Robust Iterative Learning Control Design: Application to a Robot Manipulator

A. Tayebi, S. Abdul, M. B. Zaremba, and Y. Ye

***Abstract*—This paper deals with robust iterative learning control design for uncertain single-input–single-output linear time-invariant systems. The design procedure is based upon solving the robust performance condition using the Youla parameterization and the μ-synthesis approachto obtain a feedback controller. Thereafter, a convergent iterative learning law is obtained by using the performance weighting function involved in the robust performance condition. Experimental results, on a CRS465 robot manipulator, are provided to illustrate the effectiveness of the proposed design method.**

***Index Terms*—Iterative learning control (ILC), robot manipulators, robust performance.**

I. Introduction

For many mechanical components in mechatronic systems and robotics, the motions are repeatable. Many industrial control applications, especially in robotics, use simple linear PID controllers that achieve reasonable performance if the repeatable task is relatively simple. In some situations, e.g., when the reference trajectory contains high-frequency components, it is difficult to achieve a good enough tracking accuracy using standard PID controllers [9]. It is possible, however, to compensate for effects that are difficult to compensate for by classical control techniques, by including learning capabilities into the system [2], [12], [20], [27]. One solution to this problem is to incorporate the repetition property of the desired task in the design by adding a learning component to the PID controller that allows the controller to learn from the tracking errors of the previous operations in order to improve the tracking accuracy with every new operation. This technique is referred to as iterative learning control (ILC), [5], [16], [26].

ILC is a well-established technique that presents itself as the most suitable method to improve on repetitive tasks without excessive requirements on sensor-feedback quality or control-loop bandwidth. Specifically, ILC is a technique for improving tracking performance of processes, machines, equipment, or systems that execute the same trajectory, motion, or operation over and over, starting essentially from the same initial conditions each time. In ILC, refinements are made to the input signal after each trial until the desired performance level is reached. This approach is particularly suitable for robot manipulators due to the repeatable nature of the motion usually encountered in robotics applications (see, for instance, [10], [14], [19], [23]–[25], and [27]). ILC was initially developed as a feedforward action applied directly to the open-loop system [2], [3], [6]. Several closed-loop ILC schemes were later developed in order to benefit from the feedback properties in the first iteration, e.g., [1], [7], [11], [15], and [22].

Manuscript received May 17, 2007; revised March 27, 2008 and June 20, 2008. Current version published October 8, 2008. This work was supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC), in part by the Canada Foundation for Innovation (CFI), and in part by the Ontario Innovation Trust (OIT). This paper was presented in part at the IEEE Conference on Control Applications, Toronto, ON, Canada, 2005. Recommended by Technical Editor C. Mavroidis.

A. Tayebi and Y. Ye are with the Department of Electrical Engineering, Lakehead University, Thunder Bay, ON P7B 5E1, Canada (e-mail: atayebi@lakeheadu.ca).

S. Abdul was with the Department of Electrical Engineering, Lakehead University, Thunder Bay, ON P7B 5E1, Canada. He is now with the Department of Aerospace Engineering, Ryerson University, Toronto, ON M5B 2K3, Canada (e-mail: sabdul@lakeheadu.ca).

M. B. Zaremba is with the Département d'Informatique et d'Ingénierie, Université du Québec, Gatineau, QC J8X 3X7, Canada (e-mail: marek.zaremba@uqo.ca).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMECH.2008.2004627

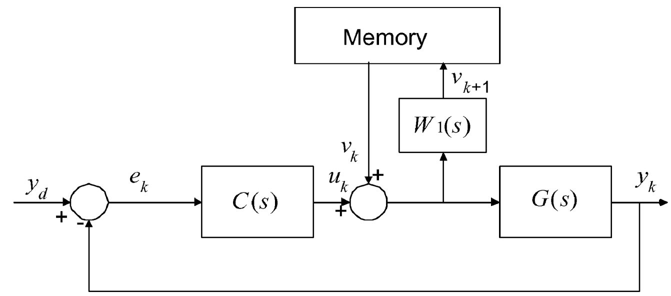

Fig. 1. ILC scheme.

Most of the ILC approaches proposed in the literature focus on the determination of the convergence conditions, which is a crucial part of the design, but there is little work that deals with solving these conditions to obtain the ILC filters especially under model uncertainties. The $\mathcal{H}_\infty$ and μ-synthesis approaches were used in [7] and [15] to design the learning filters, under model uncertainties, assuming that the feedback controller is already available. A two-step procedure based on the $\mathcal{H}_\infty$ optimization was proposed in [1] to design the feedback and learning controllers. However, as the authors pointed out in [1], this technique cannot be used for unstable systems, and the convergence condition can only be satisfied if there is no uncertainty.

The main advantage of designing the feedback and learning controllers in two separate steps is obviously the increased number of DOFs, which permits to assign the desired performance at the first iteration through the feedback controller, and the performance of the iterative process through the learning filters. The main practical drawback of this technique, besides the increased design complexity, is that it leads generally to high-order ILC filters. In fact, a feedback controller designed using robust control techniques, such as μ-synthesis, is generally of a high order; hence, robust ILC design based on the high-order feedback controller will lead to higher order ILC filters if those filters exist. From a practical point of view, it is important to keep the order of the feedback and learning filters as low as possible. One potential solution is to design the feedback and learning filters simultaneously. In fact, in [22], it is shown that if the feedback controller is designed to satisfy the robust performance condition, then the performance weighting function can be used as a learning filter guaranteeing the convergence of the iterative process. Hence, there is no need to design the learning filter if a feedback controller can be designed to satisfy the robust performance condition.

