2016 American Control Conference (ACC)
Boston Marriott Copley Place
July 6-8, 2016. Boston, MA, USA

A Position Domain Iteration Learning Control for Contour Tracking with Application to a Multi-axis Motion Testbed*

Jie Ling, Zhao Feng, Daojin Yao, *Student Member, IEEE* and Xiaohui Xiao, *Member, IEEE*

***Abstract*—To improve contour tracking performance for multi-axis precision motion systems, a position domain iteration learning control (PDILC) is presented in this paper. Traditional control approaches design controllers in time domain individually, thus suffer from poor synchronization of relevant motion axes and result in restriction for contour tracking tasks. Our approach is to combine the positon domain method with the ILC to reduce the contour errors and individual axis simultaneously for repetitive tasks. Stability and convergence analysis of the proposed method are conducted through lifted system representation method. Performance of the PDILC, traditional time domain ILC (TDILC) and feedback control lonely are evaluated and compared through numerical simulations and experimental testing based on a multi-axis precise positioning stage. The proposed method enhances the precision contour tracking performance of the testbed.**

I. Introduction

In multi-axis motion systems, motion precision depends on both individual axis tracking and contouring accuracy. Traditionally, a decoupled multi-input multi-output (MIMO) system is handled as multi single-input single-output (SISO) systems. The tracking performance is improved by applying feedback and feedforward control into each axis individually. A great deal of efforts, such as PID control [1], robust control [2], sliding-mode control [3], iterative learning control [4], repetitive control [5], [6], and polynomial-based pole placement control [7], have been investigated.

However, a good tracking performance for each individual axis does not guarantee the reduction of contour errors for a multi-axis motion system, as poor synchronization of relevant motion axes may result in diminished accuracy of the contour tracking performance [8]. To decrease contour errors, cross-coupling control (CCC) was developed by Koren [9]. The CCC utilizes coupling gains to couple the individual axis errors of SISO systems together and applies a controller to the combined signal. The variable-gain CCC was proposed by Koren and Lo for nonlinear contour tracking by applying circular contour approximation for arbitrary contour applications [10], [11]. Yeh and Hsu developed a modified variable-gain CCC based on the contour error vector in [12]. It should be mentioned that, CCC leaves the tracking performance unchanged while improving contour error [13]. Barton combined cross-coupled iterative learning control (CCILC) with individual axis ILC to improve both individual axis and contour tracking performance [14]-[16]. Being different from the aforementioned controllers designed in time domain, Ouyang proposed a novel PID feedback controller based on the position domain (PDPID) which perceives motion system as a master-slave cooperative system to guarantee synchronization and improves the contour tracking performance [17]. One unique feature of PDPID is that there is no tracking error for the master motion, and only tracking errors of the slave motions affect the final contour tracking errors [18]. The method has been applied on computerized numerical control (CNC) and robotic system.

The main motivation of this paper is to provide a novel method to improve contour tracking performance for a multi-axis motion system that executes a same task repetitively. The CCILC is an effective control method for such a system as mentioned before. But the main disadvantage of CCILC is the complexity of computation. For example, two individual ILC controllers and a CCC controller are needed for a biaxial system. The coupling gains in traditional CCILC for contours of nonlinear line nor non-circle (such as parabolic) are complex to be obtained, and the inaccuracy of coupling gain computation may influence the tracking performance of CCILC. Our method is designing ILC in position domain and generating a position domain iteration learning controller (PDILC). The proposed PDILC relies less on accurate coupling gains, so an estimation vector method can be applied in PDILC design and then the PDILC computation process can be simplified. The proposed PDILC is advantageous on maintaining multi-axis synchronization with reducing individual axis and contour errors simultaneously.

The outline of this paper is as follows. Section II gives a brief review of ILC, position domain control (PDC) and contour error estimation. In section III, PDILC control law is proposed and analyzed. Simulation and experimental results and comparison between existing time domain ILC are presented in Section IV. Conclusions are given in Section V.

II. Control Design Background

A. Iteration Learning Control

ILC is firstly proposed by Uchiyama in 1978 [19] and widely discussed in [20] and so on. It is based on the notion that tracking error of a system that executes the same task multiple times can be decreased by learning from previous iterations [21]. The error information is integrated into the

*Resrach supported by National Natural Science Foundation of China (NSFC, Grant No. 51375349 and 51175383).

