10.2478/v10170-011-0036-9

*Archives of Control Sciences*
*Volume* 22(LVIII), 2012
*No.* 4*, pages 467–480*

# Simulation-based design of monotonically convergent iterative learning control for nonlinear systems

KAMEN DELCHEV

This paper deals with a simulation-based design of model-based iterative learning control (ILC) for multi-input, multi-output nonlinear time-varying systems. The main problem of the implementation of the nonlinear ILC in practice is possible inadmissible transient growth of the tracking error due to a non-monotonic convergence of the learning process. A model-based nonlinear closed-loop iterative learning control for robot manipulators is synthesized and its tuning depends on only four positive gains of both controllers – the feedback one and the learning one. A simulation-based approach for tuning the learning and feedback controllers is proposed to achieve fast and monotonic convergence of the presented ILC. In the case of excessive growth of transient errors this approach is the only way for learning gains tuning by using classical engineering techniques for practical online tuning of feedback gains.

**Key words:** simulation-based design, iterative learning control, nonlinear dynamic systems, learning controller, feedback controller

## 1. Introduction

Iterative learning control (ILC) is designed to improve the tracking performance of repetitive processes. The ILC is based on the idea that the information from previous trial is used to update the control law in order to obtain better performance of the assigned task in the next trial [1-5]. Some similar definitions of ILC are quoted by Ahn et al. in [1]. A common feature of them is the "repetition" of the assigned-task performance and the usage of the information of previous trial (passes, iterations, cycles or repetitions). Thus, the ILC is applicable for industrial robots working in a repeatable manner in the determined environment [1].

Like most of the existing control methods, the ILC can be categorized as linear (LILC) or nonlinear (NILC) iterative learning control. LILC is an ILC for linear systems [1,2] or linearized nonlinear systems [2]. If the linear approximation of a nonlinear dynamics results in great uncertainties, the corresponding LILC may fail to ensure the admissible tracking accuracy [3]. In this case, one should resort to nonlinear models and

---

The Author is with Institute of Mechanics, Bulgarian Academy of Sciences, "Acad.G.Bonchev" Str., bl.4, BG-1113 Sofia. E-mail: kamen@imbm.bas.bg

Received 27.06.2012.

nonlinear iterative learning control (NILC) [2-4,6-8]. In this paper, a nonlinear multiple-input-multiple-output (MIMO) dynamic model of a horizontal robot arm is considered.

On the other hand, the dynamic systems can be time-invariant or time-varying [7,9,10].

A classical offline ILC scheme for a MIMO plant with a feedback controller attached is depicted in Fig. 1, where: **P** represents the plant; **C** is a feedback controller;

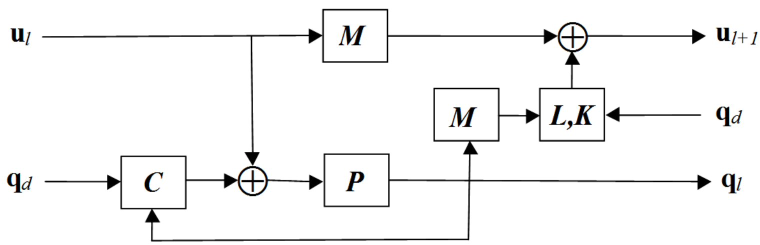


Figure 1. Learning control scheme with a feedback controller attached.

the vector of the input trajectory $\mathbf{u}_l$ is the feed-forward term of the control law; $\mathbf{q}_l$ is the actual vector of the output trajectory; $\mathbf{q}_d$ is the vector of the desired output trajectory. The learning controller $\{\mathbf{L},\mathbf{K}\}$ improves $\mathbf{u}_l$ by using the tracking error of the output trajectory $\mathbf{q}_d - \mathbf{q}_l$ multiplied by a learning gain matrix $\mathbf{K}$ (P-type learning) and/or the error derivative $\dot{\mathbf{q}}_d - \dot{\mathbf{q}}_l$ multiplied by a learning operator matrix $\mathbf{L}$ (D-type learning) [7]. If the learning operator is based on the dynamic model of the plant then the ILC can be specified as a model-based one [3,6]. In this study, we propose a model-based NILC with a feedback controller attached.

With respect to the learning controller (learning update law), ILC can be categorized as P-type, D-type, I-type, PI-type, PD-type, and PID-type (with respect to the state-space equations which describe the ILC scheme in Fig. 1) [1,9]. In similar way, a closed-loop ILC can be classified according to the type (P, PD, PID ...) of the attached feedback controller. Thus, the resultant controller-based classification of a closed-loop ILC can be P-P or P-PD, or PD-D, or any other combination of types of the learning and feedback controllers correspondingly [9]. In this work we present a PD-P type NILC.

Classical linear feedback control (PD or PID) has many applications in practice, but the linear PD/PID control of nonlinear and uncertain systems is not adequate for precise tracking [11,9]. Thus the combination of nonlinear feedback control and nonlinear iterative learning control is an effective approach to achieve good tracking performance [9]. On the other hand, a linear PD/PID controller for single-input-single-output (SISO) systems is easy to tune by using classical online techniques [12-14] because only 2-3 parameters have to be adjusted. Therefore, all efforts at minimization of the number of tuning parameters (learning and feedback gains) are reasonable. In this paper, we propose model-based nonlinear learning and feedback controller for robotic manipulators and this controller can be tuned by only four parameters so that classical engineering techniques can be used for simulation-based tuning of these parameters.

