2009 American Control Conference
Hyatt Regency Riverfront, St. Louis, MO, USA
June 10-12, 2009

Norm Optimal ILC with Time-Varying Weighting Matrices

Kira Barton and Andrew Alleyne
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@illinois.edu, alleyne@illinois.edu

Abstract— **In this paper, we focus on improving performance and robustness in precision motion control (PMC) of multi-axis systems through the use of time-varying weighting matrices. 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 time-varying weighting matrices which enable the controller to take the trajectory, position-dependent dynamics, and time-varying disturbances into consideration when designing the optimal learning controller. A general approach for designing the different weighting matrices is included. The time-varying weighting approach of this framework enables one to focus on individual components that affect the system at different times throughout the trajectory independently. The performance benefits of time-varying weighting matrices 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 multi-input multi-output (MIMO) manufacturing systems that execute the same task repetitively. In many of these systems, there are position- or time-varying dynamics that affect the performance or robustness of the system at different times throughout a single iteration. For these types of systems, it is beneficial to consider a time-varying controller which enables one to focus on the different position/time dynamics independently. Focusing on individual dynamics at different times throughout the iteration may result in a final outcome that not only improves tracking control but is more robust to time-varying disturbances.

The general norm optimal (N.O.) iterative learning control (ILC) approach for PMC on multi-axis repetitive systems has been used to implement time-invariant weighting matrices that are designed to satisfy all constraints [1]. This approach works well for systems that do not include time-varying dynamics or disturbances. However, for systems which exhibit time and/or position dependent behavior, limiting the performance of the learning controller to the worst-case dynamics may result in a more conservative controller.

In [2], [3], two novel control designs which focus on N.O. time-varying weighting matrix design were introduced. [2] presented a technique for improved contour tracking of MIMO systems which requires coordinated positioning between two or more axes. Reformatting the general N.O. framework to include weighting on the contour error results in a time-varying weighting matrix which is a function of the contour tracking and the individual axis errors. [3] seeks to maximize system performance while maintaining robust monotonic convergence by shaping the weighting matrix based on the initial tracking error.

While each of these approaches presents a motivating example for time-varying one of the weighting matrices in the cost function, this paper presents circumstances under which the performance and/or the robustness of the system would benefit by time-varying each of the weighting matrices independently. Under certain design constraints, system dynamics, and time-varying disturbances, a combination of time-varied weighting matrices may be required for any given system.

The outline of this paper is as follows. Section II motivates the use of time-varying weighting matrices for enhanced performance and robustness of a given system. The norm optimal framework with regards to iterative learning control will be discussed in Section III, including guidelines for tuning the weighting matrices and the formulation of a typical cost function. The design of time-varying weighting matrices for enhanced performance and robustness benefits using the norm optimal framework is described in Section IV and V. Results from implementation of a time-varying controller to a multi-axis robotic testbed are given in Section VI. Conclusions are given in Section VII.

II. SystemSetup

ILC has often been used to improve the tracking performance of systems that execute the same task multiple times [4]–[6]. ILC uses the tracking errors from each iteration to generate a feedforward control signal as a means of learning and compensating for repetitive disturbances, unmodelled dynamics, and tracking errors. In many manufacturing systems, the disturbances, dynamics, and tracking errors are time and position dependent. For these systems, time-varying controllers may be implemented in order to address specific challenges at specific time intervals.

In this paper, we consider MIMO systems consisting of two or more uncoupled axes (Fig. 1). The system is subject to a trajectory which includes linear and contoured trajectories, as well as low and high acceleration movements. The contoured trajectory requires coupled movements from multiple axes in order to achieve the desired trajectory (see Fig. 2).

978-1-4244-4524-0/09/$25.00 ©2009 AACC

Focusing on systems which contain multiple uncoupled axes enables one to use cross-coupled iterative learning control (CCILC) [7] to minimize contour tracking errors.

