TU Delft

Delft University of Technology

Document Version
Final published version

Licence
Dutch Copyright Act (Article 25fa)

Citation (APA)
Lee, D., Haren, M., Kon, J., Oomen, T., & Oh, S. (2026). Dynamics-Aware Iterative Learning Control for Fast and Precise Motion of Multi-Joint Robots. In *Proceedings of the IEEE 19th International Conference on Advanced Motion Control (AMC 2026)* IEEE. https://doi.org/10.1109/AMC67705.2026.11435834

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Copyright
In case the licence states “Dutch Copyright Act (Article 25fa)”, this publication was made available Green Open Access via the TU Delft Institutional Repository pursuant to Dutch Copyright Act (Article 25fa, the Taverne amendment). This provision does not affect copyright ownership.
Unless copyright is transferred by contract or statute, it remains with the copyright holder.

Sharing and reuse
Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy
Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.

This work is downloaded from Delft University of Technology.

Dynamics-Aware Iterative Learning Control for Fast and Precise Motion of Multi-Joint Robots

Deokjin Lee, Max Van Haren, Johan Kon, Tom Oomen, and Sehoon Oh

2026 IEEE 19th International Conference on Advanced Motion Control (AMC) | 979-8-3315-7049-1/26/$31.00 ©2026 IEEE | DOI: 10.1109/AMC67705.2026.11435834

***Abstract*—This paper presents a novel dynamics-aware iterative learning control (DAILC) for linear parameter-varying (LPV) robotic systems performing dynamic, non-repetitive tasks. While conventional basis function iterative learning control (BFILC) offers advantages for high precision control in non-repetitive operations, its performance degrades in the presence of dynamic variations. To overcome this limitation, the proposed method embeds physically consistent basis functions derived from LPV multi-body dynamics, enabling enhanced robustness to system variations and inherent nonlinearities while maintaining task flexibility. The effectiveness of the DAILC approach is validated through simulations on an LPV robotic manipulator, demonstrating superior tracking performance and disturbance tolerance compared with existing state-of-the-art control strategies.**

I. INTRODUCTION

The control of multi-joint robots has received increasing attention due to rapid advances in robotic technologies and their broad application in diverse domains [1]. Robotic platforms such as humanoids, manipulators, and quadrupeds are deployed in tasks ranging from repetitive industrial operations to adaptive interaction in human-centric environments. These include roles such as automated manufacturing [2], remote inspection [3], and collaborative human-robot workspaces [4].

Despite such versatility, multi-joint robots continue to face significant control challenges. These challenges include joint coupling effects [5], variations in system dynamics [6], and flexibility at the joints [7], all of which can degrade control accuracy and bandwidth. These issues ultimately result in limited motion speed and reduced performance.

To address these limitations, robust feedback control methodologies have been extensively investigated, including disturbance observers [8], H-infinity control [9], and sliding mode control [10]. Although these approaches effectively enhance disturbance rejection and robustness, they inherently encounter the waterbed effect [11], a fundamental trade-off wherein improving system performance within a specific frequency range inevitably deteriorates performance in other frequency bands. This inherent limitation motivates incorporating feedforward control techniques, which can effectively enhance tracking performance without encountering the constraints associated with the waterbed effect.

Feedforward control is widely adopted in robotic systems to overcome this limitation. Model-based feedforward approaches [12] incorporate nonlinear dynamic models to generate effective feedforward actions for multi-joint robots. However, these methods require highly accurate models, which are difficult to obtain due to unmodeled nonlinearities, friction, and parameter uncertainties. Although system identification and CAD-based modeling [13] provide partial solutions, they often fail to capture complex physical behaviors, particularly in high-DOF systems.

Iterative Learning Control (ILC) [14] offers a promising solution by learning feedforward inputs through repeated task executions without requiring explicit dynamic models. By updating control signals based on previous tracking errors, ILC has demonstrated strong performance in high-precision applications involving repetitive tasks [15]. However, conventional ILC methods depend on fixed reference trajectories and therefore struggle to generalize to non-repetitive or varying tasks.

Basis Function ILC (BFILC) [16] addresses this limitation by employing basis functions that enable generalization across changing reference trajectories. Its robustness and high accuracy have been demonstrated in applications such as printing [17], gantry-stage systems [18], and robotic platforms [19]. Nevertheless, BFILC remains limited in capturing the dynamics of linear parameter-varying (LPV) systems, as it relies on signal-oriented basis functions that do not explicitly reflect underlying system dynamics.

To overcome these challenges, this work proposes a dynamics-aware ILC (DAILC) that explicitly incorporates LPV robotic dynamics into the construction of basis functions. By leveraging inertial parameter models of multi-body robotic systems, the proposed method produces dynamics-aware basis functions that enhance control accuracy in both repetitive and non-repetitive scenarios. Additionally, the learning framework is formulated to ensure parameter consistency and dynamic relevance throughout the adaptation process.

