OPEN CLOSED-LOOP PD^{μ} / PD TYPE ILC CONTROL OF NEUROARM ROBOTIC SYSTEM

Boško P. Cvetković[1], Mihailo P. Lazarević[1], Petar D. Mandić[1], Tomislav B. Šekara[2], Paolo Lino[3]

[1] Faculty of Mechanical Engineering,
University of Belgrade, Kraljice Marije 16, Belgrade, Serbia
e-mail: boskocvetkovic@gmail.com, pmandic@mas.bg.ac.rs, mlazarevic@mas.bg.ac.rs
[2] School of Electrical Engineering,
University of Belgrade, Bulevar kralja Aleksandra 73, Belgrade, Serbia
e-mail: tomi@etf.rs
[3] Department of Electrical and Information Engineering,
Polytechnic University of Bari, Via E. Orabona 4, Bari, Italy
e-mail: paolo.lino@poliba.it

Abstract:

In this paper, an advanced iterative learning control algorithm (ILC) is introduced in order to solve the output tracking problem of a robotic manipulator with three degrees of freedom. Iterative fractional order PD^{μ} type control is located in feedforward path, and combined together with classical feedback PD controller. Fractional derivative μ provides additional flexibility in adjusting the output performances. Parameters of the feedback controller are derived using a modern approach which takes into consideration both performance and robustness characteristics of the closed loop system. This is achieved by a suitable selection of only one adjustable parameter. Mathematical model of robotic manipulator is derived together with actuator dynamics, and it is shown that in the presence of high gear ratio the nonlinear effects can be neglected, and only linear model can be observed. The efficiency of the proposed control algorithm is demonstrated by simulations, in which robotic arm needs to follow desired trajectory given in joint space. Excellent tracking performances are achieved only after few iterations.

Key words: robot control, iterative learning control, fractional calculus, PD control

1. Introduction

Robotics, as a relatively young and multidisciplinary field of modern technology, requires knowledge of electrical, mechanical and systems engineering. Rapid development of robotics over the past decades is mostly caused by advances in computer and sensor technology, as well as theoretical advances in control theory [1]. Majority of robot applications deal within industrial conditions, accomplishing tasks such as welding, packaging, cutting, paint spraying, moving objects etc. So, industrial manipulators need to fulfill high demands in terms of accuracy, precision and repeatability.

Designing such a controller is a complex task. Scientific community reports a large number of different control strategies for robotic manipulators [2]. Most of them are based on linear control theory from a practical reason. Namely, linear controllers are intuitive, easy to understand and last but not least, easy to implement. This approach gains even more importance when robot dynamics can be regarded as linear, which can be achieved by using motors with high gear ratios, or by using feedback linearization techniques. This way, dynamic coupling effects can be neglected and control of each robotic joint can be designed independently. This is the reason why classical PID controllers are still an inevitable part of industrial control of robot manipulators [3,4].

Besides classical control strategies, improving system performances can be achieved by applying intelligent control techniques [5]. One of these is iterative learning control (ILC), which has recently attracted scientists' attention as perspective field in robotic control [6]. Namely, ILC tries to emulate human learning using the trial-error concept i.e., knowledge from the previous trial is used to adjust control variable for the current trial. That way it can iteratively improve systems response over a finite time interval [7]. However, ILC as a simple offline feedforward technique cannot effectively suppress unknown disturbances. To accomplish that, ILC is combined with classical feedback control or, as in our case, with proportional-derivative (*PD*) control.

On the other side, fractional calculus, as a theory of integrals and derivatives of non-integer order [8], can be used to additionally improve control system performances. It is intuitively clear that non-integer controllers are more flexible comparing to integer order counterparts. Basically, fractional order (FO) controllers with few tuning parameters can achieve same robustness and performance characteristics as the classical high order controller. In the literature, some of the most common FO controllers are CRONE controller [9], $PI^{\lambda}D^{\mu}$ controller [10], and fractional lead/lag compensator [11].

In this paper, a combination of advanced ILC with classical feedback control scheme is used for the control of a 3 degrees of freedom (3DOFs) robot manipulator. More specifically, feedforward PD^{μ} type of ILC is used together with classical *PD* feedback control in order to obtain better output performances just after few iterations. Nonlinear mathematical model of robotic manipulator is presented in Section 2, which, because of a high gear ratio, is reduced to a linear model. Section 3 deals with the design of the aforementioned controller. Section 4 presents some simulation results, and section 5 concludes the paper.

