

Learning Feedforward Control for Industrial Manipulators

Chengyuan Liu, Atanas Popov, Alison Turner, Emma Shires, Svetan Ratchev
Centre for Aerospace Manufacturing, University of Nottingham, Nottingham NG8 1BB, U.K.
E-mail: chengyuan.liu@nottingham.ac.uk

Abstract: In this work, an iterative learning control (ILC) algorithm is proposed for industrial manipulators. The proposed ILC algorithm works coordinately with the inverse dynamics of the manipulator and a feedback controller. The entire control scheme has the ability of compensating both repetitive and non-repetitive disturbances; guaranteeing the control accuracy of the first implementation; and improving the control accuracy of the manipulator progressively with successive iterations. In order to build the convergence of the proposed ILC algorithm, a composite energy function is developed. A case study on a four degree of freedom industrial manipulator is demonstrated to illustrate the effectiveness of the proposed control scheme. By implementing the ILC algorithm, the maximum root mean square error of the control accuracy is improved from 0.0262 *rad* to 0.0016 *rad* within ten iterations.

Key Words: Iterative Learning Control (ILC), Industrial Manipulator, Control Accuracy, Convergence Analysis.

1 Introduction

Industrial robotic manipulators have achieved promising success in smart manufacturing. Comparing with the bespoke gantry-based automation, the industrial manipulators provide extra flexibility and possibility of human-robot collaboration. However, precision control [1] is challenging with industrial manipulators due to both internal and external uncertainties during operations such as drilling, machining, composite fabrication and additive manufacturing. Improving the control accuracy is particularly prominent in high precision manufacturing, e.g. aerospace manufacturing.

The conventional manipulator controller contains two components: feedforward controller and feedback controller. The feedforward controller uses desired robot motion to compute the joint input; the feedback controller takes advantage of both the desired and the measured motion to compute the joint input. The sum of the two joint inputs will be the final control input to the manipulator. Most of the research paper pay attention on the feedback controller due to its ability to compensate modelling errors and disturbances. However, the feedback controller cannot suppress hysteresis effects such as backlash since the effects are fast [3, 4]. This makes the feedforward controller, which performs good in compensating high frequency disturbances, necessary.

The traditional feedforward controller involved in the manipulator usually designed based on the inverse dynamics. Pure inverse dynamics only provide online computation of the joint input based on the desired motion without any compensational function [5]. In order to meet the requirement of compensating hysteresis effects and some other disturbances such as gravity loads and process forces, more advanced control schemes have been designed. In the absence of suitable dynamic model of the industrial manipulator, the data-driven controller, such as neural networks [6–8] and reinforcement learning [9] is often adopted to represent the dynamic effects. While due to the complexity of the nonlinear behaviour of the manipulator, model-based control is preferred when the dynamic model or model approximations are available [10]. On the other hand, uncertain modelling errors have large effect on the control performance. This drives

the development of the adaptive model and partial model-free control algorithms. For example, Grundel and his colleagues designed a model-based controller considering the external cutting forces, which enables the system to cancel the effect of the process forces while operation [11]; Fan et al proposed a fuzzy adaptation algorithm to compensate the uncertain terms in the dynamics [12].

The feedforward control of the industrial manipulator is expected to apply corrections in advance for the manipulator using predicted errors before the error actually occurring. Considering the robotic manipulator with repeating operations, compensations are made based on the previously observed errors, hence the operations become progressively more accurate with success iterations [13]. This repetitive nature in the case of manufacturing makes iterative learning control (ILC) a promising strategy for feedforward control [14–18]. The latest works have shown the superiority of the ILC algorithm as the feedforward controller of the industrial manipulator. Helfrich et al proposed an ILC approach that works together with the H_∞ feedback controller [19] to improve control performance. A dual-loop ILC controller is designed by Meng et al to compensate the vibrations during tracking [20]. The problem of joint hard constraints of the industrial manipulators has been addressed in the work of Yovchev et al through bounded error algorithm for ILC [21]. Angelini et al designed an ILC scheme in combination with low-gain feedback control for improve tracking accuracy [22]. Hofer and his colleagues presented a norm-optimal ILC scheme for articulated soft manipulator to improve the tracking control performance [23]. These existing works using the ILC controller as an alternative of the pure inverse dynamics, which is not sufficiently taking advantage of the dynamic models.