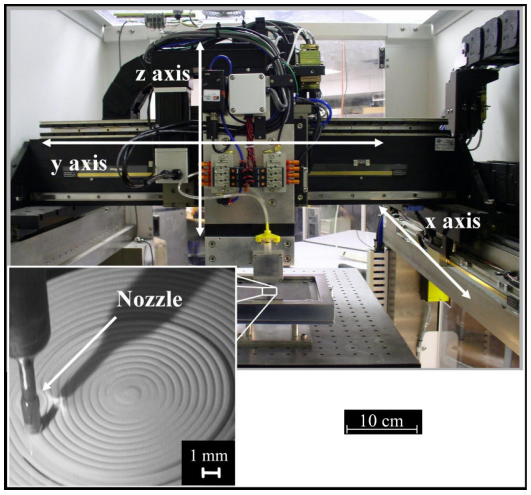

Fig. 1. *Multi-axis Robotic Testbed*

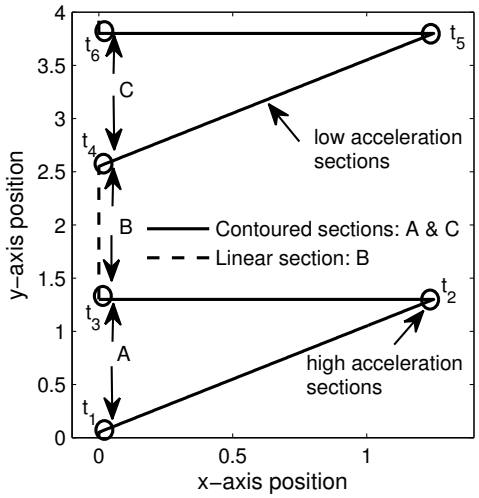

Fig. 2. *Raster trajectory containing linear, contoured, low acceleration, and high acceleration sections*

The contour tracking errors for the class of MIMO systems described above can be defined with respect to individual axis errors, e_x and e_y, and trajectory dependent gains known as coupling gains [8], $c_x(k)$ and $c_y(k)$, where k is the time interval from $k = 0, 1, \ldots, N-1$. Mathematically, this can be shown as

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

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

Linearized coupling gains have the following format

$$c_x(k) = \sin\theta(k); c_y(k) = \cos\theta(k), \quad (3)$$

where θ is defined as the instantaneous angle of the reference trajectory with respect to the x-axis of the testbed coordinate system. Note that the use of trajectory-dependent coupling gains leads to a time-varying controller.

Previous work in [2] introduced a N.O.-CCILC design which focuses on minimizing contour tracking errors. The objective of this work is to implement additional time-varying design strategies in an effort to improve contour tracking performance and robustness through the use of time-varying weighting matrices in the N.O. framework. The generalized structure for this framework, along with some guidelines for tuning the design of the controller, is given in the following section.

III. Norm Optimal ILC

ILC control design using the N.O. framework uses the lifted setting [9]. The lifted-system representation transforms the q_i-input, q_o-output, two-dimensional (time and iteration) system into an Nq_i-input, Nq_o-output, one-dimensional (iteration) system. In this setting, the discrete-time representation of a linear time invariant (LTI) system $P(k)$ is presented by its convolution matrix $\mathbf{P}$ using impulse response data $H(K)$,

$$\mathbf{P} = \begin{bmatrix} H(0) & & 0 \\ \vdots & \ddots & \\ H(N-1) & \cdots & H(0) \end{bmatrix}. \quad (4)$$

For MIMO systems, $H(K)$ represents the impulse response from each q_i input to each q_o output,

$$H(k) = \begin{bmatrix} H^{11}(k) & \cdots & H^{1q_i}(k) \\ \vdots & & \vdots \\ H^{q_o 1}(k) & \cdots & H^{q_o q_i}(k) \end{bmatrix}, \quad (5)$$

with $H^{il}(k)$ the impulse response from input l to output i.

The error at trial j is defined as, $\mathbf{e_j} = \mathbf{y_r} - \mathbf{y_j}$, where $\mathbf{y_r}$ is the reference signal and $\mathbf{y_j} = \mathbf{P}\mathbf{u_j}$, with $\mathbf{y_r}$ and $\mathbf{u_j}$ defined as,

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

$$\mathbf{u}_j = \begin{bmatrix} \mathbf{u}_j^T(0) & \mathbf{u}_j^T(1) & \cdots & \mathbf{u}_j^T(N-1) \end{bmatrix}^T, \quad (7)$$

with $\mathbf{y_r}^T(k) = \begin{bmatrix} \mathbf{y_r}^1(k) & \cdots & \mathbf{y_r}^{q_o}(k) \end{bmatrix}$

and $\mathbf{u}_j^T(k) = \begin{bmatrix} \mathbf{u}_j^1(k) & \cdots & \mathbf{u}_j^{q_i}(k) \end{bmatrix}$.