Jie Ling, Zhao Feng, Daojin Yao and Xiaohui Xiao are with the School of Power and Mechanical Engineering at Wuhan University, Wuhan, Hubei Province, P.R. China (e-mail: respectively, jamesling@whu.edu.cn, fengzhaozhao7@whu.edu.cn, ydaojin@whu.edu.cn, xhxiao@whu.edu.cn).

controller and high performance can be achieved despite large model uncertainty and repeating external disturbances. Considering a discrete LTI and SISO system

$$y_j(k) = P(q)u_j(k) + d(k) \tag{1}$$

where k stands for the time index, j is the iteration index, y_j is the output, y_j is the control signal, d is the exogenous signal, $P(q)$ is the system transfer function with a time delay and q is the forward time-shift operator $qx(k) \equiv x(k+1)$. A widely used control law of ILC is shown in (2) and the ILC system is asymptotically stable (AS) if condition (4) can be satisfied [21].

$$u_{j+1}(k) = Q(q)[u_j(k) + L(q)e_j(k+1)] \tag{2}$$

$$e_j(k+1) = y_d(k+1) - y_j(k+1) \tag{3}$$

$$\rho(\boldsymbol{Q}(\boldsymbol{I} - \boldsymbol{LP})) < 1 \tag{4}$$

where Q is a filter, L is learning function, e_j is the tracking error, y_d is the desired output, $\rho\|\bullet\|_\infty$ is the spectral radius of the matrix and the bold items stand for lifted system matrix representation.

There are some popular algorithms in designing ILC. The PID-type learning function can be applied without accurate system model. The plant inversion method can converge quickly but relies a lot on modeling accuracy and is quite sensitive to model uncertainty. The H_∞ design technique can be used to design a robustly convergent ILC but at the expense of performance. The PID-type learning function is chosen in the following PDILC design as it is tunable with no specific need for system model. The discrete-time, PD-type learning function can be written as

$$u_{j+1}(k) = u_j(k) + k_p e_j(k+1) + k_d[e_j(k+1) - e_j(k)] \tag{5}$$

where k_p is the proportional gain and k_d is the derivative gain [22].

B. Position Domain Control

The position domain control for contour tracking tasks was firstly proposed in [23]. A multi-axis motion system is treated as a master-slave cooperative motion system. The master motion is sampled equidistantly and used as a reference, while the slave motions are expressed as functions of the master motion according to desired contour requirements [8]. The PD type PDC, PID PDC and cross-coupled PID control in position domain have been proposed in [8], [24] and [25].

For a two DOF decoupled parallel motion system, a PD-type feedback control signal $u_y(x)$ of y-axis (slave motion) in position domain is related to x-axis position (master motion), which can be expressed as

$$\begin{cases} u_y(x) = K_{py}e_y(x) + K_{dy}e_y^{'}(x) \\ e_y(x) = y_d(x) - y(x) \\ e_y^{'}(x) = y_d^{'}(x) - y^{'}(x) \end{cases} \tag{6}$$

where K_{py} and K_{dy} are proportional gain and differential gain, e_y is y-axis tracking error [8]. It should be noticed that position domain PD law uses *x*-axis position as a reference rather than time. Convert (6) to time domain (8) using following (7).

$$\dot{y}(t) = \frac{dy}{dt} = \frac{dy}{dx}\frac{dx}{dt} = y'(x)\dot{x}(t) \tag{7}$$

$$u_y(t) = K_{py}(y_d(t) - y(t)) + \frac{K_{dy}}{\dot{x}(t)}(\dot{y}_d(t) - \dot{y}(t)) \tag{8}$$

From (8), it can be seen that the control force is sensitive to the noise under a low speed of the master motion [8]. To solve this problem, a modified control law using sampling distance of the master motion is shown as (9).

$$u_y(x) = K_{py}e_y(t) + \frac{K_{dy}}{\Delta x}(e_y(t) - e_y(t - \Delta t)) \tag{9}$$

The position domain PD control law can be viewed as a varying sampling rate PD control in the time domain [8]. It can also be viewed as a nonlinear PD control in the time domain when the speed of *x*-axis is not constant [23]. The PID type ILC will be chosen in the following sections in PDILC design.

C. Contour Error Computation and Estimation

Contour error is defined as the distance between actual position and the nearest point in reference trajectory [10]. In XY plane contour tracking, contour error ε can be computed by (10).

$$\varepsilon = -C_x e_x + C_y e_y \tag{10}$$

where the coupling operators are selected as (11) for a linear contour and estimated as (12) for a curved contour [9].

$$C_x = \sin\theta, C_y = \cos\theta \tag{11}$$

$$C_x = \sin\theta - \frac{e_x}{2R}, C_y = \cos\theta + \frac{e_y}{2R} \tag{12}$$

For a non-circular contour, an estimation can be made by dividing it into parts and regarding them as a part of circle on condition that the axis errors are much smaller than the instantaneous radius of curvature [10]. This approach needs a lot work of computation and is low efficient.

