Journal of Theoretical and Applied Mechanics, Sofia, Vol. 47 No. 4 (2017) pp. 3-11

GENERAL MECHANICS

FINDING THE OPTIMAL PARAMETERS FOR ROBOTIC MANIPULATOR APPLICATIONS OF THE BOUNDED ERROR ALGORITHM FOR ITERATIVE LEARNING CONTROL

KALOYAN YOVCHEV*

Faculty of Mathematics and Informatics, Sofia University, Sofia, Bulgaria

[Received 18 April 2017. Accepted 4 December 2017]

ABSTRACT: This paper continues previous research of the Bounded Error Algorithm (BEA) for Iterative Learning Control (ILC) and its application into the control of robotic manipulators. It focuses on investigation of the influence of the parameters of BEA over the convergence rate of the ILC process. This is performed first through a computer simulation. This simulation suggests optimal values for the parameters. Afterwards, the estimated results are validated on a physical robotic manipulator arm. Also, this is one of the first reports of applying BEA into robots control.

KEY WORDS: Robotic manipulator, iterative learning control (ILC), bounded error algorithm (BEA), computer simulation

1. INTRODUCTION

The Iterative Learning Control (ILC) was first introduced by Arimoto in 1984 [1]. ILC for robotic arms is a class of self-tuning algorithms, which repeatedly implement assigned tasks of robot motions in order to minimize trajectory-tracking errors [1,2].

The main advantage of the ILC is that it can compensate trajectory errors, caused by the imprecise dynamics system model as well as its deterministic disturbances. However, there are some deficiencies, which can even make the ILC inapplicable in real life operations. Despite the convergence of the ILC, the tracking performance, at early iterations, can result in a significant increase in tracking errors (see Fig. 1.a), related to violating the constraints in the motion of the joints of the robotic arm. The ILC procedure cannot be completed successfully in this case. This is called the transient growth problem. [3,4].

Previous research showed that this problem can be eliminated by using the Bounded Error Algorithm (BEA) for ILC [3,5]. This algorithm limits the maximum allowed divergence between the desired and the executed trajectory, resulting in limiting the maximum tracking error and solving the transient growth problem (see Fig. 1.b). This