To address the above problem, in this work, we propose an ILC scheme in combination with the pure inverse dynamics to improve tracking accuracy of the industrial manipulators. This structure can guarantee the control accuracy of the first iteration, reduce peak error, and improve the accuracy progressively with successive iterations. At the same time, a feedback controller is adopted to achieve basic control requirement and compensate non-repeating disturbances. Under the coordination work of the ILC, inverse dynamics, and

the feedback controller, both repetitive and non-repetitive disturbances of any frequency can be suppressed. The rigorous convergence analysis is implemented for the proposed control scheme. Finally, a four degree of freedom (DOF) manipulator – OpenManipulator [24] is used for simulation experiment.

The rest of the paper is organised as follows. The dynamic system of the industrial manipulator and the platform used for simulation are present in Section 2. In Section 3, the control scheme is developed and the convergence of the system is proved. In Section 4, simulation test on a four DOF manipulator is carried out in order to show the efficiency of the proposed control scheme. Section 5 gives the conclusion of this work.

2 Platform Overview

The following form of dynamics of a n DOF manipulator is considered:

$$M(q_k(t))\ddot{q}_k(t) + C(q_k(t), \dot{q}_k(t))\dot{q}_k(t) + G(q_k(t)) = \tau_k(t), \quad (1)$$

where $t \in \mathbb{R}$ denotes the time instant and $k \in \mathbb{Z}^+$ denotes the number of iteration. $q_k(t) \in \mathbb{R}^n$, $\dot{q}_k(t) \in \mathbb{R}^n$ and $\ddot{q}_k(t) \in \mathbb{R}^n$ are the joint position, velocity and acceleration vectors, respectively, at time t of the k th iteration. $\tau_k(t)$ is the vector of torques, which is the system control input. $M(q_k(t)) \in \mathbb{R}^{n \times n}$ is the inertia matrix, which is symmetric, bounded and positive definite, $C(q_k(t), \dot{q}_k(t)) \in \mathbb{R}^n$ represents the Coriolis and centrifugal forces, $G(q_k(t)) \in \mathbb{R}^n$ represents the gravitational and frictional forces. For robotic manipulator, the following properties are hold:

P1 $M(q)$ is positive symmetric and bounded such that there exist constants $\nu_1 > 0$ and $\nu_2 > 0$ to achieve $0 \leq \nu_1 I_n \leq M(q) \leq \nu_2 I_n$ for any $q \in \mathbb{R}^n$, where I_n denotes the identity matrix of dimension n .

P2 The matrix $(\frac{1}{2}\dot{M}(q_k(t)) - C(q_k(t), \dot{q}_k(t)))$ is skew symmetric such that

$$v^T (\frac{1}{2}\dot{M}(q_k(t)) - C(q_k(t), \dot{q}_k(t)))v = 0, \forall v \in \mathbb{R}^n. \quad (2)$$

In order to testing the efficiency of the proposed control scheme, the 4 DOF ROBOTIS OpenManipulator platform [24] is adopted. The unified robot description format (URDF) file of this robotic manipulator is imported into the MATLAB robotics system toolbox, and the robot controller model is built in Simulink. The home position of the manipulator is shown in Fig. 1 and the dynamic parameters are listed in Table. 1.

3 Controller Design

In contrast to the existing research, the proposed ILC feedforward controller is designed in combination with the pure inverse dynamics. The structure of the control diagram is shown in Fig. 2. The torque input τ_k to the robotic manipulator consists of three components: the learning feedforward control $\tau_k^{ILC}(t)$, the inverse dynamics feedforward control $\tau_k^{ID}(t)$, and a PD-type feedback control $\tau_k^{PD}(t)$, i.e.

$$\tau_k(t) = \tau_k^{ILC}(t) + \tau_k^{ID}(t) + \tau_k^{PD}(t). \quad (3)$$

Suppose $q_d(t)$ is a bounded desired trajectory and the system satisfy the following assumptions:

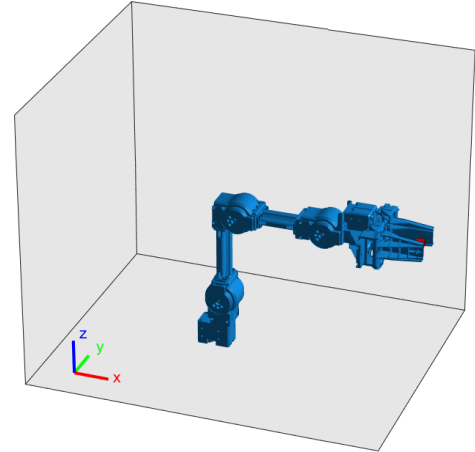


Fig. 1: The 4 DOF ROBOTIS OpenManipulator-X

Table 1: Manipulator parameters

Parameters	Link 1
Length (mm)	77
Mass (g)	98.4
Inertia ($Kg \cdot m^2$)	$\begin{bmatrix} 2.56e-4 & 2.49e-6 & -1.03e-6 \\ 2.49e-6 & 2.54e-4 & -2.20e-11 \\ -1.03e-6 & -2.20e-11 & 1.89e-6 \end{bmatrix}$
Parameters	Link 2
Length (mm)	128
Mass (g)	138.5
Inertia ($Kg \cdot m^2$)	$\begin{bmatrix} 1.76e-3 & 6.89e-6 & 1.84e-4 \\ 6.89e-6 & 1.79e-3 & 6.37e-7 \\ 1.84e-4 & 6.37e-7 & 7.51e-5 \end{bmatrix}$
Parameters	Link 3
Length (mm)	24
Mass (g)	132.7
Inertia ($Kg \cdot m^2$)	$\begin{bmatrix} 3.07e-5 & 2.31e-11 & 2.97e-6 \\ 2.31e-11 & 1.34e-3 & 5.97e-6 \\ 2.97e-6 & 5.97e-6 & 1.35e-3 \end{bmatrix}$
Parameters	Link 4
Length (mm)	124
Mass (g)	143.3
Inertia ($Kg \cdot m^2$)	$\begin{bmatrix} 9.23e-5 & 4.71e-10 & 5.75e-5 \\ 4.71e-10 & 3.67e-4 & 2.33e-9 \\ 5.75e-5 & 2.33e-9 & 3.73e-4 \end{bmatrix}$

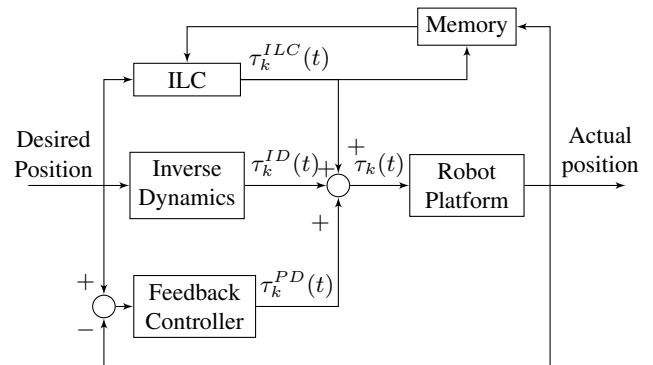


Fig. 2: Control Structure for Industry Manipulator

- A1 The desired trajectory $q_d(t)$ is first and second order continuously differentiable for all $t \in [0, T]$; $q_d(t)$ and its first and second time derivative $\dot{q}_d(t)$, $\ddot{q}_d(t)$ are bounded for all $t \in [0, T]$.
- A2 The initial conditions $q_k(0) = q_d(0)$ and $\dot{q}_k(0) = \dot{q}_d(0)$ are satisfied for all $k \in \mathbb{Z}^+$.

First, the inverse dynamics is build based on system (1) and A1, which gives

$$\tau_k^{ID}(t) = M(q_d(t))\ddot{q}_d(t) + C(q_d(t), \dot{q}_d(t))\dot{q}_d(t) + G(q_d(t)), \quad (4)$$

where $q_d(t)$ is the bounded desired trajectory, $\dot{q}_d(t)$, and $\ddot{q}_d(t)$ are the corresponding desired joint velocity and acceleration. A low pass filter is used to generate the joint velocities from the joint position. Note that the inverse dynamics is identical for each iteration since the desired trajectory is identical. Hence the inverse dynamics has no mutual influence with the other controller component.

Then the feedback controller is designed to ensure the stability of the closed loop system. In this work, four PD controllers work for each motor are utilised, that is

$$\tau_k^{PD}(t) = K_p e_k(t) + K_d \dot{e}_k(t), \quad (5)$$

where

$$e_k(t) = q_d(t) - q_k(t), \quad (6)$$

$$\dot{e}_k(t) = \dot{q}_d(t) - \dot{q}_k(t), \quad (7)$$

are the tracking error and its derivative at time t of the k th iteration, respectively. $K_p \in \mathbb{R}^{n \times n}$ and $K_d \in \mathbb{R}^{n \times n}$ are diagonal positive definite and are proportional and derivative control gain matrix, respectively.

