error is determined by the shortest distance between end-effector position $\boldsymbol{p}(t)$ and the reference contour, denoted as $\varepsilon_p = [\Delta x, \Delta y, \Delta z]^T$. Then orientation contouring error is synchronized to the same time stamp t^*. The actual orientation and the reference orientation can be presented in the format of rotation matrices as $\boldsymbol{R}_o, \boldsymbol{\Theta} \in SO(3)$. Therefore, the orientation contouring error $\Delta\boldsymbol{\Theta}(t)$ can be defined by calculating the angular deviation between the actual orientation at t and the reference orientation at t^*, i.e.,

$$\Delta\boldsymbol{\Theta}(t) = \boldsymbol{\Theta}(t)\boldsymbol{R}_o(t^*)^T \mapsto \boldsymbol{\varepsilon}_o = [\Delta\varphi, \Delta\theta, \Delta\psi]^T \quad (4)$$

where $\mapsto$ means mapping from rotation matrix to rotation angles. It should be noted that the rotation angles are merely used to evaluate the magnitude of orientation contouring error. Whilst the calculation and compensation of orientation contouring error are based on rotation matrices $\Delta\boldsymbol{\Theta}(t)$. Thus, 6-DOF end-effector contouring error can be denoted as

$$\boldsymbol{\varepsilon}(t) = [\boldsymbol{\varepsilon}_p(t); \boldsymbol{\varepsilon}_o(t)] \quad (5)$$

B. Newton-Based end-Effector CEE

In order to perform precise CEE according to the definition discussed in the above section, herein we adopt numerical Newton method [25] to calculate the end-effector contouring error after each iteration. A cost function is defined to depict the distance between the actual position at t and an arbitrary point on the reference path, i.e.,

$$J_c(\hat{t}) = \frac{1}{2}\|\boldsymbol{r}_p(\hat{t}) - \boldsymbol{p}(t)\|^2 \quad (6)$$

where $\boldsymbol{r}_p(\hat{t})$ is the reference position at time stamp $\hat{t}$, and $\|\cdot\|$ means Euclidean norm. The time stamp of the contouring error point, t^* minimizes the cost function, i.e.,

$$t^* = \arg\min_{\hat{t}} J_c(\hat{t}) \quad (7)$$

In order to calculate t^* recursively, considering the Taylor series expansion in the vicinity of $\hat{t}$, one can get

$$J_c'(t^*) \approx J_c'(\hat{t}) + (t^* - \hat{t})J_c''(\hat{t}) \quad (8)$$

where $J_c'(\hat{t}), J_c''(\hat{t})$ denote the first and second derivatives of J_c with respect to $\hat{t}$. The extremum is found when $J_c'(t^*) = 0$, which leads to $t^* = \hat{t} - J_c'(\hat{t})/J_c''(\hat{t})$. Then the recursive Newton algorithm can be described as

$$\hat{t}_{k+1} = \hat{t}_k - \eta\frac{J_c'(\hat{t}_k)}{J_c''(\hat{t}_k)} \quad (9)$$

with

$$\begin{aligned} J_c'(\hat{t}_k) &= (\boldsymbol{r}_p(\hat{t}_k) - \boldsymbol{p}(t))^T\boldsymbol{r}_p'(\hat{t}_k) \\ J_c''(\hat{t}_k) &= (\boldsymbol{r}_p(\hat{t}_k) - \boldsymbol{p}(t))^T\boldsymbol{r}_p''(\hat{t}_k) + \|\boldsymbol{r}_p'(\hat{t}_k)\|^2 \end{aligned} \quad (10)$$

where k is the iteration subscript, and η is the step size.

Remark 1: The boundedness of $J_c'(\hat{t}_k)/J_c''(\hat{t}_k)$ is guaranteed by the smoothness of the reference. Therefore, (9) converges to the global optima if the initial guess $\boldsymbol{r}_p(\hat{t}_0)$, chosen as the reference position $\boldsymbol{r}_p(t)$ in our algorithm, is located inside the convex vicinity of the contouring error point, which always holds when the tracking error is sufficiently small.

The recursive calculation of t^* only depends on the current state $\boldsymbol{p}(t)$ and the reference information. Off-line CEE can be conducted after each iteration, then the calculated contouring error can be compensated to the reference for next iteration. In the case of end-effector reference given by discretized points without explicit expression, reference information at $\hat{t}_k$ cannot be obtained directly if $\hat{t}_k$ is not a sampling point. Linear interpolation is utilized here to approximate $\boldsymbol{r}_p(\hat{t}_k)$, $\boldsymbol{r}_p'(\hat{t}_k)$, and

$\boldsymbol{r}_p''(\hat{t}_k)$. After the calculation of t^*, quaternion interpolation is adopted to calculated synchronized orientation contouring error at t^*. Assuming that t^* is located between two sampling points t_1 and t_2, i.e., $t_1 < t^* < t_2$, then

$$\begin{aligned}\mathcal{Q}(t^*) &= (\mathcal{Q}(t_2)\mathcal{Q}(t_1)^*)^{\Delta t}\mathcal{Q}(t_1)\\ \Delta t &= (t^* - t_1)/(t_2 - t_1)\end{aligned} \tag{11}$$

where $\mathcal{Q}(\cdot)$ is the quaternion representation of orientation at arbitrary given time stamp.

C. ILC-Based CEC

The calculated contouring error will be utilized for iterative compensation and refinement of the end-effector reference in this section. ILC has been widely used in repetitive tracking operation to achieve remarkable error reduction. In order to implement ILC in manipulator end-effector contouring, several assumptions should be made as follows.