2. Mathematical model of robotic manipulator with actuator dynamics

In this paper we consider a NeuroArm robotic system with seven DOFs, which is an integral part of the Laboratory of Applied Mechanics at the Faculty of Mechanical Engineering in Belgrade (Figure 1). The first three revolute joints are responsible for setting the end-effector into the demanded position, while the following three joints form the spherical wrist and achieve the end-effector's orientation. The 7th DOF is the gripper. In this paper, we consider only a 3-DOF NeuroArm with the first three revolute joints.

The mechanical structure of a NeuroArm robot could be considered as a sequence of rigid bodies (or links) interconnected by means of joints. Dynamic equations of the robotic system can be written in the following form:

$$A(\boldsymbol{q})\ddot{\boldsymbol{q}} + C(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} - \boldsymbol{Q}^{\boldsymbol{g}} = \boldsymbol{Q}^{\boldsymbol{m}}, \tag{1}$$

wherein: $\boldsymbol{q}(t) \in R^3$ is the vector of the generalized coordinates, $A(\boldsymbol{q}) \in R^{3\times 3}$ represents basic metric tensor (or inertia matrix), $C(\boldsymbol{q},\dot{\boldsymbol{q}}) \in R^{3\times 3}$ is a matrix that includes centrifugal and Coriolis

effects, $\boldsymbol{Q}^{\boldsymbol{g}} \in R^3$ and $\boldsymbol{Q}^{\boldsymbol{m}} \in R^3$ are gravity term and torque vectors applied to the joints, respectively. For details of the calculation of the basic metric tensor and matrix $C(\boldsymbol{q},\dot{\boldsymbol{q}})$ for robot manipulators, the reader is referred to [12].

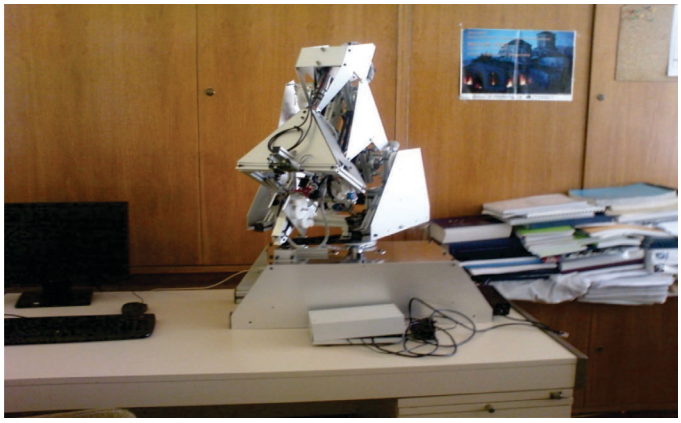

Fig. 1. NeuroArm robotic manipulator with 7DOFs

Usually, motor's optimal working condition include high angular speeds and low torques. On the other side, controlling the movement of robot link demands low speeds and high torques. Hence, it is necessary to interpose a gear transmission between motors and joints. Considering rigid robots, the following equations describe the transmission of the gears

$$\boldsymbol{q}_m = N\boldsymbol{q}, \quad \boldsymbol{Q}^{\boldsymbol{m}} = N\boldsymbol{\tau}_l, \tag{2}$$

wherein $\boldsymbol{q}_m$ represents the positions of the actuators shafts, N is the 3×3 diagonal matrix of the gear ratios, and $\boldsymbol{\tau}_l$ is the vector of torques resulting from the robot manipulator and acting on the motors shafts. It can be easily shown [13] that torque $\boldsymbol{\tau}_l$ is equal to:

$$\boldsymbol{\tau}_l = \left(N^2\right)^{-1}\left(A(\boldsymbol{q})\ddot{\boldsymbol{q}}_m + C(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}}_m\right) - N^{-1}\boldsymbol{Q}^{\boldsymbol{g}} \tag{3}$$

Torque $\boldsymbol{\tau}_l$ can be regarded as a disturbance acting on the motor shaft. As we can see from (3), the influence of this disturbance on motor's dynamics decreases with the increase of the reduction ratio N. This basically means that the presence of a large reduction ratio tends to linearize the dynamic equations of robot. In other words, we can neglect the nonlinear couplings between the motors of the various links, and use linear model instead.

