Received: 22 May 2017 | Revised: 26 March 2018 | Accepted: 26 March 2018

DOI: 10.1002/acs.2887

WILEY

Transfer learning for high-precision trajectory tracking through $\mathcal{L}_1$ adaptive feedback and iterative learning

Karime Pereida | Dave Kooijman | Rikky R. P. R. Duivenvoorden | Angela P. Schoellig

Institute for Aerospace Studies, University of Toronto, North York, ON M3H 5T6, Canada

Correspondence
Karime Pereida, Institute for Aerospace Studies, University of Toronto, North York, ON M3H 5T6, Canada.
Email: karime.pereida@robotics.utias.utoronto.ca

Funding information
Canada Foundation for Innovation John R. Evans Leaders Fund, Grant/Award Number: CFI/ORF 33000; Natural Sciences and Engineering Research Council of Canada, Grant/Award Number: CREATE-466088 and RGPIN-2014-04634; Ontario Research Fund, Grant/Award Number: CFI/ORF 33000; Mexican National Council of Science and Technology

Summary

Robust and adaptive control strategies are needed when robots or automated systems are introduced to unknown and dynamic environments where they are required to cope with disturbances, unmodeled dynamics, and parametric uncertainties. In this paper, we demonstrate the capabilities of a combined $\mathcal{L}_1$ adaptive control and iterative learning control (ILC) framework to achieve high-precision trajectory tracking in the presence of unknown and changing disturbances. The $\mathcal{L}_1$ adaptive controller makes the system behave close to a reference model; however, it does not guarantee that perfect trajectory tracking is achieved, while ILC improves trajectory tracking performance based on previous iterations. The combined framework in this paper uses $\mathcal{L}_1$ adaptive control as an underlying controller that achieves a robust and repeatable behavior, while the ILC acts as a high-level adaptation scheme that mainly compensates for systematic tracking errors. We illustrate that this framework enables transfer learning between dynamically different systems, where learned experience of one system can be shown to be beneficial for another different system. Experimental results with two different quadrotors show the superior performance of the combined $\mathcal{L}_1$-ILC framework compared with approaches using ILC with an underlying proportional-derivative controller or proportional-integral-derivative controller. Results highlight that our $\mathcal{L}_1$-ILC framework can achieve high-precision trajectory tracking when unknown and changing disturbances are present and can achieve transfer of learned experience between dynamically different systems. Moreover, our approach is able to achieve precise trajectory tracking in the first attempt when the initial input is generated based on the reference model of the adaptive controller.

KEYWORDS

iterative learning, $\mathcal{L}_1$ adaptive control, trajectory tracking, transfer learning

1 | INTRODUCTION

Robots and automated systems are being deployed in unstructured and continuously changing environments. Sophisticated control methods are required to guarantee high overall performance in these environments where model uncertainties, unknown disturbances, and changing dynamics are present. Examples of robotic applications in unknown

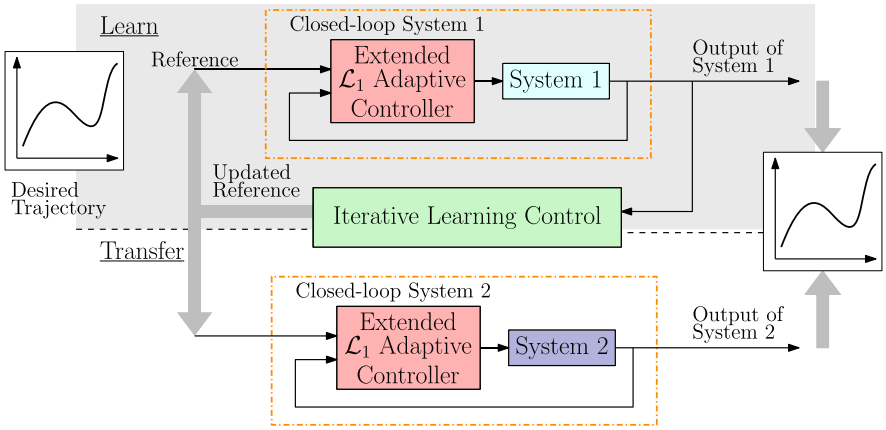

FIGURE 1 The extended $\mathcal{L}_1$ adaptive controller forces a system to behave in a repeatable predefined way. Consequently, two dynamically different systems (System 1 and System 2) can achieve the same predefined behavior, noted by orange dashed boxes. Extended $\mathcal{L}_1$ adaptive controller boxes share the same color as they have the same reference system. The iterative learning controller is capable of learning an input such that the output tracks a desired output signal. After learning a trajectory with System 1, the learned input can be applied to other systems with different dynamical behavior, such as System 2, and the output will still track the reference signal [Colour figure can be viewed at wileyonlinelibrary.com]

dynamic environments include autonomous driving, assistive robotics, and unmanned aerial vehicle applications. In applications where expensive hardware is involved, it is advantageous to use simulators or inexpensive hardware for the initial control design and learning. However, learned trajectories in the inexpensive hardware should be transferred, without further processing, to a different system and achieve a performance comparable to the one obtained in the training system (see Figure 1). Moreover, to achieve high trajectory tracking performance, the underlying controller must be robust enough as small changes in the conditions may otherwise result in a dramatic decrease in controller performance and could cause instability (see other works[1-3]).

The objective of this paper is to design a framework that makes the system achieve a repeatable behavior even in the presence of unknown disturbances and changing dynamics that improves performance over time and that is able to transfer learned trajectories to dynamically different systems, achieving high-precision trajectory tracking. Therefore, we propose a combined $\mathcal{L}_1$ adaptive control and iterative learning control (ILC) framework (see Figure 1).

Control frameworks that combine the advantages of repeatable behavior and improved performance over time have been proposed. In particular, we focus on a combined adaptive control and learning control framework. Adaptive control methods deal with model uncertainties and unknown disturbances. Model reference adaptive control uses the difference between the output of the system and the output of a desired reference model to update control parameters. The goal is for the parameters to converge so the plant response matches the reference model response.[4] Large adaptive gains help to achieve this goal; however, they result in high-frequency oscillations in the control signal.[5] The $\mathcal{L}_1$ adaptive controller is based on the model reference adaptive control architecture with the addition of a low-pass filter that decouples robustness from adaptation. This allows arbitrarily high adaptation gains to be chosen for fast adaptation and to determine uniform bounds for the system's state and control signals.[5] Attitude control based on $\mathcal{L}_1$ adaptive control was shown in the work of Mallikarjunan et al,[6] where three algorithms were successfully implemented and tested on a quadrotor, hexacopter, and octocopter, respectively. $\mathcal{L}_1$ adaptive output feedback on translational velocity was successfully implemented on a quadrotor to compensate for artificial reduction in the speed of a single motor.[7]

Iterative learning control is used to efficiently calculate the feedforward input signal by using information from previous trials to improve tracking performance in a small number of iterations. Iterative learning control has been successfully applied to a variety of trajectory tracking scenarios such as robotic arms,[8] ground vehicles,[9] manufacturing of integrated circuits,[10] swinging up a pendulum,[11] and quadrotor control.[12] An ILC method based on minimization of a quadratic performance criterion was used in the work of Schoellig et al[13] for precise quadrocopter trajectory tracking. A survey on ILC can be found in the work of Bristow et al.[14]

A framework of $\mathcal{L}_1$ adaptive feedback control with parallel ILC was proposed in other works,[15-17] where successful simulation results were presented. In the parallel framework, the input to the system is the addition of the input signal calculated by the $\mathcal{L}_1$ adaptive controller and the input signal calculated by the ILC. The addition of the 2 input signals couples the problems of providing a repeatable system behavior with the problem of improving tracking performance.

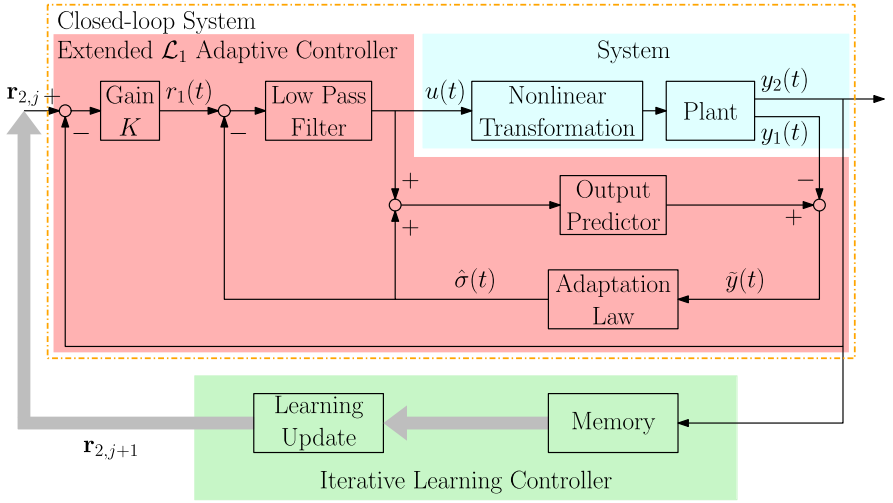

FIGURE 2 Proposed framework to achieve high-precision trajectory tracking in changing environments. The extended $\mathcal{L}_1$ adaptive controller forces the system to behave in a predefined repeatable way. The iterative learning controller improves the tracking performance in each iteration j based on experience from previous executions [Colour figure can be viewed at wileyonlinelibrary.com]

In our earlier work,[18] we proposed and showed the first experimental results on a quadrotor of a serial framework of $\mathcal{L}_1$ adaptive control and ILC under changing dynamics. The serial architecture uses the $\mathcal{L}_1$ adaptive controller as an underlying control to achieve a repeatable system behavior despite the presence of unknown and changing disturbances. It then applies ILC to the now repeatable system to improve trajectory tracking performance. In this work, we exploit the repeatable behavior of the $\mathcal{L}_1$ adaptive controller to transfer learning between dynamically different systems while achieving high-precision trajectory tracking.

Different strategies for transferring learning data from simulation to the real world have previously been proposed. In the work of Tobin et al,[19] simulated images generated by randomizing rendering in a simulator were used to train models for object localization. These models transfer to real images and are accurate enough to be used to perform grasping in cluttered environments. Furthermore, noting that policies that succeed in simulation often do not work when deployed in a real robot, Christiano et al[20] proposed to use what a simulation-based control policy expects the next state(s) will be and, based on a learned deep inverse dynamics model, calculate which real-world action is most suitable to achieve those states. However, to accurately train a deep inverse dynamics model, a significant amount of real data is required. In contrast, our work only requires both systems to behave in the same predefined way, through the use of an adaptive controller, to be able to transfer learning from simulation to real world. Moreover, a strategy that transfers learning from specific skills and robots to different skills and robots was proposed in the work of Devin et al.[21] To achieve this, "task-specific" and "robot-specific" neural network policies are composed and then trained end to end. When an unseen combination is encountered, the appropriate "task-specific" and "robot-specific" previously trained policies are composed to solve the new robot-task combination. In our work, we focus on achieving the same behavior with different systems so we are not required to relearn for each different system.

In this work, we show the capabilities of the combined $\mathcal{L}_1$ adaptive control and ILC framework to achieve high-precision trajectory tracking even if (i) changing system dynamics and uncertain environment conditions are present, and (ii) learned trajectories are transferred between dynamically different systems.

We also show that a reference trajectory generated based on the reference model of the adaptive controller achieves more precise tracking performance than reference trajectories generated with standard choices. We use the serial framework[18] where the $\mathcal{L}_1$ adaptive controller acts as an underlying controller (see Figure 2) that makes the system display a repeatable and reliable behavior (in other words, it achieves the same output when the same reference input is applied) even in the presence of unknown disturbances and changing dynamics; however, perfect trajectory tracking is not achieved. After each iteration, the ILC improves the tracking performance of the now repeatable system using knowledge from previous iterations.