The N.O. algorithm is designed to minimize a quadratic optimization problem [1], [10], [11],

$$\mathcal{J} = \mathbf{e}_{j+1}^T \mathbf{Q} \mathbf{e}_{j+1} + \mathbf{u}_{j+1}^T \mathbf{S} \mathbf{u}_{j+1} + (\mathbf{u}_{j+1} - \mathbf{u}_j)^T \mathbf{R} (\mathbf{u}_{j+1} - \mathbf{u}_j). \quad (8)$$

where $(\mathbf{Q}, \mathbf{R}, \mathbf{S})$ are symmetric, positive definite real-valued matrices of appropriate dimension and $\mathbf{P}^T\mathbf{Q}\mathbf{P} + \mathbf{S} + \mathbf{R}$ is positive definite. For many designs, $(\mathbf{Q}, \mathbf{R}, \mathbf{S}) \equiv (q\mathbf{I}, s\mathbf{I}, r\mathbf{I})$, with q, s, r real-valued positive scalars. Applying the substitution $\mathbf{e}_{j+1} = \mathbf{e}_j - \mathbf{P}(\mathbf{u}_{j+1} - \mathbf{u}_j)$, differentiating $\mathcal{J}$ with respect to $\mathbf{u}_{j+1}$, setting the result to zero, and rearranging

the solution, yields the general N.O. controller,

$$\begin{aligned} \mathbf{u}_{j+1} &= \mathbf{L_u}\mathbf{u}_j + \mathbf{L_e}\mathbf{e}_j \\ \mathbf{L_u} &= (\mathbf{P}^T\mathbf{Q}\mathbf{P} + \mathbf{S} + \mathbf{R})^{-1}(\mathbf{P}^T\mathbf{Q}\mathbf{P} + \mathbf{R}) \\ \mathbf{L_e} &= (\mathbf{P}^T\mathbf{Q}\mathbf{P} + \mathbf{S} + \mathbf{R})^{-1}\mathbf{P}^T\mathbf{Q}. \end{aligned} \tag{9}$$

An essential part of the design process involves determining the weighting matrices $(\mathbf{Q}, \mathbf{S}, \mathbf{R})$. In order to determine which weighting matrices should be time-varied depending on performance and robustness requirements, it is necessary to understand how these matrices affect monotonic convergence, performance, robust convergence, and performance in the presence of stochastic disturbances. [2] presents some guidelines for tuning the matrices based on these criteria. A brief summary of the guidelines is listed below.

A. Tuning Guidelines

Based on the analysis of monotonic convergence, performance, robust convergence, and performance in the presence of stochastic disturbances in [2], the following tuning guidelines for norm-optimal ILC control can be used.

1) Design $\mathbf{Q}$ to correspond to the desired weighting of the error.
2) Design $\mathbf{S}$ such that the system is monotonically convergent. Start with an $\mathbf{S}$ yielding $||\mathbf{S}||_{i2} \approx 0.01||\mathbf{P}||_{i2}$, where the magnitude of $||\mathbf{P}||_{i2}$ is related to system uncertainty. Subsequently, reduce $||\mathbf{S}||_{i2}$ until the system diverges. Set $||\mathbf{S}||_{i2} = 2 \cdot ||\mathbf{S}||_{i2}^{min}$ to allow for a safety factor of 2.
3) When trial-varying disturbances are present, steady state error fluctuations will occur. Start with $||\mathbf{R}||_{i2} = 0$ and increase $||\mathbf{R}||_{i2}$ until the error fluctuations are within desired bounds, or the root mean square (RMS) error does not decrease anymore.

The next section presents a design method for determining time-varying matrices based on initial contour error signals for the trajectory in Fig. 2.

IV. Q Weighting Matrix Design

When designing the $\mathbf{Q}$ weighting matrix, it is advantageous to hold the $\mathbf{S}$ and $\mathbf{R}$ matrices constant. Using the tuning guidelines from [2], the scalar gains for the $\mathbf{S}$ and $\mathbf{R}$ weighting matrices were heuristically chosen as ($s = 1e^{-2}, r = 2e^{-2}$) to meet design specifications. Previous work [2] demonstrated a technique for reformatting the N.O. algorithm to comply with contour tracking objectives, resulting in a time-varying $\mathbf{Q}$ weighting matrix. In this section, the initial error signal resulting from implementing the reference trajectory of Fig. 2 on the multi-axis system in Fig. 1 is used to design a time-varying weighting matrix which takes into consideration linear versus contoured sections, as well as low versus high acceleration sections within the given trajectory.