Model of a DC motor consists of a mechanical and electrical part, which can be described with following equations:

$$J_m\ddot{\boldsymbol{q}}_m + B_m\dot{\boldsymbol{q}}_m = \boldsymbol{\tau}_m - \boldsymbol{\tau}_l, \tag{4}$$

$$R_m\boldsymbol{i} + L\frac{d\boldsymbol{i}}{dt} + K_e\frac{d\boldsymbol{q}_m}{dt} = \boldsymbol{u}, \tag{5}$$

wherein J_m is the 3×3 diagonal matrix containing the effective motors inertias, B_m is the 3×3 diagonal matrix containing the viscous friction coefficients of the motors, $\boldsymbol{\tau}_m$ is the vector of torques supplied by the actuators, R_m is the 3×3 diagonal matrix containing the resistances of the armature circuits, $\boldsymbol{i} \in R^3$ represents the vector of the armature currents, L is the diagonal matrix of the armature inductances, K_e is the diagonal matrix containing the back EMF constants, and finally, $\boldsymbol{u}$ is the vector of the armature input voltages. Relationship between torque vector $\boldsymbol{\tau}_m$ and armature current vector $\boldsymbol{i}$ is given as:

$$\boldsymbol{\tau}_m = K_m\boldsymbol{i}, \tag{6}$$

where K_m is the 3×3 diagonal matrix containing the motor torque constants. Now, the complete block diagram of motor dynamics is depicted in Figure 2.

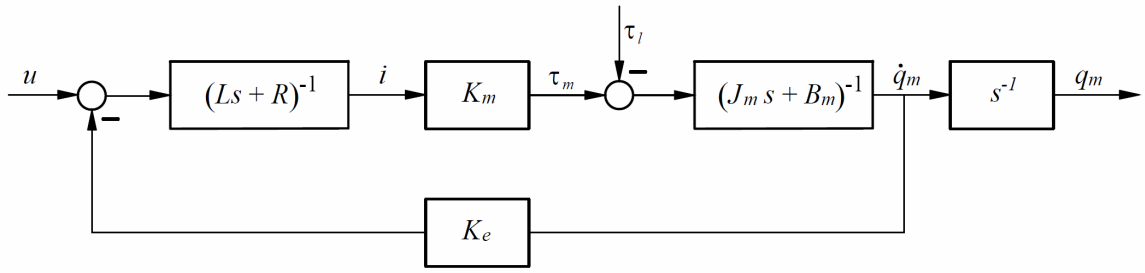

Fig. 2. Block diagram of the DC motor

Observing the above diagram, we can express the relationship between the control input u and the motor shaft output q_m using the following transfer function:

$$G_p(s)=\frac{k_m}{(\ell s+r)(j_m s+b_m)+k_m k_e}\frac{1}{s}. \tag{7}$$

NeuroArm robot uses the Maxon RE36 (70Watt) DC motors for controlling the position of the first three links. Extracting the following parameters from the manufacturer's data sheets: $j_m=65.2$ gcm^2, $r=1.71\ \Omega$, $\ell=0.89$ mH, $k_m=44.5$ mNm/A, $k_e=1/215$ V/rpm, the transfer function $G_p(s)$ becomes:

$$G_{\mathrm{m}}(s)=\frac{K}{s(Ts+1)}. \tag{8}$$

wherein K=22.515 and T=0.0056409. The second order transfer function (7) is reduced to a first order model given by (8) because one of the three poles of $G_p(s)$ is located far left in the s-plane, and its influence can be neglected. Finally, to sum up, the original nonlinear robotic system, due to high reduction ratio $\left(N=\mathrm{diag}\{185,230,74\}\right)$, resulted into 3 linear, decoupled subsystems, described with motor transfer function $G_{\mathrm{m}}(s)$.

3. Controller design

The advanced control system scheme of NeuroArm robot is shown in Figure 3. As stated before, control signal is divided into two parts, with feedforward and feedback paths. Here, ILC is proposed only in feedforward manner through the fractional PD^{μ} control law, while feedback path consists of classical *PD* control algorithm.

