



# Payload-adaptive iterative learning control for robotic manipulators

Kaloyan Yovchev and Lyubomira Miteva

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

**Correspondence:** Kaloyan Yovchev ([k.yovchev@fmi.uni-sofia.bg](mailto:k.yovchev@fmi.uni-sofia.bg))

Received: 15 December 2021 – Revised: 28 February 2022 – Accepted: 9 April 2022 – Published: 4 May 2022

**Abstract.** The main purpose of the iterative learning control (ILC) method is to reduce the trajectory tracking error caused by an inaccurate model of the robot's dynamics. It estimates the tracking error and applies a learning operator to the output control signals to correct them. Today's ILC researchers are suggesting strategies for increasing the ILC's overall performance and minimizing the number of iterations required. When a payload (or a different end effector) is attached to a robotic manipulator, the dynamics of the robot change. When a new payload is added, even the most accurately approximated model of the dynamics will be altered. This will necessitate changes to the dynamics estimates, which may be avoided if the ILC process is used to control the system. When robotic manipulators are considered, this study analyses how the payload affects the dynamics and proposes ways to improve the ILC process. The study looks at the dynamics of a SCARA-type robot. Its inertia matrix is determined by the payload attached to it. The results show that the ILC method can correct for the estimated inertia matrix inaccuracy caused by the changing payload but at the cost of more iterations. Without any additional data of the payload's properties, the suggested technique may adjust and fine-tune the learning operator. On a preset reference trajectory, the payload adaptation process is empirically tested. When the same payload is mounted, the acquired adaptation improvements are then utilized for another desired trajectory. A computer simulation is also used to validate the suggested method. The suggested method increases the overall performance of ILC for industrial robotic manipulators with a set of similar trajectories but different types of end effectors or payloads.

## 1 Introduction

Industrial robotic arm manipulators are machines that must track trajectories with high accuracy. This trajectory tracking process depends on the correct estimation of the parameters of the dynamics of both the robot itself and its payload. As a payload both the end effector and the object with which the robot interacts may be considered. Even if the dynamics of the robot is estimated precisely, the control system will have to compensate for the error introduced by the unknown dynamics or the variation of the dynamics of the payload. These uncertainties of variations will introduce trajectory tracking errors.

Nowadays, many researchers are working on adaptation of the control to the payload of the robot. Truong et al. (2019) consider control for robots with large payload variations. They are proposing backstepping sliding mode control and

are utilizing a fuzzy logic system for adjusting the control gains based on the output of the non-linear disturbance observer to compensate for the payload. Lee et al. (2020a) propose the time delay control as a promising control technique for application to robot manipulators. They are proposing the idea of adaptive gain dynamics and show through experiments that their method can be used for the control of a robot which operates under significant payload changes. In later research of Lee et al. (2020b), they consider a payload-adaptive PID control that is simple, model-free, and robust against payload variations. Their control is efficient under substantial payload variations. However, the PID type of control is working in the presence of a trajectory error and cannot be applied when a precision trajectory tracking is required. Reducing the trajectory tracking error is important, even when human-machine collaboration is considered by the modern robotic

control systems. Researchers Hu et al. (2020) proposed an integrated direct or indirect adaptive algorithm which consists of a generalized momentum-based indirect adaptation law to estimate the payload online. This research will propose the usage of the iterative learning control (ILC) techniques which are widely used for precise trajectory tracking control.

ILC is a non-dual kind of adaptive control that was first proposed by Uchiyama (1978) and further developed by Arimoto et al. (1984a, b) for systems that function in a repeatable pattern. The ILC procedure will begin by making an educated guess about the required output control signals, which will be generated using the approximated mathematical parameters of the model. The ILC will thereafter require multiple iterations. The approach evaluates the tracking error after each iteration and adjusts the output control signals using a particular learning operator. When a pre-set acceptable accuracy threshold is reached, the iterative control procedure will be terminated. Consequently, a biologically inspired iterative self-learning process is created.

The ILC must generate a convergent process before it can be applied to a robot system. In today's literature, the convergence of the ILC approach has been thoroughly researched. Heizinger et al. (1989) proved the convergence of the ILC when non-linear systems (such as robotic manipulators) are considered. Longman and Huang (2002) observed an issue with the formation of a transient error during the first few iterations. This issue necessitated more investigation due to the mechanical features of the robotic manipulators which constrain their state space. This issue prevents the ILC from being directly applicable to constrained state space robotic manipulators. The constrained output ILC (COILC) method, described in Yovchev et al. (2020a), can be applied for control of robotic manipulators without imposing any extra constraints on their workspace, as well as to solve the problem of transient growth error. Robustness and convergence of the COILC are proven when an estimation of the inertia matrix is used as a learning operator.

The focus of today's ILC investigation is on improving the overall performance of the ILC and for reducing the required number of ILC executions (iterations) for generating the control signals for precise trajectory tracking when payload variations are considered. The performance of the ILC methods can be improved by a range of machine learning methods (Wei-Liang et al., 2017; Zhang et al., 2019). The work of Nemec et al. (2017) provides an adaptive ILC method to achieve smooth and safe manipulation of fragile items, with the adaptation supervised by reinforcement learning. Neural networks (Patan et al., 2017; Xu and Xu, 2018) or fuzzy neural network methods (Wang et al., 2008; Wang et al., 2014) are used within ILC to reduce the uncertainty of the model used for the design of the controller. The basis-motion torque composition approach (Sekimoto et al., 2009) has been proposed as a solution for the main disadvantage of the ILC,

that the ILC requires a new learning process for achieving a different motion (Tanimoto et al., 2017).

In our previous research (Yovchev and Miteva, 2021) we investigated the influence of the ILC learning operator over the performance of the ILC when considering the need for using similar desired trajectories. The research by Yovchev and Miteva (2021) showed that the learning operator can be tuned for the given job, and this analysis can then be used to increase ILC performance when the job is modified. To put it another way, if the job is to paint vehicle doors, the ILC will modify its learning operator to shorten the time required for convergence when computing the precise control signals for the execution of required trajectory for painting another car door. The different learning operators are generated using a modified version of the estimated inertia matrix of the robotic manipulator. The modification analysis is done offline, and the algorithm does not require additional specific hardware. The proposed adaptation process is dependent on only one additional gain by which the estimated inertia matrix is modified when the learning operator is optimized. This research will consider this gain as a multiplier gain of the inertia matrix. The current research will extend the adaptation process of the ILC learning operator by adding an additional adaptation gain (called fine-tuning gain) which will be added element-wise to the modified estimation of the inertia matrix. This will allow for fine tuning of the adaptation of the learning operator. Also, this research will consider the changes of the dynamics of the robotic manipulator when a different payload (or different end effector) is attached to the robot. Even when a simple SCARA-type robot is considered, the dynamics and, therefore, the inertia matrix of the robot are dependent on the attached payload. The ILC process most likely will compensate for the error of the estimated inertia matrix caused by the different payload but will require a higher number of iterations. The proposed approach will adapt and fine-tune the inertia matrix estimation without the need of additional knowledge of the inertia characteristics of the payload. The adaptation process will be done on a pre-defined reference trajectory, and the adaptation gains found will be used for any other desired trajectory when the same payload is attached.