The main contributions of this work are summarized as follows:

- A DAILC framework is developed for LPV multi-body robotic systems, enabling robust performance under varying task conditions.
- A physically consistent parameter update law is formulated to accurately model the LPV robot dynamics using dynamics-aware basis functions.

This work was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education No. RS-2025-25420118.

D. Lee, and S. Oh are with Department of Robotics Engineering, DGIST, Daegu, Korea 42988. (e-mail : {djlee, sehoon}@dgist.ac.kr) (phone : +82-53-785-6209)

Max van Haren, Johan Kon, and Tom Oomen are with the Control Systems Technology Section, Department of Mechanical Engineering, Eindhoven University of Technology, 5600 MB Eindhoven, The Netherlands.

Tom Oomen is with the Delft Center for Systems and Control, Delft University of Technology, 2628 CD Delft, The Netherlands

Authorized licensed use limited to: TU Delft Library. Downloaded on October 08,2026 at 07:56:05 UTC from IEEE Xplore. Restrictions apply.

- The effectiveness of the proposed approach is validated through simulations on a 3-DOF robotic arm, showing improved tracking accuracy, accelerated convergence, and enhanced generalization in the presence of model uncertainties.

The remainder of the paper is organized as follows: Section II presents the problem formulation. Section III introduces the proposed basis function design and learning framework. Section IV provides simulation-based validation. Conclusions and directions for future work are discussed in Section V.

II. Problem Formulation

A. LPV Robotic System

The dynamics of multi-body robotic systems in joint space are generally expressed as

$$\mathcal{M}(q)\ddot{q} + \mathcal{C}(q,\dot{q})\dot{q} + \mathcal{G}(q) = \tau, \tag{1}$$

where $q, \dot{q}, \ddot{q} \in \mathbb{R}^N$ represent joint positions, velocities, and accelerations, respectively, and $\tau \in \mathbb{R}^N$ denotes applied joint torques. Matrices $\mathcal{M}, \mathcal{C} \in \mathbb{R}^{N\times N}$ are inertia and Coriolis/centrifugal matrices, respectively, and $\mathcal{G} \in \mathbb{R}^N$ is the gravity vector.

For analysis and controller design, these nonlinear dynamics are represented within a continuous-time linear parameter-varying (LPV) framework. The corresponding LPV state-space representation [20] is given by

$$P_0 : \begin{cases} \dot{x}(t) = A_l(\rho(t))x(t) + B_l(\rho(t))u(t) \\ y(t) = C_l(\rho(t))x(t) + D_l(\rho(t))u(t) \end{cases} \tag{2}$$

where the state vector is defined as $x^T = [q, \dot{q}]$, input $u = \tau$, output $y = q$, and ρ is the scheduling parameter vector that captures system variations.

When $\rho(t)$ remains constant, the LPV model reduces to a linear time-invariant (LTI) system. Applying the Laplace transform under zero initial conditions yields

$$\hat{y}(s) = \left[C_l(\rho)(sI - A_l(\rho))^{-1}B_l(\rho) + D_l(\rho)\right]\hat{u}(s), \tag{3}$$

resulting in the corresponding transfer function:

$$P_0(s,\rho) = C_l(\rho)(sI - A_l(\rho))^{-1}B_l(\rho) + D_l(\rho). \tag{4}$$

B. Conventional BFILC for LPV Robot and Its Limitation

1) Formulation of BFILC: Consider the closed-loop robotic control system employing both feedback and feedforward mechanisms, as illustrated in Fig. 1. The objective is to design the feedforward signal f_j to ensure the output y accurately tracks the reference trajectory r, where j denotes the trial index. The tracking error at trial j is

$$e_j = S_0 r - S_0 P_0 f_j, \tag{5}$$

where $S_0 = (I + KP_0)^{-1}$ is the sensitivity function associated with the feedback controller K, and P_0 represents the LPV plant model. The tracking error at trial $j+1$ is given by

$$e_{j+1} = S_o r - S_0 P_0 f_{j+1}. \tag{6}$$

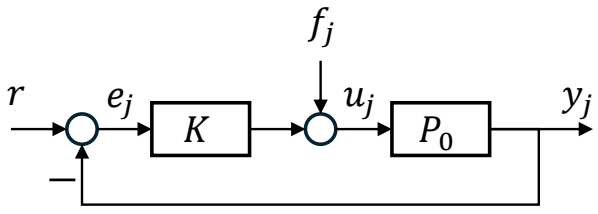


Fig. 1: Control Schematic

To promote flexibility and generalization across different motion tasks, the feedforward input f_j is parameterized as

$$f_j = \Psi(r)\theta_j, \tag{7}$$

where $\Psi(r)$ is a matrix of basis functions constructed from the reference trajectory and θ_j contains the parameters to be learned. This representation enables the feedforward input to be parameterized in a lower-dimensional and potentially more generalizable form.

