Proceedings of the
47th IEEE Conference on Decision and Control
Cancun, Mexico, Dec. 9-11, 2008



# Norm Optimal Cross-Coupled Iterative Learning Control

Kira Barton*, Jeroen van de Wijdeven†, Andrew Alleyne*, Okko Bosgra†, and Maarten Steinbuch†

*University of Illinois at Urbana-Champaign
Dept. of Mechanical Science and Engineering
140 MEB, M/C 244, 1206 W. Green Street, Urbana, IL 61801
{barton2, alleyne}@illinois.edu

†Technische Universiteit Eindhoven
Dept. of Mechanical Engineering
P.O. Box 513, 5600 MB Eindhoven, The Netherlands
{J.J.M.v.d.Wijdeven, O.H.Bosgra, M.Steinbuch}@tue.nl

*Abstract*— **In this paper, we focus on improving contour tracking in precision motion control (PMC) applications through the use of Cross-Coupled Iterative Learning Control (CCILC). Initially, the relationship between individual axis errors and contour error is discussed, including insights into the different reasons for implementing CCILC versus individual axis ILC. A Norm Optimal (N.O.) framework is used to design optimal learning filters based on design objectives. The general N.O. framework is reformatted to include the contour error, as well as individual axis errors. General guidelines for tuning the different weighting matrices are presented. The weighting approach of this framework enables one to focus on individual axis or contour tracking independently. The performance benefits of N.O. CCILC versus ILC are illustrated through simulation and experimental testing on a multi-axis robotic testbed.**

## I. Introduction

In this paper we present a method for improving the precision motion control (PMC) of multiple input multiple output (MIMO) manufacturing systems that perform the same task repetitively. PMC becomes increasingly important for tracking contoured reference trajectories in multi-axis systems [1]. Contoured trajectories require coordinated positioning between two or more axes. In these systems, shifting the focus from individual axis tracking to contour tracking may result in a final outcome that more closely resembles the desired trajectory.

The general iterative learning control (ILC) approach for PMC of contoured trajectories on multi-axis repetitive systems has been to implement individual ILC controllers on each axis. These controllers are designed to minimize individual axis errors. This approach works well for trajectories that lie well within the bandwidth of the individual systems. However, for trajectories that contain frequency content outside the bandwidth of the individual systems, the performance of the multi-axis system may begin to degrade.

In [2]–[4], a novel ILC control design which focuses on contour tracking was introduced. Cross-Coupled Iterative Learning Control (CCILC) combines feedback Cross-Coupled Control (CCC) [5]–[7] with ILC [8], [9] into a learning control design which focuses on minimizing the contour tracking of a given system. In order to place more emphasis on the contour tracking, enhanced performance for the individual axis tracking may need to be compromised.

While each of these learning controllers focus on one aspect of the tracking errors, it has been shown that a combined controller (CCILC & ILC) [3] may result in improved overall tracking performance. One of the challenges in combining the two learning controllers emerges when trying to emphasize one learning design over the other. The optimal learning controller depends on several factors, such as the system requirements, the reference trajectory and the design objectives. The goal of this work is to reformat this combined learning controller into a Norm Optimal (N.O.) framework [10]. The N.O. framework gives a structured approach for focusing on contour tracking, which results in a more intuitive design approach. The weighting approach of this framework also enables one to focus on individual axis or contour tracking as determined by the control designer.

The outline of this paper is as follows. Section II motivates the use of contour control and provides the definition of contour error with respect to individual axis errors. The N.O. framework with regards to ILC will be discussed in Section III, including guidelines for tuning the weighting matrices and the formulation of a typical cost function. The experimental setup and design of a CCILC controller using the N.O. framework are described in Section IV. Results from implementation of N.O.-CCILC and N.O.-ILC controllers to a multi-axis robotic testbed are given in Section V. Conclusions are given in Section VI.

## II. Motivation

In many multi-axis systems, the separate axes operate independently from each other. This allows the controls engineer to design individual controllers for each axis. The individual controllers are designed to minimize the tracking errors for each axis, respectively.

