Precision Engineering 81 (2023) 112–123

Contents lists available at ScienceDirect

Precision Engineering

journal homepage: www.elsevier.com/locate/precision

Neural network-based iterative learning control of a piezo-driven nanopositioning stage[☆]

Jie Ling[a,*], Zhao Feng[b], Long Chen[a], Yuchuan Zhu[a], Yongping Pan[c]

[a] National Key Laboratory of Science and Technology on Helicopter Transmission, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China
[b] College of Power and Mechanical Engineering, Wuhan University, Wuhan 430072, China
[c] School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou 510006, China

ARTICLE INFO

Keywords:
Piezoelectric actuator
Nanopositioning system
Neural network
Iterative learning control
Nonlinear control

ABSTRACT

The piezo-driven nanopositioning stage (PNS) is a key device to provide fast and precise motions for applications such as micromanipulation, microfabrication, and microscopy scanning. However, inherent nonlinearities associated with system perturbations bring difficulties to controller design. Regarding repetitive tasks for a PNS, existing control schemes are mainly dedicated to model inversion-based iterative learning control (ILC), which relies heavily on model accuracy. In this paper, a novel online identification and control scheme named neural network-based ILC (NN-ILC) is proposed for repetitive tracking of the PNS. The ILC scheme reduces repetitive errors due to the linear dynamics and invariable disturbance during each iteration. Neural networks are integrated into the ILC scheme to minimize the residual non-repetitive errors resulting from unknown nonlinear dynamics and model perturbations. Convergence results in both the time and iteration domains are demonstrated according to the Lyapunov stability theory. Comprehensive experiments of sinusoidal and triangular tracking references with different frequencies (5~20 Hz) and different peak-to-peak amplitudes (5~ 20 μm) are conducted on a real-time control testbed. Results show that the root mean square error of the proposed NN-ILC for 20 μm tracking cases is improved by up to 37% from feedback proportional-derivative (PD) control with neural networks and by up to 20% from feedforward PD-type ILC.

1. Introduction

Repetitive tasks are widely executed in industrial applications, including, but not limited to high-speed atomic force microscopy (AFM) imaging [1], wafer positioning [2], additive manufacturing [3], gait rehabilitation [4], hard disk driving [5]. For tracking of a desired trajectory in such repetitive processes, iterative learning control (ILC) has been proven as an effective method [6–8]. In an ILC scheme, the control sequence for the current iteration is refined by learning the error information from previous executions. Then, tracking performance can be improved and desired trajectory can gradually be obtained [7]. If the system is completely known or can be precisely identified, some popular techniques can be applied directly to design ILC algorithms, such as gradient-based ILC [9], norm-optimal ILC [3], model-inversion based ILC [10]. However, system dynamics or parameters are never perfectly known in many industrial applications [7–9,11–13]. For this, the ILC design considering unknown systems needs to be addressed.

A number of efforts have been made to design ILC for fully or partially unknown systems. For nonlinear systems, comprehensive surveys can be found in [8,14]. Among these existing schemes, adaptive ILC (AILC) schemes are the most widely discussed for systems with uncertain parameters. In [15], an AILC with a differential learning law is proposed to achieve error convergence in L^2 norm. The ILC is also merged with the adaptive robust control to form an adaptive robust iterative learning control (ARILC) in [16], which is proved effective for both periodic and nonperiodic structured system uncertainties. For a nonlinear system with a completely unknown control gain, an AILC with a Nussbaum-type learning function is investigated in [17]. The issue of parameter adaption in the time/iteration domain is addressed in [18,19], thus a unified AILC (UAILC) framework is proposed. In the UAILC framework, effects resulted from the iteration-varying disturbance or initial tracking errors can be eliminated in the learning processes. Unlike previous works which apply a deadzone operator (in [15,16]) or projection mechanism (in [17,18]), fuzzy ILC

[☆] This paper was recommended by Associate editor Prof. R. Leach.
[*] Corresponding author.
E-mail address: meejling@nuaa.edu.cn (J. Ling).

https://doi.org/10.1016/j.precisioneng.2023.02.006
Received 9 June 2022; Received in revised form 17 January 2023; Accepted 13 February 2023
Available online 15 February 2023

schemes are developed by combining fuzzy modules with sliding-mode-like module or neural network module [19] into the design of adaptive learning functions. With respect to the robustness of AILC to uncertainties, some efforts have been made to provide effective solutions. For norm-bounded unstructured uncertainties, a robust ILC (RILC) is developed by replacing classical resetting conditions with more practical alignment conditions in [20]. Considering the nonparametric uncertainties in a physical system, a robust ILC controller is synthesized in [20]. Another interesting and meaningful aspect of the AILC is learning from varying iteration lengths, and novel AILC schemes are demonstrated effective for nonlinear systems with partial information in [21,22] and for nonlinear dynamics which are not globally Lipschitz continuous in [23].

Although most of the aforementioned AILC schemes such as those in [15–18,20–23] are effective for a class of nonlinear systems, it is noted that a crucial premise for these methods is that the unknown parameters can be linearly parameterized with pre-defined nonlinear functions [19]. Fortunately, if a system is completely unknown or cannot be represented in such linear parameterization forms, neural networks (NNs) can be utilized to estimate the unknown nonlinear dynamics [24–28].

For a piezo-driven stage, which is widely utilized as a motion producer in an AFM, precision motion tracking is a key issue when designing the controller [29–31]. The difficulty lies in the modeling and control of nonlinearities of the system. Some offline identification and control schemes have been proposed to handle the nonlinearities. Inherent characteristics such as creep, and hysteresis can be modeled offline and compensated by feedforward terms. Specific methods can be referred to in the review in [32,33]. To be specific, multi-layer feedforward neural networks (MFNN) are applied in [34,35] to model the nonlinearities offline by training a set of input/ouput data. This is effective and accurate as proven in these works. However, on the one hand, offline training consumes a lot of data and computation for different work modes. On the other hand, external environment changes, such as temperature variation [33], load variation [29,36], etc, lead to parameter or model perturbation so that put forward higher requirements on the robustness of the control system. The controllers associated with the offline trained models will significantly deteriorate due to the parameter or model perturbation [36,37]. To some extent, online identification and control can make an improvement in terms of robustness and stability. Many closed-loop schemes have been developed as surveyed in [38]. It should be noted that an online recurrent-neural-network (RNN) identification associated with a model predictive control (MPC) is proposed in [39] for the motion tracking of piezo actuators. A composite MPC with feedforward hysteresis compensation structure is developed in [40]. To model the strong asymmetrical hysteresis nonlinearity, a nonlinear model predictive control scheme is proposed in [41]. An online neural-network-based sliding mode control is developed in [42] for robust adaptive motion tracking of the piezo stage. However, for the online identification and control of repetitive tasks of a piezo-driven nanopositioning stage, the above methods cannot be applied directly.

To tackle the problem of precision motion tracking of repetitive tasks for a piezo-driven nanopositioning stage, a novel online neural network-based ILC (NN-ILC) scheme is proposed. To reduce the repetitive errors due to the linear dynamics and invariable disturbance during each iteration, the iterative learning control scheme is built. For the residual non-repetitive errors resulted from unknown nonlinear dynamics and the parameter or model perturbation, the online neural networks are integrated into the ILC scheme. The difficulty of controller design is to make sure that the weights of the neural network can converge in both the time domain and the iteration domain in the ILC scheme. Through the cooperation of the ILC and online NNs, the motion tracking precision can be improved for the repetitive trajectories of the piezo-driven nanopositioning stages.

With comparisons to existing ILC schemes, the novelty of this work can be reflected in three aspects:

(i) Unlike the feedforward methods in [34,35] where the NNs are trained offline, the NNs in the ILC schemes are trained online in each iteration so that they can handle the parameter or model perturbation. The method in [28] is proposed for a simple first-order system with only simulation verification;
(ii) Compared to the deadzone or projection mechanism based ILCs in [15–18] or the saturation function based ILCs in [19], no truncation is operated for the identification or control processes in the NN-ILC scheme;
(iii) Compared to the adaptive ILCs in [15–18,20–23], the proposed NN-ILC can be used to a wider class of nonlinear systems whose unknown parameters cannot be linearly parameterized with known nonlinear functions. Besides, unlike the adaptive ILCs in [15–19] which simply convert the iteration domain into extended time domain and the error information from the last iteration were not utilized to enhance the tracking performance in the subsequent iteration, the proposed NN-ILC scheme aims to eliminate tracking errors along with the iteration axis by using the error information from previous iterations directly to formulate the learning function.

The main contributions of this work include:

(i) A novel NN-ILC scheme is proposed to eliminate the repetitive errors and the residual non-repetitive errors simultaneously in an online execution to further improve the tracking performance for systems subjected to unknown dynamics and nonlinearities.
(ii) An adaptive law to online update the weights of NN is developed and stability along the spanned time domain as well as convergence in the iteration domain is proven for the NN-ILC scheme.
(iii) Experimental results based on a customized piezo-driven nanopositioning stage are presented and analyzed for the reference of the controller practicers.

The remainder of this paper is arranged as follows. Section 2 formulates the control problem and system description, control objective, and some useful definitions are also presented. Section 3 presents the NN-ILC design as well as the convergence analysis. The experimental setup is introduced in Section 4, and the experimental results are discussed in Section 5. Finally, conclusions are made in Section 6.