After tuned the feedback controller to achieve a best performance, the ILC algorithm can be designed. It is worth to note that the feedback controller and the ILC algorithm could influence each other. In the work of Tayebi [25], vibrations may occur in some of or all manipulator joints as the number of iterations increases. This is due to the accumulation of the noise caused by derivative action through the iterations [26]. In order to decrease this effect, the ILC controller is proposed in the following structure:

$$\tau_k^{ILC}(t) = \tau_{k-1}^{ILC}(t) + \gamma \tau_{k-1}^{FB}(t), \quad (8)$$

where $\gamma > 0 \in \mathbb{R}$ is a learning factor, $\tau_{-1}^{ILC} = 0$.

With the above controllers and assumptions, we have the following result.

Theorem 1. Consider the robotic manipulator (1) satisfy the assumption (A1-A2), using the control law (3)-(5) and (8), then for all $t \in [0, T]$,

$$\lim_{k \rightarrow \infty} q_k(t) = q_d(t), \quad (9)$$

$$\lim_{k \rightarrow \infty} \dot{q}_k(t) = \dot{q}_d(t), \quad (10)$$

hold.

Proof. In order to facilitate the convergence analysis in Theorem 1, two fictitious signals are introduced:

$$\bar{\tau}_d := M(q_k)M^{-1}(q_d)\tau_d, \quad (11)$$

$$\xi_k := \dot{e}_k + K_d^{-1}K_p e_k, \quad (12)$$

where τ_d is the desired control input. The time instant indicator (t) is omitted for clarity in here and the following proof. Due to the property P1 and the assumptions A1, A2, we have $\bar{\tau}_d$ continuous and bounded; $\bar{\tau}_d$ approaches the actual desired control input τ_d and e_k goes to 0, when q_k approaches the desired trajectory q_d and ξ_k goes to 0. From system (1), we have

$$\ddot{q}_k = f(q_k, \dot{q}_k) + M^{-1}(q_k)\tau_k, \quad (13)$$

where

$$f(q_k, \dot{q}_k) = -M^{-1}(q_k)(C(q_k, \dot{q}_k)\dot{q}_k + G(q_k)). \quad (14)$$

Let

$$\tilde{\tau}_k := \bar{\tau}_d - \tau_k, \quad (15)$$

$$\tilde{\tau}_k^{ILC} := \bar{\tau}_d - \tau_k^{ID} - \tau_k^{ILC}, \quad (16)$$

$$\tilde{f}(q_k, \dot{q}_k) := f(q_d, \dot{q}_d) - f(q_k, \dot{q}_k), \quad (17)$$

and define the Lyapunov-like composite energy function (CEF):

$$J_k(t) = \int_0^t e^{-\lambda u} (\tilde{\tau}_k^{ILC})^T (\tilde{\tau}_k^{ILC}) du, \quad (18)$$

where $\lambda \in \mathbb{R}$ is a positive constant.

The remaining proof will firstly demonstrate the boundedness and non-increasing property of the CEF $J_k(t)$ with respect to k , then show the convergence (9) and (10).

The difference of the CEF at a time instant $t \in [0, T]$ of two consequent iterations is

$$\begin{aligned} \Delta J_k &= J_k - J_{k-1} \\ &= \int_0^t e^{-\lambda u} (|\tilde{\tau}_k^{ILC}|^2 - |\tilde{\tau}_{k-1}^{ILC}|^2) du. \end{aligned} \quad (19)$$

From the definition in (5), (8), (12) and (16) we have

$$\tilde{\tau}_k^{ILC} = \tilde{\tau}_{k-1}^{ILC} - \gamma K_d \xi_{k-1}. \quad (20)$$

Substituting this into (19) yields

$$\begin{aligned} \Delta J_k &= \int_0^t e^{-\lambda u} (|\tilde{\tau}_{k-1}^{ILC} - \gamma K_d \xi_{k-1}|^2 - |\tilde{\tau}_{k-1}^{ILC}|^2) du \\ &= \int_0^t e^{-\lambda u} (|\gamma K_d \xi_{k-1}|^2) du \\ &\quad - 2 \int_0^t e^{-\lambda u} (\gamma K_d \xi_{k-1})^T (\tilde{\tau}_{k-1}^{ILC}) du. \end{aligned} \quad (21)$$

