8th International Congress of Serbian Society of Mechanics
Kragujevac, Serbia, June 28-30, 2021

ROBUST CONSTRAINED STATE SPACE ILC FOR 3DOF ROBOT MANIPULATOR

Aleksandar Dubonjac[1], **Mihailo P. Lazarević**[2]

[1] Lola institute, Kneza Višeslava 70a,11030 Belgrade, Serbia
e-mail: aleksandar.dubonjac@li.rs
[2] Faculty of Mechanical Engineering,
The University of Belgrade, Kraljice Marije 16, 11120 Belgrade 35
e-mail: mlazarevic@mas.bg.ac.rs

Abstract:

This paper focuses on the effect of the control system parameters on the convergence speed of two constrained state space Iterative Learning Control (ILC) algorithms: Bounder Error Algorithm (BEAILC) and Constrained Output Algorithm (COILC), applied to the nonlinear model of a 3DOF robotic manipulator in presence of recurring disturbance. Analysis and comparison of previously mentioned algorithms were conducted through simulations. The obtained results have shown that COILC algorithm converges faster than BEAILC algorithm, as the BEAILC restricts the output trajectory more rigorously. Simulations have shown that change in feedback parameters' values has higher impact on the iteration interruptions (increase will lower the number of interruptions), while the learning parameters have higher impact on the whole ILC procedure duration (decrease will require more iterations to achieve the desired tracking accuracy). Additionally it's been shown that both algorithms successfully rejected the recurring disturbance.

Key words: robot control, ILC, bounded error, state space, constrained output

1. Introduction

Iterative learning control (ILC) is an intelligent control method which, through repetition in every iteration, using the information from previous trials, improves the control signal for the next trial in order to decrease (eliminate) the tracking error of a repetitive task. Proposed for the first time by Arimoto in 1984, recently it's gained researchers' attention as a viable control method for industrial robots (manipulators), which are required to execute repetitive tasks over and over with high precision. As the only information that ILC is learning from is the trajectory tracking error, it can compensate for unmodeled dynamics and recuring disturbances [1-6].

In this paper two nonlinear constrained state space ILC methods are considered for a nonlinear 3DOF robotic manipulator model: Bounded Error Algorithm (BEAILC) and Constrained Error Algorithm (COILC), which are compared through simulations. This paper extends on the [14], with addition of the disturbances.

2. Constrained State Space ILC

In reality robots have space boundaries, velocity limits and other saturations that make them constrained state space systems, thus all operations performed by those manipulators are within the constrained state space.

Violation of constraints, that can occur if the robot is following the desired trajectory close to the state space limits, can cause failure of the ILC trial as it will be interrupted and could potentially cause damage to the robot or its surroundings (Fig. 1). One of the problems of the standard ILC procedure is the transient error growth. It is possible that in the first several iterations the tracking error grows significantly before it starts to converge to zero, potentially violating the operative space boundaries. As solutions to this transient error growth problem, two of the following algorithms are proposed:

- Bounded Error Algorithm (BEA)
- Constrained Output Algorithm (CO)

Both algorithms are based around the idea of terminating the tracking process if the generalized coordinates boundaries are violated and their convergence is proven in [8].