2. Problem formulation

Notations: In the following sections, $\mathbb{R}$, $\mathbb{R}^+$ and $\mathbb{R}^n$ are denoted for the real number space, positive real number space and real n-vector space, respectively, L_∞ represents the bounded signal space, $\Omega_c := \{\mathbf{x} \mid \|\mathbf{x}\| \leq c\}$ denotes the ball of radius c, $\|\mathbf{x}\|$ is the Euclidean norm of $\mathbf{x}$, C^m is the function spaces for which are mth-order continuous differentiable, where $c \in \mathbb{R}^+$, n and m are positive integers, and $\mathbf{x} \in \mathbb{R}^n$. Thus, Theorem 1 is proved.

2.1. System description

Generally, a piezo-driven nanopositioning stage can be modeled as a Hammerstein structure [33,37,42,43]. As shown in Fig. 1, a nonlinear operator $H(\cdot)$ is cascaded with the linear dynamics $G(s)$, where s is the continuous variable in the frequency domain, $u \in \mathbb{R}$, $h \in \mathbb{R}$ and $x \in \mathbb{R}$ are the driving voltage, intermediary hysteresis signal, and the output position, respectively. $d_{in} \in \mathbb{R}$ and $d_{out} \in \mathbb{R}$ are assumed to be unknown input or output disturbances from the environment.

As described in many existing works, such as [42,44–46], the entire dynamical model of a piezo-driven nanopositioning stage with a Hammerstein structure in Fig. 1 can be expressed as,

$$m\ddot{x}(t) + b\dot{x}(t) + kx(t) + d_{out}(t) = \gamma u(t) + d_{in}(t) + h(t) \tag{1}$$

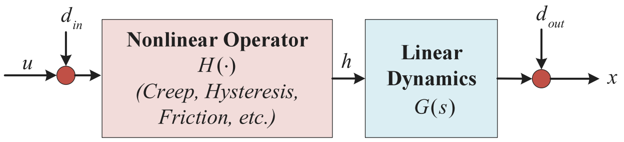

Fig. 1. The Hammerstein structure of a piezo-driven nanopositioning stage consisting of a nonlinear operator cascaded with linear dynamics.

where, t is the time variable, m, b, and k denote the mass, damping coefficient, stiffness of the system, γ is the piezoelectric coefficient with $\gamma > 0$. $h(t)$: $\mathbb{R}^n \mapsto \mathbb{R}$ is an unknown continuous nonlinear function including the hysteretic effect and other model uncertainties. Let $x_d(t) \in \mathbb{R}$ defined as a desired output. The following assumptions are made about the class of systems described in Eq. (1).

Assumption 1. $h(t)$ is of C^1.

Assumption 2. System states $x(t)$, $\dot{x}(t)$, $\ddot{x}(t)$ and the control signal $u(t)$ in Eq. (1) are measurable.

Remark 1. To facilitate the online identification process, $x(t)$, $\dot{x}(t)$ are assumed to be available. In case these variables cannot be obtained directly, some indirect approaches (sliding-mode observer with high-order functions [47] and neural network based observer [48]) can be adopted. This is not the focus of this paper and thus is omitted.

2.2. Some definitions

The following definitions and lemmas are listed as follows for derivations and analysis in subsequent sections.

Definition 1 (*Persistently Excited Condition [28]*). A matrix or vector function ϕ is persistently excited (PE) with a level of excitation $\alpha_1 > 0$ if there are positive constants T and α_2 such that

$$\alpha_1 I \leq \int_t^{t+T} \phi(\tau)\phi(\tau)^T d\tau \leq \alpha_2 I. \tag{2}$$

Lemma 1 (*Spatially Localized Approximation [24]*). *For a desired trajectory $\mathbf{x}(t) : \mathbb{R}^+ \mapsto \Omega_{c_x}$, a function $f(\mathbf{x}) : \Omega_{c_x} \mapsto \mathbb{R}$ can be approximated by the radial basis function (RBF)-NNs with a finite amount of NN nodes within a local region along $\mathbf{x}(t)$ such that*

$$f(\mathbf{x}) = W^{*T}\Phi(\mathbf{x}) + \zeta(\mathbf{x}) \tag{3}$$

with $W^* = \left[w_1^*, w_2^*, \ldots, w_N^*\right] \in \mathbb{R}^{M\times N}$ and $\Phi(\mathbf{x}) = \left[\phi_1(\mathbf{x}), \phi_2(\mathbf{x}), \ldots, \phi_M(\mathbf{x})\right]^T \in \mathbb{R}^M$ are the weight vector and the PE regressor of the NNs, where w_i and ϕ_k denote the elements of the vectors W^* and Φ with an index $i \in [1, M]$ and $k \in [1, N]$, ζ stands for the residual approximation error.

2.3. Problem statement

According to the preliminaries, the control problem can then be formulated here. The control aim is to develop a suitable ILC scheme for repetitive motion tracking of a class of piezo-driven systems described in Eq. (1). Applying the learning function designed from the converged weight vector and the regressor of the NNs, the system under designed NN-ILC should be capable of tracking the desired trajectory and achieving zero stable errors within the iteration domain.

3. Controller design

In this section, the NN-ILC scheme is proposed, and the convergence in both the time domain and iteration domain is analyzed.

3.1. Proposed NN-ILC

The system in Eq. (1) can be rewritten as,

$$m_n\ddot{x}(t) + b_n\dot{x}(t) + k_n x(t) - h(t)/\gamma + d_n(t) = u(t) \tag{4}$$

where the nominal parameters are $m_n = m/\gamma$, $b_n = b/\gamma$, $k_n = k/\gamma$, the lumped environmental disturbance d_n describes the input and output disturbances. In the following sections, the time variable t is omitted for brevity.

Denote the unknown term as $f(x,\dot{x}) = b_n\dot{x} + k_n x - h/\gamma$, and use the NNs to estimate the unknown term such that $f(x,\dot{x}) = W^{*T}\Phi + \zeta_f$ according to Lemma 1, then the system can be expressed as

$$m_n\ddot{x} + W^{*T}\Phi + \underbrace{\zeta_f + d_n}_{D} = u \tag{5}$$

where, W^* and Φ stand for the ideal weight matrix and the excitation function matrix of the NNs respectively, ζ_f is a high-order term which stands for the residual error between the estimated and the real dynamics, the lumped disturbance is denoted as $D \triangleq \zeta_f + d_n$.

Remark 2. It can be seen that D accounts for the environmental input/output disturbance as well as the residual error. Note that for a physical system, the environmental disturbance d_n is bounded, and also the residual error ζ_f is bounded, so that the lumped disturbance is bounded as $D \leqslant \bar{D}$ with $\bar{D} \in \mathbb{R}^+$.

Define the tracking error as

$$e = x_d - x \tag{6}$$

where x_d is the given reference to be tracked.

Design an auxiliary variable related to the tracking error as

$$\sigma = k_d\dot{e} + k_p e \tag{7}$$

where k_p and k_d are the feedback parameters to be determined. The differentiation of σ is obtained as,

$$\dot{\sigma} = k_d\ddot{e} + k_p\dot{e}. \tag{8}$$

Substituting Eqs. (5) and (6) into Eq. (8), it can be derived as,

$$\begin{aligned}\frac{m_n}{k_d}\dot{\sigma} &= m_n\ddot{e} + \frac{m_n k_p}{k_d}\dot{e}\\ &= m_n\ddot{x}_d - m_n\ddot{x} + \frac{m_n k_p}{k_d}\dot{e}\\ &= m_n\ddot{x}_d - u + W^{*T}\Phi + D + \frac{m_n k_p}{k_d}\dot{e}\end{aligned} \tag{9}$$

Then, the control law can be formulated as,

$$u_j = u_{1,j} + u_{2,j} \tag{10}$$

with

$$\begin{cases} u_{1,j} = \underbrace{(k_p e_j + k_d\dot{e}_j)}_{\sigma_j} + \hat{W}_j^T\Phi + m_n\ddot{x}_d + \frac{m_n k_p}{k_d}\dot{e}_j \\ u_{2,j+1} = u_{2,j} + L\sigma_j \end{cases} \tag{11}$$

where j denotes the iteration in the ILC scheme, $u_{1,j}$ accounts for the nonlinearity compensator and $u_{2,j}$ accounts for iterative control of the repetitive errors in each iteration, $L \in (0,1)$ is a learning gain to be determined, $\hat{W}_j$ is the estimated weights of the NNs in the jth iteration and it is updated based on the following law,

$$\dot{\hat{W}}_j = \Gamma\Phi\sigma_j \tag{12}$$

where $\Gamma \in \mathbb{R}^+$ is a positive parameter to be determined. The residual estimation error between the ideal weights W^* and the trained weights $\hat{W}_j$ is denoted as $\tilde{W}_j \triangleq W^* - \hat{W}_j$.

Remark 3. For the nonlinearity compensator $u_{1,j}$, the residual errors are threefold, i.e., the unmodeled environmental disturbance, the error ζ_f between NNs and dynamics, and the error $\tilde{W}_j$ between ideal NNs and trained NNs. That is the reason to introduce the ILC scheme to add an iterative control term as $u_{2,j}$ so that the residual errors can be further reduced and finally the tracking performance will be improved. It can be found that the convergence of the proposed control law is two-dimensional, i.e., both in the time domain and in the iteration domain. Herein, the convergence will be analyzed in the two domains.

