Dual Design Iterative Learning Controller for Robotic Manipulator Application

Phichitphon Chotikunnan [*] **Benjamas Panomruttanarug** [**]
Poramate Manoonpong [***]

[*] *Department of Control and Instrumentation Engineering, King Mongkut's University of Technology Thonburi, Bangkok 10140, Thailand, (e-mail: phichitphon.c@mail.kmutt.ac.th).*
[**] *Department of Control and Instrumentation Engineering, King Mongkut's University of Technology Thonburi, Bangkok 10140, Thailand, (e-mail: benjamas.pan@kmutt.ac.th)*
[***] *Bio-Inspired Robotics and Neural Engineering Laboratory, School of Information Science and Technology, Vidyasirimedhi Institute of Science and Technology, Rayong 21210, Thailand. Embodied AI and Neurorobotics Laboratory, SDU Biorobotics, The Mærsk Mc-Kinney Møller Institute, University of Southern Denmark, Odense 5230, Denmark, (e-mail: poma@mmmi.sdu.dk)*

Abstract: Iterative learning control enables high precision performance through observed historical data in previous iterations. Several techniques for designing iterative learning controllers have been developed in the existing literature. However, evidence to support the design's efficiency in real applications is, unfortunately, missing in some designs. This paper presents a practical iterative learning controller design, so-called the dual design, combining two existing controller designs using the weighted sum technique. The two controllers are designed using data-driven and frequency response approaches, distinctively selected to take benefits from each. A single gain controller designed from the gain adjustment mechanism usually has slow learning behavior but can be very robust to external uncertainty. The other design imitating the inverse of the frequency response of the system can learn extremely fast. However, its performance may not be as effective as desired when the frequency response of the system is incorrectly perceived. By taking advantage of both controllers, the dual design can achieve fast-learning behavior as well as robustness to external disturbances. Simulation and experiments were carried out to demonstrate the design efficiency.

Keywords: Iterative learning control, gain adjustment mechanism, inverse frequency response, robotic manipulator.

1. INTRODUCTION

Today, robotic manipulators are widely used in industry and hospitals. The industrial application of robot arms includes welding, gluing, and polishing. Most commonly, healthcare robot arms are used for neurorehabilitation. These tasks usually require high precision in all repeated executions. Various techniques are proposed to minimize the tracking error. In general, robot arm developers tend to design both the mechanical and electrical hardware to minimize error. However, other developers focus on the control algorithms to achieve the same purpose without adding extra cost. A Proportional Integral Derivative (PID) control is the most common technique for addressing tracking problems. Many researchers have proposed different techniques to adjust the PID gains. Among them, fuzzy logic control (FLC) is the most popular for tuning the gains. Li et al. (2020) proposed the gain tuning technique using a fuzzy neural network algorithm. A similar concept, proposed by Elmogy et al. (2020), uses an adaptive fuzzy logic controller. Other control techniques for PID tuning include optimal control Long et al. (2021), adaptive sliding mode Ejaz et al. (2019), and adaptive neural network tracking control Yang et al. (2018).

Rather than attempting to tune the PID gains, iterative learning control (ILC) is an alternative technique for improving the tracking error without adjusting the internal parameters inside the feedback control system. Based on the observed error and control input signal from the previous iteration, ILC adjusts the control input signal to the feedback control system with the aim of reducing tracking error. However, a bad learning transient can easily be observed in practical ILC implementation. Robust ILC is therefore proposed to handle the problem.

In general, there are three main categories in the design of a robust ILC: a Q-filter design, an optimization-based design, and a robust learning gain design. In the first category, the Q-filter is formulated based on a lowpass filter Tomizuka (1987). It is basically constructed using an infinite impulse response (IIR), where its cut-off frequency is determined from the spectrum of the bad learning transient. Hock and Schoellig (2019) Demonstrates the use of a Q-filter together with a D-type ILC to eliminate high-frequency disturbance in robotic applications. Huang et al. (2019) Implements a Q-filter in an active disturbance rejection control and constructs a current-cycle iterative learning control to eliminate nonlinearities and dynamic uncertainties.

Bristow et al. (2007) Develops a linear time-varying (LTV) Q-filter for ILC that incorporates time-frequency analysis. Feng et al. (2017) Combines a wavelet transform technique with a linear time-varying Q-filter to remove high-frequency errors. Lin et al. (2015) Presents another design of nonparametric Q-filter that does not require any explicit specification of the non-repetitive disturbances.

The second class of robust ILC is designed based on the optimization technique. This class is more general and extensively developed in many aspects. The original technique is known as the norm-optimal ILC Amann et al. (1996), where the optimal criterion is a trade-off between the minimization of the tracking error and input update. The cost function can be developed to include the input effort Volckaert et al. (2013) or the error rate Zhu et al. (2020). Recent implementation of the norm-optimal ILC has been presented in Johansen et al. (2019), Allahverdy et al. (2021) and Jonnalagadda and Elumalai (2021) to improve non-repetitive trajectory tracking performance.

In the optimization-based robust ILC, another type of cost function has been proposed in the frequency domain. In considering the error propagation from one iteration to the next one in the frequency domain, the cost function is minimized by assuming that all transients occur in a relatively short period of time. The frequency domain behavior of the controller produces a similar result as the system inverse, except that it does not include an unstable part like the system inverse may have Panomruttanarug and Longman (2006). The effectiveness of the design is experimentally demonstrated in Panomruttanarug (2020).

In the final robust ILC category, the learning control matrix is designed to achieve robustness without the assistance of an additional filter. Various techniques have been proposed to design and adjust the gains in the learning control matrix to robustify the ILC system. Zheng et al. (2017) proposed a robust learning filter using H_∞ optimal control, while Panomruttanarug and Longman (2009) directly designs a robust learning filter based on the frequency response data. Chotikunnan and Panomruttanarug (2022) propose a gain adjustment mechanism to adjust the learning gains based on FLC.

Since each of the three methods has pros and cons, it is of interest to investigate the potential of combining a few of them together and utilizing their advantages. Even though the robust learning control matrix can assure system robustness, it usually causes non-zero tracking error due to neglect of the high-frequency components. In contrast, the frequency domain optimization-based robust ILC can provide a fast-learning rate with a substantially low level of final tracking error. However, it can produce a bad learning transient before reaching the converging stage, making it impractical when implemented in a real experiment.

One needs to trade-off between tracking performance and the robustness of the system. Inspired by Thor and Manoonpong (2019), a combining technique using dual integral learners is considered. By fusing a fast learner with a slow learner, the dual combining concept can produce a better result than using either one individually. Similar combining techniques for dual controllers in ILC are proposed in Li et al. (2021), Lai et al. (2021), He et al. (2017) and Zhu and Xu (2018). In Li et al. (2021), a Plug-in ILC design is constructed in parallel with a feedforward signal. Whereas Lai et al. (2021) proposes a dual-loop ILC design for repetitive human-robot interaction. The first loop generates the desired joint angle trajectory for the robot to follow. The second loop is designed using an adaptive iterative learning algorithm to iteratively follow the desired path. A dual-loop ILC design for a Timoshenko beam system is proposed in He et al. (2017), where a typical ILC loop is embedded into the adaptive boundary control loop to cancel the effect of external disturbance, input backlash, and output constraint. In Zhu and Xu (2018), a dual internal model-based ILC design is proposed consisting of two loops. Using high order ILC laws, the first loop learns the control input from previous iterations, while the second loop learns the control input from previous times at the current iteration. The results from all these works demonstrate the effectiveness of using a dual design ILC.