In this paper, using the Youla parameterization and the μ-synthesis approach, we provide a robust ILC design procedure that guarantees robust performance for the feedback system and the convergence of the iterative process, for both stable and unstable uncertain linear time-invariant (LTI) systems. Finally, our approach is validated experimentally on a CRS465 robot manipulator, where we show how this approach could be used in robotics applications to generate the learning component to be added to the nominal potential difference (PD) control to improve the tracking performance from operation to operation.

II. Preliminaries and Problem Formulation

Let us consider the ILC scheme shown in Fig. 1, with the following iterative rule:

$$V_{k+1}(s) = W_1(s)\,(V_k(s) + U_k(s)) \tag{1}$$

where k denotes the iteration or operation number, and $V_1(s) = 0$. The plant G is described in the following multiplicative uncertain form:

$$G = (1 + \Delta W_2)G_n \tag{2}$$

where G_n is the nominal plant model, W_2 is a known stable transfer function, and Δ is an unknown stable transfer function satisfying $\|\Delta\|_\infty \le 1$. We assume that the reference signal $y_d(t)$ is bounded and $y_k(0) = y_d(0)$, and without any loss of generality, we consider that $y_k(0) = y_d(0) = 0$.

It is shown in [22] that if the controller $C(s)$ is designed such that the robust performance condition

$$\||W_1 S| + |W_2 T|\|_\infty < 1 \tag{3}$$

is satisfied, where $S = 1/(1 + CG_n)$ is the sensitivity function, and $T = 1 - S$ is the complementary sensitivity function, then the tracking error is bounded for all $k \in \mathbb{N}$, and is uniformly $\mathcal{L}_2$-convergent to

$$e_\infty(t) = \lim_{k\to\infty} e_k(t) = \mathcal{L}^{-1}\left(\frac{1 - W_1}{1 - W_1 + CG_n(1 + \Delta W_2)} Y_d\right) \tag{4}$$

which tends to 0 if $W_1(s)$ tends to 1.

In this paper, we propose a robust ILC design procedure, based on the Youla parameterization and the μ-synthesis approach, for both stable and unstable uncertain single-input–single-output (SISO)–LTI systems. We also show the effectiveness of the proposed approach by implementing it on a CRS465 robot manipulator.

III. Robust ILC Design via μ-Synthesis

In order to handle both stable and unstable systems, we use the Youla parameterization [28] for the feedback controller $C(s)$ as

$$C(s) = \frac{X(s) + N_2(s)Q(s)}{Y(s) - N_1(s)Q(s)} \tag{5}$$

where $N_1(s)/N_2(s)$ is a coprime factorization of $G_n(s)$, with $N_1(s)$ and $N_2(s)$ being two stable rational transfer functions. The stable rational transfer functions $X(s)$ and $Y(s)$ are solutions of the Bezout identity

$$N_1(s)X(s) + N_2(s)Y(s) = 1. \tag{6}$$

As shown in [8], [18], and [29], transfer functions $N_1(s)$, $N_2(s)$, $X(s)$, and $Y(s)$ can be obtained using the following procedure.

1) Find a state-space realization $\{A, B, C, D\}$ of $G_n(s)$, i.e.,

$$G_n(s) = D + C(sI - A)^{-1}B \triangleq \left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right].$$

2) Find F such that $A + BF$ is stable. Transfer functions $N_1(s)$ and $N_2(s)$ are given by

$$N_1(s) \triangleq \left[\begin{array}{c|c} A+BF & B \\ \hline C+DF & D \end{array}\right], \quad N_2(s) \triangleq \left[\begin{array}{c|c} A+BF & B \\ \hline F & 1 \end{array}\right].$$

3) Find H such that $A + HC$ is stable. Transfer functions $X(s)$ and $Y(s)$ are given by

$$X(s) \triangleq \left[\begin{array}{c|c} A+HF & H \\ \hline F & 0 \end{array}\right]$$

$$Y(s) \triangleq \left[\begin{array}{c|c} A+HF & -B-HD \\ \hline F & 1 \end{array}\right].$$

Note that for stable systems, one can take $N_1 = G_n$, $N_2 = 1$, $X = 0$, and $Y = 1$, which leads to the internal model parameterization [17], namely $C(s) = Q(s)/[1 - G_n(s)Q(s)]$, which is a special case of the Youla parameterization.

