

ANTICIPATIVE ITERATIVE LEARNING CONTROL OF ROBOT MANIPULATORS

Ilia G. Polushin, * Abdelhamid Tayebi **

* *Department of Systems and Computer Engineering,
Carleton University, Ottawa, ON, K1S 5B6, Canada*

** *Department of Electrical Engineering, Lakehead
University, Thunder Bay, ON, P7B 5E1, Canada*

Abstract: An iterative learning scheme for the tracking control of robot manipulators without velocity measurement is presented. The proposed learning algorithm is anticipative (noncausal) in the sense that it utilizes “future” values of the tracking error obtained during the previous iteration. Also, the standard resetting assumption is relaxed to the form of δ_q -resetting assumption. The proposed algorithm ensures convergence of the tracking error to a prescribed small domain in finite number of iterations, uniformly in time. Some experimental results on a six-degrees-of-freedom (6-DOF) robot manipulator are presented to show the effectiveness of the proposed algorithm. *Copyright ©2005 IFAC*

Keywords: Learning algorithms, Iterative improvement, Robot control

1. INTRODUCTION

Robot manipulators are generally used in manufacturing industries to execute a given task repeatedly over a finite time interval. Therefore, Iterative Learning Control (ILC) is a suitable control technique candidate for this kind of applications. In the past two decades several ILC schemes for robot manipulators have been proposed in the literature. In particular, ILC schemes that are based upon the contraction mapping theory have been proposed in (Arimoto *et al.*, 1984; Arimoto, 1996; Bondi *et al.*, 1988; Casalino and Bartolini, 1984; Craig, 1984; De Luca *et al.*, 1992; Horowitz, 1993; Kavli, 1992; Kawamura *et al.*, 1988; Moon *et al.*, 1997). On the other hand, another relatively recent type of ILC algorithms has been developed using Lyapunov and Lyapunov-like methods (Choi and Lee, 2000; Kuc *et al.*, 1991; Ham *et al.*, 2000; Xu *et al.*, 2000). Recently, some ILC schemes for the position tracking problem of rigid robot manipulators have been proposed in (Tayebi, 2004; Polushin and Tayebi, 2004).

These control schemes are build around a classical PD feedback structure, for which an iterative term is added in order to ensure asymptotic convergence along the iteration axis.

In this paper, a new iterative learning scheme for the tracking control of robot manipulators is presented. An essential feature of this iterative learning scheme is that the learning algorithm proposed is anticipative in the sense that it utilizes “future” values of the tracking error obtained during the previous iteration. This anticipative or noncausal feature provides an important advantage of ILC over conventional feedback control schemes. Another interesting feature of the results presented is that the resetting assumption, which is standard in most ILC schemes, is relaxed to the form of δ_q -resetting assumption. More precisely, instead of assuming that the initial condition is strictly determined, the same for each iteration and located on desired trajectory, we only assume that the initial states for all iterations are located in some sufficiently small neighborhood of a given

point. We first provide a general result (Theorem 1) where the convergence properties of the proposed algorithm are proved under the assumptions that include, among others, the availability of appropriate estimates for the joint velocities. We would like to emphasize that the proposed ILC scheme allows us to use different velocity estimates in the PD-type control algorithm and in the iterative learning algorithm. The main reason for using two different velocity estimates is based on the fact that the proposed control algorithm requires an on-line estimate of the current joint velocities in contrast with the learning algorithm that calculates the learning term during the k -th iteration based on measurements obtained at the $k-1$ -th iteration only. Therefore, in the latter case all the necessary calculations can be performed off-line between iterations. Thus, one may take advantage of using off-line numerical methods to obtain a possibly smoother velocity estimate in order to avoid the “accumulation of noise” in the learning term. Further, Theorem 2 presents a special case of the result described above, where a simple “dirty derivative” filter is utilized to obtain the velocity estimates for both control and learning algorithms. In this special case we show that the convergence properties of the proposed scheme can be proved, if the level of measurement disturbances is low, and the cut-off frequency of the “dirty-derivative” filter is high enough. The proposed algorithm has been implemented and tested on a 6-DOF robot manipulator CRS A465. The experimental results obtained confirm good convergence properties of the proposed algorithm. In particular, we have shown that the anticipative feature provides some additional improvement to the performance.

