Available online at www.sciencedirect.com
SCIENCE DIRECT
Control Engineering Practice 14 (2006) 843–851
CONTROL ENGINEERING PRACTICE
www.elsevier.com/locate/conengprac

Adaptive iterative learning control for robot manipulators: Experimental results ☆

A. Tayebi[a,*], S. Islam[b,1]

[a]Department of Electrical Engineering, Lakehead University, Thunder Bay, ont., Canada, P7B 5E1
[b]Department of Systems and Computer Engineering, Carleton University, Ottawa, ont., Canada, K1S 5B6

Received 30 October 2003; accepted 15 April 2005
Available online 22 June 2005

Abstract

In this paper, two adaptive iterative learning control schemes, proposed by A. Tayebi [2004, *Automatica*, *40*(7), 1195–1203], are tested experimentally on a five-degrees-of-freedom (5-DOF) robot manipulator CATALYST5. The control strategy consists of using a classical PD feedback structure plus an additional iteratively updated term designed to cope with the unknown parameters and disturbances. The control implementation is very simple in the sense that the knowledge of the robot parameters is not needed, and the only requirement on the PD and learning gains is the positive definiteness condition. Furthermore, in contrast with classical ILC schemes where the number of iterative variables is generally equal to the number of control inputs, the adaptive control schemes tested in this paper involve just one or two iterative variables.

Keywords: Iterative learning control; Robot manipulators

1. Introduction

It is well known that robot manipulators are generally used in repetitive tasks (e.g., automotive manufacturing industries). Therefore, it is interesting to take advantage of the fact that the reference trajectory is repeated over a given operation time. In this context, iterative learning control (ILC) techniques can be applied in order to enhance the tracking performance from operation to operation. Since the early works of Arimoto et al. (1984), Casalino and Bartolini (1984) and Craig (1984), several ILC schemes for robot manipulators have been proposed in the literature (see for instance Arimoto, 1996; Bondi et al., 1988; Luca et al., 1992; Horowitz, 1993; Kavli, 1992; Kawamura et al., 1988; Moon et al., 1997). These ILC algorithms, whether developed for the linearized model or the nonlinear model, are generally based upon the contraction mapping approach and require a certain a priori knowledge of the system dynamics.

On the other hand, another type of ILC algorithms have been developed using Lyapunov and Lyapunov-like methods. In fact in French and Rogers (2000), a standard Lyapunov design is used to solve ILC problems. The idea consists to use a standard adaptive controller and to start the parameter estimates with their final values obtained at the preceding iteration. In the same spirit, Choi and Lee (2000) proposed an adaptive ILC for uncertain robot manipulators, where the uncertain parameters are estimated along the time horizon whereas the repetitive disturbances are compensated along the iteration horizon. However, as in standard adaptive control design, this technique requires the unknown system parameters to be constant. In Ham et al. (1995), Ham et al. (2000), Kuc et al.

☆This work was supported by the Natural Sciences and Engineering Research Council (NSERC) of Canada, the Canada Foundation for Innovation (CFI), the Ontario Innovation Trust (OIT).
*Corresponding author. Tel.: +1 807 3438597; fax: +1 807 3438928.
E-mail address: tayebi@ieee.org (A. Tayebi).
[1]The second author was with the Department of Electrical Engineering, Lakehead University, when this work was carried out.

0967-0661/$ - see front matter © 2005 Elsevier Ltd. All rights reserved.
doi:10.1016/j.conengprac.2005.04.013

(1991), Xu (2002), Xu et al. (2000) and Xu and Tan (2001), several ILC algorithms have been proposed based upon the use of a positive-definite Lyapunov-like sequence which is made monotonically decreasing along the iteration axis via a suitable choice of the control input. In contrast with the standard adaptive control, this technique is shown to be able to handle systems with time-varying parameters since the adaptation law in this case is nothing else but a discrete integration along the iteration axis. Based on this approach, Kuc et al. (1991) proposed an ILC scheme for the linearized robot manipulator model, while in (Ham et al., 2000; Xu et al., 2000) nonlinear ILC schemes have been proposed for the nonlinear model. Again these control laws require a certain a priori knowledge of the system dynamics.

In Tayebi (2004), a simple ILC scheme, for the position tracking problem of rigid robot manipulators without any a priori knowledge on the system parameters, has been proposed. The control strategy consists of a PD term plus an additional iterative term introduced to cope with the unknown parameters and disturbances. The proof of convergence is based upon the use of a Lyapunov-like positive definite sequence, which is made monotonically decreasing through an adequate choice of the control law and the iterative adaptation rule. In contrast with classical ILC schemes where the number of iterative variables is generally equal to the number of control inputs, the proposed control strategy uses one or two iterative variables, which is interesting from a practical point of view since it contributes considerably to memory space saving. In this framework, the acceleration measurements and the bounds of the robot parameters are not needed and the only requirement on the control gains is the positive definiteness condition.

In this paper, we present some experimental results on a 5-DOF robot manipulator CATALYST5, confirming the effectiveness of the control strategy proposed in Tayebi (2004).

2. Equations of motion and problem statement

Using the Lagrangian formulation, the equations of motion of a n degrees-of-freedom rigid manipulator may be expressed by

$$M(q_k)\ddot{q}_k + C(q_k,\dot{q}_k)\dot{q}_k + G(q_k) = \tau_k(t) + d_k(t), \tag{1}$$

where $t \in \mathbb{R}_+$ denotes the time and the non-negative integer $k \in \mathbb{Z}_+$ denotes the operation or iteration number. The signals $q_k \in \mathbb{R}^n$, $\dot{q}_k \in \mathbb{R}^n$ and $\ddot{q}_k \in \mathbb{R}^n$ are the joint position, joint velocity and joint acceleration vectors, respectively, at the iteration k. $M(q_k) \in \mathbb{R}^{n\times n}$ is the inertia matrix, $C(q_k,\dot{q}_k)\dot{q}_k \in \mathbb{R}^n$ is a vector resulting from Coriolis and centrifugal forces. $G(q_k) \in \mathbb{R}^n$ is the vector resulting from the gravitational forces. $\tau_k \in \mathbb{R}^n$ is the control input vector containing the torques and forces to be applied at each joint. $d_k(t) \in \mathbb{R}^n$ is the vector containing the unmodeled dynamics and other unknown external disturbances.