By subtracting (5) from (6), the error propagation from trial j to $j+1$ is obtained as

$$e_{j+1} = e_j - S_0 P_0 \Psi(r)(\theta_{j+1} - \theta_j). \tag{8}$$

In BFILC, the parameters θ_{j+1} is updated by minimizing a cost function that balances tracking accuracy, control effort, and update smoothness:

$$J(\theta_{j+1}) = \|e_{j+1}\|^2_{W_e} + \|f_{j+1}\|^2_{W_f} + \|f_{j+1} - f_j\|^2_{W_{\Delta f}}, \tag{9}$$

where $W_e > 0$ and $W_f, W_{\Delta f} \geq 0$ are weighting matrices.

Due to the quadratic form of J and linearity in the system model, the optimal parameter update is obtained analytically:

$$\theta_{j+1} = \arg\min_{\theta_{j+1}} J(\theta_{j+1}). \tag{10}$$

2) Limitation of BFILC for LPV Robots: While BFILC significantly improves task flexibility, its effectiveness critically depends on the selection and structure of the basis functions Ψ. Typical choices include derivatives of the reference trajectory (e.g., velocity, acceleration, jerk) [17] and data-driven kernel methods [21]. However, these approaches often overlook key physical characteristics such as configuration-dependent nonlinearities and dynamic coupling present in multi-joint robotic systems.

In robotic systems with complex dynamics, joint interactions and nonlinear dependencies on inertial parameters must be explicitly considered. Effective learning and generalization thus require the basis functions to capture these dynamic properties explicitly. This motivates the need for a principled basis function design that is derived from multi-body system dynamics, which forms the foundation of the approach proposed in this work.

III. DAILC for LPV Robots

This section presents the core contribution of this work: the development of a DAILC framework tailored for position-dependent robotic systems. The proposed formulation embeds

Authorized licensed use limited to: TU Delft Library. Downloaded on October 08,2026 at 07:56:05 UTC from IEEE Xplore. Restrictions apply.

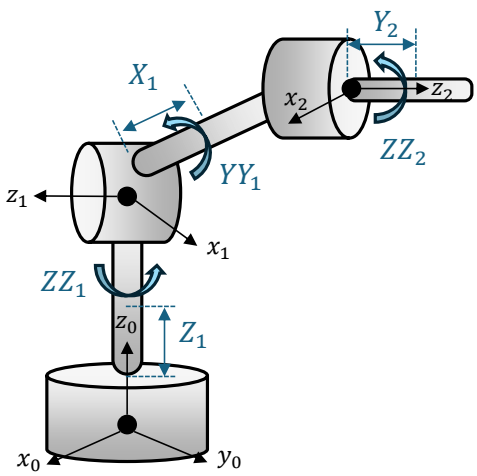


Fig. 2: Configuration of the 3-DOF LPV robot with a Z–Y–Z coordinate frame.

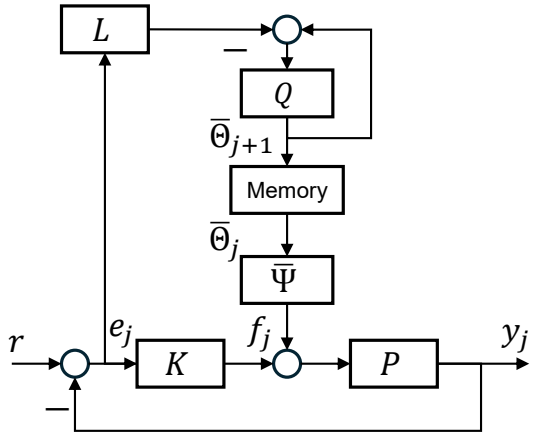


Fig. 3: Block diagram of the proposed DAILC architecture for LPV robotic systems. The basis functions are derived from LPV robot dynamics to ensure task flexibility and physics consistency.

multi-body robot dynamics directly into the basis-function design, enhancing generalization and ensuring greater physical consistency in the learning process.

