Knowledge Transfer Between Robots with Online Learning for Enhancing Robot Performance in Impromptu Trajectory Tracking

Siqi Zhou[1], Andriy Sarabakha[2], Erdal Kayacan[3], Mohamed K. Helwa[1], and Angela P. Schoellig[1]

***Abstract*— As robot dynamics become more complex, learning from data is emerging as an alternative for obtaining accurate dynamic models to assist control system designs or to enhance robot performance. Though being effective, common model learning techniques rely on rich datasets collected from the robots, and the learned experience is often platform-specific. In this work, we propose an online learning approach for transferring deep neural network (DNN) inverse dynamics models across two robots and analyze the role of dynamic similarity in the transfer problem. We demonstrate our proposed knowledge transfer approach with two different quadrotors on impromptu trajectory tracking tasks, in which the quadrotors are required to track arbitrary hand-drawn trajectories accurately from the *first attempt*. With this work, we illustrate that *(i)* we can relate the transferability of DNN inverse models to the robot dynamic properties, and *(ii)* when the transfer is feasible, we can significantly reduce data recollections that would be otherwise costly or risky for robot applications. Given a heterogeneous robot team, we envision having to train only one of the agents to allow the whole team achieving higher performance.**

I. Introduction

Trajectory tracking is fundamental to many robot applications, which include industrial inspection, autonomous driving, and search and rescue missions [1]–[3]. In these applications, high-accuracy tracking is often demanded to ensure safe operation and/or to realize optimized performance. While typical approaches such as the PID control and model predictive control (MPC) can be used to realize tracking functionalities, they often rely on sufficiently accurate dynamic models of the robots or require long tuning processes to achieve high performance. As the robot dynamics become more complex and the operating environment becomes increasingly unstructured, integrated learning and control techniques start to emerge as alternatives to ensure high performance in the presence of modeling uncertainties [4].

In the literature, there are various successful examples of applying learning techniques to robot control problems (e.g., [5]–[7]). Although being effective, these learning-based methods usually rely on rich datasets collected from the robot, and the learned experience is often robot-specific. In a multi-robot system, applying these learning techniques would imply repeating the data collection and training process on each individual robot, which can be non-economic and time-consuming. In order to increase the efficiency of robot learning, in the robotics community, approaches such as manifold alignment [8]–[10] and learning invariant features [11], [12] have been recently proposed to transfer knowledge between robots and thereby speed up the training of new robots and enhance their performance in untrained tasks.

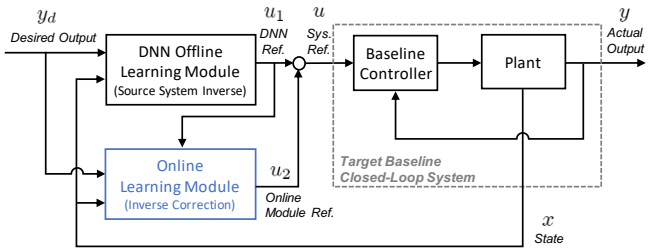

Fig. 1: Block diagram of the DNN-enhanced control architecture with online learning. The DNN offline learning module represents the inverse dynamics of a source robot system and is previously trained on a sufficiently rich dataset collected from the source robot. At test time, the DNN module is leveraged to enhance the tracking performance of a target robot. The online learning module is trained on real-time data to compensate for the dynamic differences between the robots. A demo video: http://tiny.cc/dnnTransfer

In this work, we consider the problem of impromptu trajectory tracking, in which robots are required to track arbitrary trajectories accurately from the first attempt. This problem is applicable to a robot team setting where, for instance, each robot receives a desired trajectory from an online planner to perform coordinated tasks. In [7], [13], we showed that we can effectively enhance the tracking performance of a robot by training a deep neural network (DNN) inverse dynamics module offline and pre-cascading the module to a baseline system at test time (Fig. 1). For example, on 30 arbitrary, unseen hand-drawn trajectories, the DNN-enhancement approach reduced the tracking error of a quadrotor by an average of 43% [7].