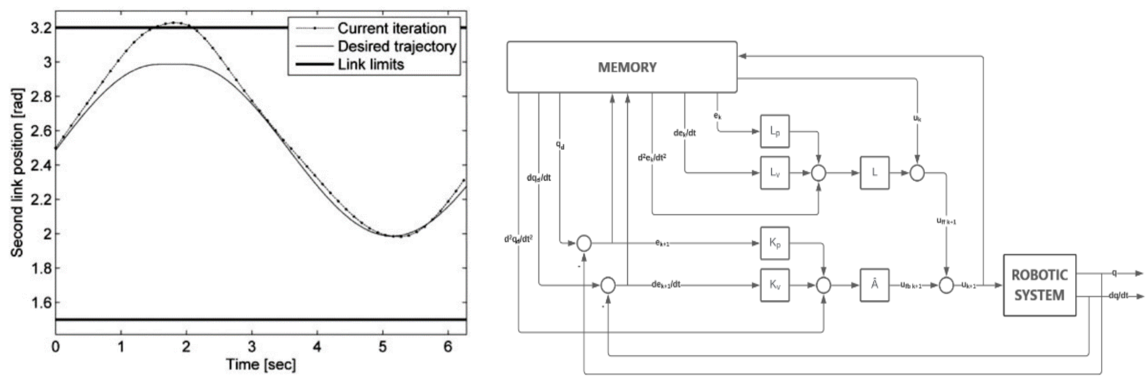

Fig. 1. Violation of generalized coordinates constraints [7] - left, Block diagram of BEAILC and COILC algorithms - right [14]

Differential equations of motion for the given robotic system are obtained in the identical covariant form of Lagrange equations of the second kind as [9, 10]:

$$\sum_{\alpha=1}^{n} a_{\gamma\alpha}(q)\ddot{q}_{\alpha} + \sum_{\alpha=1}^{n}\sum_{\beta=1}^{n} \Gamma_{\alpha\beta,\gamma}(q)\dot{q}_{\alpha}\dot{q}_{\beta} = Q_{\gamma}, \quad \gamma = 1,2,\dots n \tag{2.1}$$

where $q \in \mathfrak{R}^n$, $\dot{q} \in \mathfrak{R}^n$ are generalized coordinates and velocities respectively, the coefficients $a_{\gamma\alpha} = a_{\alpha\gamma}$ are the covariant coordinates of the basic metric tensor $A(q) = a(q) = [a_{\alpha\beta}] \in R^{nxn}$ and symbols $\Gamma_{\alpha\beta,\gamma}$ denote Christoffel symbols of the first kind which are defined as:

$$\Gamma_{\alpha\beta,\gamma} = \frac{1}{2}\left(\frac{\partial(a_{\beta\gamma})}{\partial q^{\alpha}} + \frac{\partial(a_{\gamma\alpha})}{\partial q^{\beta}} - \frac{\partial(a_{\alpha\beta})}{\partial q^{\gamma}}\right), \quad \alpha, \beta, \gamma = 1, \dots, n. \tag{2.2}$$

The generalized forces are $Q_{\gamma} = Q_{\gamma}^{g} + Q_{\gamma}^{c} + Q_{\gamma}^{f} + Q_{\gamma}^{v} + Q_{\gamma}^{a}$, $\gamma = 1,2,\dots,n$ where $Q_{\gamma}^{g} \in \mathfrak{R}^n$, $Q^c \in \mathfrak{R}^n$, $Q^f \in \mathfrak{R}^n$ and $Q^a \in \mathfrak{R}^n$ are generalized gravity, elastic, dry friction, viscous friction and actuator torques (control signals in our case), respectively. The robot arm dynamics can be presented in compact form as:

$$a(q)\ddot{q} + n(q, \dot{q}) = Q \tag{2.3}$$

The sufficient condition for both algorithms' convergence:

$$\|I - LA^{-1}\| \le \rho < 1. \tag{2.4}$$

For a high convergence rate, the learning operator is chosen as estimated inertia matrix $L(q) \equiv \hat{A}(q)$, as advised in [11].

2.1 Bounded Error Algorithm

Bounded Error Algorithm is tracking the error norm during the iteration and it terminates it as soon as the error norm reaches its limit. Correction of the control signal is affected only by information collected before the interruption.

BEA application results in the output trajectory laying inside the hypercylinder with radius of ε around the desired trajectory during each iteration. Downside of this algorithm is the over restriction of trajectory in the areas where the trajectory is far from its limits, causing more frequent interruptions of the ILC procedure [12, 13].

2.2 Constrained Output Algorithm

Constrained Output Algorithm only limits the maximum and minimum values of the output trajectory, allowing deviations from the desired trajectory in the safe areas. This results in faster convergance, when compared to BEA, due to more relaxed restrictions [8].