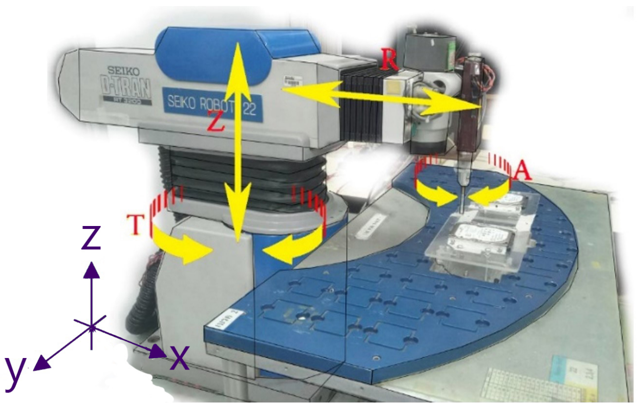

Fig. 1. Seiko D-Tran RT3200 and the interface equipment.

This work benefits from the use of a dual design ILC to combine the authors' previous controller designs. The robust learning control matrix, proposed in Chotikunnan and Panomruttanarug (2022), is considered to be a slow learner, while the frequency domain optimization-based robust ILC, proposed in Panomruttanarug (2020), is treated as a fast learner. A weighting function is also investigated to analyze the behaviors when placing unequal weight on both learners. To verify the effectiveness of the dual design ILC, both simulation and experiments are conducted based on a real robotic manipulator.

2. TRACKING PROBLEM IN ROBOTIC MANIPULATORS

This section demonstrates the tracking problem usually appearing in robotic manipulators. The cylindrical robot, Seiko D-Tran RT3200, shown in Fig. 1, is used as a testbed to perform a tracking task. It has been modified to be programmable using the LabVIEW program. The real-time controller, cRIO-9075, from National Instruments, is used to compute and execute the task.

2.1 Seiko D-Tran RT3200 and the interface equipment

The robotic arm consists of four movement joints: joint R, sliding in and out of the X-axis plane, joints T and A as the rotating angle pivot in the X-Y plane, and joint Z, respectively, for lifting and lowering points of the robotic arm.

The robot can be commanded to perform a screw fastening task by moving and rotating joints R, T, and Z simultaneously. However, joint A is not being used to execute the task and therefore omitted from the following analysis. The desired movement versus the actual movement of all joints for 15.73s is shown in Figs. 2 and 3. With a sampling time of 0.055 s,

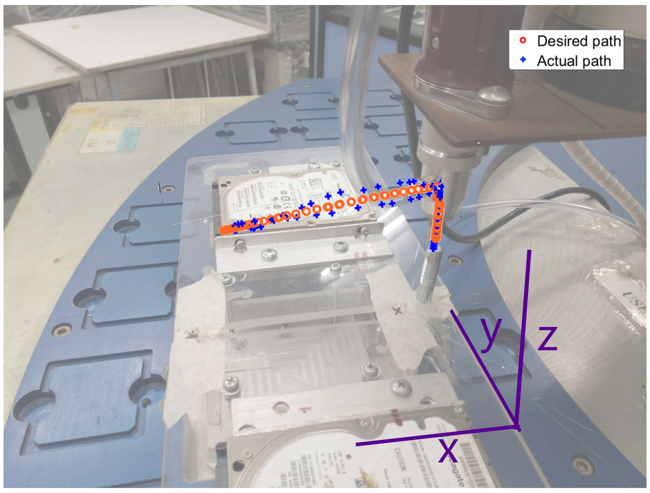

Fig. 2. Desired path versus actual path movement in 3D.

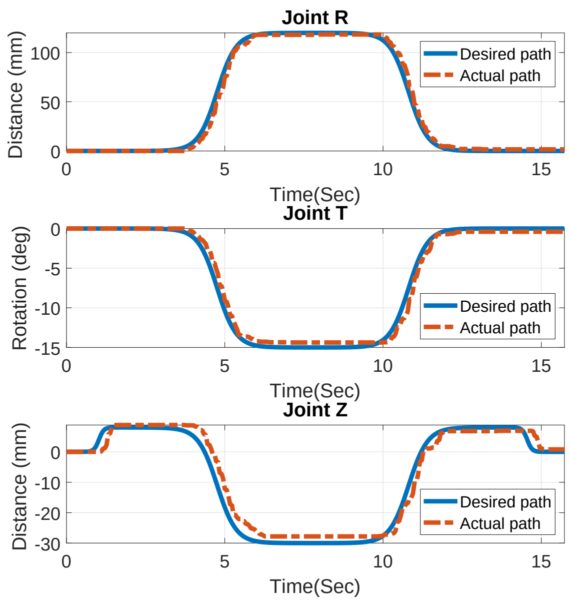

Fig. 3. Desired path versus actual path movement in each joint.

there are 287 time steps in the paths. In Fig. 2, the 3D path is created by moving the screw from one location to another and pushing into a hole. The real performance using a feedback control system is also illustrated to interpret the tracking error. Fig. 3 demonstrates the desired trajectory versus the real path in each joint. One can clearly see the tracking error along the path, especially in joint Z. In the next section, iterative learning control is introduced to eliminate the tracking error in all joints.

2.2 *System models of the robotic manipulator*

According to the input and output signals illustrated in Fig. 3, one can approximate the three feedback control systems in terms of discrete-time transfer functions as follows:

$$G(z) = \frac{\gamma_1 z}{z^2 + \beta_1 z + \beta_0} \tag{1}$$

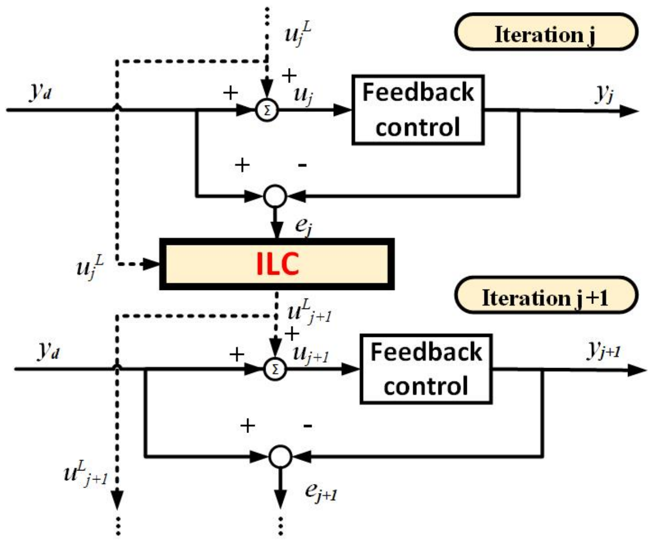

Fig. 4. Block diagram of ILC.

where $G(z)$ represents a general form of the second-order transfer function for all joints. The coefficients, γ_1, β_1, β_0, associated with each joint appear in Table 1.