*Corresponding author e-mail: k.yovchev@fmi.uni-sofia.bg

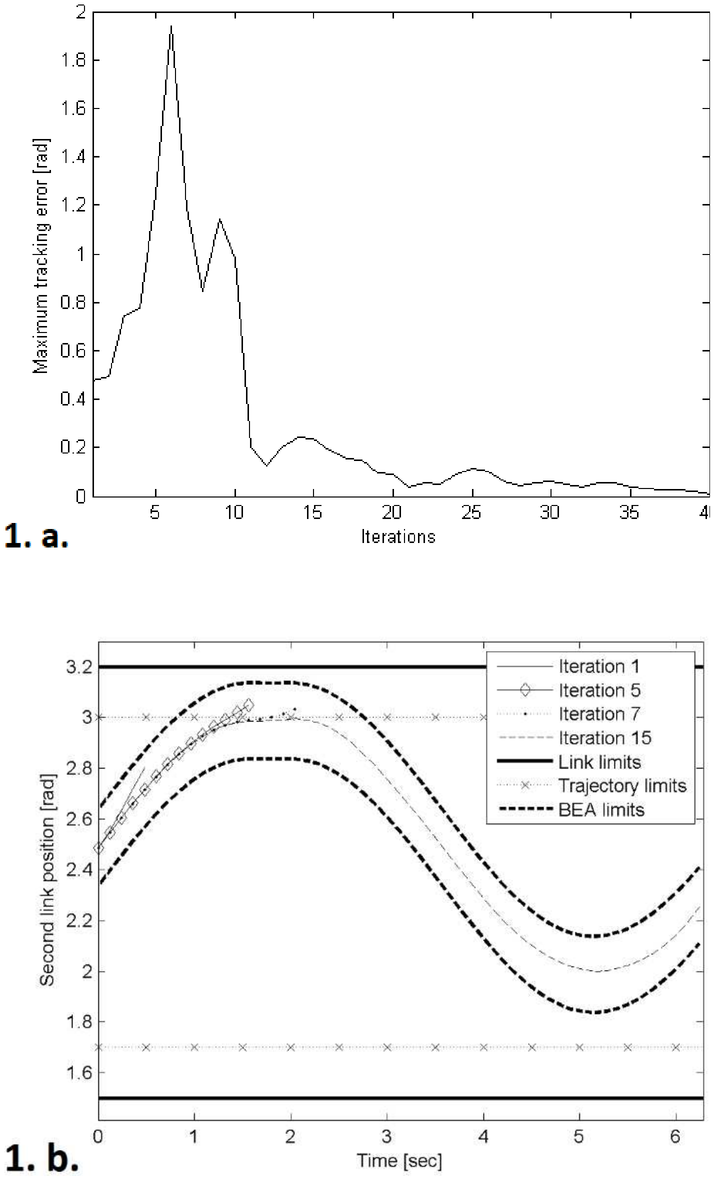


Fig. 1. (a) Transient growth problem; (b) Solution with the Bounded Error Algorithm

algorithm is proven to be robust and convergent [5]. However, there are some parameters, which influence the performance of the BEA. This research investigates them and computes the optimal values.

The paper is organized as follows. Section 2 formulates the problem. Section 3 analyses the parameters through a computer simulation. Section 4 validates those results on a real robotic arm.

2. Problem formulation

Let's consider an articulated robot manipulator (as is in [3]) with 6 degrees of freedom and dynamic model, based on the Lagrange's formulation of equations of motion in the space of generalized coordinates:

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

where $\boldsymbol{q}$ is the 6×1 vector of generalized coordinates (joint angles); $\boldsymbol{A}(\boldsymbol{q})$ is the 6×6 symmetric positive-definite inertia matrix; the 6×1 vector $\boldsymbol{b}(\boldsymbol{q},\dot{\boldsymbol{q}})$ takes into account the Coriolis and centrifugal torques; $\boldsymbol{D}=diag\left\{\delta_1,\ldots,\delta_6\right\}$ denotes the diagonal 6×6 matrix of the coefficients of viscous friction; $\boldsymbol{g(q)}$ is the 6×1 vector, representing gravity torques; $\boldsymbol{f}=[f_1 sign(\dot{q}_1)\ldots\ f_6 sign(\dot{q}_6)]^T$ is the vector of coefficients of Coulomb friction, and $\boldsymbol{u}=\boldsymbol{u}_l+\boldsymbol{u}_c$ is the 6×1 vector of generalized torques where $\boldsymbol{u}_l$ and $\boldsymbol{u}_c$ are feed-forward and feedback terms, respectively. The allowable set of generalized torques is a rectangular hyper-parallelepiped, where $u_i\in[-U_i^{\max},U_i^{\max}]$.

The input trajectory $\boldsymbol{u}_l$ is the feed-forward term of the control law $\boldsymbol{u}$ in (1), and $l=1,\ldots,N$ is the current iteration number; $\boldsymbol{q}_l$ is the actual output trajectory; and $\boldsymbol{q}_d$ is the desired output trajectory. The offline computed feed-forward term $\boldsymbol{u}_{l+1}$ decreases the tracking error of the robot's motion on the next iteration. The realization of the considered ILC scheme requires the specification of feed-forward and feedback controllers, respectively [3]. The BEA for ILC is based on this update control law that improves the feed-forward control term:

$$\begin{aligned}&\boldsymbol{u}_{l+1}(t)=\boldsymbol{u}_l(t)\\&+\begin{cases}\boldsymbol{L}(\boldsymbol{q}_l(t))[\ddot{\boldsymbol{q}}_d(t)-\ddot{\boldsymbol{q}}_l(t)+L_v(\dot{\boldsymbol{q}}_d(t)-\dot{\boldsymbol{q}}_l(t))+L_p(\boldsymbol{q}_d(t)-\boldsymbol{q}_l(t))] & t\in[0;T_l^s]\\ 0 & t\in(T_l^s;T]\end{cases},\end{aligned} \tag{2}$$

where $\boldsymbol{L}(\boldsymbol{q}_l(t))$, $l=0,1,\ldots,N$ is a learning operator; $\boldsymbol{u}_0(t)\equiv 0$ is the initial feed-forward control input; the learning gains L_p and L_v are positive constants, $t\in[0,T]$ denotes time, where $[0,\ T]$ is the robot tracking time interval [5].