The MIMO systems considered in this work consist of two or more uncoupled axes (see Fig. 1). While linear trajectories

978-1-4244-3124-3/08/$25.00 ©2008 IEEE

that only engage one axis at a time are still possible for these systems, this research focuses on contoured trajectories that require coupled movements from multiple axes in order to achieve the desired trajectory.

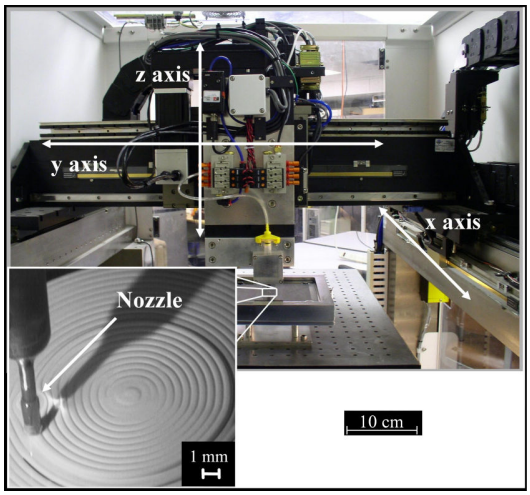


Fig. 1. *Multi-axis Robotic Testbed* [11]

The tracking performance of MIMO systems when following contoured trajectories can be defined with respect to individual axis and contour errors. Individual axis errors describe the difference between the reference position and the actual position of the axes with respect to the testbed coordinate system. Contour error is defined as the distance from the actual position to the closest point on the reference trajectory.

Although the individual controllers focus on individual axis errors, the contour error is also reduced due to the relationship between contour and individual axis errors. While this decoupled approach works well for trajectories in which the reference trajectory does not contain high frequency sections, the performance may begin to degrade for trajectories which require high frequency movements outside the bandwidth of the individual axes. For these trajectories, the multi-axis system may benefit from a controller which focuses on contour tracking, rather than individual axis tracking.

### A. Contour Error

Prior to formulating a controller which focuses on contour tracking, the definition of contour error should be analytically derived. Consider the curved trajectory of Fig. 2. The actual position of the system can be described with respect to the desired position in terms of individual axis errors, $e_x$ and $e_y$, and contour error, $\varepsilon$.

Contour error is a function of individual axis errors and time. As Fig. 2 illustrates, contour error can be defined as a linear approximation of the closest distance from the actual position to the instantaneous tangent line of the reference trajectory with respect to time. Mathematically, this can be

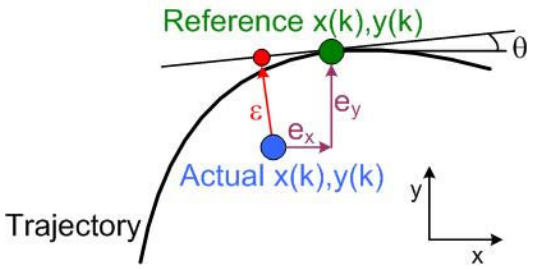


Fig. 2. *Contour versus individual axis errors*

shown as

$$\varepsilon(k) = -c_x(k) \cdot e_x(k) + c_y(k) \cdot e_y(k) \quad (1)$$

$$\varepsilon(k) = C(k) \cdot e(k), \quad (2)$$

where $C(k) \in \mathbb{R}^{1\times 2}$, $c_x(k)$ and $c_y(k)$ are known as coupling gains and are used to define the contour error with respect to the individual axis errors, and $k$ is the time interval from $k = 0, 1, \ldots, N-1$. The coupling gains are generally time-varying gains that change with respect to the trajectory. Linearized coupling gains have the following format

$$c_x(k) = \sin\theta(k); c_y(k) = \cos\theta(k),$$

where $\theta$ is defined as the instantaneous angle of the reference trajectory with respect to the x-axis of the testbed coordinate system.

Previous work in [2], [3] introduced an iterative learning control design termed CCILC which focuses on contour tracking. The objective of this work is to reformat that control structure into the N.O. framework. The generalized individual axis ILC structure for this framework, along with some guidelines for tuning the design of the N.O.-ILC controller, is given in the following section. In Section IV, the N.O. framework is reformatted to include contour tracking.

