PRECISION CONTOUR TRACKING USING FEEDBACK-FEEDFORWARD INTEGRATED CONTROL FOR A 2-DOF MANIPULATION SYSTEM

Jie Ling,* Zhao Feng,* Min Ming,* and Xiaohui Xiao*

Abstract

This paper presents a novel approach for precision contour tracking through combining feedback PID and feedforward position domain ILC (PDILC) control for a multi-axis manipulation system. Traditional control approaches in time domain suffer from poor synchronization of relevant motion axes and result in restriction for contour tracking tasks. In the proposed PID & PDILC design, a 2-DOF system is treated as a master–slave cooperative motion system. The position information of the master motion axis is integrated into the PDILC controller of the slave motion axes, which makes the PDILC learn from contour errors instead of individual axis errors. The selection and tuning of parameters for the PID and PDILC were conducted based on the computation in a lifted matrix format of the stability and convergence conditions. The performance of the PID & PDILC controller was evaluated by comparisons with PID and cross-coupled ILC controller through experiments on a multi-axis precise positioning stage. The proposed PID & PDILC design enhances the precision contour tracking of the testbed.

Key Words

Precision motion control, iterative learning control, position domain control, contour tracking

1. Introduction

Contour tracking is one of the most common control problems in multi-axis industrial manipulators and robots. The motion precision depends on both individual axis positioning accuracy and contour tracking error [1]. In many control applications, such as welding, assembling, scanning and manufacturing, contour error, rather than the individual axis positioning error, is the prime concern, although the latter is usually given as the specification of some computerized numerical control systems [2]. In some cases, each motion axis of a multi-input multi-output system is controlled by a separate control loop. A great deal of efforts concerning feedback and feedforward control algorithm, such as PID control [3], robust control [4], iterative learning control [5], predictive control [6] and master–slave control [7], were presented and implemented.

Individual axis control may bring a good tracking performance of each axis, however, it does not guarantee a reduction of contour errors for a multi-axis motion system, as poor synchronization of relevant motion axes may result in a diminished accuracy of the contour tracking. To improve the performance of contoured trajectories and maintain symmetry in the system, the cross-coupled control (CCC) was proposed by Koren in [8]. Although CCC has been applied to multi-axis motions [9], it still has some limitations. One of the difficulties in implementing CCC is determining the cross-coupling gains. Any inaccuracy in the determination of coupling gains is amplified to the control signal and results in a decreased contour performance [10]. Therefore, CCC needs an efficient algorithm to determine the variable gains of arbitrary contours because the gains are dependent on the contour trajectory in real time.

Being different from the CCC technique, Ouyang *et al.* developed a position domain PD type feedback controller for contour tracking in multi-DOF robotic systems [10], [11]. Motivated for contour tracking in repetitive tasks, the objective of this research is to take the advantages of both PDC and ILC to develop a feedforward algorithm called position domain ILC (PDILC). The proposed PDILC in this paper is a novel algorithm that designing feedforward ILC in position domain, instead of traditional time domain. An integrated feedback and feedforward control structure is more effective in precision motion control. Among feedback control algorithms, PID is considerably used in industrial processes, because its structure, consisting of only

* School of Power and Mechanical Engineering, Wuhan University, Wuhan, People's Republic of China; e-mail: {jamesling, fengzhaozhao7, mingmin_whu, xhxiao}@whu.edu.cn
Corresponding author: Xiaohui Xiao
Recommended by Prof. Wail Gueaieb
(DOI: 10.2316/Journal.206.2018.3.206-5140)

three parameters, is very simple to implement and many different techniques are nowadays available for their tuning [12]–[14]. By applying the PID technique as the feedback controller and PDILC as the feedforward controller, control signal convergence and contour errors elimination can be achieved effectively. Previous work in [15] has presented the basic framework of PDILC algorithm with comparisons to conventional time domain ILC. The main contributions in this paper lie in (1) the presentation of the derivation the feedback PID & feedforward PDILC design framework; (2) the stability and convergence analysis of the proposed design framework using the lifted matrix representation method; and (3) experimental validation of the proposed algorithm with comparisons to cross-coupled ILC (CCILC).

The rest of this paper is outlined as follows. Section 2 gives a brief review of ILC, position domain control (PDC) before introducing PDILC. In Section 3, the control law of the PID & PDILC controller is proposed with its stability, convergence and performance analysed in lifted matrix representation. Simulation and experimental results with comparisons between PID & CCILC are presented in Section 4. Conclusion and future works are given in Section 5.

2. Position Domain Iterative Learning Control

For a two input two output, linear time-invariant system, the PDILC control of x- and y-axis can be given as

$$\begin{cases} u^{ilc}_{x_{j+1}}(k) = Q_x(q)[u^{ilc}_x(k) + L_x(q)e_x(k)]_j \\ u^{ilc}_{y_{j+1}}(x) = Q_y(q)[u^{ilc}_y(x) + L_y(q)e_y(x)]_j \end{cases} \tag{1}$$

where k stands for the discretized time index, j is the iteration index, x represents the master position sequence with the function of $x = f(k)$, q is the forward time-shift operator, u^{ilc} is the control signal of ILC controller, Q is a filter, L is the learning function, $e_x(t)$ is the tracking error of x-axis versus time and $e_y(x)$ is the tracking error of y-axis versus x position. Here, the x-axis is chosen as the master motion, while the y-axis is the slave motion. It can be seen that control law for x-axis is totally in time domain, while control law for y-axis is in position domain. Applying the PID-like law of ILC, the x- and y-axis control law can be rewritten as