We know that the robust performance condition (3) is equivalent to the following condition [8]:

$$\left\|\frac{W_1 S}{1 + \Delta W_2 T}\right\|_\infty < 1, \qquad \|W_2 T\|_\infty < 1.$$

Therefore, since the sensitivity and the complementary sensitivity functions, with the Youla parameterization, are given, respectively, by $S = N_2(Y - N_1 Q)$ and $T = 1 - S = N_1(X + N_2 Q)$, one can conclude that if the following conditions:

$$\left\|\frac{W_1 N_2(Y - N_1 Q)}{1 + \Delta W_2 N_1(X + N_2 Q)}\right\|_\infty < 1, \quad \|W_2 N_1(X + N_2 Q)\|_\infty < 1 \tag{7}$$

are satisfied, then the ILC scheme in Fig. 1 guarantees the boundedness of the tracking error, for all $k \in \mathbb{N}$, and its uniform $\mathcal{L}_2$-convergence to the value given in (4), when $k \to \infty$. Robust performance is also guaranteed for the feedback system.

Now, in order to design Q satisfying the robust performance condition (7), we introduce the following generalized matrix[1]:

$$M_1 = \begin{pmatrix} -W_2 N_1(X + N_2 Q) & W_2 N_1(X + N_2 Q) \\ -W_1 N_2(Y - N_1 Q) & W_1 N_2(Y - N_1 Q) \end{pmatrix} \tag{8}$$

which has the following upper linear fractional transformation (LFT):

$$\mathcal{F}_u(M_1, \Delta) = \frac{W_1 N_2(Y - N_1 Q)}{1 + \Delta W_2 N_1(X + N_2 Q)} \tag{9}$$

which is well posed if $\|W_2 N_1(X + N_2 Q)\|_\infty < 1$. The structured singular value $\mu_{\underline{\Delta}}(M_1)$ is defined as

$$\mu_{\underline{\Delta}}(M_1) = \frac{1}{\min\{\overline{\sigma}(\underline{\Delta}) : \underline{\Delta} \in \Delta_p, \det(I - M_1\underline{\Delta}) = 0\}}$$

unless no $\underline{\Delta} \in \Delta_p$ makes $I - M_1\underline{\Delta}$ singular, in which case $\mu_{\underline{\Delta}}(M_1) = 0$.

The variable $\overline{\sigma}(\underline{\Delta})$ denotes the largest singular value of $\underline{\Delta}$, and

$$\Delta_P = \left\{\begin{pmatrix} \Delta & 0 \\ 0 & \Delta_f \end{pmatrix} : \Delta \in \mathbb{C}, \Delta_f \in \mathbb{C}\right\}$$

denotes a prescribed set of structured block diagonal matrices [29].

Now, one can state the following theorem.

Theorem 1: Consider the control scheme in Fig. 1 with the controller C parameterized as in (5). If there exists Q satisfying $\sup_{\omega\in\Re}\mu_{\underline{\Delta}}(M_1(j\omega)) < 1$, then:

1) robust performance is guaranteed for the feedback system;
2) the tracking error is bounded for all $k \in \mathbb{N}$ and is uniformly $\mathcal{L}_2$-convergent to $e_\infty(t)$ given in (4), when k tends to infinity.

Proof: Straightforward from [22, Th. 1] and [30, Th. 11.8]. □

Now, for given W_1, W_2, and G_n, one can use the μ-synthesis procedure called D-K iteration [4], [29] to obtain $Q(s)$ satisfying $\sup_{\omega\in\Re}\mu_{\underline{\Delta}}(M_1(j\omega)) < 1$. To this end, we introduce the following matrix:

$$M_Q = \left(\begin{array}{cc|c} -W_2 N_1 X & W_2 N_1 X & W_2 N_2 \\ -W_1 N_2 Y & W_1 N_2 Y & -W_1 N_2 \\ \hline -N_1 & N_1 & 0 \end{array}\right) \tag{10}$$

[1] For the sake of presentation simplicity, we omitted the details related to the robust control theory such as LFT, structured singular values, and D-K iteration. For more details, the reader may refer to [29].

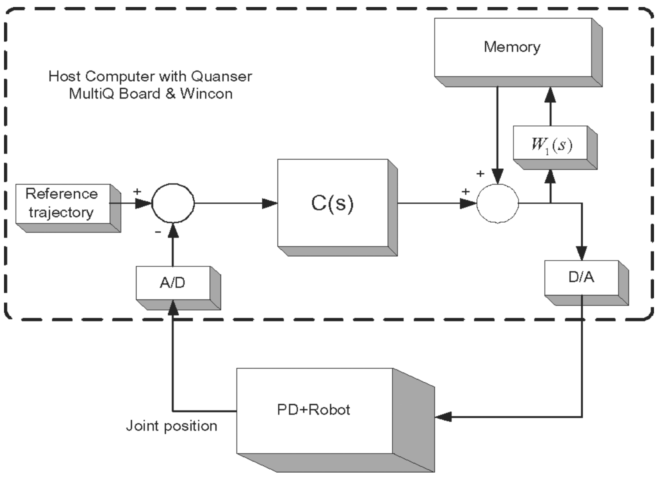

Fig. 2. Experimental setup for ILC implementation.

such that $M_1 = \mathcal{F}_l(M_Q, Q)$.

