TRANSACTIONS OF THE INSTITUTE OF MEASUREMENT & CONTROL

Article

Transactions of the Institute of Measurement and Control
2018, Vol. 40(6) 1970–1982

DOI: 10.1177/0142331217695386
journals.sagepub.com/home/tim
SAGE

Non-linear contour tracking using feedback PID and feedforward position domain cross-coupled iterative learning control

Jie Ling, Zhao Feng, Daojin Yao and Xiaohui Xiao

Abstract

In this paper, a position domain cross-coupled iterative learning controller combining proportional–integral–derivative (PID)-type iterative learning control (ILC) and proportional–derivative (PD)-type cross-coupling control (CCC) is presented aiming at non-linear contour tracking in multi-axis motion systems. Traditional individual control methods in the time domain suffer from poor synchronization of relevant motion axes. The complicated computation of coupling gains in CCC and cross-coupled ILC (CCILC) restricts their applications for non-linear contour. The proposed position domain CCILC (PDCCILC) approach introduces a position domain design concept into CCILC to improve synchronization and performance for non-linear contour tracking and it relies less on the accuracy of coupling gains than conventional CCILC. The stability and performance analysis are conducted using a lifted system representation. The contour error vector method is applied to estimate the coupling gains in simulations and experiments. Simulation and experimental results of three typical non-linear contour tracking cases (i.e. semi-circle, parabola and spiral) based on a two-axis micro-motion stage demonstrate superiority and efficacy of the proposed feedback PID and feedforward PDCCILC compared with existing ILC and CCILC in the time domain.

Keywords

Convergence analysis, cross-coupled iterative learning control, non-linear contour tracking, multi-axis motion systems, position domain control

Introduction

The rapid development of modern industries has resulted in increasing research on high-accuracy manufacturing systems. For a multi-axis system, the motion precision depends on both the individual axis and contour errors. The latter is defined as the distance between the actual position and the nearest point in the reference trajectory (Koren and Lo, 1991).

A conventional control strategy for a multi-axis motion system is independent axis control. A great deal of worthy efforts, such as proportional–integral–derivative (PID) control (Devasia et al., 2007), robust control (Raafat and Akmeliawati, 2012), sliding-mode control (Shen et al., 2014), iterative learning control (ILC; Wang et al., 2015), repetitive control (Shan and Leang, 2012) and polynomial-based pole placement control (Aphale et al., 2008), have been devoted to solving decoupled control problem. However, good tracking performance of each individual axis does not guarantee the reduction of contour errors for multi-axis systems, as poor synchronization of relevant motion axes may result in diminished accuracy of the contour tracking performance (Ouyang et al., 2012). To achieve high-precision contour performance for multi-axis systems, contour error rather than individual-axis error should be emphasized primarily. Cross-coupling control (CCC), developed by Koren (1980), utilizes coupling gains to couple the individual-axis errors of single-input–single-output (SISO) systems together. Barton et al. (Barton and Alleyne, 2006, 2008; Barton et al., 2011) combined CCC with individual-axis ILC to develop a cross-coupling iterative learning control (CCILC) to improve both individual-axis and contour tracking performance in repetitive processes. However, in the CCILC design, the synchronization between axes and the overall contour tracking performance rely heavily on the choice of coupling gains.

For non-linear contours like a circle, parabola and spiral, which are common in 3D printers (Wei et al., 2016), nano-lithography (Paul et al., 2011), nano-precision scanners (Chen et al., 2015; Tuma et al., 2013) and so on, the cross-coupling gains are time-varying and thus difficult to compute out conveniently and precisely. The variable-gain CCC was proposed by Koren and Lo (1991, 1992) for non-linear contour tracking by applying circular contour approximation for arbitrary contour applications. To improve computational efficiency for arbitrary contours in the CCC approach, Yeh and Hsu (2002) developed a modified variable-gain CCC based on the contour

[1]School of Power and Mechanical Engineering, Wuhan University, Wuhan, Hubei, China

Corresponding author:
Xiaohui Xiao, School of Power and Mechanical Engineering, Wuhan University, No.8 South Donghu Road, Wuchang District, Wuhan, Hubei 430072, China.
Email: xhxiao@whu.edu.cn

error vector. The modified method requires fewer operators than the original and may be directly extended to multi-axis motion systems. However, this vector method is an estimation of contour error instead of a precise computation. Any computation inaccuracy of coupling gains may result in weakened tracking performance in a CCILC control system. Therefore, the complicated computation of coupling gains restricts the applications of CCILC into non-linear contour tracking cases.

For multi-axis precision control in 3D nanopositioning, Tien et al. (2005), Wu et al. (2009) and Yan et al. (2012) extended the ILC technique from SISO systems to multi-input–multi-output systems to produce a multi-axis inversion-based iterative control (MAIIC) approach. This is effective for precision tracking in all 3D axes in the presence of a pronounced cross-axis dynamics coupling effect. The plant inversion learning function converges quickly but relies heavily on modelling and is sensitive to model uncertainty, while the PID type ILC is a tunable design that may be applied to a system without extensive modelling and analysis (Bristow, 2006). For different systems, the two types of ILC need to be chosen based on different control problems. Our focus in this paper will be on multi-axes PID type ILC design for the micro-positioning stages, of which the mechanical resonant changes with the increase of load.

Being different from the aforementioned controllers designed in the time domain (abbreviated as TD for the sake of simplicity), Ouyang et al. (Ouyang and Dam, 2011; Ouyang et al., 2012, 2013) proposed a technique of position domain control (PDC) for contour tracking in multi-degree-of-freedom (multi-DOF) robotic systems. A multi-axis motion system in position domain contour control is treated as a master–slave co-operative motion system to guarantee synchronization and improve the contour tracking performance. The PDC was tested effective in contour tracking cases without the problem of coupling gain computation in CCC design. However, all the PDC controllers proposed by Ouyang et al. focused on feedback control. For repetitive tasks, a position domain feedback PID may not achieve optimal performance.

Motivated by non-linear contour tracking in repetitive tasks for a multi-axis motion system, we aim to apply the estimated contour error vector method proposed in Yeh and Hsu (2002) into CCILC to extend its applications. To guarantee the tracking performance under computation inaccuracy of coupling gains, we propose to design a CCILC in the position domain to produce a PDCCILC controller. The proposed PDCCILC may take advantage of the CCILC (i.e. small contour tracking errors) and position domain design procedure (i.e. good synchronization between axes) to achieve ideal tracking performance for non-linear contours. Our previous work (Ling et al., 2015) proposed the basic PDCCILC framework for linear contour tracking. Simulation results verified its effectiveness based on an identified model of the nanopositioning stage. Contributions of this paper lie in: 1) the proposal of combined feedback PID and feedforward PDCCILC design; 2) the extended applications of CCILC into non-linear contour tracking cases under computation inaccuracy of coupling gains when estimating the contour error through contouring error vector method in Yeh and Hsu (2002); 3) the experimental demonstration of the proposed algorithm into a multi-axis precise motion stage.

The outline of this paper is as follows. The next section gives a brief review of CCILC and PDC before introducing PDCCILC. Then, the stability and performance of a combined feedforward PDCCILC and feedback PID control system for non-linear trajectory are analysed. Simulation and experimental results with comparison between existing time domain controllers and the proposed PDCCILC are presented, followed by conclusions and possibilities for future work.

Position domain cross-coupled iterative learning control

Contour error computation and estimation

CCC is a technique to reduce contour error by choosing appropriate coupling gains and coordinating the motion of two coupled axes. Determining coupling gains is vital in CCC, as they are used to calculate contour error and allocate control signal to the individual axis. In an xy-plane contour tracking case, contour error ε may be computed by:

$$\varepsilon = -C_x e_x + C_y e_y \tag{1}$$

For a non-circular contour, a modified variable-gain CCC design based on the contouring error vector was proposed in Yeh and Hsu (2002). The geometrical relations of biaxial motion systems among the desired contour, the actual position P and the reference position R in a biaxial system are shown in Yeh and Hsu (2002, figure 3). The estimated contouring error vector $\overrightarrow{\hat{\varepsilon}}$ is defined as the vector from the actual position to the nearest point on the line that passes through the reference position tangentially with direction $\vec{t}$. The vectors may be directly derived as

$$\overrightarrow{t} = \begin{bmatrix} t_x \\ t_y \end{bmatrix} \tag{2}$$

$$\overrightarrow{n} = \begin{bmatrix} n_x \\ n_y \end{bmatrix} = \begin{bmatrix} -\frac{t_y}{\sqrt{t_x^2 + t_y^2}} \\ \frac{t_x}{\sqrt{t_x^2 + t_y^2}} \end{bmatrix} \tag{3}$$

$$\overrightarrow{\hat{\varepsilon}} = \hat{\varepsilon} \cdot \overrightarrow{n} = \left\langle \overrightarrow{e}, \overrightarrow{n} \right\rangle \cdot \overrightarrow{n}. \tag{4}$$

where $\langle \cdot, \cdot \rangle$ is an inner product operator. As its simplified computation for coupling gain, the modified variable-gain CCC will be adopted in the following design of PDCCILC.