The main advantages of the offline closed-loop ILC for robotic manipulators are as follows. The ILC minimizes trajectory-tracking errors that arise from the unmodeled dynamics. The ILC procedure, can avoid closed-loop stability problems because at the end of the ILC procedure the feedback terms of the control input are getting reduced, so the feed-forward controller gets domination over the feedback controller [11]. The calculation of the ILC control update is offline and the maximum available information for the robot motion can be utilized. Anyway, the ILC applicability suffers from the following basic problems. An admissible number of iterations should produce the desired tracking accuracy. Therefore, the ILC procedure must be convergent with a high rate of convergence [6,9]. In spite of the small enough initial tracking error the ILC could fail due to non-monotonic convergence that results in a big transient error as it is shown in Fig. 2 [1,5,6,8,9,10,15].

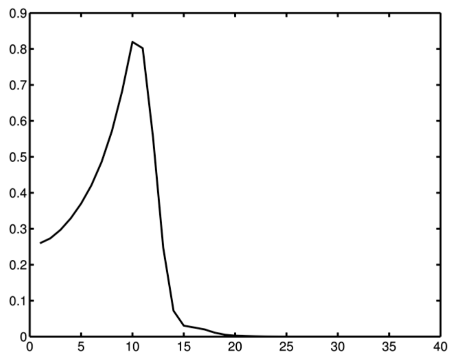


Figure 2. Maximal tracking error of TT3000 robot, $e_l^{\max}$ [rad] vs. iterations.

Computer simulations presented in [5,8,15] reveal the non-monotony of maximal tracking errors versus iterations, which leads to inapplicability of the corresponding ILC. Therefore, the monotonic convergence of the learning procedure can prevent the break-up of the mechanical system of the robot due to possible high overshoots of the desired trajectory [1]. With regard to the conditions for monotonic convergence of the ILC, numerous publications are available and most of them consider linear dynamic systems [1]. A monotonically convergent ILC for nonlinear systems (Hamiltonian systems) is proposed in [16], but an assumption for large enough feedback gains must be valid. Similar ideas of monotonically SPD-PD type ILC, for a specific class of nonlinear time-varying systems, based on increasing feedback gains are addressed in [9]. Unfortunately, this class of nonlinear systems is a subclass of systems defined by equation (1) and it can not describe the dynamic behaviour of robotic manipulators. Following these works and the ideas of accelerating the ILC convergence reported in [2], in this paper we present a simulation-based design of monotonically convergent PD type ILC for nonlinear time-varying systems. It has to be mentioned, that we propose an engineering approach to practical synthesis of a monotonically convergent ILC law while the general solution of the problem of transient errors (see Fig. 2) is presented in [8] and this solution does not concern the monotony of the ILC procedure.

Therefore, the main objective of the present study is the synthesis of nonlinear model-based PD-type learning controller and a P-type nonlinear feedback controller, for nonlinear time-varying dynamic systems of robotic manipulators. These controllers should have a minimum number of parameters to be tuned [5], so that classical techniques can be adopted for simple tuning of these parameters. Then the algorithm for simulation-based tuning of the synthesized controllers is proposed in order to ensure fast and monotonic convergence of the ILC process. The investigation of convergence rate and monotony of the synthesized ILC is based on the simulation of the dynamics of TT3000 SCARA-type robot of SEIKO Instruments Inc. Thus, the existence of monotonically convergent ILC algorithm, with high rate of convergence, can be proven using computer simulations.

This paper is organized as follows. Section 2 presents a synthesis of PD-P ILC for uncertain nonlinear time-varying systems that describe the dynamics of robotic manipulators. Section 3 presents the dynamic model of TT3000 SCARA-type robot as well as two sets of dynamic parameter estimations for a realistic computer simulation of the uncertain robot dynamics. The learning and feedback control gains of a PD-P type monotonic NILC of the TT3000 robot are determined from a computer simulation in Section 4. Conclusions are presented in Section 5.

## 2. Synthesis of PD-P type ILC for robotic manipulators

In this section we present the main result of this paper: a model-based nonlinear learning controller (learning update law) and a model-based nonlinear feedback controller (feedback control term) for robotic manipulators. These controllers are easy to tune, so classical techniques [12-14] for tuning SISO PD feedback linear controllers can also be used for synthesized MIMO nonlinear controllers.

We consider a nonlinear MIMO dynamic model of a robot based on the Lagrange's formulation of equations of motion in the space of generalized coordinates:

$$\mathbf{A}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{b}(\mathbf{q},\dot{\mathbf{q}}) + \mathbf{D}\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) + \mathbf{f} = \mathbf{u}, \tag{1}$$

where: $\mathbf{q} \equiv \mathbf{q}(t)$, $t \in [0,T]$, is the $n \times 1$ vector of generalized coordinates (joint angles), $q_i \in [Q_i^{\min}, Q_i^{\max}]$, $i = 1,\ldots,n$; $\mathbf{A}(\mathbf{q})$ is the $n \times n$ symmetric positive-definite inertia matrix; the $n \times 1$ vector $\mathbf{b}(\mathbf{q},\dot{\mathbf{q}})$ takes into account the Coriolis and centrifugal torques; $\mathbf{D} = \mathrm{diag}\{\delta_1, \ldots, \delta_n\}$ denotes the diagonal $n \times n$ matrix of the coefficients of viscous friction; $\mathbf{g}(\mathbf{q})$ is the $n \times 1$ vector representing gravity torques; $\mathbf{f} = [f_1 \ \mathrm{sign}(\dot{q}_1), \ \ldots, \ f_n \ \mathrm{sign}(\dot{q}_n)]^T$ is the vector of coefficients of Coulomb friction and $\mathbf{u} = \mathbf{u}_l + \mathbf{u}_c$ is the $n \times 1$ vector of generalized torques where $\mathbf{u}_l \equiv \mathbf{u}_l(t)$ and $\mathbf{u}_c \equiv \mathbf{u}_c(\mathbf{q},\dot{\mathbf{q}},t)$ are feed-forward and feedback terms, respectively. The allowable set of generalized torques is a rectangular hyper-parallelepiped: $u_i \in [-U_i^{\max}, U_i^{\max}]$. The state-space form of Eq. (1) describes the learning scheme in Fig. 1 for robot manipulators [7].