$$\begin{cases} u^{ilc}_{x_{j+1}}(k) = Q_x(q)\left[u^{ilc}_x(k) + k^{ilc}_{px}e_x(k) + k^{ilc}_{ix}\Gamma(e_x(k)) + k^{ilc}_{dx}\Psi(e_x(k))\right]_j \\ u^{ilc}_{y_{j+1}}(x) = Q_y(q)\left[u^{ilc}_y(x) + k^{ilc}_{py}e_y(x) + k^{ilc}_{iy}\Gamma(e_y(x)) + k^{ilc}_{dy}\Psi(e_y(x))\right]_j \end{cases} \tag{2}$$

where k^{ilc}_{px}, k^{ilc}_{ix}, k^{ilc}_{dx}, k^{ilc}_{py}, k^{ilc}_{iy} and k^{ilc}_{dy} are PID gains for x- and y-axis ILC controller, $\Gamma(\cdot)$ is the integral operator and $\Psi(\cdot)$ is the differential operator. The expansion formulas of the integral and differential operators are

$$\begin{cases} \Gamma(e_x(k)) = \dfrac{\Delta k}{2}(e_x(k) + e_x(k - \Delta k)) \\ \Psi(e_x(k)) = \dfrac{e_x(k) - e_x(k - \Delta k)}{\Delta k} \end{cases} \tag{3}$$

$$\begin{cases} \Gamma(e_y(x)) = \dfrac{\Delta x(k)}{2}(e_y(k) + e_y(k - \Delta k)) \\ \Psi(e_y(x)) = \dfrac{e_y(k) - e_y(k - \Delta k)}{\Delta x(k)} \end{cases} \tag{4}$$

Substituting the integral and differential operator in (3) and (4) into (2), we get (5) and (6) as

$$\begin{cases} u^{ilc}_{x_{j+1}}(k) = Q_x\left[u^{ilc}_x(k) + \alpha_x \cdot e_x(k) + \beta_x \cdot e_x(k - \Delta k)\right]_j \\ \alpha_x = k^{ilc}_{px} + \dfrac{\Delta k}{2}k^{ilc}_{ix} + \dfrac{k^{ilc}_{dx}}{\Delta k} \\ \beta_x = \dfrac{\Delta k}{2}k^{ilc}_{ix} - \dfrac{k^{ilc}_{dx}}{\Delta k} \end{cases} \tag{5}$$

$$\begin{cases} u^{ilc}_{y_{j+1}}(k) = Q_y\left[u^{ilc}_y(k) + \alpha_y(k)e_y(k) + \beta_y(k)e_y(k - \Delta k)\right]_j \\ \alpha_y(k) = k^{ilc}_{py} + \dfrac{\Delta x(k)}{2}k^{ilc}_{iy} + \dfrac{k^{ilc}_{dy}}{\Delta x(k)} \\ \beta_y(k) = \dfrac{\Delta x(k)}{2}k^{ilc}_{iy} - \dfrac{k^{ilc}_{dy}}{\Delta x(k)} \end{cases} \tag{6}$$

It can be seen from (5) that α_x and β_x are constants when the time delay index Δk is determined, while in (6), α_y and β_y are time variants because the position interval Δx of x-axis in time delay Δk is changing. That is to say that the control signal of y-axis is relevant with position interval of x-axis. The master axis and slave axis are coupled through position information of x-axis to guarantee the synchronization of axes. This is different from conventional time domain ILC design.

3. Combined Feedback PID and Feedforward PDILC Controller Design

3.1 Lifted Matrix Representation

The lifting operation over a finite interval allows us to represent the system using a matrix format for the time domain dynamics [16], and it can be used to analyse the system stability and convergence in the iteration domain for the ILC law. Details about this method can refer to [15], [16].

3.2 Combined PID & PDILC Controller Design

The PID & PDILC control structure for a two DOF system is shown in Fig. 1. The x-axis is chosen as the master motion, and the y-axis is the slave motion. TDILC is applied for master motion and PDILC is used for slave motion to guarantee the synchronization between axes.

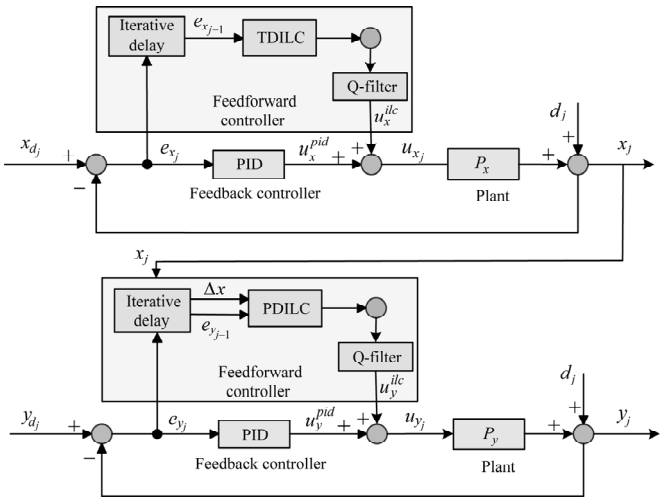

Figure 1. The PID & PDILC control structure.