A modified variable coupling operators based on the contouring error vector by applying the linear contour approximation was propose in [12]. The geometrical relations of biaxial motion systems among the desired contour, actual position *P* and reference position *R* in a biaxial systems are shown in Fig. 1. The estimated contouring error vector $\vec{\varepsilon}$ 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 tangential vector $\vec{t}$ can be computed by functional relationship between master motion and slave motion. The normalized normal vector $\vec{n}$ and the estimated contouring error vector $\vec{\varepsilon}$ then can be directly derived as

$$\vec{t}=\begin{bmatrix} t_x \\ t_y \end{bmatrix} \tag{13}$$

$$\vec{n}=\begin{bmatrix} n_x \\ n_y \end{bmatrix}=\begin{bmatrix} -\dfrac{t_y}{\sqrt{t_x^2+t_y^2}} \\ \dfrac{t_x}{\sqrt{t_x^2+t_y^2}} \end{bmatrix} \tag{14}$$

$$\vec{\varepsilon}=\varepsilon\cdot\vec{n}=\left\langle \vec{e},\vec{n}\right\rangle\cdot\vec{n} \tag{15}$$

where $\langle\bullet,\bullet\rangle$ is an inner product operator. The error vector method will be adopted in the following sections to estimate contour error for non-linear contour tracking case.

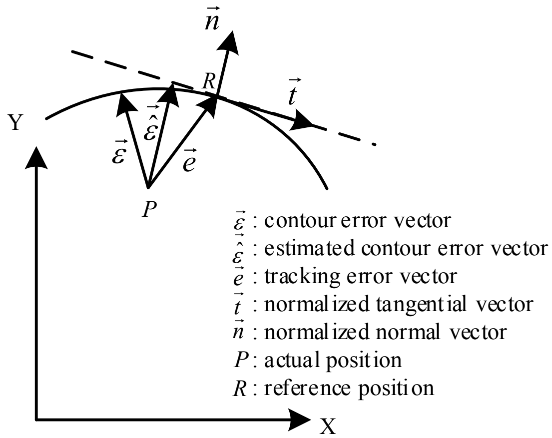

Figure 1. Geometrical relations of biaxial motion systems [12]

I. Position Domain Iteration Learning Control

A. Control Law

For a linear time invariant (LTI) system with two DOFs, the PDILC control signal of mater motion x-axis and slave motion y-axis can be given as

$$\begin{cases} u_{x_{j+1}}(x)=Q[u_x(x)+L_x e_x(x)]_j \\ u_{y_{j+1}}(x)=Q[u_y(x)+L_y e_y(x)]_j \end{cases} \tag{16}$$

where j is the iteration index, u is the control signal, Q is a filter, L is the learning function. Here, x-axis is chosen as the master motion, while y-axis is the slave motion. Applying PID type ILC, the y-axis control law can be rewritten as

$$u_{y_{j+1}}(x)=Q\left[u_y(x)+K_{py}e_y(x)+K_{iy}\int_0^{\Delta s}e_y(x)dx+K_{dy}e_y{}'(x)\right]_j \tag{17}$$

where K_{py} , K_{iy} and K_{dy} are PID gains for y-axis ILC controller. We will mainly discuss stability and performance of y-axis in the following steps.

B. Stability

Convert (17) to time domain (18) using (6) and (7). For briefness, f' and $\dot{f}$ in following analysis are short for $f'(x)$ and $f'(t)$, respectively.

$$u_{y_{j+1}}=Q\left[u_y+K_{py}e_y+K_{iy}\int_0^{\Delta t}e_y\dot{x}dt+K_{dy}\frac{\dot{e}_y}{\dot{x}}\right] \tag{18}$$

As analyzed in (8) and (9), numerical error of control signal may occur when x-axis under a low speed. A modified control law using sampling distance of the master motion Δx is shown as

$$\begin{cases} \int_0^{\Delta t}e_y(t)\dot{x}(t)dt=\dfrac{\Delta x}{2}\left(e_y(t)+e_y(t-\Delta t)\right) \\ \dfrac{\dot{e}_y(t)}{\dot{x}(t)}=\dfrac{e_y(t)-e_y(t-\Delta t)}{\Delta x} \end{cases} \tag{19}$$