The paper is organized as follows. In section 2, the necessary preliminary materials are provided. Main results of the paper are formulated in Section 3. Experimental results are presented in Section 4. Finally, some well-known properties of Euler-Lagrange equations of robot manipulators are summarized in Appendix A.

2. PRELIMINARIES

Let $k \in \{0, 1, \dots\}$. An n degree of freedom rigid fully actuated manipulator during can be described by Euler-Lagrange equations of the following standard form

$$M(q_k)\ddot{q}_k + C(q_k, \dot{q}_k)\dot{q}_k + G(q_k) = \tau_k + d. \quad (1)$$

Here $q_k(t) \in \mathbb{R}^n$, $\dot{q}_k(t) \in \mathbb{R}^n$ are vectors of generalized coordinates and generalized velocities respectively, where $k \in \{0, 1, \dots\}$ is the iteration number, and $t \in [0, T]$, $T > 0$ is the duration of each iteration. Also, $\tau_k \in \mathbb{R}^n$ is the vector of external forces, $M(q_k) \in \mathbb{R}^{n \times n}$, $C(q_k, \dot{q}_k) \in \mathbb{R}^{n \times n}$, and

$G(q_k) \in \mathbb{R}^n$ are smooth matrix-valued (vector-valued) functions of their arguments, $M(q_k)$ represents the inertia matrix of the manipulator, $C(q_k, \dot{q}_k)\dot{q}_k$ is the vector of centrifugal and Coriolis forces, and $G(q_k)$ is the vector of potential forces. Equation (1) has several well-known properties that are summarized in Appendix A.

In this paper, we address a problem of tracking a trajectory that is repeated over a given operation time. Throughout the paper, the following assumptions will be used.

Assumption 1. The reference trajectory $q_d(\cdot) \in C^2[0, T]$ (i.e., $q_d(t)$ is twice continuously differentiable on $[0, T]$).

In particular, this assumptions implies that, given a reference trajectory $q_d(\cdot)$, there exist nonnegative constants $q_{d\max}$, $\dot{q}_{d\max}$, $\ddot{q}_{d\max}$ such that $\sup_{t \in [0, T]} |q_d(t)| = q_{d\max}$, $\sup_{t \in [0, T]} |\dot{q}_d(t)| = \dot{q}_{d\max}$, and $\sup_{t \in [0, T]} |\ddot{q}_d(t)| = \ddot{q}_{d\max}$.

Assumption 2. The disturbances $d(t)$ are iteration-invariant and uniformly bounded on $[0, T]$ by some bound $k_d > 0$, i.e.,

$$\sup_{t \in [0, T]} |d(t)| \leq k_d.$$

Note that the assumption that disturbances are iteration invariant is technical. We need it to guarantee that the acceptable performance can be achieved in a certain number of iteration.

Assumption 3. The joint position of the manipulator is available for measurement subject to (small) measurement disturbances. More precisely, let \bar{q}_k denote a measured error defined as follows

$$\bar{q}_k(t) = \tilde{q}_k(t) + \Delta_{q_k}(t), \quad (2)$$

where $\tilde{q}_k = q_d - q_k$, and $\Delta_{q_k}(\cdot)$ are measurement disturbances at the k -th iteration. For all $k \in \{1, 2, \dots\}$, the measurement disturbances $\Delta_{q_k}(\cdot)$ are assumed to be absolutely continuous on $[0, T]$.