A. Dynamics-aware Basis Function Design

The robot dynamics in (1) can be expressed explicitly in terms of inertial parameters using the reference trajectory signals $q_r, \dot{q}_r, \ddot{q}_r$ and torque reference τ_r as follows:

$$\mathcal{M}(q_r,\theta)\ddot{q}_r + \mathcal{C}(q_r,\dot{q}_r,\theta)\dot{q}_r + \mathcal{G}(q_r,\theta) = \tau_r, \qquad (11)$$

where the parameter vector θ encapsulates the physical characteristics of the robot's links, including mass, center of mass, and inertia. Specifically, for the i-th link, the inertial properties are defined by its mass m_i, the first moment vector $m\mathbf{S}_i = [mX_i, mY_i, mZ_i]^T$, and the rotational inertia tensor I_i as depicted in Fig. 2:

$$I_i = \begin{bmatrix} XX_i & XY_i & XZ_i \\ XY_i & YY_i & YZ_i \\ XZ_i & YZ_i & ZZ_i \end{bmatrix}. \qquad (12)$$

These quantities are grouped into a standard inertial parameter vector for each body:

$$\theta_i = [m_i, mX_i, mY_i, mZ_i, XX_i, XY_i, XZ_i, YY_i, YZ_i, ZZ_i]^T, \qquad (13)$$

and concatenated for all bodies into the standard inertial parameter vector $\theta \in \mathbb{R}^{10N}$.

Under this parameterization, the dynamics in (11) admit a linear representation with respect to θ:

$$\tau_r = \Psi(q_r,\dot{q}_r,\ddot{q}_r)\theta, \qquad (14)$$

where $\Psi \in \mathbb{R}^{n\times 10N}$ is the regressor matrix.

However, not all parameters in θ are identifiable; some parameters either do not appear or only appear in linear combinations [22]. Any maximal linearly independent set of identifiable parameter combinations constitutes the base parameters for the robot. Standard parameters can be partitioned as:

$$\theta = [\theta_1^T \quad \theta_2^T]^T, \qquad (15)$$

where $\bar{\theta} = \theta_1 + R\theta_2$ for some matrix $R \in \mathbb{R}^{b\times(10N-b)}$.

A minimal set of base parameters $\bar{\theta} \in \mathbb{R}^b$ (with $b < 10N$) is extracted using numerical techniques such as QR or singular value decomposition (SVD). The dynamics are then expressed as:

$$\tau_r = \bar{\Psi}(q_r,\dot{q}_r,\ddot{q}_r)\bar{\theta}, \qquad (16)$$

where $\bar{\Psi} \in \mathbb{R}^{n\times b}$ is the base regressor matrix.

This formulation forms the foundation for constructing the basis function matrix in the proposed DAILC algorithm. By learning the base parameters $\bar{\theta}$ associated with physically meaningful and identifiable dynamic features, the DAILC framework enables accurate and generalizable feedforward control for LPV robotic systems performing variable tasks.

Compared to signal-based basis sets (e.g., splines or polynomials), this formulation represents LPV robot dynamics, as $\bar{\Psi}$ reflects real joint coupling and mass-inertia relationships. It also provides physically meaningful insights into model mismatch or task-specific variation, which is crucial for diagnostics and transfer learning.

B. Learning Framework of DAILC

The learning framework of DAILC integrates dynamics-aware basis functions explicitly derived from LPV robot dynamics, thereby enhancing physical interpretability and generalizability of the feedforward controller across diverse motion tasks. Fig. 3 illustrates the overall closed-loop control architecture, combining a stabilizing feedback controller K with an iterative feedforward update mechanism.

In this formulation, the feedforward input at trial j is parameterized via the base regressor matrix $\bar{\Psi}(q_r,\dot{q}_r,\ddot{q}_r)$ and

Authorized licensed use limited to: TU Delft Library. Downloaded on October 08,2026 at 07:56:05 UTC from IEEE Xplore. Restrictions apply.

the parameter vector $\bar{\theta}_j$. Utilizing the error dynamics from (8), the DAILC update law explicitly becomes:

$$e_{j+1} = e_j - S_0 P_0 \bar{\Psi}(\bar{\theta}_{j+1} - \bar{\theta}_j). \tag{17}$$

The parameter update rule for $\bar{\theta}_{j+1}$ is obtained by minimizing the quadratic cost function in (9), yielding an analytical expression:

$$\bar{\theta}_{j+1} = Q\bar{\theta}_j - Le_j, \tag{18}$$

with the learning filter Q and L given by:

$$L = \left(\bar{\Psi}^T(J^T W_e J + W_f + W_{\Delta f})\bar{\Psi}\right)^{-1} \bar{\Psi}^T J^T W_e, \tag{19}$$

$$Q = \left(\bar{\Psi}^T(J^T W_e J + W_f + W_{\Delta f})\bar{\Psi}\right)^{-1} \bar{\Psi}^T (J^T W_e J + W_{\Delta f})\bar{\Psi}. \tag{20}$$

These filters depend explicitly on the system dynamics, basis function structure, and weighting matrices, thus regulating the convergence rate, robustness, and smoothness of parameter updates.