This paper is organized as follows. Section 2 formulates the problem of applying ILC to trajectory tracking of a robotic manipulator with a payload and proposes an approach for ILC adaption for a specific payload. Section 3 validates the proposed algorithm through a numerical experiment and presents a computer simulation. Section 4 summarizes the results and proposes future research directions.

## 2 Formulation of payload-adaptive ILC

Figure 1 shows the standard ILC scheme.  $P$  denotes the robot arm,  $L$  represents the learning controller,  $M$  represents the control's system memory,  $l = 1, \dots, N$  denotes the current it-

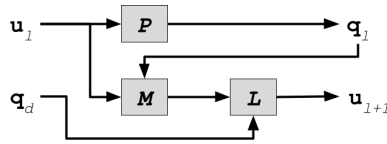


Figure 1. Iterative learning control scheme.

eration number,  $N$  represents the total number of iterations,  $u_l$  represents control law's the feed-forward term for iteration  $l$ ,  $q_l$  denotes the output trajectory for iteration  $l$ , and  $q_d$  represents the desired output trajectory. After each iteration, the feed-forward term  $u_{l+1}$  is computed offline and is used to reduce the trajectory tracking error for iteration  $l+1$ .

The synthesis of the learning update law of the conventional ILC requires the definition of feed-forward and feedback controllers. The update control law iteratively improves the feed-forward control term by the following rule:

$$u_{l+1} = u_l + L(q_l)(\ddot{q}_d - \ddot{q}_l), \quad (1)$$

where  $L(q_l(t))$ ,  $l = 0, 1, \dots, N$  is a learning operator,  $u_0(t) \equiv 0$  is the initial feed-forward control input,  $\ddot{q}_d$  are the desired joint torques,  $\ddot{q}_l$  is the output joint torque during iteration  $l$ , and  $t \in [0, T]$  denotes time, where  $[0, T]$  is the trajectory tracking time interval.

The COILC procedure described by Yovchev et al. (2020a) can be used for the control of any type of robotic manipulator with  $k$  joints and state space constraints (defined as  $Q_j^{\min} \leq q_j \leq Q_j^{\max}$ ,  $1 \leq j \leq k$ ), which is supposed to minimize the tracking error of any attainable desired trajectory  $q_d$ :

- Start with initial iteration number  $l = 0$ , and proceed with the iterative procedure.
- Starting from the initial position  $q_l(0)$ , the system tracks the desired trajectory under the control  $u_l(t)$  until the first moment  $S_l$  for which there exists  $j : 1 \leq j \leq k$ , and either  $q_j^l(S_l) = Q_j^{\min}$  or  $q_j^l(S_l) = Q_j^{\max}$  or the end position  $q_l(T)$  is reached. When  $t = S_l$ ,  $S_l \in (0, T]$ , the tracking process stops.
- After the current tracking performance has finished, the learning controller updates the feed-forward control term according to the following learning update law:

$$u_{l+1}(t, S_l) = u_l(t, S_{l-1}) + u_l^*(t, S_l),$$

$$u_0(t, S_{-1}) \equiv u_0(t),$$

$$u_l^*(t, S_l) = \begin{cases} L(q_l(t), t)(\ddot{q}_d(t) - \ddot{q}_l(t)), & t \in [0, S_l], \quad S_l \in (0, T]; \\ 0, & \forall t \in (S_l, T]. \end{cases} \quad (2)$$

- Calculate the current maximum trajectory tracking error, defined as  $|\delta q_l(t)|_{\infty}$ . If the error is less than or equal to a predefined acceptable accuracy, then exit from the learning procedure, or else set  $l = l + 1$ , and go to step (b).

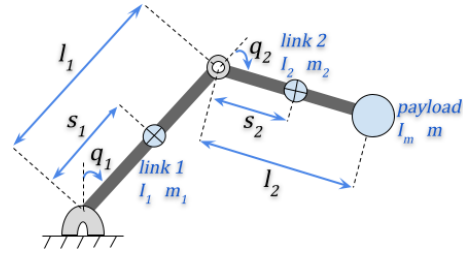


Figure 2. Schematics of SEIKO Instruments TT-3000 robot with attached payload.

Let  $\hat{A}$  be the approximation of the inertia matrix  $A$ . Then the ILC is robust and convergent if the learning operator from Eq. (1) is chosen as  $L(q) \equiv \hat{A}(q)$  (Delchev, 2012). The approximated inertia matrix evaluates how each joint's parameters influence robot motion. The convergence rate will be improved by a better estimation  $\hat{A}$  of  $A$ .

As stated in the research by Yovchev and Miteva (2021), the performance of the ILC method depends on the selection of the estimation. The research proposed an additional parameter of the COILC – the scalar learning gain  $s$  by which the matrix  $\hat{A}$  will be multiplied and the learning operator will become  $L(q) \equiv s\hat{A}(q)$ . This will alter the  $u_l^*(t, S_l)$  from the COILC update law (Eq. 2) as follows:

$$u_l^*(t, S_l) = \begin{cases} s\hat{A}(q_l(t))(\ddot{q}_d(t) - \ddot{q}_l(t)), & t \in [0, S_l], \\ S_l \in (0, T]; \\ 0, & \forall t \in (S_l, T]. \end{cases} \quad (3)$$

This scalar is used to adapt the ILC process to a specific set of trajectories. The goal of the current research is to propose a new approach for adaptation of the ILC method for the specific type of executed task. The robot's dynamics is dependent of the end effector of the robot. When the end effector is changed, the estimation of the inertia matrix must be recalculated.

This research will consider the well-known SEIKO Instruments SCARA-type robot TT-3000. The kinematics of this robot is shown in Fig. 2. The joint lengths are  $l_1 = 250$  mm and  $l_2 = 220$  mm. The dynamical equations of motions and the parameters of the dynamics of this robot are described in research by Shinji and Mita (1990).