Remark 1: It is also possible to choose M_1 and M_Q as follows:

$$M_1 = \begin{pmatrix} -W_2 N_1 (X + N_2 Q) & -W_2 N_2 (X + N_2 Q) \\ W_1 N_1 (Y - N_1 Q) & W_1 N_2 (Y - N_1 Q) \end{pmatrix} \tag{11}$$

which has the following upper LFT:

$$\mathcal{F}_u(M_1, \Delta) = \frac{W_1 N_2 (Y - N_1 Q)}{1 + \Delta W_2 N_1 (X + N_2 Q)} \tag{12}$$

and

$$M_Q = \left(\begin{array}{cc|c} -W_2 N_1 X & W_2 N_2 X & W_2 N_2 \\ -W_1 N_1 Y & W_1 N_2 Y & W_1 N_1 \\ \hline -N_1 & -N_2 & 0 \end{array}\right) \tag{13}$$

such that $M_1 = \mathcal{F}_l(M_Q, Q)$.

Remark 2: Throughout the numerous tests we have performed, we have noticed that the results obtained with the generalized plant M_Q given in (10) are relatively better than those obtained with the M_Q in (13). However, whenever G_n and W_2 are strictly proper, the M_Q in (10) generally leads to a singular $\mathcal{H}_\infty$ problem that cannot be handled by the "dkit" command of the μ-synthesis toolbox [4]. To overcome this problem, we rolled off the numerators of G_n and W_2 to obtain transfer functions with zero relative degree that approximate G_n and W_2 within a desired range of frequencies.

Remark 3: In the ideal case, i.e., $W_1 = 1$, it is obvious that the tracking error converges to 0 when k tends to infinity. However, physical systems are usually strictly proper, and hence, the problem is often not solvable with $W_1 = 1$. As an alternative solution, one can take W_1 close to 1 within the tracking bandwidth in order to minimize the tracking error.

IV. EXPERIMENTAL RESULTS

Our design procedure has been tested on a 6-DOF robot manipulator CRS465. The CRS465 is an open-chain articulated robot arm with six revolute joints powered by six dc motors. The motors are equipped with incremental encoders to measure the joint positions as well as automatic brakes to prevent the collapse of the manipulator configuration when the power supply to the motors is interrupted. The robot comes with the CRS C500 controller, which contains six independent PD controllers, one for each joint. The ILC strategy proposed in this paper has been implemented using the Quanser open architecture (OA) mode. In the OA mode, all signals are routed to and from a Quanser-MultiQ data acquisition board as opposed to the CRS controller. The Quanser-MultiQ data acquisition board is used together with a Quanser WinCon software in order to generate real-time code from the Simulink model. For the real-time implementation of the control algorithm using the Quanser OA mode, WinCon software is used together with MATLAB/Simulink/Realtime Workshop, Control System Toolbox as well as Visual C++ Professional. Our ILC scheme has been implemented, as shown in Fig. 2, for the first three links of the robot manipulator. The first three links [J1(waist), J2(shoulder), and J3(elbow)] are independently controlled by a PD feedback control with the following gains $K_p = \text{diag}\{2.5, 2.5, 2.5\}$ and $K_d = \text{diag}\{0.05, 0.05, 0.05\}$. The closed-loop transfer function (represented by the block "PD+Robot" in Fig. 2) of each link has been identified using the system identification toolbox of MATLAB [13], as follows:

$$\begin{cases} \text{J1}: \quad G_1(s) = \dfrac{-0.2622s + 1624}{s^2 + 39.42s + 1761} \\ \text{J2}: \quad G_2(s) = \dfrac{-0.2101s + 1312}{s^2 + 35.15s + 1374} \\ \text{J3}: \quad G_3(s) = \dfrac{0.6385s + 1285}{s^2 + 35.74s + 1380}. \end{cases} \tag{14}$$

The filter W_1 is selected close to 1 in order to minimize the tracking error, while W_2 is selected from a rough approximation of the relative uncertainty at steady state, and the approximate frequency at which the relative uncertainty reaches 100% [21]

$$W_2(s) = \frac{\tau s + r_0}{(\tau / r_\infty)s + 1} \tag{15}$$

where $1/\tau$ gives the approximate frequency at which relative uncertainty reaches 100%, r_0 the relative uncertainty at steady state, and r_∞ the magnitude of weight at high frequency, typically a value greater than 2.

Finally, using (10) with $N_1 = G_n$, $N_2 = 1$, $X = 0$, $Y = 1$, and

$$W_1(s) = \frac{1}{0.003s + 1} \qquad W_2(s) = \frac{2 \times 10^{-3}s + 0.4}{2 \times 10^{-4}s + 1} \tag{16}$$

for link 1,

$$W_1(s) = \frac{1}{0.0033s + 1} \qquad W_2(s) = \frac{2 \times 10^{-3}s + 0.5}{2 \times 10^{-4}s + 1} \tag{17}$$

for link 2, and

$$W_1(s) = \frac{1}{0.001s + 1} \qquad W_2(s) = \frac{10^{-3}s + 0.46}{1.25 \times 10^{-3}s + 1} \tag{18}$$

for link 3, and using the μ-synthesis toolbox [4], we obtain, after model reduction, the following controllers for link 1, link 2, and link 3, respectively.