A. Switching between CCILC and ILC

In order to explore the performance benefits of implementing individual axis versus cross-coupled ILC on different sections of the reference trajectory, we first extend the N.O. algorithm to include contour and individual axis errors. Using a cost function of the form (8), replace the error term, $\mathbf{e}$, with the contour error term, ε. Substituting (2) for the contour error and setting $\mathbf{Q} = a\mathbf{I}$ results in:

$$\begin{aligned} \mathcal{J} = \mathbf{e}_{j+1}^T(a \cdot \mathbf{Q}_{ccilc})\mathbf{e}_{j+1} + \mathbf{u}_{j+1}^T\mathbf{S}\mathbf{u}_{j+1} \\ +(\mathbf{u}_{j+1} - \mathbf{u}_j)^T\mathbf{R}(\mathbf{u}_{j+1} - \mathbf{u}_j) \end{aligned} \tag{10}$$

with $\mathbf{Q}_{ccilc}$ given by

$$\mathbf{Q}_{ccilc} = \begin{bmatrix} C^T(0)C(0) & & 0 \\ & \ddots & \\ 0 & & C^T(N-1)C(N-1) \end{bmatrix}, \tag{11}$$

where

$$C^T(k)C(k) = \begin{bmatrix} c_x(k)c_x(k) & -c_x(k)c_y(k) \\ -c_y(k)c_x(k) & c_y(k)c_y(k) \end{bmatrix}. \tag{12}$$

Combining (8) with $\mathbf{Q} = b\mathbf{I}$ and (10) results in a cost function capable of focusing on individual axis errors, contour errors, or a combination of the two.

$$\begin{aligned} \mathcal{J} = \mathbf{e}_{j+1}^T(aI \cdot \mathbf{Q}_{ccilc} + bI)\mathbf{e}_{j+1} + \mathbf{u}_{j+1}^T\mathbf{S}\mathbf{u}_{j+1} \\ +(\mathbf{u}_{j+1} - \mathbf{u}_j)^T\mathbf{R}(\mathbf{u}_{j+1} - \mathbf{u}_j). \end{aligned} \tag{13}$$

The gains a and b refer to the weighting gains applied to the contour or individual axis tracking, respectively. While generally constant, these gains can be varied throughout the trajectory using shaping criteria based on the reference trajectory.

Consider the trajectory in Fig. 2. N.O.-CCILC has been shown to result in the most improved contour tracking performance for rasters ([2]), such as sections A and C. The contour tracking performance comes from decoupling the position profile from the time profile. N.O.-ILC can be used to reestablish position-time synchronization by focusing on minimizing the individual axis errors during the linear sections. In order to maximize the contour tracking performance, while maintaining position/time synchronization at the start of each raster, N.O.-CCILC is implemented in sections A and C, while N.O.-ILC is used in section B.

In order to switch between the two controllers, the a and b gains alternate between values of 1 and 0. To facilitate smooth transitions between the sections, the time-varying vectors, $a(k)$ and $b(k)$ for $k = 1, \ldots, N-1$, are filtered using a lowpass Gaussian filter with a bandwidth of 15 Hz. This results in gain vectors of the form shown in Fig. 3.

While switching between ILC and CCILC tracking focuses on the differences between linear and contour trajectories, the high and low acceleration components of the reference trajectory can also be addressed using time-varying gains.

B. High acceleration versus low acceleration tracking

If we focus on section A from Fig. 2, the scalar weighting a represents constant performance weighting on contour error across the entire section. In many applications, the

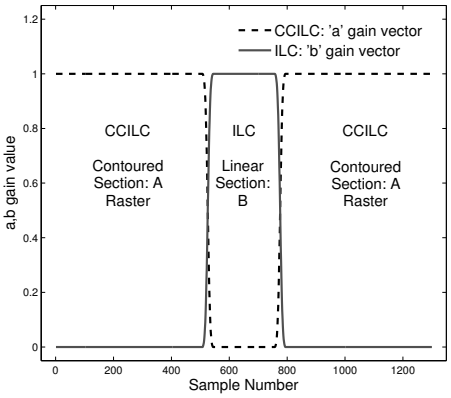

Fig. 3. *Alternating gains 'a' and 'b' to switch between CCILC and ILC*

desired trajectory does not remain uniform throughout the time interval resulting in marked differences in system performance at these locations. For example, in section A, the raster trajectory includes short durations of high acceleration content (see circled sections of Fig. 2) combined with large durations of low acceleration content. For these types of trajectories, the contour tracking performance degrades drastically in the high acceleration content locations, as illustrated in Fig. 4.