Cross-coupled iterative learning control

The CCILC algorithm may be approached by associating ILC with CCC. A general CCILC structure is presented in Barton and Alleyne (2008). A control law combining an individual-axis ILC algorithm for x- and y-axes with CCILC law was given in Barton et al. (2011) as

$$\begin{bmatrix} u_x \\ u_y \end{bmatrix}_{j+1} = Q\left(\begin{bmatrix} u_x \\ u_y \end{bmatrix} + \begin{bmatrix} L_x & 0 & -C_x L_\varepsilon \\ 0 & L_y & C_y L_\varepsilon \end{bmatrix} \begin{bmatrix} e_x \\ e_y \\ e_\varepsilon \end{bmatrix} \right)_j \tag{5}$$

where u is the control signal, Q a filter, L the learning function, e the individual-axis tracking error and C the coupling gain matrix. The subscripts x, y and ε represent the x-axis, y-axis and the overall contour, respectively.

Feedback position domain PID control

In an xy-plane contour tracking case, the master motion yields zero tracking error because it is sampled equidistantly and used as a reference, and only tracking errors caused by the slave motions will affect the final contour tracking errors (Ouyang and Dam, 2011). A proportional–derivative (PD)-type feedback control signal $u_y(x)$ of the y-axis (slave motion) in the position domain is related to the x-axis position (master motion), which may be expressed as

$$\begin{cases} u_y(x) = K_y^p e_y(x) + K_y^d e_y'(x) \\ e_y(x) = y_d(x) - y(x) \\ e_y'(x) = y_d'(x) - y'(x) \end{cases} \tag{6}$$

where K_y^p and K_y^d are the proportional and differential gains, and e_y is the y-axis tracking error (Ouyang et al., 2012). It should be noticed that the position domain PD law uses the x-axis position as a reference rather than time.

Control law of PDCCILC

For a two-input–two-output, linear time invariant (LTI) system, the PDCCILC control signal of x- and y-axes may be given as

$$\begin{cases} u_{x_{j+1}}^{pdc}(t) = Q_x[u_x^{pdc}(t) + L_x e_x(t) - C_x L_\varepsilon \varepsilon(x)]_j \\ u_{y_{j+1}}^{pdc}(x) = Q_y[u_y^{pdc}(x) + L_y e_y(x) + C_y L_\varepsilon \varepsilon(x)]_j \end{cases} \tag{7}$$

where j is the iteration index, u^{pd} the control signal of the PDCCILC controller, the superscript pdc is PDCCILC, Q is a low-pass filter to improve the control system robustness, L the learning function, $e_x(t)$ the tracking error of the x-axis versus time, and $e_y(t)$ and $\varepsilon(x)$ the tracking error of the y-axis and contour versus the x position.

There are some popular algorithms for designing the learning function in ILC and CCILC. The PID-type learning function may be applied to a system without extensive modelling. The plant inversion learning function converges quickly but relies heavily on modelling and is sensitive to model uncertainty. The H_∞ design technique may be used to design a robustly monotonically convergent ILC but at the expense of performance (Bristow et al., 2006). Although it cannot converge as quickly as the model inversion method, we adopt the PID-type learning function in our PDCCILC design, as we focus more on the model independence and the convenience of tuning parameters. Applying PID-type ILC and PD-type CCC, the ILC term and CCC term of the control law in (7) may be substituted as

$$\begin{cases} L_x e_x(t) = k_{px}^{ilc} e_x(t) + k_{ix}^{ilc} \int_{t-\Delta t}^{t} e_x(t)dt + k_{dx}^{ilc} \dot{e}_x(t) \\ L_y e_y(x) = k_{py}^{ilc} e_y(x) + k_{iy}^{ilc} \int_{s-\Delta s}^{s} e_y(x)dx + k_{dy}^{ilc} e_y'(x) \\ L_\varepsilon \varepsilon(x) = k_{p\varepsilon}^{ccc} \varepsilon(x) + k_{d\varepsilon}^{ccc} \varepsilon'(x) \end{cases} \tag{8}$$

where k_{px}^{ilc}, k_{ix}^{ilc}, k_{dx}^{ilc}, k_{py}^{ilc}, k_{iy}^{ilc} and k_{dy}^{ilc} are the PID gains for the x- and y-axes ILC controller, $k_{p\varepsilon}^{ccc}$ and $k_{d\varepsilon}^{ccc}$ are the PD gains for CCC controller.

To implement the control law in the physical system, the integral and differential items need to be discretized. An approximate calculation was made before discretization, shown as

$$\int_{t-\Delta t}^{t} e_x(t)dt \approx \frac{\Delta t}{2}(e_x(t) + \hat{e}_x(t)) \tag{9}$$

$$\dot{e}(t) \approx \frac{e_x(t) - \hat{e}_x(t)}{\Delta t} \tag{10}$$

$$\int_{s-\Delta s}^{s} e_y(x)dx = \int_{t-\Delta t}^{t} e_y(t)\dot{x}(t)dt \approx \frac{\Delta x(t)}{2}(e_y(t) + \hat{e}_y(t)) \tag{11}$$

$$e_y'(x) = \frac{\dot{e}_y(t)}{\dot{x}(t)} \approx \frac{e_y(t) - \hat{e}_y(t)}{\Delta x(t)} \tag{12}$$

$$\varepsilon'(x) = -C_x(x)e_x'(x) + C_y(x)e_y'(x) - C_x'(x)e_x(x) + C_y'(x)e_y(x) \tag{13}$$

where $\hat{e}(t) = e(t - \Delta t)$. It should be mentioned that Equation (12) is sensitive to the noise under a low speed of master axis motion before the approximation. This problem is solved using the sampling distance of the master motion Δx in the chosen time Δt.

Substituting (8)–(13) into (7) yields the control law for the x-axis as

$$u_{x_{j+1}}^{pdc}(t) = Q_x\left[u_x^{pdc}(t) + \alpha_x(t)e_x(t) + \beta_x(t)\hat{e}_x(t) + \gamma_x(t)e_y(t) + \eta_x(t)\hat{e}_y(t)\right]_j \tag{14}$$

where

$$\begin{cases} \alpha_x(t) = k_{px}^{ilc} + \dfrac{\Delta t}{2}k_{ix}^{ilc} + \dfrac{k_{dx}^{ilc}}{\Delta t} + C_x(t)^2 k_{p\varepsilon}^{ccc} + C_x(t)(2C_x(t) - \hat{C}_x(t))\dfrac{1}{\Delta x(t)}k_{d\varepsilon}^{ccc} \\ \beta_x(t) = \dfrac{\Delta t}{2}k_{ix}^{ilc} - \dfrac{k_{dx}^{ilc}}{\Delta t} - \dfrac{C_x(t)^2}{\Delta x(t)}k_{d\varepsilon}^{ccc} \\ \gamma_x(t) = -C_x(t)C_y(t)k_{p\varepsilon}^{ccc} - \dfrac{C_x(t)(2C_y(t) - \hat{C}_y(t))}{\Delta x(t)}k_{d\varepsilon}^{ccc} \\ \eta_x(t) = \dfrac{C_x(t)C_y(t)}{\Delta x(t)}k_{d\varepsilon}^{ccc} \end{cases} \tag{15}$$

and the control law for the y-axis as

$$u_{y_{j+1}}^{pdc}(t) = Q_y\left[u_y^{pdc}(t) + \alpha_y(t)e_x(t) + \beta_y(t)\hat{e}_x(t) + \gamma_y(t)e_y(t) + \eta_y(t)\hat{e}_y(t)\right]_j \tag{16}$$

where

$$\begin{cases} \alpha_y(t) = -C_x(t)C_y(t)k_{p\varepsilon}^{ccc} + \dfrac{C_y(t)(2C_x(t) - \hat{C}_x(t))}{\Delta x(t)}k_{d\varepsilon}^{ccc} \\ \beta_y(t) = -\dfrac{C_x(t)C_y(t)}{\Delta x(t)}k_{d\varepsilon}^{ccc} \\ \gamma_y(t) = k_{py}^{ilc} + \dfrac{\Delta x(t)}{2}k_{iy}^{ilc} + \dfrac{k_{dy}^{ilc}}{\Delta x(t)} + C_y(t)^2k_{p\varepsilon}^{ccc} + C_y(t)(2C_y(t) - \hat{C}_y(t))\dfrac{1}{\Delta x(t)}k_{d\varepsilon}^{ccc} \\ \eta_y(t) = \dfrac{\Delta x(t)}{2}k_{iy}^{ilc} - \dfrac{1}{\Delta x(t)}k_{dy}^{ilc} - \dfrac{C_y(t)^2}{\Delta x(t)}k_{d\varepsilon}^{ccc}. \end{cases} \tag{17}$$