3.2. Boundedness of errors in the time domain

Consider the following Lyapunov candidate function for the jth iteration in the time domain as follows, where $\eta \triangleq m_n/k_d$.

$$S_j = \frac{1}{2}\eta\sigma_j^2 + \frac{1}{2}\tilde{W}_j^T \Gamma^{-1}\tilde{W}_j \tag{13}$$

Taking its time derivative and substituting $\eta\dot{\sigma}_j$ using Eq. (9) as derived in Appendix A in details, we have

$$\dot{S}_j \leq -\frac{1}{2}\sigma_j^2 + \frac{1}{2}\tilde{u}_{2,j}^2. \tag{14}$$

Therefore, if $\tilde{u}_{2,j}$ is bounded, S_j is bounded, which means that the tracking error as well as the NN weights will asymptotically converge at the current iteration j.

3.3. Convergence in the iteration domain

Consider the following discrete-time Lyapunov candidate function as

$$V_j = \int_{\tau=0}^{T}\left(\tilde{u}_{2,j}^2\right)d\tau + L\tilde{W}_j^T(0)\Gamma^{-1}\tilde{W}_j(0). \tag{15}$$

Then, the increment function can be obtained as,

$$\begin{aligned}\Delta V_j =\ & \Delta V_{j+1} - \Delta V_j \\ =\ & \int_{\tau=0}^{T}\left(\tilde{u}_{2,j+1}^2 - \tilde{u}_{2,j}^2\right)d\tau \\ & + L\left(\tilde{W}_{j+1}^T(0)\Gamma^{-1}\tilde{W}_{j+1}(0) - \tilde{W}_j^T(0)\Gamma^{-1}\tilde{W}_j(0)\right).\end{aligned} \tag{16}$$

Note that,

$$\Delta\tilde{u}_{2,j} = \tilde{u}_{2,j+1} - \tilde{u}_{2,j} = (D - u_{2,j} - Le_j) - (D - \tilde{u}_{2,j}) = -L\sigma_j, \tag{17}$$

therefore,

$$\begin{aligned}\tilde{u}_{2,j+1}^2 - \tilde{u}_{2,j}^2 &= (\Delta\tilde{u}_{2,j} + \tilde{u}_{2,j})^2 - \tilde{u}_{2,j}^2 \\ &= \Delta\tilde{u}_{2,j}^2 + 2\Delta\tilde{u}_{2,j}\tilde{u}_{2,j} \\ &= L^2\sigma_j^2 - 2L\sigma_j\tilde{u}_{2,j}.\end{aligned} \tag{18}$$

Recall that $\sigma_j\tilde{u}_{2,j} = \dot{S}_j + \sigma_j^2$ in Eq. (14), then

$$\begin{aligned}\tilde{u}_{2,j+1}^2 - \tilde{u}_{2,j}^2 &= L^2\sigma_j^2 - 2L(\dot{S}_j + \sigma_j^2) \\ &= -L(2-L)\sigma_j^2 - 2L\dot{S}_j.\end{aligned} \tag{19}$$

Next, we have

$$\begin{aligned}\Delta V_j =& \int_{\tau=0}^{T}\left(\tilde{u}_{2,j+1}^2 - \tilde{u}_{2,j}^2\right)d\tau \\ & + L\left(\tilde{W}_{j+1}^T(0)\Gamma^{-1}\tilde{W}_{j+1}(0) - \tilde{W}_j^T(0)\Gamma^{-1}\tilde{W}_j(0)\right) \\ =& - L(2-L)\int_{\tau=0}^{T}\sigma_j^2\,d\tau - L\int_{\tau=0}^{T}2\dot{S}\,d\tau \\ & + L\left(\tilde{W}_{j+1}^T(0)\Gamma^{-1}\tilde{W}_{j+1}(0) - \tilde{W}_j^T(0)\Gamma^{-1}\tilde{W}_j(0)\right)\end{aligned} \tag{20}$$

with

$$\begin{aligned}\int_{\tau=0}^{T}2\dot{S}_j\,d\tau &= \eta\sigma_j^2\Big|_0^T + \tilde{W}_j^T\Gamma^{-1}\tilde{W}_j\Big|_0^T \\ &= \eta\sigma_j^2(T) - \eta\sigma_j^2(0) + \tilde{W}_j^T(T)\Gamma^{-1}\tilde{W}_j(T) \\ &\quad - \tilde{W}_j^T(0)\Gamma^{-1}\tilde{W}_j(0).\end{aligned} \tag{21}$$

The resetting condition for each iteration in an ILC scheme is defined as,

$$x_d(0) - x_j(0) = e_j(0) = 0 \tag{22}$$

and the iteration-domain update for the NN weights is defined as,

$$\hat{W}_{j+1}(0) = \hat{W}_j(T). \tag{23}$$

Then, as presented in Appendix A in detail, we have

$$\Delta V_j = - L(2-L)\int_{\tau=0}^{T}\sigma_j^2\,d\tau - L\eta\sigma_j^2(T) < 0 \tag{24}$$

Therefore, the value of $V_j \to 0$ during iterations so that $\tilde{u}_{2,j} \to 0$ is guaranteed. Recall that in the time domain, then $\dot{S}_j \leq -\frac{1}{2}\sigma_j^2 + \frac{1}{2}\tilde{u}_{2,j}^2 \leq 0$ can be guaranteed.

Hereto, proof of convergence in both the time domain and iteration domain of the proposed control is completed.

3.4. Overall design procedure

The overall control law is presented in Eq. (10). In practice, to avoid the effects of the sensor noise on the ILC learning process, a low-pass filter $Q = 1/(\lambda s + 1)$ is normally added to screen out useless information introduced by the sensor noise in high frequencies, where s is the frequency domain variable. Therefore, the iterative learning control component in Eq. (11) is modified as,

$$u_{2,j+1}(s) = Q(u_{2,j}(s) + L\sigma_j(s)) = \frac{u_{2,j}(s) + L\sigma_j(s)}{\lambda s + 1}. \tag{25}$$

Remark 4. The introduction of Q-filter will not affect the convergence of ILC operations as can be referred to [3,6]. When $Q = 1$ for all the frequencies, it means that all the frequency components including the sensor noise learned and stored to generate $u_{2,j+1}$, however, as the sensor noise is random and nonrepetitive, the usage of the noise information is not beneficial for $u_{2,j+1}$. Thus, the bandwidth of the Q-filter needs to be chosen according to the experimental testbed such that noise is rejected by the ILC scheme.

The control block diagram of the proposed NN-ILC is shown in Fig. 2. For the jth iteration, the control signal u_j consists of two components, i.e., a feedback term $u_{1,j}$ and a feedforward term $u_{2,j}$. Parameters associated with the scheme can be determined as following steps:

(i) Feed a step or sweep signal into the nanopositioning stage, record the input and output data, and identify the linear dynamics G as shown in Fig. 1.
(ii) Tune the feedback gains k_p and k_d alone for the system G.
(iii) According to the bandwidth of G, design the Q-filter so that the bandwidth of Q is within that of G.
(iv) Initialize the neural networks and set Γ as a small positive constant at first.
(v) According to the bandwidth of G, design the Q-filter so that the bandwidth of Q is within that of G. Set L as a small positive constant within $(0, 1)$ at first.
(vi) Perform the ILC iterations, and tune Γ and L increasingly to achieve a satisfied convergence speed during each iteration with no fluctuations.

4. Experimental test

4.1. Experimental setup

To evaluate the performance of the proposed NN-ILC, an experimental system with a custom-designed piezo-driven nanopositioning stage is built as shown in Fig. 3. For the piezo-driven nanopositioning stage, a piezo stack (model: RP150/5 × 5 × 18, from Harbin Chip

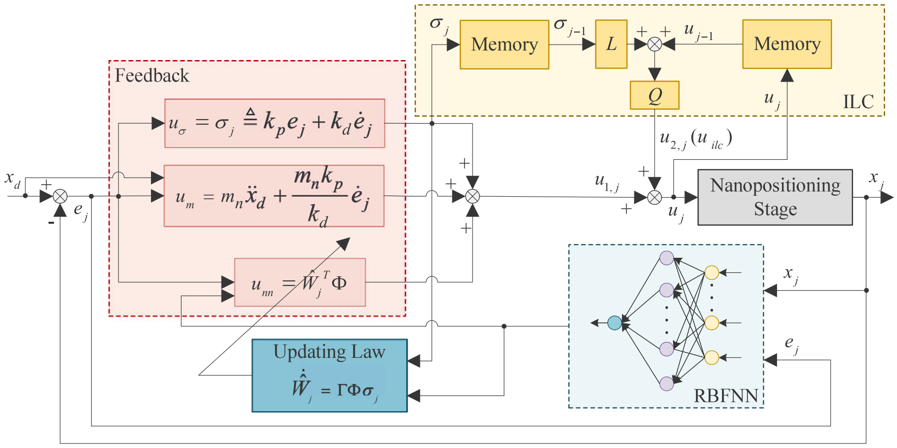
Feedback
ILC
Memory
L
Memory
Q
σ_j
σ_{j-1}
u_{j-1}
u_j
$u_{2,j}(u_{ilc})$
$u_{1,j}$
$u_\sigma = \sigma_j \triangleq k_p e_j + k_d \dot{e}_j$
$u_m = m_n \ddot{x}_d + \frac{m_n k_p}{k_d} \dot{e}_j$
$u_{nn} = \hat{W}_j^T \Phi$
x_d
e_j
x_j
Nanopositioning Stage
Updating Law
$\dot{\hat{W}}_j = \Gamma \Phi \sigma_j$
RBFNN
+
-

Fig. 2. Block diagram of the proposed NN-ILC scheme in the jth iteration.