The standard formulation of the dynamics of this robot is formulated by Shinji and Mita (1990) as follows:

$$\begin{aligned} a_{11}(q_2)\ddot{q}_1 + a_{12}(q_2)\ddot{q}_2 - 2m_2l_1s_2\sin(q_2)\dot{q}_1\dot{q}_2 \\ - m_2l_1s_2\sin(q_2)\dot{q}_2^2 + D_1\dot{q}_1 + f_1 = u_1 \\ a_{12}(q_2)\ddot{q}_1 + a_{22}\ddot{q}_2 + m_2l_1s_2\sin(q_2)\dot{q}_1^2 \\ + D_2\dot{q}_2 + f_2 = u_2. \end{aligned} \quad (4)$$

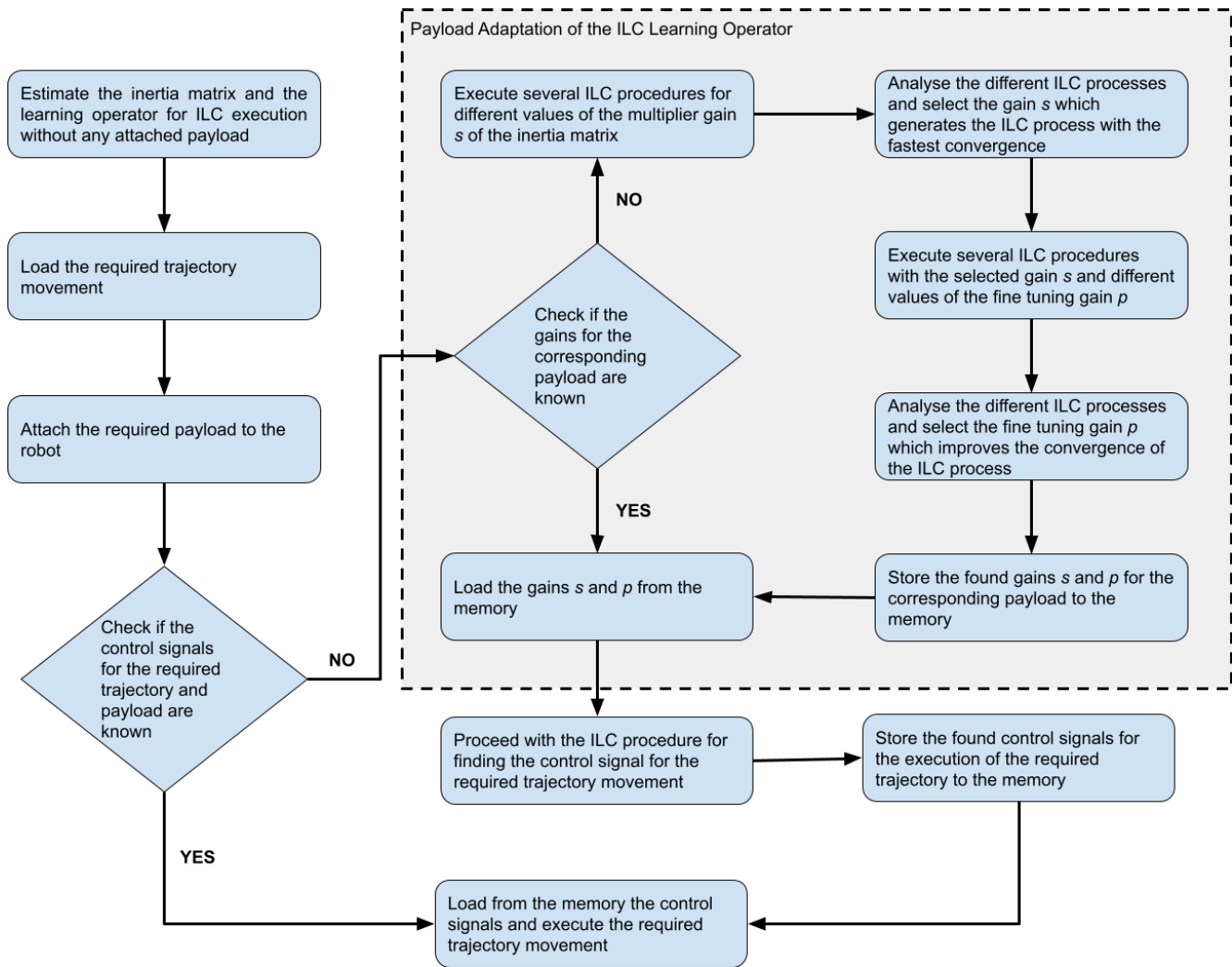


Figure 3. Schematics of the workflow of the payload-adaptive ILC.

The inertia matrix  $\mathbf{A}(\mathbf{q}) = [a_{ij}(q_2)]$ ,  $1 \leq i, j \leq 2$ , where

$$\begin{aligned}
 a_{11}(q_2) &= I_1 + m_1 s_1^2 + I_2 + m_2(I_1^2 + s_1^2) \\
 &\quad + 2m_2 l_1 s_2 \cos(q_2) \\
 a_{12}(q_2) &= I_2 + m_2 s_2^2 + m_2 l_1 s_2 \cos(q_2) \\
 a_{21}(q_2) &= a_{12}(q_2) \\
 a_{22}(q_2) &= a_{22} = I_2 + m_2 s_2^2,
 \end{aligned} \tag{5}$$

where for  $i = 1, 2$ ,  $D_i$  is the viscous damping,  $f_i$  is Coulomb friction,  $l_i$  is the link length,  $s_i$  is the position of the centre of mass,  $m_i$  is the total mass of the link,  $I_i$  is the inertia of the link about its centre of mass, and  $u_i$  is the driving torque of the link.

When we consider the attached payload, then the inertia matrix  $\bar{\mathbf{A}}(\mathbf{q}) = [\bar{a}_{ij}(q_2)]$ ,  $1 \leq i, j \leq 2$  is as follows:

$$\begin{aligned}
 \bar{a}_{11}(q_2) &= a_{11}(q_2) + I_m + m(l_1^2 + l_2^2 + 2l_1 l_2 \cos(q_2)) \\
 \bar{a}_{12}(q_2) &= a_{12}(q_2) + I_m + m(l_1^2 + l_1 l_2 \cos(q_2)) \\
 \bar{a}_{21}(q_2) &= \bar{a}_{12}(q_2) \\
 \bar{a}_{22}(q_2) &= \bar{a}_{22} = a_{22} + I_m + m l_2^2,
 \end{aligned} \tag{6}$$

where the payload is attached to the second link and has the following characteristics: mass  $m$  and inertia about its centre of mass  $I_m$ .