The synthesis of PD-P ILC consists of three steps: first, synthesis of an update control law, then, synthesis of a feedback control law, and finally, specification of a learning operator.

We assume the following notations: $h(\cdot) = h$ and $H(\cdot,\cdot) = H$.

For the nonlinear MIMO dynamic model in (1), we propose the following nonlinear PD update control law:

$$\mathbf{u}_{l+1} = \mathbf{u}_l + \mathbf{L}[\ddot{\mathbf{q}}_d - \ddot{\mathbf{q}}_l + L_v(\dot{\mathbf{q}}_d - \dot{\mathbf{q}}_l) + L_p(\mathbf{q}_d - \mathbf{q}_l)], \tag{2}$$

where: $l = 0, 1, ..., N$, $\mathbf{q}_d$ is an attainable and desired trajectory, and $\mathbf{q}_l$ is the output trajectory at the $l$th iteration; $\mathbf{L} = \mathbf{L}(\mathbf{q}_{l+1}(t))$, is a learning operator; $\mathbf{u}_0 = \mathbf{u}_0(t)$ is the initial feed-forward control input; $t : t \in [0, T]$ is the tracking time and $[0,\ T]$ is the robot tracking time interval; $L_v$ and $L_p$ are learning control gains.

Then, we consider the following stabilizing feedback control term proposed in [17]

$$\mathbf{u}_c = \hat{\mathbf{A}}[\ddot{\mathbf{q}}_d + K_v(\dot{\mathbf{q}}_d - \dot{\mathbf{q}}_l) + K_p(\mathbf{q}_d - \mathbf{q}_l)] + \hat{\mathbf{b}} + \hat{\mathbf{D}}\dot{\mathbf{q}}_l + \hat{\mathbf{g}} + \hat{\mathbf{f}}, \tag{3}$$

where $K_v$ and $K_p$ are the feedback gains, and $\hat{\mathbf{A}}$, $\hat{\mathbf{b}}$, $\hat{\mathbf{D}}$, $\hat{\mathbf{g}}$, and $\hat{\mathbf{f}}$ are the corresponding estimates of $\mathbf{A}$, $\mathbf{b}$, $\mathbf{D}$, $\mathbf{g}$, and $\mathbf{f}$ in (1). In [17] it is proven that $K_v$ and $K_p$ could be positive constants to ensure stability of the feedback controller.

The following sufficient condition for robustness and convergence of the considered update control law (2), for robotic manipulators is proven in [7]:

$$\left\| \mathbf{I} - \mathbf{L}\mathbf{A}^{-1} \right\| \leqslant \rho < 1, \tag{4}$$

where $\mathbf{I}$ is the identity matrix of size $n$; $\mathbf{A}$ is the inertia matrix in the dynamic equations of motion (1); $\mathbf{L}$ is the learning operator that is to be specified, and $||...||$ is the Euclidean matrix norm.

Unfortunately, a learning operator that satisfies the sufficient condition (4) cannot be obtained directly from the inequality (4) because the inertia matrix $\mathbf{A}$ of an actual robot is not exactly known and only an estimation $\hat{\mathbf{A}}$ of $\mathbf{A}$ could be available through identification [18,19]. That is why, following Arimoto's ideas from [2] for better convergence rate, we consider a learning operator to be as close as possible to the inertia matrix. Therefore, we propose the learning operator to be identically equal to $\hat{\mathbf{A}}$, i.e. $\mathbf{L} \equiv \hat{\mathbf{A}}$ [15,6]. From (4) we obtain the sufficient condition for convergence in case of $\mathbf{L} \equiv \hat{\mathbf{A}}$.

$$\left\| \mathbf{I} - \hat{\mathbf{A}}\, \mathbf{A}^{-1} \right\| < 1. \tag{5}$$

Now we are in a position to present the simulation-based design of the learning and feedback control gains of the synthesized in equations (2) and (3) update and feedback control laws, correspondingly. Before specifying the values of the learning and feedback gains to ensure high rate monotonic convergence for the proposed PD-P ILC, we are going to describe in the next section the dynamic model of considered robot manipulator – TT3000 of SEIKO Instruments Inc.

## 3. Dynamic model of TT3000 robot

In this paper, we consider a dynamic model of TT-3000 SCARA-type robot of SEIKO Instruments Inc. [18]. The kinematic scheme is shown in Fig. 3.

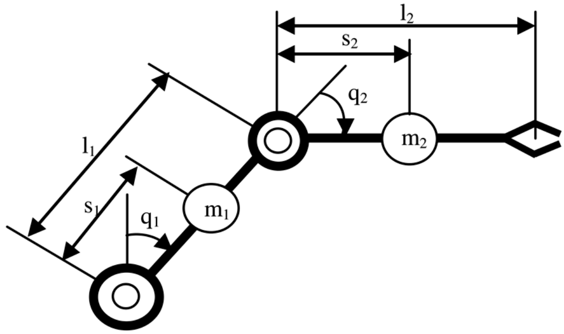


Figure 3. Kinematic scheme of SCARA-type horizontal robot arm.