Technology Co., Ltd.) with a stroke of 20 μm driven by 150 V voltage is adopted as the motion actuation. The motion is then amplified by the compliant mechanism and recorded by an eddy current displacement sensor (model: BZF-2, from Huarui Instrument Co., Ltd.).

The control signal (0–10 V) is produced by 16-bit digital to analog interfaces (model: PCI-6259 I/O Board, from National Instrument Co., Ltd.) of the data output module in real-time xPC Target. A power amplifier module (model: 7224, from AETechron Co., Ltd.) with a fixed gain (16 times) amplifies the input voltage and generates excitation voltage. The output displacement is read by a displacement sensor and is subsequently passed to the data input module in the PCI-6259 board in real-time xPC Target. The controller is designed in MATLAB®/Simulink block diagram on the host computer (running on Windows 10 and an Intel Core i5-7400 CPU @ 3.00 GHz), where the Solver is chosen as fixed-step ode 4 with a step size being 50 μs, and then built, downloaded, and executed on the xPC Target through TCP/IP communications.

4.2. System identification

A step test is performed in the experiments to identify the nominal model of the linear part $G(s)$. A second-order model is obtained using the System Identification Toolbox in Matlab®as,

$$G(s) = \frac{8.82 \times 10^7}{s^2 + 244.9s + 1.103 \times 10^7} \tag{26}$$

where the fitness between the measured data and the model is 93% as shown in Fig. 4. Therefore, we have the nominal parameter as $m_n = 1/(8.82 \times 10^7)$, and the bandwidth of G is around 820 Hz.

Remark 5. The goal to perform a system identification is twofold: (1) to obtain a nominal model of the linear part to provide a feedback term C_n as shown in Fig. 2 so that the NNs can focus on the linear part approximation, (2) to measure the bandwidth of the system to provide guidance for the tuning of the Q-filter.

4.3. Controller set

To evaluate the performance of the proposed NN-ILC scheme, another four existing schemes are also implemented in this paper to make fair comparisons. The controller set is arranged as: (1) Feedback proportional–integral–derivative (PID) control alone; (2) Feedback PD control with neural networks (PDNN); (3) Model prediction control (MPC) [40]; (4) PD-type ILC [6]; (5) The proposed NN-ILC in this paper.

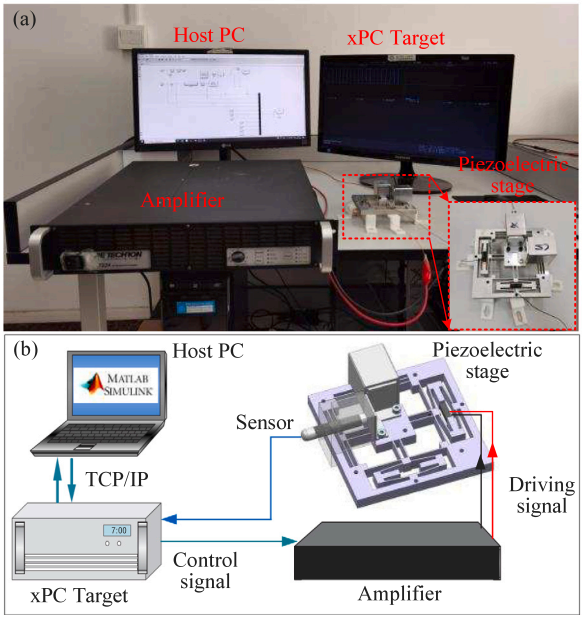

Fig. 3. Test results of step size and speed under different driving frequencies. (a) Step results. (b) Speed results.

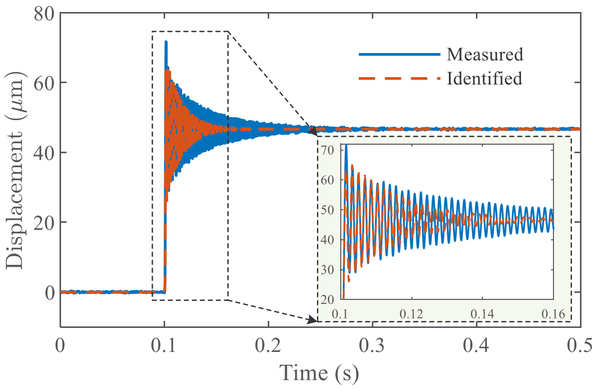

Fig. 4. Step responses of the measured data and simulated data of the identified model.

4.3.1. PID

The control law can be given as,

$$u = k_p e + k_i \int e dt + k_d \dot{e}, \tag{27}$$

where k_p, k_i, k_d are the controller gains, e is the tracking error.

4.3.2. PDNN

The control law can be given as,

$$u = k_p e + k_i \int e dt + k_d \dot{e} + \hat{W}^T \Phi, \tag{28}$$

where k_p, k_i, k_d are the PID gains as in Eq. (27), $\hat{w}$ is the estimated weights and ϕ is the regressor of the NNs.

4.3.3. MPC

The control force for an MPC scheme generates from control optimization. For the controlled plant, an identified nominal model can be used as the reference model. The control law can be obtained by minimizing the cost function J such that the constraints of variables can be satisfied in a quadratic programming (QP) problem. The detailed equations can refer to our previous work in [40].

In this work, we applied the command *mpc* in MATLAB to formulate the MPC controller, where the key parameters consist of a prediction horizon property p and a control horizon property m.

4.3.4. PD-ILC

The control law is

$$u_{j+1} = u_j + L(k_p e_j + k_d \dot{e_j}), \tag{29}$$

where k_p and k_d are the PD gains.

4.3.5. NN-ILC

The RBFNN is adopted to approximate the unknown nonlinear dynamics in the NN-ILC scheme. As given by Definition 1, a matrix or vector is PE when Eq. (2) is satisfied. This means that for the regressor $\phi(x)$ of NNs, the matrix $\int_t^{t+T} \phi(\tau)\phi(\tau)^T d\tau \in \mathbb{R}^{M \times M}$ should be positive definite over any finite interval. However, the condition of PE is restrictive and often infeasible to implement or monitor online. It should be noted that in this work, $\phi(x)$ is defined as a Gaussian function with $\Phi(\mathbf{x}) = \left[\phi_1(\mathbf{x}), \phi_2(\mathbf{x}), \dots, \phi_M(\mathbf{x})\right]^T \in \mathbb{R}^M$, and for a m-nodes neural networks, the radial basis function is designed as,

$$\phi(x)_k = exp(\frac{-\|x - c_k\|^2}{2\beta}), k = 1, 2, \dots, M \tag{30}$$

where the adjustable parameter $c_k = [c_{k,1}, c_{k,2}, \dots, c_{k,4N}]^T \in \mathbb{R}^{4N}$ is the center vector, and β is the width of the Gaussian function, $M, N \in \mathbb{R}^+$. Although the input signal x is not known as a prior, the suitable values of c_k and β can be chosen to ensure the PE condition. In this work, the center of each node is uniformly distributed in $[-10, 10] \times [-20, 20]$ as shown in [49]. Furthermore, the parameter β can be tuned online increasingly to obtain satisfied convergence of the weights for each reference. In experiments, the parameters are tuned manually to satisfy the PE condition.

4.4. Parameter tuning

The parameters of feedback PID can be tuned preliminarily according to the identified model in Eq. (26) and determined finally based on the experimental results to obtain satisfied dynamic responses, i.e., rising time and overshoot, etc. The parameters in NNs (β and Γ) are tuned online increasingly to obtain satisfied convergence of the weights without deteriorated oscillations. For the parameters in ILC, it should be noted that a PD type instead of a PID type is adopted because the iterative learning process has the effect of integration. The most commonly employed method for selecting the PD gains and L gain is by tuning [6] to obtain a satisfied convergence speed. The bandwidth of the Q-filter is set as lower than that of the identified model of the system to reject the influence of sensor noise. Finally, the parameters in the NN-ILC can be determined in two steps. First, set the NN parameters as those in the PDNN case. Second, set the ILC parameters as those in the PD-ILC case. All the parameters of the four controllers in the experiments can be tuned and selected as listed in Table 1.

Table 1
Parameters of controllers.

Controller	Notation	Value
PID	k_p, k_i, k_d	0.025, 0.0128, 0.0001
PDNN	k_p, k_i, k_d, β, Γ	0.0015, 0.00118, 0.0001, 5000, 150
MPC	p, m	100, 100
PD-ILC	k_p, k_d, L, λ	0.7, 0.1, 0.95, $1/(2\pi \cdot 800)$
NN-ILC	k_p, k_d, β, Γ, L, λ	0.7, 0.1, 5000 150, 0.95, $1/(2\pi \cdot 800)$

5. Results

To evaluate the performance of the proposed controller, tracking experiments of a set of references from 5 Hz to 20 Hz with amplitudes ranging from 10 μm to 40 μm were conducted. For quantificational comparisons among the four controllers, the root-mean-square of errors (RMSE) is calculated as an index to revealing the overall tracking performance. The RMSE can be obtained as,

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{N}(x(i) - x_d(i))^2}{N}}. \tag{31}$$

where x and x_d are the measured output and the desired reference, i is the discrete time index, N is the length of the data.