Assuming that the joint positions and the joint velocities are available for feedback, our objective is to design a bounded control law $\tau_k(t)$ guaranteeing the boundedness of $q_k(t)$, $\forall t \in [0,T]$ and $\forall k \in \mathbb{Z}_+$, and the convergence of $q_k(t)$ to the desired reference trajectory $q_d(t)$ for all $t \in [0,T]$ when k tends to infinity. Throughout this paper, we will use the $\mathscr{L}_{pe}$ norm defined as follows:

$$\|x(t)\|_{pe} \triangleq \begin{cases} \left(\int_0^t \|x(\tau)\|^p \, \mathrm{d}\tau\right)^{1/p} & \text{if } p \in [0,\infty), \\ \sup\limits_{0\leqslant\tau\leqslant t} \|x(\tau)\| & \text{if } p = \infty, \end{cases}$$

where $\|x\|$ denotes any norm of x, and t belongs to the finite interval $[0,T]$. We say that $x \in \mathscr{L}_{pe}$ when $\|x\|_{pe}$ exists (i.e., when $\|x\|_{pe}$ is finite).

We assume that all the system parameters are unknown and we make the following reasonable assumptions:

(A1) The reference trajectory and its first and second time-derivatives, namely $q_d(t)$, $\dot{q}_d(t)$ and $\ddot{q}_d(t)$, as well as the disturbance $d_k(t)$ are bounded $\forall t \in [0,T]$ and $\forall k \in \mathbb{Z}_+$.
(A2) The resetting condition is satisfied, i.e., $\dot{q}_d(0) - \dot{q}_k(0) = q_d(0) - q_k(0) = 0$, $\forall k \in \mathbb{Z}_+$.

We will also make use of the following properties, which are common to robot manipulators

(P1) $M(q_k) \in \mathbb{R}^{n\times n}$ is symmetric, bounded, and positive definite.
(P2) The matrix $\dot{M}(q_k) - 2C(q_k,\dot{q}_k)$ is skew symmetric, hence $x^{\mathrm{T}}(\dot{M}(q_k) - 2C(q_k,\dot{q}_k))x = 0$, $\forall x \in \mathbb{R}^n$.
(P3) $\|C(q_k,\dot{q}_k)\| \leqslant k_c\|\dot{q}_k\|$ and $\|G(q_k)\| < k_g$, $\forall t \in [0,T]$ and $\forall k \in \mathbb{Z}_+$, where k_c and k_g are unknown positive parameters.

3. Adaptive ILC

Let us consider system (1) under the following control law (Fig. 1):

$$\tau_k(t) = K_P\tilde{q}_k(t) + K_D\dot{\tilde{q}}_k(t) + \eta(\dot{\tilde{q}}_k)\hat{\theta}_k(t) \tag{2}$$

with

$$\hat{\theta}_k(t) = \hat{\theta}_{k-1}(t) + \Gamma\eta^{\mathrm{T}}(\dot{\tilde{q}}_k)\dot{\tilde{q}}_k(t), \tag{3}$$

where $\hat{\theta}_{-1}(t) = 0$, $\tilde{q}_k(t) = q_d(t) - q_k(t)$ and $\dot{\tilde{q}}_k(t) = \dot{q}_d(t) - \dot{q}_k(t)$. The matrices $K_P \in \mathbb{R}^{n\times n}$ and $K_D \in \mathbb{R}^{n\times n}$ are symmetric positive definite.

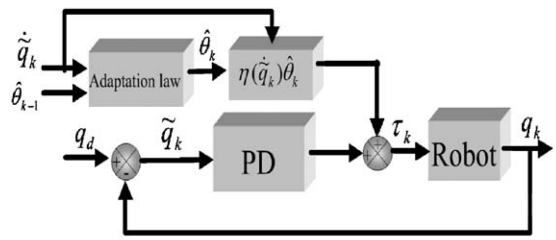

Fig. 1. Adaptive ILC structure.

According to the choice of $\eta(\dot{\tilde{q}}_k)$, we have two different control laws CL1 and CL2:

(CL1) Using two iterative variables (i.e., $\hat{\theta}_k(t) \in \mathbb{R}^2$): $\eta(\dot{\tilde{q}}_k) = [\dot{\tilde{q}}_k \, sgn(\dot{\tilde{q}}_k)] \in \mathbb{R}^{n \times 2}$ where $sgn(\dot{\tilde{q}}_k)$ is the vector obtained by applying the signum function to all elements of $\dot{\tilde{q}}_k$. $\Gamma \in \mathbb{R}^{2 \times 2}$ is symmetric positive definite.

(CL2) Using a single iterative variable (i.e., $\hat{\theta}_k(t) \in \mathbb{R}$): $\eta(\dot{\tilde{q}}_k) = sgn(\dot{\tilde{q}}_k(t)) \in \mathbb{R}^n$. Γ is a positive scalar and $(K_D - \alpha I)$ is positive semi-definite, with $\alpha = k_c \, sup_{t \in [0,T]} \|\dot{q}_d(t)\|$.

Then, under assumptions (A1 and A2) and properties (P1–P3), one has

- $\tilde{q}_k(t) \in \mathscr{L}_{\infty e}$, $\dot{\tilde{q}}_k(t) \in \mathscr{L}_{\infty e}$, $\tau_k(t) \in \mathscr{L}_{2e}$ for all $k \in \mathbb{Z}_+$.
- $\lim_{k \to \infty} \tilde{q}_k(t) = \lim_{k \to \infty} \dot{\tilde{q}}_k(t) = 0$, $\forall t \in [0, T]$.

The proof of this statement can be found in Tayebi (2004).

Remark 1. Generally, in contraction-mapping-based ILC schemes, the number of iterative parameters is equal to the number of the control inputs which is equal to the number of degrees of freedom n. For instance, for a P-type ILC of the form $u_{k+1} = u_k + K\tilde{q}_k$ or a D-type ILC of the form $u_{k+1} = u_k + K\dot{\tilde{q}}_k$, we need to save the signals $u_k(t)$ and ($\tilde{q}_k(t)$ or $\dot{\tilde{q}}_k(t)$) in the memory in order to be able to generate $u_{k+1}(t)$. Therefore, at each sampling time, we need to save $2n$ parameters in the memory. In our approach, we use only two iterative parameters in CL1 which is an interesting fact from the practical point of view since it contributes considerably to memory space saving (i.e., at each sampling time just two parameters are saved in the memory). In CL2, we bring down the number of iterative parameters to one at the expense of a certain knowledge of the system parameters (i.e., the parameter k_c defined in P3 is needed).