It may be seen from (15) and (17) that the position information of the x-axis is integrated into the control signal to guarantee the synchronization between the master and slave motions. This is the difference between conventional CCILC in the time domain and the proposed PDCCILC design.

Combined feedback PID with feedforward PDCCILC

In this section, the lifted system framework is reviewed first for the following analysis. Then, the recursion formula of the control signal in a combined PDCCILC and PID control system for non-linear contour tracking ise given. Stability, convergence properties and performance analysis are conducted based on the lifted system representation.

Lifted system representation

The lifted matrix is a method that analyses system stability and convergence using matrix representation of the time-domain system dynamics (Barton et al., 2011). Compared with the general approach for system analysis in the frequency domain, the lifted system matrix is advantageous on analysing time-varying (LTV) dynamics. The PDILC algorithm for the y-axis in this paper contains LTV parameters and therefore the lifted matrix method is adapted.

Consider the discrete-time, linear time-invariant SISO system:

$$y_j(k) = P(q)u_j(k) + d(k) \tag{18}$$

where k is the time index, j the iteration index, y_j the output, u_j the control signal, d the exogenous signal, $P(q)$ the system transfer function with a time delay and q the forward time-shift operator $qx(k) \equiv x(k+1)$.

Applying an impulse input to the system dynamics of (18), the lifted matrix representation may be formed as (19).

$$\underbrace{\begin{bmatrix} y_i(1) \\ y_i(2) \\ \vdots \\ y_i(N) \end{bmatrix}}_{\boldsymbol{Y}_j} = \underbrace{\begin{bmatrix} p_1 & 0 & \cdots & 0 \\ p_2 & p_1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ p_N & p_{N-1} & \cdots & p_1 \end{bmatrix}}_{\boldsymbol{P}} \underbrace{\begin{bmatrix} u_j(0) \\ u_j(1) \\ \vdots \\ u_j(N-1) \end{bmatrix}}_{\boldsymbol{U}_j} + \underbrace{\begin{bmatrix} d(1) \\ d(2) \\ \vdots \\ d(N) \end{bmatrix}}_{\boldsymbol{D}} \tag{19}$$

where $\boldsymbol{Y}_j$, $\boldsymbol{P}$, $\boldsymbol{U}_j$ and $\boldsymbol{D}$ are the lifted representation of y_j, $P(q)$, y_j and d with a matrix form, respectively (Barton et al., 2011). It should be mentioned that the components of $\boldsymbol{Y}_j$ and $\boldsymbol{D}$ are shifted by one time step to accommodate the one-step delay in the plant, which ensures that the diagonal entries of $\boldsymbol{P}$ are nonzero. For an m-step delay, the lifted system representation of $\boldsymbol{Y}_j$ and $\boldsymbol{D}$ then start from $y_j(m)$and $d(m)$. As all of the entries along each diagonal are identical, the lifted framework may easily accommodate LTV dynamics.

Combined PID and PDCCILC design

A combined feedback and feedforward control structure is more effective in precision motion control (Ahn et al., 2007; Devasia et al., 2007; Raafat and Akmeliawati, 2012; Shan and Leang, 2013; Shen et al., 2014). Among feedback control algorithms, PID is commonly used in industrial processes because of its simplicity and the ease of implementation (Ang et al., 2005). By applying a PID technique as the feedback controller and PDCCILC as the feedforward controller, the control signal convergence and contour errors elimination may be achieved effectively.

The combined PID and PDCCILC design for a two-DOF system is shown in Figure 1. The x-axis is chosen as the master motion and the y-axis is the slave motion. The position tracking errors of slave motion versus master motion position are learned by PDCCILC as the master motion position information is integrated into the controller.

To discuss the convergence and error performance in (14) and (16), the recursion formula of control signal should be derived. The error items of $e(t)$ may be substituted as

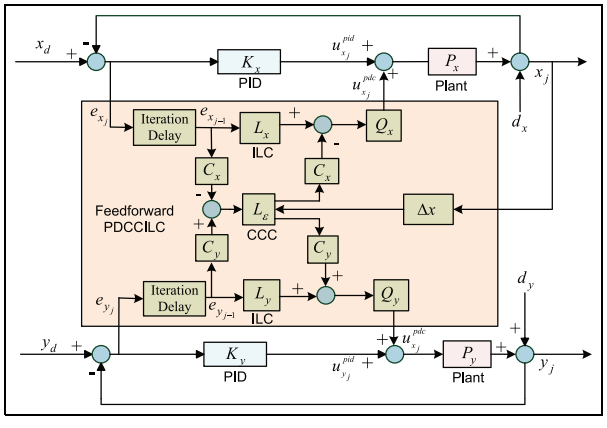

Figure 1. Combined feedback proportional–integral–derivative (PID) and feedforward position domain cross-coupled iterative learning control (PDCCILC) control structure.

$$e_r = r_d - r = r_d - (u_r^{pdc} + u_r^{pid}) \cdot P_r - d_r \tag{20}$$

where $r \in \{\mathrm{x}, \mathrm{y}\}$, $u_r^{pid} = e_r \cdot K_\mathrm{r}$ is the control signal of PID controller with $K_r = k_{pr}^{pid} + k_{ir}^{\mathrm{pid}} \cdot s^{-1} + k_{dr}^{pid} \cdot s$, P_r is the transfer function of axis and the $\hat{e}(t)$ may be substituted as

$$\hat{e}_r = \hat{r}_d - \hat{r} = \hat{r}_d - (u_r^{pdc} + u_r^{pid}) \cdot \hat{P}_r - \hat{d}_r \tag{21}$$

where $\hat{r}_d$, $\hat{P}_r$ and $\hat{d}_r$ represent the signal and system transfer function with a time delay of Δt. Here, e and $\hat{e}$ are short for $e(t)$ and $\hat{e}(t)$ for briefness.

Solving Equations (20) and (21) generates

$$e_r = S_r \cdot (r_d - d_r) - S_r P_r \cdot u_r^{pdc} \tag{22}$$

$$\hat{e}_r = -K_r S_r \hat{P}_r \cdot (r_d - d_r) - S_r \hat{P}_r u_r^{pdc} + (\hat{r}_d - \hat{d}_r) \tag{23}$$

where $S_r = (1 + K_r P_r)^{-1}$ is the sensitivity transfer function of the axis system dynamics.