In the y-axis ILC design, the position information of x-axis is integrated into the controller as described in (6), while a time domain ILC is applied into x-axis control. To discuss the convergence and error performance in (5) and (6), the error items of $e(k)$ and $e(k-\Delta k)$ need to be substituted by polynomial of u_x^{ilc}, u_y^{ilc}, x_d and y_d. The lifted matrix format of errors are

$$\begin{cases} \boldsymbol{e}_x = \boldsymbol{x}_d - \boldsymbol{x} = \boldsymbol{x}_d - \boldsymbol{P}_{\boldsymbol{x}} \cdot (\boldsymbol{u}_x^{ilc} + \boldsymbol{u}_x^{pid}) \\ \hat{\boldsymbol{e}}_x = \hat{\boldsymbol{x}}_d - \hat{\boldsymbol{x}} = \hat{\boldsymbol{x}}_d - \hat{\boldsymbol{P}}_{\boldsymbol{x}} \cdot (\boldsymbol{u}_x^{ilc} + \boldsymbol{u}_x^{pid}) \end{cases} \tag{7}$$

where $\boldsymbol{u}_x^{pid} = \boldsymbol{K}_x \cdot \boldsymbol{e}_x$ is the control output of the PID controller for x-axis with the non-lifted format of the PID controller as $K_x = k_{px}^{pid} + k_{ix}^{pid} \cdot s^{-1} + k_{dx}^{pid} \cdot s$ (s is the frequency domain variable), $\hat{e}_x$, $\hat{x}_d$ and $\hat{P}_x$ represent the signal or system transfer function with Δk steps time delay (*i.e.* $\hat{e}_x(k) = e_x(k) - e_x(k-\Delta k)$), and

$$\begin{cases} \boldsymbol{e}_y = \boldsymbol{y}_d - \boldsymbol{y} = \boldsymbol{y}_d - \boldsymbol{P}_y \cdot (\boldsymbol{u}_y^{ilc} + \boldsymbol{u}_y^{pid}) \\ \hat{\boldsymbol{e}}_y = \hat{\boldsymbol{y}}_d - \hat{\boldsymbol{y}} = \hat{\boldsymbol{y}}_d - \hat{\boldsymbol{P}}_y \cdot (\boldsymbol{u}_y^{ilc} + \boldsymbol{u}_y^{pid}) \end{cases} \tag{8}$$

where $\boldsymbol{u}_y^{pid} = \boldsymbol{K}_y \cdot \boldsymbol{e}_y$, and the non-lifted format of the y-axis PID controller is $K_y = k_{py}^{pid} + k_{iy}^{pid} \cdot s^{-1} + k_{dy}^{pid} \cdot s$.

Solving (7) and (8) generates (9) and (10) as

$$\begin{cases} \boldsymbol{e}_x = \boldsymbol{S}_x \cdot \boldsymbol{x}_d - \boldsymbol{S}_x \boldsymbol{P}_x \cdot \boldsymbol{u}_x^{ilc} \\ \hat{\boldsymbol{e}}_x = -\boldsymbol{K}_x \boldsymbol{S}_x \hat{\boldsymbol{P}}_x \cdot \boldsymbol{x}_d - \boldsymbol{S}_x \hat{\boldsymbol{P}}_x \boldsymbol{u}_x^{ilc} + \hat{\boldsymbol{x}}_d \end{cases} \tag{9}$$

$$\begin{cases} \boldsymbol{e}_y = \boldsymbol{S}_y \cdot \boldsymbol{y}_d - \boldsymbol{S}_y \boldsymbol{P}_y \cdot \boldsymbol{u}_y^{ilc} \\ \hat{\boldsymbol{e}}_y = -\boldsymbol{K}_y \boldsymbol{S}_y \hat{\boldsymbol{P}}_y \cdot \boldsymbol{y}_d - \boldsymbol{S}_y \hat{\boldsymbol{P}}_y \cdot \boldsymbol{u}_y^{ilc} + \hat{\boldsymbol{y}}_d \end{cases} \tag{10}$$

where $\boldsymbol{S}$ is the lifted format of the sensitivity transfer function of the axis system dynamic with $S = (1 + KP)^{-1}$.

Substituting (9) and (10) into (5) and (6) yields the recursion formula of ILC control signal for x- and y-axis as

$$\begin{cases} \boldsymbol{u}_{x_{j+1}}^{ilc} = \boldsymbol{M}_x \cdot \boldsymbol{u}_{x_j}^{ilc} + \boldsymbol{N}_x \\ \boldsymbol{u}_{y_{j+1}}^{ilc} = \boldsymbol{M}_y \cdot \boldsymbol{u}_{y_j}^{ilc} + \boldsymbol{N}_y \end{cases} \tag{11}$$

where

$$\begin{cases} \boldsymbol{M}_x = \boldsymbol{Q}_x (\boldsymbol{I} - \boldsymbol{\alpha}_x \boldsymbol{S}_x \boldsymbol{P}_x - \boldsymbol{\beta}_x \boldsymbol{S}_x \hat{\boldsymbol{P}}_x) \\ \boldsymbol{N}_x = \boldsymbol{Q}_x \left[(\boldsymbol{\alpha}_x \boldsymbol{S}_x - \boldsymbol{\beta}_x \boldsymbol{K}_x \boldsymbol{S}_x \hat{\boldsymbol{P}}_x) \cdot \boldsymbol{x}_d + \boldsymbol{\beta}_x \hat{\boldsymbol{x}}_d \right] \\ \boldsymbol{M}_y = \boldsymbol{Q}_y (\boldsymbol{I} - \boldsymbol{\alpha}_y \boldsymbol{S}_y \boldsymbol{P}_y - \boldsymbol{\beta}_y \boldsymbol{S}_y \hat{\boldsymbol{P}}_y) \\ \boldsymbol{N}_y = \boldsymbol{Q}_y \left[(\boldsymbol{\alpha}_y \boldsymbol{S}_y - \boldsymbol{\beta}_y \boldsymbol{K}_y \boldsymbol{S}_y \hat{\boldsymbol{P}}_y) \boldsymbol{y}_d + \boldsymbol{\beta}_y \hat{\boldsymbol{y}}_d \right] \end{cases} \tag{12}$$