The overall control input can be obtained as:

$$u_{i+1}(t)=u_{ffi+1}(t)+u_{fbi+1}(t). \tag{9}$$

where $u_{ffi+1}(t)$ and $u_{fbi+1}(t)$ are two aforementioned parts of control signal in the $i+1-$th iteration. More specifically, we can write the following:

$$u_{ffi+1}(t)=u_i(t)+\Gamma_1\left(P_1 e_i^{(\mu)}(t)+P_2 e_i(t)\right) \tag{10}$$

$$u_{fbi+1}(t)=K_d\dot{e}_{i+1}(t)+K_p e_{i+1}(t) \tag{11}$$

where $q_d(t)$ is the desired output trajectory, $e_i(t) = q_d(t) - q_{m,i}(t)$ and $e_{i+1}(t) = q_d(t) - q_{m,i+1}(t)$ are trajectory tracking error in the $i-$th iteration, and $i+1-$th iteration, respectively. Also, $P_1, P_2, \Gamma_1, K_d, K_p \in R^{3\times 3}$ are feedforward and feedback loop positive-definite diagonal matrices. The term $e_i^{(\mu)}(t)$ in (9) represents error derivative of μ-th order. A sufficient convergent condition of a proposed feedforward-feedback control law can be found in [14].

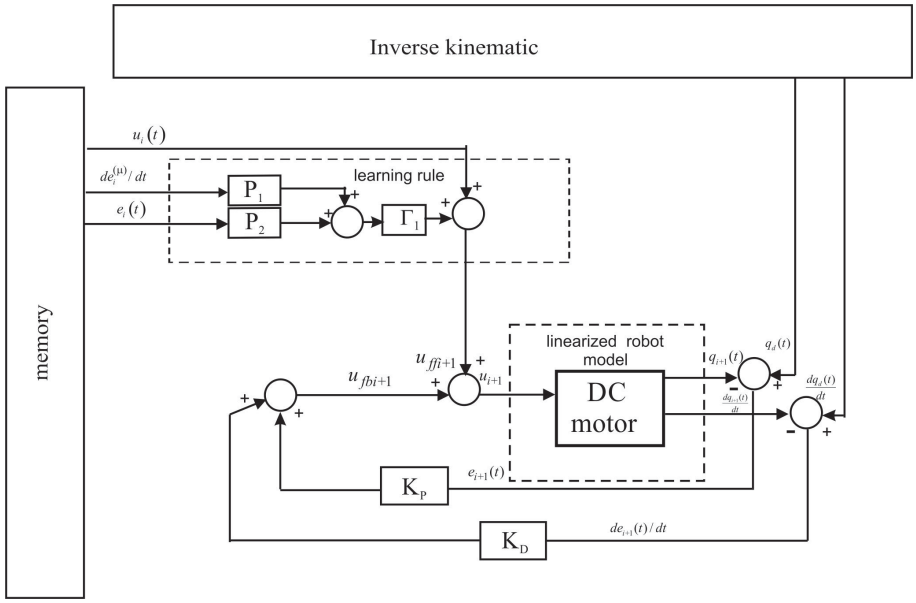

Fig. 3. Block diagram of the advanced feedforward-feedback PD^{μ}/PD control system

Learning gain matrices P_1, P_2 and Γ_1 are chosen by trial and error, while few words must be said now regarding the tuning of K_d and K_p matrices. Feedback control law for each one of the three motors that put in motion robotic arm, can be expressed by the following transfer function:

$$C(s) = \frac{k_d s + k_p}{T_f s + 1}, \tag{12}$$

where T_f is the filter time constant, k_d and k_p are derivative and proportional gain, respectively. Tuning method derived below is explained thoroughly in [15], and reader is referred to it for detailed explanations. In this paper, parameters of the *PD* feedback controller are derived specifically for the plant $G_m(s)$ given by (8), since the nonlinear robot dynamics is equivalent to that linear model. To begin with, we start with the complementary transfer function $T(s)$ given by [16]:

$$T(s) = \frac{\eta_1 s + \eta_0}{(\lambda s + 1)^3} \tag{13}$$

where time constant $\lambda > 0$, η_1 and η_0 are free parameters which will be determined to obtain the desired dynamic characteristics of the closed loop system. Having in mind that $T(s)$ can be formulated also as a $T(s) = C(s)G_m(s)/(1 + C(s)G_m(s))$, then, after some calculations, one obtains controller $C(s)$ in the following form:

$$C(s)=\frac{\eta_1 s+\eta_0}{(\lambda s+1)^3-\eta_1 s-\eta_0}\frac{1}{G_m(s)}, \tag{14}$$