### 3.1. Equations of motion

The Lagrange's formulation of equations of motion for this robot is:

$$\mathbf{A}\ddot{\mathbf{q}} + \mathbf{b} + \mathbf{D}\dot{\mathbf{q}} + \mathbf{f}_c = \mathbf{u}, \tag{6}$$

where [18]

$$\mathbf{A} = \begin{bmatrix} a_{11}(\mathbf{q}) & a_{12}(\mathbf{q}) \\ a_{12}(\mathbf{q}) & a_{22}(\mathbf{q}) \end{bmatrix}, \quad \mathbf{b} = [-\beta \sin q_2(2\dot{q}_1\dot{q}_2 + \dot{q}_2^2), \beta\, \dot{q}_1^2 \sin q_2]^{\mathrm{T}},$$
$$\mathbf{D} = \begin{bmatrix} d_1 & 0 \\ 0 & d_2 \end{bmatrix}, \quad \mathbf{f}_c = [f_{c1}\mathrm{sign}\,(\dot{q}_1), f_{c2}\mathrm{sign}\,(\dot{q}_2)]^{\mathrm{T}}, \quad \mathbf{u} = [u_1, u_2]^{\mathrm{T}}, \tag{7}$$
$$\mathbf{q} = [q_1, q_2]^{\mathrm{T}}, \quad \dot{\mathbf{q}} = [\dot{q}_1, \dot{q}_2]^{\mathrm{T}}, \quad \ddot{\mathbf{q}} = [\ddot{q}_1, \ddot{q}_2]^{\mathrm{T}},$$

$$\begin{aligned} a_{11}(\mathbf{q}) &= \alpha + 2\beta\cos q_2, & \alpha &= I_1 + m_1 s_1^2 + I_2 + m_2(l_1^2 + s_2^2), \\ a_{12}(\mathbf{q}) &= \chi + \beta\cos q_2, & \beta &= m_2 l_1 s_2, \\ a_{22}(\mathbf{q}) &= \chi, & \chi &= I_2 + m_2 s_2^2, \end{aligned} \tag{8}$$

and $d_i$, $i = 1, 2$ is viscous friction, $f_{ci}$ is Coulomb friction, $l_i$ is the length of the link given in the specifications of the robot, $s_i$ is the position of the centre of the mass, $m_i$ is the total mass of the link, and $I_i$ is the inertia of the link about its centre of the mass, $\mathbf{u}$ is the vector of generalized torques, $\mathbf{u} = \mathbf{u}_l + \mathbf{u}_c$, and $\mathbf{u}_l$, $\mathbf{u}_c$ are the feedforward and feedback terms described by (2) and (3) respectively.

### 3.2. Model parameters

If the model parameters α, β, χ and the viscous friction $d_i$, $i = 1, 2$, and the Coulomb friction $f_{ci}$ are estimated, the inertia matrix $\mathbf{A}(\mathbf{q}) = [a_{ij}]$, $i, j = 1,2$ could be calculated and the robot dynamics could be simulated through numerical solution of the associated differential equations given in (6)–(8).

For the realistic computer simulation of the PD-P NILC procedure, proposed in the next section, we are going to use two sets of model parameters of the TT3000 robot which are reported in [18]. The first set of model parameters, called the standard ones, is provided by the manufacturer. We use this set to calculate the left-hand side of (6) (the virtual robot arm of TT-3000). The second set of dynamic parameters is estimated in [18] by means of an effective identification method. We use it to calculate the control input $\mathbf{u}$ (the right-hand side of (6)). Using inaccurate values of estimated dynamic parameters (uncertain robot dynamics) the simulation of a learning control of robot motion corresponds to a great extent to the same learning process of an actual robot.

Table 1 shows the comparison between experimentally estimated values of the model parameters $\hat{\alpha}$, $\hat{\beta}$ and $\hat{\chi}$ [18], and the assumed standard values of α, β and χ.

Table 9. Estimated and standard values of the model parameters.

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Standard parameters** $\times 10^{-2}$ | $\alpha$ [Nm$^2$] | $\beta$ [Nm$^2$] | $\chi$ [Nm$^2$] | $d_1$ [Nms] | $d_2$ [Nms] | $f_{c1}$ [Nm] | $f_{c1}$ [Nm] |
| | 65.962 | 15.056 | 14.534 | 17.29 | 12.49 | 6.46 | 3.44 |
| **Experimental estimations** $\times 10^{-2}$ | $\hat{\alpha}$ [Nm$^2$] | $\hat{\beta}$ [Nm$^2$] | $\hat{\chi}$ [Nm$^2$] | $\hat{d}_1$ [Nms] | $\hat{d}_2$ [Nms] | $\hat{f}_{c1}$ [Nm] | $\hat{f}_{c1}$ [Nm] |
| | 65.553 | 14.718 | 18.478 | 17.29 | 12.49 | 8.13 | 5.51 |

## 4. Simulation results

In this section we present simulation-based tuning of the learning control gains in (2), $(\mathbf{L} \equiv \hat{\mathbf{A}})$, and the feedback gains given in (3) to obtain monotonic, high-rate convergence of the proposed PD-P NILC for TT3000 robotic manipulator. All simulation programs were written in MATLAB-6 R12 and executed on a personal computer with Intel® Core™ Duo CPU E8500@3.16GHz. Suppose that a desired trajectory is defined in the space of generalized coordinates as follows:

$$
\begin{aligned}
q_1^d(t) &= -0.25\cos(2t) + 0.25 \\
q_2^d(t) &= -0.5\cos(t) + 0.5
\end{aligned}
\tag{9}
$$

where $t \in [0,\ 2\pi]$.