The payload (the attached end effector) is supposed to change the dynamics characteristics of the robotic manipulator, as can be seen from Eqs. (5) and (6). This will lead to a different performance of the ILC process when different payloads are used. Also, for optimal ILC the inertia characteristics of the payload must be known. The goal of this research is to propose a new approach with an added ILC adaptation phase when a new payload is attached. The estimation

$\hat{\mathbf{A}}(\mathbf{q}) = [\hat{a}_{ij}(q_2)]$ ,  $1 \leq i, j \leq 2$  of the inertia matrix  $\mathbf{A}(\mathbf{q})$  of the robot without payload will be modified by execution of a predefined training trajectory when a new payload is attached.

The estimated inertia matrix  $\hat{\mathbf{A}}(\mathbf{q})$  considers how the characteristics of each joint affect the robot's motion. Correct estimation of this matrix will lead to a robust and convergent ILC process. The learning operator cannot be randomly chosen due to a slow or non-convergent ILC process. For the purposes of this research, we will assume that the inertia matrix  $\hat{\mathbf{A}}$  is estimated correctly for a robotic manipulator with no payload. We will introduce two additional parameters of the COILC – the scalar learning multiplier gain  $s$  and fine-tuning gain  $p$  by which the matrix  $\hat{\mathbf{A}}$  will be modified to be used as a learning operator:  $\mathbf{L}(\mathbf{q}) \equiv s\hat{\mathbf{A}}(\mathbf{q}) + p$ . This will alter the  $\mathbf{u}_l^*(t, S_l)$  from the COILC update law (Eq. 2) as follows:

$$\mathbf{u}_l^*(t, S_l) = \begin{cases} (s\hat{\mathbf{A}}(\mathbf{q}_l(t)) + p)(\ddot{\mathbf{q}}_d(t) - \ddot{\mathbf{q}}_l(t)), & t \in [0, S_l], \\ S_l \in (0, T]; \\ 0, & \forall t \in (S_l, T]. \end{cases} \quad (7)$$

When a new payload is attached, the robot will execute several consequent ILC processes with a predefined reference trajectory. These consequent ILC executions will evaluate multiple pairs of parameters  $(s, p)$ .

The COILC constrains the output trajectory at each iteration  $l$ , so that the state space constraints cannot be violated during the ILC procedure, and the adaptation step can be safely executed.

The workflow of the proposed payload adaptation ILC process is shown in Fig. 3. The steps which are bordered with the dashed rectangle are the new addition to the ILC procedure – the payload adaptation phase. During this phase, multiple ILC procedures are executed to find the optimal multiplier gain  $s$  and fine-tuning gain  $p$  for the current payload. This adaptation requires several additional ILC executions and is applicable when the current payload is supposed to be used for multiple execution of multiple trajectories. The goal of this payload adaption phase is to reduce the number of required ILC iterations for achieving a precise trajectory tracking when the dynamics and the mathematical parameters of the payload are unknown. As a payload, we are referring to both the tools (end effectors) and the objects with which the robot operates.

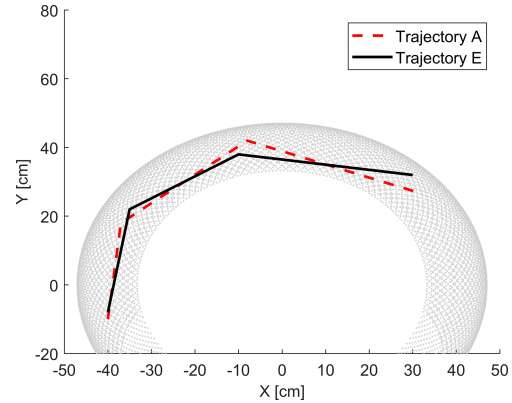
The performance of the proposed approach is evaluated through numerical experiments and computer simulation in the following section.

### 3 Numerical experiments and simulation results

For evaluation on the viability and applicability of the proposed payload-adaptive ILC, we conduct numerical experiments and a computer simulation with a specialized software. For the numerical experiments, we will use GNU Octave software to solve the ordinary differential equations of the dynamics (Eaton et al., 2019). For the computer simulations, we will use the Cyberbotics Ltd. Webots™ robotic

**Table 1.** TT-3000 hard constraints for the computer simulation.

Joint constraint [rad]	$Q_1^{\min}$	$Q_1^{\max}$	$Q_2^{\min}$	$Q_2^{\max}$
	-2.50	2.50	-1.57	1.57



**Figure 4.** The grey dots represent the workspace of the robot. The dashed red and the solid black lines represent two similar desired trajectories A and E in the workspace.

simulator (Michel, 2004) and its simulation of the Universal Robots' UR5e robotic manipulator. The computer simulation will be used for better explanation of the advantages of the payload adaption of the ILC learning operator.

For the numerical experiments of the adaptive COILC algorithm, a computer simulation is used, as described in Sect. 2. The robot will have the following joint constraints  $Q_i^{\min} \leq q_i \leq Q_i^{\max}$ , where  $1 \leq i \leq 2$  as described in Table 1.

The inertia matrix  $\mathbf{A}$  from Eq. (5) of the simulated robot with no payload will be as follows:

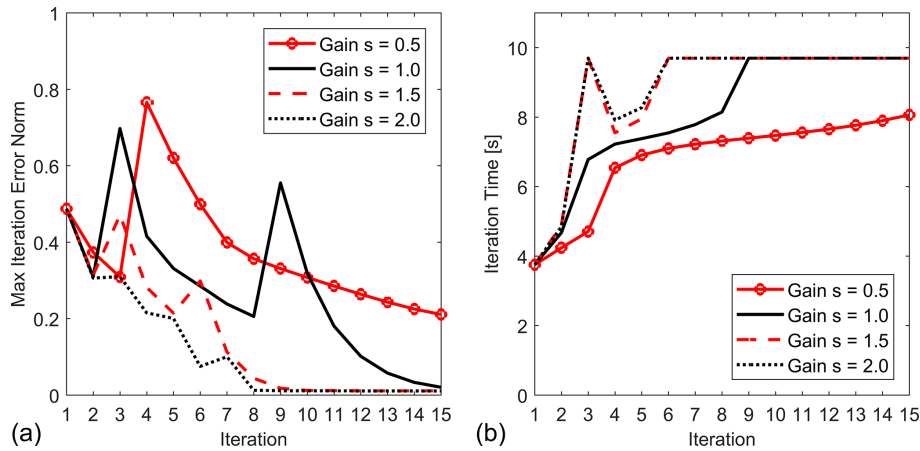
$$\begin{aligned} a_{11}(q_2) &= 6596.2 + 3011.2 \cos(q_2) \text{ kg cm}^2 \\ a_{12}(q_2) &= 1353.4 + 1505.6 \cos(q_2) \text{ kg cm}^2 \\ a_{21}(q_2) &= a_{12}(q_2) \text{ kg cm}^2 \\ a_{22}(q_2) &= 1353.4 \text{ kg cm}^2. \end{aligned} \quad (8)$$