3. Simulation results

Trajectory tracking simulations of 3DOFs robotic system (Table 1) were conducted in MATLAB and Simulink environment, using the Runge-Kutta method (ODE4), where the simulation step was 0.00001.

e_i	ρ_{ii}	m [kg]	l [m]
$e_{11}^{(1)} = \begin{Bmatrix} 0 \\ 0 \\ 1 \end{Bmatrix}$	$\rho_{11}^{(1)} = \begin{Bmatrix} 0 \\ 0 \\ 0.15 \end{Bmatrix}$	0.15	0.15
$e_{22}^{(2)} = \begin{Bmatrix} -1 \\ 0 \\ 0 \end{Bmatrix}$	$\rho_{22}^{(2)} = \begin{Bmatrix} 0 \\ 0 \\ 0.5 \end{Bmatrix}$	0.5	0.5
$e_{33}^{(3)} = \begin{Bmatrix} -1 \\ 0 \\ 0 \end{Bmatrix}$	$\rho_{33}^{(3)} = \begin{Bmatrix} 0 \\ 0.35 \\ 0 \end{Bmatrix}$	0.35	0.35

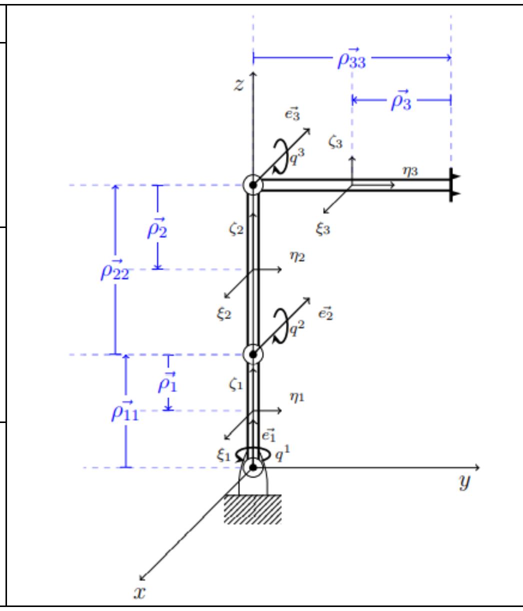

Table 1. Robot configuration and it's parameters

The feedback term used in both algorithms is:

$$u_{fb} = \hat{A}(q)\left[\ddot{q}_d(t) + K_v\left(\dot{q}_d(t) - \dot{q}_k(t)\right) + K_p\left(q_d(t) - q_k(t)\right)\right] \tag{3.1}$$

where K_v and K_p are feedback gains.

BEA and CO algorithms can be described through the following steps [8][12], taking into the account the feedback term (3.1):

1. Set the initial iteration number $\mathrm{k} = 0$ and begin the iterative procedure

2. a) (BEA) - Starting from the initial position $q_k(0) = q_d(0)$ the system is tracking the desired trajectory under the control $u(q,t) = u_k(t) + u_{fb}(t)$ while $|q_k(t) - q_d(t)| < \varepsilon$ and $t < T$. When $t = T$ or for the first $T_k: 0 < T_k < T$, $\|q_k(t) - q_d(t)\| = \varepsilon$, then the tracking process is stopped and T_k is set to the stop time of iteration k.
   b) (CO) - Starting from the initial position $q_k(0) = q_d(0)$ the system is tracking the desired trajectory under the control $u(q,t) = u_k(t) + u_{fb}(t)$ while $Q_i^{min} < q_i^k < Q_i^{max}, i = 1,2,..,n$ and $t < T$. When $t = T$ or for the first $T_k: 0 < T_k < T$, $q_i^k = Q_i^{min}$ or $q_i^k = Q_i^{min}$, then the tracking process is stopped and T_k is set to the stop time of iteration k.
3. At the end of the current iteration the learning controller updates the input control signals for the next iteration u_{k+1} according to the following learning update law:

$$u_{k+1}(t) = u_k(t) + \begin{cases} L\big(q_k(t)\big)[\ddot{q}_d(t) - \ddot{q}_k(t) + L_v\big(\dot{q}_d(t) - \dot{q}_k(t)\big) \\ \quad +L_p(q_d(t) - q_k(t))], t \in [0, T_k]; \\ 0, t \in (T_k, T] \end{cases} \tag{3.2}$$

   where L_p and L_v are learning gains.
4. If the overall output error is less than or equal to an acceptable tracking accuracy and T_k equals T, then the learning procedure terminates successfully and the optimal feedforward control signal is u_k. Otherwise, set $k = k + 1$ and go to step 2.

The desired trajectories defined in the space of generalized coordinates for joints are taken from [14]:

$$\begin{gathered} q_d^1(t) = 4sin(t), \\ q_d^2(t) = 2cos(t), \\ q_d^3(t) = 0.8cos(2t), \\ \forall \mathrm{t} \in [0, \mathrm{T}], \qquad \mathrm{T} = 2\pi. \end{gathered} \tag{3.3}$$

The initial resetting conditions hold for all iterations, that is, $q^i(0) = q_d^i(0)$, $\dot{q}^i(0) = \dot{q}_d^i(0)$ and control in initial iteration is:

$$u_0(t) \equiv 0, t \in [0, T]. \tag{3.4}$$

Sufficient condition for convergence is met, when $L(q) \equiv \hat{A}(q)$:

$$\max_{q^i}\left\|I - \hat{A}(q)A^{-1}(q)\right\| = 0.9282 < 1, \; q^i \in [-2\pi, 2\pi] \tag{3.5}$$

The rest of the control system parameters (K_p, K_v, L_p и L_v) were taken from the previous paper, for further comparison [14].

Generalized coordinates boundaries and control system parameters for both algorithms were set so that the simulation results are comparable (matching maximum values). Hypercylinder radius for BEAILC algorithm is set as: $\varepsilon = 0.3$

The desired tracking accuracy that has to be obtained by both algorithms $\left\|e_{max}^i\right\| < \mu$ is:

$$\mu = 0.005\,. \tag{3.6}$$

The disturbances affecting the robot's output trajectory are:

$$\begin{gathered} \eta_1 = 3te^{-0.5t} \\ \eta_2 = 1 - e^{-4t}(1 + 4t) \\ \eta_3 = 0.5(1 - \cos 2t)\,. \end{gathered} \tag{3.7}$$

3.1 First set of parameters BEA and CO

Parameters for the first set are chosen as diagonal matrices [13, 14]:

$$\begin{aligned} K_p &= 120 * I\,, \quad & K_v &= 60 * I \\ L_p &= 100 * I\,, \quad & L_v &= 20 * I \end{aligned} \tag{3.8}$$