Remark 2. The tracking error and its time derivative, at the first iteration, can be made arbitrarily small, over the finite time interval $[0, T]$, by increasing the minimum eigenvalues of the control and learning gains K_P, K_D and Γ.

Remark 3. The ILC schemes CL1 and CL2 can be used in a straightforward manner for industrial robot manipulators already functioning under a PD controller by just adding the iterative term to the control input in order to enhance the tracking performance from operation to operation.

Remark 4. The sign function used in the control laws CL1 and CL2 might lead to the chattering phenomenon. In practical applications, the sign function can be replaced by a continuous approximation (e.g., saturation or sigmoid) in order to smooth out the control input and reduce the chattering.

4. Experimental results

In this section, the two adaptive ILC algorithms (CL1 and CL2) are implemented and evaluated on the 5-DOF robot manipulator CRS255 (CATALYST5) shown in Fig. 2.

The CRS255 is a 5-DOF open-chain articulated robot arm. The arm is constructed of high tensile strength aluminum alloy components. It has five revolute joints powered by five DC motors. At each joint, an incremental encoder is employed for joint position measurement. As the gravity forces are not counter balanced, motors for vertical joints are equipped with automatic brakes to prevent the collapse of the manipulator configuration if the power supply to the joint motors is interrupted. This robot comes with the CRS-C500 controller, which contains five PID feedback controllers operating about each motor and their structures cannot be modified. In order to implement our control strategy, we have to by-pass the CRS-C500 controller through the Quanser open-architecture mode

Fig. 2. CRS-255 Robot manipulator.

which allows to use Simulink for real-time control implementation. In order to do so, a Quanser–MultiQ acquisition board is used together with a Quanser–WinCon software (allowing to generate real-time code from Simulink). A switch mounted on the CRS-C500 control box allows us to switch back and forth from the Quanser-open-architecture mode to the CRS mode. For real-time implementation of our control algorithm using the Quanser open-architecture mode, the following components are installed in our host/supervisor Pentium III PC: MATLAB/Simulink/Realtime Workshop/Control systems toolbox, WinCon, Visual C++ professional.

Our objective consists to track a circle in the $(Y\text{–}Z)$-plane, described by the following equations:

$$y(t) = 10 + \sqrt{200}\cos 0.1t \text{ (mm)},$$

$$z(t) = 10 + \sqrt{200}\sin 0.1t \text{ (mm)}, \quad (4)$$

where $t \in [0, 63s]$. To realize this motion, only joints 1, 2 and 3 are needed. From the world coordinates $x(t)$, $y(t)$ and $z(t)$, we generate the desired joint variables $q_d^1(t)$,

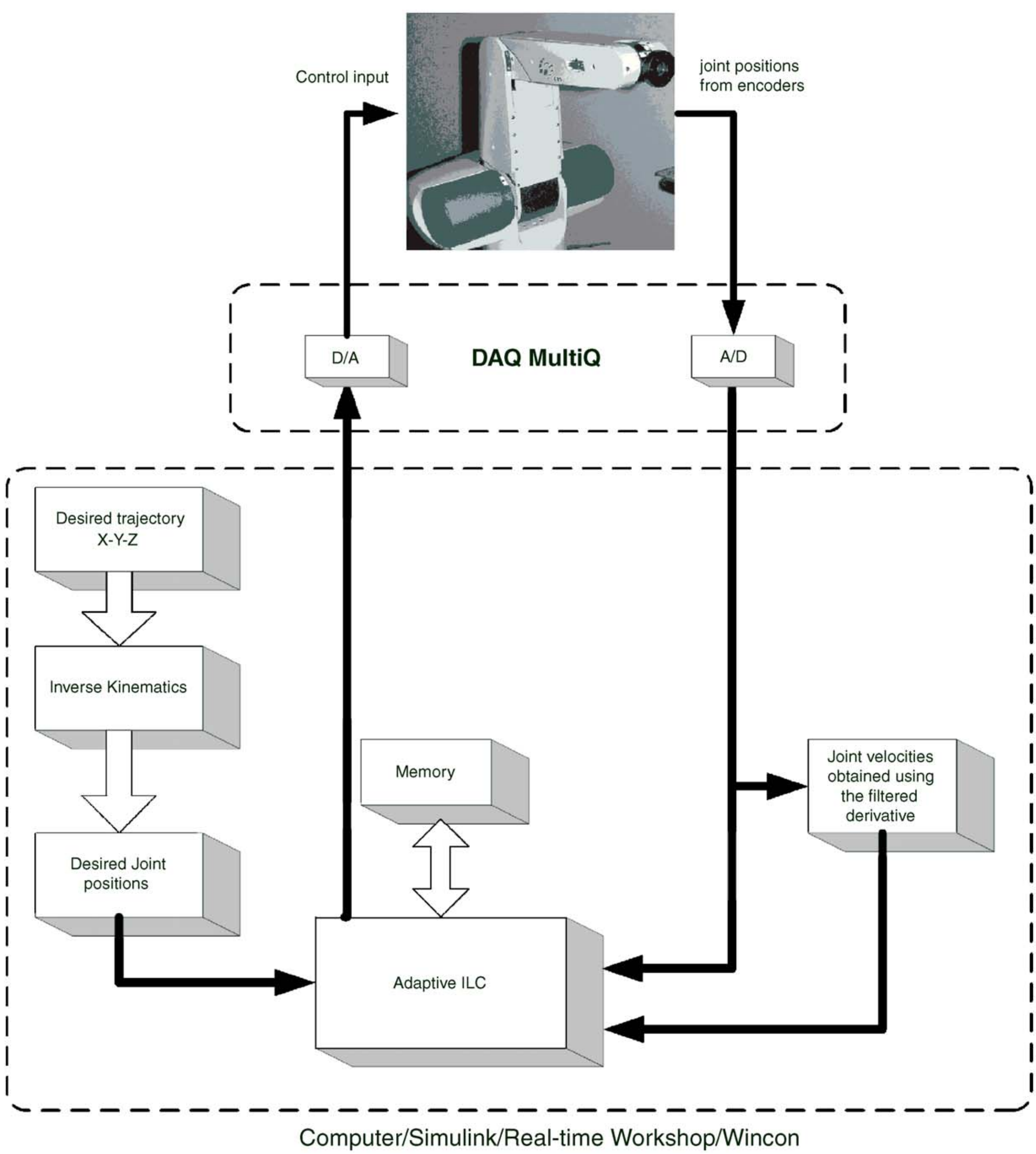

Fig. 3. ILC implementation.