Assumption 1: The initial conditions are exactly the same in each iteration, i.e.,

$$\boldsymbol{q}^j(0) = \boldsymbol{q}_0 \tag{12}$$

where $\boldsymbol{q}_0$ is constant initial joint angles. This assumption is quite reasonable in industrial implementations.

Assumption 2: The end-effector contouring error is sufficiently small, and the kinematic Jacobian along the reference is free of singularities, thus joint trajectory deformation caused by end-effector error compensation takes small values, i.e.,

$$\boldsymbol{X}(t) + \boldsymbol{\varepsilon}(t) \approx F(\boldsymbol{q}(t)) + \boldsymbol{J}(\boldsymbol{q}(t))\boldsymbol{\Delta q}(t) \tag{13}$$

$$\boldsymbol{X}(t) + \boldsymbol{QL}\boldsymbol{\varepsilon}(t) \approx F(\boldsymbol{q}(t)) + \boldsymbol{J}(\boldsymbol{q}(t))\boldsymbol{\delta q}(t) \tag{14}$$

where $\boldsymbol{\Delta q}(t) = \boldsymbol{q}_d(t) - \boldsymbol{q}(t)$ and $\boldsymbol{\delta q}(t)$ denote joint space tracking error and joint reference compensation at time stamp t, respectively. $\boldsymbol{QL}\boldsymbol{\varepsilon}(t)$ denotes task space compensation, where $\boldsymbol{Q}$ is the low-pass filter, $\boldsymbol{L}$ is the positive learning gain in ILC.

Assumption 3: Since $\boldsymbol{\delta q}$ takes small values over iteration, the effect of nonlinear coupling can be approximated as repetitive disturbance [27], which leads to the following discretized decoupled state-space model, i.e.,

$$\boldsymbol{q}^j(t+1) = \boldsymbol{\Phi}\boldsymbol{q}^j(t) + \boldsymbol{\Gamma}\boldsymbol{S}^j(t) + \boldsymbol{d}^j(t+1) \tag{15}$$

where j denotes learning iteration, $\boldsymbol{q}^j(t)$ is joint angle at discretized time stamps, $\boldsymbol{S}^j(t)$ is input reference, $\boldsymbol{\Phi} = diag[\phi_1(t), \ldots, \phi_6(t)]$, $\boldsymbol{\Gamma} = diag[\gamma_1(t), \ldots, \gamma_6(t)]$ are 6×6 coefficient matrices, and $\boldsymbol{d}^j(t)$ represents the effect of joint coupling at t. Referring to [28], by defining

$$\begin{aligned}p_{i,\Delta T}(T) &= \phi_i(T-1)\cdots\phi_i(T-\Delta T+1)\gamma_i(T-\Delta T)\\ m_{i,\Delta T}(T) &= \phi_i(T-1)\phi_i(T-2)\cdots\phi_i(T-\Delta T)\\ \Lambda_i^j(T) &= \phi_i(T-1)\phi_i(T-2)\cdots\phi_i(0)q_i^j(0)\\ T &= 1,2,\ldots,N \qquad \Delta T = 0,1,\ldots,T\end{aligned} \tag{16}$$

LTV system (15) can be presented in lift-system format for joint i, i.e.,

$$\begin{aligned}\boldsymbol{q}_i^j &= \left[q_i^j(1), q_i^j(2), \ldots, q_i^j(N)\right]^T\\ &= \begin{bmatrix} p_{i,1}(1) & 0 & \cdots & 0\\ p_{i,2}(2) & p_{i,1}(2) & \cdots & 0\\ \vdots & \vdots & \ddots & \vdots\\ p_{i,N}(N) & p_{i,N-1}(N) & \cdots & p_{i,1}(N)\end{bmatrix}\\ &\quad\times \begin{bmatrix} S_i^j(0)\\ S_i^j(1)\\ \vdots\\ S_i^j(N-1)\end{bmatrix} + \begin{bmatrix}\Lambda_i^j(1)\\ \Lambda_i^j(2)\\ \vdots\\ \Lambda_i^j(N)\end{bmatrix}\\ &\quad+ \begin{bmatrix} 1 & 0 & \cdots & 0\\ m_{i,1}(2) & 1 & \cdots & 0\\ \vdots & \vdots & \ddots & \vdots\\ m_{i,N-1}(N) & m_{i,N-2}(N) & \cdots & 1\end{bmatrix} \times \begin{bmatrix} d_i^j(1)\\ d_i^j(2)\\ \vdots\\ d_i^j(N)\end{bmatrix}\\ &= \boldsymbol{P}_i\boldsymbol{S}_i^j + \boldsymbol{\Lambda}_i^j + \boldsymbol{M}_i\boldsymbol{D}_i^j\end{aligned} \tag{17}$$

where the matrix $\boldsymbol{P}_i$ and $\boldsymbol{M}_i$ are the Markov parameters for input $\boldsymbol{S}_i^j$ and coupling effect $\boldsymbol{D}_i^j$, and $\boldsymbol{\Lambda}_i^j$ is the initial condition response. It should be noted that $\boldsymbol{P}_i$ and $\boldsymbol{M}_i$ are iteration-invariant. Then the integral joint angles in j-th iteration can be denoted as

$$\boldsymbol{q}^j = \left[\boldsymbol{q}_1^j; \boldsymbol{q}_2^j; \boldsymbol{q}_3^j; \boldsymbol{q}_4^j; \boldsymbol{q}_5^j; \boldsymbol{q}_6^j\right] = \boldsymbol{P}\boldsymbol{S}^j + \boldsymbol{\Lambda}^j + \boldsymbol{M}\boldsymbol{D}^j \tag{18}$$

with

$$\begin{aligned}\boldsymbol{P} &= diag\left[\boldsymbol{P}_1, \ldots, \boldsymbol{P}_6\right] \;, & \boldsymbol{S}^j &= \left[\boldsymbol{S}_1^j; \cdots; \boldsymbol{S}_6^j\right]\\ \boldsymbol{M} &= diag\left[\boldsymbol{M}_1, \ldots, \boldsymbol{M}_6\right] \;, & \boldsymbol{\Lambda}^j &= \left[\boldsymbol{\Lambda}_1^j; \cdots; \boldsymbol{\Lambda}_6^j\right]\end{aligned} \tag{19}$$