The $\mathcal{L}_1$ adaptive controller forces systems to follow a predefined behavior defined through a so-called reference model, even if the systems are dynamically different. Therefore, learned trajectories in one system can be transferred among dynamically different systems (that have an underlying $\mathcal{L}_1$ adaptive controller with the same reference model) to achieve

perfect tracking or to significantly decrease the initial tracking error in a different system (see Figure 1). Experimental results on two dynamically different quadrotors show that the proposed approach achieves high trajectory tracking performance despite the presence of unknown disturbances. Furthermore, we show that our approach allows us to train on a simulator or on a quadrotor and then transfer the learned trajectory to a dynamically different quadrotor and achieve a high-precision tracking performance even in the first iteration. The tracking performance achieved by our approach cannot, under changing dynamics, be achieved by baseline proportional-derivative (PD) and proportional-integral-derivative (PID) controllers combined with ILC. However, ILC has the limitation of not being able to generalize previously learned tasks to new unseen tasks. In future work, a linear map generated using prior knowledge from previously learned trajectories (see the work of Hamer et al[22]) could be used to achieve transfer learning between different robots and tasks.

The remainder of this paper is organized as follows. We define the problem in Section 2. The details of the proposed approach and proofs of key features are presented in Section 3. Section 4 shows our experimental results, including examples where learned trajectories are transferred between dynamically different systems. We compare our approach to two frameworks with standard underlying feedback controllers. Conclusions are provided in Section 5.

2 | PROBLEM STATEMENT

The objectives of this work are to achieve high-precision trajectory tracking (i) when changing system dynamics and uncertain environment conditions are present, (ii) in a new and dynamically different system by transferring the previously learned trajectories, and (iii) by calculating the initial reference input based on the model reference defined in the $\mathcal{L}_1$ adaptive controller.

For a given desired trajectory, the system optimizes its performance over multiple executions and, if required, transfers the learned trajectory to a dynamically different system that is able to achieve a similar optimized performance. Moreover, even if the system dynamics continue to change, there is no need to relearn.

We assume that the uncertain and changing dynamics ("System" block in Figure 2) can be described by a single-input–single-output (SISO) system (this approach can be extended to multi-input–multioutput (MIMO) systems as described in Section 3.1.5) identical to the work of Hovakimyan and Cao[5] for output feedback:

$$y_1(s) = A(s)\left(u(s) + d_{\mathcal{L}_1}(s)\right), \quad y_2(s) = \frac{1}{s}y_1(s), \tag{1}$$

where $y_1(s)$ and $y_2(s)$ are the Laplace transforms of the translational velocity $y_1(t)$ and position $y_2(t)$, respectively; $A(s)$ is a strictly-proper *unknown* transfer function that can be stabilized by a proportional-integral controller; $u(s)$ is the Laplace transform of the input signal; and $d_{\mathcal{L}_1}(s)$ is the Laplace transform of the disturbance signal defined as $d_{\mathcal{L}_1}(t) \triangleq f(t, y_1(t))$, where $f : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ is an *unknown* map subject to the assumption:

Assumption 1. (Global Lipschitz continuity)
There exist constants $L > 0$ and $L_0 > 0$ such that the following inequalities hold uniformly in t:

$$|f(t,v) - f(t,w)| \leq L|v - w|, \text{ and } |f(t,w)| \leq L|w| + L_0, \quad \forall v, w \in \mathbb{R}.$$

The system is tasked to track a desired position trajectory $y_2^*(t)$, which is defined over a finite-time interval and is assumed to be feasible with respect to the true dynamics of the $\mathcal{L}_1$-controlled system (red and blue boxes in Figure 2). This signal is discretized because the input of computer-controlled systems is sampled, and measurements are only available at fixed time intervals. We introduce the lifted representation (see the work of Gunnarson and Norrlöf[8]) for the desired trajectory $\mathbf{y}_2^* = (y_2^*(1), \ldots, y_2^*(N))$, the output of the plant $\mathbf{y}_2 = (y_2(1), \ldots, y_2(N))$, and the reference input $\mathbf{r}_2 = (r_2(1), \ldots, r_2(N))$, where $N < \infty$ is the number of discrete samples. The tracking performance criterion J is defined as:

$$J \triangleq \min_{\mathbf{e}} \mathbf{e}^T \mathbf{Q} \mathbf{e}, \tag{2}$$

where $\mathbf{e} = \mathbf{y}_2 - \mathbf{y}_2^*$ is the tracking error and $\mathbf{Q}$ is a positive definite matrix. In this way, the reference input $\mathbf{r}_2$ is updated to improve the trajectory tracking iteratively.

3 | METHODOLOGY

We consider two main subsystems: the extended $\mathcal{L}_1$ adaptive controller (red box in Figure 2) and the ILC (green box in Figure 2). The extended $\mathcal{L}_1$ adaptive controller is presented in Section 3.1, including proofs on its transient behavior when subjected to (dynamic) disturbances. Section 3.2 introduces the ILC and includes a remark on convergence. Section 3.3 discusses the transfer of learned trajectories between dynamically different systems.

3.1 | $\mathcal{L}_1$ adaptive feedback

The goal of the $\mathcal{L}_1$ adaptive controller in this framework is to make the system behave in a repeatable, predefined way, even when unknown and changing disturbances affect the system. A description of the extended $\mathcal{L}_1$ adaptive controller and transient behavior proofs are presented next.

The extended architecture used in this work is identical to the work of Michini and How[7] where the typical $\mathcal{L}_1$ adaptive output feedback controller for SISO systems acting on velocity[5] is nested within a proportional controller (see Figure 2). The outer-loop proportional controller enables the system to remain within certain position boundaries.

3.1.1 | Problem formulation

The $\mathcal{L}_1$ adaptive output feedback controller aims to design a control input $u(t)$ such that the output $y_2(t)$ tracks a bounded piecewise continuous reference input $r_2(t)$. We aim to achieve a desired closed-loop behavior where the output of the $\mathcal{L}_1$ adaptive controller $y_1(t)$, nested within a proportional feedback loop, tracks $r_1(t)$ according to a first-order reference dynamic system:

$$M(s) = \frac{m}{s+m}, \quad m > 0. \tag{3}$$

3.1.2 | Definitions and $\mathcal{L}_1$-norm condition

The system in (1) can be rewritten in terms of the reference system (3):

$$y_1(s) = M(s)\left(u(s) + \sigma(s)\right), \tag{4}$$

where uncertainties in $A(s)$ and $d_{\mathcal{L}_1}(s)$ are combined into σ:

$$\sigma(s) \triangleq \frac{(A(s) - M(s))u(s) + A(s)d_{\mathcal{L}_1}(s)}{M(s)}. \tag{5}$$

We consider a strictly-proper low-pass filter $C(s)$ (see Figure 2) with $C(0) = 1$ and a proportional gain $K \in \mathbb{R}^+$ such that:

$$H(s) \triangleq \frac{A(s)M(s)}{C(s)A(s) + (1 - C(s))M(s)} \quad \text{is stable,} \tag{6}$$

$$F(s) \triangleq \frac{1}{s + H(s)C(s)K} \quad \text{is stable,} \tag{7}$$

and the following $\mathcal{L}_1$-norm condition is satisfied:

$$\|G(s)\|_{\mathcal{L}_1} L < 1, \quad \text{where} \quad G(s) \triangleq H(s)(1 - C(s)), \tag{8}$$

where L is the Lipschitz constant defined in Assumption 1. The transfer function $H(s)$ helps to describe the relationship between $y_1(s)$ and $r_1(s)$ and between $y_1(s)$ and $d_{\mathcal{L}_1}(s)$. It is obtained from eqs. (4.90) to (4.92) in the work of Hovakimyan and Cao.[5] The transfer function $F(s)$ helps to describe the relationship between $y_2(s)$ and $r_2(s)$ and between $y_2(s)$ and $d_{\mathcal{L}_1}(s)$. To obtain $F(s)$, we substitute $r_1(s) = K(r_2(s) - y_2(s))$ into eq. (4.94) in the work of Hovakimyan and Cao[5] and use the result to solve for $y_2(s)$ in (1). Finally, the transfer function $G(s)$ describes the relationship between $y_2(s)$ and $d_{\mathcal{L}_1}(s)$.

To prove the bounded-input–bounded-output (BIBO) stability of a reference model, which describes the repeatable behavior of the extended $\mathcal{L}_1$ controlled system, the $\mathcal{L}_1$-norm condition is used. The solution of the $\mathcal{L}_1$-norm condition in (8) exists under the following assumptions.

Assumption 2. (Stability of $H(s)$)
The transfer function $H(s)$ is assumed to be stable for appropriately chosen low-pass filter $C(s)$ and first-order reference eigenvalue $-m < 0$.

This assumption holds when $A(s)$ can be stabilized by a proportional-integral controller.[5]

Assumption 3. (Stability of $F(s)$)
The transfer function $F(s)$ is assumed to be stable for appropriately chosen proportional gain K.

For this assumption to be valid, a sufficient condition is that $A(s)$ is minimum phase stable, which holds if there is a controller within the system $A(s)$ that is stabilizing the plant without any unstable zeros. This assumption is valid in the case of velocity control of a quadrotor.

3.1.3 | Extended $\mathcal{L}_1$ adaptive control architecture

The SISO extended $\mathcal{L}_1$ adaptive controller architecture is shown in Figure 2. This architecture (from r_1 to y_1) is identical to the work of Hovakimyan and Cao[5] with the exception of the proportional feedback loop. The integrator from y_1 to y_2 allows the outer loop to control position, while the $\mathcal{L}_1$ adaptive feedback controls the velocity. The equations that describe the implementation of the extended $\mathcal{L}_1$ output feedback architecture are presented as follows.

Output Predictor: The output predictor used within the $\mathcal{L}_1$ adaptive output feedback architecture is:

$$\dot{\hat{y}}_1(t) = -m\hat{y}_1(t) + m(u(t) + \hat{\sigma}(t)), \quad \hat{y}_1(0) = 0,$$

where $\hat{\sigma}(t)$ is the adaptive estimate of $\sigma(t)$. In the Laplace domain, this is:

$$\hat{y}_1(s) = M(s)(u(s) + \hat{\sigma}(s)). \tag{9}$$

Adaptation Law: The adaptive estimate $\hat{\sigma}$ is updated according to the following update law:

$$\dot{\hat{\sigma}}(t) = \Gamma \mathrm{Proj}\left(\hat{\sigma}(t), -\tilde{y}(t)\right), \quad \hat{\sigma}(0) = 0, \tag{10}$$

where $\tilde{y}(t) \triangleq \hat{y}_1(t) - y_1(t)$. The adaptation rate $\Gamma \in \mathbb{R}^+$ is subject to the lower bound specified in the work of Hovakimyan and Cao.[5] For a fast adaptation, Γ is set very large. The projection operator is defined in[5] and ensures that the estimation of σ is guaranteed to remain within a specified convex set, which contains all possible values of $d_{\mathcal{L}_1}(s)$ and the range of uncertainties in $A(s)$. Intuitively, this convex set includes all the values that σ in (4) could take.

Control Law: The control input is a low-pass filtered signal by $C(s)$ of the difference between the $\mathcal{L}_1$ desired trajectory r_1 and the adaptive estimate $\hat{\sigma}$:

$$u(s) = C(s)(r_1(s) - \hat{\sigma}(s)). \tag{11}$$

Hence, it only compensates for the low frequencies of the uncertainties within $A(s)$ and $d_{\mathcal{L}_1}$, which the system is capable of counteracting. The high-frequency portion is attenuated by the low-pass filter.

Closed-Loop Feedback: The objective of the closed-loop feedback is for y_2 to track r_2. It acts on the input to the $\mathcal{L}_1$ adaptive output feedback controller r_1 based on the output of the system y_1. From the aforementioned statement, $y_2(s) \triangleq \frac{1}{s} y_1(s)$ and the negative feedback is defined as:

$$r_1(s) = K(r_2(s) - y_2(s)). \tag{12}$$

3.1.4 | Transient and steady-state performance

The extended $\mathcal{L}_1$ adaptive controller guarantees that the difference between the output of a given BIBO stable reference system and the output of the actual system is uniformly bounded. In other words, the actual system behaves close to the reference system. Intuitively, the extended $\mathcal{L}_1$ adaptive controller makes the system perform repeatably and consistently.

We first introduce the following assumption necessary to prove uniform boundedness of the difference between the output of a given BIBO stable reference system and the output of the actual system.