Motivated by the work on recent knowledge transfer (or transfer learning), in current work, we study feasibility of leveraging the DNN inverse module trained on one robot to enhance the performance of another robot in impromptu tracking tasks. In contrast to the existing transfer learning approaches, where transfer mappings are usually found offline based on a set of sample tasks (e.g., [9], [11]), we propose an online learning approach (Fig. 1) that allows a target robot using the DNN module from a source robot to achieve high-accuracy tracking *impromptu* — i.e., without additional data collection on sample tasks and offline training. The proposed online learning approach is expected to *(i)* significantly reduce the data required to train new robots by leveraging

* This extended abstract is a condensed version of [14], where we present full details of this work.

[1]Siqi Zhou, Mohamed K. Helwa, and Angela P. Schoellig are with the Dynamic Systems Lab (http://www.dynsyslab.org), Institute for Aerospace Studies, University of Toronto, Canada. Emails: {siqi.zhou, mohamed.helwa}@robotics.utias.utoronto.ca, schoellig@utias.utoronto.ca

[2]Andriy Sarabakha is with the School of Mechanical and Aerospace Engineering, Nanyang Technological University, Singapore. Email: andriy001@e.ntu.edu.sg

[3]Erdal Kayacan is with the Department of Engineering, Aarhus University, Denmark. Email: erdal@eng.au.dk

a prior DNN inverse module from a source robot, and *(ii)* additionally compensate for real-time changes in robot dynamics (e.g., due to mechanical wear or other unforeseen effects) that may otherwise degrade a robot's or a robot team's performance. Our contributions are as follows:

(1) analytically derive the ideal mapping that the online module should represent to achieve exact tracking,
(2) present first results on characterizing system similarity between source and target robots and how it relates to the stability of the proposed overall learning system given modeling uncertainties, and
(3) verify the effectiveness of the proposed approach in impromptu trajectory tracking experiments on quadrotors.

Note that, with the proposed knowledge transfer approach, we ultimately aim to enhance the tracking of trajectories provided by high-level planners. The approach is agnostic to the nature of the high-level planner and can be potentially used for general multi-agent coordination tasks where trajectories for individual agents are given. This work is originally presented in [14] where we provide full details of our derivations and discussions; this abstract highlights the main theoretical and experimental results.

II. Background on Offline DNN Inverse Dynamics Learning

In this section, we provide a brief summary of the DNN inverse learning approach in [7], [13] to facilitate our discussion. We consider a nonlinear baseline system represented by

$$x(k+1) = f\left(x(k)\right) + g\left(x(k)\right)u(k), \;\; y(k) = h\left(x(k)\right), \tag{1}$$

where k is the discrete-time index, x is the system state, u is the reference signal, y is the system output, and $f(\cdot)$, $g(\cdot)$, and $h(\cdot)$ are smooth functions. For many practical cases, we may relate the system input and output by

$$y(k+r) = \mathcal{F}\left(x(k)\right) + \mathcal{G}\left(x(k)\right)u(k), \tag{2}$$

where $\mathcal{F}\left(x(k)\right) = h \circ f^r(x(k))$, $\mathcal{G}\left(x(k)\right) = \frac{\partial}{\partial u} h \circ f^{r-1}(f(x(k)) + g(x(k))u(k))$, and r is the relative degree (or inherent delay) of system (1) [15]. One can show that the reference for achieving exact tracking (i.e., $y(k+r) = y_d(k+r)$ with y_d denotes the desired output) is given by

$$u(k) = \frac{1}{\mathcal{G}\left(x(k)\right)}\left(y_d(k+r) - \mathcal{F}\left(x(k)\right)\right). \tag{3}$$

Eqn. (3) can be thought as the exact inverse of the baseline system (1). In [7], [13], we showed that, for an unknown, nonlinear baseline system that has stable inverse dynamics and a well-defined relative degree r, we can train a DNN module offline with input $\mathcal{I} = [x(k), y(k+r)]$ and output $\mathcal{O} = [u(k)]$ to approximate the exact inverse in Eqn. (3) to enhance the baseline system tracking performance.

III. Problem Statement

We consider the control architecture shown in Fig. 1 and study the impromptu knowledge transfer problem that allows the DNN inverse dynamics module trained on a source robot to enhance the tracking performance of a target robot having different dynamics. Following [13], we consider source and target robot systems represented by Eqns. (1) and (2). In order for the DNN inverse learning approach to be safely applied, we assume that (A1) the source and the target systems are input-to-state stable and have stable inverse dynamics [13]. We also assume that (A2) the source and the target systems have well-defined and the same relative degree to simplify the analysis. This holds, for instance, if the robots have similar structures but different mass or dimensions.