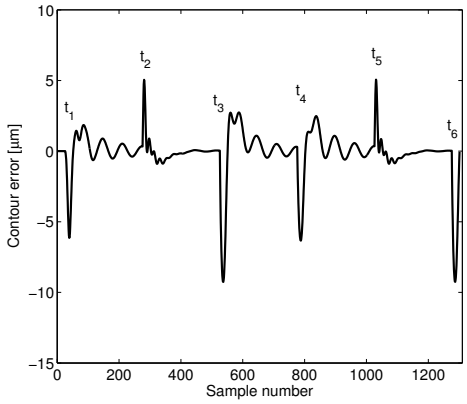

Fig. 4. *Initial contour tracking errors without the use of learning*

Figure 4 clearly indicates the locations where increased emphasis on the contour error (larger a gains) may result in better contour tracking. From this information, the locations where the a gain will be increased from 1 to 30 have been identified as t_1, t_2, t_3, t_4, t_5, and t_6. As with the switching process described in Section IV-A, the transitions between high and low gain were smoothed out using a lowpass Gaussian filter with a 15 Hz bandwidth. The modified time-varying a profile is given in Fig. 5.

The value of the gain during the high acceleration sections was chosen by increasing the value until the performance of the system either became erratic or unstable, and then reducing the gain to provide for a safety factor on the real system. For this system we chose a safety factor of 2.

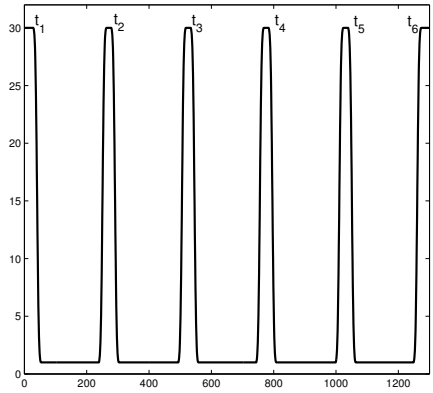

Fig. 5. *Profile for the weighting gain on contour error*

Combining the two methods results in a time-varying weighting matrix that includes switching between CCILC and ILC and incorporating high gains at locations requiring high acceleration tracking. The resulting weighting vectors for gains $a(k)$ and $b(k)$ are plotted in Fig. 6. Note the high gain section for b at the beginning and end of the switching sections. The transitions between CCILC and ILC occur at high acceleration sections of the trajectory, thereby requiring higher gain values for these locations.

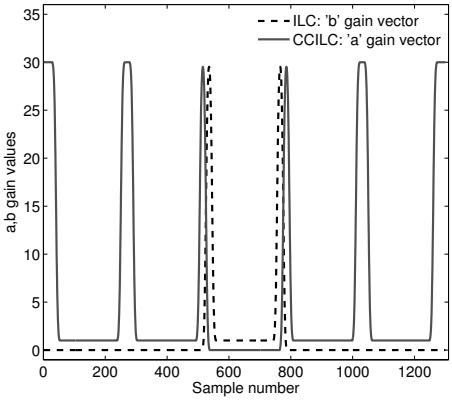

Fig. 6. *Profile for the weighting gains for a time-varying Q matrix*

C. Monotonic Convergence

An important consideration in ILC design is stability and convergence. In some cases, a stable ILC design may produce a system which exhibits large transients before convergence [6]. Therefore, this paper focuses on monotonic convergence.

Given the ILC controller (9) and the system dynamics $\mathbf{y_j} = \mathbf{P}\mathbf{u_j}$, the trial domain update law can be given by

$$\mathbf{u}_{j+1} = (\mathbf{L_u} - \mathbf{L_e}\mathbf{P})\mathbf{u}_j + \mathbf{L_e}\mathbf{y_r}. \tag{14}$$

Monotonic convergence requires the relatively well known condition $|\mathbf{L_u} - \mathbf{L_e}\mathbf{P}|_{i2} < 1$ such that $\|\mathbf{u}_{j+1}\|_2 < \|\mathbf{u}_j\|_2$. Here $\|\cdot\|_{i2}$ is defined as the largest singular value of a

given matrix. For the norm optimal controller, we have $\mathbf{L_u} - \mathbf{L_e P} = (\mathbf{P}^T\mathbf{Q}\mathbf{P} + \mathbf{S} + \mathbf{R})^{-1}\mathbf{R}$. As a result, convergence is guaranteed for any symmetric positive definite $(\mathbf{Q}, \mathbf{S}, \mathbf{R})$ with $\mathbf{P}^T\mathbf{Q}\mathbf{P} + \mathbf{S} + \mathbf{R}$ positive definite.

V. $\mathbf{S}$ and $\mathbf{R}$ Weighting Matrix Design

The previous section presented a technique for designing a time-varying weighting matrix for performance benefits. An equally important aspect in control design is ensuring robustness of the controller. This section focuses on implementing time-varying weighting matrices in order to provide robustness in the presence of position dependent dynamics and position/time dependent disturbances and noise.