Free parameters η_1 and η_0 are determined in order to cancel the poles of $G_m(s)$, and these values are given as:

$$\eta_0=1,\ \eta_1=\frac{\lambda}{T^2}\left[3T^2-\lambda(3T-\lambda)\right]. \tag{15}$$

Finally, parameters of the *PD* controller $C(s)$ are defined by

$$k_p=\frac{T^2}{(3T-\lambda)K\lambda^2},\ k_d=\frac{3T^2-(3T-\lambda)\lambda}{(3T-\lambda)K\lambda},\ T_f=\frac{T\lambda}{3T-\lambda}. \tag{16}$$

Tuning formulae (16) contain one free parameter: time constant λ. By observing the above relations, we can conclude that system response is faster as λ gets smaller. So, by adjusting λ one can obtain very good robustness/performance trade-off, which is a key issue in modern control system design.

4. Simulation results

Now, simulation results are presented to illustrate the validity of the proposed advanced ILC algorithm – PD^{μ}/PD type. The desired output trajectories of the NeuroArm robotic links are as follows:

$$\begin{aligned} &q_{d1}=0.2\cdot t^2(6-t)[rad], \qquad q_{d2}=0.25\cdot cos(t)[rad],\\ &q_{d3}=0.2\cdot sin(2t)[rad], \qquad \forall t\in[0,6sec]. \end{aligned} \tag{17}$$

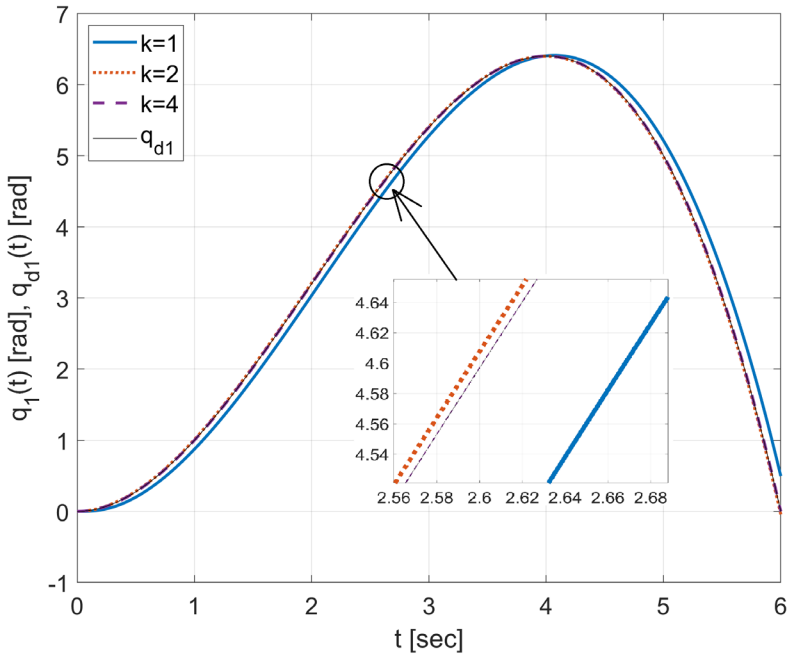

Fig. 4. The tracking performance of the first robot link through the 1st, 2nd and 4th iteration, and $q_{d1}(t)$.

The following values of learning gain matrices are adopted:

$$P_1=diag(1,1,1),\quad P_2=diag(1,1,1),\quad \Gamma_1=diag(0.1,\ 0.01,\ 0.1). \tag{18}$$

Parameters of the feedback controller are determined according to (16) in order to obtain the best trade-off between robustness and performances of closed loop system:

$$K_p = diag(1{,}3165;\ 1{,}31651;\ 1{,}3165), \quad K_d = diag(0.2255;\ 0.2255;\ 0.2255) \tag{19}$$

Filter constant is calculated as $T_f = 0.2714$. For all the selected parameters shown above and for $\mu = 1.1$, simulation results are shown in Figures 4-6. Figures show that the control system successfully tracks output trajectories just after few iterations.