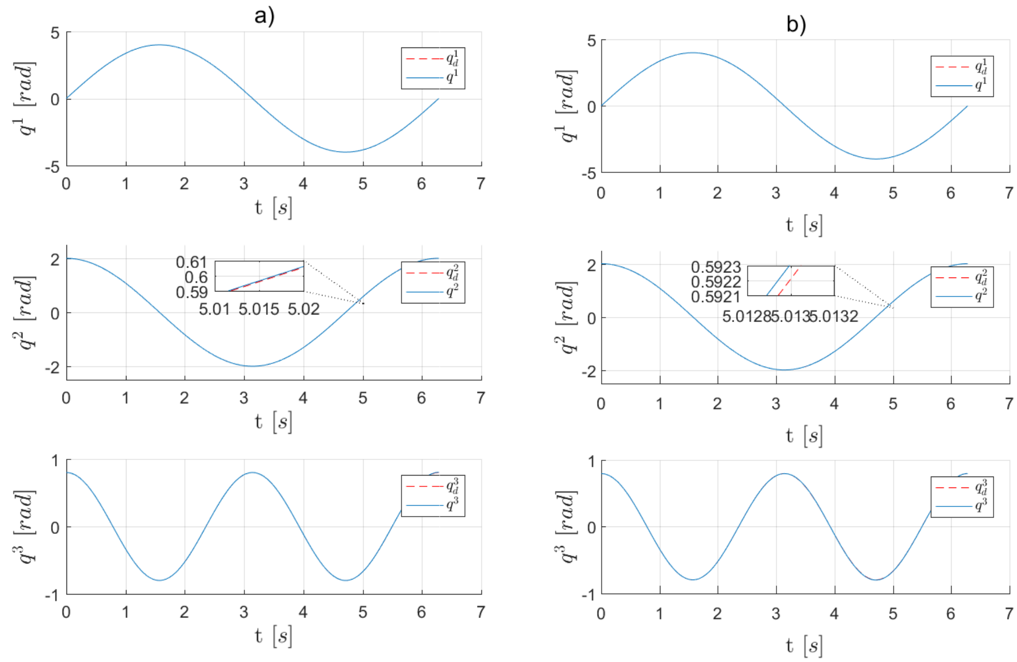

Fig. 2. Trajectory tracking: a) BEA, b) CO

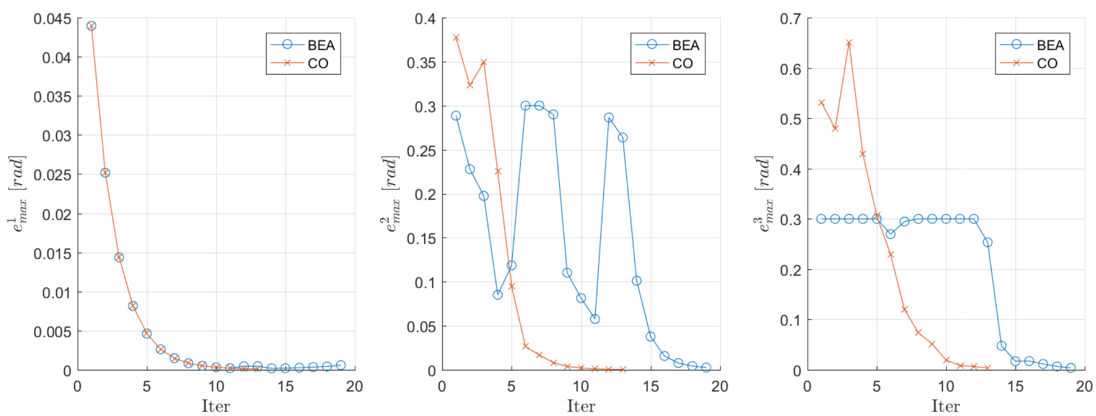

Fig. 3. Maximum error norm

BEA algorithm with (3.8) parameters, obtained the desired accuracy after 19 iterations, with maximum tracking errors (Fig. 3): $e^1_{max} = 5.964584554207877e(-04)$, $e^2_{max} = 0.002605102606072$, $e^3_{max} = 0.003707387054551$.

CO algorithm with (3.8) parameters, obtained the desired accuracy after 13 iterations, with maximum tracking errors (Fig. 3): $e^1_{max} = 1.224670704455836e(-04)$, $e^2_{max} = 3.931029416956999(\mathrm{e}-04)$, $e^3_{max} = 0.003784062605802$.