## III. Norm Optimal ILC

The ILC control problem in this paper is studied in the lifted setting, [10], [12]. In this setting, the discrete-time behavior of a linear time invariant (LTI) system $P(k)$ for time $k = 0, 1, \ldots, N-1$ is represented by its convolution matrix $P$ using impulse response data $H(k)$, (3).

$$P = \begin{bmatrix} H(0) & & 0 \\ \vdots & \ddots & \\ H(N-1) & \cdots & H(0) \end{bmatrix}. \quad (3)$$

For MIMO systems, $H(k)$ contains the impulse response from each of the $q_i$ inputs to each of the $q_o$ outputs, (4).

$$H(k) = \begin{bmatrix} H^{11}(k) & \cdots & H^{1q_i}(k) \\ \vdots & & \vdots \\ H^{q_o 1}(k) & \cdots & H^{q_o q_i}(k) \end{bmatrix}, \quad (4)$$

with $H^{il}(k)$ the impulse response from input $l$ to output $i$. Given $H(k) \in \mathbb{R}^{q_o \times q_i}$, system $P \in \mathbb{R}^{Nq_o \times Nq_i}$ is a lower triangular matrix with a block Toeplitz structure.

During trial $j$, system $P$ maps the input signal $u_j$ to the measured output signal $y_j$, i.e., $y_j = Pu_j$, with $u_j$ and $y_j$ defined in (5) and (6), respectively.

$$u_j = \begin{bmatrix} u_j^T(0) & u_j^T(1) & \cdots & u_j^T(N-1) \end{bmatrix}^T \quad (5)$$

$$y_j = \begin{bmatrix} y_j^T(0) & y_j^T(1) & \cdots & y_j^T(N-1) \end{bmatrix}^T, \quad (6)$$

$$\text{with} \quad u_j^T(k) = \begin{bmatrix} u_j^1(k) & \cdots & u_j^{q_i}(k) \end{bmatrix}$$

$$\text{and} \quad y_j^T(k) = \begin{bmatrix} y_j^1(k) & \cdots & y_j^{q_o}(k) \end{bmatrix}.$$

The ILC controller for our contour tracking problem results from a quadratic optimization problem, [13]. In this problem, we want to minimize an objective function $\mathcal{J}$, with $\mathcal{J}$ corresponding to the sum of weighted norms of the error $||e_{j+1}||_Q$, the command signal $||u_{j+1}||_S$, and the rate of change of the command signal $||u_{j+1} - u_j||_R$,

$$\mathcal{J} = e_{j+1}^T Q e_{j+1} + u_{j+1}^T S u_{j+1} + (u_{j+1} - u_j)^T R (u_{j+1} - u_j), \quad (7)$$

where $e_j = y_r - y_j$, with $y_r$ as the reference signal and $(Q, R, S)$ symmetric positive definite matrices (often $(Q, R, S) =: (qI, rI, sI)$). Note that in some cases $(Q, R, S)$ may be semi-definite matrices, as long as $P^TQP + S + R$ is positive definite.

Applying $y_r = e_j + Pu_j$ and $e_{j+1} = y_r - Pu_{j+1}$ leads to (8).

$$e_{j+1} = e_j - P(u_{j+1} - u_j). \quad (8)$$

By placing (8) in (7) and subsequently differentiating $\mathcal{J}$ with respect to $u_{j+1}$ and setting this derivative equal to zero, we find the norm-optimal ILC controller (9).

$$\begin{aligned} u_{j+1} &= \mathcal{Q}u_j + Le_j \quad (9) \\ \mathcal{Q} &= (P^TQP + S + R)^{-1}(P^TQP + R) \\ L &= (P^TQP + S + R)^{-1}P^TQ. \end{aligned}$$

Note the difference between the weight $Q$ and the ILC controller $\mathcal{Q}$.

Although this ILC control strategy is relatively well known, there is little in the available literature on how to tune $(Q, S, R)$. Therefore, we derive here some guidelines by studying the properties of the ILC controlled system with respect to convergence, performance, robust monotonic convergence, and performance in the presence of stochastic disturbances.