Assumption 4. (Boundedness of $r_1(t)$)
The signal $r_1(t)$ is assumed to be a bounded piecewise continuous signal. Therefore, it has a bounded norm $\|r_1\|_{\mathcal{L}_\infty}$.

The aforementioned assumption is justifiable since the $\mathcal{L}_1$ adaptive controller makes the system behave close to the linear reference system $M(s)$. In other words, the low-pass filter, output predictor, adaptation law, and system (see Figure 2) behave close to the linear model $M(s)$. By choosing $\det(M(0)) > 0$ and using theorem 1 in the work of Morari,[23] we know that there exists a $K = kI$, with $k > 0$ such that the closed-loop system from r_2 to y_2 is stable; hence, $r_1(t)$ is bounded. The proof is part of future work as stability of integral controllers for nonlinear systems is an ongoing research topic (see, eg, the work of Konstantopoulos et al[24]) and outside the scope of the present work.

We then present the BIBO stable closed-loop reference system.

Lemma 1. *Let $C(S)$, $M(S)$, and K satisfy the $\mathcal{L}_1$-norm condition in (8). Then, the following closed-loop reference system:*

$$y_{1,\text{ref}}(s) = M(s)\left(u_{1,\text{ref}}(s) + \sigma_{\text{ref}}(s)\right), \tag{13}$$

$$u_{1,\text{ref}}(s) = C(s)\left(r_{1,\text{ref}}(s) - \sigma_{\text{ref}}(s)\right), \tag{14}$$

$$y_{2,\text{ref}}(s) = \frac{1}{s} y_{1,\text{ref}}(s), \tag{15}$$

$$r_{1,\text{ref}}(s) = K\left(r_2(s) - y_{2,\text{ref}}(s)\right), \tag{16}$$

where

$$\sigma_{\text{ref}}(s) = \frac{(A(s) - M(s))u_{1,\text{ref}}(s) + A(s)d_{1,\text{ref}}(s)}{M(s)}, \tag{17}$$

and $d_{1,\text{ref}}(s)$ is the Laplace transform of $d_{1,\text{ref}}(t) \triangleq f(t, y_{1,\text{ref}}(t))$ is BIBO stable.

The proof of this lemma is found in Appendix A. Next, we show that error in the estimation is bounded and that the system behaves close to the BIBO stable reference system.

Theorem 1. *Consider the system in (1), with a control input from the extended $\mathcal{L}_1$ output feedback adaptive controller defined in (9) to (12). Suppose $C(s)$, $M(s)$, and K satisfy the $\mathcal{L}_1$-norm condition in (8). Then, the following bounds hold:*

$$\|\tilde{y}\|_{\mathcal{L}_\infty} \leq \gamma_0, \tag{18}$$

$$\|y_{2,\text{ref}} - y_2\|_{\mathcal{L}_\infty} \leq \gamma_1, \tag{19}$$

where $\tilde{y}(t) \triangleq \hat{y}_1(t) - y_1(t)$, $\gamma_0 \propto \sqrt{\frac{1}{\Gamma}}$ is defined in the work of Hovakimyan and Cao[5] and

$$\gamma_1 \triangleq \left(\|F(s)G(s)\|_{\mathcal{L}_1} L \frac{\|H_2(s)\|_{\mathcal{L}_1}}{1 - \|G(s)\|_{\mathcal{L}_1} L} + \left\| \frac{F(s)H(s)C(s)}{M(s)} \right\|_{\mathcal{L}_1} \right) \gamma_0. \tag{20}$$

The proof of this theorem is found in Appendix B.

The difference between the output predictor and the system output $y_1(t)$ and the difference between the reference system and the system output $y_2(t)$ are uniformly bounded with bounds inversely proportional to the square root of the adaptation gain Γ. For high adaptation gains, the actual system approaches the behavior of the reference system (A4). Hence, the system achieves repeatable and consistent performance, which is required for ILC.

3.1.5 | Multi-input multi-output implementation

The SISO architecture derived so far can be extended to a MIMO implementation. In our application, it can be assumed that states are decoupled (after applying an appropriate feedback linearization). Hence, for n different states, the low-pass filter $C(s)$ and the first-order output predictor (9) are implemented as $(n \times n)$ diagonal transfer function matrices:

$$C(s) = \text{diag}(C_1(s), \ldots, C_n(s)), \; M(s) = \text{diag}(M_1(s), \ldots, M_n(s)),$$

where $C_i(s) = \frac{\omega_i}{s+\omega_i}$, $M_i(s) = \frac{m_i}{s+m_i}$, and $i = 1, \ldots, n$. Moreover, the proportional gain K is implemented as an $(n \times n)$ matrix:

$$K = \text{diag}(k_1, \ldots, k_n),$$

where $k_i \in \mathbb{R}^+$.

3.2 | Iterative learning control

In this work, we use the extended $\mathcal{L}_1$ adaptive controller to achieve a repeatable system, even in the presence of disturbances, and ILC to improve tracking performance of the resulting repeatable system. We assume we have an approximate model of the repeatable system (orange dashed line in Figure 2):

$$\dot{x}(t) = f(x(t), r_2(t)) \quad y_2(t) = h(x(t)), \tag{21}$$

where $r_2(t) \in \mathbb{R}$ is the control input, $x(t) \in \mathbb{R}^{n_x}$ is the state, and $y_2(t) \in \mathbb{R}$ is the output. In order to be consistent with the assumptions made in Section 3.1, we have $r_2(t)$ and $y_2(t) \in \mathbb{R}$; however, the approach described in this section can be extended to MIMO systems as described in Section 3.2.1. We assume that the system states can be directly measured or observed from the output. In many control applications, constraints must be placed on the process variables to ensure safe and smooth operations. The system may be subjected to input or output constraints of the form:

$$V_c y_2(t) \leq y_{2,\max}, \quad Z_c r_2(t) \leq r_{2,\max}, \tag{22}$$

where V_c and Z_c are matrices of appropriate size that can represent lower and upper limits. Iterative learning control seeks to update the feedforward signal $r_2(t)$ based on data gathered during previous iterations. The ILC implementation in this work is based on the work of Schoellig et al.[13]

The goal is to track a desired trajectory $y_2^*(t)$ over a finite-time interval. The desired output trajectory is assumed to be feasible based on the nominal model (21) and the constraints in (22), ie, there exist nominal reference state and output trajectories $(r_2^*(t), x^*(t), y_2^*(t))$ that satisfy (21) and (22). We also assume that the system stays close to the reference trajectory; hence, we only consider small deviations from the aforementioned nominal trajectories, ie, $\tilde{r}_2(t)$, $\tilde{x}(t)$, and $\tilde{y}_2(t)$, respectively. The system is linearized about the nominal trajectory to obtain a time-varying linear state-space model, which approximates the system dynamics along the reference trajectory. This system is then discretized and written as:

$$\tilde{x}(k+1) = A(k)\tilde{x}(k) + B(k)\tilde{r}_2(k), \tag{23}$$

where $k \in \{0, 1, \ldots, N-1\}$, $N < \infty$, represents the discrete-time index.

Using the lifted representation introduced in Section 2, we define $\bar{\mathbf{y}}_{2,j} = (\tilde{y}_2(1), \ldots, \tilde{y}_2(N)) \in \mathbb{R}^N$ and $\bar{\mathbf{r}}_{2,j} = (\tilde{r}_2(0), \ldots, \tilde{r}_2(N-1)) \in \mathbb{R}^N$ and write the extended system as:

$$\bar{\mathbf{y}}_{2,j} = \mathbf{F}_{\text{ILC}}\bar{\mathbf{r}}_{2,j} + \mathbf{d}_\infty, \tag{24}$$

where the subscript j represents the iteration number, $\mathbf{F}_{\text{ILC}}$ is a constant matrix derived from the discretized model (23) as described in the work of Schoellig et al,[13] and $\mathbf{d}_\infty$ represents a repetitive disturbance that is initially unknown but is identified during the learning process. The constraints of the system can be written in the lifted representation accordingly:

$$\mathbf{V}_c\bar{\mathbf{y}}_{2,j} \leq \bar{\mathbf{y}}_{2,\max}, \quad \mathbf{Z}_c\bar{\mathbf{r}}_{2,j} \leq \bar{\mathbf{r}}_{2,\max},$$

where $\mathbf{V}_c$ and $\mathbf{Z}_c$ are matrices of appropriate size.

We follow the learning approach presented in the works of Mueller et al[12] and Schoellig et al[13] for a single system. An iteration-domain Kalman filter for the system (24) is used to compute the estimate $\hat{\mathbf{d}}_j$ based on measurements from iterations $1, \ldots, j$. The disturbance estimate is obtained from a Kalman filter based on the following model:

$$\begin{aligned} \mathbf{d}_{j+1} &= \mathbf{d}_j + \omega_j \\ \bar{\mathbf{y}}_{2,j} &= \mathbf{F}_{\text{ILC}}\bar{\mathbf{r}}_{2,j} + \mathbf{d}_j + \mu_j, \end{aligned} \tag{25}$$

where $\omega_j \sim \mathcal{N}(0, \mathbf{E}_j)$ and $\mu_j \sim \mathcal{N}(0, \mathbf{H}_j)$. The covariances $\mathbf{E}_j$ and $\mathbf{H}_j$ may be regarded as design parameters to adapt the learning rate of the algorithm. A common choice are diagonal covariances, such that $\mathbf{E}_j = \eta\mathbf{I}$ and $\mathbf{H}_j = \epsilon\mathbf{I}$, where $\eta, \epsilon \in \mathbb{R}$, and $\mathbf{I}$ represent an identity matrix of appropriate size. The estimation equations are:

$$\hat{\mathbf{y}}_{j|j-1} = \mathbf{F}_{\text{ILC}}\bar{\mathbf{r}}_{2,j} + \hat{\mathbf{d}}_{j-1|j-1}, \tag{26}$$

where

$$\hat{\mathbf{d}}_{j|j} = \hat{\mathbf{d}}_{j-1|j-1} + \mathbf{K}_j\left(\bar{\mathbf{y}}_{2,j} - \hat{\mathbf{y}}_{j|j-1}\right), \tag{27}$$

and $\mathbf{K}_j$ is the optimal Kalman gain.

An update step, based on the optimization of a cost function, computes the next input sequence $\bar{\mathbf{r}}_{2,j+1}$ that compensates for the identified disturbance $\hat{\mathbf{d}}_{j|j}$ and estimated output error $\hat{\mathbf{y}}_{j+1|j}$ in the following way:

$$J(\bar{\mathbf{r}}_{2,j+1}) = \min_{\bar{\mathbf{r}}_{2,j+1} \in \hat{\Omega}_{j+1}} \left[\hat{\Phi}_{j+1} \triangleq \frac{1}{2}\left\{\hat{\mathbf{y}}_{j+1|j}^T\mathbf{Q}\hat{\mathbf{y}}_{j+1|j} + \bar{\mathbf{r}}_{2,j+1}^T\mathbf{W}\bar{\mathbf{r}}_{2,j+1}\right\}\right], \tag{28}$$

subject to

$$\mathbf{V}_c\hat{\mathbf{y}}_{j+1|j} \leq \hat{\mathbf{y}}_{\max}, \quad \mathbf{Z}_c\bar{\mathbf{r}}_{2,j} \leq \bar{\mathbf{r}}_{2,\max}, \tag{29}$$

where $\mathbf{V}_c$ and $\mathbf{Z}_c$ are matrices of appropriate size and $\hat{\mathbf{y}}_{j+1|j}$ is defined in (26). The set $\hat{\Omega}_{j+1}$ is a convex set defined by the constraints in (29). The constant matrix $\mathbf{Q}$ is symmetric positive definite, and the constant matrix $\mathbf{W}$ is symmetric positive semidefinite and both weight different components of the cost function. The cost function tries to minimize the tracking error of the system (weighted by $\mathbf{Q}$) and a function of the control effort (weighted by $\mathbf{W}$). The resulting convex optimization problem can be solved very efficiently with state-of-the-art optimization libraries. A common approach is to define the weighting matrix as $\mathbf{W} = w\mathbf{I}$, where $w \in \mathbb{R}$ and $\mathbf{I}$ represents an identity matrix of appropriate size. In Section 2, we defined the cost function (2), which tried to minimize the error $\mathbf{e}$. In Equation (28), we specify a cost function that tries to minimize the estimate $\hat{\mathbf{y}}_{j+1,j}$ of the error $\mathbf{e}$ and further ensures that a smooth and executable reference input is obtained as a result of the optimization process.