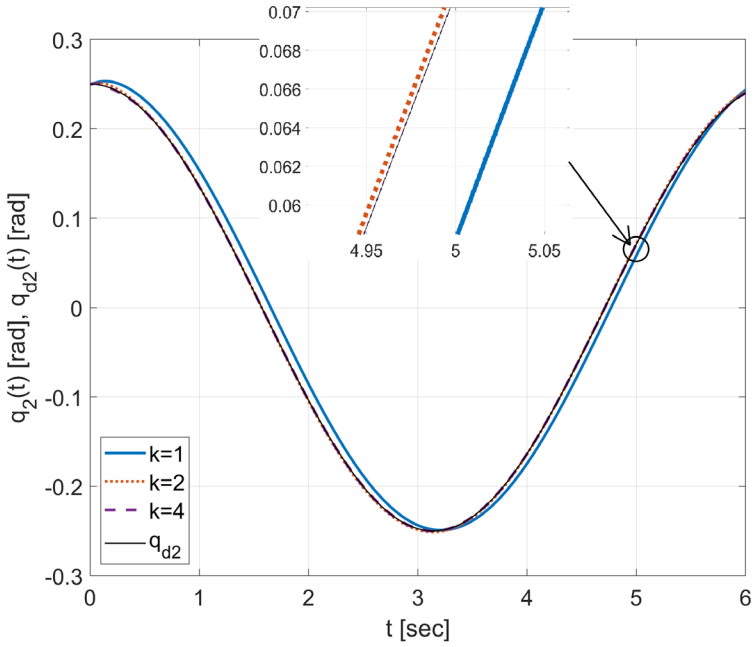

Fig. 5. The tracking performance of the second robot link through the 1st, 2nd and 4th iteration, and $q_{d2}(t)$.

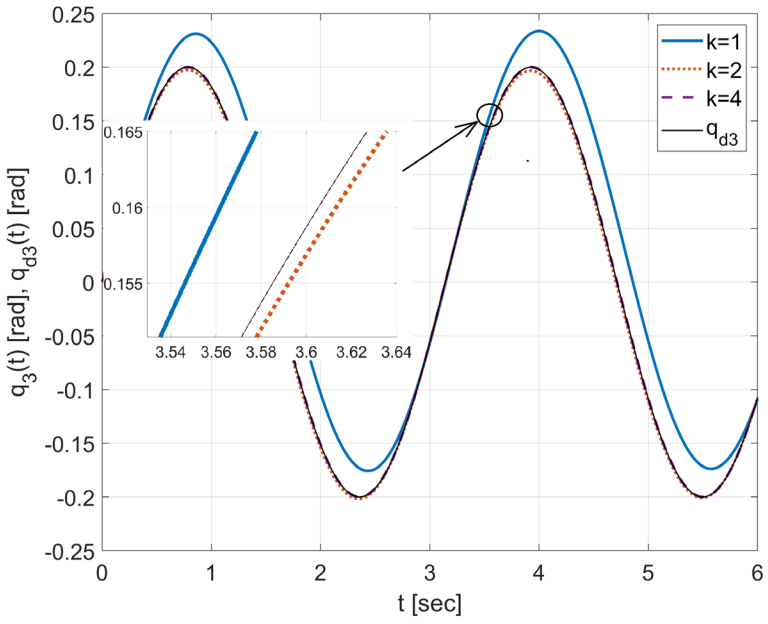

Fig. 6. The tracking performance of the third robot link through the 1st, 2nd and 4th iteration, and $q_{d3}(t)$.

Now, in order to show the influence of fractional derivative μ on tracking error, we run the following simulation. We changed gradually the value of parameter μ from 0.8 to 1.2, and observed how it effects on system's response. Results are shown in Figures 7-9, in terms of maximum absolute tracking error for each of three robot links. It can be seen that tracking errors decrease as number of iterations increase. Also, the best results in terms of the speed of convergence of output signal are obtained for the value $\mu = 1{,}1$ for the first and third robot link, and $\mu = 1{,}2$ for the second joint.