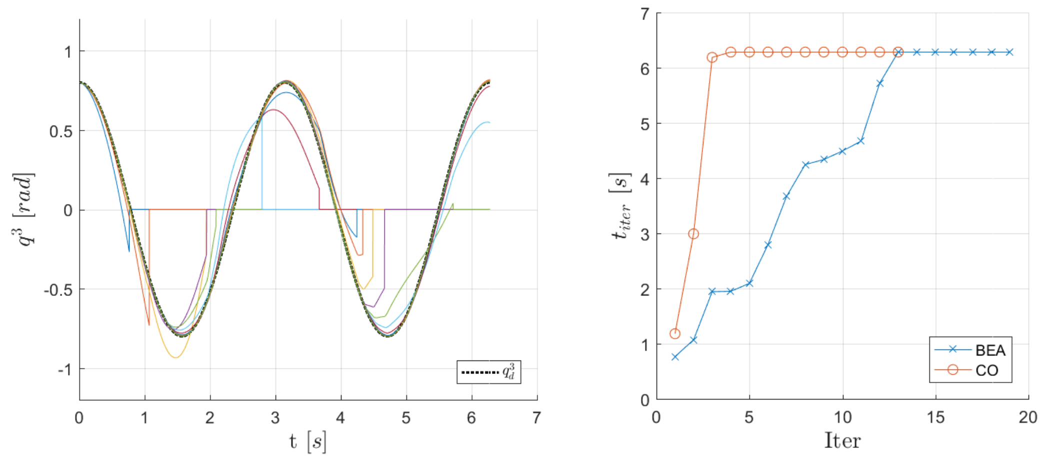

Fig. 4. Trajectory tracking trough iterations – BEA - left, Iteration duration time - right

On Fig. 2 the final trajectory tracking can be seen. On Fig. 3 it can be seen that in case of the BEA the maximum error norm was capped at the value of $\varepsilon = 0.3$, where the CO algorithm allowed for higher deviations from the desired trajectory. Due to more relaxed constrains, interruptions were less frequent and more learning information about the trajectory was available, which resulted in faster convergence of the CO algorithm (Fig. 4). Both algorithms successfully learned to reject repeatable disturbance.

3.2 Second and third set of parameters BEA and CO

In case of the third set, feedback and learning gains were tuned separately for individual joints.

Second set:

$$K_p = 150 * I, \quad K_v = 80 * I$$
$$L_p = 70 * I, \; L_v = 15 * I \tag{3.11}$$

Third set:

$$K_p = \begin{bmatrix} 100 & 0 & 0 \\ 0 & 120 & 0 \\ 0 & 0 & 130 \end{bmatrix}, K_v = \begin{bmatrix} 20 & 0 & 0 \\ 0 & 40 & 0 \\ 0 & 0 & 60 \end{bmatrix}$$
$$L_p = \begin{bmatrix} 65 & 0 & 0 \\ 0 & 70 & 0 \\ 0 & 0 & 80 \end{bmatrix}, L_v = \begin{bmatrix} 15 & 0 & 0 \\ 0 & 20 & 0 \\ 0 & 0 & 45 \end{bmatrix} \tag{3.12}$$