Substituting (19) into (18) yielding,

$$\begin{cases} u_{y_{j+1}}(t)=Q\left[u_y(t)+\alpha\cdot e_y(t)+\beta\cdot e_y(t-\Delta t)\right]_j \\ \alpha=K_{py}+\dfrac{\Delta x}{2}\cdot K_{iy}+\dfrac{K_{dy}}{\Delta x} \\ \beta=\dfrac{\Delta x}{2}\cdot K_{iy}-\dfrac{K_{dy}}{\Delta x} \end{cases} \tag{20}$$

To discuss the asymptotically stable condition of the control law in (20), lifted system representation is adopted. The lifted system matrix can be formed experimentally by applying an impulse input to the system dynamic of (1). The linear plant dynamics then can be rewritten as

$$\underbrace{\begin{bmatrix} y_i(1) \\ y_i(2) \\ \vdots \\ y_i(N) \end{bmatrix}}_{Y_j}=\underbrace{\begin{bmatrix} p_1 & 0 & \cdots & 0 \\ p_2 & p_1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ p_N & p_{N-1} & \cdots & p_1 \end{bmatrix}}_{P}\underbrace{\begin{bmatrix} u_j(1) \\ u_j(2) \\ \vdots \\ u_j(N) \end{bmatrix}}_{U_j}+\underbrace{\begin{bmatrix} d(1) \\ d(2) \\ \vdots \\ d(N) \end{bmatrix}}_{d} \tag{21}$$

where Y_j, P, U_j and d are lifted representation of y_j, P, u_j and d with a matrix form, respectively [26]. Equation (20) can be rewritten as

$$\boldsymbol{u}_{y_{j+1}}(k)=\boldsymbol{Q}\left[(\boldsymbol{I}-\alpha\boldsymbol{P}_y-\beta\boldsymbol{P}'_y)\boldsymbol{u}_{y_j}(k)+(\alpha+\beta q^{-\Delta k})(\boldsymbol{y}_d(k)-\boldsymbol{d}(k))\right] \tag{22}$$

where $\boldsymbol{P}'_y(q)=q^{-\Delta k}\bullet\boldsymbol{P}_y(q)$, $q^{-\Delta k}$ stands for time delay with discrete step length of Δk. System is asymptotically stable **(AS)** if there exists appropriate parameters α and β satisfying

$$\rho(\boldsymbol{Q}(\boldsymbol{I}-\alpha\boldsymbol{P}_y-\beta\boldsymbol{P}'_y))<1 \tag{23}$$

where the parameter α and β are bounded on condition that the trajectory and motion of x-axis are planned appropriate with the position increment Δx bounded. For further analysis in computational burden of large matrix manipulations, one may refer to [14], [15].

C. Performance

If the AS condition (23) is satisfied, the asymptotic control input and steady state error is described in (24). Performance is often judged by the decrease from initial error to converged error. In simulation and experimental data analysis, the root mean square (RMS) value of error is chosen as evaluation index for the proposed algorithm.

$$\begin{cases} \boldsymbol{e}_{y_\infty}(k) = \boldsymbol{y}_d(k) - \boldsymbol{P}_y \boldsymbol{u}_\infty(k) - \boldsymbol{d}(k) \\ \boldsymbol{u}_{y_\infty}(k) = \boldsymbol{Q} \dfrac{(\alpha + \beta q^{-\Delta k})}{\boldsymbol{I} - \boldsymbol{Q}(\boldsymbol{I} - \alpha \boldsymbol{P}_y - \beta \boldsymbol{P}_y')} (\boldsymbol{y}_d(k) - \boldsymbol{d}(k)) \end{cases} \tag{24}$$

II. Evaluation: Simulation and Experiment Case

The proposed PDILC was evaluated through both simulation and experiment case based on a three DOF precise positioning stage shown in Fig. 2. A triangle trajectory and a parabolic trajectory were used to test the algorithm effectiveness under linear and nonlinear references displayed in Fig. 3. The velocity profiles were planned smoothly to avoid numerical error in (18).