In (12), $\boldsymbol{M}$, $\boldsymbol{Q}$, $\boldsymbol{S}$ and $\boldsymbol{P}$ are $N \times N$ square matrixes, $\boldsymbol{I}$ is the N-dimensional unit matrix, $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are N-dimensional diagonal matrixes, and $\boldsymbol{N}$, $\boldsymbol{x}_d$, $\hat{\boldsymbol{x}}_d$, $\boldsymbol{y}_d$ and $\hat{\boldsymbol{y}}_d$ are $N \times 1$ matrixes as displayed in the uniformed equation as

$$\underbrace{\begin{bmatrix} u^{ilc}(1) \\ u^{ilc}(2) \\ \vdots \\ u^{ilc}(N) \end{bmatrix}_{j+1}}_{\mathrm{U_i}} = \underbrace{\begin{bmatrix} m_1 & 0 & \cdots & 0 \\ m_2 & m_1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ m_N & m_{N-1} & \cdots & m_1 \end{bmatrix}}_{\boldsymbol{M}_i} \underbrace{\begin{bmatrix} u^{ilc}(0) \\ u^{ilc}(1) \\ \vdots \\ u^{ilc}(N-1) \end{bmatrix}_{j}}_{\mathrm{U_i}} + \underbrace{\begin{bmatrix} n(1) \\ n(2) \\ \vdots \\ n(N) \end{bmatrix}}_{\boldsymbol{N}_i} \tag{13}$$

where $i \in \{x, y\}$, $\boldsymbol{M}_i$ and $\boldsymbol{N}_i$ are computed by (14) as

$$\begin{cases} \boldsymbol{M}_i = \boldsymbol{Q}_i \cdot (\boldsymbol{I} - \boldsymbol{\alpha}_i \boldsymbol{S}_i \boldsymbol{P}_i - \boldsymbol{\beta}_i \boldsymbol{S}_i \hat{\boldsymbol{P}}_i) \\ \boldsymbol{N}_i = \boldsymbol{Q}_i \cdot \left[(\boldsymbol{\alpha}_i \boldsymbol{S}_i - \boldsymbol{\beta}_i \boldsymbol{K}_i \boldsymbol{S}_i \hat{\boldsymbol{P}}_i) \boldsymbol{i}_d + \boldsymbol{\beta}_i \hat{\boldsymbol{i}}_d \right] \end{cases} \tag{14}$$

3.3 Monotonic Convergence Condition

As stated in [17]–[19], a sufficient condition for monotonic convergence of the composite iterative algorithm can be given by

$$\bar{\sigma}(\boldsymbol{M}) < 1 \tag{15}$$

where $\bar{\sigma}(\cdot)$ denotes the upper bound value of the induced 2-norm of a matrix, $\boldsymbol{M} = \begin{bmatrix} \boldsymbol{M}_x \\ \boldsymbol{M}_y \end{bmatrix}$ is the combined matrix of the two axes.

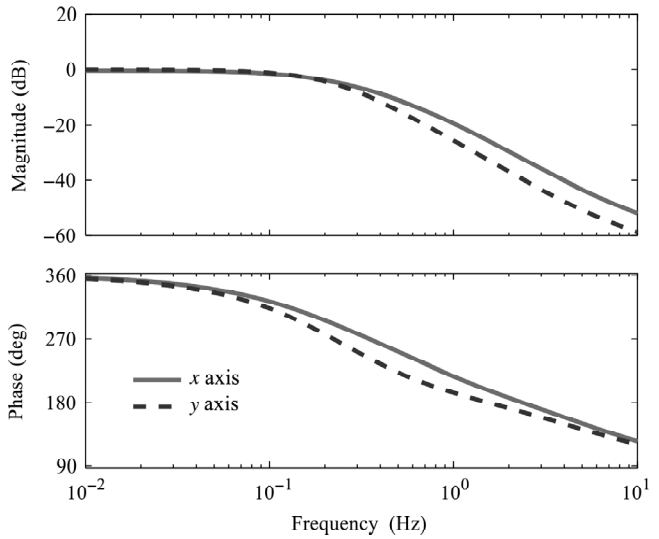

Figure 2. Bode plot of x- and y-axis.

4. Evaluation: Convergence Computation and Experiments

In this section, the PID & PDILC design was evaluated through both simulation and experiment cases based on a three DOF precise positioning stage with a stroke of 50, 50 and 20 mm for x, y and z-axis, respectively. The tested platform shown in Fig. 8 is a serial system with its each axis driven by a DC servo motor. The position information of the axis is detected by an incremental encoder with a resolution of 0.5 μm which provides baseline information for understanding the behaviour of general multi-axis systems. 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 PID & PDILC design is to improve the contour error in this testbed in the following sections.