Table 1. Parameters used in the system models

Joint	γ_1	β_1	β_0
Joint R	0.1413	-1.5458	0.6884
Joint T	0.1297	-1.5780	0.7111
Joint Z	0.0935	-1.6584	0.7526

3. ITERATIVE LEARNING CONTROLLER DESIGNS

In this section, ILC is constructed to improve the tracking performance in each feedback control system. Fig. 4 presents a block diagram of ILC added in between two successive iterations, i.e., j and $j+1$. The ILC mechanism updates the learning control input in iteration $j+1$, or u_{j+1}^L, based on the learning control input and the error from the previous iteration by aiming for the output, y, to get closer and closer to the desired trajectory, y_d, when the iteration progresses. In other words, the objective of using ILC is to eliminate the tracking error, $e = y_d - y$, in each iteration.

The feedback control system at iteration j in the block diagram refers to the transfer function in (1), which can be rewritten in the state space representation as

$$\begin{aligned} x_j(k+1) &= \boldsymbol{A}x_j(k) + \boldsymbol{B}u_j(k) \\ y_j(k) &= \boldsymbol{C}x_j(k) + \boldsymbol{D}u_j(k) \end{aligned} \tag{2}$$

where $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}$, and $\boldsymbol{D}$ are the Markov parameters. For simplicity, let us suppose that $\boldsymbol{D} = 0$ for simplicity. The subscript j denotes the iteration number j. One can formulate a relationship between the input sequence $\boldsymbol{u}_j$ and the output sequence $\boldsymbol{y}_j$ in a package form as

$$\underbrace{\begin{bmatrix} y_j(1) \\ \vdots \\ y_j(N) \end{bmatrix}}_{\boldsymbol{y}_j} = \underbrace{\begin{bmatrix} p_1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ p_N & \dots & p_1 \end{bmatrix}}_{\mathbf{P}} \underbrace{\begin{bmatrix} u_j(0) \\ \vdots \\ u_j(N-1) \end{bmatrix}}_{\boldsymbol{u}_j} + \underbrace{\begin{bmatrix} q_1 \\ \vdots \\ q_N \end{bmatrix}}_{\mathbf{Q}} x_j(0) \tag{3}$$

where $p_i = \boldsymbol{CA}^{i-1}\boldsymbol{B}$ and $q_i = \boldsymbol{CA}^i$, for $i \in [1, N]$. N and $x_j(0)$ denote the total number of time steps in an iteration and the initial condition, respectively. For simplicity, let us assume the initial states are zero. This results in $\boldsymbol{y}_j = \boldsymbol{P}\boldsymbol{u}_j$ where the matrix $\boldsymbol{P}$ represents the feedback control system.

Referring to the block diagram in Fig. 4, the control input to the feedback control system, $\boldsymbol{u}_j$, is a summation of the learning control input $\boldsymbol{u}_j^L$ and the desired trajectory $\boldsymbol{y}_d$. At the initial trial, the learning algorithm has yet to activate. The input to the feedback control system comes directly from the desired trajectory, resulting in $\boldsymbol{u}_j = \boldsymbol{y}_d$. In later iterations when ILC is activated, $\boldsymbol{u}_j = \boldsymbol{y}_d + \boldsymbol{u}_j^L$, its input is adjusted based on the following ILC law:

$$u_{j+1}^L(k) = u_j^L(k) + \boldsymbol{L}e_j(k+1) \tag{4}$$

where $\boldsymbol{L}$ is the learning gain matrix. Two existing learning gain matrix designs are discussed in the following subsections: a single gain design based on the GAM, and a multiple gain design based on the IFR, as well as the proposed design that integrates the advantages of both controllers.

3.1 A single gain design based on the GAM

In this controller design, only the gains along the main diagonal in (5) are non-zero and set to a constant at the first learning iteration. The learning control matrix can be expressed as

$$\boldsymbol{L} = \begin{bmatrix} l & 0 & \cdots & 0 \\ 0 & l & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & l \end{bmatrix} \tag{5}$$

This leads to the following learning control law with a single gain design:

$$u_{j+1}^L(k) = u_j^L(k) + le_j(k+1) \tag{6}$$

It can obviously be observed that gain l is associated with the error at time step $k+1$. The gains are adjusted all over the main diagonal in the following trial using the gain adjustment mechanism (GAM). The values of gains are slowly changed in early learning iterations. However, they remain constant after passing through some iterations.

In the GAM process, the gains are adjusted so that the output can follow the desired trajectory as well as the desired learning control input. The desired learning control input, or $\boldsymbol{u}_d^L$, is generated based on the system's behavior. Starting with the execution of ILC law in (6) and the learning control matrix in (5), one can observe error decay during the early stage, followed by error growth due to model mismatch or the propagation of noise and non-repetitive disturbances. The desired learning control input can be obtained from its input signal in the last iteration before observing the error growth. The error due to the mismatch of the learning control input signals can be calculated as

$$e_j^u(k) = \begin{cases} 0 & , for \quad \|\boldsymbol{u}_d^L - \boldsymbol{u}_j^L\|_\infty = 0 \\ \left| \dfrac{u_d^L(k) - u_j^L(k)}{\|\boldsymbol{u}_d^L - \boldsymbol{u}_j^L\|_\infty} \right| & , otherwise \end{cases} \tag{7}$$

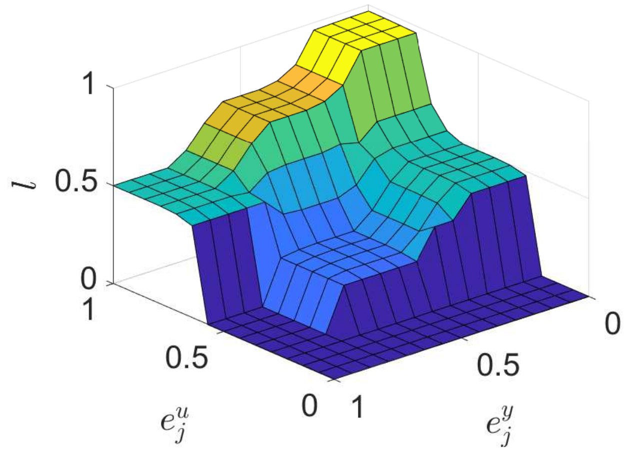

Fig. 5. Surface plot for GAM.