To prove the asymptotic zeroing of the tracking error under the constrained optimization-based ILC, we make the following assumptions.

Assumption 5. (Rank of $\mathbf{F}_{\text{ILC}}$)
The matrix $\mathbf{F}_{\text{ILC}}$ has full row rank.

If $\mathbf{F}_{\text{ILC}}$ does not have full row rank, a projection operator onto the image space of $\mathbf{Q}^{\frac{1}{2}}\mathbf{F}_{\text{ILC}}$ must be introduced to prove convergence of the controllable part of the system.[25]

Assumption 6. (Input constraints)
Given the input constraints in (29), reference trajectory $\mathbf{y}_2^* = (y_2^*(1), \ldots, y_2^*(N)) \in \mathbb{R}^N$, and the actual steady-state disturbance $\mathbf{d}_\infty$, the zeroing of the error is possible with an input $\bar{\mathbf{r}}_{2,\infty}$ in the feasible set. We further assume that the active equality constraints are defined such that $[\mathbf{V}_{c,\text{act}}\mathbf{F}_{\mathbf{ILC}}\mathbf{Z}_{c,\text{act}}]^T$ is full rank.

In other words, there exists $\bar{\mathbf{r}}_{2,\infty}$ such that $\mathbf{F}_{\mathbf{ILC}}\bar{\mathbf{r}}_{2,\infty} + \mathbf{d}_\infty = 0$. In addition, (29) holds.

Remark 1. Under Assumptions 5 and 6, system (24) converges to the global minimum under the Kalman filter–based, constrained optimization ILC with (28) and (29).

A discussion of Remark 1 is found in Appendix C.

3.2.1 | Multi-input multi-output implementation

The SISO architecture derived so far can be extended to a MIMO implementation. We make use of the assumption that states are decoupled (after applying an appropriate feedback linearization). Hence, the control input is a vector $\mathbf{r}_{2,j} \in \mathbb{R}^{n_u}$ and the output is a vector $\mathbf{y}_{2,j} \in \mathbb{R}^{n_y}$. Moreover, the matrices A and B are implemented as:

$$A = \text{diag}(A_1, \ldots, A_n), \quad B = \text{diag}(B_1, \ldots, B_n).$$

In the lifted representation, we define the input and the output as:

$$\bar{\mathbf{r}}_{2,j} = \left(r_{2,1}(0), \ldots, r_{2,n_u}(0), \ldots, r_{2,1}(N-1), \ldots, r_{2,n_u}(N-1)\right)$$
$$\bar{\mathbf{y}}_{2,j} = \left(y_{2,1}(1), \ldots, y_{2,n_y}(1), \ldots, y_{2,1}(N), \ldots, y_{2,n_y}(N)\right),$$

respectively. We modify $\mathbf{F}_{\text{ILC}}$ accordingly. Finally, we redefine the weighing matrices in the cost function as:

$$\mathbf{Q} = \text{diag}\left(\text{diag}\left(q_1, \ldots, q_{n_y}\right), \ldots, \text{diag}\left(q_1, \ldots, q_{n_y}\right)\right)$$
$$\mathbf{R} = \text{diag}\left(\text{diag}\left(r_1, \ldots, r_{n_u}\right), \ldots, \text{diag}\left(r_1, \ldots, r_{n_u}\right)\right)$$
$$\mathbf{S} = \text{diag}\left(\text{diag}\left(s_1, \ldots, s_{n_u}\right), \ldots, \text{diag}\left(s_1, \ldots, s_{n_u}\right)\right).$$

3.3 | Transfer learning

The purpose of transfer learning is to exchange learned trajectories between dynamically different systems and achieve a performance comparable to the one obtained from learning in the training system. The $\mathcal{L}_1$ adaptive controller makes two systems behave in a repeatable predefined way, even under unknown and changing disturbances. Therefore, learned trajectories can usually be exchanged without any modification when the underlying reference model (3) is the same for both systems. The learned input $\bar{\mathbf{r}}_{2,j}$ on the training system can be transfered without any modifications to the new

system. In order to allow the new system to continue learning after the initial transfer, we need to provide the ILC with an initial estimate of the repetitive disturbance $\mathbf{d}_j$, which is also obtained from the training system without any modifications. Equations (25) to (29) compute the next input sequence $\bar{\mathbf{r}}_{2.j+1}$ such that the system continues learning.

4 | EXPERIMENTAL RESULTS

This section shows the experimental results of the proposed framework combining $\mathcal{L}_1$ adaptive control and ILC ($\mathcal{L}_1$-ILC) applied to quadrotors for high-precision trajectory tracking. We compare the performance of the proposed framework to the performance of two baseline controllers: a PD (proportional-derivative) controller combined with ILC (PD-ILC) and a PID (proportional-integral-derivative) controller combined with ILC (PID-ILC). We consider four scenarios:

1. learning under unknown and changing disturbances,
2. transfer learning between dynamically different systems,
3. transfer learning from simulation to real-world experiments, and
4. initializing the robot learning with a reference input generated based on the $\mathcal{L}_1$ adaptive controller reference model.

Section 4.1 describes the experimental setup, introduces the two quadrotors used in this study, and compares their dynamical behavior under $\mathcal{L}_1$, PD, and PID control. Section 4.2 discusses the tracking performance under changing conditions. Section 4.3 focuses on the transferability of learned trajectories between dynamically different quadrotors. The transferability from simulation to real world is assessed in Section 4.4, and Section 4.5 discusses the ability to compute the initial reference input, assuming that the system behaves as the $\mathcal{L}_1$ reference model.

4.1 | Experimental setup

The vehicles used in the experiments are the Parrot AR.Drone 2.0 and the Parrot Bebop 2 (see Figure 3). The signals $r_1(t)$, $r_2(t)$, $y_1(t)$, and $y_2(t)$ in Figure 2 are the desired translational velocity, desired position, quadrotor translational velocity, and quadrotor position, respectively. We implement a MIMO extended $\mathcal{L}_1$ adaptive controller for position control as described in Section 3.1.5 where we assume that the x, y, and z directions are decoupled. A central overhead motion capture camera system provides velocity, roll-pitch-yaw Euler angles, and rotational velocity measurements. The output of the extended $\mathcal{L}_1$ adaptive controller $\mathbf{u}(t) = (u_{\mathsf{x}}(t), u_{\mathsf{y}}(t), u_{\mathsf{z}}(t))$, commanded x and y translational acceleration, and commanded z velocity, respectively, is specified in the global coordinate frame. However, the interface to the real quadrotor ("Plant" in Figure 2) requires commanded roll (ϕ_{des}), pitch (θ_{des}), vertical velocity ($\dot{z}_{\text{des}}$), and rotational velocity around the z axis (ω_{z}) (see the work of Powers et al[26]). Therefore, the signal $\mathbf{u}(t)$ is transformed through the following nonlinear transformation:

$$\begin{aligned}
\phi_{\text{des}} &= -\arcsin\left(-u_{\mathsf{x}}\sin(\psi) + u_{\mathsf{y}}\cos(\psi)\right)\\
\theta_{\text{des}} &= \arcsin\left(u_{\mathsf{x}}\cos(\psi) + u_{\mathsf{y}}\sin(\psi)\right)\\
\dot{z}_{\text{des}} &= u_{\mathsf{z}},
\end{aligned}$$

FIGURE 3 Vehicles used in the experiments. On the left the Bebop 2, on the right the AR.Drone 2.0 [Colour figure can be viewed at wileyonlinelibrary.com]

TABLE 1 Parameters used in the extended $\mathcal{L}_1$ adaptive controller

Parameter	Value
Γ	5000
m_{x}	-1.1
m_{y}	-1.1
m_{z}	-1.75
k_{x}	0.4
k_{y}	0.4
k_{z}	0.4

TABLE 2 Drone-dependent $\mathcal{L}_1$ adaptive controller parameters for the low-pass filter $C(s) = \text{diag}(\frac{\omega_{\mathsf{x}}}{s+\omega_{\mathsf{x}}}, \frac{\omega_{\mathsf{y}}}{s+\omega_{\mathsf{y}}}, \frac{\omega_{\mathsf{z}}}{s+\omega_{\mathsf{z}}})$

Parameter	AR.Drone 2.0	Bebop 2
ω_{x}	3.5	23
ω_{y}	3.5	23
ω_{z}	3.5	3.8

where ψ is the current yaw angle. During the experiment, the desired yaw angle (u_ψ) is set to zero and controlled through a simple proportional controller $u_{\omega_{\mathsf{z}}} = k_\psi(u_\psi - \psi)$, where k_ψ is the control gain.

We implement three different position controllers for comparison. For the extended $\mathcal{L}_1$ adaptive controller, the controller parameters for the adaption rate Γ; reference model eigenvalues m_{x}, m_{y}, and m_{z}, respectively; and gain matrix K are given in Table 1. We choose a first-order low-pass filter $C(s) = \text{diag}(\frac{\omega_{\mathsf{x}}}{s+\omega_{\mathsf{x}}}, \frac{\omega_{\mathsf{y}}}{s+\omega_{\mathsf{y}}}, \frac{\omega_{\mathsf{z}}}{s+\omega_{\mathsf{z}}})$. The low-pass filter is tuned for each quadrotor separately; the parameters are given in Table 2.

We compare the performance of the proposed $\mathcal{L}_1$-ILC approach with that of PD-ILC and PID-ILC. The PD controller is given by:

$$u_i(t) = \frac{2\zeta}{\tau_i}(\dot{r}_{2,i}(t) - y_{1,i}(t)) + \frac{1}{\tau_i^2}\left(r_{2,i}(t) - y_{2,i}(t)\right), \quad \text{for } i = \mathsf{x}, \mathsf{y}, \mathsf{z}, \tag{30}$$

where τ_i and ζ are the time constant and damping coefficient, respectively. The PID controller is given by:

$$u_i(t) = \alpha(\dot{r}_{2,i}(t) - y_{1,i}(t)) + \beta(r_{2,i}(t) - y_{2,i}(t)) + \gamma \int (r_{2,i}(t) - y_{2,i}(t))\mathrm{d}t, \quad \text{for } i = \mathsf{x}, \mathsf{y}, \mathsf{z}, \tag{31}$$

where α, β, and γ are the controller gains, which could be defined as in the work of Schoellig et al,[27] ie, $\alpha = \tau_i(1 + 2\zeta)$, $\beta = \tau_i^2(1 + 2\zeta)$, and $\gamma = \tau_i^3$.

In this application, constraints are imposed on the input acceleration as it is intimately related to the physical capabilities of the actuators of the system and are expressed through the following mathematical inequality:

$$\ddot{\mathbf{r}}^{\mathbf{low}} \leq \ddot{\bar{\mathbf{r}}}_{2,j+1} \leq \ddot{\mathbf{r}}^{\mathbf{hi}}, \tag{32}$$

where the sequence $\ddot{\bar{\mathbf{r}}}_{2,j+1}$ represents the discrete approximation of the second derivative of the input reference. The aforementioned constraint can be rearranged as linear inequality with respect to $\bar{\mathbf{r}}_{2,j+1}$. Under the assumption that $\ddot{\bar{r}}_{2,j+1}(N) = \ddot{\bar{r}}_{2,j+1}(N-1)$, $\ddot{\bar{\mathbf{r}}}_{2,j+1}$ can be written as:

$$\ddot{\bar{\mathbf{r}}}_{2,j+1} = \begin{bmatrix} \left(\bar{r}_{2,j+1}(2) - 2\bar{r}_{2,j+1}(1) + \bar{r}_{2,j+1}(0)\right)/(\Delta t)^2 \\ \left(\bar{r}_{2,j+1}(3) - 2\bar{r}_{2,j+1}(2) + \bar{r}_{2,j+1}(1)\right)/(\Delta t)^2 \\ \vdots \\ \left(\bar{r}_{2,j+1}(N) - 2\bar{r}_{2,j+1}(N-1) + \bar{r}_{2,j+1}(N-2)\right)/(\Delta t)^2 \end{bmatrix} = \mathbf{D}\ddot{\bar{\mathbf{r}}}_{2,j+1}, \tag{33}$$

where

$$\mathbf{D} = \begin{bmatrix} 1/(\Delta t)^2 & -2/(\Delta t)^2 & 1/(\Delta t)^2 & 0 & \cdots & 0 \\ 0 & 1/(\Delta t)^2 & -2/(\Delta t)^2 & 1/(\Delta t)^2 & \ddots & 0 \\ \vdots & & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & \cdots & 1/(\Delta t)^2 & -2/(\Delta t)^2 & 1/(\Delta t)^2 \end{bmatrix}, \tag{34}$$

and Δt is the time interval between discrete samples.