4.1 System Identification

In this work, only x- and y-axis were selected for simulation and experiment case. Dynamic models of the x- and y-axis were achieved through step response method with a sample rate of 1 kHz. The continuous transfer functions are

$$\begin{cases} P_x = \dfrac{-0.1402s + 5.291}{s^2 + 5.795s + 5.564} \\ P_y = \dfrac{-0.0631s + 2.132}{s^2 + 2.76s + 2.127} \end{cases} \tag{16}$$

As the bode diagram shown in Fig. 2, the bandwidth of x- and y-axis are 0.164 and 0.162 Hz.

4.2 Lifted Matrix Computation

A sector trajectory and a parabolic trajectory were used to test the effectiveness of the proposed algorithm. The contours in Fig. 3 lasted 14 s and the step size in simulation

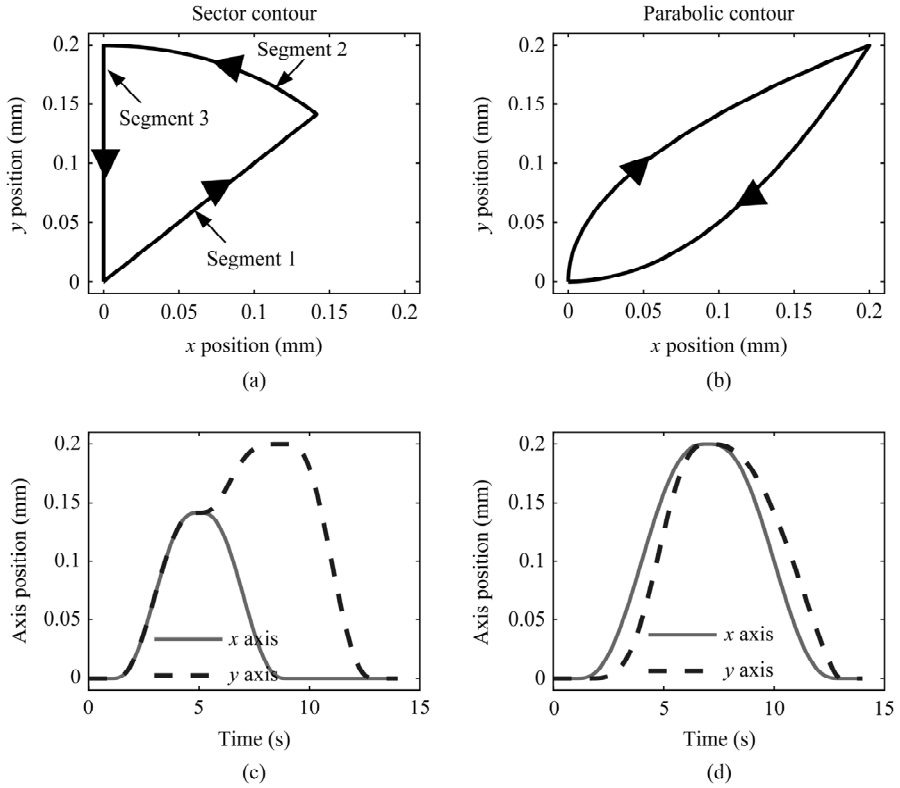

Figure 3. Reference contours: (a) sector motion in the xy plane; (b) parabolic motion in the xy-plane; and (c) and (d) are individual axis motion versus time.

Table 1
Controller Parameters

Contour Case	PDILC			CCILC				
				ILC			CCC	
	P	I	D	P	I	D	P	D
Sector	0.2	0.1	0	0.2	0.1	0	1	0.5
Parabolic	0.3	0.1	0	0.3	0.1	0	1	0.5

Table 2
Computational Results of Monotonic Convergence for the Reference Contours

Contour Case	Computational Results of $\bar{\sigma}(\boldsymbol{M})$	
	PDILC	CCILC
Sector	0.9913	0.9663
Parabolic	0.9896	0.9622

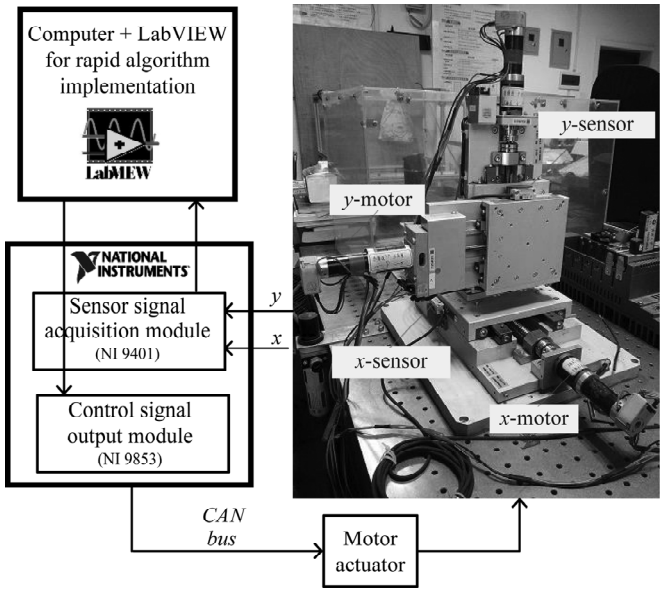

Figure 4. Closed loop control system in the experiment.

was set 0.005 s, so the dimension of the AS condition matrix in (15) was 2,800 × 2,800. This made it compatible within the memory and calculation capability in a normal computer. If the matrix size is too big, some indirect computation methods can be applied as described in [17].