We propose the following PD-type control algorithm

$$\tau_k(t) = K_P \bar{q}_k(t) + K_D \nu_k(t) + \hat{\theta}_k(t) \left(\nu_k(t) + \frac{\nu_k(t)}{\max\{\sigma, |\nu_k(t)|\}} \right), \quad (3)$$

with $\hat{\theta}_0(t) \equiv 0$, where $K_D, K_P \in \mathbb{R}^{n \times n}$ are symmetric positive definite matrices, $\sigma > 0$ is a constant, $\nu_k(\cdot)$ is a velocity error estimate obtained from the joint position measurement during k -th iteration, and $\hat{\theta}_k(\cdot): [0, T] \rightarrow \mathbb{R}^+$, $k \in \{0, 1, 2, \dots\}$, is an iteratively learned function. For each iteration $k \in \{1, 2, \dots\}$ and each $t \in [0, T]$, a learning algorithm is either

$$L_1: \hat{\theta}_k(t) := \max \left\{ \hat{\theta}_{k-1}(t), \min \left\{ \hat{\theta}_{k-1}(t), \theta^* \right\} + \gamma \sup_{s \in [t-\tau_b, t+\tau_f] \cap [0, T]} \mathcal{W}_{k-1}(s) \right\}, \quad (4)$$

or

$$L_2: \hat{\theta}_k(t) := \hat{\theta}_{k-1}(t), \quad (5)$$

where

$$\mathcal{W}_{k-1}(s) := |w_{k-1}(s)|^2 + \frac{|w_{k-1}(s)|^2}{\max \{\sigma, |w_{k-1}(s)|\}}, \quad (6)$$

and $w_{k-1}(\cdot)$ is an estimate for the joint velocity error $\tilde{q}_{k-1}(\cdot)$, $\tau_b \geq 0$, $\tau_f \geq 0$ are constants that determine the size of the “time window” of the learning algorithm, $\gamma > 0$ is a learning gain, and $\theta^* > 0$ is a sufficiently large constant. More precisely, θ^* should be chosen to satisfy $\theta^* \geq 2\beta + \alpha$, where

$$\alpha = k_c \dot{q}_{d \max} \quad (7)$$

$$\beta = \lambda_{\max}(M) \ddot{q}_{d \max} + k_c \dot{q}_{d \max}^2 + k_g + k_d, \quad (8)$$

where parameters $\lambda_{\max}(M)$, k_c , k_g are defined in Appendix A.

Remark 1. Algorithm (4) can be rewritten in the following (longer but somewhat simpler to understand) form:

i) If $\hat{\theta}_{k-1}(t) \leq \theta^*$, then

$$\hat{\theta}_k(t) := \hat{\theta}_{k-1}(t) + \gamma \sup_{s \in [t-\tau_b, t+\tau_f] \cap [0, T]} \mathcal{W}_{k-1}(s);$$

ii) If

$$\theta^* \leq \hat{\theta}_{k-1}(t) \leq \theta^* + \gamma \sup_{s \in [t-\tau_b, t+\tau_f] \cap [0, T]} \mathcal{W}_{k-1}(s),$$

then

$$\hat{\theta}_k(t) := \theta^* + \gamma \sup_{s \in [t-\tau_b, t+\tau_f] \cap [0, T]} \mathcal{W}_{k-1}(s);$$

iii) If

$$\hat{\theta}_{k-1}(t) \geq \theta^* + \gamma \sup_{s \in [t-\tau_b, t+\tau_f] \cap [0, T]} \mathcal{W}_{k-1}(s),$$

then

$$\hat{\theta}_k(t) := \hat{\theta}_{k-1}(t). \quad \bullet$$

Remark 2. Note that we use different notations $\nu(\cdot)$ and $w(\cdot)$ for the velocity error estimates used in the control algorithm (3) and the learning algorithm (4) respectively. By doing so, we would like to emphasize that these two estimates may be different and, in particular, may be obtained by using different methods. Indeed, in the control algorithm (3) we need the information about the current velocities, therefore the estimation process for $v_k(\cdot)$ has to be done on-line. On the other hand, the learning algorithm (4) utilizes

only the information about velocity error during the previous iteration, therefore the estimation process for $w_{k-1}(\cdot)$ can be done off-line between the $k-1$ -th and k -th iterations. Thus, one may take advantage of using a variety of numerical methods to obtain an appropriate velocity error estimate $w_{k-1}(\cdot)$. Specifically, from the practical point of view it seems to be particularly important to use a smooth enough estimate $w_{k-1}(\cdot)$ in order to avoid “accumulation of noise” in the learning term $\hat{\theta}_k(\cdot)$. \bullet