IV. Theoretical Results

In this section, we provide theoretical results on the knowledge transfer problem. We denote u_1 as the reference from the DNN module trained on the source system and u_2 as the reference from the online learning module. The overall reference send to the target baseline system $u(k)$ is given by

$$u(k) = u_1(k) + u_2(k). \tag{4}$$

Below we provide an expression of $u_2(k)$ for achieving exact tracking in Sec. IV-A, propose a characterization of system similarity in Sec. IV-B, and analyze the stability of the overall system in the presence of uncertainties in Sec. IV-C.

A. Online Learning Module

We propose an online learning approach that adapts the reference of the DNN module $u_1(k)$ based on the tracking error. In particular, by considering source and target robot systems represented by Eqns. (1) and (2), which the exact inverse dynamics are represented by Eqn. (3), we can show that the ideal mapping which the online learning module should represent to achieve exact tracking is

$$u_2(k) = \alpha^* e_p^*(k+r), \tag{5}$$

where $\alpha^* = \frac{1}{\mathcal{G}_t(x(k))}$ is an adaptation gain,

$$e_p^*(k+r) = y_d(k+r) - \mathcal{F}_t\left(x(k)\right) - \mathcal{G}_t\left(x(k)\right)u_1(k) \tag{6}$$

is a prediction of the tracking error as the result of applying $u_1(k)$, and $\mathcal{F}_t\left(x(k)\right) = h_t \circ f_t^r(x(k))$ and $\mathcal{G}_t\left(x(k)\right) = \frac{\partial}{\partial u} h_t \circ f_t^{r-1}(f_t(x(k)) + g_t(x(k))u(k))$, and $f_t(\cdot)$, $g_t(\cdot)$, and $h_t(\cdot)$ are the nonlinear functions in Eqn. (1) for the target system.

Note that the error prediction in Eqn. (6) depends on the state $x(k)$, the reference $u_1(k)$ from the DNN module, and the desired output $y_d(k+r)$. When the dynamics of the source and the target systems are not known, one may train an online learning model to approximate Eqn. (6).

Remark 1. Online Learning for Error Prediction. For training an online model to approximate Eqn. (6), at each time step k, one may construct a dataset with paired inputs $\{x(p-r), u(p-r), y_d(p)\}$ and outputs $\{y_d(p) - y(p)\}$ over the past N time steps $p = k-N, ..., k$. The error $e_p(k+r)$ can then be predicted using the online model with input $\mathcal{I} = [x(k), u_1(k), y_d(k+r)]$.

Given an online model $F\left(x(k), u_1(k), y_d(k+r)\right)$ approximating Eqn. (6), we can estimate α^* by $-\left(\partial F/\partial u_1\right)^{-1}$.

B. Characterization of System Similarity

The concept of task similarity has been introduced in the reinforcement learning (RL) literature to address the issue of negative knowledge transfer in task transfer problems. In this work, we propose a characterization of system similarity for impromptu knowledge transfer problems, where an inverse module is transferred across two robot systems. We consider two systems are similar if at any given state $x(k)$, the application of an input $u(k)$ to the systems results in similar outputs $y(k+r)$. For this discussion, we assume linear source and target systems to simplify our analysis:

$$x(k+1) = Ax(k) + Bu(k), \;\; y(k) = Cx(k), \qquad (7)$$

where (A, B, C) are constant matrices. It can be shown that the input and output of the system are related by

$$y(k+r) = \mathcal{A}x(k) + \mathcal{B}u(k), \qquad (8)$$

where $\mathcal{A} = CA^r$ and $\mathcal{B} = CA^{r-1}B$, and r is the relative degree of system (7). Based on Eqn. (8), we define a vector S to characterize the similarity of the source and target systems:

$$S = \begin{bmatrix} S_1 & S_2 \end{bmatrix}, \qquad (9)$$

where $S_1 = 1 - \frac{\mathcal{B}_t}{\mathcal{B}_s}$, $S_2 = \mathcal{A}_t - \frac{\mathcal{B}_t}{\mathcal{B}_s}\mathcal{A}_s$, and the subscripts s and t denote the source and the target system. The terms S_1 and S_2, respectively, characterize the differences in the input-to-output gain and state-to-output gain vector of the source and target systems. Note that $S = 0$ if and only if $\mathcal{A}_t = \mathcal{A}_s$ and $\mathcal{B}_t = \mathcal{B}_s$ (i.e., the state-to-output and input-to-output gains of the systems are identical).