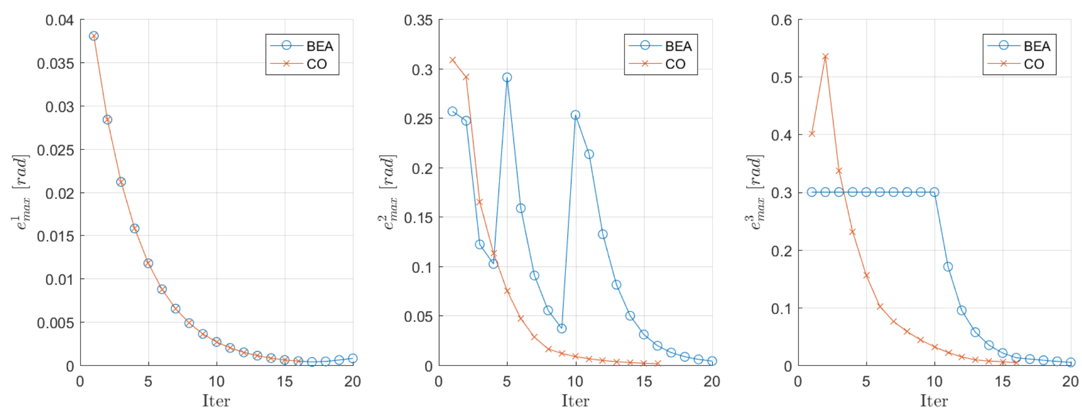

Fig. 5. Maximum error norm - set 2

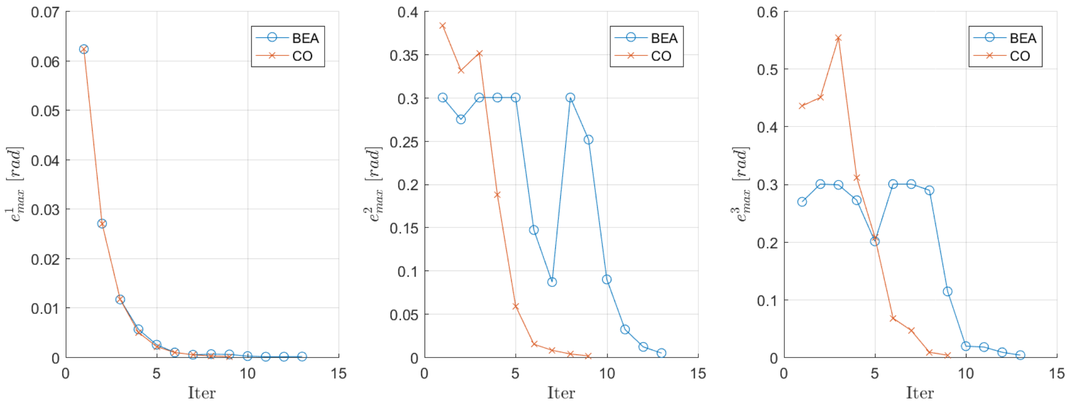

Fig. 6. Maximum error norm – set 3

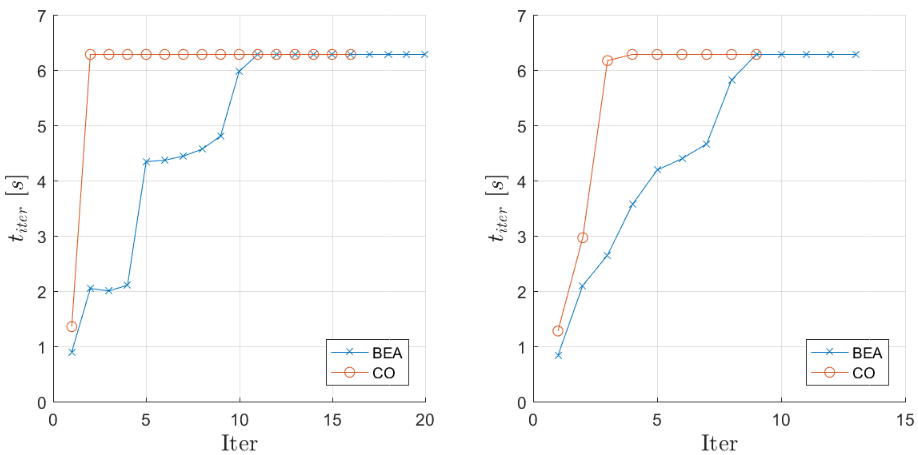

Fig. 7. Iteration duration time: set 2 – left, set 3 - right

BEA algorithm with (3.11) parameters, obtained the desired accuracy after 20 iterations, while the CO algorithm with the same (3.11) parameters required 16 iterations (Fig. 5). Number of iterations was increased for both algorithms due to lower learning gains (Fig. 7). On the other hand, the number of iteration interruptions was decreased due to higher feedback gains.

BEA algorithm with (3.12) parameters, the desired accuracy was obtained after 13 iterations, while the CO algorithm with the same (3.12) parameters required 9 iterations (Fig. 6). In comparison with previous (3.12) parameter set where the joint 3 caused most of the iteration terminations in case of BEA algorithm (Fig. 6), with this parameter set it can be seen that now the joint 2 caused most of the trial terminations. The recurring disturbance was successfully rejected as well. More simulation results without the disturbances can be seen in [14].

3. Conclusions

From previously shown simulation results, it can be observed that both algorithms successfully managed to decrease the tracking error under the desired accuracy in the presence of the recurring disturbance.