Given the desired trajectory in (9) and using the standard model parameters and the parameter estimations described in Tab. 1, from equations (6), (2) and (3) we obtain the differential equations of the model-based PD-P NILC controller for TT3000 robot

$$\begin{aligned}&\mathbf{A}(\mathbf{q}_{l+1})\ddot{\mathbf{q}}_{l+1} + \Delta\mathbf{b} + \Delta\mathbf{D}\dot{\mathbf{q}}_{l+1} + \Delta\mathbf{f} = \mathbf{u}_l + \hat{\mathbf{A}}(\mathbf{q}_l)(\ddot{\mathbf{e}}_l + L_v\dot{\mathbf{e}}_l + L_p\mathbf{e}_l)\\ &+\hat{\mathbf{A}}(\mathbf{q}_{l+1})(\ddot{\mathbf{q}}_d + K_v\dot{\mathbf{e}}_{l+1} + K_p\mathbf{e}_{l+1})\end{aligned} \tag{10}$$

where: $l = 0,1,\ldots,N$ is the iteration number; $\Delta\mathbf{A} = \mathbf{A}(q_2^{l+1},\alpha,\ \beta,\ \chi) - \hat{\mathbf{A}}(q_2^{l+1},\hat{\alpha},\ \hat{\beta},\hat{\chi})$ and $\Delta\mathbf{b} = \mathbf{b}(q_1^{l+1},q_2^{l+1},\alpha,\ \beta,\ \chi) - \hat{\mathbf{b}}(q_1^{l+1},q_2^{l+1},\hat{\alpha},\ \hat{\beta},\hat{\chi})$; $\Delta\mathbf{D} = \mathbf{D}(d_1,d_2) - \hat{\mathbf{D}}(\hat{d}_1,\hat{d}_2)$, and $\Delta\mathbf{f} = \mathbf{f}(q_1^{l+1},q_2^{l+1},f_{c1},f_{c2}) - \hat{\mathbf{f}}(q_1^{l+1},q_2^{l+1},\hat{f}_{c1},\hat{f}_{c2})$. The trajectory-tracking error is defined as $\mathbf{e}_l : \mathbf{e}_l = \mathbf{q}_d(t) - \mathbf{q}_l(t),\ t \in [0,2\pi]$. As the basic postulates of classical ILC described in [1] are required, the repetition of the initial setting is satisfied and therefore the initial conditions are: $q_1^l(0) = q_1^d(0) = 0$, $q_2^l(0) = q_2^d(0) = 0$, $\dot{q}_1^l(0) = \dot{q}_1^d(0) = 0$, and $\dot{q}_2^l(0) = \dot{q}_2^d(0) = 0$. Also we assume that $\mathbf{u}_0(t) \equiv 0$. One of the advantages of using numerical simulation is the ability to track the desired trajectory $\mathbf{q}_d$ assuming that there are no limits on $\mathbf{q}_l$ and $\mathbf{u}_l$. Further, we will use this assumption to examine large tracking errors.

First of all, it has to be proven that the above proposed PD-P iterative learning algorithm is uniformly asymptotically convergent. Therefore, using equations (7) and (8), and model parameters in Tab. 1, and assuming that $z = \cos q_2,\ z \in [-1,1]$ from the condition given by inequality (5) we obtain the sufficient condition for convergence of the proposed learning procedure (Eq. (10)) for TT3000 robot:

$$\left\|\mathbf{I} - \hat{\mathbf{A}}(z,\hat{\alpha},\ \hat{\beta},\hat{\chi})\ \mathbf{A}^{-1}(z,\alpha,\ \beta,\ \chi)\right\| < 1. \tag{11}$$

The calculation of the left hand side of the inequality (11) for $z \in [-1,1]$ is depicted in Fig. 4.

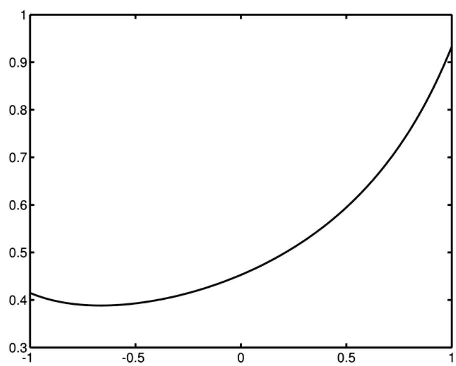


Figure 4. Graph of $\left\|\mathbf{I} - \hat{\mathbf{A}}(z)\ \mathbf{A}^{-1}(z)\right\|$, $z = \cos(q_2), z \in [-1,1]$.

Referring to the curve in Fig. 4 it can be seen that the condition (11) holds and the existence of an asymptotically convergent PD-P NILC control law for TT3000 robot, Eq. (10), is proven. It has to be mentioned that this convergence does not depend on the values of the learning and the feedback gains given in (10).

Now we are aimed to prove the existence of a monotonic convergence of the proposed PD-P NILC control algorithm in (10). For this purpose we will tune (by simulation) the corresponding learning and feedback gains in order to ensure a high rate of monotonic convergence of the learning algorithm.

Let us define maximal tracking error as:

$$e_l^{\max} = \max_t x \left\| \mathbf{e}_l(t) \right\|, \ \mathbf{e}_l(t) = \mathbf{q}_d(t) - \mathbf{q}_l(t), t \in [0,T], \ l = 1,...,N$$

and the maximal error of the iterative learning procedure:

$$e^{\max} = \max_l x(e_l^{\max}), \ l = 1,...,N$$

Thus, the examination of $e_l^{\max}$ in iterations reveals the convergence of the learning algorithm and therefore, we shall investigate the monotony and the rate of the ILC convergence by varying the learning and feedback gains and calculating the corresponding maximal errors $e_l^{\max}$.

Assuming $\mathbf{u}_0(t) \equiv 0$ and $\mathbf{u}_c(\mathbf{q}_0, \dot{\mathbf{q}}_0, t) = \hat{\mathbf{A}}(\mathbf{q}_0)(\ddot{\mathbf{q}}_d + K_v\dot{\mathbf{e}}_0 + K_p\mathbf{e}_0) + \hat{\mathbf{b}} + \hat{\mathbf{D}}\dot{\mathbf{q}}_0 + \hat{\mathbf{g}} + \hat{\mathbf{f}}$ we specify the initial maximal error $e_0^{\max}$, as a result of the initial trajectory tracking. If values of $K_v$ and $K_p$ are not specified properly, a significant maximal initial tracking error is expected. That is why, varying values of $K_v$ and $K_p$, we can examine the initial error $e_0^{\max}$ by simulation of the initial tracking. In similar way we can vary the learning gains $L_v$ and $L_p$ in order to investigate the convergence of $e_l^{\max}$ by simulating the PD-P NILC process.