Remark 3. If $\tau_f > 0$ in (4), we see that, for given iteration $k \in \{1, 2, \dots\}$ and time instant $t \in [0, T]$, algorithm (4) utilizes “future” (with respect to t) values of estimate $w_{k-1}(\cdot)$ for the velocity during the previous iteration $k-1$. In this sense, algorithm (4) has anticipative (noncausal) nature with respect to time axis. \bullet

A simple switching rule between algorithms L_1 and L_2 can be described as follows. For each $k \in \{1, 2, \dots\}$, let us denote by C_k a learning algorithm which is applied to the system at the k -th step (*i.e.* which is used to calculate $\hat{\theta}_k(\cdot)$). Thus, C_k is “equal” to either L_1 or L_2 . We propose the following switching rule:

$$C_k = \begin{cases} L_1 & \text{if } \min_{j \in \{0, 1, \dots, k-1\}} \sup_{t \in [0, T]} |w_j(t)| > v_0/2; \\ L_2 & \text{otherwise,} \end{cases} \quad (9)$$

where $v_0 > 0$ is a constant to be determined. The switching algorithm (9) can be simply interpreted as follows: if during some iteration $k \in \{0, 1, \dots\}$ we have $\sup_{t \in [0, T]} |w_k(t)| \leq v_0/2$, then the learning

process is stopped, and we have $\hat{\theta}_i(t) = \hat{\theta}_k(t)$ for all $t \in [0, T]$ and all $i \in \{k, k+1, \dots\}$.

3. MAIN RESULTS

Our first result is presented below.

Theorem 1. Consider the system (1), (3), (4) (5), (9), and suppose Assumptions 1 – 3 are satisfied. Given $\epsilon > 0$, there exist $\sigma^* > 0$, $\delta_q > 0$, $\Delta^* > 0$, $\nu^* > 0$, and $w^* > 0$ such that if

i) the constant v_0 in (9) satisfies $v_0 \leq \max \{\epsilon, \frac{\epsilon}{T}\}$;

ii) the constant σ in (3), (6) satisfies $\sigma \in (0, \sigma^*)$;

iii) the following δ_q -resetting assumption

$$\max \{ |\tilde{q}_k(0)|, |\dot{\tilde{q}}_k(0)| \} \leq \delta_q \quad (10)$$

holds for all $k \in \{0, 1, 2, \dots\}$;

iv) the measurement disturbances satisfy

$$\sup_{t \in [0, T]} |\Delta_{q_k}(t)| \leq \Delta^* \quad \text{for all } k \in \{0, 1, 2, \dots\}, \quad (11)$$

v) the estimates $\nu_k(\cdot)$, $w_k(\cdot)$ satisfy

$$\begin{aligned} \sup_{k \in \{0,1,2,\dots\}} \sup_{t \in [0,T]} |\nu_k(t) - \dot{q}_k(t)| &\leq \nu^*, \\ \sup_{k \in \{0,1,2,\dots\}} \sup_{t \in [0,T]} |w_k(t) - \dot{q}_k(t)| &\leq w^*, \end{aligned} \quad (12)$$

then the closed-loop system (1), (3), (4) (5), (9) has the following properties:

a) trajectories of the closed loop system are uniformly bounded for all $k \in \{0, 1, 2, \dots\}$, *i.e.*, there exists $Q^* \geq 0$ such that

$$\sup_{k \in \{0,1,2,\dots\}} \sup_{t \in [0,T]} \max \{|\tilde{q}_k(t)|, |\dot{\tilde{q}}_k(t)|\} \leq Q^*;$$

b) there exists $k^* \in \{0, 1, \dots\}$ such that

$$\sup_{k \in \{k^*, k^*+1, \dots\}} \sup_{t \in [0,T]} \max \{|\tilde{q}_k(t)|, |\dot{\tilde{q}}_k(t)|\} \leq \epsilon,$$

and

$$\hat{\theta}_k(\cdot) \equiv \hat{\theta}_{k^*}(\cdot) \quad \text{for all } k \in \{k^*, k^* + 1, \dots\}. \quad \bullet$$