Substituting (22) and (23) into (14) and (16) yields the recursion formula of PDCCILC control signal for the x- and y-axes as

$$\begin{cases} u_{x_{j+1}}^{pdc}(t) = M_x \cdot u_{x_j}^{pdc}(t) + N_x \\ u_{y_{j+1}}^{pdc}(t) = M_y \cdot u_{y_j}^{pdc}(t) + N_y \end{cases} \tag{24}$$

where

$$\begin{cases} M_r = Q_r(1 - \alpha_r S_r P_r - \beta_r S_r \hat{P}_r) \\ N_r = Q_r\Big[(\alpha_r S_r - \beta_r K_r S_r \hat{P}_r) \cdot (r_d - d_r) + \beta_r(\hat{r}_d - \hat{d}_r)\Big]. \end{cases} \tag{25}$$

Stability, monotonic convergence and performance

Equation (25) then may be converted into lifted matrices using the impulse response as

$$\begin{cases} \boldsymbol{M}_r = \boldsymbol{Q}_r(\boldsymbol{I} - \boldsymbol{\alpha}_r \boldsymbol{S}_r \boldsymbol{P}_r - \boldsymbol{\beta}_r \boldsymbol{S}_r \hat{\boldsymbol{P}}_r) \\ \boldsymbol{N}_r = \boldsymbol{Q}_r\Big[(\boldsymbol{\alpha}_r \boldsymbol{S}_r - \boldsymbol{\beta}_r \boldsymbol{K}_r \boldsymbol{S}_r \hat{\boldsymbol{P}}_r) \cdot (\boldsymbol{r}_d - \boldsymbol{d}_r) + \boldsymbol{\beta}_r(\hat{\boldsymbol{r}}_d - \hat{\boldsymbol{d}}_r)\Big] \end{cases} \tag{26}$$

where $\boldsymbol{M}_r$ and $\boldsymbol{N}_r$ are $N \times N$ matrices, $\boldsymbol{\alpha}_r$ and $\boldsymbol{\beta}_r$ are diagonal matrices, $\boldsymbol{r}_d$, $\hat{\boldsymbol{r}}_d$, $\boldsymbol{d}_r$ and $\hat{\boldsymbol{d}}_r$ are $N \times 1$ matrices.

Let $\rho(\boldsymbol{A}) = \max|\lambda_k(\boldsymbol{A})|$ be the spectral radius of the matrix $\boldsymbol{A}$ and $\lambda_k(\boldsymbol{A})$ the kth eigenvalue of $\boldsymbol{A}$. The control system is asymptotically stable (AS) if there exists appropriate parameters of PDCCILC and PID controllers satisfying

$$\rho(\boldsymbol{M}_r) < 1 \tag{27}$$

The spectral radius condition in (27) satisfies the stability criteria, but does not ensure monotonicity in the iteration domain. A sufficient condition for monotonic convergence of the combined system is given by the upper bound value of the induced norm as

$$\bar{\sigma}(\boldsymbol{M}) < 1 \tag{28}$$

where

$$\boldsymbol{M} = [\,\boldsymbol{M}_x \quad \boldsymbol{M}_y\,]^T$$

is the combined matrix of the two axes. It should be noted that $\rho(\boldsymbol{M}_r) \le \bar{\sigma}(\boldsymbol{M})$ for the matrix $\boldsymbol{M}$. Thus condition (28) ensures monotonicity and stability of the given system simultaneously.

The performance of a control system is often judged by the decrease from the initial error to converged error and convergence speed. While the monotonic convergence condition is satisfied, the asymptotic control input and steady state error of the system may be computed in (29) and (30). In the following simulation and experiment data analysis, the root mean square (RMS) of the error is chosen as the evaluation index for the proposed algorithm.

$$\underbrace{\begin{bmatrix} u^{ilc}(1) \\ u^{ilc}(2) \\ \vdots \\ u^{ilc}(N) \end{bmatrix}_\infty}_{\boldsymbol{U}_r} = \underbrace{\begin{bmatrix} 1-m_1 & 0 & \cdots & 0 \\ -m_2 & 1-m_1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ -m_N & -m_{N-1} & \cdots & 1-m_1 \end{bmatrix}^{-1}}_{\boldsymbol{M}_r} \cdot \underbrace{\begin{bmatrix} n(1) \\ n(2) \\ \vdots \\ n(N) \end{bmatrix}}_{\boldsymbol{N}_r} \tag{29}$$

$$\underbrace{\begin{bmatrix} e(1) \\ e(2) \\ \vdots \\ e(N) \end{bmatrix}_\infty}_{\boldsymbol{e}_r} = \underbrace{\begin{bmatrix} r(1) \\ r(2) \\ \vdots \\ r(N) \end{bmatrix}}_{\boldsymbol{r}_d} - \underbrace{\begin{bmatrix} p_1 & 0 & \cdots & 0 \\ p_2 & p_1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ p_N & p_{N-1} & \cdots & p_1 \end{bmatrix}}_{\boldsymbol{P}_r} \cdot \underbrace{\begin{bmatrix} u^{ilc}(1) \\ u^{ilc}(2) \\ \vdots \\ u^{ilc}(N) \end{bmatrix}_\infty}_{\boldsymbol{U}_r} - \underbrace{\begin{bmatrix} d(1) \\ d(2) \\ \vdots \\ d(N) \end{bmatrix}}_{\boldsymbol{d}_r} \tag{30}$$

Overall computation procedure

The overall computation procedure of the proposed PID & PDCCILC control is described in Figure 2. There are mainly two stages concerning computation.

Stage 1 is the reference pre-processing. First, the reference contour to be tracked needs to be discretized. The sampling time *ts* is chosen according to two principles: 1) meeting the Shannon's sampling theorem, which is the upper limit; 2) satisfying that the matrix size in (19) is within the PC's memory capacity, which is the lower limit. The second step is the determination of the time delay index Δt in (9) and (10).

Stage 2 contains the lifted matrix computation. Before computation, the Q-filter and PID & PDCCILC gains need to be chosen. Then, the lifted matrix in (26) may be obtained using *Impulse* and *Toeplitz* commands in MATLAB software.

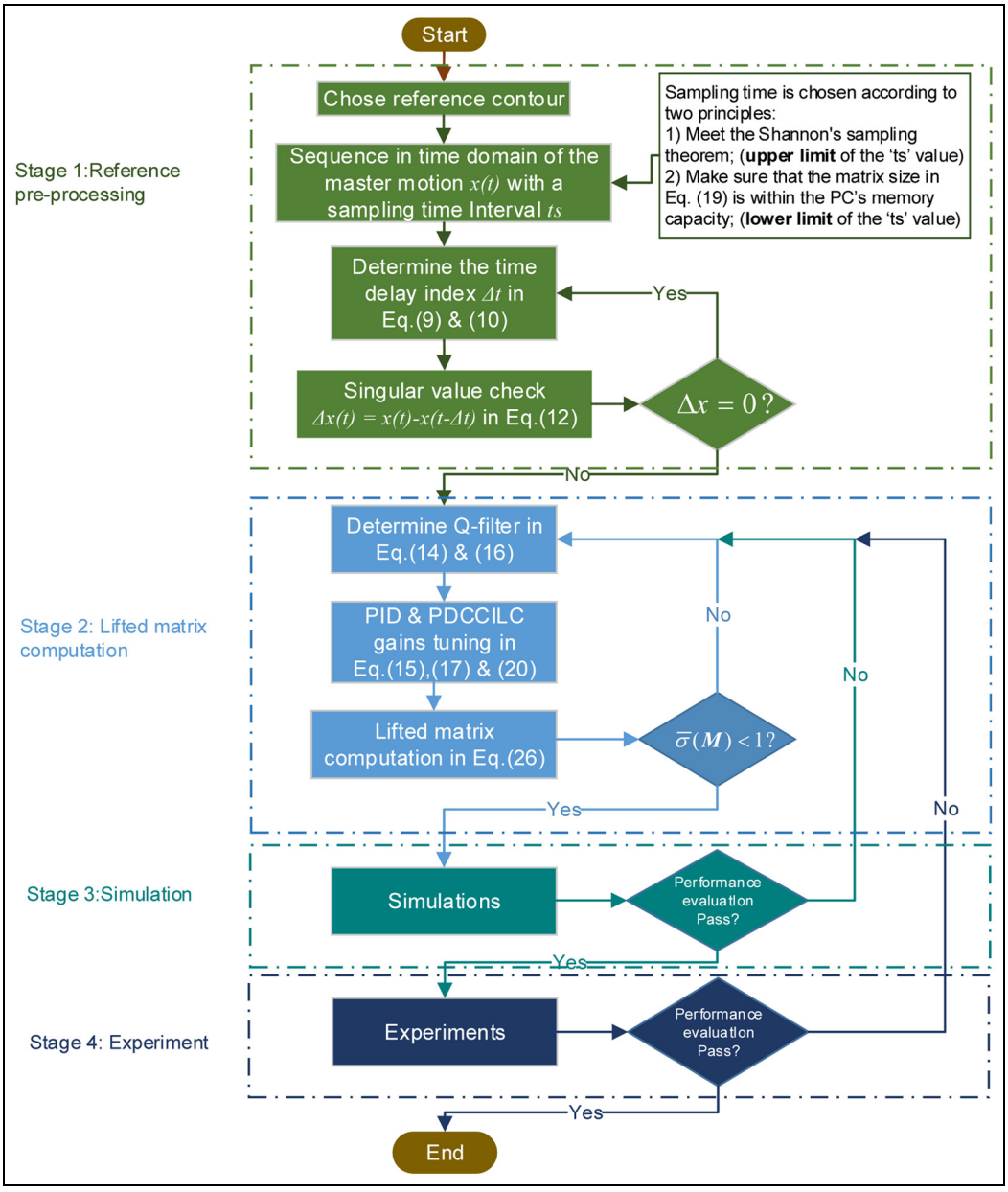

Figure 2. The overall computation procedure for the proposed proportional–integral–derivative (PID) and position domain cross-coupled iterative learning control (PDCCILC).