A. Time-varying $\mathbf{S}$ Weighting Matrix

Using analysis provided in [2], [12] it can be shown that the $\mathbf{S}$ weighting matrix should be designed to ensure robust monotonic convergence in the presence of model uncertainty. Assuming a weighting matrix of the form $\mathbf{S} = s\mathbf{I}$, the weighting gain s provides constant weighting for uniform model uncertainty. However, in some applications, the dynamics are position dependent [13]. For applications which extend into locations with different dynamics, a time-varying weighting gain $s(k)$ enables the controller to adequately address the model uncertainty at each location.

B. Time-varying $\mathbf{R}$ Weighting Matrix

In this subsection, we consider performance in the presence of stochastic disturbances. As is shown in [3], the influence of stochastic disturbances can be reduced by reducing the convergence speed. In [2] the dominating factor in convergence speed was shown to be the $\mathbf{R}$ weighting matrix. While a constant weighting gain r in $\mathbf{R} = r\mathbf{I}$ provides consistent influence on the effect of stochastic disturbances, many applications include disturbances that change depending on time or position. For these cases, designing a time-varying weighting gain $r(k)$ results in a more robust controller that is capable of handling different types of disturbances without being overly conservative or becoming unstable.

The next section presents some results from implementing time-varying N.O. controllers on a multi-axis robotic testbed.

VI. Results

A. Experimental Setup

In this section we apply the methodology presented in Section IV to design a time-varying N.O. controller for 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 [14]. Numerical values identified for the plant models along with controller coefficients can be found in the Appendix.

$$P_i(z) = \frac{K(z+\alpha_{i1})(z^2-\alpha_{i2}z+\alpha_{i3})(z^2-\alpha_{i4}z+\alpha_{i5})}{(z-\beta_{i1})(z-1)(z^2-\beta_{i2}z+\beta_{i3})(z^2-\beta_{i4}z+\beta_{i5})}. \quad (15)$$

$$k_{pi}(z) = \frac{k(z-\alpha_{i1})(z-\alpha_{i2})(z-\alpha_{i3})}{(z-\beta_{i1})(z-\beta_{i2})(z-\beta_{i3})}, i = x, y. \quad (16)$$

The reference signal applied to the system is the combined raster scanning trajectory of Fig. 2 ($N = 1300$), in which the motion consists of linear and contour sections, as well as long periods of low acceleration content followed by short periods of high acceleration 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. Sections A and C of Fig 2 correspond to locations where a N.O.-CCILC design focuses on minimizing contour tracking, while in section B N.O.-ILC is used to improve individual axis tracking and reestablish position/time synchronization. The high acceleration transition points, identified using circles on Fig. 2, correspond to locations within the trajectory where the desired trajectory results in increased contour tracking errors. These areas indicate potential opportunities for large weighting gains to provide improved tracking capabilities as compared to small weighting gains with respect to contour tracking.

B. Simulation Results

The following results were obtained using the stabilized dynamic models from (15) and (16), the reference trajectory from Fig 2, and the N.O. controllers (9).

Since [2] demonstrates the contour tracking performance benefits of N.O.-CCILC over N.O.-ILC, this paper focuses on comparing the modified N.O. controller with time-varying weighting matrices to the N.O.-CCILC design. Figure 7 illustrates the effect of using time-varying weighting gains, $a(k)$ and $b(k)$, on the RMS contour error, as compared to basic feedback control and N.O.-CCILC. Figure 8 shows the improvement in the contour tracking at the corners as a result of high weighting gains at these particular locations. Figure 9 demonstrates the improvement in the individual y-axis tracking, and therefore enhanced position/time synchronization, at the locations where the controller switches between CCILC and ILC.

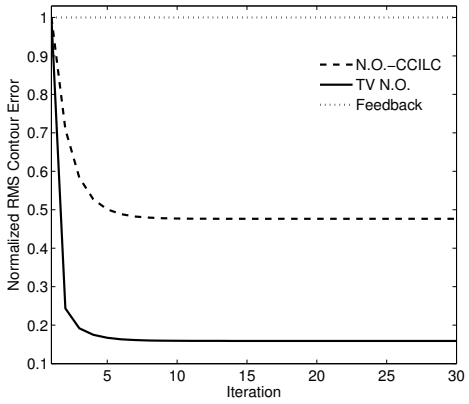

Fig. 7. *Comparison of RMS contour errors for feedback, N.O.-CCILC and TV N.O. control [Simulation]*