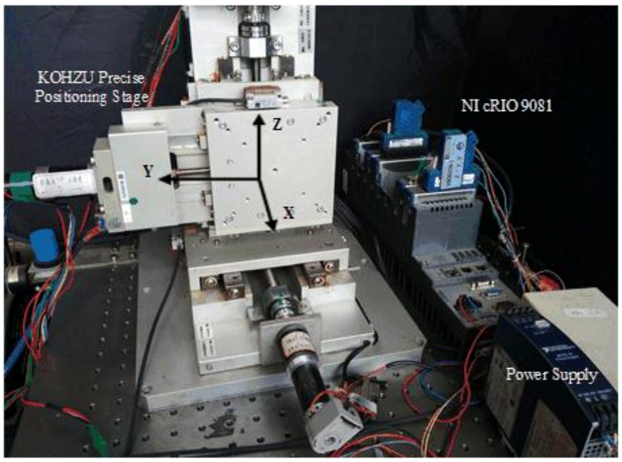

Figure 2. Experiment system

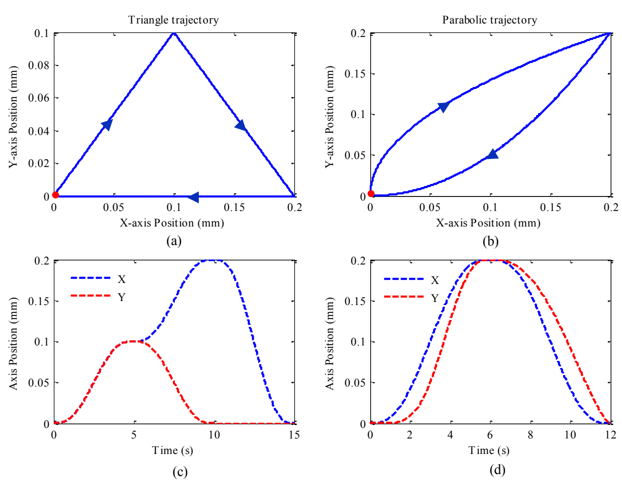

Figure 3. Reference trajectories. (a)Triangle motion. (b)Parabolic motion. (c)Triangle axis motion versus time. (d) Parabolic axis motion versus time.

A. System Description

The tested platform shown in Fig. 2 is a serial system. Each axis is driven by a DC motor and the position is detected by a grating ruler. For 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 1kHz. The continuous transfer functions are listed in (25).

$$\begin{cases} Gx = \dfrac{6.878e^{-5}s^2 - 0.1402s + 5.291}{s^2 + 5.795s + 5.564} \\ Gy = \dfrac{-0.0631s + 2.132}{s^2 + 2.76s + 2.127} \end{cases} \tag{25}$$

The proposed PDILC is an open-loop control and has no feedback mechanism to reject unexpected, nonrepeating disturbances. A PID controller combined with a PDILC controller for each axis was designed to perform simulation and experiment. The control structure then can be described as Fig. 4 shows.

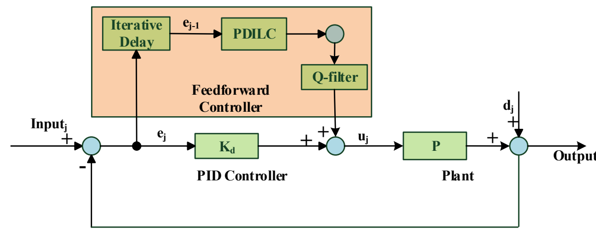

Figure 4. Control structure

B. Simulation Results

Based on the asymptotically stable condition in (23), a set of parameter was chosen as listed in Tab. 1 for the feedback PDILC controller. The feedback PID gains were determined according to system stability and tracking performance in step response simulation. It should be noticed that the choice of feedback PID gains was not the emphasis in this paper, so the gains were set the same for time domain ILC (TDILC) and PDILC to make a persuasive comparison between TDILC and PDILC control performance.

TABLE I. Controller Parameters

Axis	Feedback PID Gains			Feedforward PDILC Gains		
	Kp	*Ki*	*Kd*	*Kp*	*Ki*	*Kd*
X	3	1	1	0.3	0.1	0
Y	3	2	1	0.2	0.05	0

Applying parameters in Tab. 1 and reference trajectory in Fig. 3, performance of feedback PID (FB), FB combined with TDILC, FB combined with PDILC on the system in (25) were evaluated. Simulation results of triangle motion are provided in Fig. 5 and Table 2.

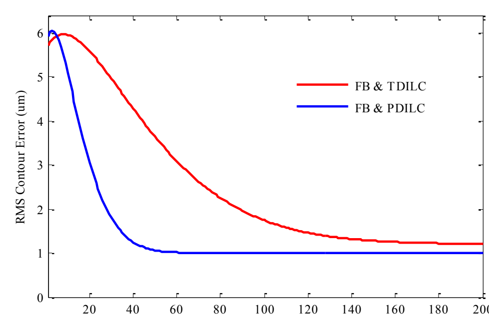

Figure 5. RMS contour error

Figure 5 shows the RMS contour error of FB combined with TDILC and FB combined with PDILC in the iteration process. Under the condition of a same set of gain parameters, PDILC converges more quickly (less than 40 iterations) and results in a lower final RMS value than TDILC (about 160 iterations).