In the control law, a Q-filter is used. In this case, it was set as a third-order low-pass filter to ensure robustness in the presence of noise and uncertainty. From Fig. 2, it can be found that the bandwidth of the x-axis is very close to that of the y-axis. The Q-filter for the two axes is set the same with its continuous transfer function of Q is shown as

$$Q = \frac{0.7294s^3 - 167s^2 - 5.213 \times 10^4 s}{s^3 + 1262s^2 + 7.961 \times 10^5 s + 2.543 \times 10^8} \qquad (17)$$

For comparison with the proposed PID & PDILC controller, the PID & CCILC controller is used in the following computation, simulation and experiment sections. The PID gains for feedback controller are determined by tuning [20] with the designed results as 3, 0.8 and 0.05, respectively. The parameters for feedforward PDILC and CCILC in the two sets are displayed in Table 1.

The convergence results under the two sets of controllers (PID & PDILC, PID & CCILC) are shown in Table 2, which illustrates that the upper bound condition of (15) is valid the monotonic convergence condition is satisfied by appropriate design of the different learning functions and PID gains.

4.3 Experiments

As shown in Fig. 4, the tested platform is a three DOF precise positioning stage with a stroke of 50, 50 and 20 mm for x-, y- and z-axis, 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 PID & PDILC design is to improve the contour error in this testbed in the following sections. The product of National Instrument (Compact-RIO 9081) was used for implementation of sensor signal acquisition and control signal output based on rapid control prototype.

4.3.1 Sector Case

The experimental results for sector tracking are shown in Figs. 5 and 6. The PID & PDILC control produces the better improved tracking performance than PID & CCILC with a 95% decrease in the contour RMS error. The improvement includes a 5% reduction in RMS contour error from CCC design to PDC design. The increase in experimental RMS error values is to be expected due to model uncertainty in system identification that appears in the actual system.

Figure 6 displays the tracking performance in the last iteration for sector experiment. The contour under PID & PDILC is closer to the reference compared with PID & CCILC.

4.3.2 Parabolic Case

Figures 7 and 8 are the experimental results for the parabolic tracking under a PID & PDILC system versus

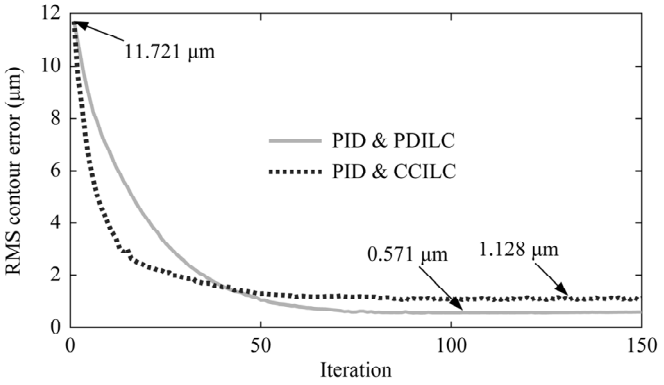

Figure 5. RMS contour error of sector case versus iteration in the experiments.

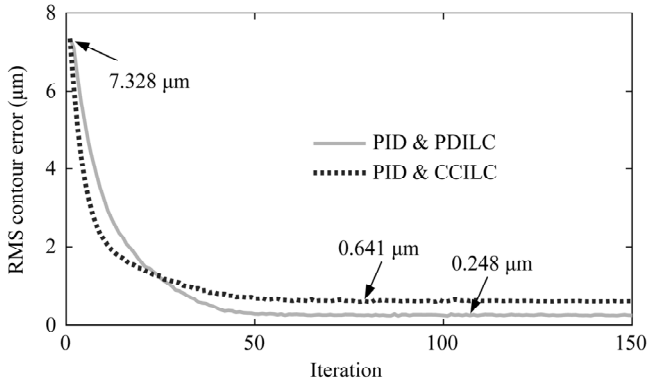

Figure 7. RMS contour error of parabolic case versus iteration in the experiments.

PID & CCILC system. The PID & PDILC controller produces the better improved tracking performance than PID & CCILC with a 97% decrease in the contour RMS error as shown in Fig. 7. The improvement includes a 5% reduction in RMS contour error from CCC design to PDC design.

Move to a closer look into Fig. 8, it can be seen that the contour under PID & PDILC are closer to the reference compared with PID & CCILC, which is consistent to the aforementioned results in Fig. 8.

The above experimental results show that both the PID & PDILC and PID & CCILC can achieve obvious improvements for contour tracking performance. However, to be more precise, the PID & PDILC is better in decreasing steady converged errors. What's more, the PDILC algorithm does not rely on the computation of cross-coupling gains, which is the main advantage over the CCILC.