The estimation  $\hat{\mathbf{A}}$  of the matrix  $\mathbf{A}$  which will be used for the initial ILC learning operator will be defined as follows:

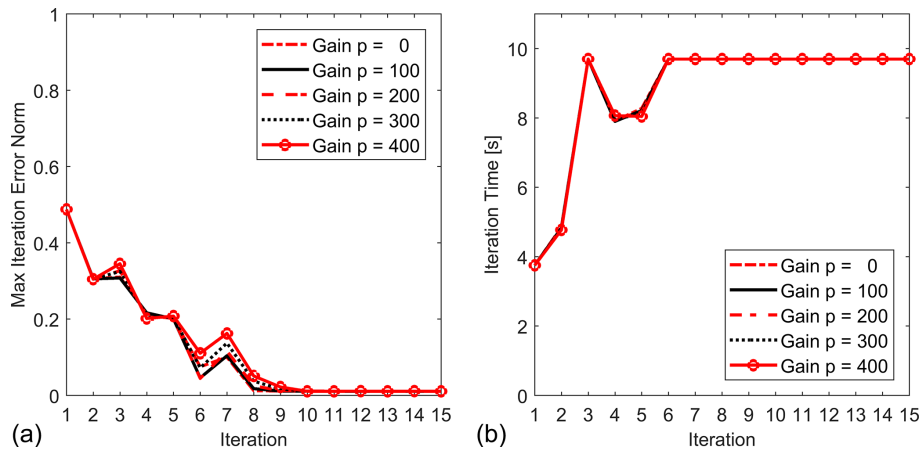
$$\begin{aligned} \hat{a}_{11}(q_2) &= 6600 + 3000 \cos(q_2) \text{ kg cm}^2 \\ \hat{a}_{12}(q_2) &= 1400 + 1500 \cos(q_2) \text{ kg cm}^2 \\ \hat{a}_{21}(q_2) &= \hat{a}_{12}(q_2) \text{ kg cm}^2 \\ \hat{a}_{22}(q_2) &= 1400 \text{ kg cm}^2. \end{aligned} \quad (9)$$

When a payload with inertia about its centre of mass  $I_m$  and mass  $m$  is attached, the inertia matrix  $\bar{\mathbf{A}}$  of the simulated robot will be altered in accordance with the payload as fol-





**Figure 5.** Convergence comparison of the ILC process for Trajectory A for different values of the multiplier gain  $s$  and fine-tuning gain  $p = 0$ : (a) maximum tracking error and (b) iteration times.



**Figure 6.** Convergence comparison of the ILC process for Trajectory A for different values of the fine-tuning gain  $p$  and multiplier gain  $s = 2$ : (a) maximum tracking error and (b) iteration times.

lows:

$$\begin{aligned}
 \bar{a}_{11}(q_2) &= 6596.2 + 3011.2 \cos(q_2) + I_m \\
 &\quad + m(1109 + 1100 \cos(q_2)) \text{ kg cm}^2 \\
 \bar{a}_{12}(q_2) &= 1353.4 + 1505.6 \cos(q_2) + I_m \\
 &\quad + m(484 + 550 \cos(q_2)) \text{ kg cm}^2 \\
 \bar{a}_{21}(q_2) &= \bar{a}_{12}(q_2) \text{ kg cm}^2 \\
 \bar{a}_{22}(q_2) &= 1353.4 + I_m + 484m \text{ kg cm}^2.
 \end{aligned} \quad (10)$$

The trajectories used for adaptation (Trajectory A) and for evaluation (Trajectory E) of the optimized learning operator are shown in Fig. 4. They are trajectories with similar characteristics. It is supposed that after the ILC learning operator is adapted for Trajectory A, then the ILC process will have a better performance (faster convergence rate) when the robot must be trained for execution of Trajectory E.

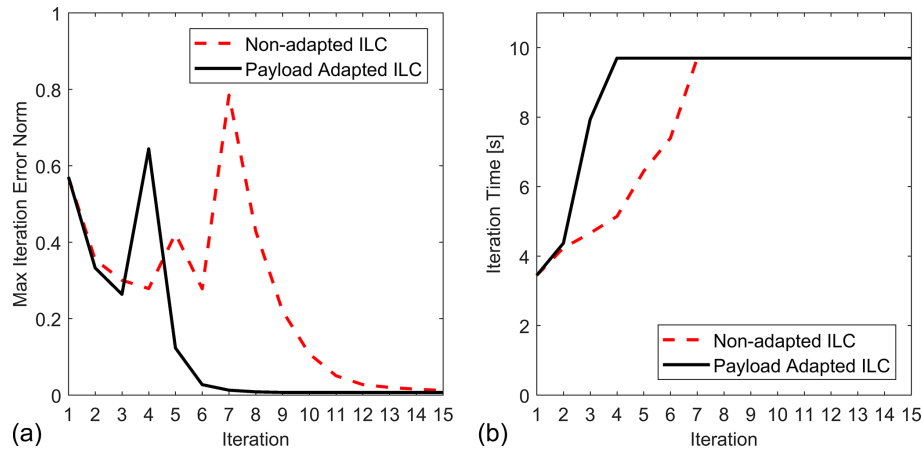
The numerical experiments consider two types of payloads, Payloads A and B, with the following characteris-

tics:  $I_m = 150 \text{ kg cm}^2$  and  $m = 7.5 \text{ kg}$  for Payload A and  $I_m = 52.8 \text{ kg cm}^2$  and  $m = 2.5 \text{ kg}$  for Payload B.