Evaluation: computation, simulation and experiment

In this section, the combined PID and PDCCILC design was evaluated through both simulation and experiment cases.

System description

As shown in Figure 3, the tested platform is a three-DOF precise positioning stage with a stroke of 50, 50 and 20 mm for the *x*-, *y*- and *z*-axes, respectively. It is a serial system with its each axis driven by a DC servo motor. The position information of each axis is detected by a grating ruler with the resolution of 0.5 μm. For such a commercial product, the original controller is commonly designed decoupled for each individual axis. Therefore, the contour error always exists, as the synchronization between axes is not guaranteed. The proposed PDCCILC design improves the contour error in this testbed in the following sections.

Before simulations, the system model needs to be identified. In this work, only the *x*- and *y*-axes were selected for simulation and experiment cases. Dynamic models of the *x*- and *y*-axes were achieved through step response method with a sample rate of 1 kHz. The continuous transfer functions are:

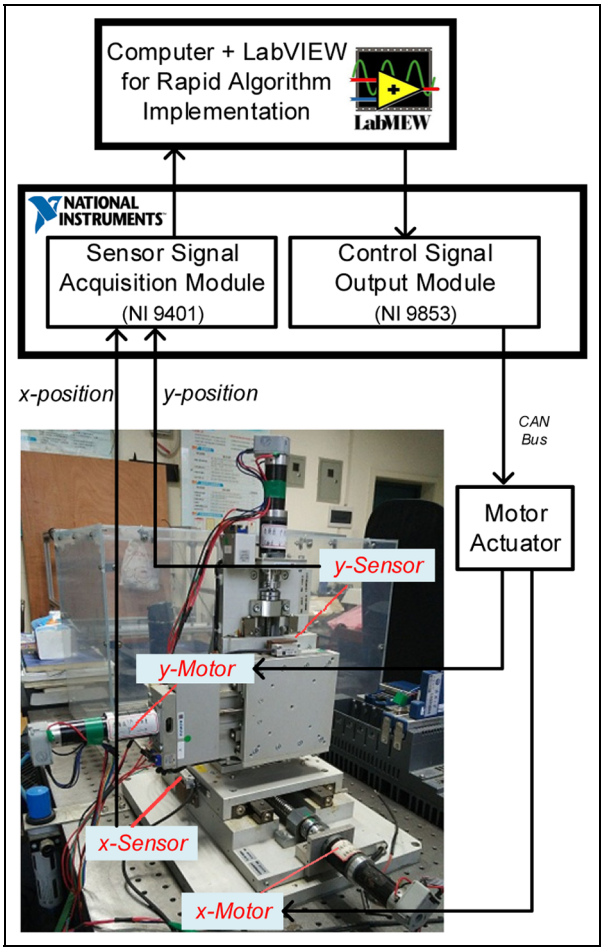

Figure 3. The three-degree-of-freedom precise positioning stage used as the testbed.

$$\begin{cases} P_x = \dfrac{-0.1402s + 5.291}{s^2 + 5.795s + 5.564} \\ P_y = \dfrac{-0.0631s + 2.132}{s^2 + 2.76s + 2.127} \end{cases} \tag{31}$$

Then, the Q-filter in (14) and (16) for x- and y-axes is chosen. The most generally applicable approach to achieving monotonic convenience is to apply a low-pass Q-filter to disable learning at high frequencies, as stated in Bristow et al. (2006). From the bode plot in Figure 4, it may be found that the bandwidth of the x-axis is very close to that of the y-axis, so the Q-filter for the two axes is set the same. Here, we designed a Butterworth low-pass filter with a cut-off frequency at 0.5 Hz in (32) according to the working bandwidth of the two axes.

$$Q_x = Q_y = \frac{31.01}{s^3 + 6.283s^2 + 19.74s + 31.01} \tag{32}$$

Computation results

The monotonic convergence condition in (28) was computationally verified before simulations and experiments. Three

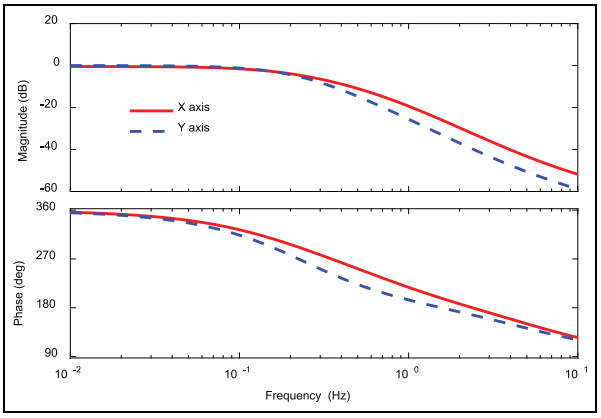

Figure 4. Bode plot of x- and y-axes. The bandwidths of x- and y-axes are 0.164 and 0.162 Hz.

Table 1. Controller parameters.

	Gains		
Controller	K_p	K_i	K_d
PID	3	1	0
ILC	0.3	0.1	0.1
CCC	1	0.5	0

PID, proportional–integral–derivative; ILC, iterative learning control; CCC, cross-coupling control.

sets of non-linear reference contours were used to evaluate the proposed PDCCILC as shown in Figure 5.

The maximum time length of the three reference contours is 12 s. Setting the step size in simulations and experiments as 0.005 s, then the dimensions of the monotonic convergence condition in (28) will be 2400 $\times$ 1200. This makes it compatible with the memory and calculation capabilities in a normal computer. If the matrix size is too big, some indirect computation methods may be applied, as described in Barton and Alleyne (2008).

Following the above preparation work is the selection and tuning of parameters in PID and PDCCILC. The well-known Ziegler–Nichols method is adopted for tuning of the PID parameters. Then, the most commonly employed method for selecting the gains of the PID-type learning function is by tuning, as described in Bristow et al. (2006).

For comparison with the proposed PDCCILC & PID, the conventional TDCCILC & PID, the PDILC (set the CCC gains in PDCCILC as zeros) & PID and the TDILC & PID were applied in computation, simulation and experiment sections, and all the parameters of the four sets were set the same as displayed in Table 1. It should be mentioned that sufficient comparison work has been performed in Ouyang et al. (2012, 2013) between PDC and CCC, and the superiority of CCILC has been verified in Barton and Alleyne (2006, 2008) and Barton et al. (2011) compared with other control algorithms. Therefore, we here focus on the comparison between PDCCILC and TDCCILC in non-linear contour tracking

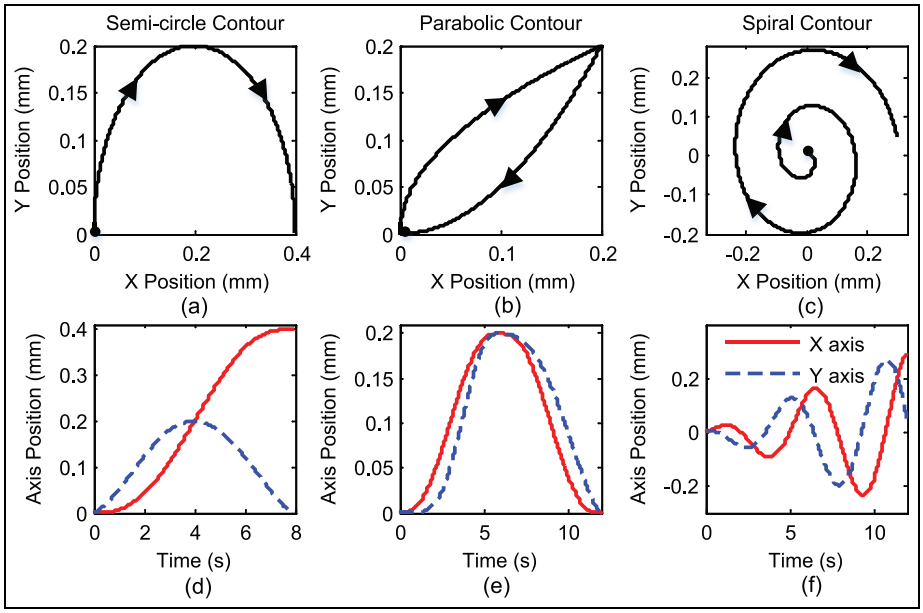

Figure 5. Reference contours. Contours in (a), (b) and (c) start from the bold dot and move along with the arrows; (d), (e) and (f) are the *x*- and *y*-axes positions versus time of the three contours, respectively.