Link 1:

$$C(s) = \frac{1.0657s^2 + 3.0346 \times 10^7 s + 2.09301 \times 10^8}{s^2 + 2.5045 \times 10^5 s + 4.4749 \times 10^8}. \tag{19}$$

Link 2:

$$C(s) = \frac{-0.4550s^2 + 9.6595 \times 10^6 s + 9.9943 \times 10^7}{s^2 + 2.1621 \times 10^5 s + 2.5113 \times 10^8}. \tag{20}$$

Link 3:

$$C(s) = \frac{6.6528 \times 10^8 s + 2.6408 \times 10^{10}}{s^2 + 1.3071 \times 10^5 s + 1.8263 \times 10^{10}}. \tag{21}$$

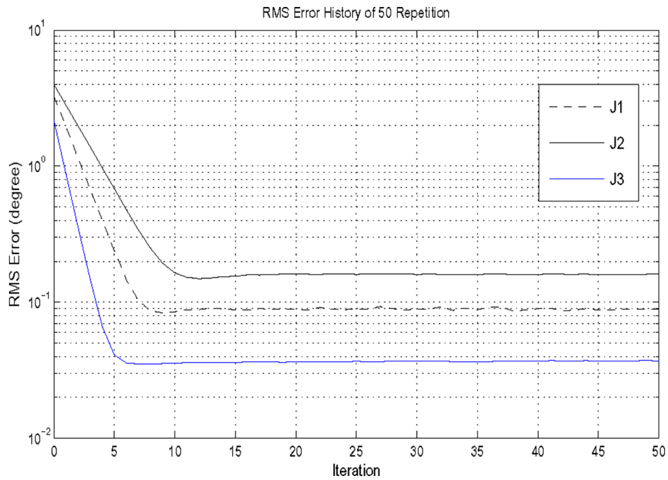

Fig. 3. RMS norm of the tracking error (in degrees) versus iteration number for the three links using (16)–(18) and the controllers (22)–(24).

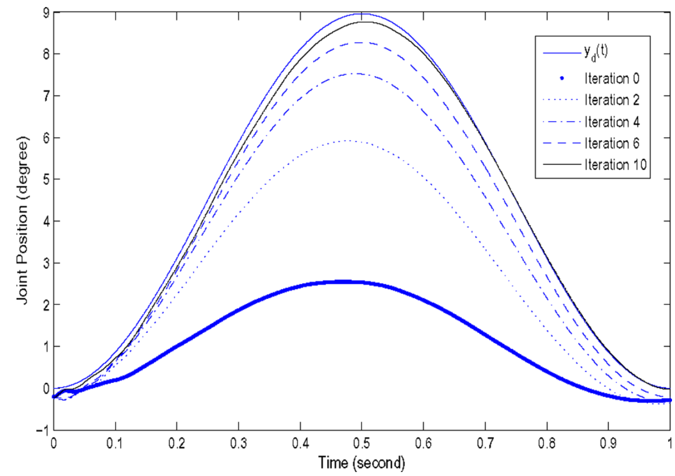

Fig. 5. Reference trajectory and actual trajectories for joint 2 (in degrees) versus time, using $W_1(s)$ given in (16) and the controller (22).

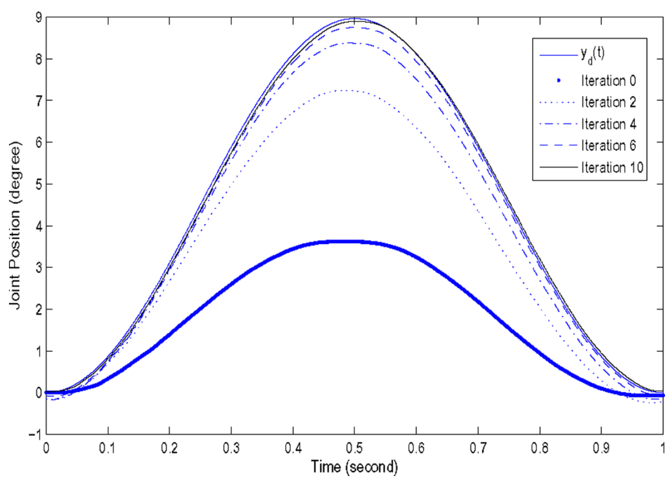

Fig. 4. Reference trajectory and actual trajectories for joint 1 (in degrees) versus time, using $W_1(s)$ given in (16) and controller (22).

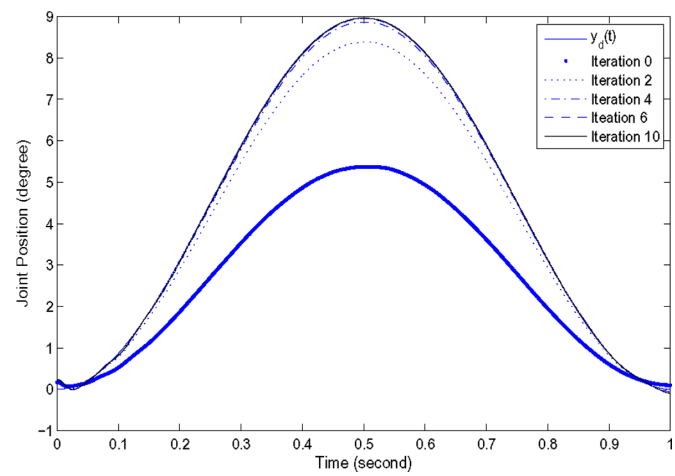

Fig. 6. Reference trajectory and actual trajectories for joint 3 (in degrees) versus time, using $W_1(s)$ given in (18) and the controller (24).