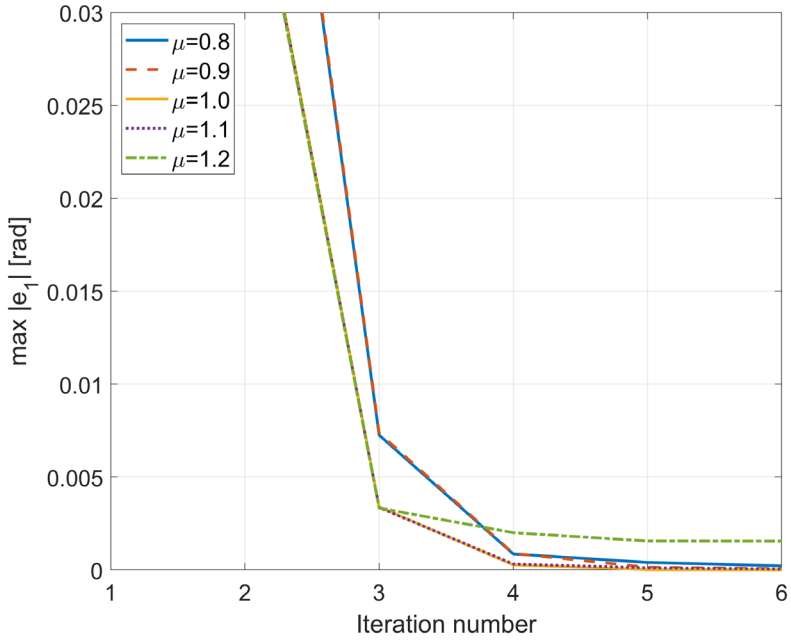

Fig. 7. Convergence of the maximum absolute tracking error through the iterations for the first robot link

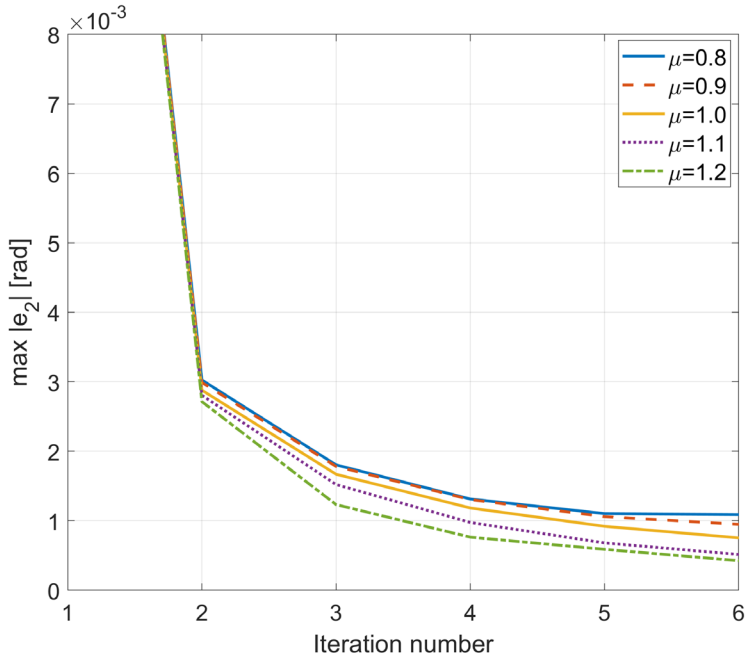

Fig. 8. Convergence of the maximum absolute tracking error through the iterations for the second robot link

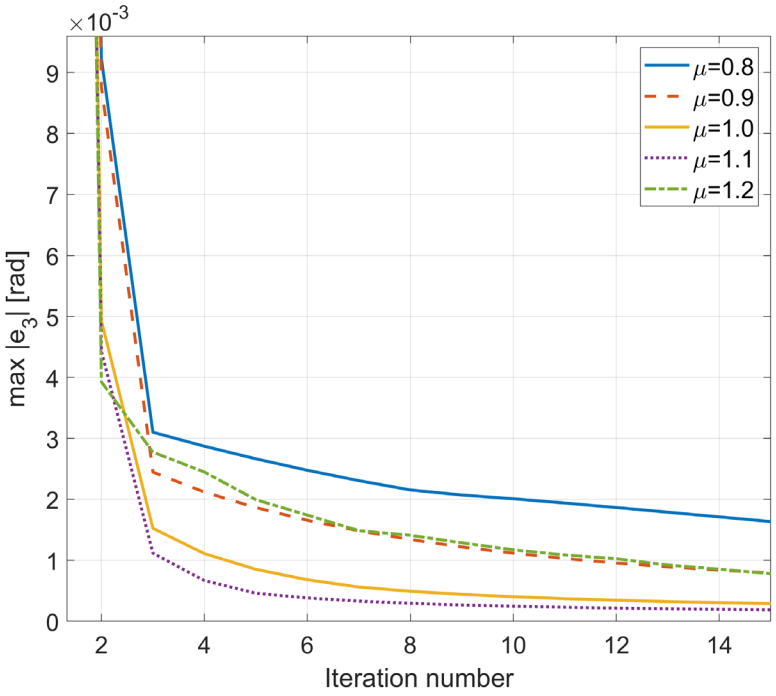

Fig. 9. Convergence of the maximum absolute tracking error through the iterations for the third robot link