The computer simulation of the PD-P learning process (Eq. (10) for the virtual TT3000 robot consists of the following steps:

- Solution of the differential equation of motion (Eq. (10)) to obtain the resultant output trajectory;
- Update of the control input according to the learning update law, (Eq. (2));
- Increment the iteration $l = l + 1$ number until the specified maximum $l = N$ is reached.

Based on the above analysis, we present the following algorithm for simulation-based tuning of the learning and feedback gains of the PD-P learning controller (Eq. (10) to achieve an acceptable initial error and fast monotonic convergence of $e_l^{\max}$:

- Solve the differential equation of motion (3) with initial $\mathbf{u}_0(t)$, $\mathbf{u}_c(\mathbf{q}_0, \dot{\mathbf{q}}_0, t)$ and $K_v = K_p = 0$ to examine the value of the initial error $e_0^{\max}$, and equation (10) with $K_v = K_p = 0$, and $L_v = L_p = 0$ to calculate at each iteration $e_l^{\max}$, and the convergence of the maximal tracking error in the iteration domain;

- If the initial error $e_0^{\max}$ is unacceptable, stepwise increase the values of the feedback gains following the ideas reported in [12]:
  - First, increase $K_p > 0$ ($K_v = 0$ and $L_v = L_p = 0$), obtain the output trajectory $\mathbf{q}_l(t), t \in [0, T]$ and compute the maximal tracking error $e_l^{\max}$ at each iteration $l = 0, 1, ..., N$ ($N$ depends on the desired convergence accuracy). Make your decision to repeat the simulation or to stop it depending on the simulation results for reduction of $e_0^{\max}$.
  - If no satisfactory reduction of $e_0^{\max}$ is obtained, increase $K_v > 0$, ($L_v = L_p = 0$) and compute $e_l^{\max}$ in iterations. If needed, repeat the simulation (Eq. (3) and Eq. (10)) until an acceptable initial error $e_0^{\max}$ is achieved.
- If the resultant error $e_l^{\max}$ is not monotonic convergent with the iteration $l = 0, 1, ..., N$ and the learning error $e^{\max}$ is inadmissible, then increase, in the same way as described in the last two steps, the learning gains in order to achieve fast monotonic convergence of the considered PD-P NILC algorithm.

The implementation of the first step of the above tuning algorithm results in the state space trajectory at the first iteration and in the corresponding trajectory at the twentieth iteration which are shown in Fig. 5a together with the desired trajectory.

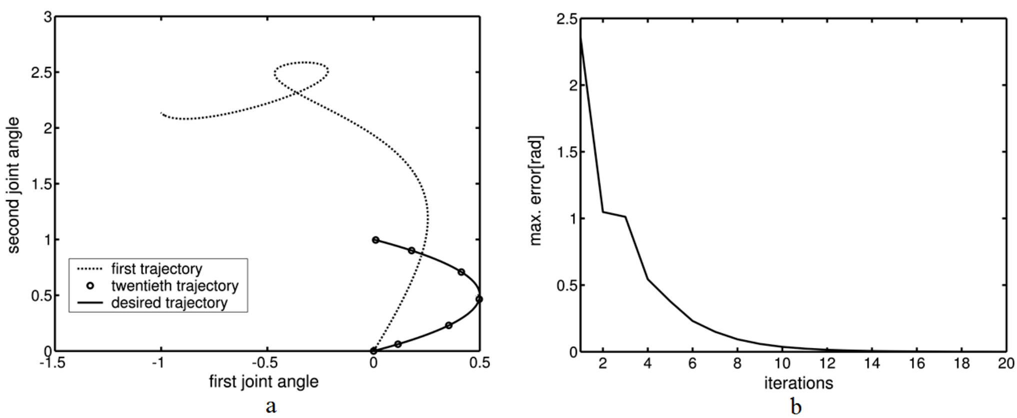


Figure 5. a) The initial, twentieth, and desired trajectory; b) Convergence of the maximal tracking error $e_l^{\max}$ for $K_v = 0$, $L_v = 0$ and $K_p = 0$, $L_p = 0$.

In this case, the simulation result for the maximal tracking error $e_l^{\max}$ in iterations ($l = 0, 1, ..., 20$) is depicted in Fig. 5b. Obviously, the convergence rate is very good and $e_l^{\max}$ is monotonically decreasing but the initial error $e_0^{\max}$ is unacceptable. Therefore, in the second step of the tuning algorithm the increasing of the feedback gain $K_p > 0$ ($K_v = 0$ and $L_p = L_v = 0$) results in the graphs shown in Fig. 6a.