C. Stability of Learning-Enhanced Target System

We apply the concept of system similarity and analyze the stability of the target system when the gain α^* is approximated by a constant α and the prediction of the future error $e_p^*(k+r)$ is not exact. We discuss the main result in this abstract; details of the proof can be found in [14].

We focus on system (7) and additionally assume that: (A3) The output of the offline DNN $u_1(k)$ corresponds to the inverse of the source system $u_1(k) = \frac{1}{\mathcal{B}_s}\left(y_d(k+r) - \mathcal{A}_s x(k)\right)$, and (A4) the error in the prediction $\Lambda = e_p^*(k+r) - e_p(k+r)$ can be bounded as $\Lambda \leq \beta_1||y_d(k+r)|| + \beta_2||x(k)|| + \beta_3$, where $(\beta_1, \beta_2, \beta_3)$ are positive constants, and $||\cdot||$ is the Euclidean norm. Moreover, by (A1), the state can be bounded as $||x||_\infty \leq L_1||u||_\infty + L_2||x_0||$, where $||x||_\infty = \sup_k\{||x(k)||\}$, $||u||_\infty = \sup_k\{||u(k)||\}$, and (L_1, L_2) are positive constants.

Under (A1)-(A4), we can show that the overall target system is bounded-input-bounded-state (BIBS) stable if

$$|\alpha|\left(||S_2|| + \beta_2\right) < \frac{\beta_4}{L_1}, \qquad (10)$$

where $\beta_4 = 1 - L_1\left|\left|\frac{\mathcal{A}_s}{\mathcal{B}_s}\right|\right|$. The stability condition in Eqn. (10) can be interpreted for two scenarios: *(i)* when $|\alpha| = 0$ (i.e., the online module is inactive) and *(ii)* when $|\alpha| \neq 0$ (i.e., the online module is active). In scenario *(i)*, the condition in Eqn. (10) reduces to $L_1 < \frac{1}{||\mathcal{A}_s/\mathcal{B}_s||}$, which can be interpreted as an upper bound on the relative aggressiveness of the source and target systems. When this condition is satisfied, the target system with the source DNN module is stable. In scenario *(ii)*, the condition in Eqn. (10) implies that if the source and target systems are similar (i.e., $||S_2||$ is closer to 0), then there will be a greater margin for selecting α and higher tolerance for having uncertainties in the online prediction model.

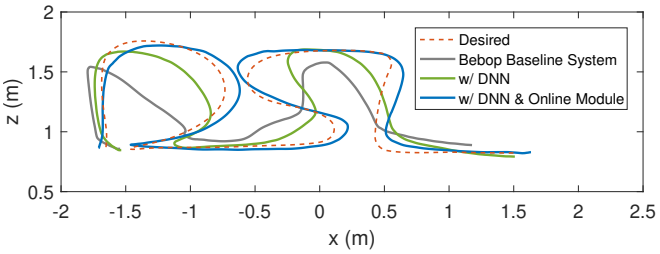

Fig. 2: Comparison of three control strategies for the Bebop target system: The RMS error is 0.42 m for the Bebop baseline system (grey), 0.26 m for the baseline system enhanced by the ARDrone DNN (green), and 0.14 m for the baseline system further enhanced by the online learning module (blue).

V. Quadrotor Experiments

We verify the proposed online learning approach by transferring a DNN module trained on a source quadrotor vehicle (Parrot ARDrone) to enhance the tracking of a target quadrotor vehicle (Parrot Bebop). A demo video of the experiment can be found here: http://tiny.cc/dnnTransfer

A. Experiment Setup

1) Control Objective: The dynamics of a quadrotor vehicle can be characterized by 12 states: translational positions $\mathbf{p} = (x, y, z)$, translational velocities $\mathbf{v} = (\dot{x}, \dot{y}, \dot{z})$, roll-pitch-yaw angles $\boldsymbol{\theta} = (\phi, \theta, \psi)$, and rotational velocities $\boldsymbol{\omega} = (p, q, r)$. The objective is to design a control system such that the position of the quadrotor $\mathbf{p}_a$ tracks desired trajectories $\mathbf{p}_d$ generated from arbitrary hand drawings.