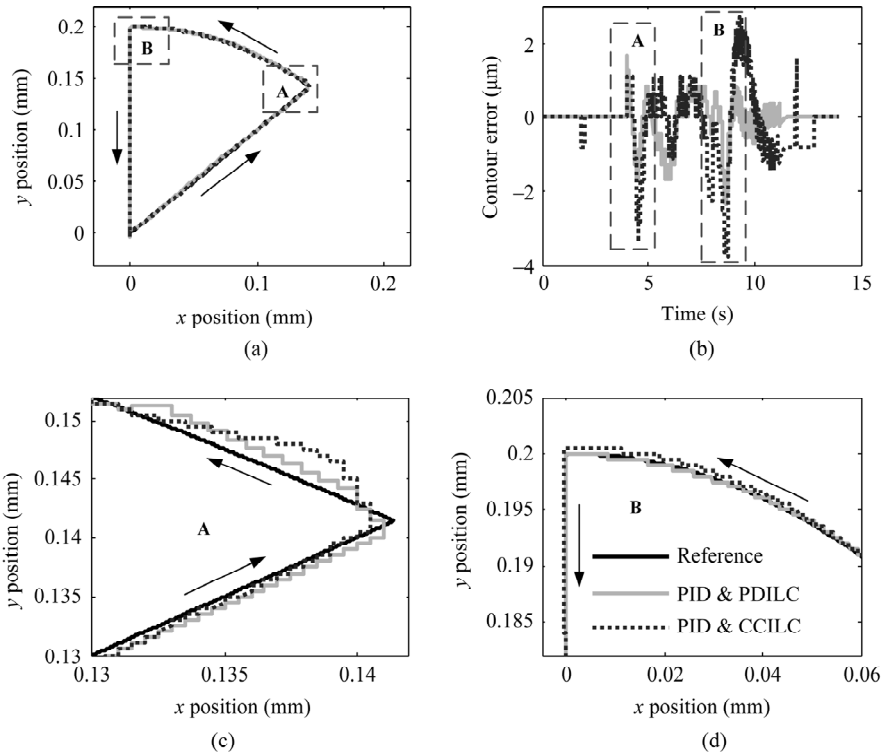

Figure 6. Tracking results of the sector contour in experiments: (a) the overall look in the xy-plane; (b) contour errors versus time; (c) and (d) are the partial enlarged view of part A and B marked in subplot (a) and (b).

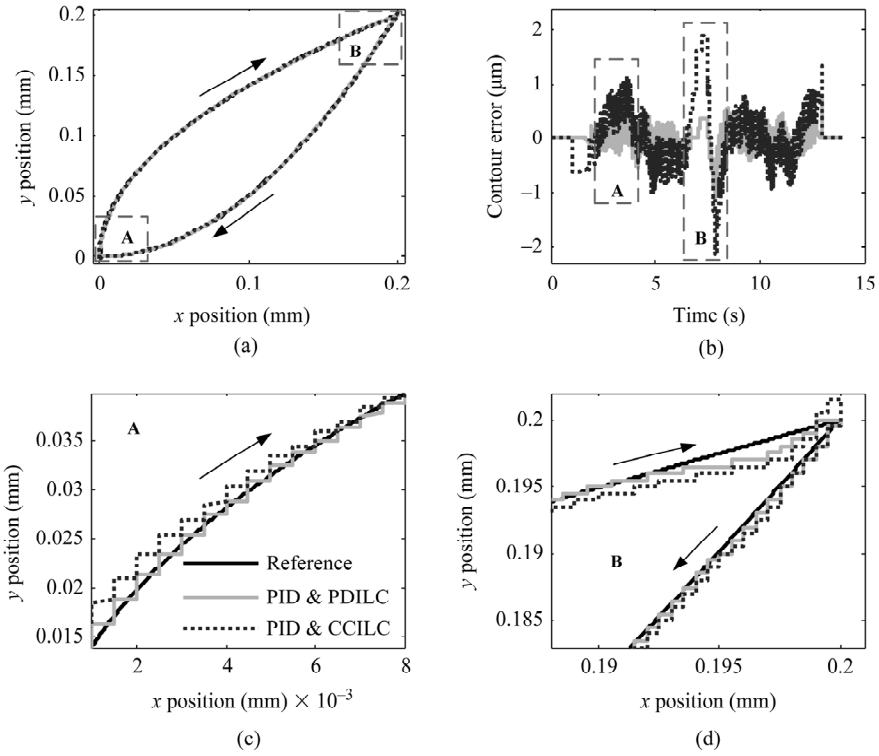

Figure 8. Tracking results of the parabolic contour in experiments: (a) the overall look in the xy-plane; (b) contour errors; and (c) and (d) the partial enlarged view of part A and B marked in subplot (a) and (b).

5. Conclusion

This paper has presented the framework of the composite PID & PDILC and demonstrated the contour tracking performance befits for such a controller. The stability and convergence conditions were discussed in lifted matrix representation.

1. Smaller contour tracking errors can be obtained from the PID & PDILC controller than the PID & CCILC.
2. The proposed PID & PDILC controller relies less on the computation of cross-coupling gains compared with traditional CCC or CCILC. The position information of the master motion is integrated into the slave motion control to guarantee the synchronization between the axes.
3. The PID & PDILC control has broad application prospects. As the control strategy is originated from precision motion control for a multi-axis stage in this paper, it can be easily integrated into multi-DOFs robotic manipulators, multi-axle vehicles and so on, where the synchronization between axes should be satisfied.

In this paper, the PID & PDILC controller was illustrated on a 2-DOF motion system; however, this technique could be extended to three and more axes systems in our future work. Also, the robustness of the algorithm needs to be investigated in the future.

Acknowledgement

This work was supported by the Natural Science Foundation of China under Grant 51375349 and Shenzhen Science and Technology Program under Grant JCYJ20170306171514468.

References

[1] P.R. Ouyang, T. Dam, J. Huang, and W. J. Zhang, Contour tracking control in position domain, *Mechatronics*, *22*(7), 2012, 934–944.

[2] Y. Koren and C.C. Lo, Variable-gain cross-coupling controller for contouring, *CIRP Annals – Manufacturing Technology*, *40*(1),1991, 371–374.