### A. Convergence

Given the ILC controller (9) and the system dynamics $y_j = Pu_j$, the trial domain dynamics can be given by

$$u_{j+1} = (\mathcal{Q} - LP)u_j + Ly_r. \quad (10)$$

For this system to be convergent, the spectral radius $\max_i |\lambda_i(\mathcal{Q} - LP)| < 1$. Furthermore, for monotonic convergence, we require the relatively well known condition $||\mathcal{Q} - LP||_{i2} < 1$ such that $\|u_{j+1}\|_2 < \|u_j\|_2$. Here $||\cdot||_{i2} = \overline{\sigma}(\cdot)$ and $\overline{\sigma}(\cdot)$ is defined as the largest singular value of a given matrix. Note that $\max_i |\lambda_i(\cdot)| \leq ||\cdot||_{i2}$.

For the norm-optimal ILC controller, we have $\mathcal{Q} - LP = (P^TQP + S + R)^{-1}R$. As a result, convergence is guaranteed for any symmetric positive (semi-)definite $(Q, S, R)$ with $P^TQP + S + R$ positive definite. Moreover, convergence speed, i.e. the rate $\kappa = \frac{\|u_{j+1}\|_2}{\|u_j\|_2}$, strongly depends on $R$. For $R = 0$ ($\kappa = 0$) deadbeat control is achieved, as $R \to \infty$ ($\kappa \to 1$) the convergence speed can be arbitrarily slow.

### B. Performance

For performance, we study the steady state error $e_\infty := \lim_{j\to\infty} e_j$. Consequently, we assume the ILC control system to be convergent.

The steady state error is derived from the steady state command signal $u_{j+1} = u_j = u_\infty$.

$$u_\infty = (\mathcal{Q} - LP)u_\infty + Ly_r \to u_\infty = (P^TQP + S)^{-1}P^TQy_r. \quad (11)$$

With (11) and $e_\infty = y_r - Pu_\infty$, $e_\infty$ is given by

$$e_\infty = (I - P(P^TQP + S)^{-1}P^TQ)y_r. \quad (12)$$

From (12), we can now conclude the following: the smallest attainable error, i.e., optimal performance, requires $S = 0$ and hence $P^TQP$ to be positive definite. Furthermore, $e_\infty$ is not a function of $R$, and hence performance is not a function of convergence speed.

### C. Robust monotonic convergence

In this subsection, we consider the true system $P_t$ to correspond to the nominal model $P$ plus an uncertainty $\Delta_P$: $P_t = P(I + \Delta_P)$, with the multiplicative uncertainty $\Delta_P = W\Delta$ and $||\Delta||_{i2} \leq 1$. As a result, for robust monotonic convergence we require

$$\begin{aligned} &||\mathcal{Q} - LP_t||_{i2} < 1 \Rightarrow \quad (13) \\ &\max_\Delta ||(P^TQP + S + R)^{-1}(R - P^TQPW\Delta)||_{i2} < 1. \end{aligned}$$

*Lemma 1:* Consider (13) with $R = 0$. Then a sufficient condition for robust monotonic convergence is given by $||(P^TQP + S)^{-1}P^TQPW||_{i2} < 1$.

*Proof:* Follows directly from (13) and the inequality $||(P^TQP + S)^{-1}P^TQPW\Delta||_{i2} \leq ||(P^TQP + S)^{-1}P^TQPW||_{i2} \cdot ||\Delta||_{i2}$. ■

*Lemma 2:* Consider (13) with $||(P^TQP + S)^{-1}P^TQPW||_{i2} < 1$, and assume $P^TQP + S$ symmetric and positive definite. Then robust monotonic convergence is guaranteed for all $R = rI,\ r \in \mathbb{R} \geq 0$.

*Proof:* See [14] for detailed proof. ■

From Lemma 2, we can conclude that for an ILC controlled system which is robustly monotonically convergent for $R = 0$, the design parameter $R = rI$ does *not* influence the robust monotonic convergence properties of the ILC controlled system. $S$, on the other hand, should be designed such that the robust monotonic convergence condition in Lemma 1 holds.

### D. Performance in the presence of stochastic disturbances

In this subsection, we extend performance aspects of norm-optimal ILC of subsection III-B by including stochastic disturbances $d_j$, i.e., by considering $e_j = y_r - y_j - d_j$. As is shown in [15], the influence of stochastic disturbances can

be reduced by reducing the convergence speed. With $R$ in the N.O.-ILC controller the dominant factor in convergence speed, the influence of $d_j$ on $e_j$ can be reduced by increasing $R$.