In this implementation, we define a cost function that minimizes the estimated error $\hat{\mathbf{y}}_{j+1|j}$ while achieving a smooth input with the minimum control effort $\bar{\mathbf{r}}_{2,j+1}$. Hence, we include the estimated error, the control effort, and input accelerations in the cost function of this implementation:

$$J(\bar{\mathbf{r}}_{2,j+1}) = \min_{\bar{\mathbf{r}}_{2,j+1}\in\hat{\Omega}_{j+1}} \left[\hat{\Phi}_{j+1} \triangleq \frac{1}{2}\left\{\hat{\mathbf{y}}_{j+1|j}^T \mathbf{Q}\hat{\mathbf{y}}_{j+1|j} + \bar{\mathbf{r}}_{2,j+1}^T \mathbf{W}\bar{\mathbf{r}}_{2,j+1}\right\}\right], \tag{35}$$

subject to (32), where $\mathbf{Q} = \mathbf{I}$. Moreover, we define $\mathbf{W} = \mathbf{R} + \mathbf{D}^T\mathbf{S}\mathbf{D}$ to penalize control effort (weighted by $\mathbf{R} = r\mathbf{I}$ with $r = 0.001$) and the acceleration of the reference signal (weighted by $\mathbf{S} = s\mathbf{I}$ with $s = 0.0025$). We use the IBM CPLEX optimizer to solve the aforementioned optimization problem. Using this definition, it can be shown that $\mathbf{W}$ is symmetric positive definite.

If the constraints (32) are inactive, according to Remark 1, system (24) converges to the global minimum under the Kalman filter–based constrained optimization ILC approach. However, if constraints are active, they are included in the Lagrangian in the following way:

$$\begin{aligned}\mathcal{L}(\bar{\mathbf{r}}_{2,j}, \lambda_1, \lambda_2) = \frac{1}{2}\left\{\left(\mathbf{F}_{\text{ILC}}\bar{\mathbf{r}}_{2,j+1} + \hat{\mathbf{d}}_{j|j}\right)^T \mathbf{Q}\left(\mathbf{F}_{\text{ILC}}\bar{\mathbf{r}}_{2,j+1} + \hat{\mathbf{d}}_{j|j}\right) + \bar{\mathbf{r}}_{2,j+1}^T\mathbf{W}\bar{\mathbf{r}}_{2,j+1}\right\} \\ - \sum\nolimits_{l\in M}\lambda_{1,l}\left(\mathbf{D}_l\bar{\mathbf{r}}_{2,j+1} - \ddot{r}^{\text{hi}}\right) - \sum\nolimits_{l\in P}\lambda_{2,l}\left(\mathbf{D}_l\bar{\mathbf{r}}_{2,j+1} - \ddot{r}^{\text{low}}\right)\end{aligned}$$

where $\mathbf{D}_l$ is the l^{th} row of (34) that corresponds to an active constraint and $\lambda_{1,l}$ and $\lambda_{2,l}$ are Lagrange multipliers for the set M of maximum acceleration and the set P of minimum acceleration active constraints. Assumption 6 holds because, at any given point in the trajectory, only one set of constraints, either minimum or maximum, can be active. Hence, $M \cap P = \{0\}$ and $\mathbf{Z}_{c,\text{act}}$, the matrix whose rows are the rows of $\mathbf{D}$ that correspond to the active constraints, is full rank. We can conclude then that $\bar{\mathbf{r}}_{2,j+1}$ is the unique global solution to the minimization problem.

To show that the two quadrotors have different dynamical behavior, we use the same controller gains for the PD and PID controller for both quadrotors. Each of the two quadrotors is tasked to track a 3-dimensional straight line reference trajectory using the PD, PID, and $\mathcal{L}_1$ controller. Figure 4 compares the time response in x direction of the two quadrotors for each controller. The dynamical difference between the AR.Drone 2.0 and Bebop 2 are significant for both the PD and PID controller, while using the extended $\mathcal{L}_1$ adaptive controller, both drones behave similarly and close to the $\mathcal{L}_1$ model reference system. This confirms that the $\mathcal{L}_1$ adaptive controller framework implemented as an underlying controller enforces the same dynamic behavior on dynamically different systems. It is also interesting to observe that, for repeated

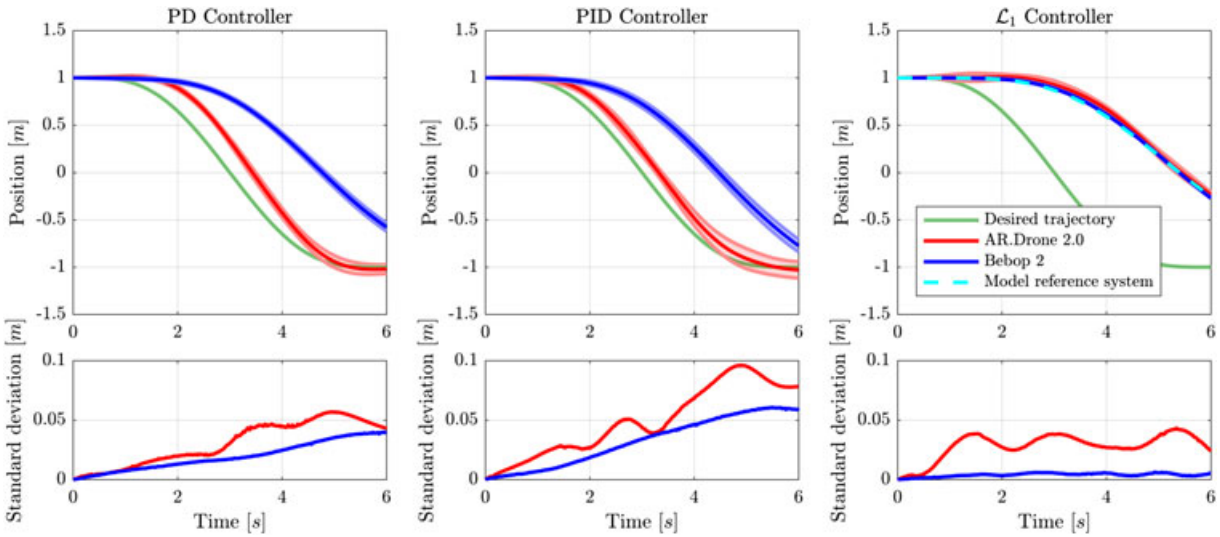

FIGURE 4 Time response in x direction for the AR.Drone 2.0 and Bebop 2 for a given reference trajectory using three different controllers: left: proportional-derivative (PD), middle: proportional-integral-derivative (PID), right: $\mathcal{L}_1$. The response of the model reference system of the $\mathcal{L}_1$ controller is depicted in the plot of the $\mathcal{L}_1$ controller. The line in the top figures denotes the mean over 5 repetitions, and the envelope denotes the standard deviation. The bottom figures show the standard deviation over time. It can be seen that, with the use of the PD and PID controller, the drones have different dynamic behavior, whereas the $\mathcal{L}_1$ adaptive controller forces the systems to behave as the model reference system [Colour figure can be viewed at wileyonlinelibrary.com]

experiments, the standard deviation (Figure 4, lower row) over time stays constant for the $\mathcal{L}_1$ controller and increases with time for the PD and PID controller. This shows that the $\mathcal{L}_1$ controller renders a more repeatable system overall.

To quantify the performance of the ILC, an average position error along the trajectory is defined as:

$$e = \frac{\sum_{i=1}^{N} \sqrt{(r_{2,x}(i) - y_{2,x}(i))^2 + (r_{2,y}(i) - y_{2,y}(i))^2 + (r_{2,z}(i) - y_{2,z}(i))^2}}{N}. \tag{36}$$

4.2 | Learning under disturbance

To assess the performance under changing conditions, an external wind is introduced as a disturbance. This wind is generated by a fan placed on the floor, blowing wind in the direction perpendicular to the trajectory path of the quadrotor.

In this experiment, the AR.Drone 2.0 learns to track a desired trajectory (same diagonal trajectory as in Section 4.1) using each of the three frameworks, ie, PD-ILC, PID-ILC, and $\mathcal{L}_1$-ILC. This experiment is repeated five times, and the mean of the tracking error as defined in (36) for this initial learning process (iterations 1 to 10) is depicted in Figure 5; the average standard deviation (average over iterations 1 to 10) during this initial learning process is given in Table 3. The proposed $\mathcal{L}_1$-ILC shows a higher position error during the first iteration, which is expected since the model

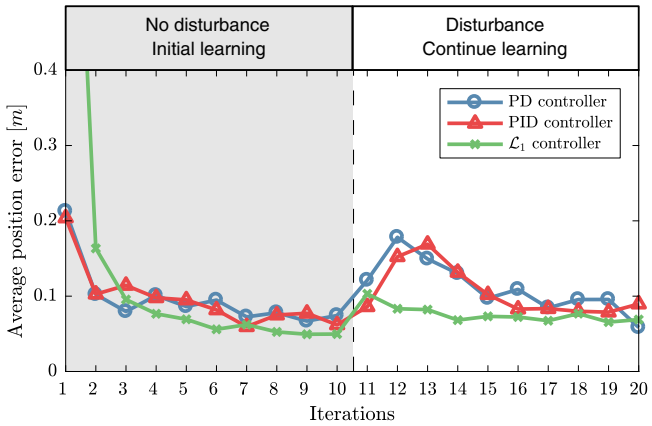

FIGURE 5 Mean of the average position error (36) over five separate learning experiments using the proportional-derivative (PD)–iterative learning control (ILC), proportional-integral-derivative (PID)–ILC, and $\mathcal{L}_1$-ILC frameworks. No disturbance is applied in iterations 1 to 10. After iteration 10, an external disturbance (wind) is applied and learning is continued for iterations 11 to 20. The initial value of the error using the $\mathcal{L}_1$ controller is 0.99 [m] in iteration 1, which is larger than the PD and PID controllers because of the relatively slow model reference system. After the wind disturbance is applied, the PD-ILC and PID-ILC frameworks must relearn and show a significantly larger error in iteration 11 than the $\mathcal{L}_1$-ILC setup. Both the PD and PID controllers show that the error increases within the first 3 iterations after the disturbance is applied, caused by the dynamical changes for which the ILC is not tuned; the $\mathcal{L}_1$ controller compensates for these dynamical changes and converges quickly [Colour figure can be viewed at wileyonlinelibrary.com]

TABLE 3 Average standard deviation of the tracking error over iterations. The full learning experiment (20 iterations in total) was repeated five times for each framework, ie, proportional-derivative (PD)–iterative learning control (ILC), proportional-integral-derivative (PID)–ILC, and $\mathcal{L}_1$-ILC

	Average Standard Deviation [m]	
	No Disturbance	**Disturbance**
PD-ILC	0.0167	0.0210
PID-ILC	0.0177	0.0182
$\mathcal{L}_1$-ILC	0.0130	0.0128