Subsequently, the updated feedforward torque at trial $j+1$ is computed by:

$$f_{j+1} = \bar{\Psi}(q_r, \dot{q}_r, \ddot{q}_r)\bar{\theta}_{j+1}. \tag{21}$$

A principal advantage of the DAILC framework is its inherent physical consistency. By embedding robot-specific dynamic characteristics directly into the basis functions, the learned parameters retain meaningful interpretations, enabling effective generalization across varying reference trajectories and task conditions. Furthermore, the explicit dynamics-based structure facilitates diagnostics, interpretability of learning outcomes, and potential transfer learning applications.

The DAILC learning procedure is summarized concisely in Algorithm 1. Each iteration refines the feedforward parameters, progressively improving tracking performance through iterative updates based on direct physical insights from robot dynamics.

Algorithm 1 Proposed DAILC Algorithm

1: **Initialize:** Reference signal $r = [q_r, \dot{q}_r, \ddot{q}_r]$, iteration index $j = 0$, parameter vector $\bar{\theta}_0 = 0$
2: Construct $\bar{\Psi}$ and initial parameter vector $\bar{\theta}$ (see (16))
3: **for** $j = 0, 1, \ldots, N-1$ **do**
4: Update feedforward $f_j = \bar{\Psi}(r)\bar{\theta}_j$;
5: Perform simulation or experiment using r and f_j, measure error e_j
6: Compute parameter update: $\bar{\theta}_{j+1} = Q\bar{\theta}_j - Le_j$;
7: j→j+1;
8: **end for**
9: **Output:** Final learned parameter vector $\bar{\theta} = \bar{\theta}_N$.

IV. Validation

This section presents simulation-based validation of the proposed DAILC framework using a 3-DOF serial robotic arm as shown in Fig. 4. The goal is to demonstrate improved tracking accuracy, convergence behavior, and robustness to model uncertainty compared to existing control approaches.

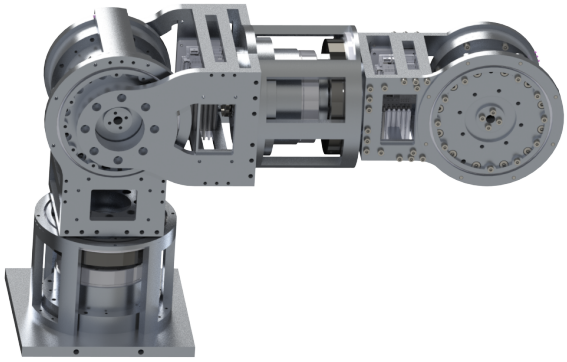


Fig. 4: Configuration of the 3-DOF robotic arm used for simulation validation.

A. Simulation Setup

The simulated robot consists of three revolute joints configured in a serial kinematic chain with a Z-Y-Z joint arrangement, as depicted in Fig. 2 and Fig. 4. This structure captures essential characteristics of multi-joint manipulators while maintaining simplicity for validation purposes. The simulation runs at a sampling period of 1 ms, and a PD controller is employed for feedback stabilization.

The performance of the proposed DAILC is benchmarked against two established methods:

- Norm-optimal ILC [14]: As described in Section II-B, this method learns task-specific feedforward signals based on iterative tracking error minimization.
- CTC [12]: A model-based controller that uses inverse robot dynamics to generate feedforward torques.

While CTC demonstrates high accuracy with accurate models, it is sensitive to model uncertainties. In contrast, DAILC and norm-optimal ILC use the system model mainly for designing the learning filter, which influences convergence rather than steady-state accuracy, thus mitigating sensitivity to modeling inaccuracies.

To evaluate robustness, model uncertainty is introduced by scaling the mass and inertia parameters of each link:

$$\bar{m}_i = \Delta m_i, \quad \bar{I}_i = \Delta I_i, \tag{22}$$

where Δ is a scalar representing uncertainty.

B. DAILC Implementation

This section details the DAILC framework implementation, including basis function construction, learning filter design, and validation task configuration.

1) Basis function construction: The basis function matrix Ψ is initially derived from the robot dynamics as described in (14), involving all inertial parameters for the three-joint configuration. However, due to inherent rank deficiencies arising from the particular robot configuration and selected reference trajectory, the regressor matrix is numerically reduced to a full-rank basis $\bar{\Psi} \in \mathbb{R}^{n\times 17}$. Consequently, the DAILC learning parameters are reduced to the minimal identifiable parameter set $\bar{\theta} \in \mathbb{R}^{17}$.

Authorized licensed use limited to: TU Delft Library. Downloaded on October 08,2026 at 07:56:05 UTC from IEEE Xplore. Restrictions apply.