5.1. Overall comparative results of the controller set

Tracking results of 5 Hz sinusoidal reference with peak-to-peak amplitude of 20 μm are shown in Fig. 5.

It can be intuitively seen from the overall tracking results that, the traditional PID controller obtains the largest tracking errors with apparent periodicity. This is mainly due to the failure to handle system nonlinearity and the inevitable lag induced by the PID as can be seen from the control voltage in Fig. 5(c). The MPC achieves better performance compared with PID and PDNN, whereas the gap between MPC and PDNN is narrow in this case. The tracking errors of PD-ILC and NN-ILC are close to each other for this case, except that it is more stable with slighter fluctuations for NN-ILC compared with PD-ILC. The tracking error with the proposed NN-ILC is the smallest and the most windless among the five controllers. It also should be noted that the control voltage of NN-ILC is the largest with the smallest lag owing to the cooperation mechanism of the feedback and feedforward terms in the control law.

5.2. Evolution process of the control voltage in NN-ILC

Taking a closer view to the evolution process of the control voltage of the proposed NN-ILC. The control components of the NN-ILC controller in the last iteration are shown in Fig. 6(a) to investigate the contributions of each control term in the NN-ILC mechanism. It can be seen that the feedback control voltage (i.e., $u_\sigma + u_m + u_{nn}$ accounts for a small proportion of the total control voltage u, while the feedforward control voltage (i.e., u_{ilc} generated from NNs contributes the most and dominates the total control voltage u. Due to the effect of iterative learning, the u_{ilc} grows instantly and the error drops so that the feedback control voltage is reduced accordingly. Finally, the converged control voltage is dominated by the feedforward. In addition,

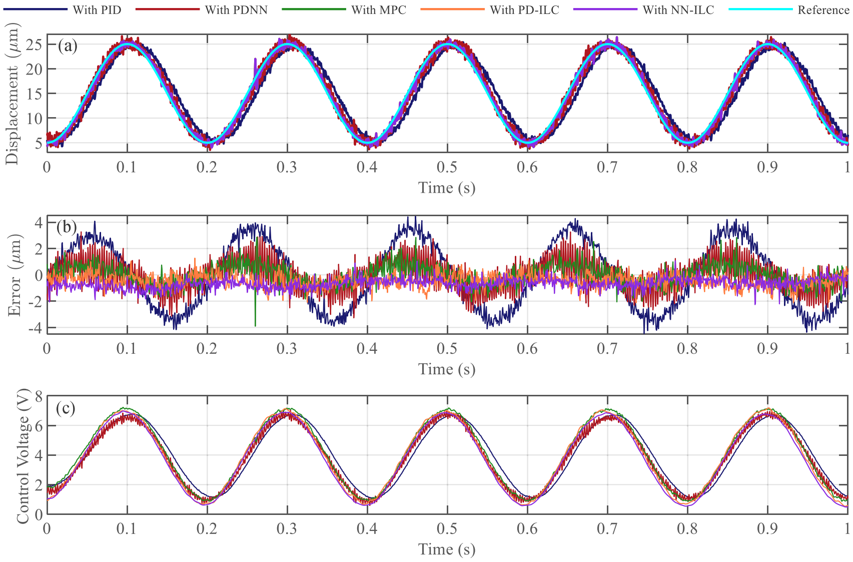

Fig. 5. Tracking results of sinusoidal reference at 5 Hz with the five controllers. (a) Overall tracking view. (b) Tracking errors. (c) Control voltages.

it can be seen that the phase lag resulting from the feedback terms can be effectively compensated by the feedforward term.

The evolution process of the control voltage in the NN-ILC scheme during the iterations is shown in Fig. 6(b) to investigate the dynamic evolution of the control voltage. The control voltage of NN-ILC converges to a steady value with the increasing iteration. At the 10th iteration, the control voltage has reached closely the level at the 50th iteration, which shows fast convergence of the NN-ILC mechanism.

According to the observations, it can be revealed that: (1) The feedback term is important to provide a stabilized plant for the installation of the feedforward term and is beneficial for the fast convergence of the feedforward term to some extent, which can be found in the following comparisons with existing PD-ILC. (2) The iterative learning mechanism is effective to compensate for the residual errors as the tracking precision is improved from the first to the last iteration. (3) The control voltage u_m from the nominal model is minor which can also be covered by the u_{nn}, however, we here identify the nominal model to obtain the bandwidth of the system so that the Q-filter can be designed properly.

5.3. Tracking results of different frequencies

To make comprehensive comparisons among the controllers, a set of tracking experiments of sinusoidal and triangular references ranging from 5 to 20 Hz and from 5 to 20 μm (peak-to-peak magnitude) were conducted. More results of the tracking cases can be found in Figs. 7 and 8. The corresponding statistic results of the RMSEs of all the cases are depicted in Fig. 9.

As can be seen from Figs. 7 and 8, for the cases with different frequencies and the same amplitudes, the errors are increased more or less for each controller. Among the five controllers, PD-ILC and NN-ILC are better than the feedback PID, PDNN and MPC. This is because the feedback controller alone is always deteriorated due to phase lag, while the feedforward controller can compensate for the errors without lags. The NN-ILC can achieve the best tracking performance compared with others in terms of smaller tracking errors and slighter fluctuations.

For the cases with different amplitudes and the same frequencies, the observations are similar to the above cases. However, the cause is different. With the increase of amplitudes, the nonlinearity effects (e.g., the hysteresis) is enlarged gradually for the piezo-driven nanopositioning system. Owing to the learning mechanism in the ILCs, the repetitive nonlinearity in each iteration can be learned and compensated notably to obtain a much smaller steady error.

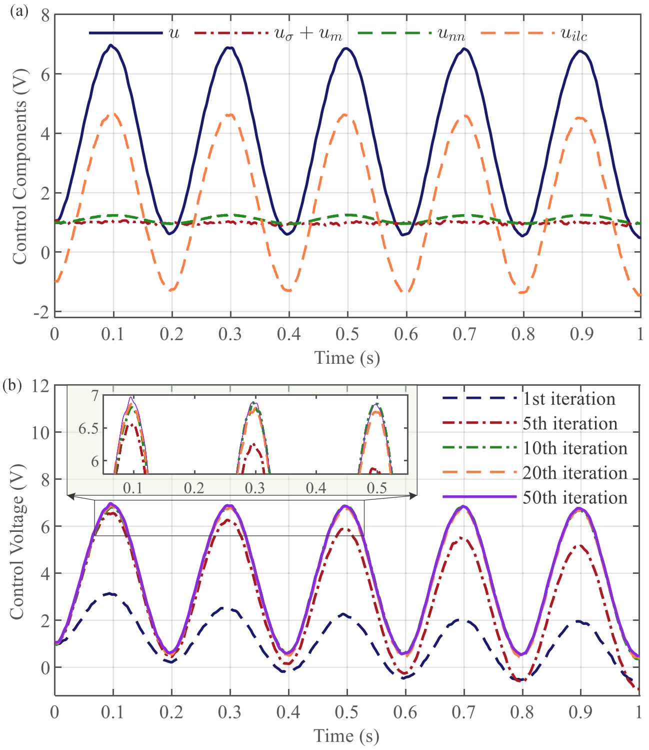

Fig. 6. Components and evolution analysis of the control voltage. (a) Control components of the NN-ILC controller in the last iteration. (b) The evolution process of the control voltage for NN-ILC during the iterations.

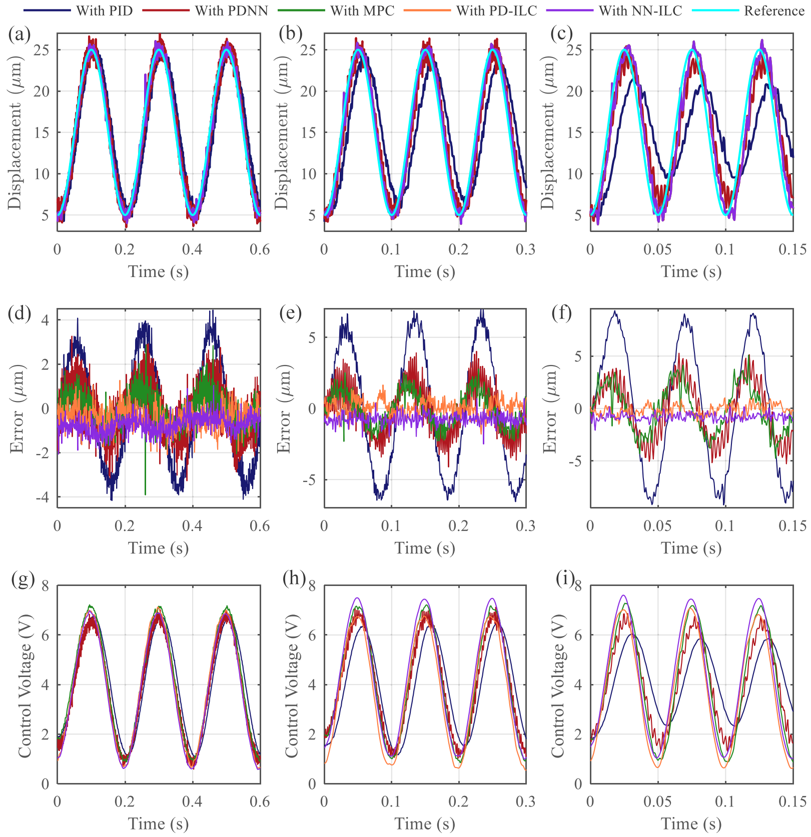

Fig. 7. Comparative tracking results of sinusoidal references with the five controllers at three frequencies. (a)–(c) Output displacements for 5, 10, and 20 Hz, respectively. (d)–(f) Tracking errors for 5, 10, and 20 Hz, respectively. (g)–(i) Control voltages for 5, 10, and 20 Hz, respectively.