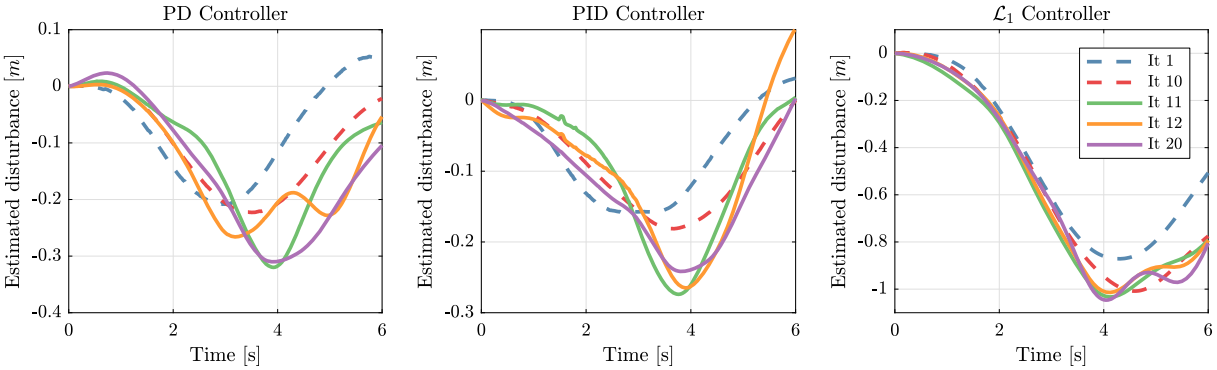

FIGURE 6 Disturbance estimate for y direction obtained from the Kalman filter using the proportional-derivative (PD) (left), proportional-integral-derivative (PID) (middle), and $\mathcal{L}_1$ (right) controller. In iterations 1 and 10 (dashed lines), no external disturbance is applied, and in iterations 11, 12, and 20 (solid lines), the external wind disturbance is applied. Note that the scale for each controller is different. For the PD–iterative learning control (ILC) and PID-ILC, we can see a significantly larger change in estimated disturbance after the external wind disturbance is applied compared with the $\mathcal{L}_1$-ILC setup [Colour figure can be viewed at wileyonlinelibrary.com]

reference system is slow (see Figure 4). From iteration 4, the $\mathcal{L}_1$-ILC shows lower errors consistently. It has also the highest repeatability (ie, lowest standard deviation over different learning experiments); see Table 3. There may be PID gains that improve the performance of the PID over the PD controller in Figure 5; however, we do not expect fundamental differences in the results.

After this initial learning process, an external wind disturbance is applied in iterations 11 to 20, and the ILC continues learning. While all frameworks show an increase in error in iteration 11, the $\mathcal{L}_1$-ILC setup exhibits only a minor increase and quickly adapts to the new conditions (within two to three iterations). The PD-ILC shows the largest increase. Because of the change in dynamics caused by the disturbance, the model of the ILC is not representing the real system anymore; therefore, the error increases in iterations 12 and 13 for the PD-ILC and PID-ILC where the $\mathcal{L}_1$ controller is capable of adapting to this change of dynamics. Figure 6 depicts the Kalman filter estimated disturbance $\hat{d}_j$ for the y direction. It can be seen that the disturbance is overestimated in iterations 11 and 12 when using the PD and PID controller, causing the error to increase in the next iteration. When using the $\mathcal{L}_1$ controller, the estimated disturbance in the ILC component does not change much after applying the external wind disturbance since the underlying $\mathcal{L}_1$ controller compensates for the change in dynamics. Overall, the three frameworks converge to a slightly higher average tracking error (iterations 17 to 20) due to the fact that the wind disturbance is partially nonrepetitive (or noisy); learning is only able to compensate for systematic disturbances. Table 3 shows that the variance significantly increases when the external wind disturbance is applied within the PD-ILC framework while there is little or no increase when using the PID-ILC and $\mathcal{L}_1$-ILC framework, respectively.

4.3 | Transfer learning between dynamically different systems

In this experiment, we assess the performance of transfer learning between dynamically different systems. In an initial learning phase, both the AR.Drone 2.0 and Bebop 2 quadrotors learn over 10 iterations with the PD-ILC, PID-ILC, and $\mathcal{L}_1$-ILC framework where both quadrotors use the same underlying model reference system (3). Convergence of the error for the first 10 iterations for the AR.Drone 2.0 and Bebop 2 under each control framework is shown in Figure 7. After iteration 10, the learned trajectory is transferred from AR.Drone 2.0 to Bebop 2, and vice versa. Learning is continued in iterations 11 to 20. The increase in tracking error after transfer learning is shown in Table 4. The $\mathcal{L}_1$-ILC approach shows only a marginal increase in error, while the PD-ILC and PID-ILC approaches show a significant increase in error. Moreover, it can be noted that transferring the learned trajectory from a system with a low variance (Bebop 2) to a system with a larger variance (AR.Drone 2.0) increases the error. This result shows that, in the $\mathcal{L}_1$-ILC case, the learned knowledge can be transferred to a dynamically different second system; the second system must be controlled by a corresponding underlying $\mathcal{L}_1$ adaptive controller with the same reference model. More generally, this proves the potential of the $\mathcal{L}_1$-ILC method to significantly speed up learning as one robot can learn from the other.

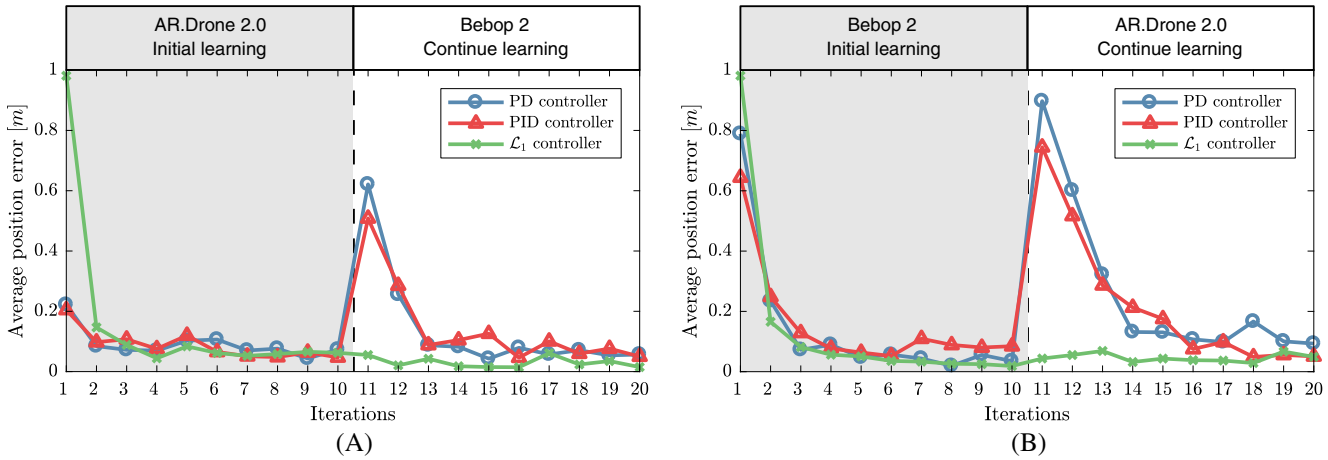

FIGURE 7 Learning behavior of the proportional-derivative (PD)–iterative learning control (ILC), proportional-integral-derivative (PID)–ILC and $\mathcal{L}_1$-ILC framework when, after iteration 10, the learned input and disturbance are transferred (A) from the AR.Drone 2.0 to the Bebop 2 and (B) from Bebop 2 to AR.Drone 2.0. The PD-ILC and PID-ILC approaches show a significantly larger error after transfer. Note that the scale is different than in Figure 5 [Colour figure can be viewed at wileyonlinelibrary.com]

TABLE 4 Increase in error after transferring the learned trajectory from AR.Drone 2.0 to Bebop 2 and vice versa for the proportional-derivative (PD)–iterative learning control (ILC), proportional-integral-derivative (PID)–ILC, and $\mathcal{L}_1$-ILC approach. It is obvious that the $\mathcal{L}_1$-ILC framewok handles dynamic changes significantly better (by a factor of 4 to 10)

	Factor of Error Increase		
	PD-ILC	PID-ILC	$\mathcal{L}_1$-ILC
AR.Drone 2.0 to Bebop 2	8.492	10.795	0.884
Bebop 2 to AR.Drone 2.0	25.613	8.807	2.327

4.4 | Transfer learning from simulation to real system

In this experiment, we aim to assess the performance of transfer learning from simulation to real systems. The transfer performance depends on how close the simulator is to the real dynamics of the system. For this experiment, simulations have been performed in the Robot Operating System environment using the Gazebo simulator running a simulation of the AR.Drone 2.0. Learning was performed in the simulator over 10 iterations using the PD-ILC, PID-ILC, and $\mathcal{L}_1$-ILC approach. Here, the $\mathcal{L}_1$-ILC framework uses the same underlying model reference system (3) for both the simulation and the real quadrotor. After iteration 10, the learned trajectory is transferred to the real AR.Drone 2.0 quadrotor, and 10 additional learning iterations are performed. Convergence of the error is shown in Figure 8. Again, the $\mathcal{L}_1$-ILC framework shows the best transfer capabilities and needs no relearning. Overall, since the simulator is very consistent (only minor sources of random noise added), it is possible to achieve very low tracking errors. With the real system, this error increases as not all disturbances are repeatable and can be compensated for. However, the AR.Drone 2.0 achieves comparable tracking errors as before (see Figure 7).

4.5 | Reference model–based input to initialize learning

In this experiment, we want to calculate, based on the $\mathcal{L}_1$ reference model, an initial input for the ILC. In Section 4.1 and Figure 4, we showed that the real system behaves as the model reference system (3) when using the $\mathcal{L}_1$ adaptive controller. We use this feature to simplify the closed-loop system (as defined in Figure 2) and substitute the $\mathcal{L}_1$ adaptive controller and the system with the $\mathcal{L}_1$ reference model to obtain the block diagram shown in Figure 9. Using this simplified system, we obtain a state-space representation defined by:

$$A = \begin{bmatrix} 0 & 1 \\ -Km & -m \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ Km \end{bmatrix}.$$

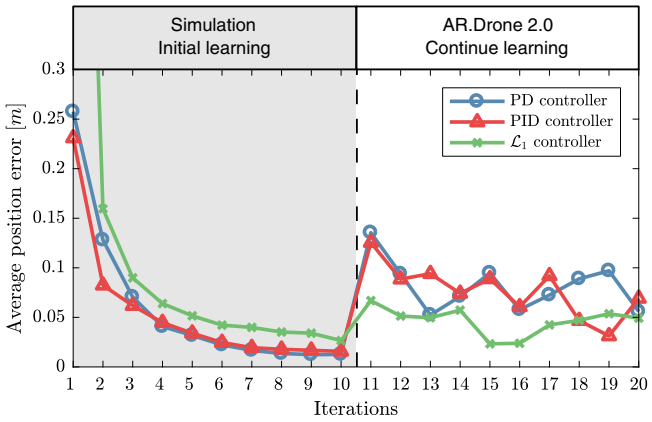

FIGURE 8 Learning behavior of the proportional-derivative (PD)–iterative learning control (ILC), proportional-integral-derivative (PID)–ILC, and $\mathcal{L}_1$-ILC framework in the simulator for iterations 1 to 10. The learned reference input and disturbance are transferred to the AR.Drone 2.0 after iteration 10, and learning continues in iterations 11 to 20. Note that the scale is different than in Figure 7A and 7B [Colour figure can be viewed at wileyonlinelibrary.com]

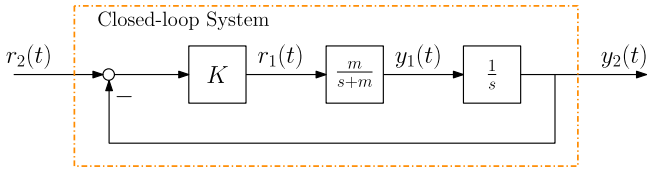

FIGURE 9 Simplified closed-loop system (defined in Figure 2) that uses the fact that the system behaves like the reference model specified by the $\mathcal{L}_1$ adaptive controller [Colour figure can be viewed at wileyonlinelibrary.com]

Furthermore, we define the initial state as x_0. With the aforementioned definitions, it is possible to compute an input based on (3) such that the system tracks the reference exactly. Since we know the desired output $\mathbf{y}_2^*$, we can compute an initial input $\mathbf{r}_{2,1}$ and initial disturbance estimate $\mathbf{d}_1$ using (24), resulting in:

$$\begin{aligned} \mathbf{r}_{2,1} &= \mathbf{F}_{\text{ILC}}^{-1}\left(\mathbf{y}_2^* - \mathbf{d}^0\right), \\ \mathbf{d}_1 &= \mathbf{F}_{\text{ILC}}\left(\mathbf{r}_{2,1} - \mathbf{y}_2^*\right), \\ \mathbf{d}^0 &= \left[(Ax_0)^T, (A^2x_0)^T, \ldots, (A^{N-1}x_0)^T\right]^T, \end{aligned} \tag{37}$$

with $\mathbf{d}^0$ defined as in the work of Schoellig et al.[13] Note that (37) does not use the deviations from the nominal trajectories in contrast to (24). The calculated input $\mathbf{r}_{2,1}$ is applied to both the AR.Drone 2.0 and Bebop 2 using the $\mathcal{L}_1$ adaptive controller without learning. The response over time for a 3-dimensional trajectory is shown in Figure 10. Since there is still a small error in the response of both drones, learning is initialized, and a 10-iteration learning experiment is performed. The experiment is repeated five times, and mean and standard deviation of the error are shown in Figure 11.