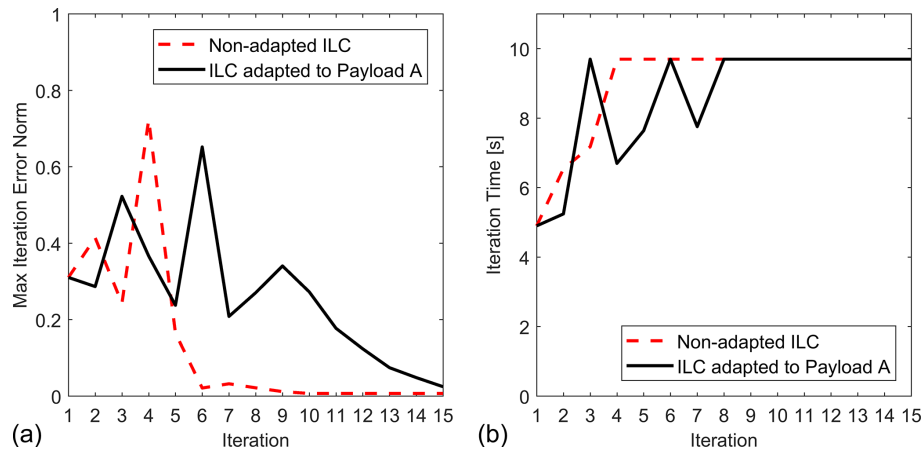
During the initial adaption for Payload A the following values for the multiplier gain  $s$  are considered: 0.5, 1.0, 1.5, and 2.0. The fine-tuning gain  $p$  is set to 0. The COILC is executed with the desired trajectory set to Trajectory A. The convergence of the ILC process is shown in Fig. 5.

Further, the multiplier gain  $s$  is set to the best value found of 2.0, and additional adaptation is done by considering the following values for the fine-tuning gain  $p$ : 0, 100, 200, 300, and 400. The convergence results are shown in Fig. 6.

Figure 7 shows comparison of the generated ILC process when the desired trajectory is set to Trajectory E. The results show that the total ILC execution time is reduced from 109 to 55 s, which is about a 50 % faster ILC process with the learning operator that is adapted to Payload A with gains  $s = 2$  and  $p = 0$ .



**Figure 7.** Convergence comparison of the ILC process for Trajectory E for non-adapted ILC and ILC adapted to the payload: (a) maximum tracking error and (b) iteration times.



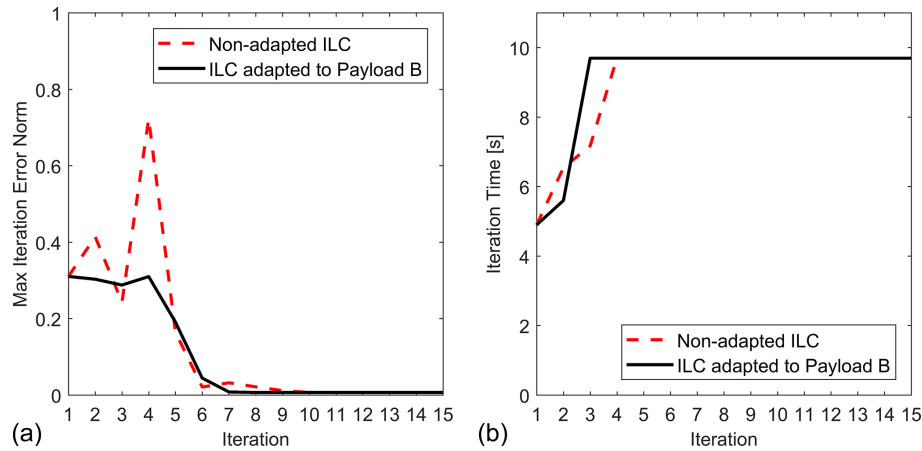
**Figure 8.** Convergence comparison of the ILC process for Trajectory E with Payload B for non-adapted ILC and ILC adapted to Payload A: (a) maximum tracking error and (b) iteration times.

The next experiments are conducted for Payload B. Figure 8 shows the ILC convergence for Trajectory E when non-adapted ILC and ILC adapted to Payload A is used. It can be concluded that with the learning operator adapted to Payload A, the ILC process will need a very high number of iterations for trajectory tracking with Payload B. The ILC process will still be convergent, but the number of iterations will exceed more than 20.

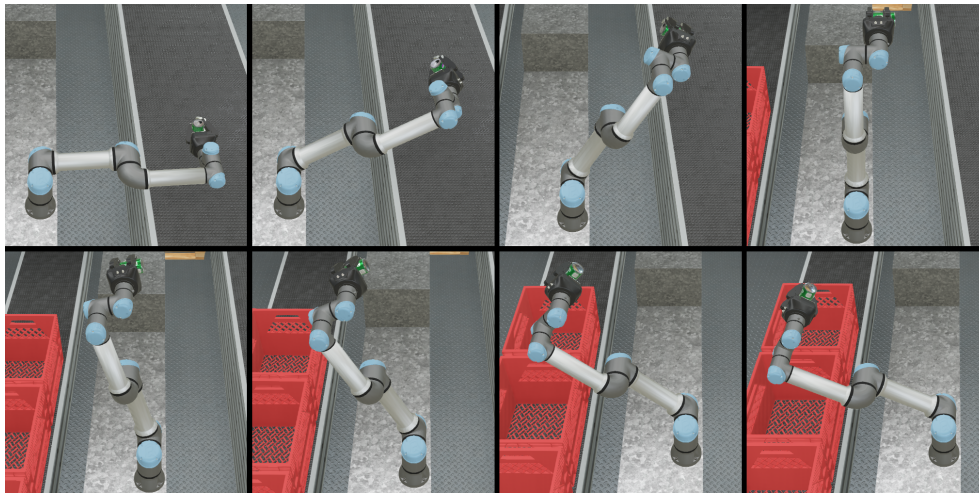
Then, the adaptation process is repeated for Payload B over Trajectory A. The found adaptation gains are  $s = 1.5$  and  $p = 200$ . With the learning operator that is adapted to Payload B, the total ILC execution time for Trajectory E is reduced from 77 to 59 s, which is about a 23 % faster ILC process. The convergence comparison is shown in Fig. 9. The results confirm that after adaption to the payload, the ILC performance will be improved, and a smaller number of iterations will be required. The results from Figs. 8 and 9 also confirm that the payload adaption is a process specific

to the payload. When the payload is changed, the ILC cannot be used with gains which correspond to another payload. Either a new adaption must be executed, or the ILC should use the non-adapted learning operator. The adaption to payload makes sense when a specific payload is used for multiple different trajectories. Such payloads are the different end effectors or tools with which the robot operates.