Utilising the definition of (11), (13) and (15) - (17), we can derive the the following result:

$$\begin{aligned} M(q_{k-1})\dot{\xi}_{k-1} &= M(q_{k-1})\ddot{e}_{k-1} + M(q_{k-1})K_d^{-1}K_p \dot{e}_{k-1} \\ &= M(q_{k-1})\tilde{f}(q_{k-1}, \dot{q}_{k-1}) + \tilde{\tau}_{k-1} + M(q_{k-1})K_d^{-1}K_p \dot{e}_{k-1} \\ &= \tilde{\tau}_{k-1} - C(q_{k-1}, \dot{q}_{k-1})\xi_{k-1} + \eta_{k-1} \\ &= \tilde{\tau}_{k-1}^{ILC} - \tilde{\tau}_{k-1}^{FB} - C(q_{k-1}, \dot{q}_{k-1})\xi_{k-1} + \eta_{k-1}, \end{aligned} \quad (22)$$

where

$$\begin{aligned} \eta_{k-1} &= C(q_{k-1}, \dot{q}_{k-1})\xi_{k-1} + M(q_{k-1})\tilde{f}(q_{k-1}, \dot{q}_{k-1}) \\ &\quad + M(q_{k-1})K_d^{-1}K_p \dot{e}_{k-1}. \end{aligned} \quad (23)$$

Hence, we have

$$\tilde{\tau}_{k-1}^{ILC} = M(q_{k-1})\dot{\xi}_{k-1} + \tilde{\tau}_{k-1}^{FB} + C(q_{k-1}, \dot{q}_{k-1})\xi_{k-1} - \eta_{k-1}. \quad (24)$$

Substituting this into (21), we have

$$\begin{aligned} \Delta J_k &= \int_0^t e^{-\lambda t} (\gamma K_d \xi_{k-1})^T (\gamma K_d \xi_{k-1} + 2\eta_{k-1}) dt \\ &\quad - 2 \int_0^t e^{-\lambda t} (\gamma K_d \xi_{k-1})^T (M(q_{k-1})\dot{\xi}_{k-1}) dt \\ &\quad - 2 \int_0^t e^{-\lambda t} (\gamma K_d \xi_{k-1})^T (\tilde{\tau}_{k-1}^{FB}) dt \\ &\quad - 2 \int_0^t e^{-\lambda t} (\gamma K_d \xi_{k-1})^T (C(q_{k-1}, \dot{q}_{k-1})\xi_{k-1}) dt, \end{aligned} \quad (25)$$

where $\tau_{k-1}^{FB} = K_d \xi_{k-1}$, and

$$\begin{aligned} &- 2e^{-\lambda t} (\gamma K_d \xi_{k-1})^T (M(q_{k-1})\dot{\xi}_{k-1}) \\ &= - \frac{d}{dt} (e^{-\lambda t} (\gamma K_d \xi_{k-1})^T (M(q_{k-1})\xi_{k-1})) \\ &\quad + e^{-\lambda t} (\gamma K_d \xi_{k-1})^T (\dot{M}(q_{k-1})\xi_{k-1}) \\ &\quad - \lambda e^{-\lambda t} (\gamma K_d \xi_{k-1})^T (M(q_{k-1})\xi_{k-1}). \end{aligned} \quad (26)$$

Substituting the above equation into (25) and considering property P2, we have

$$\begin{aligned} \Delta J_k &= - e^{-\lambda t} (\gamma K_d \xi_{k-1})^T (M(q_{k-1})\xi_{k-1}) \\ &\quad - \lambda \int_0^t e^{-\lambda t} (\gamma K_d \xi_{k-1})^T (M(q_{k-1})\xi_{k-1}) dt \\ &\quad - 2 \int_0^t e^{-\lambda t} (\gamma K_d \xi_{k-1})^T (K_d \xi_{k-1}) dt \\ &\quad + \int_0^t e^{-\lambda t} (\gamma K_d \xi_{k-1})^T (\gamma K_d \xi_{k-1} + 2\eta_{k-1}) dt. \end{aligned} \quad (27)$$