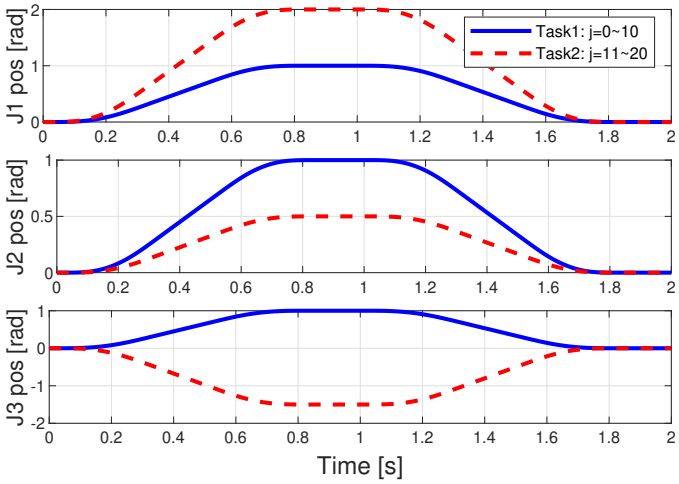


Fig. 5: Trajectories in two tasks: initial segment ($j = 0 \sim 10$) for parameter optimization and subsequent validation segment ($j = 11 \sim 20$) for generalization testing.

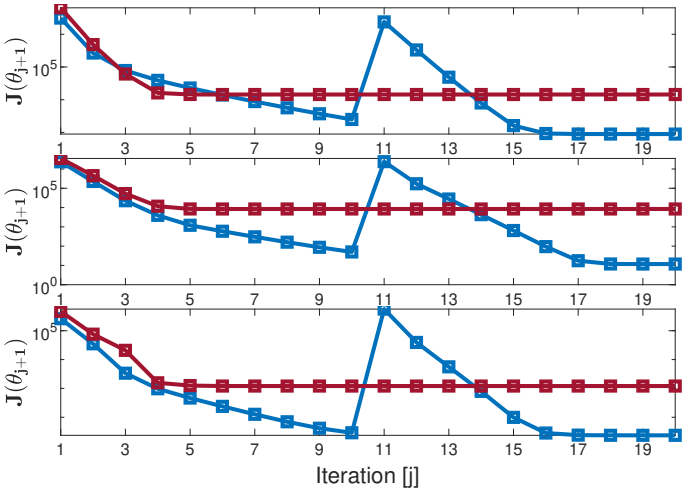


Fig. 6: Cost function values $J(\theta_{j+1})$ where DAILC (—) is insensitive to the reference changes at $j = 11$, in contrast to norm-optimal ILC (—). Top to bottom: Joint 1 to Joint 3.

2) Learning filter design: The learning filters, characterized by matrices Q and L, are computed using weighting matrices W_e, W_f, and $W_{\Delta f}$, as well as the process sensitivity function S_0P_0. Among these, the weighting matrix W_e significantly affects convergence and is manually adjusted to achieve an optimal learning rate while maintaining stability. The process sensitivity function S_0P_0 is approximated by employing a LTI model of the robot dynamics. Simulation results demonstrate that the learning filters designed using the LTI model exhibit robust performance without compromising tracking accuracy.

3) Task design and validation: DAILC and norm-optimal ILC are trained with two distinct reference trajectories as shown in Fig. 5. Fig. 6 shows DAILC rapidly converges, maintaining stable performance across task changes ($j = 10$). Norm-optimal ILC achieves slightly higher accuracy initially but suffers significantly increased errors after switching tasks at $j = 11$. These results highlight the superior adaptability of DAILC for non-repetitive tasks.

C. Simulation results

After optimization with two tasks, performances of controllers are evaluated in sequential trajectory tracking shown in Fig. 7. Norm-optimal ILC uses feedforward signals learned at iteration $j = 10$; CTC is manually tuned for optimal performance.

1) Task flexibility: Fig. 8 illustrates position errors in the absence of model uncertainty. Norm-optimal ILC exhibits the lowest error during 0-2s but undergoes significant performance degradation after the task change at 2s. In contrast, both CTC and DAILC maintain consistently low errors across all trajectories – approximately 0.01% of the reference – demonstrating robust adaptability across varying tasks. The comparable performance of CTC and DAILC is expected, as CTC benefits from the availability of an accurate dynamic model when no uncertainties are present.

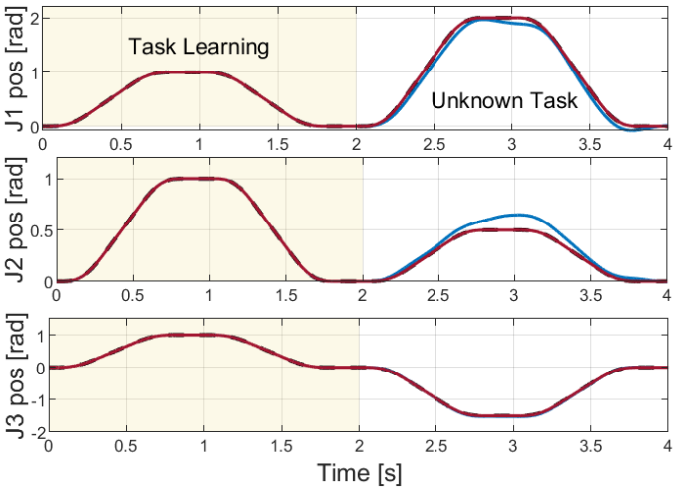


Fig. 7: Task-flexible reference tracking for the DAILC (--), norm-optimal ILC (--), CTC (--) and reference trajectory (—).