[3] R.A. Hooshmand, A. Akbar Nasiri, and M. Ataei, Trajectory angle control of fish-like robot motion by using fuzzy-PID controller, *International Journal of Robotics and Automation*, *27*(2), 2012, 163.

[4] Q. Guo, C.J. Macnab, and J.K. Pieper, Robust control of a rigid articulated hopper, *International Journal of Robotics and Automation*, *27*(1), 2012, 1.

[5] J. Ling, Z. Feng, D.J. Yao, and X.H. Xiao, Non-linear contour tracking using feedback PID and feedforward position domain cross-coupled iterative learning control, *Transactions of the Institute of Measurement and Control*, *40*(6), 2018, 1970–1982.

[6] Y. Wang, Y. Tuo, S.X. Yang, and M. Fu, Nonlinear model predictive control of dynamic positioning of deep-sea ships with a unified model, *International Journal of Robotics and Automation*, *31*(6), 2016, 519–529.

[7] A. Rodriguez-Angeles, M.A. Arteaga-Perez, R.d.J. Portillo-Velez, and C.A. Cruz-Villar, Transparent bilateral master–slave control based on virtual surfaces: stability analysis and experimental results, *International Journal of Robotics and Automation*, *30*(2), 2015.

[8] Y. Koren, Cross-coupled biaxial computer controls for manufacturing systems, *Journal of Dynamic Systems Measurement & Control*, *102*(4), 1980, 265–272.

[9] L. Ren, J.K. Mills, and D. Sun, Experimental comparison of control approaches on trajectory tracking control of a 3-DOF parallel robot, *IEEE Transactions on Control Systems Technology*, *15*(5), 2007, 982–988.

[10] P.R. Ouyang, V. Pano, and J. Acob, Position domain contour control for multi-DOF robotic system, *Mechatronics*, *23*(8), 2013, 1061–1071.
[11] P.R. Ouyang and T.D.V. Pano, Cross-coupled PID control in position domain for contour tracking, *Robotica*, *33*(6), 2014, 1–24.
[12] G.P. Liu and S. Daley, Optimal-tuning PID control for industrial systems, *Control Engineering Practice*, *9*(11), 2001, 1185–1194.
[13] K.H. Ang, G. Chong, and Y. Li, PID control system analysis, design, and technology, *IEEE Transactions on Control Systems Technology*, *13*(4), 2005, 559–576.
[14] P. Nordfeldt and T. Hägglund, Decoupler and PID controller design of TITO systems, *Journal of Process Control*, *16*(9), 2006, 923–936.
[15] J. Ling, Z. Feng, D.J. Yao, and X.H. Xiao, A position domain iteration learning control for contour tracking with application to a multi-axis motion testbed, *American Control Conference (ACC), IEEE*, Boston, United states, 2016, 1247–1252.
[16] D.A. Bristow, M. Tharayil, and A.G. Alleyne, A survey of iterative learning control, *IEEE Control Systems*, *26*(3), 2006, 96–114.
[17] K.L. Barton and A.G. Alleyne, A cross-coupled iterative learning control design for precision motion control, *IEEE Transactions on Control Systems Technology*, *16*(6), 2008, 1218–1231.
[18] K.L. Barton, D.J. Hoelzle, A.G. Alleyne, and A.J.W. Johnson, Cross-coupled iterative learning control of systems with dissimilar dynamics: Design and implementation, *International Journal of Control*, *84*(7), 2011, 1223–1233.
[19] K. Barton, d.W. Van, J.A. Alleyne, and O. Bosgra, Norm optimal cross-coupled iterative learning control, *47th IEEE Conf. on Decision and Control*, Cancun, Mexico, 2008, 3020–3025.
[20] K.H. Ang, G. Chong, and Y. Li, PID control system analysis, design, and technology, *IEEE Transactions on Control Systems Technology*, *13*(4), 2005, 559–576.

Biographies

Jie Ling received the B.S. degree in Mechanical Engineering from the School of Power and Mechanical Engineering, Wuhan University, Wuhan, China, in 2012. He is currently pursuing the Ph.D. degree in Mechatronic Engineering at Wuhan University, Wuhan, China. His research interests include precise motion control, iterative learning control applications as well as design and control of nano-positioning systems.

Zhao Feng received the B.S. degree in Mechanical Engineering from the School of Power and Mechanical Engineering, Wuhan University, Wuhan, China, in 2014. He is currently pursuing the Ph.D. degree in Mechatronic Engineering at Wuhan University, Wuhan, China. His research interests include vibration damping control, iterative learning control, nano-positioning and robotics.

Min Ming received the B.S. degree in Mechanical Engineering from the School of Power and Mechanical Engineering, Wuhan University, Wuhan, China, in 2016. She is currently a Ph.D. student at the Department of Mechatronic Engineering, Wuhan University. Her research interests include hysteresis control, iterative learning control and nano-positioner.

Xiaohui Xiao received the B.S. and M.S. degrees in Mechanical Engineering from Wuhan University, Wuhan, China, in 1991 and 1998, respectively, and the Ph.D. degree in mechanical engineering from Huazhong University of Science and Technology, Wuhan, China, in 2005. She joined the Wuhan University, Wuhan, China, in 1998, where she is currently a Full Professor with the Mechanical Engineering Department, School of Power and Mechanical Engineering. Her current research interests include mobile robotics, high-precision positioning control and signal processing.