Table 2. Computational results of monotonic convergence.

	Computational results of $\bar{\sigma}(\boldsymbol{M})$		
Control method	Semi-circle	Parabolic	Spiral
PID & PDCCILC	0.878	0.879	0.875
PID & TDCCILC	0.901	0.903	0.895
PID & PDILC	0.916	0.899	0.898
PID & TDILC	0.925	0.912	0.902

PID, proportional–integral–derivative; PDCCILC, position domain cross-coupled iterative learning control; TDCCILC, time domain cross-coupled iterative learning control; PDILC, position domain iterative learning control; TDILC, time domain iterative learning control.

cases. The PDILC & PID and TDILC & PID groups are presented to testify the effectiveness of the CCC in PDCCILC design.

The computational convergence results of the four sets of controllers (PDCCILC & PID, TDCCILC & PID, TDILC & PID and PDILC & PID) are shown in Table 2. As Table 2 illustrates, the upper bound condition of (31) is valid.

Simulation results

Using the PID gains and learning functions presented in Table 1, three typical non-linear trajectories were used in simulations. The RMS contour errors versus iteration for the three type contours in the simulations are shown in Figure 6.

Figure 6(a) shows the RMS contour errors for the four sets of controllers. The control system of PDCCILC & PID results in a 93% decrease of the RMS contour error from the initial to the last iteration, which is the best improvement among the

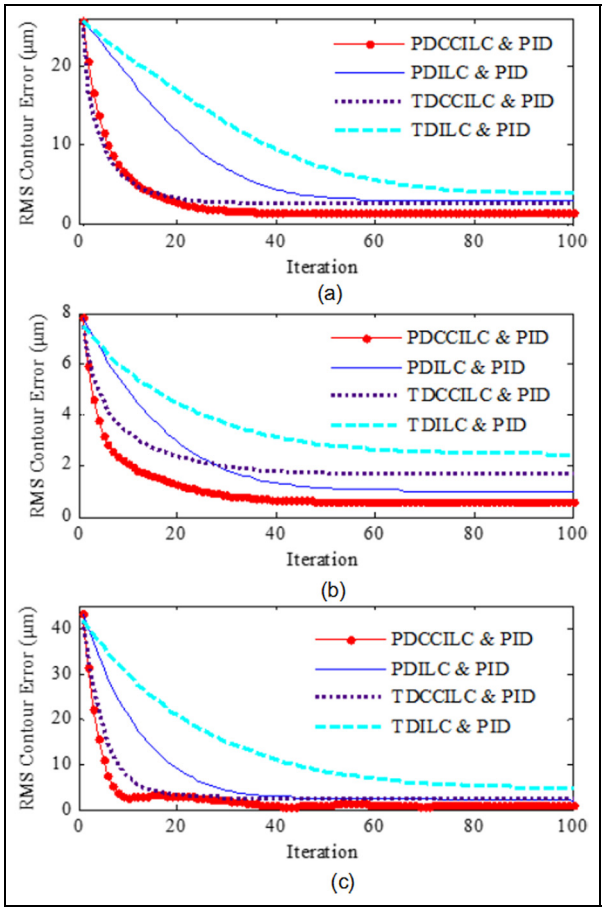

Figure 6. Root mean square (RMS) contour error versus iteration for the three contours in the simulations: (a) semi-circular contour; (b) parabolic contour; (c) spiral contour.

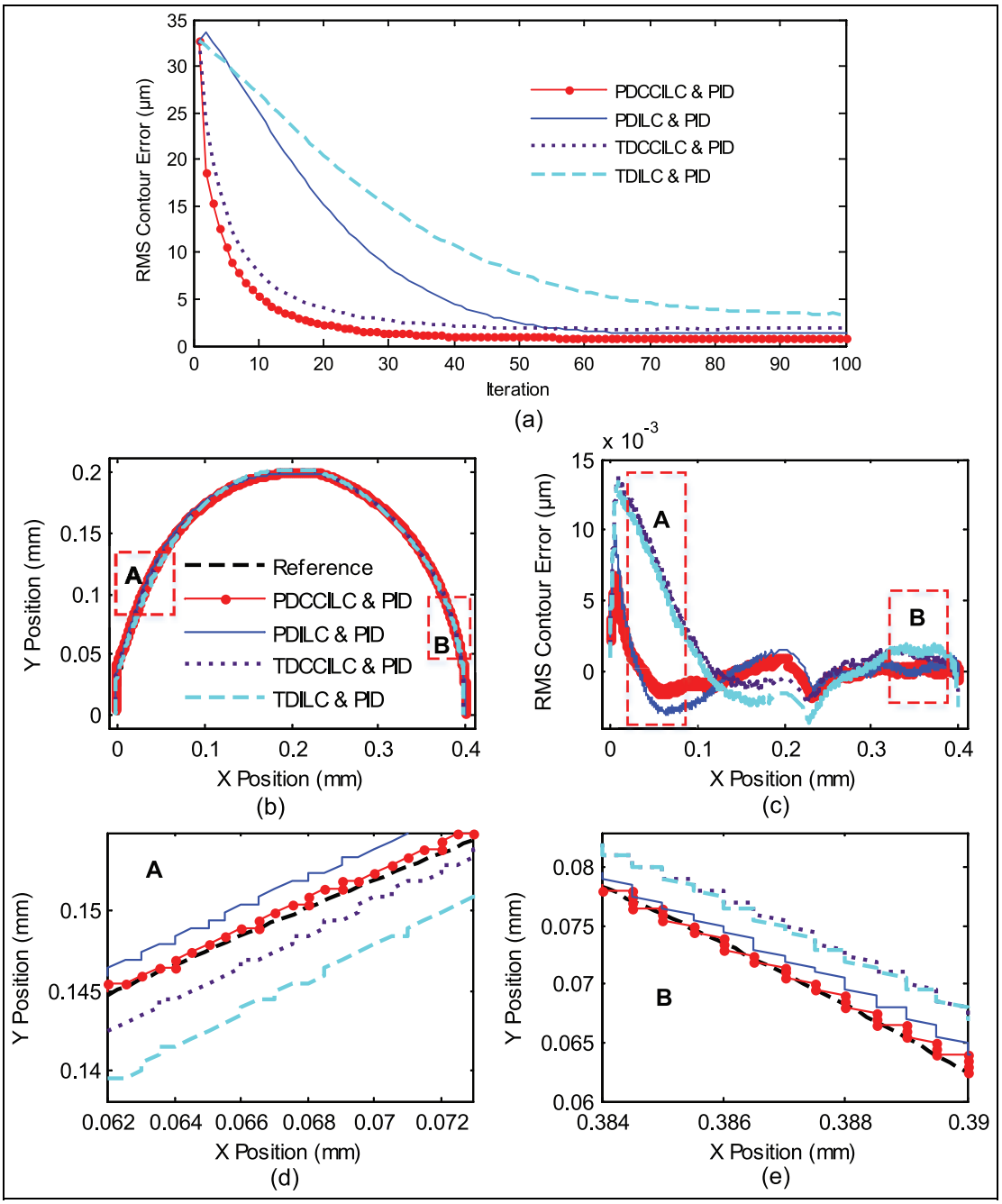

Figure 7. Tracking results of the semi-circle contour in the experiments: (a) root mean square (RMS) contour error versus iteration; (b) the overall look in the *xy*-plane; (c) contour errors versus *x*-axis position; (d) and (e) are the partial enlarged view of parts A and B in (b).

four sets. The TDCCILC & PID combination is superior to the left controller sets in the iteration process.

Results of the parabolic contour tracking case are displayed in Figure 6(b). It may be seen that PDCCILC & PID results in the best improvements from the initial to the last iteration (about 93% decrease). The PDILC & PID set achieves the second best performance, which is different from the results of semi-circle contour tracking.