The feedback term is considered to be the following:

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

where $\boldsymbol{q}_l$ and $\dot{\boldsymbol{q}}_l$, $l=1,\ldots,N$ are respectively, the feedback generalized coordinates and velocities of the output trajectory; $\hat{A}$, $\hat{b}$, $\hat{D}$, $\hat{g}$ and $\hat{f}$ are the corresponding es-

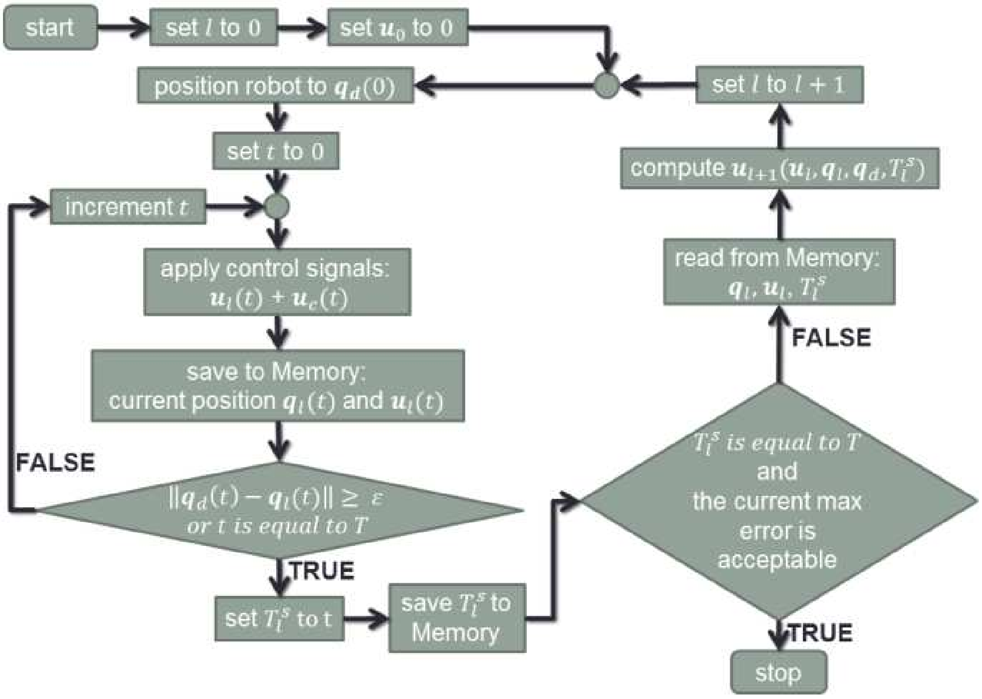

start
set l to 0
set $\boldsymbol{u}_0$ to 0
position robot to $\boldsymbol{q}_d(0)$
set l to $l+1$
set t to 0
compute $\boldsymbol{u}_{l+1}(\boldsymbol{u}_l, \boldsymbol{q}_l, \boldsymbol{q}_d, T_l^s)$
increment t
apply control signals: $\boldsymbol{u}_l(t) + \boldsymbol{u}_c(t)$
read from Memory: $\boldsymbol{q}_l, \boldsymbol{u}_l, T_l^s$
FALSE
save to Memory: current position $\boldsymbol{q}_l(t)$ and $\boldsymbol{u}_l(t)$
FALSE
T_l^s is equal to T and the current max error is acceptable
$\|\boldsymbol{q}_d(t) - \boldsymbol{q}_l(t)\| \geq \varepsilon$ or t is equal to T
TRUE
set T_l^s to t
save T_l^s to Memory
TRUE
stop

Fig. 2. BEA workflow chart.

timates of $\boldsymbol{A}$, $\boldsymbol{b}$, $\boldsymbol{D}$, $\boldsymbol{g}$ and $\boldsymbol{f}$ from (1). The feedback gains K_v and K_p are positive constants [5].

The Bounded Error Algorithm can be illustrated with the workflow chart, shown in Fig. 2. From the update control law (2) follows, that the learning gains L_p and L_v are positive constants, which should be preselected. From the workflow chart, there is another constant ε which determines whether iteration must be stopped or continued during the trajectory tracking process. The goal of this paper is to find the influence of those parameters and to suggest their optimal values, which will lead to the fastest convergence rate.