Considering the end-effector contouring error defined by (4) and (5), under *Assumption* 2, one can get

$$\mathbf{vec}(\varepsilon^j) = \boldsymbol{J}_d\boldsymbol{K}_{nN}\boldsymbol{\Delta q}^j \tag{20}$$

$$\boldsymbol{QL}\mathbf{vec}(\varepsilon^j) = \boldsymbol{J}_d\boldsymbol{K}_{nN}\boldsymbol{\delta q}^j \tag{21}$$

where $\mathbf{vec}(\varepsilon^j) : \mathbb{R}^{n\times N} \mapsto \mathbb{R}^{nN\times 1}$ denotes column vectorization of the contouring error $\varepsilon^j = [\varepsilon^j(1), \varepsilon^j(2), \ldots, \varepsilon^j(N)]$, $\boldsymbol{\Delta q}^j$, $\boldsymbol{\delta q}^j$ are joint space tracking error and compensation in the j-th iteration, $\boldsymbol{K}_{nN}$ is the commutation matrix, and $\boldsymbol{J}_d$ is the Jacobian matrix along the desired joint space reference $\boldsymbol{q}_d$, i.e.,

$$\boldsymbol{J}_d = diag\left[\boldsymbol{J}(\boldsymbol{q}_d(1)), \boldsymbol{J}(\boldsymbol{q}_d(2)), \ldots, \boldsymbol{J}(\boldsymbol{q}_d(N))\right] \tag{22}$$

According to the compensation strategy shown in Fig. 1, the joint space input $\boldsymbol{S}$ is updated by

$$\boldsymbol{S}^{j+1} = \boldsymbol{S}^j + \boldsymbol{\delta q}^j \tag{23}$$

Substituting (18), (21), (23) into (20), the update of end-effector contouring error can be expressed as

$$
\begin{aligned}
\mathbf{vec}(\boldsymbol{\varepsilon}^{j+1}) &= \boldsymbol{J}_d(\boldsymbol{K}_{nN}(\boldsymbol{q}_d - \boldsymbol{q}^{j+1})) \\
&= \boldsymbol{J}_d(\boldsymbol{K}_{nN}(\boldsymbol{q}_d - \boldsymbol{P}\boldsymbol{S}^{j+1} - \boldsymbol{\Lambda}^{j+1} - \boldsymbol{M}\boldsymbol{D}^{j+1})) \\
&= \boldsymbol{J}_d(\boldsymbol{K}_{nN}(\boldsymbol{q}_d - \boldsymbol{P}(\boldsymbol{S}^j + \boldsymbol{\delta q}^j) - \boldsymbol{\Lambda}^{j+1} - \boldsymbol{M}\boldsymbol{D}^{j+1})) \\
&= \boldsymbol{J}_d(\boldsymbol{K}_{nN}(\boldsymbol{\Delta q}^j - \boldsymbol{P}\boldsymbol{\delta q}^j + \boldsymbol{\Delta\Lambda}^j + \boldsymbol{M}\boldsymbol{\Delta D}^j)) \\
&= (\boldsymbol{I} - \boldsymbol{ZQL})\mathbf{vec}(\boldsymbol{\varepsilon}^j) + \boldsymbol{\epsilon}^j \qquad (24)
\end{aligned}
$$

with

$$
\begin{aligned}
\boldsymbol{Z} &= \boldsymbol{J}_d\boldsymbol{K}_{nN}\boldsymbol{P}\boldsymbol{K}_{Nn}\boldsymbol{J}_d^{-1} \\
\boldsymbol{\epsilon}^j &= \boldsymbol{J}_d\boldsymbol{K}_{nN}\boldsymbol{\Delta\Lambda}^j + \boldsymbol{J}_d\boldsymbol{K}_{nN}\boldsymbol{M}\boldsymbol{\Delta D}^j \qquad (25)
\end{aligned}
$$

where $\boldsymbol{K}_{Nn} = \boldsymbol{K}_{nN}^T$ is also commutation matrix. $\boldsymbol{\Delta\Lambda}^j = \boldsymbol{\Lambda}^j - \boldsymbol{\Lambda}^{j+1}$ and $\boldsymbol{\Delta D}^j = \boldsymbol{D}^j - \boldsymbol{D}^{j+1}$ are sufficiently small under *Assumptions 1* and *3*. $\boldsymbol{\epsilon}^j$ is Lipschitz continuous with respect to $\boldsymbol{\Delta\Lambda}^j$ and $\boldsymbol{\Delta D}^j$. Therefore, $\boldsymbol{\epsilon}^j$ is sufficiently small as iteration progresses.

Theorem 1 ([28]): The ILC system (18), (23) is locally *asymptotically stable* (AS), and the end-effector contouring error converges exponentially if and only if

$$\rho(\boldsymbol{I} - \boldsymbol{ZQL}) < 1 \qquad (26)$$

where $\rho(\cdot)$ denotes the spectral radius.