While the tracking error can be computed by

$$e_j^y(k) = \begin{cases} 0 & , for \quad \|\boldsymbol{y}_d - \boldsymbol{y}_j\|_\infty = 0 \\ \left| \dfrac{y_d^L(k) - y_j^L(k)}{\|\boldsymbol{y}_d - \boldsymbol{y}_j\|_\infty} \right| & , otherwise \end{cases} \tag{8}$$

The two errors are subsequently considered as the inputs of a fuzzy logic control with the output as the learning gain. The relationship between the two inputs and the output can be illustrated by the surface plot in Fig. 5.

As evidenced by Chotikunnan and Panomruttanarug (2022), even though the GAM can track the desired trajectory with an acceptable final error level, the learning rate is rather slow due to the simplicity of the learning control matrix, which contains only the gains along the main diagonal. A compensator design that fills the learning control matrix with a set of gains is discussed in the following subsection.

3.2 A multiple gain design based on the IFR

This controller design makes use of multiple gains to form a banded matrix with the same entries as

$$\boldsymbol{L} = \begin{bmatrix} l_m & \cdots & l_n & 0 & \cdots & 0 \\ \vdots & l_m & \ddots & \ddots & \ddots & \vdots \\ l_1 & 0 & \ddots & 0 & \ddots & 0 \\ 0 & l_1 & \ddots & \ddots & \ddots & l_n \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & l_1 & \cdots & l_m \end{bmatrix} \tag{9}$$

The notation m indicates the index associated with the error at time step $k+1$, whereas the notation n represents the total number of gains used in the controller design.

To explicitly express the gains in the learning control law, one can write

$$\begin{aligned} u_{j+1}^L(k) = {} & u_j^L(k) + l_1 e_j(k+m-1) + l_2 e_j(k+m-2) \\ & + \cdots + l_n e_j(k+m-n) \end{aligned} \tag{10}$$

The controller can be considered as a finite impulse response (FIR) filter with the following discrete-time transfer function

$$F(z) = l_1 z^{m-1} + l_2 z^{m-2} + \ldots + l_n z^{m-n} \tag{11}$$

The frequency response of the FIR compensator can be expressed as

$$F\left(e^{i\omega}\right) = l_1 e^{i\omega(m-1)} + l_2 e^{i\omega(m-2)} + \ldots + l_n e^{i\omega(m-n)} \tag{12}$$

Similarly, the frequency response of the feedback control system, $G(z)$, can be written in terms of magnitude and phase as

$$G\left(e^{i\omega}\right) = M(\omega)\, e^{i\phi(\omega)} \tag{13}$$

where $M(\omega)$ and $\phi(\omega)$ represent the magnitude response and the phase response of the system, respectively.

Based on the inverse frequency response design in Panomruttanarug (2020), one needs to find the optimal gains to minimize the cost function

$$J = \sqrt{\sum_{j=1}^{NP}\left[1 - F(e^{i\omega_j T})G(e^{i\omega_j T})\right]\left[1 - F\left(e^{i\omega_j T}\right)G\left(e^{i\omega_j T}\right)\right]^* + \sum_{k=1}^{n} l_k^2} \tag{14}$$

where ω_j represents the sample frequency from zero to Nyquist with a total of NP points. It can be observed that the multi-objective cost function is composed of the two terms: the first term represents the effect of error from one iteration to the next, while the second expresses the size of the gains used in the FIR compensator. Minimizing only the first term can easily result in gains that are too large to be implemented in the real application. The second term is therefore needed for a practical design.

To find the optimal gains associated with the cost function in (14), one can equivalently solve the linear equation, $\boldsymbol{\alpha x} = \boldsymbol{\beta}$, where

$$\boldsymbol{\alpha} = \sum_{j=1}^{NP} M^2\left(\omega_j\right) \begin{bmatrix} 1 & \cos\left(\omega_j T\right) & \ldots & \cos\left((n-1)\,\omega_j T\right) \\ \cos\left(\omega_j T\right) & 1 & \ldots & \cos\left((n-2)\,\omega_j T\right) \\ \vdots & \vdots & \ddots & \vdots \\ \cos\left((n-1)\,\omega_j T\right) & \cos\left((n-2)\,\omega_j T\right) & \ldots & 1 \end{bmatrix} + NP \times \boldsymbol{I}_{n\times n} \tag{15}$$

$$\boldsymbol{\beta} = \sum_{j=1}^{NP} M(\omega_j) \begin{bmatrix} \cos((m-1)\omega_j T) + \phi(\omega_j) \\ \cos((m-2)\omega_j T) + \phi(\omega_j) \\ \vdots \\ \cos((m-n)\omega_j T) + \phi(\omega_j) \end{bmatrix} \tag{16}$$

$$\boldsymbol{x} = \begin{bmatrix} l_1 & l_2 & \cdots & l_n \end{bmatrix}^T$$

The compensator design based on the IFR provides a fast-learning rate due to the set of gains used in the design. However, it requires the magnitude response and phase response of the system to achieve the optimal gains. The efficiency of this controller design depends directly on the accuracy of the frequency response obtained from the system.

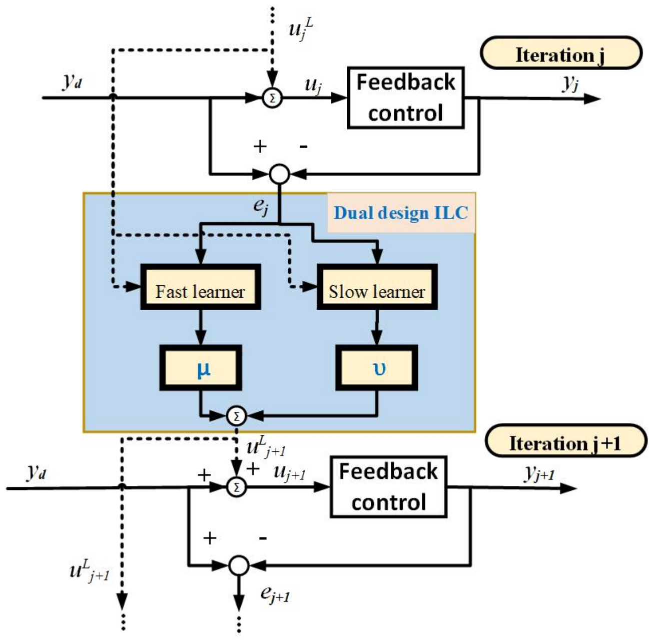

Fig. 6. Block diagram of dual design ILC.

3.3 A dual design

This subsection introduces a dual design that integrates the prior designs using the weighted sum technique. Its structure is illustrated in Fig. 6.

The fast and slow learners in Fig. 6 refer to the multi-gain design based on the IFR and the single gain design based on the GAM, respectively. The learning control input and error from the last iteration are fed to the two learners in the next iteration. The outputs from the two learners are combined using the weighted sum of each individual learner's output. The weight μ belongs to the fast learner, while the weight ν belongs to the slow learner. In general, the sum of the weights is unity, i.e., $\mu + \nu = 1$. One can explicitly write the learning control input from the dual design as

$$u^L_{j+1}(k) = \mu u^{L_f}_{j+1}(k) + \nu u^{L_s}_{j+1}(k) \tag{17}$$

where $\boldsymbol{u}^{L_f}_{j+1}$ and $\boldsymbol{u}^{L_s}_{j+1}$ can be calculated from (10) and (6), respectively. Using the weighted sum of the outputs from the two learners allows designers to tune the weights to achieve better tracking performance. It combines the advantages of the two learners by providing a fast-learning rate without relying too much on the frequency response.