4.6 | Discussion on input initializing approaches and transfer learning performance

The experiments in the previous subsections demonstrate the capability of the $\mathcal{L}_1$-ILC framework to achieve high-precision trajectory tracking in the first iteration of a new experiment by (i) transferring learned experience from a dynamically different system, (ii) transferring learned experience from a simulation, and (iii) generating the first input trajectory based on the $\mathcal{L}_1$ reference model. Table 5 shows the initial errors and average converged errors obtained when applying methods (i) to (iii) described to an AR.Drone 2.0. We compare these errors to a standard first trial where the desired output $\mathbf{y}_2^*$ is used as the reference input $\mathbf{r}_2$. We make the following observations.

- We begin by comparing the error achieved in the first iteration with the naive input to the input obtained by transfer learning from a different system (Section 4.3). When the PD and PID controllers are used, the initial error of the transfer learning approach can be up to 4 times larger than the error of the naive input. This means that transfer learning from a dynamically different system has an adverse effect on the error for the PD and PID controllers and should not be done. In contrast, when the $\mathcal{L}_1$ adaptive controller is used, the error in the first iteration using transfer learning from a dynamically different system is comparable to the converged error achieved after learning with the naive input. This

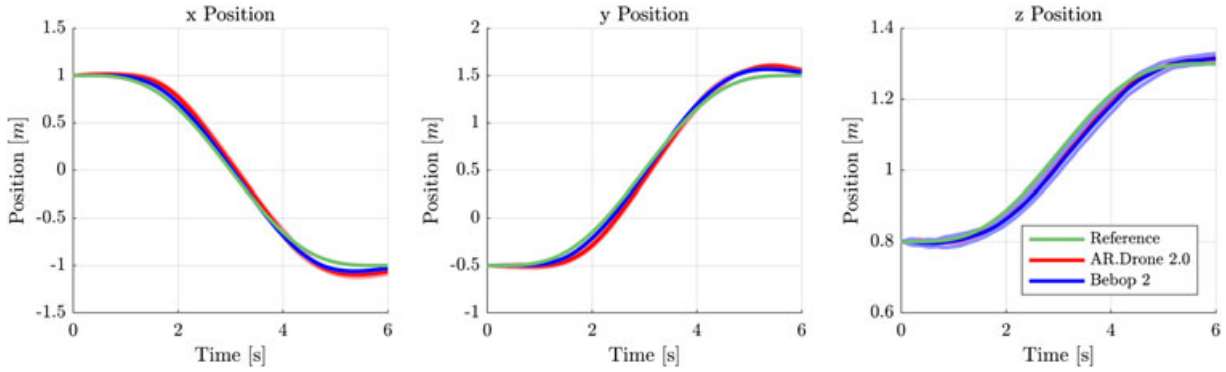

FIGURE 10 Response over time for a 3-dimensional trajectory with the input calculated based on the $\mathcal{L}_1$ reference model applied to the AR.Drone 2.0 and Bebop 2 using the $\mathcal{L}_1$ adaptive controller. Shown are the mean and standard deviation over 5 experiments. Both systems track the reference trajectory closely [Colour figure can be viewed at wileyonlinelibrary.com]

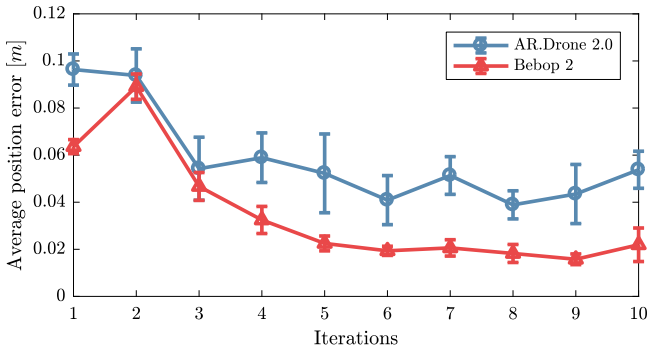

FIGURE 11 Mean of the error for five 10-iteration sets when the $\mathcal{L}_1$ reference model based input is used to initialize the learning for the $\mathcal{L}_1$–iterative learning control framework. The initial error of both systems is very low and is further reduced through learning. An average position error of 2 to 5 centimeters is very low by all standards [Colour figure can be viewed at wileyonlinelibrary.com]

TABLE 5 Average position error of a 3-dimensional trajectory flown with the AR.Drone 2.0 for the 1st and 8th to 10th iteration in an iterative learning control (ILC) experiment for initializing without and with learned experience obtained from (i) transferring learning from a dynamically different system, (ii) transferring learning from a simulation, and (iii) generating the input based on the $\mathcal{L}_1$ reference model

Initial Input	**Naive**		**Transfer Learning from System**		**Transfer Learning from Simulation**		**Calculated Input**	
Iteration	**1st**	**8*th*-10th**	**1st**	**8*th*-10th**	**1st**	**8*th*-10th**	**1st**	**8*th*-10th**
PD-ILC [m]	0.213	0.073	0.898	0.120	0.135	0.081	–	–
PID-ILC [m]	0.204	0.071	0.745	0.052	0.126	0.049	–	–
$\mathcal{L}_1$-ILC [m]	0.991	0.051	0.044	0.048	0.067	0.050	0.096	0.045

is possible as both systems run an adaptive controller with the same predefined model reference, which defines the system behavior. Consequently, transfer learning from a dynamically different system is highly effective for the $\mathcal{L}_1$-ILC approach.

- For the PD-ILC and PID-ILC approaches, using trajectories learned in simulation results in a better performance in the first iteration than transferring learning from a different real system. This is because the simulator closely resembles the behavior of the real system. In our experiments, transferring knowledge from simulation to the real system was beneficial for all frameworks compared with the initial performance with the naive input. However, partial relearning is necessary for the PD and PID case while not necessary for the $\mathcal{L}_1$ case.
- Using the $\mathcal{L}_1$ adaptive controller allows us to calculate an input based on the $\mathcal{L}_1$ reference model and to achieve an error ten times smaller than the error obtained with a naive input.

Overall, the best results are obtained with the $\mathcal{L}_1$-ILC framework. Within this framework, using an input transferred from an initial learning process in a different system achieves the lowest tracking errors in the first iteration. For the PD-ILC and PID-ILC frameworks, it is only possible to transfer experience if the system used for initial learning closely resembles the real system.

5 | CONCLUSION

In this paper, we have shown the capabilities of an $\mathcal{L}_1$-ILC framework to achieve precise trajectory tracking and to enable transfer learning. The $\mathcal{L}_1$ adaptive controller forces the system to remain close to a predefined nominal system behavior, even in the presence of unknown and changing disturbances. This makes it possible for two dynamically different systems to have the same predefined behavior. However, having a repeatable system does not imply achieving zero tracking error. We use ILC to learn from previous iterations and improve the tracking performance over time. We derive performance bounds for the $\mathcal{L}_1$-ILC approach analytically. Experimental results on quadrotors show significant performance improvements of the proposed $\mathcal{L}_1$-ILC approach compared with nonadaptive PD-ILC and PID-ILC approaches in terms of disturbance attenuation, transfer learning capability between dynamically different systems, and transfer learning from simulation to the real system. Since the $\mathcal{L}_1$ adaptive controller makes the system behave in a predefined way, it also allows us to compute a near optimal input from the $\mathcal{L}_1$ reference model, which achieves a small tracking error in the first trial. Overall, the $\mathcal{L}_1$-ILC framework promises to make robot learning simpler and more effective as robots can learn from each other and from simulations.

ACKNOWLEDGEMENTS

This work was supported by Canada Foundation for Innovation John R. Evans Leaders Fund under grant CFI/ORF 33000, the Natural Sciences and Engineering Research Council of Canada under grants CREATE-466088 and RGPIN-2014-04634, the Ontario Research Fund for Small Infrastructure Funds under grant CFI/ORF 33000, Alfred P. Sloan Foundation Sloan Research Fellowship and Ontario Early Researcher Award, and the Mexican National Council of Science and Technology.

ORCID

Karime Pereida http://orcid.org/0000-0002-8576-2164
Dave Kooijman http://orcid.org/0000-0002-1808-2441
Rikky R. P. R. Duivenvoorden http://orcid.org/0000-0002-6807-3738
Angela P. Schoellig http://orcid.org/0000-0003-4012-4668

REFERENCES

1. Skelton R. Model error concepts in control design. *Int J Control*. 1989;49(5):1725-1753.
2. Morari M, Lee JH. Model predictive control: past, present and future. *Comput Chem Eng*. 1999;23(4-5):667-682.
3. Skogestad S, Postlethwaite I. *Multivariable Feedback Control: Analysis and Design*. Vol 2. New York, NY: Wiley; 2007.
4. Parks P. Liapunov redesign of model reference adaptive control systems. *IEEE Trans Autom Control*. 1966;11(3):362-367.
5. Hovakimyan N, Cao C. $\mathcal{L}_1$ *Adaptive Control Theory: Guaranteed Robustness with Fast Adaptation*. Philadelphia, PA: Society for Industrial and Applied Mathematics; 2010.
6. Mallikarjunan S, Nesbit B, Kharisov E, Xargay E, Hovakimyan N, Cao C. $\mathcal{L}_1$ adaptive controller for attitude control of multirotors. Paper presented at: 2012 AIAA Guidance, Navigation and Control Conference; 2012; Minneapolis, MN.
7. Michini B, How JP. $\mathcal{L}_1$ adaptive control for indoor autonomous vehicles: Design process and flight testing. Paper presented at: 2009 AIAA Guidance, Navigation and Control Conference; 2009; Chicago, IL.
8. Gunnarsson S, Norrlöf M. On the design of ILC algorithms using optimization. *Automatica*. 2001;37(12):2011-2016.
9. Ostafew CJ, Schoellig AP, Barfoot TD. Visual teach and repeat, repeat, repeat: Iterative learning control to improve mobile robot path tracking in challenging outdoor environments. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS); 2013; Tokyo, Japan.
10. Yu D, Zhu Y, Yang K, Hu C, Li M. A time-varying Q-filter design for iterative learning control with application to an ultra-precision dual-stage actuated wafer stage. *Proc Inst Mech Eng Part I: J Syst Control Eng*. 2014;228(9):658-667.