2) Robustness to model uncertainty: To closely emulate realistic scenarios, model uncertainties are introduced by applying the scaling factor $\Delta = 0.9$ as defined in (22). Fig. 9 shows that norm-optimal ILC still achieves the lowest error during the initial task (0-2s), but exhibits a significant increase in error upon task switching. The performance of CTC notably deteriorates due to model uncertainty compared to the ideal-case scenario in Fig. 8. Conversely, DAILC maintains consistent error magnitudes, similar to the ideal case as depicted in Fig. 8, highlighting its robustness against model uncertainties and demonstrating superior adaptability to varying tasks.

V. CONCLUSION

This paper introduced a novel Dynamics-Aware Iterative Learning Control (DAILC) framework tailored specifically

Authorized licensed use limited to: TU Delft Library. Downloaded on October 08,2026 at 07:56:05 UTC from IEEE Xplore. Restrictions apply.

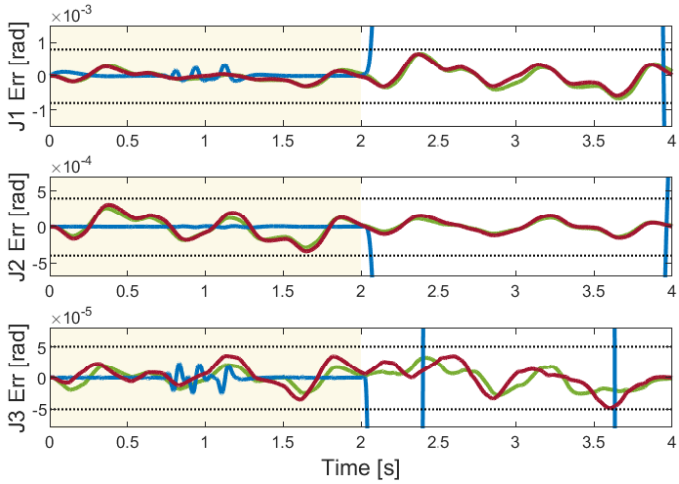


Fig. 8: Position tracking error without model uncertainty for DAILC (—), norm-optimal ILC (—), and CTC (—). The norm-optimal ILC error increases at 2 s, while CTC and DAILC remain stable.

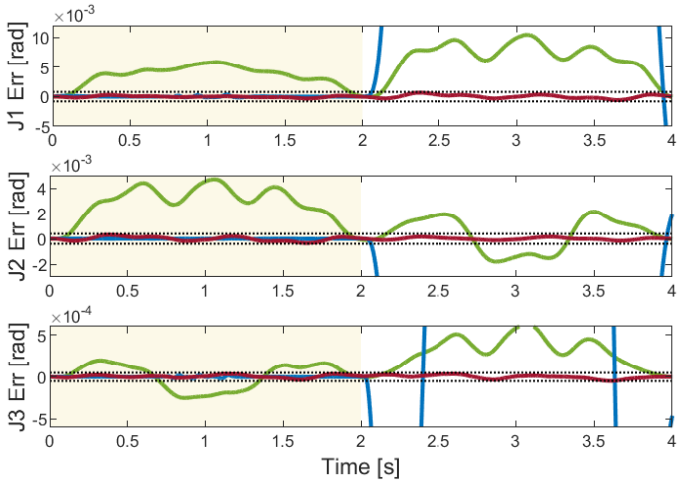


Fig. 9: Position tracking error with model uncertainty for DAILC (—), norm-optimal ILC (—), and CTC (—). Under model uncertainty, CTC performance deteriorates, while the proposed DAILC maintains performance, consistent with the results shown in Fig. 8.

for linear parameter-varying (LPV) robotic systems executing dynamic, non-repetitive tasks. Unlike conventional BFILC methods that rely on generic signal-oriented basis functions, the proposed approach integrates physically consistent, robot dynamics-derived basis functions, enhancing its robustness to system variations and intrinsic nonlinearities typical of multi-body robotic platforms. Simulation validations on a 3-DOF robotic arm demonstrated that the developed framework achieves significant improvements in tracking accuracy, convergence speed, and generalization capabilities compared to existing control techniques. Future work includes experimental validation, extension to higher-DOF robots, and adaptation to environments with disturbance and interaction dynamics.

REFERENCES

[1] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: Modelling, Planning and Control*. Springer, 2010.