For the sake of implementation simplicity, we further reduced the controllers designed earlier to obtain simple PD controllers $C(s) = 0.0678s + 0.6548$ for link 1, $C(s) = 0.0385s + 0.3980$ for link 2, and $C(s) = 0.0364s + 1.4459$ for link 3. Finally, these controllers have been implemented using a first-order low-pass filter (with a cutoff frequency of 100 rad/s) with the derivative action as follows.

Link 1:

$$C(s) = \frac{0.0678s}{10^{-2}s + 1} + 0.6548. \qquad (22)$$

Link 2:

$$C(s) = \frac{0.0385s}{10^{-2}s + 1} + 0.3980. \qquad (23)$$

Link 3:

$$C(s) = \frac{0.0364s}{10^{-2}s + 1} + 1.4459. \qquad (24)$$

Using (16)–(18) and the controllers (22)–(24), the robust performance condition is satisfied for $\omega < 55$ rad/s and $\omega > 400$ rad/s. Using $W_1(s) = 1$ for the three links, and using the controllers (22)–(24), the robust performance condition is satisfied for $\omega < 55$ rad/s.

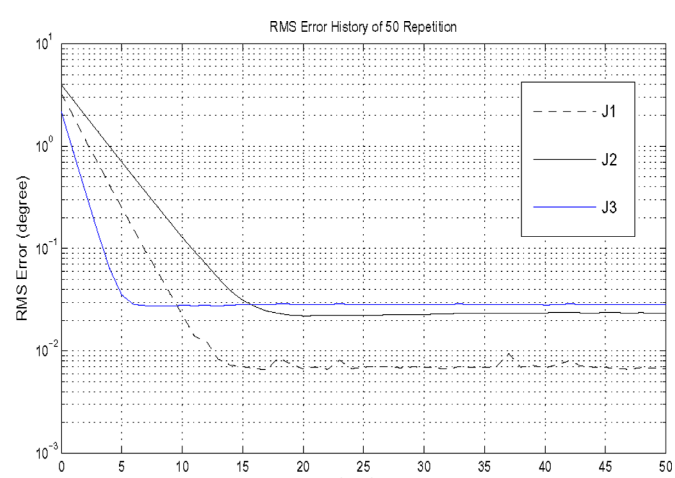

Fig. 7. RMS norm of the tracking error (in degrees) versus iteration number for the three links using $W_1(s) = 1$ and the controllers (22)–(24).

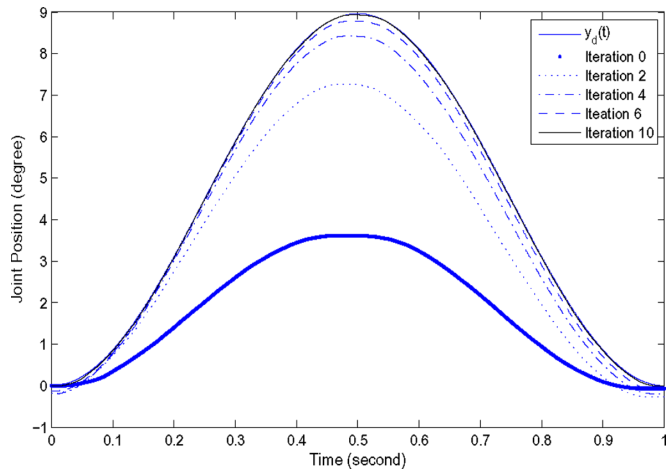

Fig. 8. Reference trajectory and actual trajectories for joint 1 (in degrees) versus time, using $W_1(s) = 1$ and the controller (22).

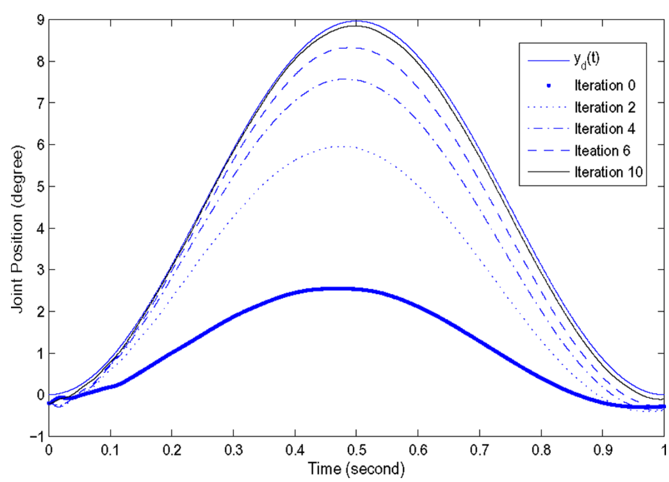

Fig. 9. Reference trajectory and actual trajectories for joint 2 (in degrees) versus time, using $W_1(s) = 1$ and the controller (22).