Proof: Lyapunov-style function can be constructed for the ILC system in iteration domain,

$$V(j) = \mathbf{vec}(\boldsymbol{\varepsilon}^j)^T\mathbf{vec}(\boldsymbol{\varepsilon}^j) \qquad (27)$$

The iteration difference of V, using (24), is calculated as:

$$
\begin{aligned}
\Delta V &= V(j+1) - V(j) \\
&= \mathbf{vec}(\boldsymbol{\varepsilon}^j)^T(\boldsymbol{A}^T\boldsymbol{A} - \boldsymbol{I})\mathbf{vec}(\boldsymbol{\varepsilon}^j) + \zeta \qquad (28)
\end{aligned}
$$

where $\boldsymbol{A} = \boldsymbol{I} - \boldsymbol{ZQL}$ and $\zeta = \mathbf{vec}(\boldsymbol{\varepsilon}^j)^T(\boldsymbol{A} + \boldsymbol{A}^T)\boldsymbol{\epsilon}^j + \boldsymbol{\epsilon}^{jT}\boldsymbol{\epsilon}^j$. $\boldsymbol{A}^T\boldsymbol{A} - \boldsymbol{I}$ is negative definite if condition (26) is met, and ζ is sufficiently small according to the discussion above. Therefore, $\Delta V < 0$ can be deduced.

Thus, stability and contouring error convergence of the proposed task space Newton-ILC method are guaranteed.

IV. Experimental Results

Comparative experiments are conducted on a COMAU Racer3 industrial manipulator to verify the effectiveness of the proposed task space Newton-ILC method. The detailed experimental settings are shown in Fig. 3. The sampling frequency of the control board is set as 5 kHz.

A. Performance Indexes

Position contouring error and orientation contouring error are evaluated respectively in the experiments. The following indexes are introduced:

1) $\|\varepsilon_p\|_{rms} = \sqrt{\frac{1}{N}\sum_{k=1}^{N}\|\varepsilon_p\|_2^2}$ is the root-mean-square (RMS) contouring error.
2) $\|\varepsilon_p\|_{max} = \max\|\varepsilon_p\|_2$ is the maximum absolute contouring error.
3) $|\Delta\varphi|_{rms}, |\Delta\theta|_{rms}, |\Delta\psi|_{rms}$ are the root-mean-square synchronized orientation contouring error of 3 dimensions.

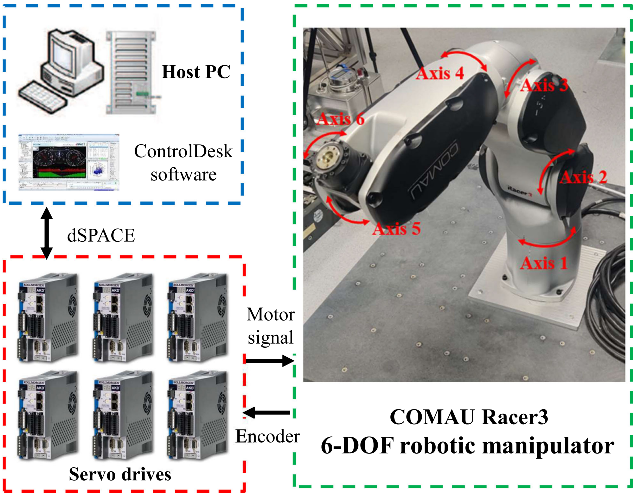

Fig. 3. Experimental setting.

B. Experimental Results

To demonstrate the effectiveness of the proposed task space Newton-ILC control scheme, this letter compares the performances of different control methods in end-effector contouring tasks. Specifically, the following control strategies are compared:

C1: PID—PID controllers are designed for each joint to achieve asymptotic joint space tracking control.

C2: joint space ILC—Based on *C1*, an additional P-type joint space ILC controller is designed to reduce end-effector contouring error indirectly. Herein the learning gain $L = 1$ and the low-pass filter is set as $Q = \frac{(2\pi f_s)^2}{s^2 + 2\zeta(2\pi f_s)s + (2\pi f_s)^2}$ with $f_s = 15\,\mathrm{Hz}$ and $\zeta = 0.7$.

C3: task space CCILC—Based on *C1*, position contouring error is estimated by linear approximation, while orientation contouring error is replaced by orientation tracking error. Then, the end-effector reference is iteratively refined using task space ILC as discussed in Section III-C. The settings of ILC algorithm is the same with *C2*.

C4: task space Newton-ILC—The proposed end-effector contouring error controller, as depicted in Fig. 1. End-effector contouring error is estimated using Newton algorithm and other control parameters are exactly the same with *C3*.

Case 1—Large-Curvature High-Speed Spatial Elliptical Contouring: To test the contouring performance of each controller under extreme conditions, the following parameterized end-effector reference is given for contouring control, i.e.,

$$
\boldsymbol{r}_p = \begin{bmatrix} -\frac{\sqrt{3}}{400}\sin(\mathrm{u}) + 0.55 \\ \frac{1}{10}\cos(\mathrm{u}) \\ \frac{1}{400}\sin(\mathrm{u}) + 0.65 \end{bmatrix}, \qquad \boldsymbol{r}_o = \begin{bmatrix} 0 \\ \pi/6 \\ 0 \end{bmatrix}. \qquad (29)
$$

where u is the contour parameter, which is planned to realize a constant end-effector feed rate at $0.15 m/s$, and the orientation reference $\boldsymbol{r}_o$ is given in the rotation angle format (XYZ). The spatial contour is depicted in Fig. 4.