$q_d^2(t)$ and $q_d^3(t)$ using the inverse kinematics. In order to smooth out the control input and reduce the chattering, the sign function has been replaced by a saturation function defined as follows:

$$sat(\dot{\tilde{q}}_k) = [sat(\dot{\tilde{q}}_{1,k}), \ldots, sat(\dot{\tilde{q}}_{3,k})]^{\mathrm{T}}$$

with

$$sat(\dot{\tilde{q}}_{i,k}) = \begin{cases} \dot{\tilde{q}}_{i,k} & \text{if } |\dot{\tilde{q}}_{i,k}| < 0.01, \\ 1 & \text{if } \dot{\tilde{q}}_{i,k} \geqslant 0.01, \\ -1 & \text{if } \dot{\tilde{q}}_{i,k} \leqslant -0.01 \end{cases} \tag{5}$$

for $i \in \{1, 2, 3\}$.

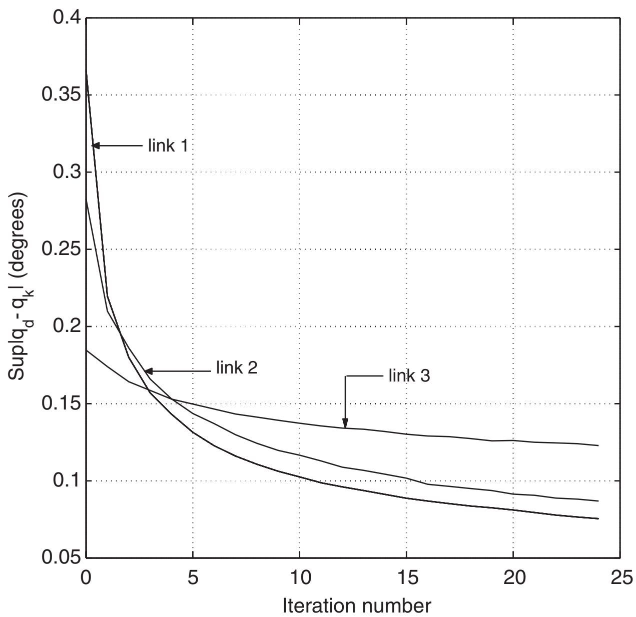

Fig. 4. Experiment 1: Sup-norm of the tracking error (in degrees) versus the number of iterations for links 1, 2 and 3 under CL1.

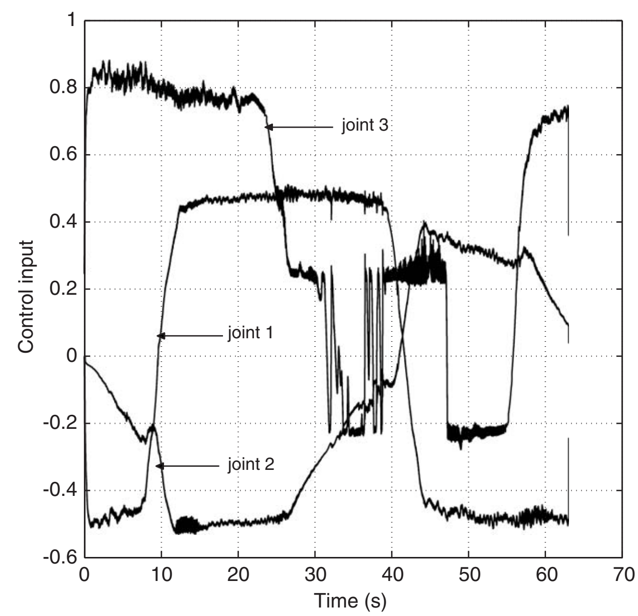

Fig. 6. Experiment 1: Control action for joints 1, 2 and 3 at the first iteration under CL1.

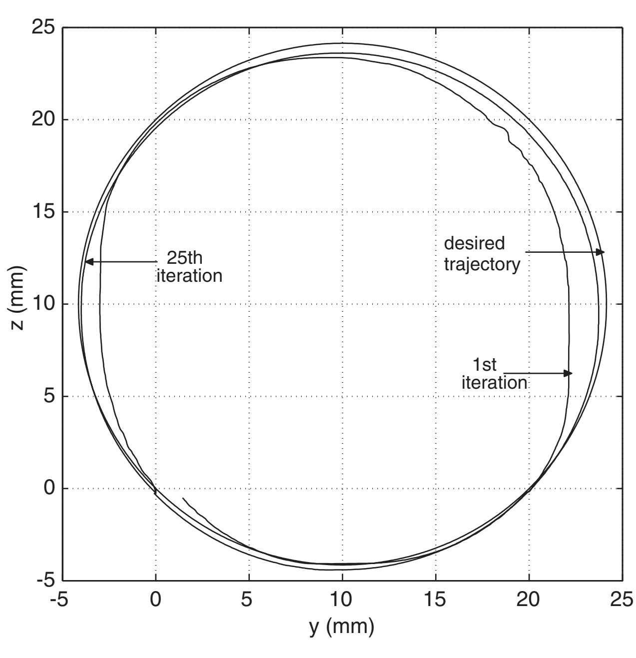

Fig. 5. Experiment 1: Desired and actual trajectories at the first and 25th iteration under CL1.

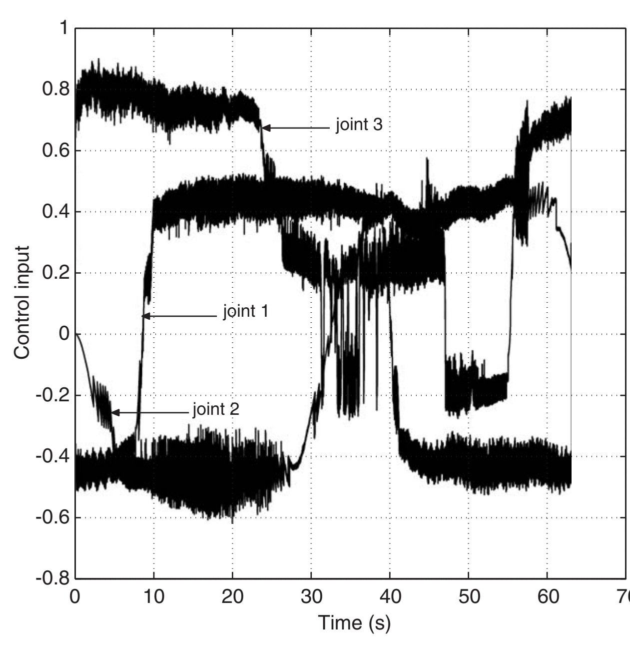

Fig. 7. Experiment 1: Control action for joints 1, 2 and 3 at the 25th iteration under CL1.