TABLE II. SIMULAITON RESULTS FOR RMS ERRORS

Controller	RMS Value (μm)		
	X	***Y***	***ε***
FB	19.28	10.65	5.72
FB & TDILC	3.43	3.03	1.21
FB & PDILC	2.14	2.30	1.01

Table 2 lists the RMS values of each axis and the contour errors. The PDILC improves the *x*-axis RMS error by 89%, the *y*-axis RMS error by 78% and the contour RMS error by 82%, resulting in the best among the three.

C. Experimental Results

To validate the simulation results, the proposed PDILC was tested on actual platform in Fig. 2. The NI cRIO 9081 was used for implement of controllers. Fig. 6 and Tab. 3 are results of triangle motion tracking. Figure 7 and Table 4 display the results of parabolic motion tracking.

Similar to the simulation results, FB & PDILC control produces the best tracking performance with an 84% decrease in x-axis RMS value, an 88% decrease in y-axis RMS value and a 72% decrease in contour RMS value. The improvement from TDILC to PDILC in contour RMS is about 6%. Besides, the convergence process of PDILC is more quickly (less than 50 iterations) than TDILC (about 180 iterations) as expected in simulation case. The increase in experimental error results is to be expected as the existence of unmodelled dynamics in the actual environment.

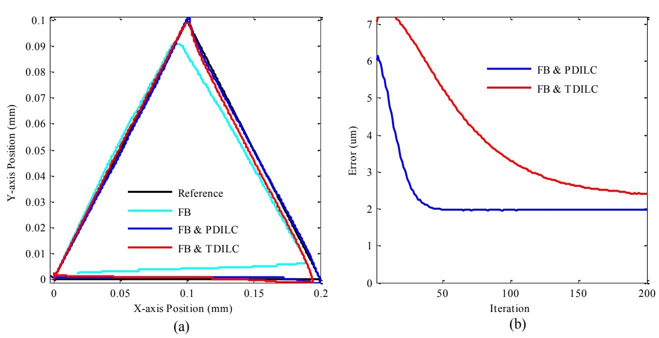

Figure 6. Experimental results of triangle contour tracking. (a) Tracking in xy plane. (b) RMS contour error in iteration process

TABLE III. EXPERIMENT RESULTS OF TRIANGLE MOTION

Controller	RMS Value (μm)		
	X	***Y***	***ε***
FB	18.51	9.93	7.05
FB & TDILC	4.80	2.28	2.40
FB & PDILC	2.94	1.15	1.98

Figure 7 shows experiment results of each individual axis tracking performance. Errors under FB & PDILC are smallest and smoothest versus time compared with FB and FB & TDILC control. The *x*-axis and *y*-axis under FB & PDILC get a 10% improvement and an 11% improvement in RMS value from TDILC to PDILC.

From the control law in (18) and (20), it can be found that the decoupled two axis are coupled through introducing the velocity of *x*-axis into *y*-axis control. The coupling control law results in the improvement of contour error and individual tracking performance as the experiment results show.

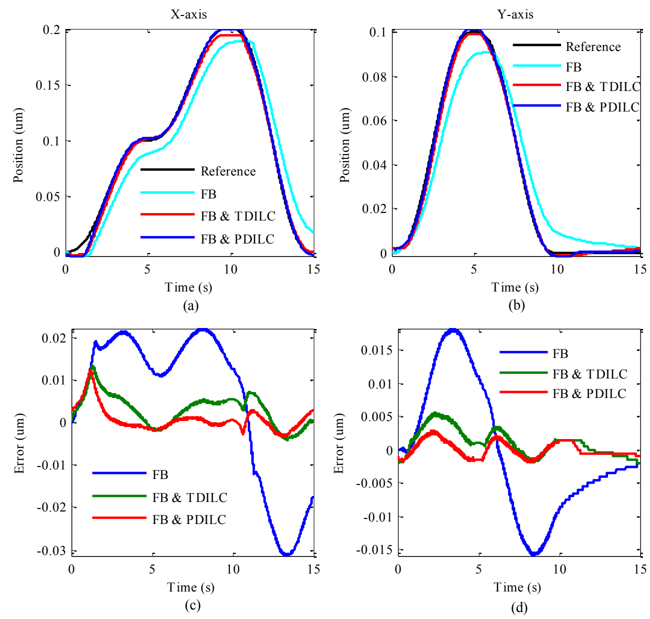

Figure 7. Experimental results of triangle axis tracking. (a) X-axis Tracking versus time. (b) Y-axis Tracking versus time. (c) X-aixs error versus time. (d) Y-aixs error versus time.