### *E. Summary*

Based on the previous subsections, the following tuning guidelines for norm-optimal ILC control can be given.

1) Design $Q$ to correspond to the desired weighting of the error (see Section IV-B).
2) Design $S$ such that the system is monotonically convergent. Start with an $S$ with $||S||_{i2}$ relatively large compared to $||P||_{i2}$. Subsequently, reduce $S$ until the system diverges.
3) Design of $R$: Start with $||R||_{i2} = 0$ and increase $R$ until the steady state error fluctuations are within desired bounds, or do not decrease anymore.

## IV. Implementation

### *A. N.O.-CCILC*

Up until this point we have been focusing on N.O.-ILC. In order to explore the performance benefits of contour tracking versus individual axis tracking, we now extend the N.O.-ILC design to N.O.-CCILC, in which the controller seeks to comply with contour tracking objectives. For this extension, we first construct a cost function, $\mathcal{J}$, of the form given in Section III, with contour error, $\varepsilon$, replacing individual axis errors, $e_x$ and $e_y$. The reader is asked to compare (14) with (7).

$$\mathcal{J} = \varepsilon_{j+1}^T Q \varepsilon_{j+1} + u_{j+1}^T S u_{j+1} + (u_{j+1} - u_j)^T R (u_{j+1} - u_j). \tag{14}$$

Substituting (2) for the contour error and setting $Q = aI$ results in:

$$\begin{aligned} \mathcal{J} = e_{j+1}^T (a \cdot Q_{ccilc}) e_{j+1} + u_{j+1}^T S u_{j+1} \\ + (u_{j+1} - u_j)^T R (u_{j+1} - u_j), \end{aligned} \tag{15}$$

with $Q_{ccilc}$ given by

$$Q_{ccilc} = \begin{bmatrix} C^T(0)C(0) & & 0 \\ & \ddots & \\ 0 & & C^T(N-1)C(N-1) \end{bmatrix} \tag{16}$$

where

$$C^T(k)C(k) = \begin{bmatrix} c_x(k)c_x(k) & -c_x(k)c_y(k) \\ -c_y(k)c_x(k) & c_y(k)c_y(k) \end{bmatrix}. \tag{17}$$

Contrary to the general practice of setting $Q = qI$ in N.O.-ILC [16], the modified matrix is singular, time-varying and block-diagonal. The change in the structure of the $Q$ weighting matrix results in a controller that only penalizes certain combinations of the individual axis errors, rather than every combination equally. This provides flexibility within the system by allowing combinations of the individual axis errors which result in improved contour tracking. These combinations may result in an increase of the individual axis errors by effectively *decoupling the position profile from the time profile*. This decoupling enables the system to perform movements to improve the contour tracking while subsequently decreasing the individual axis performance without being penalized.

Combining (7) (with $Q = bI$) and (16), a combined cost function capable of focusing on either individual axis tracking, contour tracking or a combination of the two can be derived.

$$\begin{aligned} \mathcal{J} = e_{j+1}^T (a \cdot Q_{ccilc} + bI) e_{j+1} + u_{j+1}^T S u_{j+1} \\ + (u_{j+1} - u_j)^T R (u_{j+1} - u_j). \end{aligned} \tag{18}$$

The gains $a$ and $b$ refer to the weighting gains applied to the contour tracking or individual axis tracking, respectively. In order to ensure an equivalent comparison between N.O.-CCILC, N.O.-ILC, and N.O.-CCILC plus N.O.-ILC, the relationship $a + b = 1$ must be satisfied. Using this relationship, it can be shown that increasing $a$ results in a decrease in contour errors, while increasing $b$ results in a decrease in individual axis errors.