2) DNN Module Trained on ARDrone (Source Robot): In [7], [13], a DNN module is trained offline to approximate the inverse of the ARDrone baseline system dynamics. The DNN module consists of fully-connected feedforward networks with 4 hidden layers of 128 rectified linear units (ReLU). The training dataset of the DNN module is constructed from the ARDrone baseline system response on a 400-second, 3-dimensional sinusoidal trajectory. Overall, approximately 2,800 data points are used for training.

3) Online Learning for Bebop (Target Robot): Based on Remark 1, we design an online learning module to predict the error of the Bebop that would result from applying the reference of the ARDrone DNN inverse module. At each time step k, the most recent 40 observations are used for constructing the training dataset. We implemente the online learning module using a Gaussian process (GP) model with a standard squared-exponential kernel. Details of the online learning module implementation can be found in [14].

B. Experiment Results

Figure 2 compares the tracking performance of three control strategies on the Bebop on one of the test hand-

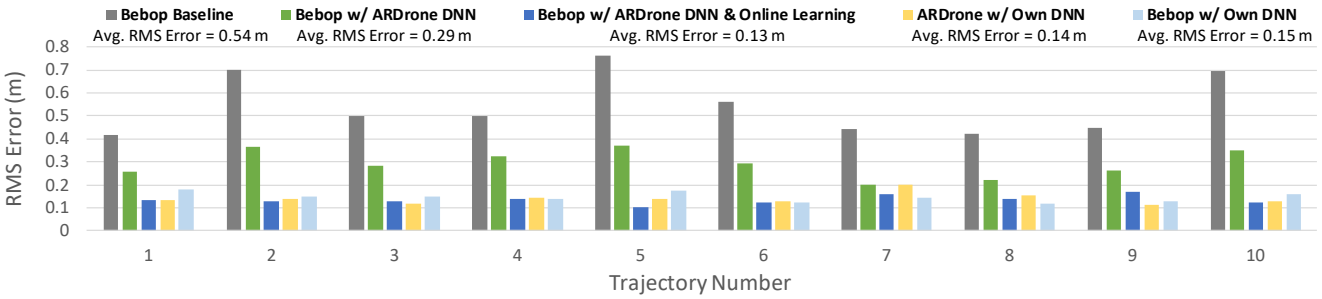

Fig. 3: Performance comparison on 10 hand-drawn trajectories. The ARDrone DNN module alone (green) and the ARDrone DNN module with the online learning module (blue) reduce the tracking error of the Bebop baseline system (grey) by 46% and 74% on average. With the proposed approach (blue), the performance of the Bebop (target robot) is comparable to cases where the quadrotors are enhanced by their own DNN modules (yellow and light blue).

drawn trajectories. When comparing the performance of the Bebop system enhanced by the ARDrone DNN (green) and the performance of the Bebop baseline system (grey), the ARDrone DNN reduces the delay and the amplitude errors in the Bebop tracking response. Along this particular trajectory, the ARDrone DNN module alone reduces the root-mean-square (RMS) tracking error of the Bebop from approximately 0.42 m to 0.26 m. With the addition of the online module, the Bebop RMS tracking error is further reduced to approximately 0.14 m.

Figure 3 shows the testing results on 10 arbitrary hand-drawn trajectories. As compared with the Bebop baseline system (grey), with the transferred ARDrone DNN module and the online learning module, the proposed approach (blue) reduces the tracking error of the Bebop by an average of 74%. Without further offline training, the proposed approach (blue) effectively reduces the tracking error of the Bebop to values that are comparable to the cases where the quadrotors are enhanced by their own offline DNN modules (yellow and light blue). This result demonstrates the efficiency of the proposed knowledge transfer for leveraging past experience and reducing data collection in training new robots.