[2] Y. Zhang, W. Liu, and L. Wang, "Intelligent manufacturing in the context of industry 4.0: A review," *Engineering*, vol. 8, no. 1, pp. 1–12, 2022.

[3] J. Valente, H. Oliveira, and J. Sousa, "Robotic systems for remote inspection of industrial sites: A review," in *OCEANS 2016-Shanghai*, pp. 1–7, IEEE, 2016.

[4] A. Ajoudani, A. M. Zanchettin, S. Ivaldi, A. Albu-Schäffer, K. Kosuge, and O. Khatib, "Progress and prospects of the human–robot collaboration," *Autonomous Robots*, vol. 42, no. 5, pp. 957–975, 2018.

[5] T. Nozaki, T. Mizoguchi, and K. Ohnishi, "Decoupling strategy for position and force control based on modal space disturbance observer," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 2, pp. 1022–1032, 2013.

[6] D. Lee, J. Back, and S. Oh, "Workspace nonlinear disturbance observer for robust position control of flexible joint robots," *IEEE Robotics and Automation Letters*, 2024.

[7] S. Ozgoli and H. Taghirad, "A survey on the control of flexible joint robots," *Asian journal of control*, vol. 8, no. 4, pp. 332–344, 2006.

[8] W.-H. Chen, D. J. Ballance, P. J. Gawthrop, and J. O'Reilly, "A nonlinear disturbance observer for robotic manipulators," *IEEE Transactions on industrial Electronics*, vol. 47, no. 4, pp. 932–938, 2000.

[9] M. Makarov, M. Grossard, P. Rodriguez-Ayerbe, and D. Dumur, "Modeling and preview h backslash infty control design for motion control of elastic-joint robots with uncertainties," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 10, pp. 6429–6438, 2016.

[10] S. Islam and X. P. Liu, "Robust sliding mode control for robot manipulators," *IEEE Transactions on industrial electronics*, vol. 58, no. 6, pp. 2444–2453, 2010.

[11] E. Sariyildiz, S. Hangai, T. Uzunovic, T. Nozaki, and K. Ohnishi, "Stability and robustness of the disturbance observer-based motion control systems in discrete-time domain," *IEEE/ASME Transactions on Mechatronics*, vol. 26, no. 4, pp. 2139–2150, 2020.

[12] K. M. Lynch and F. C. Park, *Modern robotics*. Cambridge University Press, 2017.

[13] O. Nelles and O. Nelles, *Nonlinear dynamic system identification*. Springer, 2020.

[14] L. Blanken and T. Oomen, "Multivariable iterative learning control design procedures: From decentralized to centralized, illustrated on an industrial printer," *IEEE Transactions on Control Systems Technology*, vol. 28, no. 4, pp. 1534–1541, 2019.

[15] C. Wang, M. Zheng, Z. Wang, C. Peng, and M. Tomizuka, "Robust iterative learning control for vibration suppression of industrial robot manipulators," *Journal of Dynamic Systems, Measurement, and Control*, vol. 140, no. 1, p. 011003, 2018.

[16] J. Bolder, T. Oomen, S. Koekebakker, and M. Steinbuch, "Using iterative learning control with basis functions to compensate medium deformation in a wide-format inkjet printer," *Mechatronics*, vol. 24, no. 8, pp. 944–953, 2014.

[17] J. Bolder and T. Oomen, "Rational basis functions in iterative learning control—with experimental verification on a motion system," *IEEE Transactions on Control Systems Technology*, vol. 23, no. 2, pp. 722–729, 2014.

[18] D. Morgan, J. Gilmour, D. Dawson, and R. Patel, "Trajectory generalization in robotic manipulation via basis function ilc," *Robotics and Autonomous Systems*, vol. 127, p. 103462, 2020.

[19] M. Pierallini, F. Angelini, R. Mengacci, A. Palleschi, A. Bicchi, and M. Garabini, "Iterative learning control for compliant underactuated arms," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 53, no. 6, pp. 3810–3822, 2023.

[20] R. Tóth, *Modeling and identification of linear parameter-varying systems*, vol. 403. Springer, 2010.

[21] J. Ding, M. A. van Löben Sels, F. Angelini, J. Kober, and C. Della Santina, "Robust jumping with an articulated soft quadruped via trajectory optimization and iterative learning," *IEEE Robotics and Automation Letters*, vol. 9, no. 1, pp. 255–262, 2023.

[22] C. G. Atkeson, C. H. An, and J. M. Hollerbach, "Estimation of inertial parameters of manipulator loads and links," *The International Journal of Robotics Research*, vol. 5, no. 3, pp. 101–119, 1986.

Authorized licensed use limited to: TU Delft Library. Downloaded on October 08,2026 at 07:56:05 UTC from IEEE Xplore. Restrictions apply.