Results of the spiral contour tracking case are displayed in Figure 6(c). The PDCCILC & PID combination results in a 98% decrease for the RMS contour error in the iteration process, which is the best among the four. The TDCCILC & PID achieves a similar result compared with PDILC & PID control.

From the results of the above four sets of controllers, it may be seen that the PDCCILC & PID may always achieve the best performance among the listed four sets, whereas a PDC or CCC alone may not guarantee a consistent result for different non-linear contours.

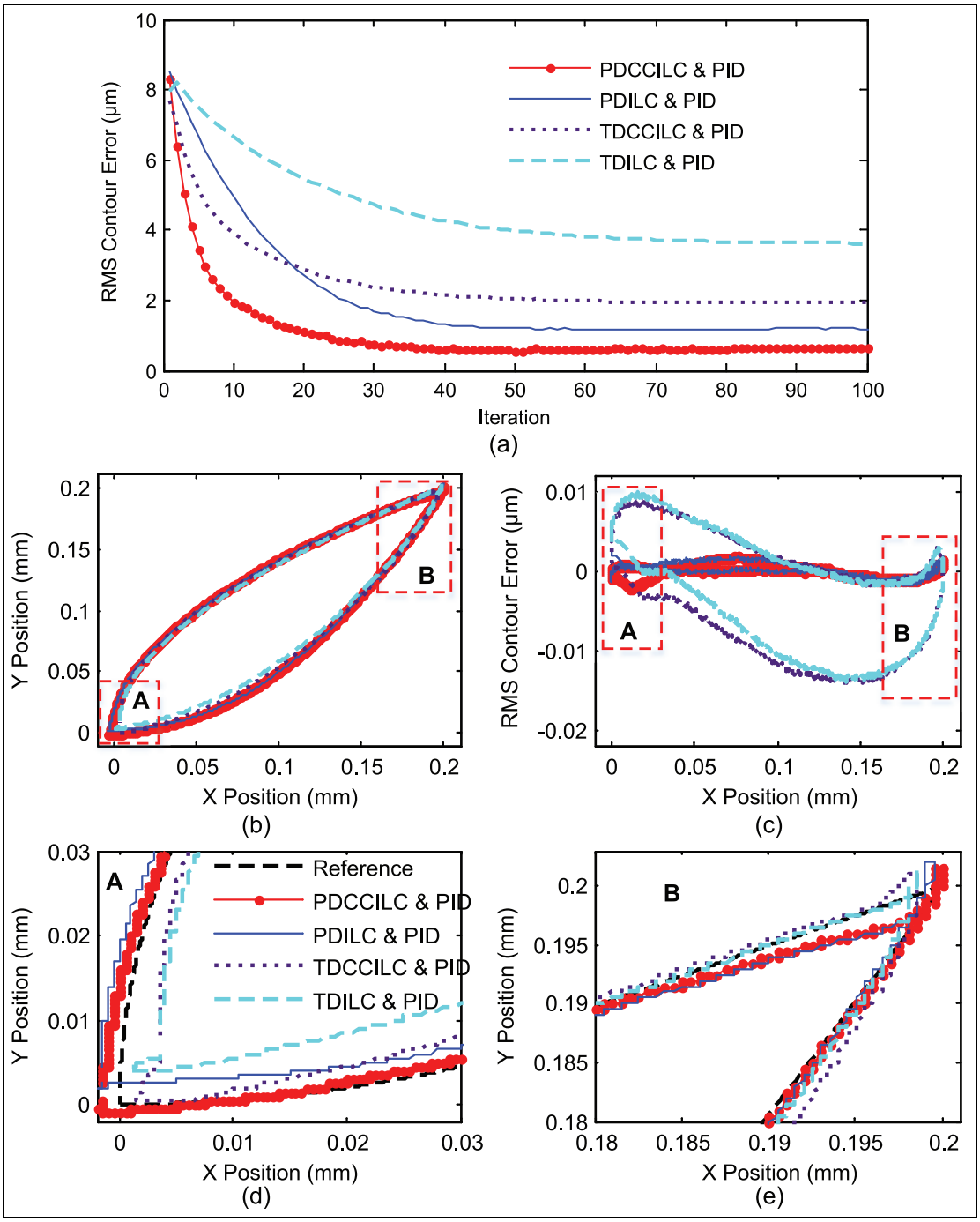

Figure 8. Tracking results of the parabolic contour in the experiments: (a) root mean square (RMS) contour error versus iteration; (b) the overall look in the *xy*-plane; (c) contour errors versus *x*-axis position; (d) and (e) are the partial enlarged view of parts A and B in (b).

Experimental results

All the simulation cases were performed in the experimental set-up as shown in Figure 3 to validate the above-mentioned results. A National Instrument product (CompactRIO 9081) was used for implementation of sensor signal acquisition and control signal output. The specific statistics may be found in Table 3.

Semi-circle contour. The results of semi-circle tracking are shown in Figure 7. Analogously to the simulation results, the combined PDCCILC & PID control system produces the best tracking performance with a 97% decrease in the RMS contour error. Furthermore, the convergence speed of PDCCILC & PID is also the fastest (about 25 iterations). The partial enlarged views of parts A and B are also compatible with those of simulations.

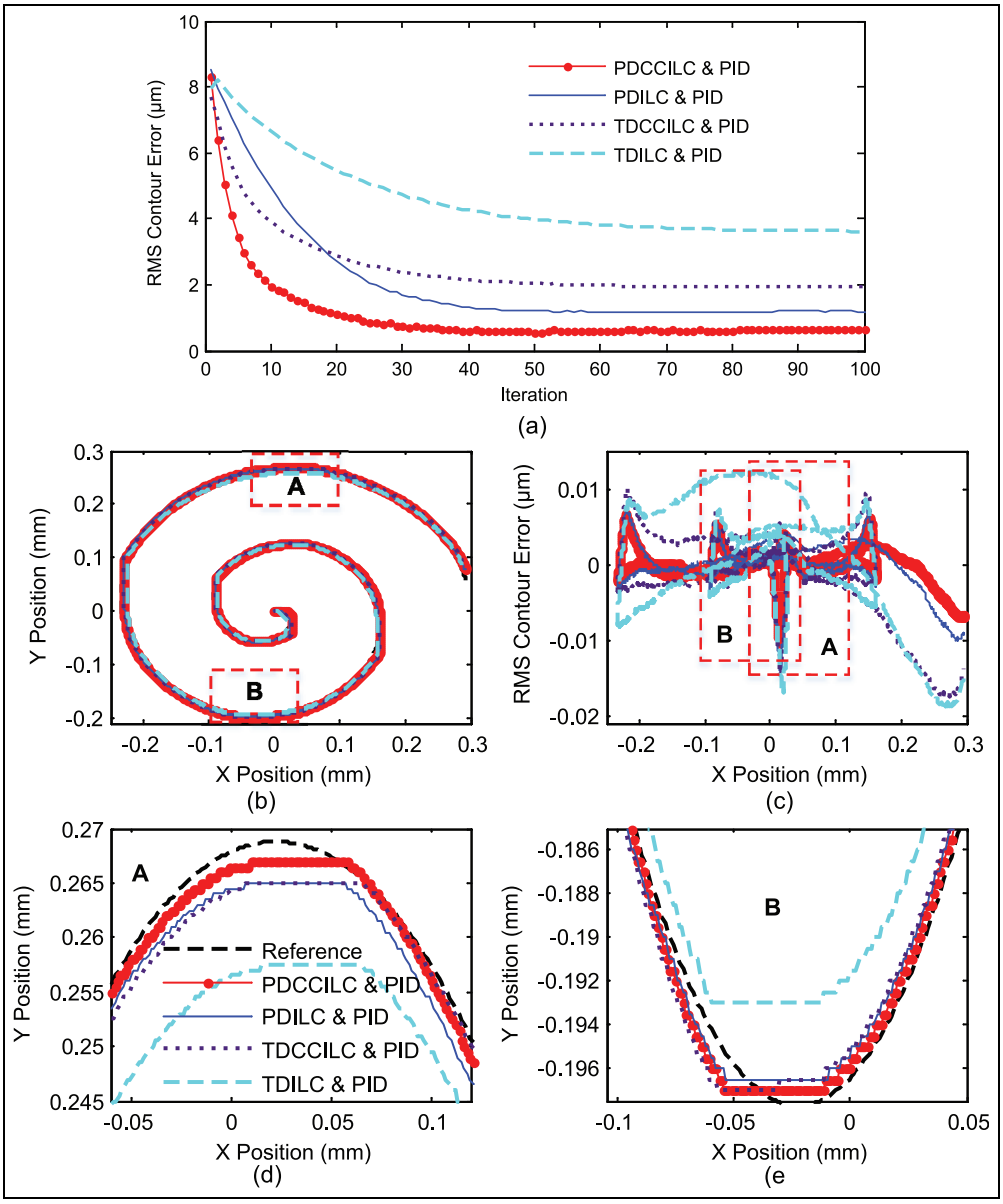

Figure 9. Tracking results of the spiral contour in the experiments: (a) root mean square (RMS) contour error versus iteration; (b) the overall look in the *xy*-plane; (c) contour errors versus *x*-axis position; (d) and (e) are the partial enlarged view of parts A and B in (b).