3. Investigation and optimization of the parameters through a computer simulation

The computer simulation had a similar setup as the one presented in [4]. The Puma 560 robotic manipulator with six degrees of freedom was simulated. Two different sets of parameters for the dynamics model of the robot were used. One reported from Armstrong in 1986 [6] and another one reported by Tarn [7]. This allows taking in consideration the errors, due to imprecise modelling of the robot arm dynamics in a simulation procedure, applying BEA [3].

The goal of the first group of experiments was to evaluate the influence of the L_p and the L_v parameters. This was performed by executing multiple BEA procedures with equal desired trajectory and different combinations of values for those parameters. The ε was set to 1, which is high enough and at the same time prevents the transient growth problem. The number of iterations was limited to 25. The results are illustrated in Figs 3 and 4, where the horizontal axis represents the iterations number and the vertical represents the maximum error for the corresponding iteration. Higher convergence rate is achieved when the value of L_p is significantly higher than the value of L_v. This confirms the results, presented in another research [8], but for a completely different robotic configuration.

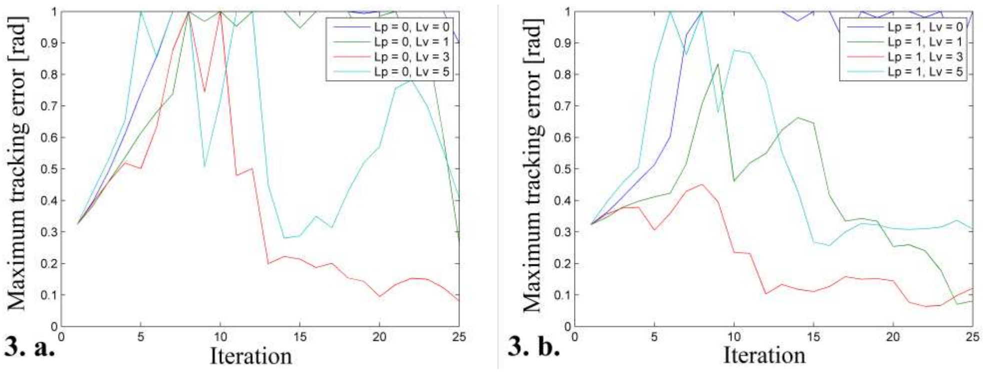


Fig. 3. Influence of the L_p and the L_v parameters: (a) L_p value of 0; (b) L_p value of 1.

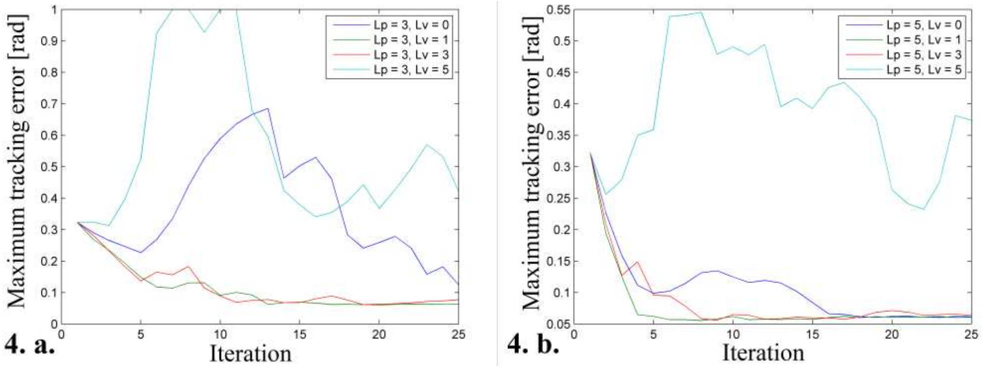


Fig. 4. Influence of the L_p and the L_v parameters: (a) L_p value of 3; (b) L_p value of 5.