Based on (12) and (23), we have constants $a_1 > 0$ and $a_2 > 0$ [27] such that

$$|\eta_{k-1}| \leq a_1 |\xi_{k-1}| + a_2 |\xi_{k-1}|^2. \quad (28)$$

The bounded property P1 gives a constant $\nu > 0 \in \mathbb{R}$ such that

$$0 \leq \nu I_n \leq K_d^T M(q_{k-1}). \quad (29)$$

Hence we have

$$\begin{aligned} \Delta J_k &\leq -\nu \gamma e^{-\lambda t} |\xi_{k-1}|^2 - (\lambda + 2) \gamma \int_0^t e^{-\lambda t} |\xi_{k-1}|^2 dt \\ &\quad + \gamma \int_0^t e^{-\lambda t} h(|\xi_{k-1}|) |\xi_{k-1}|^2 dt, \end{aligned} \quad (30)$$

where $h(|\xi_{k-1}|) = |K_d|(\gamma |K_d| + 2a_1 + 2a_2 |\xi_{k-1}|)$. It can be shown that there exist a positive constant $a_3 > 0$ such that

$$0 < h(|\xi_{k-1}|) < a_3, \quad (31)$$

we have

$$\begin{aligned} \Delta J_k &\leq -(\lambda + 2 - a_3) \gamma \int_0^t e^{-\lambda t} |\xi_{k-1}|^2 dt \\ &\quad - \nu \gamma e^{-\lambda t} |\xi_{k-1}|^2. \end{aligned} \quad (32)$$

It follows that if λ is selected such that $\lambda + 2 - a_3 > 0$, then $\Delta J_k \leq 0$, which shows the non-increasing property of the CEF J_k .

Implementing a cumulative sum for (32) shows the convergence of the tracking error since J_{-1} is bounded and J_k is positive, which also guarantees the convergence of the control input in the iteration axis. Furthermore, as (22) is uniformly bounded, the uniform continuity of ξ_k is guaranteed in the time axis $[0, T]$. Hence the tracking error $e_k(t)$ is uniformly convergent, and $\lim_{k \rightarrow \infty} q_k(t) = q_d(t)$ and $\lim_{k \rightarrow \infty} \dot{q}_k(t) = \dot{q}_d(t)$ hold. \square

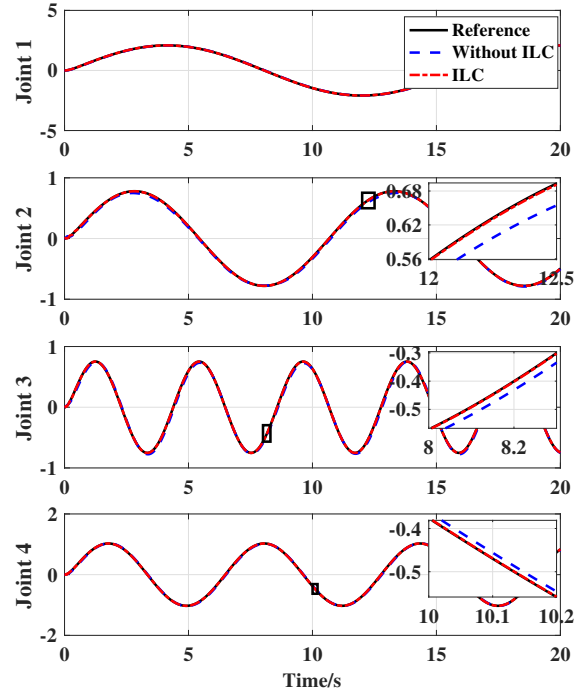


Fig. 3: Tracking performance in the four joints. The black solid line indicates the desired trajectory, the blue dashed line indicates the tracking path without ILC algorithm, the red dash-dotted line indicates the tracking path after ten iterations using ILC.

4 Simulation Test

Consider the manipulator platform described in Section 2, the desired trajectories for the four joints are chosen as

$$q_d(t) = \begin{bmatrix} \frac{2}{3}\pi \sin(0.4t) \\ \frac{\pi}{4} \sin(0.6t) \\ \frac{\pi}{4} \sin(1.5t) \\ \frac{\pi}{3} \sin(t) \end{bmatrix}. \quad (33)$$

The proposed control law (3)-(5) and (8) are applied with $K_p = 8I_4$, $K_d = 0.1I_4$, and $\gamma = 0.3$. The tracking performance of before and after utilising the ILC method have