For the differences between sinusoidal and triangular reference tracking, it can be found that the errors are bigger for triangular cases compared with sinusoidal cases under the same magnitude and the same frequency. It can be explained that triangular signals consist of fundamental frequency and high-order harmonic frequency, while sinusoidal signals contain a single frequency. The high-order frequency exists in the zig-zag corners and tends to deteriorate the tracking performance of the controller.

5.4. Statistic tracking error analysis

To make a quantitative analysis, the statistic RMSEs of experimental cases are shown in a radar figure for sinusoidal cases in Fig. 9(a) and triangular case in Fig. 9(b).

In the radar figure, the radial line stands for the experiment case and the latitude line stands for the RMSE value. Hence, the inner curve represents smaller RMSEs than the outer curve so that the performance is better among the controllers. It can be obtained that the proposed NN-ILC controller achieves the best tracking performances with the smallest RMSEs and locates in the innermost among the controllers.

To be more specific, for the sinusoidal references with 20 μm cases, the RMSEs of the NN-ILC in Case 7, Case 8, and Case 9 are 1.12, 1.53, 2.05 μm, respectively, which are reduced by 16%, 21%, and 37% from those of PDNN, and by 19%, 13%, 20% from those of PD-ILC respectively. For the triangular references with 20 μm cases, the RMSEs of the NN-ILC in Case 7, Case 8, and Case 9 are 1.36, 1.95, and 2.78 μm, respectively, which are reduced by 23%, 21%, and 28% from those of PDNN, and by 26%, 18%, 23% from those of PD-ILC respectively.

To be more specific, the advantages of the proposed NN-ILC over the traditional PD-ILC can be observed from the following two aspects.

(i) The initial tracking error in the first iteration of the proposed NN-ILC is much smaller than that of the traditional PD-ILC. For the 5 Hz, 10 Hz, and 20 Hz sinusoidal reference cases, the initial tracking errors of the NN-ILC are 5.87, 6.72, and 7.6 μm, respectively, while those errors of the PD-ILC are 22.5, 23.6, and 23.9 μm, respectively. This is because the feedback mechanisms are different for the two controllers. For the NN-ILC, the nonlinearity is compensated by the PDNN term to some extent.

(ii) The steady tracking errors in the last iteration of the two controllers are also different. This can be easily observed from the statistic RMSEs in Fig. 9 for that the learning mechanisms are different. For the NN-ILC, the weights of the NNs are updated along with the iterations so that the ability to handle the nonlinearity is superior to the PID mechanism in the PD-ILC.

5.5. Comparative observations between PD-ILC and NN-ILC

To evaluate the effectiveness of the learning mechanism in the proposed NN-ILC, comparative observations between the PD-ILC and NN-ILC can be made as shown in Fig. 10. On the one hand, for the tracking cases with the same frequencies, the steady tracking errors of the NN-ILC (solid lines) are consistently smaller with faster convergence speeds than those of the PD-ILC (dashed lines). On the other hand, the steady tracking errors are enlarged with the increase of the reference frequency for each controller.

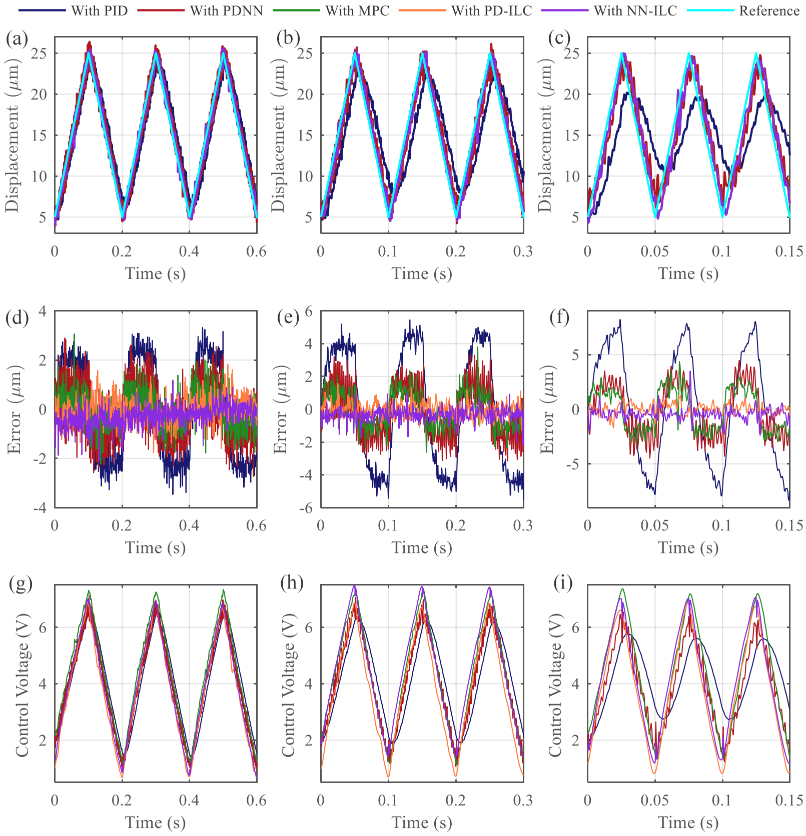

Fig. 8. Comparative tracking results of triangular references with the five controllers at three frequencies. (a)–(c) Output displacements for 5, 10, and 20 Hz cases. (d)–(f) Tracking errors for 5, 10, and 20 Hz cases. (g)–(i) Control voltages for 5, 10, and 20 Hz cases.

6. Discussions

In this section, discussions are made in terms of tracking precision and convergence speed compared with the other controllers in the experiments.

6.1. Tracking precision

Diving the controllers into feedback and feedforward groups, it can be seen that the inevitable lag induced by the limited bandwidth of the feedback controller (i.e., PID, PDNN, and MPC) deteriorates the performance of the controller. While this can be effectively compensated by the learning mechanism in the feedforward controllers, i.e., the PD-ILC and NN-ILC. For the cases with different amplitudes and different frequencies in the experiments, the NN-ILC can achieve the best tracking performance compared with others in terms of smaller tracking errors and slighter fluctuations.

6.2. Convergence speed

This can be mainly discussed between PD-ILC and NN-ILC. It can be revealed from the experimental observations that: (1) Both the two controllers can converge into steady errors during 10 iterations for all the cases, this demonstrates that the design is proper and the learning mechanisms work well in the iterations. (2) The steady tracking errors are smaller so that the learning mechanism of NN-ILC is superior to that of the PD-ILC in terms of the nonlinearity compensation for this system. (3) Both the two controllers are sensitive to the frequencies of the reference to be tracked, this is mainly due to the same Q-filter adopted in the two control schemes. In addition, this is a common issue to be addressed for all the ILC schemes.

6.3. Challenges and future directions

Future improvements of the proposed NN-ILC can be made from the following two aspects.

(i) **Persistently excited condition**: It is well known that the persistent excitation (PE) condition is the key to guaranteeing the parameter error converges exponentially to zero. If PE is not satisfied, parameter convergence to the true values often cannot be ensured in practice for many adaptive control applications. In this work, the parameters of the controller are tuned manually to satisfy the PE condition. Thus, in future work, improvements can be focused on the selection of centers automatically by making use of historical data to guarantee the PE condition.

(ii) **Fixed-time convergence**: In this work, the controller contains two parts, i.e., the current feedback signal and the feedforward signal calculated by ILC. Therefore, the stability and convergence analysis are also divided into two parts: the boundedness of error signals in the time domain and convergence in the iteration domain. The control system is demonstrated as asymptotically stable according to the Lyapunov theorem of stability. Improvements can also be made to handle fixed-time convergence considering the controller saturation.