### *B. Experimental Setup*

The experimental system used to verify the tracking performance of the N.O.-CCILC and N.O.-ILC controllers is the multi-axis robotic testbed of Fig. 1. For simulation purposes, dynamics models of the $x$ and $y$ axes, along with stabilizing feedback controllers, were developed in [11].

$$P_i(z) = \frac{K(z+\alpha_{i1})(z^2 - \alpha_{i2}z + \alpha_{i3})(z^2 - \alpha_{i4}z + \alpha_{i5})}{(z-\beta_{i1})(z-1)(z^2 - \beta_{i2}z + \beta_{i3})(z^2 - \beta_{i4}z + \beta_{i5})}. \tag{19}$$

$$k_{pi}(z) = \frac{k(z-\alpha_{i1})(z-\alpha_{i2})(z-\alpha_{i3})}{(z-\beta_{i1})(z-\beta_{i2})(z-\beta_{i3})}, i = x, y. \tag{20}$$

The reference signal applied to the system is a raster scanning trajectory ($N = 1050$), in which the motion consists of long periods of low frequency content followed by short periods of high frequency transitions from one direction to another. This type of trajectory is commonly used in atomic force microscopy (AFM), as well as other manufacturing systems which require sharp transitions between signals. The transition points, labeled $A$ and $B$ on Fig. 3, correspond to locations within the trajectory where the desired trajectory requires movements outside the bandwidth of the individual axes. These areas indicate potential opportunities for N.O.-CCILC to provide improved tracking capabilities as compared to N.O.-ILC with respect to contour tracking.

Learning filters, $(\mathcal{Q}, L)$, of the form given in (9) were designed using the combined cost function, (18),

$$\begin{aligned} \overline{Q} &= (a \cdot Q_{ccilc} + bI) \\ \mathcal{Q} &= (P^T \overline{Q} P + 1e^{-3} I + 1e^{-3} I)^{-1} (P^T \overline{Q} P + 1e^{-3} I) \\ L &= (P^T \overline{Q} P + 1e^{-3} I + 1e^{-3} I)^{-1} P^T \overline{Q}. \end{aligned} \tag{21}$$

where the weighting gains for the $S$ and $R$ tuning matrices were found using the tuning rules summarized at the end of Section III. By varying the values of $a$ and $b$ over the interval $[0, 1]$ while satisfying the relationship $a + b = 1$, controllers

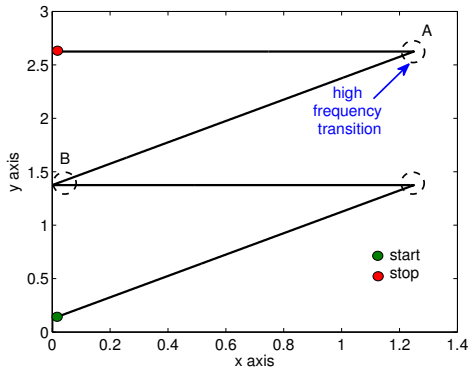


Fig. 3. *Raster scan trajectory*

with adjustable focus on individual axis or contour tracking errors could be implemented on the testbed.

## V. Results

### A. Simulation Results

Using the N.O. controllers (21), the models of the experimental testbed (19) stabilized with (20), and the reference trajectory described above, the following simulation results were obtained. Figure 4 shows converged root mean square (RMS) contour tracking errors for increasing values of $a$. Increasing $a$ corresponds to an increasing focus on the contour errors versus the individual axis errors. As one would expect, focusing on contour error results in decreasing values of converged RMS contour error. A controller which focuses entirely on contour errors versus individual axis errors results in a 28% improvement in the converged RMS contour error.

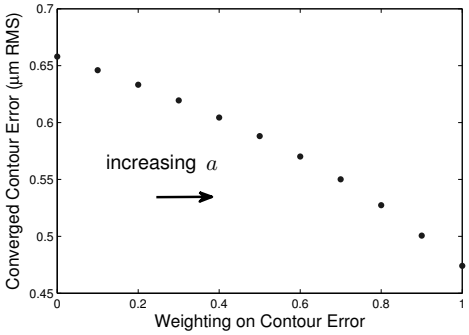


Fig. 4. *Converged RMS Contour Error for Increasing $a$*

Although the converged contour error has decreased, as Fig. 5 illustrates, the improvement in contour tracking is at the expense of individual axis tracking. The converged individual axis errors increase as the value of $a$ is increased. This indicates that the enhanced flexibility in position and time is directly related to a decrease in individual axis errors. As long as position and time synchronization is not a critical design objective, the reduction in individual axis tracking may be an acceptable trade-off for improved contour tracking.