Below we will also investigate a special case of Theorem 1 where both the estimates $\nu_k(\cdot)$, $w_k(\cdot)$ are equal to each other and obtained by using a simple “dirty-derivative” filter of the form

$$T_f(s) = \frac{\rho s}{s + \rho}, \quad (13)$$

where $\rho > 0$ is a constant to be determined. Thus, $\nu_k(\cdot)$ and $w_k(\cdot)$ can be defined in the Laplace domain as follows

$$\nu_k(s) := w_k(s) := T_f(s)\bar{q}_k(s). \quad (14)$$

Let the initial condition of the filter (14) in the time domain be $\nu_k(0) = w_k(0) = 0$. For such defined estimates, one can show that assumption v) of Theorem 1 is actually satisfied, if the cut-off frequency of the filter is high enough. Consequently, one can state a modified version of Theorem 1 as follows.

Theorem 2. Consider a system (1), (3), (4) (5), (9), (14) under Assumptions 1 – 3. Given $\epsilon > 0$, there exist $\rho^* > 0$, $\sigma^* > 0$, $\delta_q > 0$, and $\Delta^* > 0$, such that if assumptions i)-iv) of Theorem 1 are satisfied, and, additionally, the constant ρ in (13) satisfies $\rho \geq \rho^*$, then the closed-loop system has the properties a), and b) of Theorem 1.

4. EXPERIMENTAL RESULTS

In this section, we present some results of experimental evaluation of the iterative learning control algorithm (3), (4). The algorithm proposed in this paper has been implemented and evaluated on a 6-DOF robot manipulator CRS A465OA (see figure 1). The CRS465 is an open-chain articulated robot arm with 6 revolute joints

powered by 6 motors. The iterative learning control algorithm proposed can be implemented using Quanser open architecture (OA) mode. For the real-time implementation of the control algorithm using the Quanser open architecture mode, WinCon software is used together with MATLAB/Simulink/Realtime Workshop/Control System Toolbox as well as Visual C++ Professional.



Fig. 1. CRS A465OA Robotic Manipulator

The reference trajectory traces an ellipse in the horizontal (X-Y) plane, described by the following equations

$$\begin{aligned} x(t) &= X_0 (1 - \cos(2\pi \cdot 0.1t)), \\ y(t) &= Y_0 (\sin(2\pi \cdot 0.1t)), \\ z(t) &\equiv Z_0, \end{aligned}$$

where $X_0 = 100$ mm, $Y_0 = 150$ mm, and $Z_0 = 70$ mm. The reference trajectories for pitch, yaw, and roll of the end-effector are set to be constantly equal to zero.

The parameters of algorithm (3), (4), are chosen as follows: $K_D = 0.5 \cdot \mathbb{I}^{6 \times 6}$, $K_P = 0.5 \cdot \mathbb{I}^{6 \times 6}$, $\gamma = 10$, $\sigma = 1$. The estimate ν in (4) is obtained using a “dirty-derivative” filter of the form (13), (14), with $\rho = 0.5$ rad/sec. On the other hand, the estimate w in (4) is also obtained using a dirty-derivative filter of the form (13), (14) where, for the sake of using a possibly smoother signal to avoid the “accumulation of noise” in the learning term, the cut-off frequency is decreased to $\rho = 0.2$ rad/sec.