The desired trajectory (in degrees) for all the joints is shown in Fig. 4, and is given by

$$y_d(t) = \sum_{n=1}^{6} 2400 e^{-2(n-1)\pi} [1 - \cos(2(n-1)\pi t)]. \quad (25)$$

To avoid the problem of noise accumulation from iteration to iteration, a cutoff realized by a zero-phase filter discrete Fourier transform (DFT)/inverse DFT (IDFT) has been imposed on v_{k+1} before using the signal in the next trial. The cutoff frequencies are 10 Hz for joint 1, 13 Hz for joint 2, and 15 Hz for joint 3. The experiment was conducted with a sampling period of 1 ms. We performed 51 iterations over a time interval of 1 s for each iteration. The rms norm of the tracking error (in degrees) versus iteration number for the three links using (16)–(18) and the controllers (22)–(24) is shown in Fig. 7. The reference trajectory and the actual trajectories for joints 1, 2 and 3 (in degrees), at different iterations, versus time, using $W_1(s)$ given in (16)–(18) and the controllers (22)–(24) are shown, respectively, in Figs. 4–6. The rms norm of the tracking error (in degrees) versus iteration number for the three links using $W_1(s) = 1$ and the controllers (22)–(24) is shown in Fig. 7. The reference trajectory and the actual trajectories for joints 1, 2, and 3 (in degrees), at different iterations, versus time, using $W_1(s) = 1$ and the controllers (22)–(24) are shown, respectively, in Figs. 8–10. The experimental results confirmed the theoretical results stating that the best performance in terms of convergence is obtained with W_1 close to 1. From Fig. 7, one can see that the rms error has been, roughly, reduced by a factor of 375 for the first link after 15 iterations, a factor of 160 for the second link after 18 iterations, and a factor of 72 for the third link after 6 iterations.

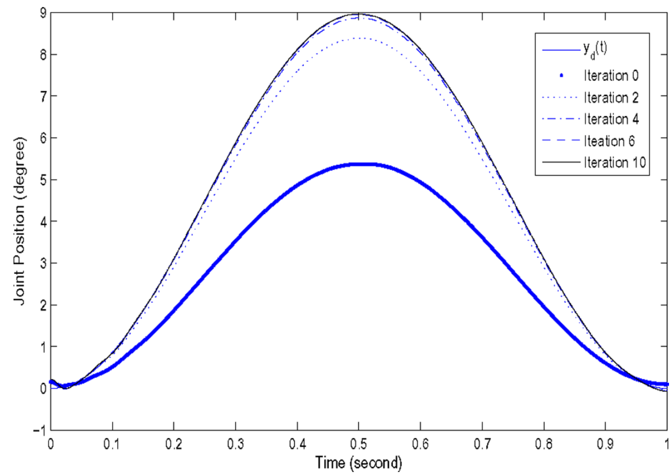

Fig. 10. Reference trajectory and actual trajectories for joint 3 (in degrees) versus time, using $W_1(s) = 1$ and the controller (24).

V. Conclusion

A robust ILC design procedure, based on the Youla parameterization and the μ-synthesis approach, for both stable and unstable uncertain LTI systems, has been proposed. Owing to the fact that the convergence of the proposed ILC scheme is guaranteed under the robust performance condition, we show that it is possible to design a single filter $Q(s)$ that ensures, simultaneously, robust performance for the feedback system and the convergence of the iterative process. Since the best ILC performance one can achieve is obtained with $W_1 = 1$, one can take W_1 as close as possible to 1 within the tracking bandwidth and solve the robust performance condition, using the wide range of tools from the robust control theory, to obtain $Q(s)$. One of the possible tools is the μ-synthesis approach, which is used in this paper. It is worth noting that our approach involves a certain tradeoff between the performance of the feedback system at the first iteration and the performance of the iterative process. In fact, the controller $C(s)$ obtained with W_1 close to 1 leads to the best ILC performance, but does not necessarily lead to the best feedback performance at the first iteration.

In this paper, we dealt with continuous-time systems, but the proposed design procedure is the same for discrete-time systems except that the unit circle should be used rather than the imaginary axis for norm computation (Matlab μ-analysis and synthesis toolbox deals with both continuous-time and discrete-time systems).

Since our theoretical results do not take into account the measurement noise and its accumulation from iteration to iteration, in our experiment on a CRS465 robot manipulator, we had to clean up the iterative signals at the end of each iteration before calculating the updated control input for the next iteration. We used a zero-phase offline filtering procedure (DFT/IDFT) so that no phase shift is introduced on the filtered signals. The obtained experimental results are satisfactory and conform to the theory, and show that the proposed control scheme—although designed for SISO-LTI systems—could handle, to a certain extent, coupled multiple-input–multiple-output (MIMO) nonlinear systems such as robot manipulators.

References

[1] N. Amman, D. H. Owens, E. Rogers, and A. Wahl, "An $\mathcal{H}_\infty$ approach to linear iterative learning control design," *Int. J. Adapt. Control Signal Process.*, vol. 10, pp. 767–781, 1996.

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

[3] S. Arimoto and T. Naniwa, "Equivalence relations between learnability, output-dissipativity and strict positive realness," *Int. J. Control*, vol. 73, no. 10, pp. 824–831, 2000.