On the other hand, the joint velocities are not measured but estimated from the joint positions using a filtered derivative (i.e., substituting $\dot{\tilde{q}}_k$ in the control and adaptation laws by $(s/(1+T_f s))\tilde{q}_k$, with $T_f = (1/2\pi f_c)$, where the cut-off frequency $f_c = 0.08$ Hz). The sampling period is taken as 1 ms.

In order to investigate the effect of the learning gain, we performed two experiments with each control law.

Experiment 1. The control and learning gains for CL1 are taken as follows: $K_p = diag(1, 1, 4)$, $K_D = diag(0.05, 0.05, 0.05)$ and $\Gamma = diag(8, 8)$. The control and learning gains for CL2 are taken as follows: $K_p = diag(1, 1, 4)$, $K_D = diag(0.05, 0.05, 0.05)$ and $\Gamma = 8$.

Experiment 2. The control and learning gains for CL1 are taken as follows: $K_p = diag(1, 1, 4)$,

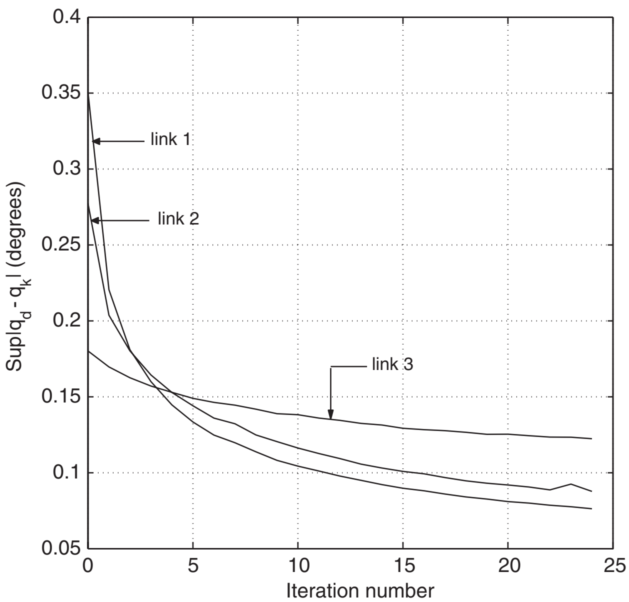

Fig. 8. Experiment 1: Sup-norm of the tracking error (in degrees) versus the number of iterations for links 1, 2 and 3 under CL2.

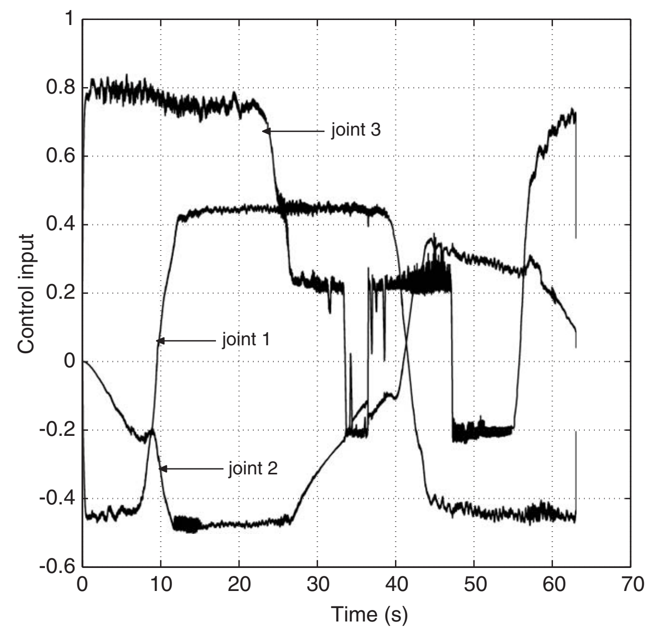

Fig. 10. Experiment 1: Control action for joints 1, 2 and 3 at the first iteration under CL2.

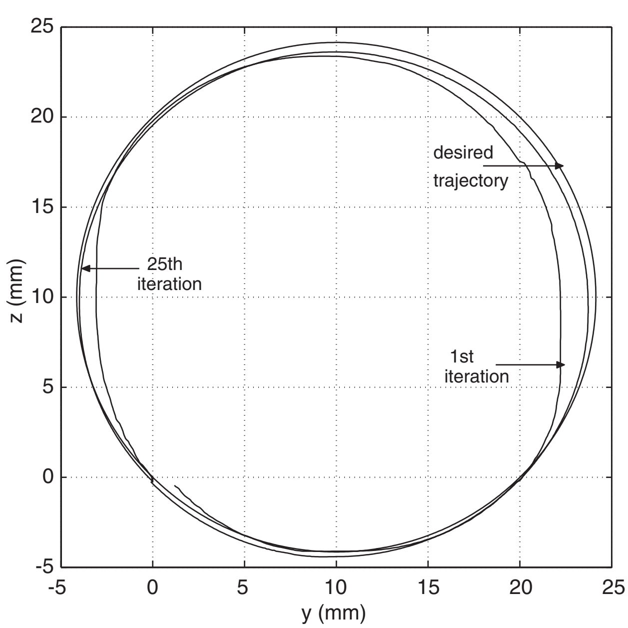

Fig. 9. Experiment 1: Desired and actual trajectories at the first and 25th iteration under CL2.