Based on the linearity property in (17), it can be modified by expressing the dual design controller in terms of the learning matrix as

$$\boldsymbol{L}_d = \mu \boldsymbol{L}_f + \nu \boldsymbol{L}_s \tag{18}$$

where $\boldsymbol{L}_d$ is the learning matrix of the dual design developed from the IFR in (9) and the GAM in (5).

The choice of weights μ and ν can be either constant or a function with the summation of one. For fixed weight, one may consider putting more weight on the slow learner to maintain stability while the fast learner has little influence. Therefore, μ is set to 0.35, resulting in $\nu = 0.65$.

Alternatively, the weighting function can be used, where the weight of the fast learner can be expressed as

$$\mu(j) = \begin{cases} 1 - \dfrac{0.65}{1+e^{(12.2-(2.2j))}} & , for \quad j \le 10 \\ 0.35 & , otherwise \end{cases} \tag{19}$$

where j denotes the iteration number. The weighting function is a sigmoid function gradually decreasing from one at the first iteration to 0.35 at iteration 10. The weight is fixed to a constant 0.35 after iteration 10. This technique takes advantage of the fast learner in early iterations. However, the weight is gradually decreased to slow down the learning process after some iterations. The weight of the slow learner is associated with $\nu(j) = 1 - \mu(j),\ \forall j$.

4. STABILITY ANALYSIS

It has been widely proven that the stability of ILC systems can be considered using asymptotic and monotonic convergence conditions. Both conditions are proven from the expression of the error from one iteration to the next, which can be written as

$$\boldsymbol{e}_{j+1} = (\boldsymbol{I} - \boldsymbol{PL})\boldsymbol{e}_j \tag{20}$$

According to (21) and (22), the asymptotic convergence condition can be expressed as

$$\max_i |\lambda_i(\boldsymbol{I} - \boldsymbol{PL})| < 1 \tag{21}$$

where λ is the eigenvalue of the matrix. Similarly, the monotonic convergence condition can be written as

$$\max_i \ \sigma_i(\boldsymbol{I} - \boldsymbol{PL}) < 1 \tag{22}$$

where σ represents the singular value of the matrix.

Table 2. Stability indicators

Type	Joint R		Joint T		Joint Z	
	(21)	(22)	(21)	(22)	(21)	(22)
1 (GAM)	0.9750	1.0191	0.9771	1.0235	0.9835	1.0324
2 (IFR-small)	0.9979	0.9988	0.9984	0.9993	0.9990	0.9994
3 (IFR-large)	0.9974	0.9975	0.9978	0.9979	0.9988	0.9989
4 (Dual-small, fixed)	0.9979	0.9988	0.9984	0.9993	0.9993	0.9998
5 (Dual-large, fixed)	0.9974	0.9975	0.9978	0.9979	0.9988	0.9989
6 (Dual-small, function)	0.9979	0.9988	0.9984	0.9993	0.9993	0.9998
7 (Dual-large, function)	0.9979	0.9974	0.9975	0.9978	0.9988	0.9989

To study and evaluate the stability of the aforementioned controllers, three classes of ILC are designed to improve the tracking error from the three axes of the robotic manipulator. Seven controllers are designed based on the GAM, IFR, and dual design as previously presented. Table 2 demonstrates the stability indicators, or the left term in (21) and (22), using each type of controller. To comply with the two conditions, the indicator values must be less than one.

Controller 1 represents a slow learner using the single gain design based on the GAM. The learning gain tuned by the GAM is varied from 0 to 0.1769 in joint R, 0 to 0.1928 in joint T, and 0 to 0.2674 in joint Z. The indicator values shown in Table 2 are subject to the worst cases when each learning matrix is filled with the maximum gains along the main diagonal. It can be observed that the values for the condition in (21) are all less than one, implying that the error will eventually converge to a certain value. In contrast to the asymptotic convergence condition, the indicator values for the condition in (22) are slightly greater than one, violating the monotonic convergence condition. This is not an issue when using the GAM-based controller since the gains are iteratively tuned to prevent transient growth.

Controllers 2 and 3, where the multiple gain design is based on the IFR, are fast learners. The minimum number of gains for stabilizing ILC systems are used in controller 2. Joints R and T require at least six gains, while joint Z needs at least eight to achieve stability. For comparison of robustness, many gains are used in controller 3 with $n = 30$ in each joint. For simplicity, the parameter m is set to be equal to n in all these types of controllers. According to Table 2, all indicator values for controllers 2 and 3 are less than one. Due to a greater number of gains being used in the controller, the indicator values for controller 3 are lower in all systems when compared with those for controller 2. This implies that controller 3 is more robust than controller 2.

For the dual design, four different controllers are used to study the effectiveness of the proposed method. Controllers 4 and 5 combine the slow learner from controller 1 with the fast learners from controllers 2 and 3, respectively. The fixed weights of $\mu = 0.35$ and $\nu = 0.65$ are applied to these types of controllers. Similarly, controllers 6 and 7 represent a combination of the slow learner from controller 1 and the fast learners from controllers 2 and 3 using the weighting function in (19). As can be observed, the indicator values from the dual design are more or less equal to those from its IFR-based design, no matter what fixed weight or weighting function is used. One might wonder if their tracking behaviors are similar. The tracking performance using all addressed controllers is demonstrated in the following section.

5. SIMULATION AND EXPERIMENTAL RESULTS

Seven controllers have been designed and evaluated to ensure stability. This section shows their tracking performance in both simulation and experiments.

To provide a clear demonstration, let us compare the addressed controllers by separating them into two situations. The first situation demonstrates what happens to the tracking performance when using a minimum number of gains in both IFR-based controllers and dual design controllers. Therefore, controllers 1, 2, 4, and 6 are compared in the first case. The latter compares tracking performance when using many gains in the IFR-based controllers as well as dual design controllers. Controllers 1, 3, 5, and 7 are included in the comparison.