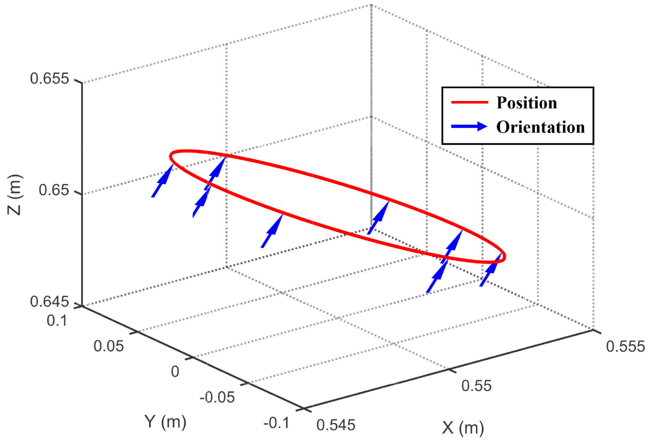

Fig. 4. End-effector reference of Case 1. The blue colored arrow represents the direction of z-axis along the reference.

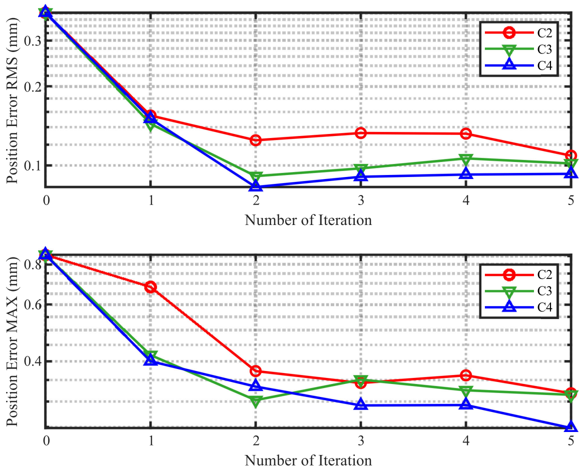

Fig. 5. Position contouring error convergence over iteration. Contouring error of *C1* is exactly the error without iteration, i.e., Iteration 0.

TABLE I
EXPERIMENTAL RESULTS OF CASE 1

Controller	$\|\varepsilon_p\|_{rms}$ (mm)	$\|\varepsilon_p\|_{max}$ (mm)	$\|\Delta\varphi\|_{rms}$ $(mrad)$	$\|\Delta\vartheta\|_{rms}$ $(mrad)$	$\|\Delta\psi\|_{rms}$ $(mrad)$
C1	0.383	0.857	0.773	0.812	0.195
C2	0.109	0.318	0.412	0.215	0.239
C3	0.102	0.315	0.119	0.174	0.075
C4	0.0929	0.248	0.0337	0.161	0.0282

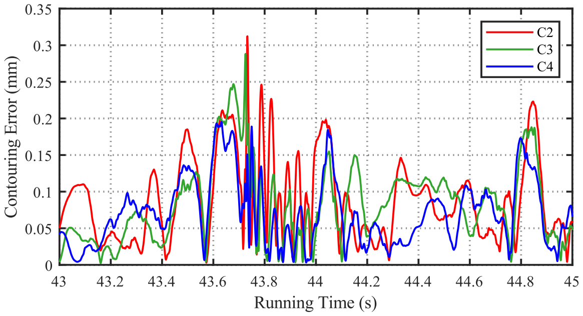

Fig. 6. Position contouring error in the final iteration. For better comparison, *C1* is excluded because its contouring error is significantly larger than other groups.

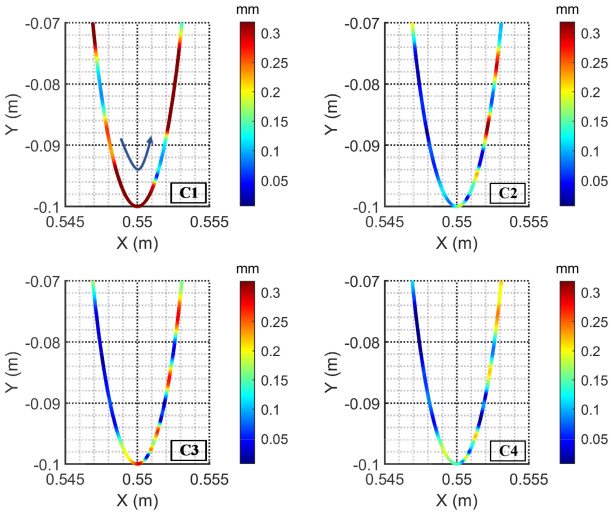

Fig. 7. Position contouring error near extreme condition point, depicted as distribution with respect to spatial position. The direction of motion is marked with blue arrow in *C1*.

The convergence of end-effector position contouring errors in *C2*∼*C4* are depicted in Fig. 5. One can see that the contouring error converges uniformly over iteration, which is consistent with the proof in Section III-C. Specifically, both position contouring error and orientation contouring error in the final iteration are listed in Table I. It can be observed that the proposed task space Newton-ILC method (*C4*) outperforms other controllers both in $\|\varepsilon_p\|_{rms}$ and $\|\varepsilon_p\|_{max}$. Compared with *C3*, contouring error RMS and maximum are reduced by 9% and 21%, respectively, and this error reduction comes to 15% and 22% when compared with *C2*, which implies that the proposed method has significant improvement on the contouring performances near extreme condition points. To better demonstrate this phenomenon, the position contouring errors in the final iteration are shown in Fig. 6. It can be seen that the performance of *C4* is better than others during the whole control process, especially near the error peaks. More intuitively, the position contouring error near extreme condition point is shown in Fig. 7. It is obvious that *C4* has better contouring performance around extreme point $(0.55, -0.1)$ compared with *C3*, which is due to more accurate CEE under large-curvature condition. Meanwhile, although the performance of *C2* is comparable to *C4* around $(0.55, -0.1)$, its contouring error peaks in other segments. This is because *C2* is essentially not designed for contouring error reduction, and error transfer occurs due to improper compensation. Meanwhile, the proposed method achieves synchronous convergence of position and orientation contouring error. It can be noted in Table I that the ultimate orientation contouring error in *C4* is significantly smaller than *C1*∼*C3*, because synchronized orientation contouring error is not considered in these methods. Additionally, since orientation error is evaluated at the contouring error point but not the current reference point, $|\Delta\psi|_{rms}$ of *C2* even deteriorates when compared with uncompensated case (*C1*). Consequently, the actual end-effector contour in *C4* is closest to the 6-DOF reference.