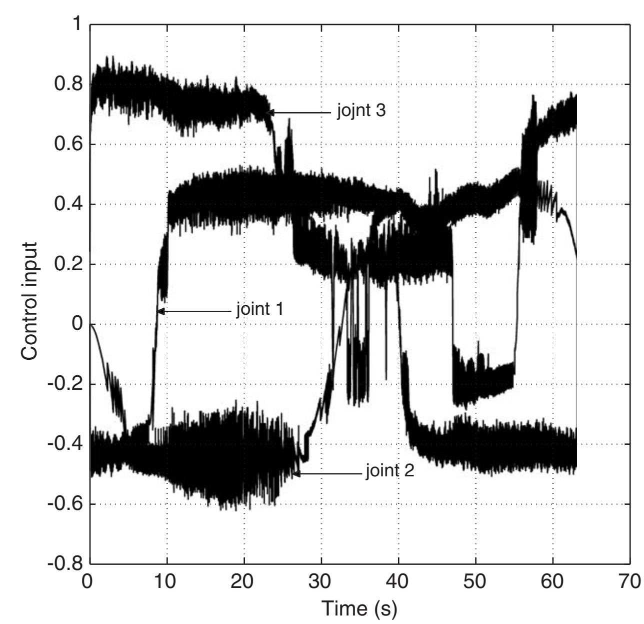

Fig. 11. Experiment 1: Control action for joints 1, 2 and 3 at the 25th iteration under CL2.

$K_D = diag(0.05, 0.05, 0.05)$ and $\Gamma = diag(2, 2)$. The control and learning gains for CL2 are taken as follows: $K_p = diag(1, 1, 4)$, $K_D = diag(0.05, 0.05, 0.05)$ and $\Gamma = 2$.

Figs. 4–11 are obtained for experiment 1. The ILC implementation block diagram is depicted in Fig. 3. Fig. 4 shows the evolution of the sup-norm of the tracking error with respect to the iteration number, under CL1. Fig. 5 shows the reference trajectory and the actual trajectory at the first and 25th iteration under CL1. Figs. 6 and 7 show the control inputs at the first and the 25th iterations, respectively, under CL1. Fig. 8 shows the evolution of the sup-norm of the tracking error with respect to the iteration number, under CL2. Fig. 9 shows the reference trajectory and the actual trajectory at the first and 25th iteration under CL2. Figs. 10 and 11 show the control inputs at the first and

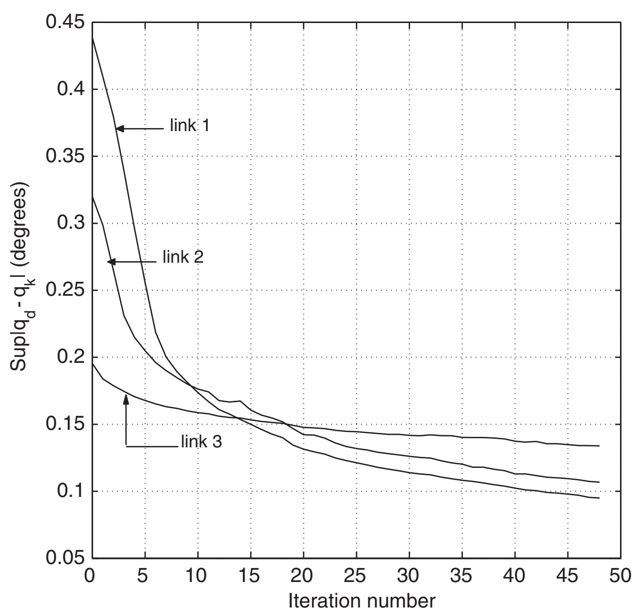

Fig. 12. Experiment 2: Sup-norm of the tracking error (in degrees) versus the number of iterations for links 1, 2 and 3 under CL1.

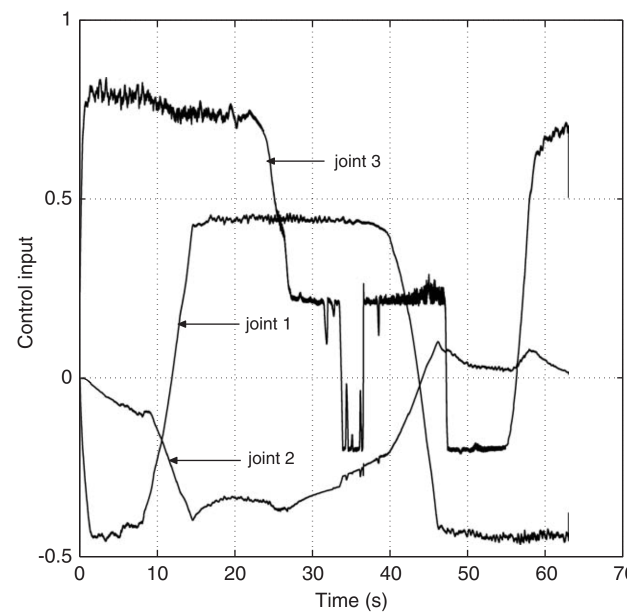

Fig. 14. Experiment 2: Control action for joints 1, 2 and 3 at the first iteration under CL1.

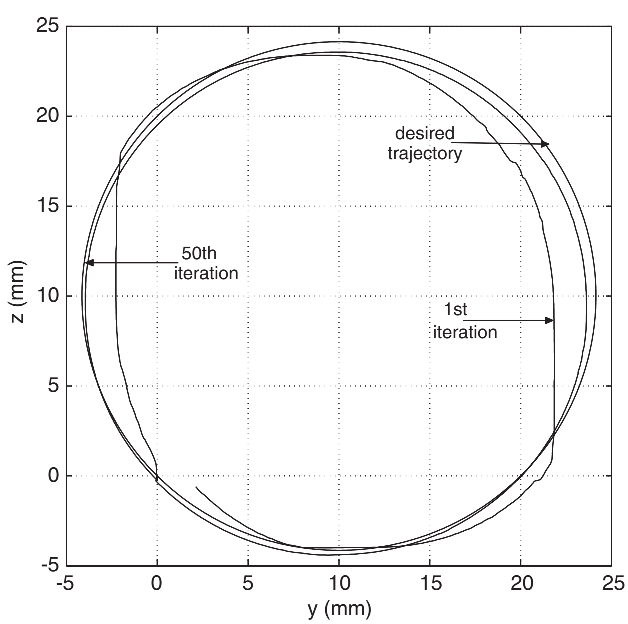

Fig. 13. Experiment 2: Desired and actual trajectories at the first and 50th iteration under CL1.