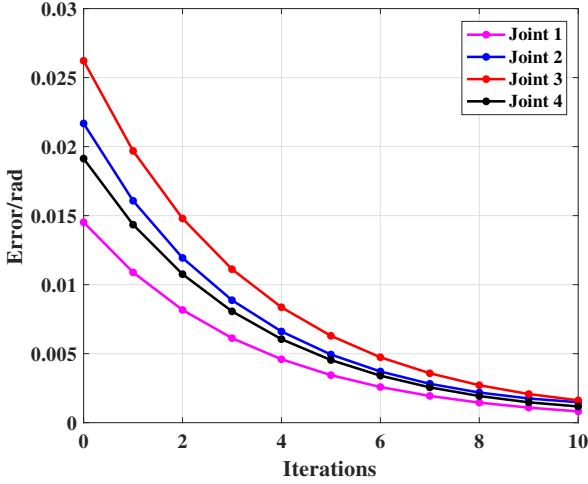


Fig. 4: Tracking performance in root mean square errors versus the number of iterations in the four joints. The performance of joint 1 – 4 are shown in pink, blue, red, and black line, respectively.

Table 2: Before iteration versus after 10 iterations tracking performance in root mean square errors of the joint position (rad).

Joint	Before iteration	After 10 iterations
Joint 1	0.0145	0.0008
Joint 2	0.0217	0.0015
Joint 3	0.0262	0.0016
Joint 4	0.0191	0.0012

been shown in Fig 3. The desired trajectories are shown in black solid line, the tracking trajectories are shown in blue dashed line, the tracking trajectories after ten iterations are shown in red dash-dotted line. Three random portions of the tracking performance of joint 2 – 4 are enlarged in the plot to show more details of the tracking difference. The root mean square errors versus the number of iterations for the four joints are shown in Fig. 4. The root mean square errors of before iteration and after 10 iterations are shown in Table 2.

The simulation results shows that the proposed ILC scheme can improve the tracking accuracy with the increase of the iteration number. However, due to the noise accumulation problem described in Section 3, the learning process must be stopped once the tracking error meet the requirement to prevent any possible joint oscillations. As addressed by Tayebi et al [26], trade off between the convergence rate and the joints chattering phenomenon is considered by tuning the learning gain γ . Furthermore, the cut-off frequency of the filter used to generate the joint velocity also has large effect on the convergence property.

5 Conclusion

The control scheme designed in this work contains an ILC feedforward controller, an inverse dynamics feedforward controller, and a PD feedback controller. The coordination work of these three control method makes the compensation for both repetitive and non-repetitive disturbances of any frequency possible. The control accuracy of the indus-

trial manipulator at the first implementation is guaranteed by taking advantage of the inverse dynamics. The PD feedback controller provides compensation of non-repeating and low frequency disturbances. The ILC algorithm is used to suppress the repetitive and high frequency disturbances. The control accuracy is improved progressively with the successive iterations. Both the feedback and ILC algorithm contribute to the convergence of the control scheme, which is proved using CEF method. The simulation on the OpenManipulator shows promising results in the tracking accuracy.

References

- [1] D. A. Bristow and A. G. Alleyne, "A high precision motion control system with application to microscale robotic deposition," *IEEE Transactions on Control Systems Technology*, vol. 14, no. 6, pp. 1008–1020, 2006.
- [2] S. Devasia, E. Eleftheriou, and S. R. Moheimani, "A survey of control issues in nanopositioning," *IEEE Transactions on Control Systems Technology*, vol. 15, no. 5, pp. 802–823, 2007.
- [3] M. Goldfarb and N. Celanovic, "Modeling piezoelectric stack actuators for control of micromanipulation," *IEEE Control Systems Magazine*, vol. 17, no. 3, pp. 69–79, 1997.
- [4] S. Yamada and H. Fujimoto, "Precise joint torque control method for two-inertia system with backlash using load-side encoder," *IEEJ Journal of Industry Applications*, vol. 8, no. 1, pp. 75–83, 2019.
- [5] L. Sciavicco and B. Siciliano, *Modelling and control of robot manipulators*. Springer Science & Business Media, 2012.
- [6] D. Braganza, D. M. Dawson, I. D. Walker, and N. Nath, "A neural network controller for continuum robots," *IEEE transactions on robotics*, vol. 23, no. 6, pp. 1270–1277, 2007.
- [7] N. Kumar, V. Panwar, N. Sukavanam, S. P. Sharma, and J.-H. Borm, "Neural network based hybrid force/position control for robot manipulators," *International Journal of Precision Engineering and Manufacturing*, vol. 12, no. 3, pp. 419–426, 2011.
- [8] M. T. Gillespie, C. M. Best, E. C. Townsend, D. Wingate, and M. D. Killpack, "Learning nonlinear dynamic models of soft robots for model predictive control with neural networks," in *2018 IEEE International Conference on Soft Robotics (RoboSoft)*, pp. 39–45. IEEE, 2018.
- [9] Y. Ansari, M. Manti, E. Falotico, M. Cianchetti, and C. Laschi, "Multiobjective optimization for stiffness and position control in a soft robot arm module," *IEEE Robotics and Automation Letters*, vol. 3, no. 1, pp. 108–115, 2017.
- [10] V. Falkenhahn, A. Hildebrandt, R. Neumann, and O. Sawodny, "Model-based feedforward position control of constant curvature continuum robots using feedback linearization," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 762–767. IEEE, 2015.
- [11] L. Gründel, L. Lienenlücke, S. Storms, and C. Brecher, "Compensation of process forces with a model-based feed-forward control for robot machining," in *2019 4th International Conference on Control and Robotics Engineering (ICCRE)*, pp. 115–121. IEEE, 2019.
- [12] Y. Fan, K. Xing, and X. Jiang, "Fuzzy adaptation algorithms' control for robot manipulators with uncertainty modelling errors," *Complexity*, 2018.
- [13] C. Atkeson and J. McIntyre, "An application of adaptive feedforward control to robotics," in *Adaptive Systems in Control and Signal Processing*, pp. 283–288. Elsevier, 1987.
- [14] S. Arimoto, S. Kawamura, and F. Miyazaki, "Bettering opera-