Referring to Fig. 6a one may conclude that the reduction of $e_0^{\max}$ is not sufficient and $e_0^{\max}$ reaches its minimum for $K_p \approx 1$. In addition significant transient errors $e^{\max}$

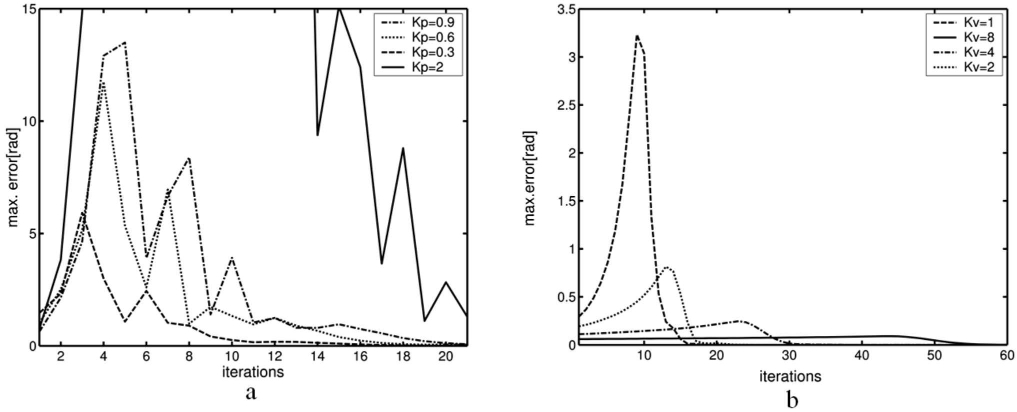


Figure 6. a) The profiles of maximal tracking error $e_l^{\max}$ for $K_v = 0$, $L_v = 0$; and $K_p = 0.3,\ 0.6,\ 0.9,\ 2$, $L_p = 0$ b) The profiles of maximal tracking error for $K_v = 1,\ 2,\ 4,\ 8$, $L_v = 0$ and $K_p = 0$, $L_p = 0$.

exist due to the non monotonic convergence of $e_l^{\max}$ versus iterations. Anyway, in this case, the convergence rate of $e_l^{\max}$ is similar to the one shown in Fig. 5b. Note that this examination is possible only by computer simulation because such large learning errors $e^{\max}$, as shown in Fig. 6a, are inadmissible for a real robot due to the limits of the generalized coordinates $\mathbf{q}_{l+1}(t)$, $t \in [0, T]$. Thus, we fix the value of $K_p = 1$ and continuing the tuning algorithm we increase the feedback gain $K_v => 0$ ($L_v = L_p = 0$), which yields the graphs plotted in Fig. 6b. Here we see a very good reduction of the initial error $e_0^{\max}$, so we assume that $e_0^{\max} = 0.06028$[rad] for $K_p = 1$ and $K_v = 8$ ($L_v = L_p = 0$) satisfies the desired tracking accuracy in the first iteration. Note that in this case, the convergence of $e_l^{\max}$, $l = 0, 1, \ldots, 80$, is not monotonic and the convergence rate is not so good as the one shown in Fig. 5b. Thus, we can increase the learning gain $L_p > 0$ ($L_v = 0$, $K_p = 1$, $K_v = 8$) in order to ensure fast monotonic convergence of the considered PD-P learning algorithm (Eq. (10)). The simulation results for $e_l^{\max}$, $l = 0, 1, \ldots, 60$, and $L_p = 1$, $L_p = 2$, $L_p = 4$, and $L_p = 8$ ($L_v = 0$, $K_p = 1$, $K_v = 8$) are shown in Fig. 7a).

The examination of Fig. 7a reveals that the convergence of $e_l^{\max}$ is not monotonic and the maximal learning errors $e_l^{\max}$ for $L_p = 4$, and $L_p = 8$ exceeds the initial error $e_0^{\max}$, and consequently, it is reasonable to assume that $L_p = 2$.

Finally, we have to increase step-by-step the learning gain $L_v > 0$ ($L_p = 2$, $K_p = 1$, $K_v = 8$). The resultant graphs of $e_l^{\max}$, $l = 0, 1, \ldots, 40$, where $L_v = 1$, $L_v = 2$, $L_v = 4$ and $L_v = 8$, are shown in Fig. 7b.

Given feedback gains $K_p = 1$ and $K_v = 8$, and learning gains $L_p = 2$ and $L_v = 8$, the corresponding curve of $e_l^{\max}$ shows very fast monotonic convergence of the maximal tracking error versus iterations. The final accuracy at iteration 40 is $e_{40}^{\max} = 3.6057e-6$, taking into account that the accuracy of the Runge-Kutta method for all computer simulations is $1.0e-6$. Thus, the implementation of the tuning algorithm for simulation-based

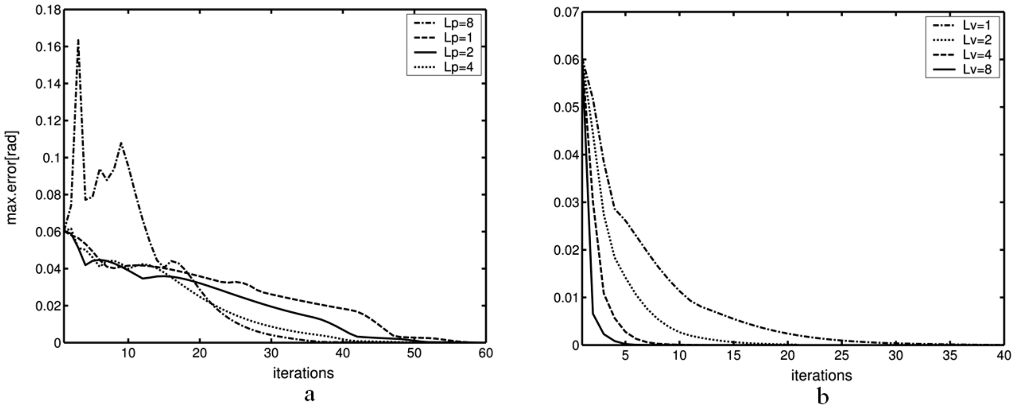


Figure 7. a) The profiles of maximal tracking error for $K_v = 8$, $L_v = 0$ and $K_p = 1$, $L_p = 1, 2, 4, 8$; b) The profiles of maximal tracking error for $K_v = 8$, $L_v = 1, 2, 4, 8$ and $K_p = 1$, $L_p = 2$.

design of monotonically convergent PD-P NILC (Eq. (10)) for TT3000 robotic manipulator is completed by deriving the above values of the learning and feedback gains.