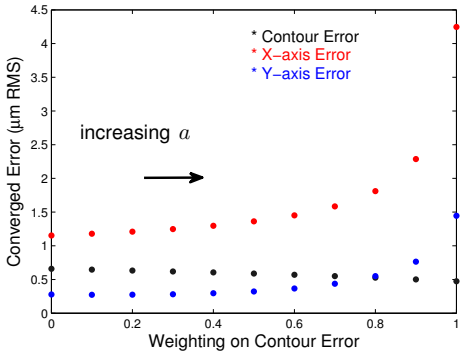


Fig. 5. *Converged RMS Errors for Increasing $a$*

Having seen that varying the values of $a$ and $b$ affects the converged RMS tracking errors, we want to explore how these differences translate to the actual position tracking as a function of time. Figure 6 shows an enlarged image of one raster corner in which the system is attempting to track an instantaneous change in direction. As Fig. 6 illustrates, a N.O.-ILC controller that focuses on individual axis tracking, $a = 0$ and $b = 1$, deviates from the contoured trajectory prior to the high frequency corner and performs oscillatory behavior directly following the corner before settling back onto the desired trajectory. On the other hand, an N.O.-CCILC controller that focuses on contour tracking, $a = 1$ and $b = 0$, results in a system that more closely tracks the reference trajectory throughout the time interval.

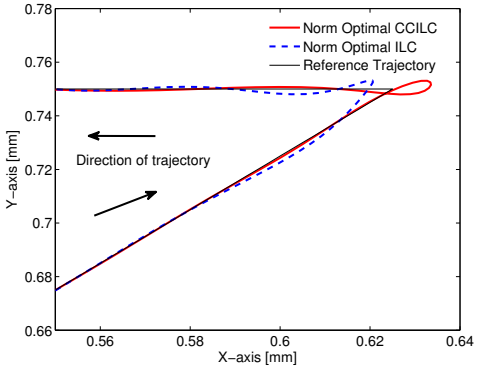


Fig. 6. *Trajectory tracking for N.O.-CCILC vs N.O.-ILC [Simulation]*

### B. Experimental Results

In order to validate the simulation results, the N.O.-CCILC and N.O.-ILC controllers used in simulation were implemented on the actual robotic testbed from Fig. 1. Analogous to the simulation results, N.O.-CCILC resulted in more precise contour tracking as compared to individual axis N.O.-ILC. The simulation results are shown to closely predict the behavior of the N.O. learning controllers on the robotic testbed, as can be seen by comparing Fig. 6 and Fig. 7.

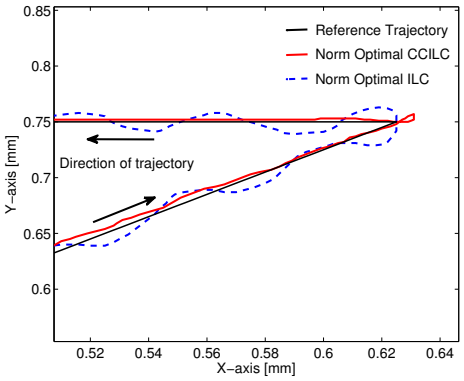


Fig. 7. *Trajectory tracking for N.O.-CCILC vs N.O.-ILC [Experimental]*

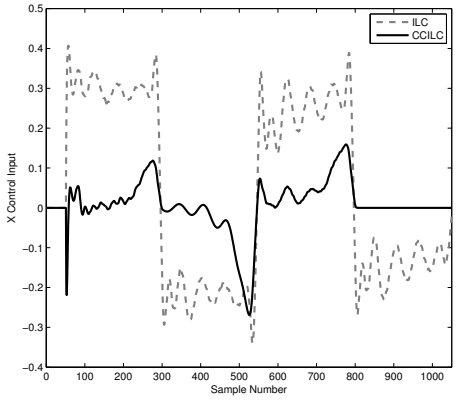


Fig. 8. *Control input signal for the x-axis [Experimental] Similar results for the y-axis*

Comparing the RMS converged error values for individual axis and contour tracking, the trade-off between individual axis tracking and contour tracking matches the trends shown in Fig. 4 and Fig. 5. N.O.-CCILC results in a 27% improvement in the converged RMS contour error versus N.O.-ILC. This improvement results from decoupling time and position. This decoupling effect shows up in the converged individual axis RMS errors, in the form of higher RMS errors for the N.O.-CCILC controller, and in the individual axis control signals, in the form of lower input signals for the N.O.-CCILC controller, as seen in Fig. 8. Although the individual axis errors may be higher in the N.O.-CCILC case, these results illustrate how the optimal contour controller yields more precise tracking of this specific trajectory without the expense of large control signals.