[4] G. J. Balas, J. C. Doyle, K. Glover, A. Packard, and R. Smith, *μ-Analysis and Synthesis Toolbox*. Natick, MA: The MathWorks, Inc., 1998.

[5] Z. Bien and J.-X. Xu, *Iterative Learning Control: Analysis, Design, Integration and Applications*. Norwell, MA: Kluwer, 1998.

[6] Y. Chen, C. Wen, Z. Gong, and M. Sun, "An iterative learning controller with initial state learning," *IEEE Trans. Autom. Control*, vol. 44, no. 2, pp. 371–376, Feb. 1999.

[7] D. de Roover, "Synthesis of a robust iterative learning controller using an H_∞ approach," in *Proc. 35th Conf. Decis. Control*, Kobe, Japan, 1996, pp. 3044–3049.

[8] J. Doyle, B. Francis, and A. Tannenbaum, *Feedback Control Theory*. New York: Macmillan, 1990.

[9] V. Duchaine, S. Bouchard, and C. M. Gosselin, "Computationally efficient predictive robot control," *IEEE/ASME Trans. Mechatronics*, vol. 12, no. 5, pp. 570–578, Oct. 2007.

[10] K. Hamamoto and T. Sugie, "Iterative learning control for robot manipulators using the finite dimensional input subspace," *IEEE Trans. Robot. Autom.*, vol. 18, no. 4, pp. 632–635, Aug. 2002.

[11] T. Y. Kuc, J. S. Lee, and K. Nam, "An iterative learning control theory for a class of nonlinear dynamic systems," *Automatica*, vol. 28, no. 6, pp. 1215–1221, 1992.

[12] T. H. Lee, K. K. Tan, S. Y. Lim, and H. F. Dou, "Iterative learning control of permanent magnet linear motor with relay automatic tuning," *Mechatronics*, vol. 10, pp. 169–190, 2000.

[13] L. Ljung, *System Identification Toolbox*. Natick, MA: The MathWorks, Inc., 2001.

[14] R. W. Longman, "Iterative learning control and repetitive control for engineering practice," *Int. J. Control*, vol. 73, no. 10, pp. 930–954, 2000.

[15] J. H. Moon, T. Y. Doh, and M. J. Chung, "A robust approach to iterative learning control design for uncertain systems," *Automatica*, vol. 34, no. 8, pp. 1001–1004, 1998.

[16] K. L. Moore, "Iterative learning control: An expository overview," *Appl. Comput. Controls, Signal Process., Circuits*, vol. 1, pp. 151–214, 1999.

[17] M. Morari and E. Zafriou, *Robust Process Control*. Englewood Cliffs, NJ: Prentice-Hall, 1989.

[18] C. N. Nett, C. A. Jacobson, and M. J. Balas, "A connection between state space and doubly coprime fractional representations," *IEEE Trans. Autom. Control*, vol. 29, no. 9, pp. 831–832, Sep. 1984.

[19] M. Norrlöf and S. Gunnarsson, "Experimental comparison of some classical iterative learning control algorithms," *IEEE Trans. Robot. Autom.*, vol. 18, no. 4, pp. 636–641, Aug. 2002.

[20] J. Park, P.-H. Chang, H.-S. Park, and E. Lee, "Design of learning input shaping technique for residual vibration suppression in an industrial robot," *IEEE/ASME Trans. Mechatronics*, vol. 11, no. 1, pp. 55–65, Feb. 2006.

[21] S. Skogestad and I. Postlewaite, *Multivariable Feedback Control*. New York: Wiley, 1996.

[22] A. Tayebi and M. B. Zaremba, "Robust iterative learning control design is straightforward for uncertain LTI systems satisfying the robust performance condition," *IEEE Trans. Autom. Control*, vol. 48, no. 1, pp. 101–106, Jan. 2003.

[23] A. Tayebi, "Adaptive iterative learning control for robot manipulators," *Automatica*, vol. 40, no. 7, pp. 1195–1203, 2004.

[24] A. Tayebi and S. Islam, "Adaptive iterative learning control for robot manipulators: Experimental results," *Control Eng. Practice*, vol. 14, no. 7, pp. 843–851, 2006.

[25] D. Wang and Y. Ye, "Design and experiments of anticipatory learning control: Frequency domain approach," *IEEE/ASME Trans. Mechatronics*, vol. 10, no. 3, pp. 305–313, Jun. 2005.

[26] J.-X. Xu and Y. Tan, *Linear and Nonlinear Iterative Learning Control*. (Lecture Notes in Control and Information Sciences). New York: Springer-Verlag, 2003.

[27] Y. Ye and D. Wang, "DCT basis function learning control," *IEEE/ASME Trans. Mechatronics*, vol. 10, no. 4, pp. 449–454, Aug. 2005.

[28] D. C. Youla, H. A. Jabr, and J. J. Bongiorno, "Modern Wiener–Hopf design of optimal controllers—Part II: The multivariable case," *IEEE Trans. Autom. Control*, vol. AC-21, no. 3, pp. 319–338, Jun. 1976.

[29] K. Zhou, J. C. Doyle, and K. Glover, *Robust and Optimal Control*. Englewood Cliffs, NJ: Prentice-Hall, 1996.