## 5. Conclusions

We came to the following conclusions:

- A model-based PD-P type (with respect to the state state-space form of Eq. (1)) nonlinear iterative learning control for robotic manipulators is proposed. The PD-P ILC parameter design consists of two steps: specification of the learning operator **L** (for D-type ILC) and design of the learning (for P-type ILC), and feedback (for P-type feedback) gains.

- The learning operator is specified as equal to an estimate of the inertia matrix in the dynamic equations of the robot motion, $(\mathbf{L} \equiv \hat{\mathbf{A}})$, and the uniform asymptotic convergence of the proposed PD-P learning algorithm for TT3000 robot (equation (10)) is proven by numerically validation of the corresponding sufficient condition (11).

- The constant learning gains are multiplied by **L** in the learning update law (2) which implies a nonlinear learning controller and the constant feedback gains are multiplied by $\hat{\mathbf{A}}$ in the feedback control law (3) which implies a nonlinear feedback controller. So, the PD-P ILC tuning depends on only four positive gains: $K_p$ and $K_v$ of the feedback controller, $L_p$ and $L_v$, and of the learning one. The number of these gains does not depend on the system dimension. Thus, classical engineering techniques can be used for tuning by simulations of both controllers. Moreover, in the case of excessive growth of transient errors (see Fig. 6a), the computer simulation of the proposed learning procedure is the only way for learning gains tuning.

- A simulation-based design of the feedback and the learning gains, respectively, is performed for TT3000 robot. Thus, it was shown by numerical simulation that a monotonically convergent model-based nonlinear PD-P NILC for TT3000 SCARA-type robot exists and in this case the convergence rate is very high.

- The existence of powerful robot control devices equipped with high-quality displays and accurate CAD-oriented methods for identification of model parameters ensures the applicability of the proposed simulation-based approach for design of monotonically-convergent nonlinear iterative learning controllers.

## References

[1] H.-S. AHN, K. MOORE and Y. CHEN: Iterative learning control robustness and monotonic convergence for interval systems. Springer-Verlag London Limited, London, 2007.

[2] S. ARIMOTO, S. KAWAMURA and F. MIYAZAKI: Bettering operation of dynamic systems by learning: A new control theory for servomechanism of mechatronics systems. *Proc. of 23rd Conf. on Decision and Control*, (1984), 1064-1068.

[3] K. SMOLDERS, M. VOLCKAERT and J. SWEVERS: Tracking control of nonlinear lumped mechanical continuous-time systems: A model-based iterative learning approach. *Mechanical Systems and Signal Processing*, **22**(8), (2008), 1896-1916. doi:10.1016/j.ymssp.2008.03.004

[4] J.-X. XU and Y. TAN: Linear and nonlinear iterative learning control. (Lecture Notes in Control and Information Sciences). Springer, New York, 2003.

[5] R. LONGMAN: Iterative learning control and repetitive control for engineering practice. *Int. J. of Control*, **73**(10), (2000), 930-954.

[6] K. DELCHEV and E. ZAHARIEV: Computer simulation based synthesis of learning control law of robots. *Mechanics based design of structures and machines*, **36**(3), (2008), 225-248.

[7] D. HEINZINGER, B. FENWICK, B. PADEN and F. MIYAZAKI: Robust learning control. *Proc. of 28th Conf. on Decision and Control*, (1989), 436-440.

[8] K. DELCHEV: Iterative learning control for nonlinear systems: A bounded-error algorithm. *Asian J. of Control*, (2012). DOI: 10.1002/asjc.554

[9] P. OUYANG, B. PETZ and F. XI: Iterative learning control with switching gain feedback for nonlinear systems. *Computational and Nonlinear Dynamics*, **6**(1), (2011), 011020-1 - 011020-7.

[10] D. BRISTOW and J. SINGLER: Robustness analysis of slow learning in iterative learning control systems. *Proc. of American Control Conference*, (2011), 3669-3673.

[11] S. GOPINATH and I. KAR: Iterative learning control scheme for manipulators including actuator dynamics. *Mechanism and Machine Theory*, **39** (2004), 1367-1384.

[12] T. BULLOCK: Adjusting PID gains. *Motion Control Magazine*, (1991), 17-18. Available at:
http://www.ctc-control.com/customer/elearning/registered/servoBasicsForTheLayman.pdf

[13] K. ANG, G. CHONG and Y. LI: PID control system analysis, design, and technology. *IEEE Trans. on Control Systems Technology*, **13**(4), (2005), 559-576.

[14] C. SMITH and A. CORRIPIO: Principles and practices of automatic process control. (3rd Edition). John Wiley & Sons Inc., New York, 2006.

[15] K. DELCHEV and G. BOYADJIEV: Synthesis and computer simulation of learning control of horizontal robot arm. *Proc. of 9th IFAC Symp. on Robot Control "SYROCO'09"*, (2009), 301-306.

[16] K. FUJIMOTO and T. SUGIE: Iterative learning control of hamiltonian systems: I/O based optimal control approach. *IEEE Trans. on Automatic Control*, **48**(10), (2003), 1756-1761.

[17] V. KOZLOV, V. MAKARYCHEV, A. TIMOFEEV and E. YUREVICH: Dynamics of robot control. Science, Moscow, 1984. (in Russian)

[18] W. SHINJI and T. MITA: A parameter identification method of horizontal robot arms. *Advanced Robotics*, **4**(4), (1990), 337-352.

[19] B. ARMSTRONG, O. KHATIB and J. BURDICK: The explicit dynamic model and inertial parameters of the PUMA 560 arm. *Proc. of IEEE Int. Conf. Robotics and Automation*, **1** (1986), 510-518.