To evaluate the control law in nonlinear contour tracking case, a parabolic contour tracking experiment was conducted. The difficulty in nonlinear contour tracking lies in contour error estimation as the coupling operators in (10) are not constant nor specific functions. The error vector method in (15) was applied in this case. Figure 8, figure 9 and Table 4 are the results.

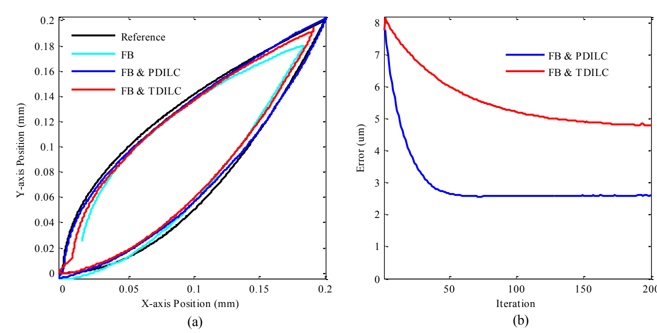

Figure 8. Experimental results of parabolic contour tracking (a) Tracking in xy plane. (b) RMS contour error in iteration process

Analogous to the results in triangle tracking case, FB & PDILC control produces the best tracking performance. The RMS error decrease of x-axis, y-axis and contour under FB and PDILC are, respectively, 61%, 72% and 67%. The improvements from TDILC to PDILC are 24%, 22% and 28% for x-axis, y-axis and contour, respectively.

TABLE IV. EXPERIMENT RESULTS OF PARABOLIC MOTION

Controller	**RMS Value (μm)**		
	X	***Y***	***ε***
FB	22.04	22.94	7.80
FB & TDILC	14.01	11.61	4.79
FB & PDILC	8.52	6.49	2.59

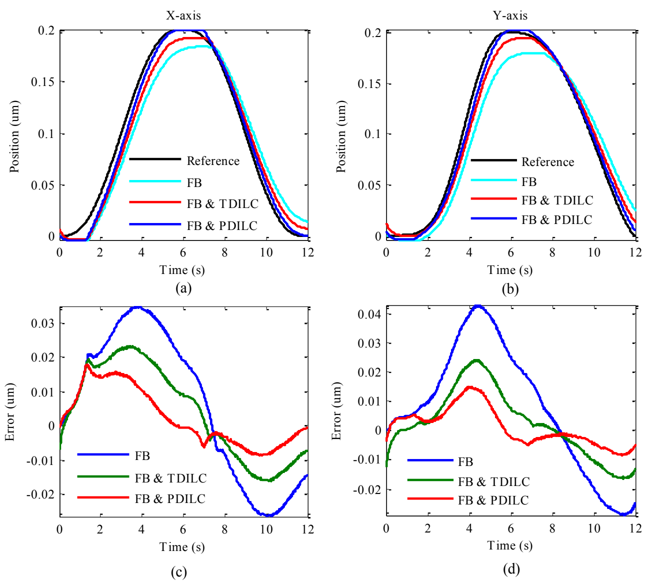

Figure 9. Experimental results of parabolic axis tracking. (a) X-axis Tracking versus time. (b) Y-axis Tracking versus time. (c) X-aixs error versus time. (d) Y-aixs error versus time.

III. CONCLUSION

In this paper, a position domain iteration learning control algorithm has been presented for contour error improvement in a multi-axis motion system. The control law and stability condition were discussed in lifted system representation. Simulation and experiment of both linear and nonlinear contour tracking cases were conducted in a three DOF precise positioning stage. A set of controllers (FB, FB & TDILC, FB & PDILC) were tested to make comparisons. The combination of FB and PDILC was found to be best control system for both individual axis and contour tracking performance.

One future work is the optimal design of parameters in the combined PID and PDILC structure.

ACKNOWLEDGMENT

This research is sponsored by National Natural Science Foundation of China (NSFC, Grant No.51375349 and No.51175383).

REFERENCES

[1] S. Devasia, E. Eleftheriou, and S. Moheimani, "A survey of control issues in nanopositioning," *IEEE Trans. Control Syst. Technol.*, vol. 15, no. 5, pp. 802–823, Sep. 2007.