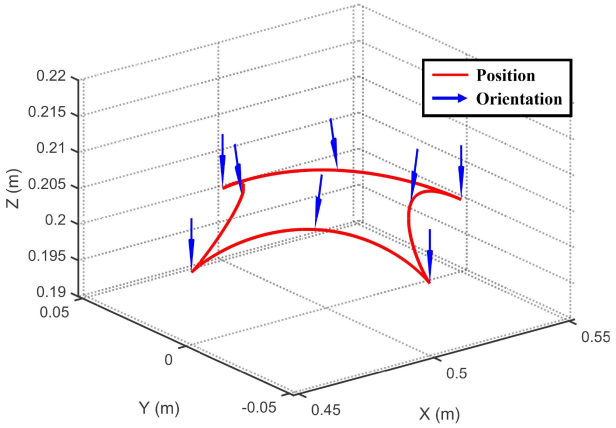

Fig. 8. End-effector reference of Case 2. The blue colored arrow represents the direction of z-axis along the reference.

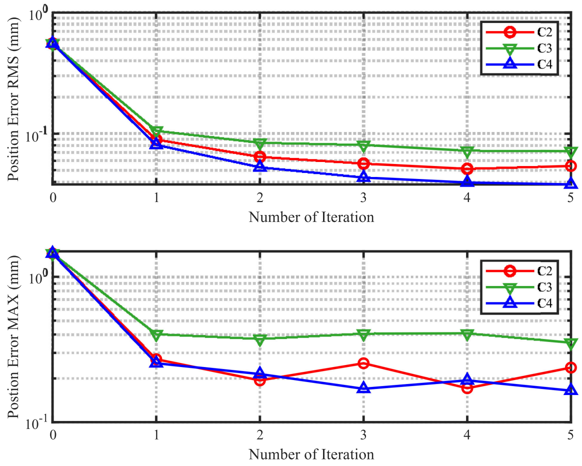

Fig. 9. Position contouring error convergence over iteration. Contouring error of *C1* is exactly the error without iteration, i.e., Iteration 0.

Case 2—Spatial Astroid Contouring: To further demonstrate the effectiveness of the proposed method, a spatial astroid contour with sharp corners is designed as the end-effector reference and tested in this case, as depicted in Fig. 8. The reference contour can be parameterized as

$$\boldsymbol{r}_p = \begin{bmatrix} 0.05\cos^3(\mathrm{u})+0.5 \\ 0.05\sin^3(\mathrm{u}) \\ 0.005\sin^2(2\mathrm{u})+0.2 \end{bmatrix}, \boldsymbol{r}_o = \begin{bmatrix} 0 \\ \mathbf{P2P}: \pi \leftrightarrows \frac{17}{18}\pi \\ \mathbf{P2P}: 0 \leftrightarrows \frac{1}{4}\pi \end{bmatrix}. \tag{30}$$

with the contouring period set as $2s$, i.e., $u = \pi t$. The point-to-point orientation reference $\boldsymbol{r}_o$ is planned to have constrained acceleration.

The convergence of end-effector position contouring errors in *C2*∼*C4* are depicted in Fig. 9. For better illustration, the convergence of synchronized orientation contouring error is also given in Fig. 10. It can be seen that as the iteration progresses, the superiority of the proposed method becomes gradually obvious. Table II demonstrates both position and orientation contouring error in the final iteration. Apparently, the proposed method outperforms other methods in all the performance indexes. For $\|\varepsilon_p\|_{rms}$, the index in *C4* is reduced by 30% and 47% compared with *C2* and *C3*. As for $\|\varepsilon_p\|_{\max}$, *C4* also stands out with 31% and 53% error reduction, respectively. These results verify that the proposed method not only improves contouring performance at extreme condition points, but also achieves considerable contouring error reduction during the whole contouring process. Moreover, the control accuracy of synchronized orientation in *C4* is also improved greatly, which further validates the convergence of the proposed method under synchronized definition. Same as those in Case 1, Figs. 11 and 12 display the position contouring error in the last iteration and its distribution near the sharp corner. It can be observed that the proposed method maintains excellent contouring quality even at the sharp corner. Whereas *C3* suffers from dramatic performance deterioration under the extreme condition due to poor CEE accuracy. It should also be noted that the orientation contouring error in *C3* is even worse than *C1*, which implies that for 6-DOF end-effector contouring, synchronized orientation error should be taken into consideration in CEE for overall contouring error reduction. As for *C2*, the contouring quality is much better than *C3* near the sharp corner, whereas its overall contouring performance is still worse than *C4*, which is consistent with the conclusion of

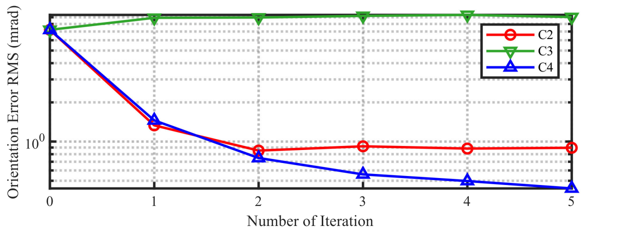

Fig. 10. Synchronized orientation contouring error convergence over iteration. The orientation contouring error is evaluated by $\|\varepsilon_o\| = \sqrt{\Delta\varphi^2 + \Delta\theta^2 + \Delta\psi^2}$, which is used here to compare orientation deviation qualitatively.