## VI. Conclusion

In this paper we presented N.O.-CCILC, which focuses on contour tracking for multi-axis systems. After examining the concept of contour tracking versus individual axis tracking and introducing CCILC, the N.O. framework was presented. A detailed design approach, including tuning guidelines, was provided. Using this approach, N.O.-CCILC and N.O.-ILC controllers were designed for comparison on a multi-axis robotic testbed. The results showed that N.O.-CCILC results in an alternative technique for improving the contour tracking of multi-axis systems.

## VII. Acknowledgements

This work was supported in part by NSF IREE, an NSF Travel Grant, and the University of Illinois at Urbana-Champaign Nano-CEMMS Center NSF Award $DMI - 0328162$.

## References

[1] H. Chuang and C. Liu, "Cross-coupled adaptive feedrate control for multiaxis machine tools," *Journal of Dynamic Systems, Measurement, and Control*, pp. 451–457, 1991.

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

[3] ——, "Cross-coupled ILC for improved precision motion control," in *Proc. of IEEE American Control Conference*, 2007.

[4] ——, "A cross-coupled iterative learning control design for precision motion control," *to appear in Control Systems Technology*, vol. 000, p. 000, 2008.

[5] Y. Koren, "Cross-coupled biaxial computer control for manufacturing systems," *Journal of Dynamic Systems, Motion, and Control*, vol. 102, pp. 265–272, 1980.

[6] Y. Koren and C. Lo, "Variable-gain cross-coupling controller for contouring," *CIRP Annals*, vol. 40, pp. 371–374, 1991.

[7] G. Chiu and M. Tomizuka, "Contouring control of machine tool feed drive system: A task coordinate frame approach," *Transactions on Control Systems Technology*, pp. 130–139, 2001.

[8] R. Longman, "Iterative learning control and repetitive control for engineering practice," *International Journal of Control*, vol. 73, pp. 930–954, 2000.

[9] D. Bristow, M. Tharayil, and A. Alleyne, "A survey of iterative learning control," *Control Systems Magazine*, vol. 26, pp. 96–114, 2006.

[10] J. van de Wijdeven and O. Bosgra, "Hankel iterative learning control for residual vibration suppression with MIMO flexible structure experiments," in *Proc. of IEEE American Control Conference*, 2007.

[11] D. Bristow and A. Alleyne, "A high precision motion control system with application to microscale robotic deposition," *IEEE Transactions on Control Systems Technology*, vol. 16, pp. 1008–1020, 2006.

[12] M. Phan and R. Longman, "A mathematical theory of learning control for linear discrete multivariable systems," in *Prod. of the AIAA/AAS Astrodynamics Specialist Conference*, 1988.

[13] M. Norrlöf and S. Gunnarsson, "Experimental comparison of some classical iterative learning control algorithms," *IEEE Transactions on Robotics and Automation*, vol. 18, pp. 636–641, 2002.

[14] T. Donkers, J. van de Wijdeven, and O. Bosgra, "Robustness against model uncertainties of norm optimal Iterative Learning Control," in *Proc. of the American Control Conference*, Seattle, WA, USA, June 11-13 2008, pp. 4561–4566.

[15] D. Bristow, "Frequency domain analysis and design of Iterative Learning Control for systems with stochastic disturbances," in *Proc. of the American Control Conference*, Seattle, WA USA, June 11-13 2008, pp. 3901–3907.

[16] J. Ratcliffe, P. Lewin, E. Rogers, J. Hätönen, and D. Owens, "Norm-optimal iterative learning control applied to gantry robots for automation applications," *IEEE Transactions on Robotics*, vol. 22, pp. 1303–1307, 2006.