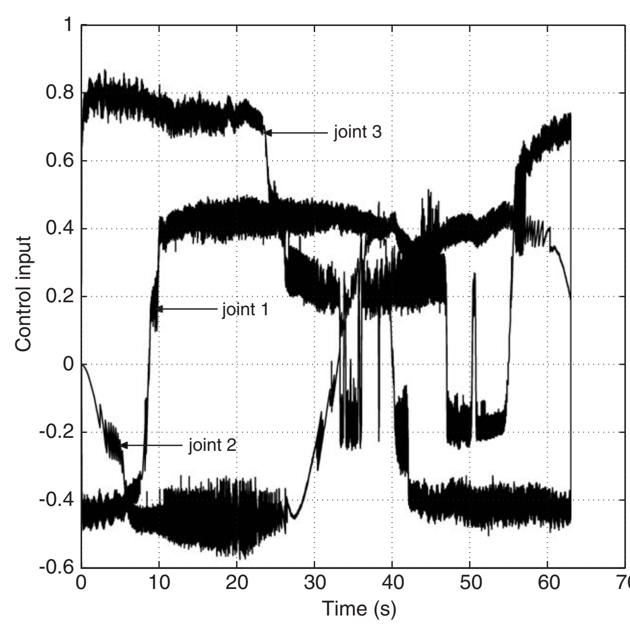

Fig. 15. Experiment 2: Control action for joints 1, 2 and 3 at the 50th iteration under CL1.

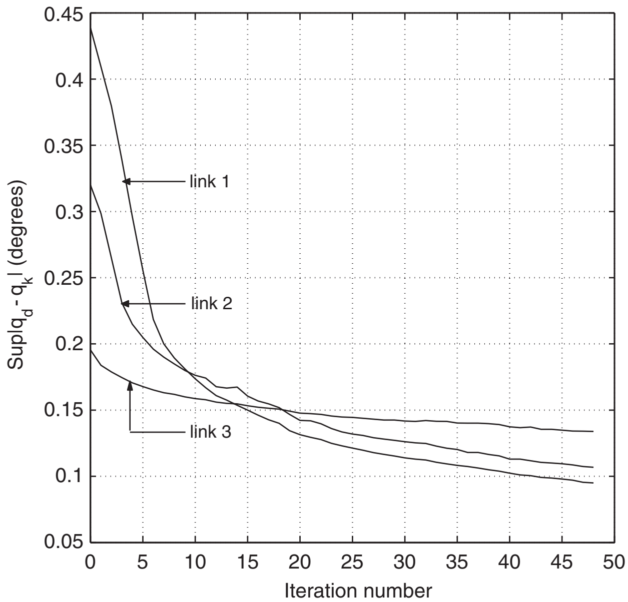

Fig. 16. Experiment 2: Sup-norm of the tracking error (in degrees) versus the number of iterations for links 1, 2 and 3 under CL2.

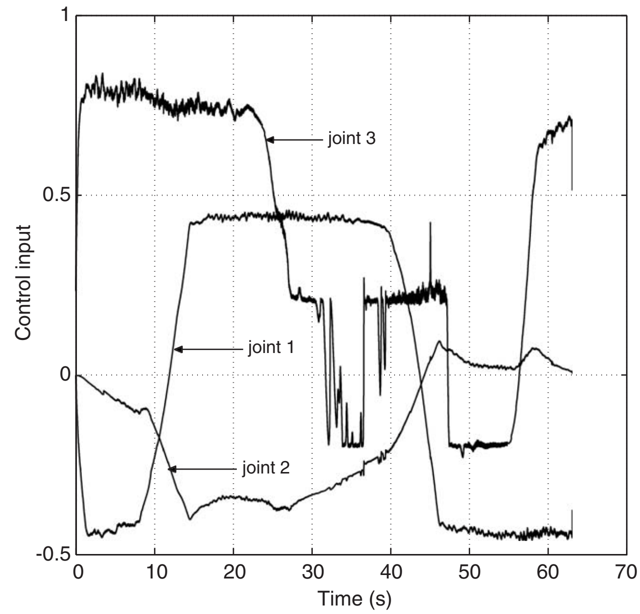

Fig. 18. Experiment 2: Control action for joints 1, 2 and 3 at the first iteration under CL2.

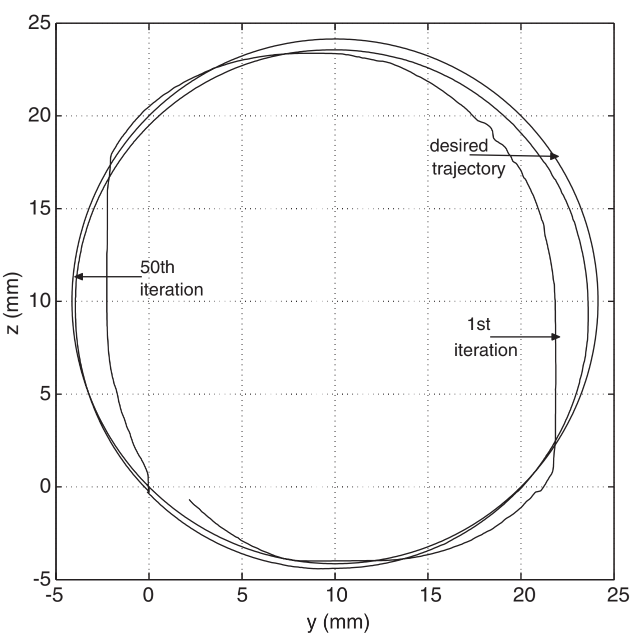

Fig. 17. Experiment 2: Desired and actual trajectories at the first and 50th iteration under CL2.

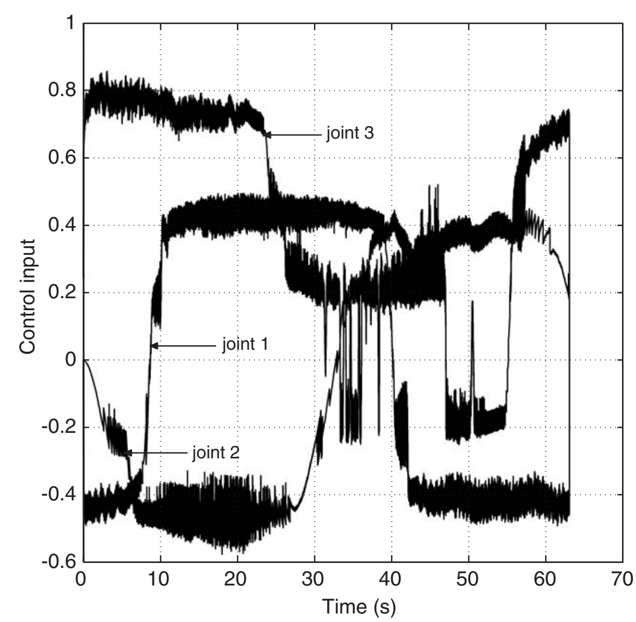

Fig. 19. Experiment 2: Control action for joints 1, 2 and 3 at the 50th iteration under CL2.

the 25th iterations, respectively, under CL2. Figs. 12–19 are obtained similarly for experiment 2.