TABLE II
EXPERIMENTAL RESULTS OF CASE 2

Controller	$\|\varepsilon_p\|_{rms}$ (mm)	$\|\varepsilon_p\|_{max}$ (mm)	$\|\Delta\varphi\|_{rms}$ $(mrad)$	$\|\Delta\vartheta\|_{rms}$ $(mrad)$	$\|\Delta\psi\|_{rms}$ $(mrad)$
C1	0.553	1.45	6.96	1.69	0.504
C2	0.0541	0.238	0.869	0.213	0.0629
C3	0.0718	0.353	8.74	1.97	0.526
C4	0.0381	0.165	0.413	0.127	0.0309

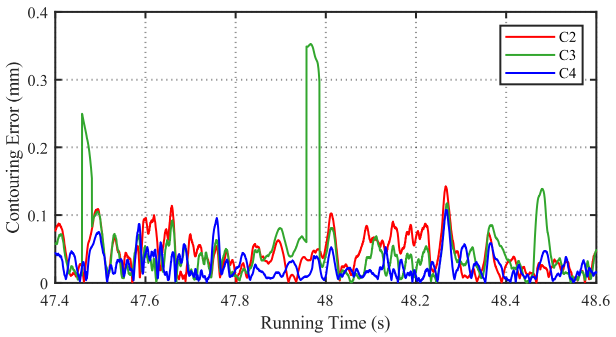

Fig. 11. Position contouring error in the final iteration. For better comparison, *C1* is excluded because its contouring error is significantly larger than other groups.

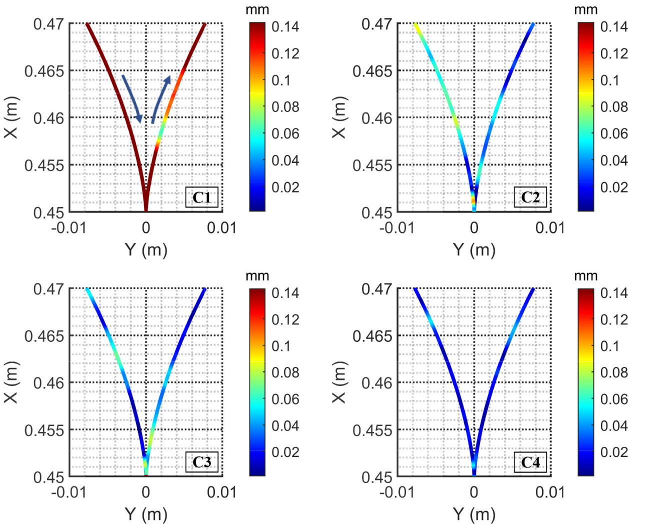

Fig. 12. Position contouring error near extreme condition point, depicted as distribution with respect to spatial position. The direction of motion is marked with blue arrow in *C1*.

Case 1. Therefore, the effectiveness of the proposed task space Newton-ILC method is further validated.

V. Conclusion

In this letter, a task space Newton-ILC control scheme has been developed to improve the end-effector contouring performance of industrial robotic manipulators. In order to better describe end-effector contouring performance, 6-DOF synchronized contouring error is defined in the task space. Based on the above definition, accurate CEE is realized using numerical Newton algorithm. The estimated contouring error is further implemented in task space error compensation to achieve high-quality contouring performance. The stability and error convergence of the proposed method are proved theoretically. Comparative experiments have been carried out on a 6-DOF industrial manipulator. The experimental results consistently indicate that the proposed method is capable of maintaining excellent contouring performance even under extreme conditions, such as high speed, large curvature and sharp corner. The proposed method has advantages in terms of both control accuracy and implementation simplicity, which provides a new practical solution for industrial manipulators in high-precision contouring applications.

References

[1] A. Rout, B. Deepak, and B. Biswal, "Advances in weld seam tracking techniques for robotic welding: A review," *Robot. Comput.- Integr. Manuf.*, vol. 56, pp. 12–37, 2019.

[2] J. Xiao, S. Dou, W. Zhao, and H. Liu, "Sensorless human-robot collaborative assembly considering load and friction compensation," *IEEE Robot. Automat. Lett.*, vol. 6, no. 3, pp. 5945–5952, Jul. 2021.

[3] L. Yuan *et al.*, "Application of multidirectional robotic wire arc additive manufacturing process for the fabrication of complex metallic parts," *IEEE Trans. Ind. Informat.*, vol. 16, no. 1, pp. 454–464, Jan. 2020.

[4] Z. Wang, R. Zhou, C. Hu, and Y. Zhu, "Online iterative learning compensation method based on model prediction for trajectory tracking control systems," *IEEE Trans. Ind. Informat.*, vol. 18, no. 1, pp. 415–425, Jan. 2022.

[5] A. Carron, E. Arcari, M. Wermelinger, L. Hewing, M. Hutter, and M. N. Zeilinger, "Data-driven model predictive control for trajectory tracking with a robotic arm," *IEEE Robot. Automat. Lett.*, vol. 4, no. 4, pp. 3758–3765, Oct. 2019.

[6] J. Hu, C. Li, Z. Chen, and B. Yao, "Precision motion control of a 6-DoFs industrial robot with accurate payload estimation," *IEEE/ASME Trans. Mechatronics*, vol. 25, no. 4, pp. 1821–1829, Aug. 2020.