[2] S. M Raafat and R. Akmeliawati, "Survey on robust control of precision positioning systems," *Recent patents on mechanical engineering*, vol. 5, pp. 55-68, Jan. 2012.

[3] X. Yu and O. Kaynak, "Sliding-mode control with soft computing: A survey," *IEEE Trans. Ind Electron.*, vol. 56, pp. 3275-3285, Sep. 2009.

[4] H. S. Ahn, Y. Chen, and K. L. Moore, "Iterative learning control: brief survey and categorization," *IEEE Trans. Syst Man Cybern C Appl Rev.*, vol. 37, p. 1099, Nov. 2007.

[5] Y. Shan and K. K. Leang, "Design and control for high-speed nanopositioning: serial-kinematic nanopositioners and repetitive control for nanofabrication," *Control Syst Mag.*, vol. 33, pp. 86-105, Dec. 2013.

[6] Y. Shan and K. K. Leang, "Accounting for hysteresis in repetitive control design: Nanopositioning example," *Automatica*, vol. 48, pp. 1751-1758, Aug. 2012.

[7] S. S. Aphale, B. Bhikkaji, and S. R. Moheimani, "Minimizing scanning errors in piezoelectric stack-actuated nanopositioning platforms," *IEEE Trans. Nanotechnol.*, vol. 7, pp. 79-90, Jan. 2008.

[8] P. Ouyang, T. Dam, J. Huang, and W. Zhang, "Contour tracking control in position domain," *Mechatronics*, vol. 22, pp. 934-944, Oct. 2012

[9] Y. Koren, "Cross-coupled biaxial computer control for manufacturing systems," *J. Dyn. Syst., Meas. Contr.*, vol. 102, pp. 265-272, 1980.

[10] Y. Koren and C.-C. Lo, "Variable-gain cross-coupling controller for contouring," CIRP Proc.-Manufacturing Systems, vol. 40, pp. 371-374, 1991.

[11] Y. Koren and C. Lo, "Advanced controllers for feed drives," CIRP Proc.-Manufacturing Systems, vol. 41, pp. 689-698, 1992.

[12] S. S. Yeh and P.-L. Hsu, "Estimation of the contouring error vector for the cross-coupled control design," *IEEE/ASME Trans. Mechatronics*, vol. 7, pp. 44-51, Mar. 2002.

[13] K. L. Barton, and A. G. Alleyne, "Cross-coupled ILC for improved precision motion control: Design and implementation." in *Proc. Amer. Control Conf*, 2007, pp. 5496-5502.

[14] K. L. Barton, and A. G. Alleyne, "Cross-coupled iterative learning control: Design and implementation." *Proceedings of IMECE*. 2006.

[15] K. L. Barton, and A. G. Alleyne, "A cross-coupled iterative learning control design for precision motion control," *IEEE Trans. Control Syst. Technol.*, vol. 16, no. 6, pp. 1218-1231, Nov. 2008

[16] 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," *Int. J. Control*, vol. 84, pp. 1223-1233, Jul. 2011.

[17] P. Ouyang, V. Pano, and J. Acob, "Position domain contour control for multi-DOF robotic system," *Mechatronics*, vol. 23, pp. 1061-1071, Dec. 2013.

[18] T .Dam, and P. Ouyang, "Contour tracking control in position domain for CNC machines," in ICIA, Shenzhen, China, 2011, pp. 8-13.

[19] M. Uchiyama, "Formation of high-speed motion pattern of a mechanical arm by trial," *Trans. Soc. Instrument Contr. Engineers*, vol. 14, no. 6, pp.706–712, 1978.

[20] J.J. Craig, "Adaptive control of manipulators through repeated trials," in *Proc. Amer. Contr. Conf*, 1984., pp. 1566–1573.

[21] D. A. Bristow, M. Tharayil, and A. G. Alleyne, "A survey of iterative learning control," *Control Syst Mag.*, vol. 26, pp. 96-114, Jun. 2006.

[22] S. Arimoto, S. Kawamura, and F. Miyazaki, "Bettering operation of robots by learning," *J. Robot. Syst.*, vol. 1, pp. 123–140, 1984.

[23] P. Ouyang, V. Pano, T .Dam, "PID position domain control for contour tracking," *International Journal of Systems Science*, vol. 46, no. 1, pp. 111-124, 2015

[24] P. Ouyang, T. Dam, and V. Pano, "Cross-coupled PID control in position domain for contour tracking," *Robotica*, pp. 1-24, 2014.

[25] C. T. Chen, *Linear System Theory and Design*. New York: Oxford Univ.Press, 1999.