5. Conclusion

Two adaptive ILC schemes, proposed in Tayebi (2004), have been successfully tested on a 5-DOF robot manipulator CATALYST5. The control schemes consist of a PD feedback plus an additional adaptive term introduced to cope with the unknown parameters and disturbances. The overall control strategy is very simple to implement since no a priori knowledge of the robot parameters is needed, and the only requirement on the control gains is the positive definiteness condition. Another clear advantage of this approach is the fact that it uses just one or two iterative variables, which helps to reduce the memory space requirements in practical implementations.

During our several tests, we noticed that the cut-off frequency of the low pass filter, used to generate the joint velocities from the joint positions, plays a crucial role. In fact, from a theoretical point of view, a high cut-off frequency would result in a good approximation of the derivative action and hence would lead to a good performance. However, according to our experiments, we noticed that at high cut-off frequencies the robot joints start to vibrate after a certain number of iterations forcing us to stop the learning process. This is mainly due to the fact that the noise amplification, caused by the derivative action, is accumulating through the iterative process (see Figs. 6, 7, 10 and 11). By gradually reducing the cut-off frequency, we noticed a considerable improvement in the tracking performance in terms of measurement noise rejection. However, a very low cut-off frequency would result in a bad approximation of the joint velocity and therefore the stability and convergence of the iterative process are not guaranteed any more.

We also noticed that the convergence rates could be improved by increasing the learning gain as seen from the results of experiments 1 and 2. However, the noise level increases with the learning gain causing the joints to vibrate earlier in the iteration domain as evidenced by the two experiments. In fact, in the first experiment we had to stop the learning process at the 25th iteration while for the second experiment we were able to go to the 50th iteration without any problem.

A potential solution to this crucial problem, related to the use of the filtered derivative also known as the ''dirty derivative'', is to design P-type iterative parametric updating rules that do not require the joint velocities measurements. In this case, the noise effect will be reduced considerably, but will not be totally eliminated since the joint positions measurements are also noisy, and there will be an accumulation of noise from iteration to iteration. Consequently, it is important to stop the learning process after a certain number of iterations once the tracking error reaches a certain acceptable level. The theoretical implications of this crucial problem will be investigated in our future research work.

Acknowledgements

The authors would like to thank the anonymous reviewers for their helpful comments.

References

Arimoto, S. (1996). *Control theory of non-linear mechanical systems*. Oxford: Oxford Science Publications.

Arimoto, S., Kawamura, S., & Miyazaki, F. (1984). Bettering operation of robots by learning. *Journal of Robotics Systems*, *1*, 123–140.

Bondi, P., Casalino, G., & Gambardella, L. (1988). On the iterative learning control theory for robotic manipulators. *IEEE Journal of Robotics and Automation*, *4*(1), 14–22.

Casalino, G., & Bartolini, G. (1984). A learning procedure for the control of movements of robotic manipulators. *IASTED symposium on robotics and automation*, Amsterdam (pp. 108–111).

Choi, J. Y., & Lee, J. S. (2000). Adaptive iterative learning control for uncertain robotic systems. *IEE-Proceedings, Control Theory Applications*, *147*(2), 217–223.

Craig, J. J. (1984). Adaptive control of manipulators through repeated trials. *Proceedings of the American control conference*, San Diego (pp. 1566–1574).

De Luca, A., Paesano, G., & Ulivi, G. (1992). A frequency domain approach to learning control: Implementation for a robot manipulator. *IEEE Transactions On Industrial Electronics*, *39*(1), 1–10.

French, M., & Rogers, E. (2000). Nonlinear iterative learning by an adaptive Lyapunov technique. *International Journal of Control*, *73*(10), 840–850.

Ham, C., Qu, Z., & Johnson, R. (2000). A nonlinear iterative learning control for robot manipulators in the presence of actuator dynamics. *International Journal of Robotics and Automation*, *15*(3), 119–130.

Ham, C., Qu, Z. H., & Kaloust, J. H. (1995). Nonlinear learning control for a class of nonlinear systems based on Lyapunov's direct method. *Proceedings of the American control conference*, Seattle, USA (pp. 3024–3028).

Horowitz, R. (1993). Learning control of robot manipulators. *ASME Journal of Dynamic Systems, Measurement and Control*, *115*, 402–411.

Kavli, T. (1992). Frequency domain synthesis of trajectory learning controller for robot manipulators. *Journal of Robotic Systems*, *9*(5), 663–680.

Kawamura, S., Miyazaki, F., & Arimoto, S. (1988). Realization of robot motion based on a learning method. *IEEE Transactions on Systems, Man and Cybernetics*, *18*(1), 126–134.

Kuc, T. Y., Nam, K., & Lee, J. S. (1991). An iterative learning control of robot manipulators. *IEEE Transactions on Robotics and Automation*, *7*(6), 835–841.

Moon, J. H., Doh, T. Y., & Chung, M. J., 1997. An iterative learning control scheme for manipulators. *Proceedings of international conference on intelligent robots and systems*, Grenoble, France.

Tayebi, A. (2004). Adaptive iterative learning control for robot manipulators. *Automatica*, *40*(7), 1195–1203.

Xu, J.-X. (2002). The frontiers of iterative learning control—Part II. *Journal of Systems, Control and Information*, *46*(5), 233–243.

Xu, J.-X., Badrinath, V., & Qu, Z. (2000). Robust learning control for robotic manipulators with an extension to a class of nonlinear systems. *International Journal of Control*, *73*(10), 858–870.

Xu, J.-X., & Tan, Y. (2001). A suboptimal learning control scheme for nonlinear systems with time-varying parametric uncertainties. *Journal of Optimal Control—Applications and Theory*, *22*, 111–126.