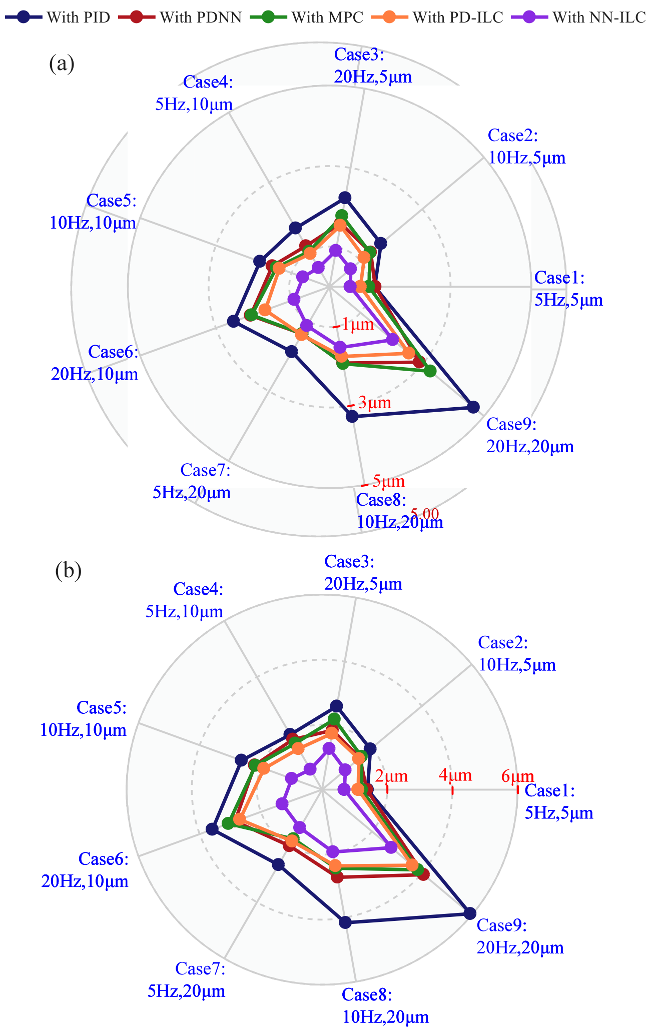

Fig. 9. Statistic RMSEs of experimental cases with different amplitudes and different frequencies for the five controllers. (a) Sinusoidal cases. (b) Triangular cases.

7. Conclusion

In this paper, a novel identification and control scheme named neural network-based iterative learning control (NN-ILC) is proposed for precision motion tracking of repetitive tasks for a piezo-driven nanopositioning stage. To reduce the repetitive errors due to the linear dynamics and invariable disturbance during each iteration, the iterative learning control scheme is built. For the residual non-repetitive errors resulting from unknown nonlinear dynamics and the parameter or model perturbation, the online neural networks are integrated into the ILC scheme. The convergence of the proposed scheme is demonstrated in both the time domain and the iteration domain. The design procedure is given to control practicers. Comprehensive experiments are conducted based on a real-time control testbed. Results show that the proposed NN-ILC is superior to existing controllers (i.e., the traditional PID, the PDNN, the MPC, and the PD-ILC) in terms of the root-mean-square errors of tracking results. Future works will seek to handle fixed-time convergence considering the controller saturation. In addition, the issue of flexible tracking control with ILC for varying references are worthy to be addressed.

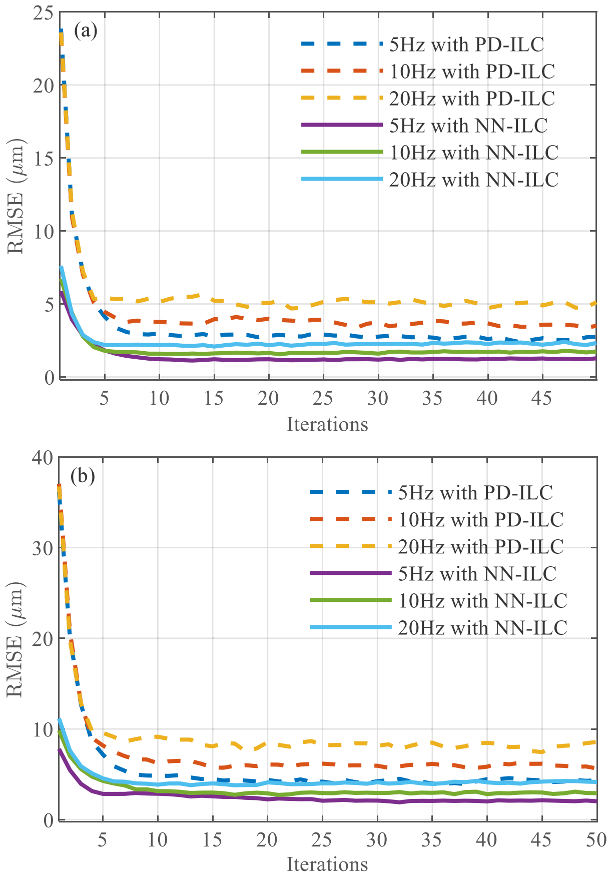

Fig. 10. Iterative process of the RMSE with PD-ILC and NN-ILC under three frequencies. (a) Sinusoidal cases. (b) Triangular cases.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant Number 51975275), the Primary Research & Development Plan of Jiangsu Province (Grant No. BE2021034), the Natural Science Foundation of Jiangsu Province (Grant No. BK20210294), the Fundamental Research Funds for the Central Universities (Grant No. NS2022051), the National Key Laboratory of Science and Technology on Helicopter Transmission (Grant No. HTL-A-22G03) and the Graduate Research and Innovation Projects of Jiangsu Province, China (Grant Number SJCX20_0066).

Appendix A. Convergence proof

(1) Convergence proof in time domain as in (14):

$$\begin{aligned}\dot{S}_j &= \sigma_j \eta \dot{\sigma}_j + \tilde{W}_j^T \Gamma^{-1} \dot{\tilde{W}}_j \\ &= \sigma_j\left(m_n \ddot{x}_d - u_j + W^{*T}\Phi + D + \frac{m_n k_p}{k_d}\dot{e}_j\right) + \tilde{W}_j^T \Gamma^{-1} \dot{\tilde{W}}_j\end{aligned}$$

$$\begin{aligned}
&= \sigma_j \left(m_n \ddot{x}_d - (\sigma_j + \hat{W}_j^T \Phi + m_n \ddot{x}_d + \frac{m_n k_p}{k_d} \dot{e}_j + u_{2,j}) \right. \\
&\quad \left. + W^{*T} \Phi + D + \frac{m_n k_p}{k_d} \dot{e}_j \right) + \tilde{W}_j^T \Gamma^{-1} \dot{\hat{W}}_j \\
&= \sigma_j \left(-\sigma_j - \hat{W}_j^T \Phi - u_{2,j} + W^{*T} \Phi + D \right) + \tilde{W}_j^T \Gamma^{-1} \dot{\hat{W}}_j \\
&= -\sigma_j^2 + \sigma_j (D - u_{2,j}) + \sigma_j \tilde{W}_j^T \Phi - \tilde{W}_j^T \Gamma^{-1} \dot{\hat{W}}_j \\
&= -\sigma_j^2 + \sigma_j \underbrace{(D - u_{2,j})}_{\tilde{u}_{2,j}} + \underbrace{\sigma_j \tilde{W}_j^T \Phi - \tilde{W}_j^T \Gamma^{-1} \Gamma \Phi \sigma_j}_{=0} \\
&= -\sigma_j^2 + \sigma_j \tilde{u}_{2,j} \\
&\leq -\sigma_j^2 + \frac{1}{2} \sigma_j^2 + \frac{1}{2} \tilde{u}_{2,j}^2 \\
&= -\frac{1}{2} \sigma_j^2 + \frac{1}{2} \tilde{u}_{2,j}^2 .
\end{aligned} \tag{32}$$

(2) Convergence proof in iteration domain as in (24):

$$\begin{aligned}
\Delta V_j &= -L(2-L) \int_{\tau=0}^{T} \sigma_j^2 \, d\tau - L \int_{\tau=0}^{T} 2\dot{S} \, d\tau \\
&\quad + L \left(\tilde{W}_{j+1}^T(0) \Gamma^{-1} W_{j+1}(0) - \tilde{W}_j{}^T(0) \Gamma^{-1} \tilde{W}_j(0) \right) \\
&= -L(2-L) \int_{\tau=0}^{T} \sigma_j^2 \, d\tau - L\eta\sigma_j^2(T) \\
&\quad - L \left(\tilde{W}_j^T(T) \Gamma^{-1} \tilde{W}_j(T) - \tilde{W}_j^T(0) \Gamma^{-1} \tilde{W}_j(0) \right) \\
&\quad + L \left(\tilde{W}_{j+1}^T(0) \Gamma^{-1} \tilde{W}_{j+1}(0) - \tilde{W}_j{}^T(0) \Gamma^{-1} \tilde{W}_j(0) \right) \\
&= -L(2-L) \int_{\tau=0}^{T} \sigma_j^2 \, d\tau - L\eta\sigma_j^2(T) < 0.
\end{aligned} \tag{33}$$

Appendix B. Abbreviations

- **PNS:** Piezo-driven nanopositioning stage.
- **AFM:** Atomic force microscopy.
- **ILC:** Iterative learning control.
- **NN-ILC:** Neural network-based iterative learning control.
- **AILC:** Adaptive iterative learning control.
- **ARILC:** Adaptive robust iterative learning control.
- **UAILC:** Unified adaptive iterative learning control.
- **MFNN:** Multi-layer feedforward neural networks.
- **MPC:** Model predictive control.
- **PE:** Persistently excited.
- **RBF:** Radial basis function.
- **PID:** Proportional–integral–derivative.
- **PDNN:** Proportional-derivative control with neural networks.
- **RMSE:** Root-mean-square of error.

References

[1] Chen Z, Zhong X, Shi J, Zhang X. Damping-enabling technologies for broadband control of piezo-stages: A survey. Annu Rev Control 2021;52:120–34.