VI. Conclusion and Future Work

We present an online learning approach that allows us to transfer a DNN inverse dynamics module trained on a source robot to enhance the impromptu tracking performance of a target robot that has different dynamics. We provide theoretical analysis of our proposed approach and verify the approach experimentally with quadrotors. In the experiments, we demonstrate that the performance of the target quadrotor enhanced by the proposed knowledge transfer approach is comparable to the cases where the quadrotors are enhanced by their own offline DNN inverse modules. This result verified that the proposed knowledge transfer approach can efficaciously circumvent data recollection on the target robot, and thus, the costs and risks associated with training new robots. The knowledge transfer approach acts to enhance the lower-level control performance of the robots and can be potentially used for multi-agent coordination tasks where desired trajectories for the individual agents are given. For a heterogeneous team, the efficacy of knowledge transfer implies that we can leverage the experience of a single robot to improve the lower-level control of each agent and thereby enhance the overall performance of the robot team.

As a future work, we would like to study the extent of robot similarity required for exploiting knowledge transfer between structurally-different robots. This will be important as the diversity of robot teams increases and the coordinated tasks become more complex.

References

[1] J. Nikolic, M. Burri, J. Rehder, S. Leutenegger, C. Huerzeler, and R. Siegwart, "A UAV system for inspection of industrial facilities," in *IEEE Aerospace Conf.*, 2013, pp. 1–8.

[2] G. M. Hoffmann, C. J. Tomlin, M. Montemerlo, and S. Thrun, "Autonomous automobile trajectory tracking for off-road driving: Controller design, experimental validation and racing," in *Proc. of the IEEE American Control Conf. (ACC)*, 2007, pp. 2296–2301.

[3] B. Grocholsky, J. Keller, V. Kumar, and G. Pappas, "Cooperative air and ground surveillance," *IEEE Robotics & Automation Magazine*, vol. 13(3), pp. 16–25, 2006.

[4] D. Nguyen-Tuong and J. Peters, "Model learning for robot control: a survey," *Cognitive Processing*, vol. 12(4), pp. 319–340, 2011.

[5] H. D. Patino, R. Carelli, and B. R. Kuchen, "Neural networks for advanced control of robot manipulators," *IEEE Transactions on Neural networks*, vol. 13, no. 2, pp. 343–354, 2002.

[6] A. P. Schoellig, F. L. Mueller, and R. D'Andrea, "Optimization-based iterative learning for precise quadrocopter trajectory tracking," *Autonomous Robots*, vol. 33(1-2), pp. 103–127, 2012.

[7] Q. Li, J. Qian, Z. Zhu, X. Bao, M. K. Helwa, and A. P. Schoellig, "Deep neural networks for improved, impromptu trajectory tracking of quadrotors," in *Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2017, pp. 5183–5189.

[8] H. B. Ammar, E. Eaton, P. Ruvolo, and M. Taylor, "Unsupervised cross-domain transfer in policy gradient reinforcement learning via manifold alignment," in *Proc. of the AAAI Conf. on Artificial Intelligence*, 2015.

[9] B. Bcsi, L. Csat, and J. Peters, "Alignment-based transfer learning for robot models," in *Proc. of the Intl. Joint Conf. on Neural Networks (IJCNN)*, 2013, pp. 1–7.

[10] M. K. Helwa and A. P. Schoellig, "Multi-robot transfer learning: A dynamical system perspective," in *Proc. of the IEEE Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2017, pp. 4702–4708.

[11] A. Gupta, C. Devin, Y. Liu, P. Abbeel, and S. Levine, "Learning invariant feature spaces to transfer skills with reinforcement learning," in *Proc. of the Intl. Conf. on Learning Representations*, 2017.

[12] S. Daftry, J. A. Bagnell, and M. Hebert, "Learning transferable policies for monocular reactive MAV control," in *Proc. of the Intl. Symposium on Experimental Robotics*. Springer, 2016, pp. 3–11.

[13] S. Zhou, M. K. Helwa, and A. P. Schoellig, "Design of deep neural networks as add-on blocks for improving impromptu trajectory tracking," in *Proc. of the IEEE Conf. on Decision and Control (CDC)*, 2017, pp. 5201–5207.

[14] S. Zhou, A. Sarabakha, E. Kayacan, M. K. Helwa, and A. P. Schoellig, "Knowledge transfer between robots with similar dynamics for high-accuracy impromptu trajectory tracking," in *Proc. of the European Control Conf. (ECC)*, 2019, Available: https://arxiv.org/abs/1904.00249.

[15] M. Sun and D. Wang, "Analysis of nonlinear discrete-time systems with higher-order iterative learning control," *Dynamics and Control*, vol. 11(1), pp. 81–96, 2001.