[7] R. Zhou, C. Hu, Z. Wang, S. He, and Y. Zhu, "Nonlinearity compensation and high-frequency flexibility suppression based RIC method for precision motion control systems," *IEEE Trans. Ind. Informat.*, to be published, doi: 10.1109/TII.2022.3158944.

[8] C.-C. Cheah and H. C. Liaw, "Inverse jacobian regulator with gravity compensation: Stability and experiment," *IEEE Trans. Robot.*, vol. 21, no. 4, pp. 741–747, Aug. 2005.

[9] D. Chen, Y. Zhang, and S. Li, "Tracking control of robot manipulators with unknown models: A jacobian-matrix-adaption method," *IEEE Trans. Ind. Informat.*, vol. 14, no. 7, pp. 3044–3053, Jul. 2018.

[10] D. Lee, W. Lee, J. Park, and W. K. Chung, "Task space control of articulated robot near kinematic singularity: Forward dynamics approach," *IEEE Robot. Automat. Lett.*, vol. 5, no. 2, pp. 752–759, Apr. 2020.

[11] S. Kawamura, F. Miyazaki, and S. Arimoto, "Realization of robot motion based on a learning method," *IEEE Trans. Syst., Man, Cybern.*, vol. 18, no. 1, pp. 126–134, Jan./Feb. 1988.

[12] A. Steinhauser and J. Swevers, "An efficient iterative learning approach to time-optimal path tracking for industrial robots," *IEEE Trans. Ind. Informat.*, vol. 14, no. 11, pp. 5200–5207, Nov. 2018.

[13] J. Shi, J. Xu, J. Sun, and Y. Yang, "Iterative learning control for time-varying systems subject to variable pass lengths: Application to robot manipulators," *IEEE Trans. Ind. Electron.*, vol. 67, no. 10, pp. 8629–8637, Oct. 2020.

[14] T. Ou, C. Hu, Y. Zhu, M. Zhang, and L. Zhu, "Intelligent feedforward compensation motion control of maglev planar motor with precise reference modification prediction," *IEEE Trans. Ind. Electron.*, vol. 68, no. 9, pp. 7768–7777, Sep. 2021.

[15] W. Wang, C. Hu, K. Zhou, and Z. Wang, "Time parameter mapping and contour error precompensation for multi-axis input shaping," *IEEE Trans. Ind. Informat.*, to be published, doi: 10.1109/TII.2022.3158960.

[16] L. Wang, T. Chai, and C. Yang, "Neural-network-based contouring control for robotic manipulators in operational space," *IEEE Trans. Control Syst. Technol.*, vol. 20, no. 4, pp. 1073–1080, Jul. 2012.

[17] P. Ouyang, V. Pano, and J. Acob, "Position domain contour control for multi-DoF robotic system," *Mechatronics*, vol. 23, no. 8, pp. 1061–1071, 2013.

[18] P. Ouyang, Y. Hu, W. Yue, and D. Liu, "Cross-coupled contouring control of multi-DOF robotic manipulator," *Algorithms*, vol. 9, no. 4, 2016, Art. no. 81.

[19] C. Wang, Y. Zhao, Y. Chen, and M. Tomizuka, "Nonparametric statistical learning control of robot manipulators for trajectory or contour tracking," *Robot. Comput.- Integr. Manuf.*, vol. 35, pp. 96–103, 2015.

[20] Z.-Y. Jia, J.-W. Ma, D. N. Song, F. J. Wang, and W. Liu, "A review of contouring-error reduction method in multi-axis CNC machining," *Int. J. Mach. Tools Manufacture*, vol. 125, pp. 34–54, 2018.

[21] A. El Khalick M and N. Uchiyama, "Estimation of tool orientation contour errors for five-axismachining," *Robot. Comput.- Integr. Manuf.*, vol. 29, no. 5, pp. 271–277, 2013.

[22] J. Yang and Y. Altintas, "A generalized on-line estimation and control of five-axis contouring errors of CNC machine tools," *Int. J. Mach. Tools Manufacture*, vol. 88, pp. 9–23, 2015.

[23] X. Yang, R. Seethaler, C. Zhan, D. Lu, and W. Zhao, "A model predictive contouring error precompensation method," *IEEE Trans. Ind. Electron.*, vol. 67, no. 5, pp. 4036–4045, May 2020.

[24] Z. Wang, C. Hu, and Y. Zhu, "Dynamical model based contouring error position-loop feedforward control for multiaxis motion systems," *IEEE Trans. Ind. Informat.*, vol. 15, no. 8, pp. 4686–4695, Aug. 2019.

[25] Z. Wang, C. Hu, Y. Zhu, S. He, M. Zhang, and H. Mu, "Newton-ILC contouring error estimation and coordinated motion control for precision multiaxis systems with comparative experiments," *IEEE Trans. Ind. Electron.*, vol. 65, no. 2, pp. 1470–1480, Feb. 2018.

[26] H. Zhao, X. Li, K. Ge, and H. Ding, "A contour error definition, estimation approach and control structure for six-dimensional robotic machining tasks," *Robot. Comput.- Integr. Manuf.*, vol. 73, 2022, Art. no. 102235.

[27] M. W. Spong, S. Hutchinson, and M. Vidyasagar, *Robot Modeling and Control*. vol. 3, New York, NY, USA: Wiley, 2006.

[28] D. A. Bristow, M. Tharayil, and A. G. Alleyne, "A survey of iterative learning control," *IEEE Control Syst. Mag.*, vol. 26, no. 3, pp. 96–114, Jun. 2006.