[2] Boeren F, Bareja A, Kok T, Oomen T. Frequency-domain ILC approach for repeating and varying tasks: With application to semiconductor bonding equipment. IEEE/ASME Trans Mechatronics 2016;21(6):2716–27.

[3] Barton KL, Alleyne AG. A norm optimal approach to time-varying ILC with application to a multi-axis robotic testbed. IEEE Trans Control Syst Technol 2010;19(1):166–80.

[4] Li X, Liu Y, Yu H. Iterative learning impedance control for rehabilitation robots driven by series elastic actuators. Automatica 2018;90:1–7.

[5] Wu S, Tomizuka M. An iterative learning control design for self-servowriting in hard disk drives. Mechatronics 2010;20(1):53–8.

[6] Shen D, Xu J. Adaptive learning control for nonlinear systems with randomly varying iteration lengths. IEEE Trans Neural Netw Learn Syst 2018;30(4):1119–32.

[7] Huang D, Yang W, Huang T, Qin N, Chen Y, Tan Y. Iterative learning operation control of high-speed trains with adhesion dynamics. IEEE Trans Control Syst Technol 2021;29(6):2598–608.

[8] Xu J. A survey on iterative learning control for nonlinear systems. Internat J Control 2011;84(7):1275–94.

[9] Owens D, Hatonen J, Daley S. Robust monotone gradient-based discrete-time iterative learning control. Internat J Robust Nonlinear Control 2009;19(6):634–61.

[10] van Zundert J, Oomen T. On inversion-based approaches for feedforward and ILC. Mechatronics 2018;50:282–91.

[11] Son TD, Pipeleers G, Swevers J. Robust monotonic convergent iterative learning control. IEEE Trans Automat Control 2015;61(4):1063–8.

[12] Janssens P, Pipeleers G, Swevers J. A data-driven constrained norm-optimal iterative learning control framework for LTI systems. IEEE Trans Control Syst Technol 2012;21(2):546–51.

[13] Kim KS, Zou Q. A modeling-free inversion-based iterative feedforward control for precision output tracking of linear time-invariant systems. IEEE/ASME Trans Mechatronics 2012;18(6):1767–77.

[14] Shen D. Iterative learning control with incomplete information: A survey. IEEE/CAA J Autom Sin 2018;5(5):885–901.

[15] French M, Rogers E. Non-linear iterative learning by an adaptive Lyapunov technique. Internat J Control 2000;73(10):840–50.

[16] Xu J, Viswanathan B. Adaptive robust iterative learning control with dead zone scheme. Automatica 2000;36(1):91–9.

[17] Shen D, Xu J. Adaptive learning control for nonlinear systems with randomly varying iteration lengths. IEEE Trans Neural Netw Learn Syst 2018;30(4):1119–32.

[18] Tayebi A, Chien CJ. A unified adaptive iterative learning control framework for uncertain nonlinear systems. IEEE Trans Automat Control 2007;52(10):1907–13.

[19] Wang YC, Chien CJ. Design and analysis of fuzzy-neural discrete adaptive iterative learning control for nonlinear plants. Int J Fuzzy Syst 2013;15(2):149–58.

[20] Li X, Huang D, Chu B, Xu J. Robust iterative learning control for systems with norm-bounded uncertainties. Internat J Robust Nonlinear Control 2016;26(4):697–718.

[21] Cichy B, Hładowski Ł, Gałkowski K, Rauh A, Aschemann H. Iterative learning control of an electrostatic microbridge actuator with polytopic uncertainty models. IEEE Trans Control Syst Technol 2015;23(5):2035–43.

[22] Zeng C, Shen D, Wang J. Adaptive learning tracking for uncertain systems with partial structure information and varying trial lengths. J Franklin Inst B 2018;355(15):7027–55.

[23] Shen D, Xu J. Adaptive learning control for nonlinear systems with randomly varying iteration lengths. IEEE Trans Neural Netw Learn Syst 2018;30(4):1119–32.

[24] Pan Y, Sun T, Liu Y, Yu H. Composite learning from adaptive backstepping neural network control. Neural Netw 2017;95:134–42.

[25] Pan Y, Yu H. Biomimetic hybrid feedback feedforward neural-network learning control. IEEE Trans Neural Netw Learn Syst 2016;28(6):1481–7.

[26] Zheng D, Xie W, Ren X, Na J. Identification and control for singularly perturbed systems using multitime-scale neural networks. IEEE Trans Neural Netw Learn Syst 2016;28(2):321–33.

[27] Guo K, Pan Y, Yu H. Composite learning robot control with friction compensation: a neural network-based approach. IEEE Trans Ind Electron 2018;66(10):7841–51.

[28] Zheng D, Pan Y, Guo K, Yu H. Identification and control of nonlinear systems using neural networks: A singularity-free approach. IEEE Trans Neural Netw Learn Syst 2019;30(9):2696–706.

[29] Ling J, Rakotondrabe M, Feng Z, Ming M, Xiao X. A robust resonant controller for high-speed scanning of nanopositioners: Design and implementation. IEEE Trans Control Syst Technol 2019;28(3):1116–23.

[30] Feng Z, Ling J, Ming M, Liang W, Tan KK, Xiao X. Signal-transformation-based repetitive control of spiral trajectory for piezoelectric nanopositioning stages. IEEE/ASME Trans Mechatronics 2020;25(3):1634–45.

[31] Rana M, Pota HR, Petersen IR. Improvement in the imaging performance of atomic force microscopy: A survey. IEEE Trans Autom Sci Eng 2016;14(2):1265–85.

[32] Sabarianand D, Karthikeyan P, Muthuramalingam T. A review on control strategies for compensation of hysteresis and creep on piezoelectric actuators based micro systems. Mech Syst Signal Process 2020;140:106634.

[33] Gan J, Zhang X. A review of nonlinear hysteresis modeling and control of piezoelectric actuators. AIP Adv 2019;9(4):040702.

[34] Cheng L, Liu W, Hou Z, Yu J, Tan M. Neural-network-based nonlinear model predictive control for piezoelectric actuators. IEEE Trans Ind Electron 2015;62(12):7717–27.

[35] Liu W, Cheng L, Hou Z, Yu J, Tan M. An inversion-free predictive controller for piezoelectric actuators based on a dynamic linearized neural network model. IEEE/ASME Trans Mechatronics 2015;21(1):214–26.

[36] Ling J, Feng Z, Kang X, Xiao X. Bandwidth enhancement in damping control for piezoelectric nanopositioning stages with load uncertainty: Design and implementation. J Vib Control 2021;27(11–12):1382–94.

[37] Al Janaideh M, Al Saaideh M, Rakotondrabe M. On hysteresis modeling of a piezoelectric precise positioning system under variable temperature. Mech Syst Signal Process 2020;145:106880.

[38] Gu G, Zhu L, Su C, Ding H, Fatikow S. Modeling and control of piezo-actuated nanopositioning stages: A survey. IEEE Trans Autom Sci Eng 2014;13(1):313–32.

[39] Xie S, Ren J. Recurrent-neural-network-based predictive control of piezo actuators for trajectory tracking. IEEE/ASME Trans Mechatronics 2019;24(6):2885–96.

[40] Ming M, Ling J, Feng Z, Xiao X. A model prediction control design for inverse multiplicative structure based feedforward hysteresis compensation of a piezo nanopositioning stage. Int J Precis Eng Manuf 2018;19(11):1699–708.

[41] Flores G, Aldana N, Rakotondrabe M. Model predictive control based on the generalized Bouc-Wen model for piezoelectric actuators in robotic hand with only position measurements. IEEE Control Syst Lett 2021;6:2186–91.

[42] Ling J, Feng Z, Zheng D, Yang J, Yu H, Xiao X. Robust adaptive motion tracking of piezoelectric actuated stages using online neural-network-based sliding mode control. Mech Syst Signal Process 2021;150:107235.

[43] Flores G, Rakotondrabe M. Robust nonlinear control for a piezoelectric actuator in a robotic hand using only position measurements. IEEE Control Syst Lett 2021;6:872–7.

[44] Wen Z, Ding Y, Liu P, Ding H. An efficient identification method for dynamic systems with coupled hysteresis and linear dynamics: Application to piezoelectric-actuated nanopositioning stages. IEEE/ASME Trans Mechatronics 2019;24(1):326–37.

[45] Mishra JP, Xu Q, Yu X, Jalili M. Precision position tracking for piezoelectric-driven motion system using continuous third-order sliding mode control. IEEE/ASME Trans Mechatronics 2018;23(4):1521–31.

[46] Xu Q. Precision motion control of piezoelectric nanopositioning stage with chattering-free adaptive sliding mode control. IEEE Trans Autom Sci Eng 2016;14(1):238–48.

[47] Na J, Ren X, Zheng D. Adaptive control for nonlinear pure-feedback systems with high-order sliding mode observer. IEEE Trans Neural Netw Learn Syst 2013;24(3):370–82.

[48] Chen M, Ge SS. Adaptive neural output feedback control of uncertain nonlinear systems with unknown hysteresis using disturbance observer. IEEE Trans Ind Electron 2015;62(12):7706–16.

[49] Li J, Du J, Chen CP. Command-filtered robust adaptive NN control with the prescribed performance for the 3-D trajectory tracking of underactuated AUVs. IEEE Trans Neural Netw Learn Syst 2022;33(11):6545–57.