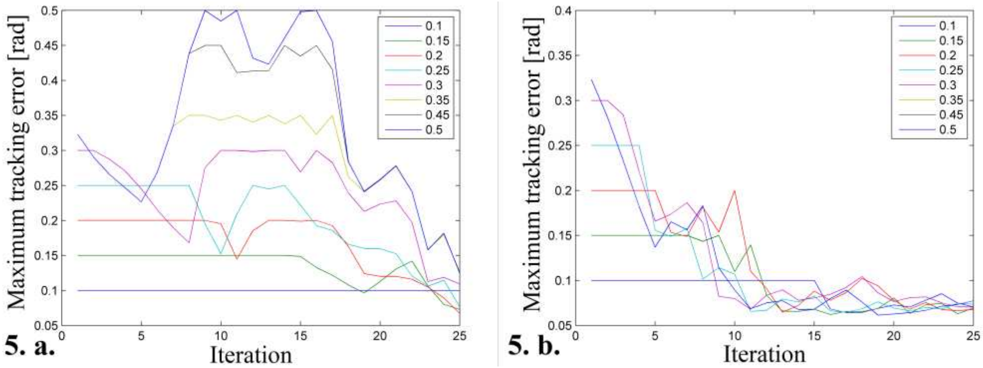


Fig. 5. Influence of ε on the convergence rate of the BEA: (a) non-optimal values for $L_p = 3$ and $L_v = 3$; (b) with more optimal values of $L_p = 3$ and $L_v = 0$.

The next group of experiments investigated how the value of ε changed the convergence rate of the BEA for ILC. Figure 5. a shows the results for non-optimal values for L_p and L_v. It can be concluded that for such values the value of ε have a low impact on the convergence rate. However, the presented in Fig. 5.b results show that for optimal values of L_p and L_v the convergence rate is much higher for high values of ε.

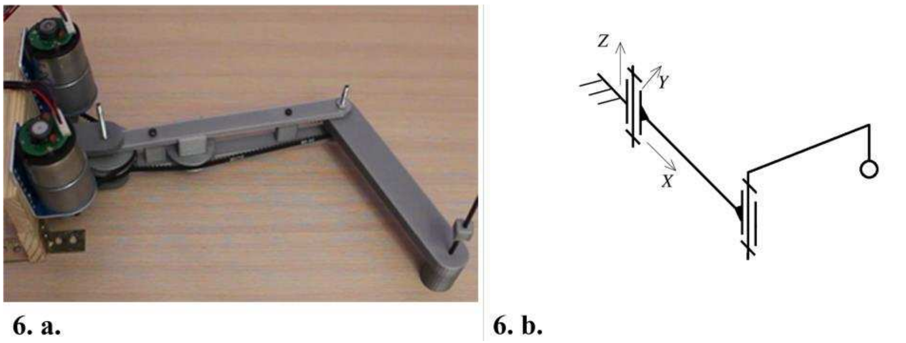
After those experiments it can be concluded, that faster convergence rate of the BEA for ILC can be achieved if the selected value for L_p is significantly higher than the selected value for L_v and if the maximum possible value for ε is chosen.

4. Applying the Bounded Error Algorithm on a robotic arm

BEA was incorporated into the control of a SCARA type two link robot (see Fig. 6). The motors of that robot were brushed DC geared motors. The control board allowed setting their torques at any moment. Those motors are very similar to those simulated in the above section. They had Hall sensor type quadrature encoders. The robot had a belt driven mechanics. This resulted in less than 0.0017 radians positioning precision for both links.

Figure 7 is a screenshot of the robot arm real-time telemetry. The green trajectory, shown in Fig. 7, is the desired one. The two yellow lines represent the robot links. The yellow circle illustrates where the robot end effector should be at the desired time since the trajectory had a fixed duration. The red trajectory represents the tracked trajectory.

In all of the experiments, the parameter L_v was set to 0.01 and L_p was set to 5, as suggested by the analyses in the previous chapter. Multiple values for ε were tested. The first experiment used a value of 0.1, which leads to a high number of iterations

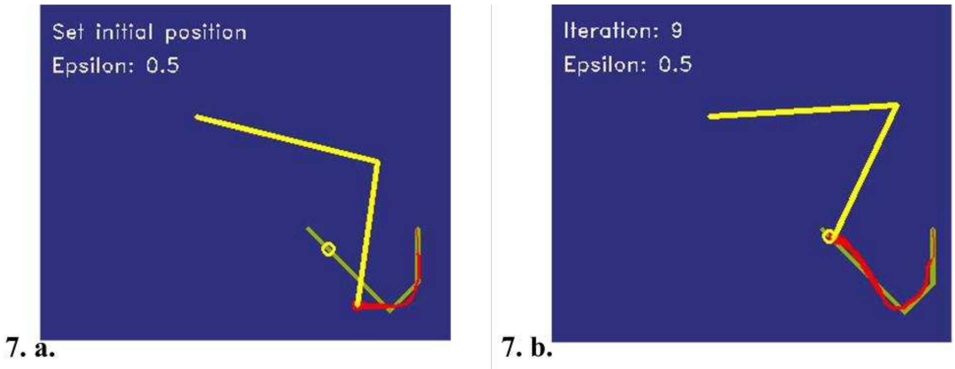

6. a.