For the next experiment, a computer simulation is created. For the computer simulation, we consider trajectory movement of the UR5e with a duration of 2 s. All the joints of the robot are fixed, except the second one (the elbow lift joint), which moves from  $-0.01$  rad to  $-2.60$  rad. The robot is in its maximum extended state, which requires the maximum torque when the robot holds the payload in its gripper. For the experiment, two different payloads with a mass of 1 and 7 kg are considered. Some positions from the simulated robot movement are shown in Fig. 10.



**Figure 9.** Convergence comparison of the ILC process for Trajectory E with Payload B for non-adapted ILC and ILC adapted to Payload B: (a) maximum tracking error and (b) iteration times.



**Figure 10.** Positions of the simulated movement of the UR5e robot holding the payload in the gripper.

The desired trajectory of the second link is shown with a thin solid black line in Fig. 11a. The trajectory executions with the two different payloads with the default robot control without applying ILC are shown in Fig. 11a. For the computer simulation, multiple ILC procedures are executed with different values of the multiplier gain. In Fig. 11b the ILC executions are shown for values 0.4 and 0.7 of the multiplier adaptation gain.

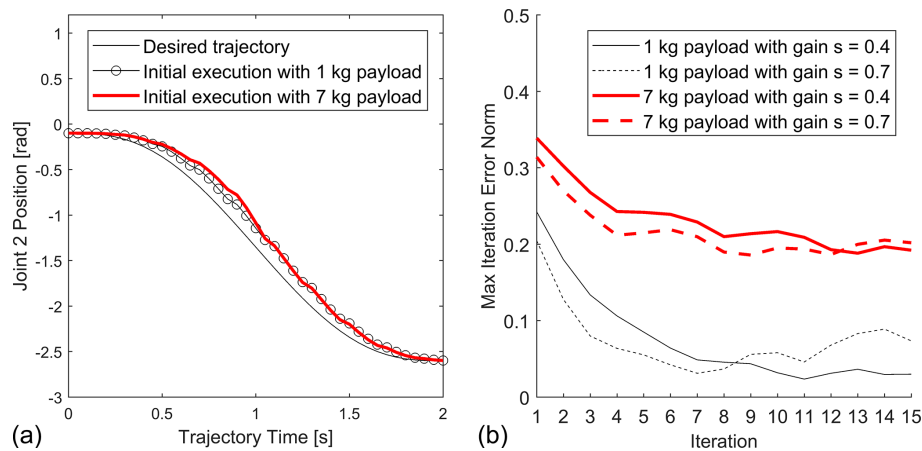
These results clearly show that for different payloads, a different multiplier adaptation gain will change the behaviour of the ILC process. For the payload of 1 kg, we can conclude the higher multiplier gain  $s = 0.7$  will result in a non-convergent ILC process. For the higher payload of 1 kg, both 0.4 and 0.7 gains will preserve the convergence of the ILC process, but the higher gain will need 9 iterations instead of 13 iterations for the lower to achieve the same value of maximum tracking error. For both payloads, we can use a gain value of 0.4, and the ILC process will minimize the trajectory

tracking error. But with the payload adaptation proposed in this research, the overall performance of the ILC can be improved when the payload is also taken into consideration by the robot control. Fewer iterations will be needed for minimizing the trajectory tracking error.

#### 4 Conclusion

This research considers the convergence of the ILC process when it is applied to control of the same robotic manipulator with different type of payloads (end effectors). The proposed approach uses the estimation of the inertia matrix of the robotic manipulator with no payload attached as the basis ILC learning operator. Afterwards, an approach for adaptation of this learning operator to a specific payload is proposed. When the robot is equipped with a new payload, the ILC learning operator must be adapted for the changes of





**Figure 11.** (a) Desired trajectory and its execution with different payloads. (b) Comparison of the convergence of the ILC process with different payloads and different values of the multiplier gain  $s$ .

the dynamics model and especially to the inertia matrix of the robot. After this adaption is done, the performance of the ILC will be higher, and the required number of ILC iterations will be reduced for any desired trajectory when this payload is being used. The conducted experiments confirmed the proposed approach. They also concluded that the adaption must be repeated when the payload is changed. This research considers the control of state-space-constrained robotic manipulators, and since the adaption process alters the learning operator, it is required that those constraints are considered by the control method. The research proposes the use of the robust and convergent COILC method. The proposed approach for learning operator adaptation can be considered to be an adaptive extension to the COILC. After the adaption there are no additional computational costs during further execution of the ILC iterations. The proposed approach improves the overall performance of the ILC for industrial robotic manipulators, which are supposed to execute a set of similar trajectories but with different types of end effectors attached or payloads carried. Further research should be done on applying payload adaption to ILC for redundant robotic manipulators. Also, the multiplier and the fine-tuning gains can be considered to be vector gains, and experiments can be conducted to show whether these vector gains can further reduce the required number of ILC iterations. Metaheuristic algorithms (such as genetic algorithms, hill climbing, or simulated annealing) can be used for automatization of the adaption process.

**Data availability.** All the data used in this paper can be obtained from the corresponding author upon request.

**Author contributions.** KY and LM designed the proposed approach algorithm and experiments. LM carried them out and pre-

pared the results. KY prepared the manuscript with contributions from all co-authors.

**Competing interests.** The contact author has declared that neither they nor their co-author has any competing interests.

**Disclaimer.** Publisher's note: Copernicus Publications remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

**Special issue statement.** This article is part of the special issue "Advances in Service and Industrial Robotics – RAAD2021". It is a result of the The 30th International Conference on Robotics in Alpe-Adria-Danube Region, RAAD 2021, Futuroscope-Poitiers, France, 21–23 June 2021.

**Acknowledgements.** The research presented in this paper was supported by the Bulgarian National Science Fund under contract no. KP-06-M47/5-27.11.2020.

**Financial support.** This research has been supported by the Bulgarian National Science Fund (grant no. KP-06-M47/5-27.11.2020).