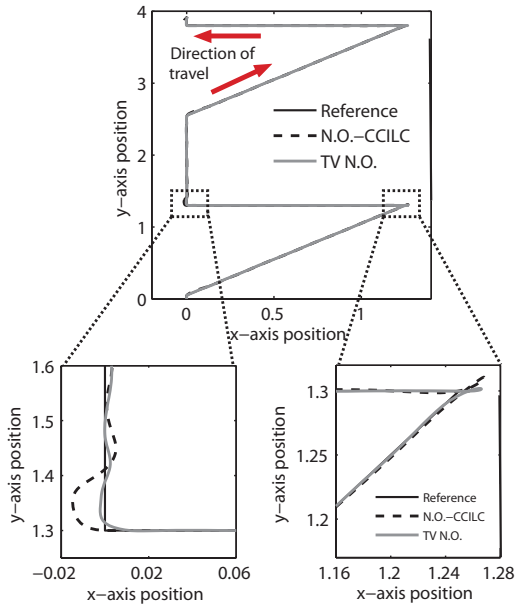

Fig. 8. *Trajectory tracking comparison of N.O.-CCILC and TV N.O. controllers. Notice that the time-varying controller produces tighter tolerances around the corners as a result of increasing the gain at specific locations [Simulation].*

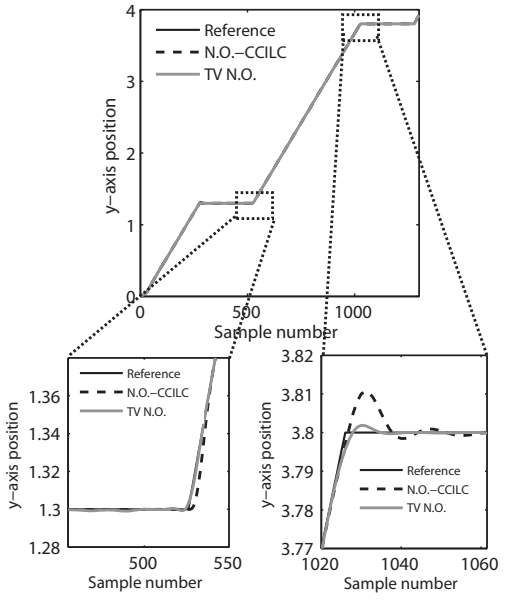

Fig. 9. *Tracking performance for the y-axis. Notice the reduction in the error resulting from the switch from CCILC to ILC [Simulation].*

Figures 7-9 clearly indicate the performance improvements obtained by implementing the modified N.O. controller with a time-varying $\mathbf{Q}$ weighting matrix in simulation. The tracking improvements from experimental testing are given in the next section.

C. Experimental Results

In order to validate the simulation results, N.O.-CCILC and TV N.O. controllers were implemented on the experimental testbed from Fig. 1. Analogous to the simulation results, the TV N.O. controller results in the most improved contour tracking performance as illustrated in Fig. 10 and Fig. 11.

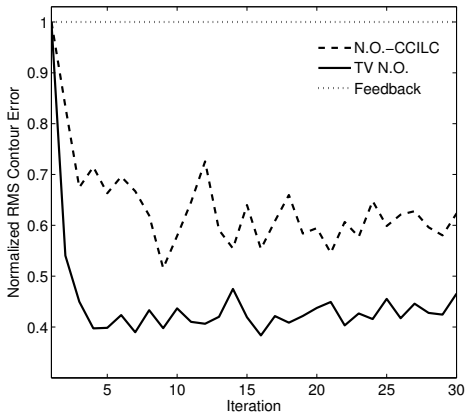

Fig. 10. *Comparison of RMS contour errors for Feedback, N.O.-CCILC and TV N.O. control [Experimental]*

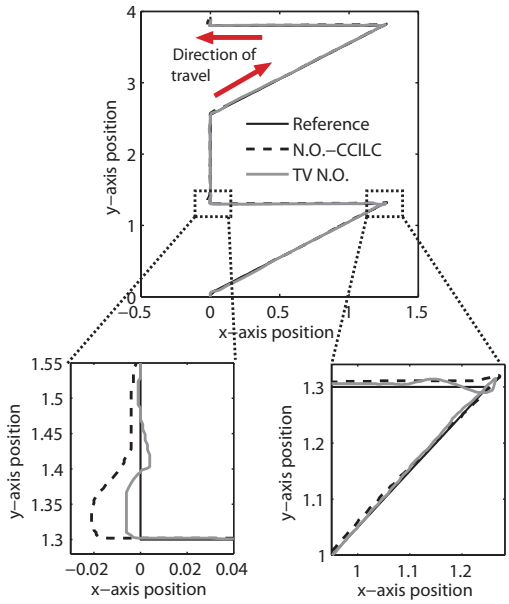

Fig. 11. *Trajectory tracking comparison of N.O.-CCILC and TV N.O. controllers. Notice that the time-varying controller produces tighter tolerances around the corners as a result of increasing the gain at specific locations. [Experimental]*

In Fig. 10, TV N.O. control produces the lowest normalized RSM contour tracking errors as compared to N.O.-CCILC and Feedback control with a 32% reduction from N.O.-CCILC to TV N.O. control. The contour tracking performance improvements resulting from this reduction in

RMS contour error can be see in Fig. 11. These results indicate how the addition of a time-varying $\mathbf{Q}$ weighting matrix in the N.O. learning control design results in more precise contour tracking for this particular trajectory.