5. Conclusions

This paper deals with the tracking output task of a 3-DOFs robot manipulator. First, a mathematical model of robotic system is introduced. Due to high gear transmission ratio the model is reduced to the linear dynamics of actuators. Then, an advanced PD^{μ} type iterative learning control is established in the feedforward path, while classical PD controller is applied to the feedback loop. Simulation results verify the efficiency of the proposed advanced ILC algorithm. Convergence rate is analyzed for different values of fractional order parameter μ and for different desired trajectories of robot links.

Acknowledgment. Authors gratefully acknowledge the support of Ministry of Education, Science and Technological Development of the Republic of Serbia under the contract 451-03-68/2020-14/200105 (Projects TR 33047 (P.D.M), TR 35006, III 41006 (M.P.L.)) and Project TR 33020 (T.B.Š), as well as partially supported by the Serbia-Italian bilateral project ADFOCMEDER

References

[1] B. Siciliano, L. Sciavicco, L. Villani, G. Oriolo, *Robotics*, London: Springer-Verlag, 2009
[2] H. G. Sage, M. F. De Mathelin, E. Ostertag, "Robust control of robot manipulators: A survey," *Inter J Control*, vol. 76, no. 16, pp. 1498-1522, 1999.
[3] G.K. McMillan, *Industrial applications of PID control*. In: Vilanova R, Visioli A, eds. PID control in the third millenium. London: Springer, pp. 415-461, 2012.

[4] T.B. Šekara, M.R. Mataušek, *Optimization of PID Controller Based on Maximization of the Proportional Gain Under Constraints on Robustness and Sensitivity to Measurement Noise*, IEEE Trans Automat Contr, vol. 54, pp. 184-189, 2009.
[5] Yung CS and Chengying X. *Intelligent Systems: Modeling, Optimization, and Control*, 1st ed. Boca Raton USA: CRC Press, 2017.
[6] Arimoto S, Kawamura S and Miyazaki F. *Bettering operation of robots by learning*. Journal of Robotic Systems, 1984; 2(1):123-140.
[7] Lazarević M and Panagiotis T. *Robust second-order PDalpha type iterative learning control for a class of uncertain fractional order singular systems*. Journal of Vibration and Control, Sage Journals, 2016;22(8).
[8] S.G. Samko, A.A. Kilbas, O.I. Marichev, *Fractional Integrals and Derivatives: Theory and Applications.* Gordon and Breach, 1993.
[9] A. Oustaloup, X. Moreau, M. Nouillant, *The CRONE suspension*, Control Eng. Practice, vol. 4, no.8, pp.1101-1108, 1996.
[10] I. Podlubny, *Fractional-order systems and PI$^{\lambda}$D$^{\mu}$-controllers*, IEEE Trans.Autom.Cont, vol.44, pp.208–214, 1999.
[11] M. Safaei, S. Tavakoli, *Tuning of robust fractional-order phase-lead compensators using pole placement and pole-zero ratio minimization*, Journal of Vibration and Control, pp.1-12, January 2018.
[12] V. Čović, M. P. Lazarević, *Robot Mechanics*, Faculty of Mechanical Engineering, Belgrade, Serbia, 2009.
[13] P. D. Mandić, M. P. Lazarević, T. B. Šekara, M. Č. Bošković, G. Maione, *Robust control of robot manipulators using fractional order lag compensator,* 7th International Congress of Serbian Society of Mechanics, Sr. Karlovci, Serbia, June 24-26, 2019.
[14] Zhao Y, Zhou F., Wang Y. ·Li Y., *Fractional-order iterative learning control with initial state learning design*, Nonlinear Dyn (2017) 90:1257–1268.
[15] M. Č. Bošković, M. R. Rapaić, T. B. Šekara, P. D. Mandić, M. P. Lazarević, *A novel method for design of complex compensators in control systems,* 18th International Symposium INFOTEH-JAHORINA, March 20-22, 2019.
[16] K. J. Åström, R. M. Murray, *Feedback Systems- An Introduction for Scientists and Engineers*, Princeton, USA: Princeton University Press, 2008.