- tion of robots by learning,” *Journal of Robotic systems*, vol. 1, no. 2, pp. 123–140, 1984.
- [15] M. Norrlof, “An adaptive iterative learning control algorithm with experiments on an industrial robot,” *IEEE Transactions on robotics and automation*, vol. 18, no. 2, pp. 245–251, 2002.
- [16] D. A. Bristow, M. Tharayil, and A. G. Alleyne, “A survey of iterative learning control,” *IEEE control systems magazine*, vol. 26, no. 3, pp. 96–114, 2006.
- [17] K. L. Moore, *Iterative learning control for deterministic systems*. Springer Science & Business Media, 2012.
- [18] Z. Bien and J.-X. Xu, *Iterative learning control: analysis, design, integration and applications*. Springer Science & Business Media, 2012.
- [19] B. Helfrich, C. Lee, D. Bristow, X. Xiao, J. Dong, A. G. Alleyne, S. M. Salapaka, and P. M. Ferreira, “Combined h_∞ -feedback and iterative learning control design with application to nanopositioning systems,” in *2008 American Control Conference*, pp. 3893–3900. IEEE, 2008.
- [20] T. Meng and W. He, “Iterative learning control of a robotic arm experiment platform with input constraint,” *IEEE Transactions on Industrial Electronics*, vol. 65, no. 1, pp. 664–672, 2017.
- [21] K. Yovchev, K. Delchev, and E. Krastev, “State space constrained iterative learning control for robotic manipulators,” *Asian Journal of Control*, vol. 20, no. 3, pp. 1145–1150, 2018.
- [22] F. Angelini, C. Della Santina, M. Garabini, M. Bianchi, G. M. Gasparri, G. Grioli, M. G. Catalano, and A. Bicchi, “Decentralized trajectory tracking control for soft robots interacting with the environment,” *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 924–935, 2018.
- [23] M. Hofer, L. Spannagl, and R. D’Andrea, “Iterative learning control for fast and accurate position tracking with an articulated soft robotic arm,” *arXiv preprint arXiv:1901.10187*, 2019.
- [24] “Openmanipulator: Openmanipulator - x.” [Online]. Available: <https://manual.robotis.com/>
- [25] A. Tayebi, “Adaptive iterative learning control for robot manipulators,” *Automatica*, vol. 40, no. 7, pp. 1195–1203, 2004.
- [26] A. Tayebi and S. Islam, “Adaptive iterative learning control for robot manipulators: Experimental results,” *Control Engineering Practice*, vol. 14, no. 7, pp. 843–851, 2006.
- [27] G. Sebastian, Z. Li, Y. Tan, and D. Oetomo, “On implementation of feedback-based pd-type iterative learning control for robotic manipulators with hard input constraints,” in *2019 IEEE 15th International Conference on Control and Automation (ICCA)*, pp. 43–48. IEEE, 2019.