Table 3. Experimental statistics of tracking performance.

Steady contour error		Semi-circle	Parabola	Spiral
TDILC	RMS (μm)	3.405	3.637	7.566
	MAX (μm)	8.501	8.334	17.429
TDCCILC	RMS (μm)	1.869	1.924	4.897
	MAX (μm)	5.170	6.001	15.420
PDILC	RMS (μm)	1.344	1.197	2.992
	MAX (μm)	4.348	4.696	11.106
PDCCILC	RMS (μm)	0.821	0.631	1.551
	MAX (μm)	2.108	2.907	7.240

TDILC, time domain iterative learning control; RMS, root mean square; TDCCILC, time domain cross-coupled iterative learning control; PDILC, position domain iterative learning control; PDCCILC, position domain cross-coupled iterative learning control.

The increase in experimental RMS error values is to be expected due to model uncertainty in system identification that appears in the actual system and the external disturbance.

Parabolic contour. Figure 8 presents the experimental results of parabolic contour tracking cases. A 93% decrease is achieved by PDCCILC & PID combination and the improvement from TDCCILC to PDCCILC is 16%.

Spiral contour. Figure 9 displays the tracking results of the spiral contour in the experiments. As seen, the PDCCILC & PID combination achieves the smallest steady contour RMS error (about 1.552 μm) and the fastest convergence speed (less than 20 iterations) among the four. The decrease from the initial to the last iteration under PDCCILC & PID is 98%.

Conclusions

This paper has presented the basic framework of PDCCILC & PID design and demonstrated the tracking performance benefits for a combined PDCCILC & PID control system for the three typical non-linear contours. The stability and convergence conditions were discussed in a lifted matrix representation. The best contour tracking performance improvements were obtained from the combined PDCCILC & PID compared with TDCCILC & PID, PDILC & PID and TDILC & PID. These results were supported by experimental data from an actual experimental multi-axis system. Simulation and experiment results of semi-circle, parabola and spiral pattern tracking show that RMS errors under PDCCILC & PID all led to more than a 90% decrease from the initial to the last iteration. The proposed PDCCILC extends the application of CCILC into non-linear contour tracking cases under computation inaccuracy of coupling gains.

Future work involves the adaptive design of the parameters in the combined PDCCILC & PID control approach to obtain superior results in both tracking performance and robustness.

Declaration of conflicting interest

The authors declare that there is no conflict of interest.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported by the National Natural Science Foundation of China (Grant No. 51375349).

References

Ahn HS, Chen Y and Moore KL (2007) Iterative learning control: brief survey and categorization. *IEEE Transactions on Systems Man and Cybernetics Part C Applications and Reviews* 37: 1099.

Ang KH, Chong G and Li Y (2005) PID control system analysis, design, and technology. *IEEE Transactions on Control System Technology* 13: 559–576.

Aphale SS, Bhikkaji B and Moheimani SOR (2008) Minimizing scanning errors in piezoelectric stack-actuated nanopositioning platforms. *IEEE Transactions on Nanotechnology* 7: 79–90.

Barton KL and Alleyne AG (2006) Cross-coupled iterative learning control: design and implementation. In: *The 2007 American Control Conference*, pp. 5496–5502.

Barton KL and Alleyne AG (2008) A cross-coupled iterative learning control design for precision motion control. *IEEE Transactions on Control System Technology* 16: 1218–1231.

Barton KL, Hoelzle DJ, Alleyne AG, et al. (2011) Cross-coupled iterative learning control of systems with dissimilar dynamics: design and implementation. *International Journal of Control* 84: 1223–1233.

Bristow DA, Tharayil M and Alleyne AG (2006) A survey of iterative learning control. *IEEE Control System Magazine* 26: 96–114.

Chen C, Wu J, Lin Y, et al. (2015) Precision sinusoidal local scan for large-range atomic force microscopy with auxiliary optical microscopy. *IEEE/ASME Transactions on Mechatronics* 20: 226–236.

Devasia S, Eleftheriou E and Moheimani SOR (2007) A survey of control issues in nanopositioning. *IEEE Transactions on Control System Technology* 15: 802–823.

Koren Y (1980) Cross-coupled biaxial computer control for manufacturing systems. *Journal of Dynamic Systems, Measurement, and Control* 102: 265–272.

Koren Y and Lo CC (1991) Variable-gain cross-coupling controller for contouring. *CIRP Annals – Manufacturing Systems* 40: 371–374.

Koren Y and Lo CC (1992) Advanced controllers for feed drives. *CIRP Annals – Manufacturing Systems* 41:689–698.

Ling J, Feng Z and Xiao XH (2015) A position domain cross-coupled iteration learning control for contour tracking in multi-axis precision motion control systems. In: *The 2015 International Conference on Intelligent Robotics and Applications*, pp. 667–679.

Ouyang P and Dam T (2011) Position domain PD control: stability and comparison. In: *2011 IEEE International Conference on Information and Automation*, pp. 8–13.

Ouyang P, Dam T, Huang J, et al. (2012) Contour tracking control in position domain. *Mechatronics* 22: 934–944.

Ouyang P, Pano V and Acob J (2013) Position domain contour control for multi-DOF robotic system. *Mechatronics* 23: 1061–1071.

Paul PC, Knoll AW and Holzner F (2011) Rapid turnaround scanning probe nanolithography. *Nanotechnology* 22: 275–306.

Raafat SM and Akmeliawati R (2012) Survey on robust control of precision positioning systems. *Recent Patents on Mechanical Engineering* 5: 55–68.

Shan Y and Leang KK (2012). Accounting for hysteresis in repetitive control design: nanopositioning example. *Automatica* 48: 1751–1758.

Shan Y and Leang KK (2013) Design and control for high-speed nanopositioning: serial-kinematic nanopositioners and repetitive control for nanofabrication. *IEEE Control System* 33: 86–105.

Shen JC, Lu QZ, Wu CH, et al. (2014) Sliding-mode tracking control with DNLRX model-based friction compensation for the precision stage. *IEEE/ASME Transactions on Mechatronics* 19: 788–797.

Tien S, Zou Q and Devasia S (2005) Iterative control of dynamics-coupling caused errors in piezoscanners during high-speed AFM operation. *IEEE Transactions on Control System Technology* 13: 921–931.

Tuma T, Sebastian A and Lygeros J (2013) The four pillars of nanopositioning for scanning probe microscopy: the position sensor, the scanning device, the feedback controller, and the reference trajectory. *IEEE Control System Magazine* 33: 68–85.

Wang Z, Witthauer A, Zou Q, et al. (2015). Control of a magnetostrictive-actuator-based micromachining system for

optimal high-speed microforming process. *IEEE/ASME Transactions on Mechatronics* 20: 1046–1055.

Wei Y, Chen Y, Yang Y, et al. (2016) Novel design and 3-D printing of nonassembly controllable pneumatic robots. *IEEE/ASME Transactions on Mechatronics* 21: 649–659.

Wu Y, Shi J, Su C, et al. (2009) A control approach to cross-coupling compensation of piezotube scanners in tapping-mode atomic force microscope imaging. *Review of Scientific Instruments* 80: 043709.

Yan Y, Wang H and Zou Q (2012) A decoupled inversion-based iterative control approach to multi-axis precision positioning: 3D nanopositioning example. *Automatica* 48: 167–176.

Yeh SS and Hsu PL (2002) Estimation of the contouring error vector for the cross-coupled control design. *IEEE/ASME Transactions on Mechatronics* 7: 44–51.