Below we present the results of two sets of experiments. In the first set, we put $\tau_f = \tau_b = 0$ in (4), *i.e.*, the learning control algorithm implemented does not have anticipative properties. On the other hand, in second set of experiments, we put $\tau_b = 0$, $\tau_f = 1$, which means that the algorithm has anticipative feature with anticipatory time 1 sec. The results of these two sets of experiments are shown in figures 2–4. In particular, figure 2 represents the desired trajectory as well as the actual trajectories at 1st and 20th iterations for the algorithm with $\tau_f = 0$, while in figure 3 the

same are shown for the algorithm with $\tau_f = 1$ sec. To estimate convergence properties of the algorithms, for each iteration $k \in \{1, 2, \dots, 20\}$, the supremum norm of sum of all joint tracking errors is calculated using the following formula

$$e_k^{sum} = \sup_{t \in [0, 20 \text{ sec}]} \left(\sum_{i=1}^6 |\tilde{q}_k^i(t)| \right),$$

where $\tilde{q}_k^i(t)$ is the tracking error of i -th joint, $i \in \{1, 2, \dots, 6\}$, at time $t \in [0, 10 \text{ sec}]$. The plots of e_k^{sum} vs. iteration number for both the cases $\tau_f = 0$ and $\tau_f = 1$ sec are shown in figure 4. We see that in the both cases, the algorithm proposed provides good uniform convergence of the actual trajectory to the desired one along the iteration axis. However, the use of anticipatory feature provides some improvement to the convergence properties.

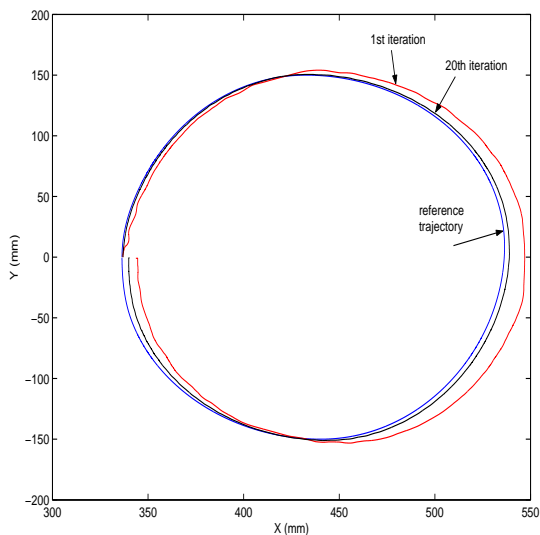


Fig. 2. Experimental results for $\tau_f = 0$: the desired trajectory and the actual trajectories at 1st and 20th iterations

5. CONCLUSIONS

In this paper, an iterative learning control scheme for robot manipulators without velocity measurement has been presented. The proposed learning algorithm is anticipative (noncausal) in the sense that “future” values of the tracking error obtained during the previous iteration are used to calculate the learning parameter. Also, the structure of the algorithm proposed makes it possible to use off-line estimation of the velocity error, which may lead to better performance of the learning process. The proposed algorithm has been implemented and tested on 6-DOF robot manipulator CRS A465. Experimental results are provided that confirm good convergence properties of the proposed algorithm. In particular, we have demonstrated that additional improvement of the convergence

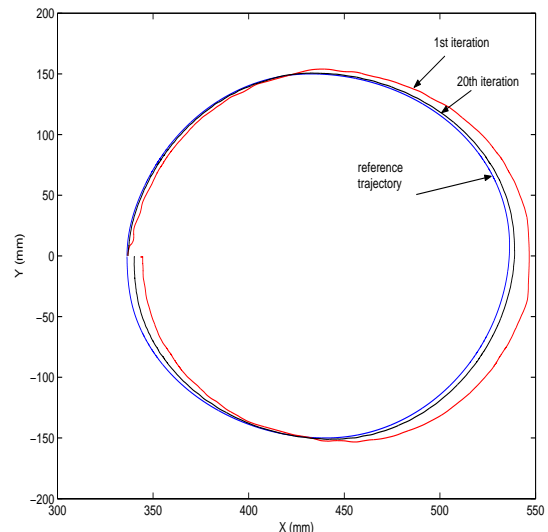


Fig. 3. Experimental results for $\tau_f = 1$ sec: the desired trajectory and the actual trajectories at 1st and 20th iterations