Fig. 7 displays the tracking performance from the first situation when using a minimum number of gains in both IFR-based controllers and dual design controllers in the simulation. Each subfigure shows the root mean square error (RMSE) occurring in each joint by executing controllers 1, 2, 4, and 6 up to 20,000 iterations. One can clearly see that the IFR-based controller

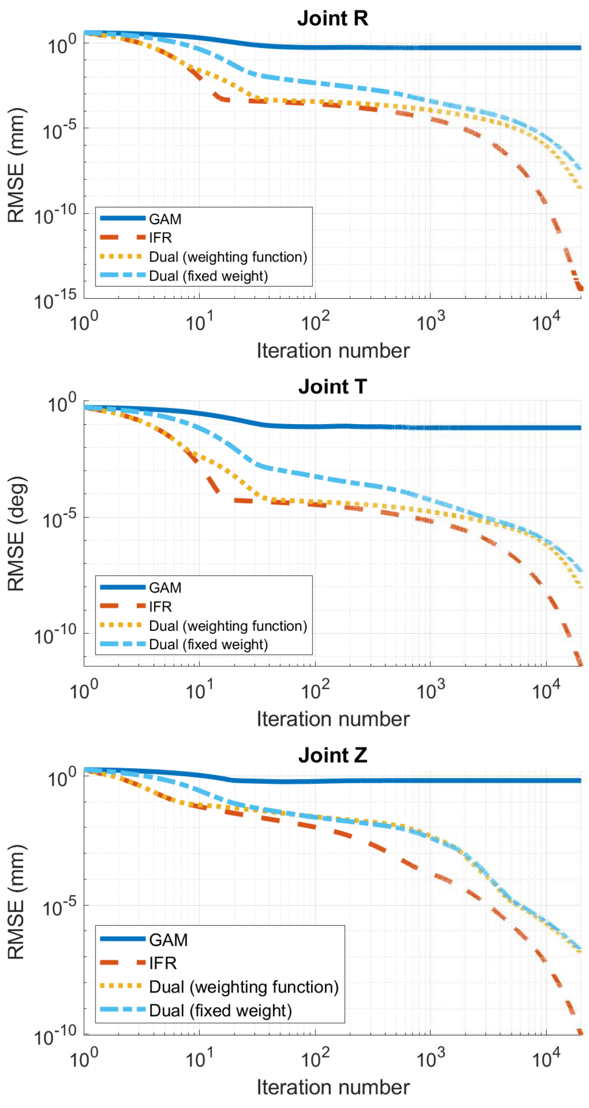

Fig. 7. RMSE from simulation using controllers 1, 2, 4, and 6.

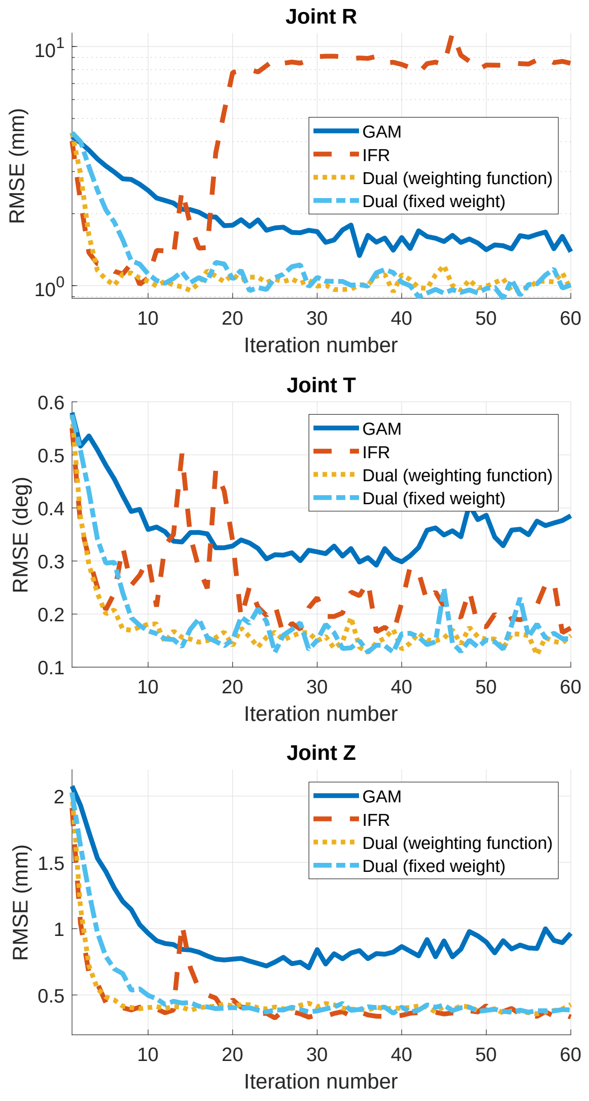

Fig. 8. RMSE from experiment using controllers 1, 2, 4, and 6.

achieves the best tracking performance, followed by the dual design controller and the GAM-based controller, respectively. The tracking performance of the dual design controllers falls in between the other two controllers. As previously discussed, the tracking behavior from the dual design controller with the weighting function appears to be exactly the same as that exhibited by the IFR-based controller in some early iterations. In later iterations, it slowly converges to the behavior of the dual design controller with a fixed weight.

Fig. 8 demonstrates the tracking performance using a minimum number of gains in both the IFR-based controllers and dual design controllers from the experiment. Up to 60 iterations were executed in each joint. The tracking performance from the IFR-based controller does not perform as well as in the simulation. Even though the IFR-based controller can learn faster than other controllers at the beginning, it somehow generates error growth in some later iterations. In joint R, the tracking error continuously grows from iterations 10 to 20 and remains constant after that, resulting in a larger tracking error than that of the pure feedback controller. However, the IFR-based controller can eventually get rid of the transient growth in joints T and Z. The GAM-based controller performs similarly to the simulation results. It iteratively lowers the error from one run to the next with a slow learning rate. The final error level is larger than that of the other controllers in joints T and Z. Interestingly, the dual design controllers practically perform best in all systems. They can learn much faster than the GAM-based controller without any appearance of error growth. The design with weighting function can even learn as fast as the IFR-based controller with no evidence of error growth.

To analyze the final error level of RMSE, the error reduction rates at iteration 60 are computed for comparison with its initial iteration. The GAM-based controller can reduce the tracking error by 67.09% in joint R, 33.55% in joint T, and 53.45% in joint Z. The IFR-based controller can reduce the tracking error by -111.32% in joint R, 68.44% in joint T, and 82.56%

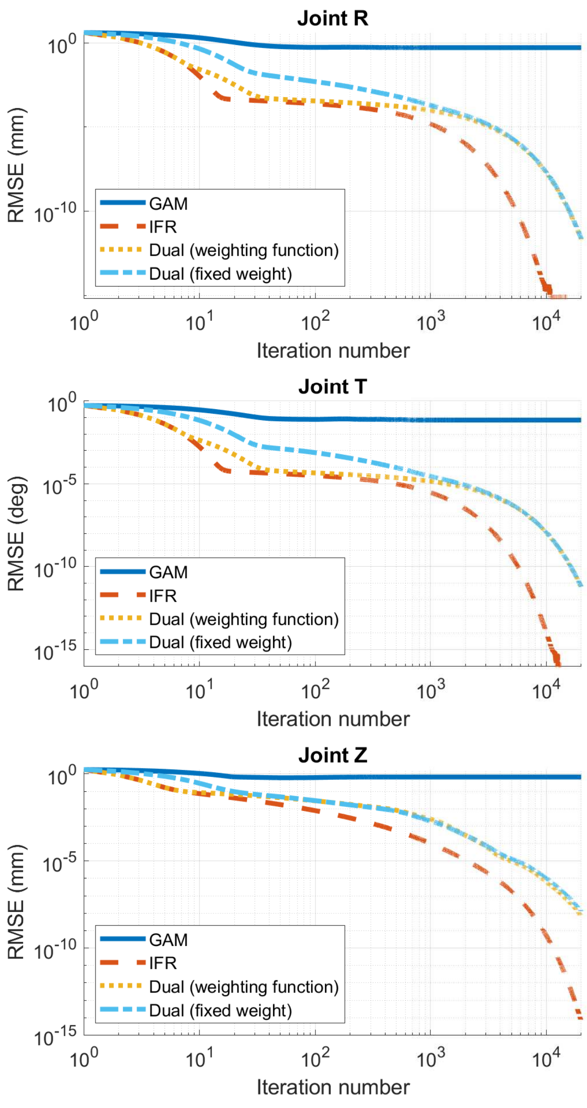

Fig. 9. RMSE from simulation using controllers 1, 3, 5, and 7.