6. b.

Fig. 6. (a) Two link SCARA type robotic arm; (b) Robot kinematics scheme.

and quite a slow convergence process. The iteration was very short, but the time for returning the robot arm to the initial position, between two iterations, slowed down the whole process. With increasing the value for ε the iteration time is increasing, too. The best value was experimentally proven to be 0.5, which resulted in a total count of about 12 iterations, before correct execution of the desired trajectory. Figure 7 shows the executed trajectories at 3$^{\text{rd}}$ and 9$^{\text{th}}$ iterations.

It is worth mentioning, that ILC without the BEA, was also tested for the given trajectory. But it needed a higher number of iterations. That was due to the selected trajectory, which in the middle needed a change of the first motor rotation direction. The error accumulated on that point and needed more iteration to be reduced. The BEA prevented such accumulation simply by premature termination of the trajectory tracking. Last experiment used a straight line trajectory. Higher value of ε reduced the number of iterations.


Set initial position
Epsilon: 0.5

7. a.

Iteration: 9
Epsilon: 0.5

7. b.

Fig. 7. SCARA robot recorded (red curve) and desired (green curve) trajectories at: (a) 3-rd iteration; (b) 9t-h iteration.

5. Conclusion

In conclusion, this paper investigated the influence of different combinations of the BEA for ILC parameters on two completely different types of robots. The investigation started on a computer simulated 6 degree of freedom robotic arm. Then, the results were populated on a physical robotic manipulator with a different configuration. This paper reports the use of the BEA for ILC on a real robotic system. This research proposes how the BEA parameters should be selected for achieving optimal convergence rate and confirmed the following statements:

- BEA can be applied safely on a robotic arm;
- The positioning error must have a higher priority than the velocity error within the ILC update procedure;
- Higher value of BEA's ε leads to a faster convergence rate;
- For some cases the BEA is faster than classic ILC, because it prevents error accumulation when movement direction changes.

Acknowledgment

This work is supported by the Fund for Scientific Research at Sofia University "St. Kl. Ohridski" under grant 153/2017.

References

[1] Arimoto, S., S. Kawamura, F. Miyazaki. Bettering Operation of Dynamic Systems by Learning: A New Control Theory for Servomechanism of Mechatronics Systems, Proc. of 23-rd Conference on Decision and Control, NV, Las Vegas, 1984, 1064-1068,

[2] Schoöllig, A., R. D'Andrea. Optimization-Based Iterative Learning Control for Trajectory Tracking, In European Control Conference, 2009, 1505-1510,

[3] Yovchev, K., K. Delchev, E. Krastev. Computer Simulation of Bounded Error Algorithm for Iterative Learning Control, In International Conference on Robotics in Alpe-Adria Danube Region, Springer International Publishing, 2016, 136-143.

[4] Longman, R. W. Iterative Learning Control and Repetitive Control for Engineering practice. *International Journal of Control*, **73** (2000), No. **10**, 930-954.

[5] Delchev, K. Iterative Learning Control for Robotic Manipulators: A Bounded-error Algorithm. *International Journal of Adaptive Control and Signal Processing*, **28** (2014), No. 12, 1454.

[6] Armstrong, B., O. Hatib, J. Burdick. The Explicit Dynamic Model and Inertial Parameters of the Puma 560 Arm, Proc. IEEE Int. Conf. Robotics and Automation, USA, San Francisco, **1** (1986), 510-518.

[7] TARN, T. J., A. K. BEJCZY, S. HAN, X. YUN. Inertia Parameters of Puma 560 Robot Arm, Tech. Rep. SSMRL-85-01, MO, St. Louis, Washington University, 1988.

[8] DELCHEV, K. Simulation-based Design of Monotonically Convergent Iterative Learning Control for Nonlinear Systems. *Archives of Control Sciences*, **22** (2012), No. **4**, 371–384.