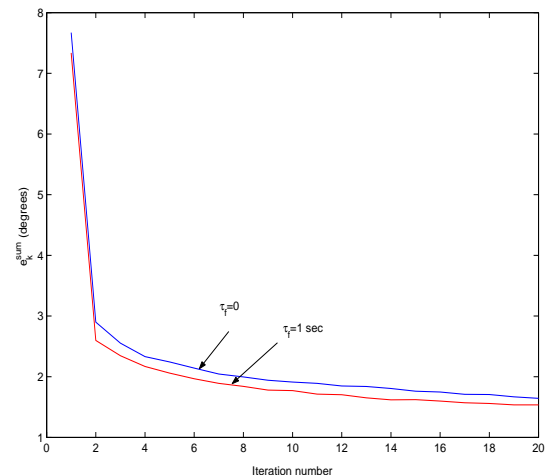


Fig. 4. Experimental results: e_k^{sum} vs. number of iteration k for algorithms with $\tau_f = 0$ and $\tau_f = 1$ sec

properties of the iterative learning algorithm can be achieved by using the anticipative property of the proposed algorithm.

Acknowledgments

The authors would like to acknowledge the support of the Institute for Robotics and Intelligent Systems (IRIS) and the Natural Sciences and Engineering Research Council (NSERC) of Canada.

REFERENCES

- Arimoto, S., S. Kawamura and F. Miyazaki (1984). Bettering operation of robots by learning. *Journal Robot. Syst.*, **1**, 123–140.

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

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

Casalino, G. and G. Bartolini (1984). A Learning procedure for the control of movements of robotic manipulators. In: *IASTED Symposium on Robotics and Automation*, pp. 108–111. Amsterdam.

Choi, J.Y. and J.S. Lee (2000). Adaptive iterative learning control for uncertain robotic systems. *IEE-Proc., Control Theory Appl.*, **147**, 217–223.

Craig, J. J., 1984. Adaptive Control of Manipulators Through Repeated Trials. In: *Proceedings of the American Control Conference*, pp. 1566–1574. San Diego, CA.

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

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

Horowitz, R. (1993). Learning control of robot manipulators. *ASME Journal of Dyn. Syst., Meas. and Control*, **115**, 402–411.

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

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

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

Moon, J.H., T.Y. Doh and M.J. Chung (1997). An iterative learning control scheme for manipulators. In: *Proceedings of International Conference on Intelligent Robots and Systems*. Grenoble, France.

Polushin, I. G. and A. Tayebi (2004). Iterative learning control scheme for robot manipulators without velocity measurement: Theory and experiments. In: *Proceedings of IFAC Workshop on Adaptation and Learning in Control and Signal Processing ALCOSP 04*, pp.675 – 679. Yokohama, Japan.

Takegaki, M. and S. Arimoto (1981). A new feedback method for dynamic control of manipulators. *Journal of Dynamic Systems, Measurement and Control*, **103**, 119–125.

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

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

APPENDIX A. PROPERTIES OF EULER-LAGRANGE EQUATIONS 1

The following properties of the dynamical model (1) are well-known and utilized in the paper.

Property 1. The inertia matrix $M(q)$ is symmetric, positive definite, and globally uniformly bounded

$$\lambda_{\min}(M) \leq |M(q)| \leq \lambda_{\max}(M)$$

for all $q \in \mathbb{R}^n$, where $\lambda_{\min}(M)$, $\lambda_{\max}(M)$ are positive constants.

Property 2. The matrix $\dot{M}(q) - 2C(q, \dot{q})$ is skew-symmetric.

Property 3. The matrix $C(q, \dot{q})$ satisfies

$$|C(q, \dot{q})| \leq k_c |\dot{q}|$$

for some $k_c > 0$, and for all $q, \dot{q} \in \mathbb{R}^n$.

Property 4. The vector of potential forces $G(q)$ is globally uniformly bounded, *i.e.*,

$$|G(q)| \leq k_g$$

for some $k_g > 0$, and for all $q \in \mathbb{R}^n$.