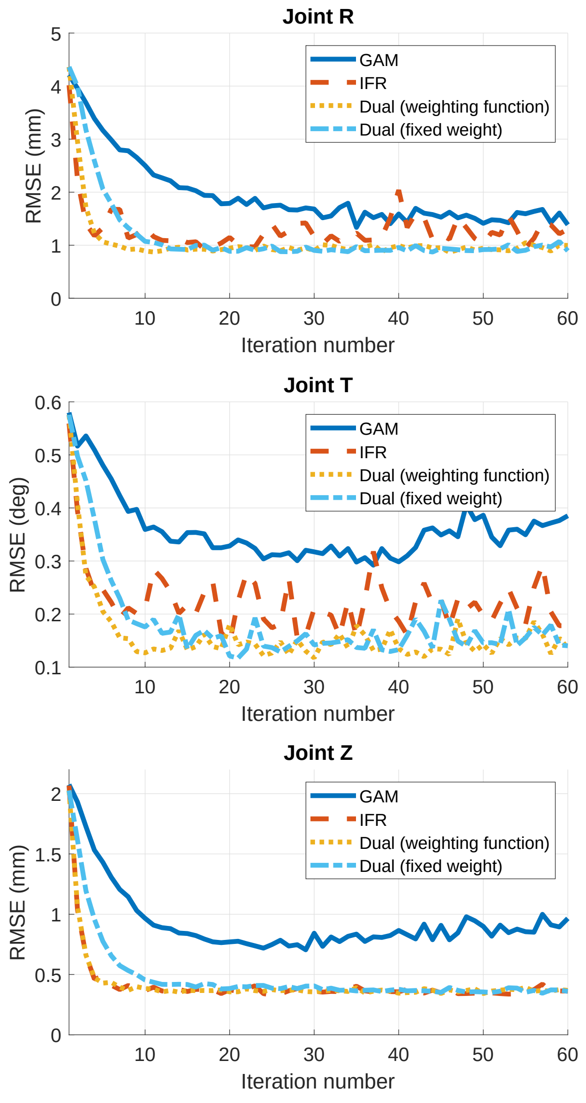

Fig. 10. RMSE from experiment using controllers 1, 3, 5, and 7.

in joint Z. It should be noted that the minus sign determines the amplification of the error. Meanwhile, the dual design controllers with the fixed weight and weighting function can perform similarly, with a reduction rate of 76.95% and 77.99% in joint R, 73.39% and 72.28% in joint T, and 80.87% and 78.84% in joint Z, respectively.

In the next situation, many gains are used in the IFR-based controllers as well as dual design controllers to compare their tracking efficiency. Controllers 1, 3, 5, and 7 were executed in both simulation and the experiment. Fig. 9 compares the tracking performance in each joint from the simulation. The results look similar to those in Fig. 7 where the minimum number of gains is used in the controller design. According to Fig. 9, the IFR-based controller can still achieve the best performance while the GAM-based performs slowest but is still able to maintain stability. The dual design controller with the weighting function can learn as fast as the IFR-based controller in up to 10 iterations. However, its behavior slowly converges to that of the dual design controller with the fixed weight in later iterations.

Fig. 10 illustrates the tracking performance when executing many gain controller designs in the experiment. By using many gains in the IFR-based controller, one cannot obviously see the error growth in its tracking result. In other words, the controller is more robust when applying many gains in the design. Both dual design controllers perform similar tracking results with lower final RMSE values in comparison to the IFR-based controller. The error reduction rates at iteration 60 are computed to compare with its initial iteration. The IFR-based controller can reduce the tracking error by 67.60% in joint R, 68.13% in joint T, and 82.49% in joint Z. Meanwhile, the dual design controllers with the fixed weight and weighting function can perform similarly, with reduction rates of 79.52% and 77.14% in joint R, 75.65% and 76.42% in joint T, and 82.23% and 81.83% in joint Z, respectively. For a clear illustration, Table 3 summarizes the percentage of improvement in RMSE exhibited by each controller. It can be clearly observed that not only can the dual design controllers be robust to error growth, but also achieve the lowest RMSE level with a small number of gains used in the designs.

Table 3. Percentage of improvement in RMSE exhibited by each controller design

Type	Joint R (%)	Joint T (%)	Joint Z (%)
1 (GAM)	67.09	33.55	53.49
2 (IFR-small)	-111.32	68.44	**82.56**
3 (IFR-large)	67.60	68.13	82.49
4 (Dual-small, fixed)	76.95	73.39	80.87
5 (Dual-large, fixed)	**79.52**	75.65	82.23
6 (Dual-small, function)	77.99	72.28	78.84
7 (Dual-large, function)	77.14	**76.42**	81.83

6. CONCLUSION

This paper proposes a dual design for an iterative learning controller based on fast and slow learners. The fast learner is designed based on the inverse of the frequency response from the system that can learn very fast in early iterations. However, by taking advantage of fast learning, the design might achieve the desired robustness in practice, especially when using a small number of gains. The slow learner designed from the gain adjustment mechanism is therefore introduced and combined to increase robustness from the fast learner. The dual design can adaptively choose to learn as quickly as the fast learner up to the desired iteration and slowly learn in later iterations to maintain stability, thus imitating the slow learner. Two different learning weights are proposed and demonstrated to adjust the learning behavior, namely fixed weight and weighting function. One can clearly see that with the use of the weighting function, the dual design can learn as quickly as the fast learner. Based on the results, even though the fast learner achieves the best tracking performance in the simulation, robustness evidently becomes an issue in the experiment, especially when using a minimum number of gains. This issue can be solved by the dual design, which is robust to error growth and efficiently reduces the error to a decent level. An extension of the existing results includes some other choices for the two learners that can practically provide fast learning with low computational complexity.

ACKNOWLEDGEMENTS

This work was supported by Petchra Pra Jom Klao, Ph.D. Research Scholarship (Grant No. 1/2561) from King Mongkut's University of Technology Thonburi.