Due to more limiting constrains, BEAILC algorithm takes more iterations to obtain desired tracking accuracy compared to COILC algorithm, which obtains more information from the less frequently interrupted iterations. Tuning the learning and feedback parameters for each joint individually can help speeding up the convergence, in case of a particular joint causing most of the ILC interruptions. Increase in feedback gains can decrease the number of iteration terminations during learning process, while the decrease in learning parameter values will increase the number of iterations required to obtain the desired tracking accuracy

Taking into the account the results from [14], the presence of the recuring disturbance didn't have significant impact to convergence speed of both algorithms in this case.

4. Acknowledgement

The presented research was supported by the Ministry of Education, Science and Technological Development of the Republic of Serbia by contract no. 451-03-9/2021-14/200105 from 05.02.2021 and contract no. 451-03-9/2021-14/200066.

References

[1] Ahn HS, Moore K and Chen Y. *Iterative learning control robustness and monotonic convergence for interval systems.* 1st ed. London: Springer-Verlag London Limited, 2007.

[2] Arimoto S, Kawamura S and Miyazaki F. *Bettering operation of robots by learning.* Journal of Robotic Systems, 1984; 2(1):123-140.

[3] Isao T and Hunag PH. *Iterative Learning Control for Trajectory Tracking of Robot Manipulators*. International Journal of Automation and Smart Technology, 2017; 7(3) 133-139.

[4] Lazarević M and Panagiotis T. *Robust second-order PD alpha type iterative learning control for a class of uncertain fractional order singular systems*. Journal of Vibration and Control, Sage Journals, 2016;22(8):2004-2018, DOI: 10.1177/ 1077546314562241.

[5] Lazarević M, Mandić P, Cvetković B, et al. *Advanced open-closed-loop PIDD2 /PID type ILC control of a robot arm. In:* Proceedings of the INISTA2018 conference, Thessaloniki, Greece, 2018, pp.1-8, DOI: 10.1109/INISTA.2018.8466308.

[6] Cai Z., *Iterative Learning Control: Algorithm Development and Experimental Benchmarking*, University of Southampton, Faculty of Engineering and Applied Science, 2009.

[7] Yovchev K., Delchev K., Krastev E., *State Space Constrained Iterative Learning Control for Robotic Manipulators,* Asian Journal of Control, Vol. 20, No. 1, pp. 1–6, DOI: 10.1002/asjc.1680, 2018.

[8] Yovchev K., Delchev K., Krastev E., *Constrained Output Iterative Learning Control*, Faculty of Mathematics and Informatics, Sofia University, 2020.

[9] Lazarević, M., Čović. V., *Robot Mechanics,* Faculty of Mechanical Engineering, Belgrade, 2009, (*in Serbian*).

[10] M.Lazarević, M. Cajić *Determination of Joint Reactions in a Rigid Multibody System, Two Different Approaches,* Journal FME Transactions, Faculty of Mechanical Engineering, Belgrade, Vol.44. No2, 2016.

[11] Delchev K., Zahariev E., *Computer Simulation-Based Synthesis of Learning-Control Law of Robots*, Institute of Mechanics, Bulgarian Academy of Sciences, Sofia, Bulgaria 2008.

[12] Delchev K., *Iterative learning control for robotic manipulators: A bounded-error algorithm*, Institute of Mechanics, Bulgarian Academy of Sciences, 2013.

[13] Yovchev K., *Finding The Optimal Parameters for Robotic Manipulator Applications of The Bounded Error Algorithm for Iterative Learning Control*, Journal of Theoretical and Applied Mechanics, Sofia, Vol. 47 No. 4 (2017) pp. 3-11 DOI: 10.1515/jtam-2017-0016, 2017.

[14] Dubonjac A., Lazarević M., *State Space Constrained Iterative Learning Control For Robotic Manipulator With 3DOFs*, FME Transactions, Belgrade, Vol. 49 No. 2 (2021), doi: 10.5937/fme2102429D, 2021.