VII. Conclusion

In this paper we discussed a time-varying norm optimal controller (TV N.O.), which uses iterative learning to focus on contour tracking for multi-axis systems. Time-varying weighting matrices provide a means for improving both performance and robustness of a given system.

After introducing the N.O. framework and presenting a short description of tuning guidelines for weighting matrix design, a detailed design approach for a time-varying $\mathbf{Q}$ weighting matrix was provided. Time-varying $\mathbf{S}$ and $\mathbf{R}$ weighting matrices, which focus on robustness issues, were also presented briefly. Using the design approach detailed in the paper, N.O.-CCILC and TV N.O. controllers were designed for comparison on a multi-axis robotic testbed. Simulation and experimental results showed that TV N.O. is an alternative technique for improving the contour tracking of multi-axis systems. Future work will focus on implementing a TV N.O. controller which includes time-varying $\mathbf{S}$ and $\mathbf{R}$ weighting matrices.

Appendix: Coefficients for the Plant and Controller Models

$$\begin{bmatrix} Symbol & & & Quantity & & \\ Num & \alpha_1 & \alpha_2 & \alpha_3 & \alpha_4 & \alpha_5 \\ P_x & 0.759 & 1.706 & 0.9596 & 0.0324 & 0.8968 \\ P_y & 0.9963 & 1.768 & 0.9567 & 0.2238 & 0.7933 \\ Den & \beta_1 & \beta_2 & \beta_3 & \beta_4 & \beta_5 \\ P_x & 0.9972 & 1.676 & 0.9479 & 0.3736 & 0.4904 \\ P_y & 0.9972 & 1.764 & 0.9562 & 0.1784 & 0.7898 \\ Gain & K & & & & \\ P_x & 0.0172 & & & & \\ P_y & 0.0459 & & & & \end{bmatrix} \quad (17)$$

$$\begin{bmatrix} Symbol & & Quantity & \\ Num & \alpha_1 & \alpha_2 & \alpha_3 \\ k_{px} & 1.92 & 0.8881 & 0.8583 \\ k_{py} & 1.377 & 0.9147 & 0.776 \\ Den & \beta_1 & \beta_2 & \beta_3 \\ k_{px} & 1.001 & 0.5182 & 0.1691 \\ k_{py} & 1.001 & 0.5182 & 0.1691 \\ Gain & K & & \\ k_{px} & 3.5 & & \\ k_{py} & 1.5 & & \end{bmatrix} \quad (18)$$

References

[1] M. Norrlöf and S. Gunnarsson, "On the design of ilc algorithms using optimization," *Automatica*, vol. 37, pp. 2011–2016, 2001.

[2] K. Barton, J. van de Wijdeven, A. Alleyne, M. Steinbuch, and O. Bosgra, "Norm optimal cross-coupled iterative learning control," in *Proc. of IEEE Conference on Decision and Control*, 2008.

[3] D. Bristow, "Weighting matrix design for robust monotonic convergence in norm optimal iterative learning control," in *Proc. of IEEE American Control Conference*, 2008.

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

[5] K. L. Moore, *Iterative Learning Control for Deterministic Systems*. Springer-Verlag, 1993.

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

[7] K. Barton and A. Alleyne, "A cross-coupled iterative learning control design for precision motion control," *to appear in Control Systems Technology*, vol. 000, p. 000, 2008.

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

[9] 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.

[10] N. Amann, D. Owens, and E. Rogers, "Iterative learning control for discrete-time systems with exponential rate of convergence," *IEE Proceedings: Control Theory and Applications*, vol. 143, pp. 217–224, 1996.

[11] J. Lee, K. Lee, and W. Kim, "Model-based iterative learing control with a quadratic criterion for time-varying linear systems," *Automatica*, vol. 36, pp. 641–657, 2000.

[12] 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.

[13] B. Hennen, I. Rotariu, and M. Steinbuch, "Time-frequency analysis of position-dependent dynamics in a motion system with ilc," in *Proc. of the American Control Conference*, Seattle, WA, USA, July 11-13 2007.

[14] 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.