**Review statement.** This paper was edited by Mohamed Amine Laribi and reviewed by two anonymous referees.

## References

- Arimoto, S., Kawamura, S., and Miyazaki, F.: Iterative learning control for robot systems, in: *Proceedings of IECON, Tokyo, Japan, 1984a* (in Japanese).
- Arimoto, S., Kawamura, S., and Miyazaki, F.: Bettering operation of robots by learning, *J. Robotic Syst.*, 1, 123–140, <https://doi.org/10.1002/rob.4620010203>, 1984b.
- Delchev, K.: Simulation-based design of monotonically convergent iterative learning control for nonlinear systems, *Archives of Control Sciences*, 22, 371–384, <https://doi.org/10.2478/v10170-011-0036-9>, 2012.
- Eaton, J. W., Bateman, D., Hauberg, S., and Wehbring, R.: GNU Octave version 5.2.0 manual: a high-level interactive language for numerical computations, <https://octave.org/doc/octave-5.2.0.pdf> (last access: 29 April 2022), 2019.
- Heizinger, G., Fenwick, D., Paden, B., and Miyazaki, F.: Robust Learning Control, in: *Proc. of 28th Conference on Decision and Control*, 13–15 December 1989, Tampa, FL, <https://doi.org/10.1109/CDC.1989.70152>, 1989.
- Hu, J., Li, C., Chen, Z., and Yao, B.: Precision Motion Control of a 6-DoFs Industrial Robot With Accurate Payload Estimation, *IEEE-ASME T. Mech.*, 25, 1821–1829, <https://doi.org/10.1109/TMECH.2020.2994231>, 2020.
- Lee, J., Chang, P. H., and Jin, M.: An Adaptive Gain Dynamics for Time Delay Control Improves Accuracy and Robustness to Significant Payload Changes for Robots, *IEEE T. Ind. Electron.*, 67, 3076–3085, <https://doi.org/10.1109/TIE.2019.2912774>, 2020a.
- Lee, J., P. H. Chang, P. H., Yu, B., and Jin, M.: An Adaptive PID Control for Robot Manipulators Under Substantial Payload Variations, *IEEE Access*, 8, 162261–162270, <https://doi.org/10.1109/ACCESS.2020.3014348>, 2020b.
- Longman, R. and Huang, Y.: The Phenomenon of Apparent Convergence Followed by Divergence in Learning and Repetitive Control, *Intell. Autom. Soft Co.*, 8, 107–128, <https://doi.org/10.1080/10798587.2002.10644210>, 2002.
- Michel, O.: Cyberbotics Ltd. Webots™: Professional Mobile Robot Simulation, *Int. J. Adv. Robot. Syst.*, 1, 39–42, <https://doi.org/10.5772/5618>, 2004.
- Nemec, B., Simonič, M., Likar, N., and Ude, A.: Enhancing the performance of adaptive iterative learning control with reinforcement learning, in: *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 24–28 September 2017, Vancouver, BC, 2192–2199, <https://doi.org/10.1109/IROS.2017.8206038>, 2017.
- Patan, K., Patan, M., and Kowalów, D.: Neural networks in design of iterative learning control for nonlinear systems, *IFAC PapersOnLine*, 50, 13402–13407, <https://doi.org/10.1016/j.ifacol.2017.08.2277>, 2017.
- Sekimoto, M., Kawamura, S., Ishitsubo, T., Akizuki, S., and Mizuno, M.: Basis-motion torque composition approach: generation of feedforward inputs for control of multi-joint robots, in: *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 10–15 October 2009, St. Louis, MO, USA, 3127–3132, <https://doi.org/10.1109/IROS.2009.5354729>, 2009.
- Shinji, W. and Mita, T.: A Parameter Identification Method of Horizontal Robot Arms, *Adv. Robotics*, 4, 337–352, <https://doi.org/10.1163/156855390X00170>, 1990.
- Tanimoto, N., Sekimoto, M., Kawamura, S., and Kimura, H.: Generation of feedforward torque by reuse of ILC torque for three-joint robot arm in gravity, in: *2017 56th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE)*, 19–22 September 2017, Kanazawa, Japan, 927–931, <https://doi.org/10.23919/SICE.2017.8105507>, 2017.
- Truong, H. V. A., Tran, D. T., To, X. D., Ahn, K. K., and Jin, M.: Adaptive Fuzzy Backstepping Sliding Mode Control for a 3-DOF Hydraulic Manipulator with Nonlinear Disturbance Observer for Large Payload Variation, *Appl. Sci.*, 9, 3290, <https://doi.org/10.3390/app9163290>, 2019.
- Uchiyama, M.: Formulation of high-speed motion pattern of a mechanical arm by trial, *Trans. SICE (Soc. Instrum. Contr. Eng.)*, 14, 706–712, <https://doi.org/10.9746/sicetr1965.14.706>, 1978.
- Wang, Y. C., Chien, C. J., and Lee, D. T.: A fuzzy neural network direct adaptive iterative learning controller for robotic systems, *IFAC Proceedings Volumes*, 41, 6519–6524, <https://doi.org/10.3182/20080706-5-KR-1001.01099>, 2008.
- Wang, Y., Chien, C., and Chi, R.: A fuzzy neural network based adaptive terminal iterative learning control for nonaffine nonlinear discrete-time systems, in: *Proceedings of the 2014 International Conference on Fuzzy Theory and Its Applications (iFUZZY2014)*, 26–28 November 2014, Kaohsiung, Taiwan, <https://doi.org/10.1109/iFUZZY.2014.7091252>, 163–167, 2014.
- Wei-Liang, K., Cheng, M., and Lin, H.: Reinforcement Q-Learning and ILC with Self-Tuning Learning Rate for Contour Following Accuracy Improvement of Biaxial Motion Stage, *AD Publications*, Rajasthan, India, <http://ijoeer.com/Paper-March-2017/IJOER-MAR-2017-8.pdf> (last access: 29 April 2022), 2017.
- Xu, P., and Xu, Q.: Neural network-based AILC for nonlinear discrete-time system with iteration varying initial error and reference trajectory, in: *Proceedings of the 2018 5th International Conference on Information Science and Control Engineering (ICISCE)*, 20–22 July 2018, Zhengzhou, China, <https://doi.org/10.1109/ICISCE.2018.00179>, 854–858, 2018.
- Yovchev, K. and Miteva, L.: Preliminary Study of Self-adaptive Constrained Output Iterative Learning Control for Robotic Manipulators, in: *Advances in Service and Industrial Robotics. RAAD 2021. Mechanisms and Machine Science*, edited by: Zeghloul, S., Laribi, M. A., and Sandoval, J., Springer, Cham, [https://doi.org/10.1007/978-3-030-75259-0\\_30](https://doi.org/10.1007/978-3-030-75259-0_30), 2021.
- Yovchev, K., Delchev, K., and Krastev, E.: Constrained Output Iterative Learning Control, *Archives of Control Sciences*, 30, 157–176, <https://doi.org/10.24425/acs.2020.132590>, 2020a.
- Zhang, Y., Chu, B., and Shu, Z.: A Preliminary Study on the Relationship Between Iterative Learning Control and Reinforcement Learning, *IFAC PapersOnLine*, 52, 314–319, <https://doi.org/10.1016/j.ifacol.2019.12.669>, 2019.