REFERENCES

Allahverdy, D., Fakharian, A. and Menhaj, M. B. (2021). Application of PID and Norm Optimal Iterative Learning Control to Swash Mass Helicopter. *In 2021 7th International Conference on Control, Instrumentation and Automation (ICCIA)*, 1–6, Doi:10.1109/ICCIA52082.2021.9403592.

Amann, N., Owens, D. H. and Rogers, E. (1996). Iterative learning control using optimal feedback and feedforward actions. *International Journal of Control*, 65(2), 277–293, Doi:10.1080/00207179608921697

Bristow, D. A., Alleyne, A. and Tharayil, M. (2007) A time-varying q-filter design for iterative learning control. *In 2007 American Control Conference*, 5503–5508, Doi:10.1109/ACC.2007.4282553

Chotikunnan, P. and Panomruttanarug, B. (2022). Practical design of a time-varying iterative learning control law using fuzzy logic. *Journal of Intelligent & Fuzzy Systems*, (Preprint), 1–16, Doi:10.3233/JIFS-213082.

Ejaz, F., Hamayun, M. T., Hussain, S., Ijaz, S., Yang, S., Shehzad, N. and Rashid, A. (2019). An adaptive sliding mode actuator fault tolerant control scheme for octorotor system. *International Journal of Advanced Robotic Systems*, 16(2), Doi:10.1177/1729881419832435.

Elmogy, A., Bouteraa, Y. and Elawady, W. (2020). An adaptive fuzzy self-tuning inverse kinematics approach for robot manipulators. *Journal of Control Engineering and Applied Informatics*, 22(4), 43–51.

Feng, Z., Ling, J., Ming, M. and Xiao, X. H. (2017). High-bandwidth and flexible tracking control for precision motion with application to a piezo nanopositioner. *Review of Scientific Instruments*, 88(8), Doi:10.1063/1.4998303.

He, W., Meng, T., Zhang, S., Liu, J. K., Li, G. and Sun, C. (2017). Dual-loop adaptive iterative learning control for a Timoshenko beam with output constraint and input backlash. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* , 49(5), 1027–1038, Doi:10.1109/TSMC.2017.2692529.

Huang, D., Min, D., Jian, Y. and Li, Y. (2019). Current-cycle iterative learning control for high-precision position tracking of piezoelectric actuator system via active disturbance rejection control for hysteresis compensation. *IEEE Transactions on Industrial Electronics*, 67(10), 8680–8690, Doi:10.1109/TIE.2019.2946554.

Hock, A. and Schoellig, A. P. (2019). Distributed iterative learning control for multi-agent systems. *Autonomous Robots* , 43(8), 1989–2010, Doi:10.1007/s10514-019-09845-4

Johansen, S. V., Jensen, M. R., Chu, B., Bendtsen, J. D., Mogensen, J. and Rogers, E. (2019). Broiler FCR optimization using norm optimal terminal iterative learning control. *IEEE Transactions on Control Systems Technology*, 29(2), 580–592, Doi:10.1109/TCST.2019.2954300.

Jonnalagadda, V. K. and Elumalai, V. K. (2021). Norm Optimal Iterative Learning Control for Non-Repetitive Trajectory Tracking of Servo System. *International Review of Automatic Control (IREACO)*, 14(3), 153–161, Doi:10.15866/ireaco.v14i3.20692.

Lai, J., Jin, S., Xia, P., Huang, J. and Yuan, P. (2021). Dual-Loop Iterative Learning Control of Robot Manipulator for Human-Robot cooperation. *In Journal of Physics: Conference Series*, 1846(1), 1–9, Doi:10.1088/1742-6596/1846/1/012090.

Li, B., Tan, Y., Chen, J., Liu, X. and Yang, S. (2020). Precise active seeding downforce control system based on fuzzy PID. *Mathematical Problems in Engineering*, Doi:10.1155/2020/5123830.

Li, M., Chen, T., Cheng, R., Yang, K., Zhu, Y. and Mao, C. (2021). Dual-Loop Iterative Learning Control With Application to an Ultraprecision Wafer Stage. *IEEE Transactions on Industrial Electronics*, 1–10, Doi:10.1109/TIE.2021.3120481.

Lin, C. Y., Sun, L. and Tomizuka, M. (2015). Matrix factorization for design of Q-filter in iterative learning control. *In 2015 54th IEEE Conference on Decision and Control (CDC)*, 18(11), 6076–6082, Doi:10.1109/CDC.2015.7403175.

Long, X., He, Z. and Wang, Z. (2021). Online optimal control of robotic systems with single critic NN-based reinforcement learning. *Complexity*, Doi:10.1155/2021/8839391.

Panomruttanarug, B. and Longman, R. W. (2006). Converting repetitive control into stable learning control by iterative adjustment of initial state. *Advances in the Astronautical Sciences*, 124, 667—686.

Panomruttanarug, B., Longman, R. W. and Phan, M. Q. (2009). Multiple model robustification of iterative learning and repetitive control laws including design from frequency response data. *Advances in the Astronautical Sciences*, 134(01), 2259—2278.

Panomruttanarug, B. (2020). Position Control of Robotic Manipulator Using Repetitive Control Basedon Inverse Frequency Response Design. *International Journal of Control, Automation and Systems*, 18(11), 2830–2841, Doi:10.1007/s12555-019-0518-2.

Thor, M. and Manoonpong, P. (2019). Error-based learning mechanism for fast online adaptation in robot motor control. *IEEE transactions on neural networks and learning systems*, 31(6), 2042–2051, Doi:10.1109/TNNLS.2019.2927737.

Tomizuka, M. (1987). Zero phase error tracking algorithm for digital control. *Journal of Dynamic Systems Measurement and Control*, 109(1), 65–68, Doi:10.1115/1.3143822

Volckaert, M., Diehl, M. and Swevers, J. (2013). Generalization of norm optimal ILC for nonlinear systems with constraints. *Mechanical Systems and Signal Processing*, 39(1-2), 280–296, Doi:10.1016/j.ymssp.2013.03.009.

Yang, J., Na, J., Gao, G. and Zhang, C. (2018). Adaptive neural tracking control of robotic manipulators with guaranteed nn weight convergence. *Complexity*, Doi:10.1155/2018/7131562.

Zheng, M., Wang, C., Sun, L. and Tomizuka, M. (2017). Design of arbitrary-order robust iterative learning control based on robust control theory. *Mechatronics*, 47, 67–76, Doi:10.1016/j.mechatronics.2017.08.009

Zhu, Q. and Xu, J. X. (2018). Dual IM-based ILC scheme for linear discrete-time systems with iteration-varying reference. *IET Control Theory & Applications*, 12(1), 129–139, Doi:10.1049/iet-cta.2017.0592.

Zhu, M., Ye, L. and Ma, X. (2020). Estimation-based quadratic iterative learning control for trajectory tracking of robotic manipulator with uncertain parameters. *IEEE Access*, 8, 43122–43133, Doi:10.1109/ACCESS.2020.2977687.