11. Schoellig AP, D'Andrea R. Optimization-based iterative learning control for trajectory tracking. Paper presented at: 2009 European Control Conference (ECC); 2009; Budapest, Hungary.
12. Mueller FL, Schoellig AP, D'Andrea R. Iterative learning of feed-forward corrections for high-performance tracking. Paper presented at: 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS); 2012; Vilamoura, Portugal. https://doi.org/10.1109/IROS.2012.6385647
13. Schoellig AP, Mueller FL, D'Andrea R. Optimization-based iterative learning for precise quadrocopter trajectory tracking. *Auton Robots*. 2012;33(1-2):103-127.
14. Bristow D, Tharayil M, Alleyne AG. A survey of iterative learning control. *IEEE Control Syst*. 2006;26(3):96-114.
15. Barton K, Mishra S, Xargay E. Robust iterative learning control: $\mathcal{L}_1$ adaptive feedback control in an ILC framework. In: Proceedings of the 2011 American Control Conference (ACC); 2011; San Francisco, CA.
16. Altin B, Barton K. $\mathcal{L}_1$ adaptive control in an iterative learning control framework: Stability, robustness and design trade-offs. In: Proceedings of the 2013 American Control Conference (ACC); 2013; Washington, DC.
17. Altın B, Barton K. Robust iterative learning for high precision motion control through $\mathcal{L}_1$ adaptive feedback. *Mechatronics*. 2014;24(6):549-561.
18. Pereida K, Duivenvoorden RR, Schoellig AP. High-precision trajectory tracking in changing environments through $\mathcal{L}_1$ adaptive feedback and iterative learning. 2017. *arXiv preprint arXiv*:1705.04763.
19. Tobin J, Fong R, Ray A, Schneider J, Zaremba W, Abbeel P. Domain randomization for transferring deep neural networks from simulation to the real world. 2017. *arXiv preprint arXiv*:1703.06907.
20. Christiano P, Shah Z, Mordatch I, et al. Transfer from simulation to real world through learning deep inverse dynamics model. 2016. *arXiv preprint arXiv*:1610.03518.
21. Devin C, Gupta A, Darrell T, Abbeel P, Levine S. Learning modular neural network policies for multi-task and multi-robot transfer. 2016. *arXiv preprint arXiv*:1609.07088.
22. Hamer M, Waibel M, D'Andrea R. Knowledge transfer for high-performance quadrocopter maneuvers. Paper presented at: 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS); 2013; Tokyo, Japan.
23. Morari M. Robust stability of systems with integral control. *IEEE Trans Autom Control*. 1985;30(6):574-577.
24. Konstantopoulos GC, Zhong QC, Ren B, Krstic M. Bounded integral control of input-to-state practically stable nonlinear systems to guarantee closed-loop stability. *IEEE Trans Autom Control*. 2016;61(12):4196-4202.
25. Lee JH, Lee KS, Kim WC. Model-based iterative learning control with a quadratic criterion for time-varying linear systems. *Automatica*. 2000;36(5):641-657.
26. Powers C, Mellinger D, Kumar V. Quadrotor kinematics and dynamics. In: Valavanis K, Vachtsevanos G, eds. *Handbook of Unmanned Aerial Vehicles*. Dordrecht, The Netherlands: Springer; 2015:307-328.
27. Schoellig AP, Augugliaro F, D'Andrea R. Synchronizing the motion of a quadrocopter to music. Paper presented at: 2010 IEEE International Conference on Robotics and Automation (ICRA); 2010; Anchorage, AK.
28. Wright S, Nocedal J. Numerical optimization. *Springer Sci*. 1999;35(67-68):7.

How to cite this article: Pereida K, Kooijman D, Duivenvoorden RRPR, Schoellig AP. Transfer learning for high-accuracy trajectory tracking through $\mathcal{L}_1$ adaptive feedback and iterative learning. *Int J Adapt Control Signal Process*. 2019;33:388–409. https://doi.org/10.1002/acs.2887

APPENDIX A: PROOF OF LEMMA 1

Proof. Substitution of (17) into (14) results in

$$u_{1,\text{ref}}(s) = \frac{C(s)M(s)r_{1,\text{ref}}(s) - C(s)A(s)d_{1,\text{ref}}(s)}{C(s)A(s) + (1 - C(s))M(s)}. \tag{A1}$$

From (13) and (17), we obtain,

$$y_{1,\text{ref}}(s) = A(s)(u_{1,\text{ref}}(s) + d_{1,\text{ref}}(s)). \tag{A2}$$

Substitution of (A1) into (A2) results in

$$y_{1,\text{ref}}(s) = A(s)M(s)\frac{C(s)r_{1,\text{ref}}(s) + d_{1,\text{ref}}(s)(1 - C(s))}{C(s)A(s) + (1 - C(s))M(s)}.$$

Using (6), we can rewrite

$$y_{1,\text{ref}}(s) = H(s)\left(C(s)r_{1,\text{ref}}(s) + (1 - C(s))d_{1,\text{ref}}(s)\right). \tag{A3}$$

Substitution of (16) into (A3) and using the definition in (15) results in the following expression:

$$y_{2,\text{ref}}(s) = \frac{1}{s}H(s)\left(C(s)K\left(r_2(s) - y_{2,\text{ref}}(s)\right) + (1 - C(s))d_{1,\text{ref}}(s)\right),$$

and hence

$$y_{2,\text{ref}}(s) = F(s)H(s)(C(s)Kr_2(s) + (1 - C(s))d_{1,\text{ref}}(s)). \tag{A4}$$

In Lemma 4.1.1 in the work of Hovakimyan and Cao,[5] using the $\mathcal{L}_1$-norm condition in (8), it is shown that the following upper bound holds uniformly:

$$\|y_{1,\text{ref}_\tau}\|_{\mathcal{L}_\infty} \leq \frac{\|H(s)C(s)\|_{\mathcal{L}_1}\|r_{1,\text{ref}}\|_{\mathcal{L}_\infty} + \|G(s)\|_{\mathcal{L}_1}L_0}{1 - \|G(s)\|_{\mathcal{L}_1}L},$$

where $\|y_{1,\text{ref}_\tau}\|_{\mathcal{L}_\infty}$ is the truncated $\mathcal{L}_\infty$-norm of $y_{1,\text{ref}}(t)$ up to $t = \tau$. Hence, $\|y_{1,\text{ref}}\|_{\mathcal{L}_\infty}$ is bounded. Since $H(s)$, $F(s)$, and $C(s)$ are strictly proper stable transfer functions, taking the norm of the reference system (A4) and making use of Assumption 1 and Lemma 4.1.1 in the work of Hovakimyan and Cao[5] yields the following bound:

$$\|y_{2,\text{ref}_\tau}\|_{\mathcal{L}_\infty} \leq \|F(s)H(s)C(s)\|_{\mathcal{L}_1}K\|r_2\|_{\mathcal{L}_\infty} + \|F(s)G(s)\|_{\mathcal{L}_1}\left(L\|y_{1,\text{ref}}\|_{\mathcal{L}_\infty} + L_0\right). \tag{A5}$$

This results holds uniformly, so $\|y_{2,\text{ref}}\|_{\mathcal{L}_\infty}$ is bounded. Hence, the closed-loop reference system in (13) to (16) is BIBO stable. □

APPENDIX B: PROOF OF THEOREM 1

Proof. Theorem 4.4.1 in the work of Hovakimyan and Cao[5] proves the bound in (18) under the same assumptions made in this paper. It remains to show the bound in (19). We use the following definitions:

$$H_0(s) \triangleq \frac{A(s)}{C(s)A(s) + (1 - C(s))M(s)}, \tag{B1}$$

$$H_1(s) \triangleq \frac{(A(s) - M(s))C(s)}{C(s)A(s) + (1 - C(s))M(s)}, \text{ and} \tag{B2}$$

$$H_2(s) \triangleq \frac{C(s)H(s)}{M(s)}. \tag{B3}$$

All $H_0(s)$, $H_1(s)$, and $H_2(s)$ are strictly proper stable transfer functions, as shown in the work of Hovakimyan and Cao.[5] The following expressions using (B1) and (B2) can be verified:

$$M(s)H_0(s) = H(s), \text{ and} \tag{B4}$$

$$M(s)\left(C(s) + H_1(s)(1 - C(s))\right) = H(s)C(s). \tag{B5}$$

Let $\tilde{\sigma}(t) \triangleq \hat{\sigma}(t) - \sigma(t)$, where $\hat{\sigma}$ is the adaptive estimate and σ is defined in (5). The control law in (11) can be expressed as:

$$u(s) = C(s)r_1(s) - C(s)(\tilde{\sigma}(s) + \sigma(s)). \tag{B6}$$

Substitution of (B6) into (5) and making use of the definitions in (B1) and (B2) results in the following expression for $\sigma(s)$:

$$\sigma(s) = H_1(s)(r_1(s) - \tilde{\sigma}(s)) + H_0(s)d_{\mathcal{L}_1}(s). \tag{B7}$$

Substitution of (B6) and (B7) into the system (4) results in:

$$y_1(s) = M(s)\left(C(s) + H_1(s)(1 - C(s))\right)(r_1(s) - \tilde{\sigma}(s)) + M(s)H_0(s)(1 - C(s))d_{\mathcal{L}_1}(s).$$

Using (B4) and (B5), this expression simplifies to:

$$y_1(s) = H(s)C(s)\left(r_1(s) - \tilde{\sigma}(s)\right) + H(s)(1 - C(s))d_{\mathcal{L}_1}(s). \tag{B8}$$

We obtain $y_2(s)$ by substituting (12) and (B8) into (15) and making use of the definition in (7):

$$y_2(s) = F(s)H(s)\left(C(s)Kr_2(s) + (1 - C(s))d_{\mathcal{L}_1}(s)\right) - F(s)H(s)C(s)\tilde{\sigma}(s). \tag{B9}$$

Substituting (4) and (9) into the definition of $\tilde{y}(s)$ in the adaptation law results in:

$$\tilde{y}(s) = M(s)\tilde{\sigma}(s). \tag{B10}$$

Recalling the reference system in (A4) and using the expression for $y_2(s)$ in (B9), the error between reference and actual systems, $y_{2,\text{ref}}(s) - y_2(s)$ is:

$$y_{2,\text{ref}}(s) - y_2(s) = F(s)H(s)(1 - C(s))(d_{\text{ref}}(s) - d_{\mathcal{L}_1}(s)) + \frac{F(s)H(s)C(s)}{M(s)}M(s)\tilde{\sigma}(s).$$

Substituting the expression for $\tilde{y}(s)$ in (B10) and the definition of G(s) in (8), we obtain:

$$y_{2,\text{ref}}(s) - y_2(s) = F(s)G(s)(d_{\text{ref}}(s) - d_{\mathcal{L}_1}(s)) + \frac{F(s)H(s)C(s)}{M(s)}\tilde{y}(s).$$

In theorem 4.1.1 in the work of Hovakimyan and Cao,[5] using the same assumptions in this work, the following bound is derived:

$$\|y_{1,\text{ref}} - y_1\|_{\mathcal{L}_\infty} \leq \frac{\|H_2(s)\|_{\mathcal{L}_1}}{1 - \|G(s)\|_{\mathcal{L}_1}L}\gamma_0.$$

Finally, since the $\mathcal{L}_1$-norm of $F(s)G(s)$ exists and $\frac{F(s)H(s)C(s)}{M(s)}$ is strictly proper and stable, the following bound can be derived by taking the truncated $\mathcal{L}_\infty$-norm and by making use of Assumption 1:

$$\|y_{2,\text{ref}_\tau} - y_{2_\tau}\|_{\mathcal{L}_\infty} \leq \|F(s)G(s)\|_{\mathcal{L}_1}L\|y_{1,\text{ref}_\tau} - y_{1_\tau}\|_{\mathcal{L}_\infty} + \left\|\frac{F(s)H(s)C(s)}{M(s)}\right\|_{\mathcal{L}_1}\|\tilde{y}_\tau\|_{\mathcal{L}_\infty},$$

which holds uniformly. Making use of the bounds in theorem 4.1.1 in the work of Hovakimyan and Cao[5] results in:

$$\|y_{2,\text{ref}} - y_2\|_{\mathcal{L}_\infty} \leq \left(\|F(s)G(s)\|_{\mathcal{L}_1}L\frac{\|H_2(s)\|_{\mathcal{L}_1}}{1 - \|G(s)\|_{\mathcal{L}_1}L} + \left\|\frac{F(s)H(s)C(s)}{M(s)}\right\|_{\mathcal{L}_1}\right)\gamma_0, \tag{B11}$$

proving the bound in (19). □

APPENDIX C: DISCUSSION OF REMARK 1

We begin our discussion with the case where the constraints in (29) are inactive. For this case, using (26) and deriving (28) with respect to $\bar{\mathbf{r}}_{2,j+1}$, we obtain:

$$\nabla_{\bar{\mathbf{r}}_{2,j+1}} = \left(\mathbf{F}_{\text{ILC}}^T\mathbf{Q}\mathbf{F}_{\text{ILC}} + \mathbf{W}\right)\bar{\mathbf{r}}_{2,j+1} + \mathbf{F}_{\text{ILC}}^T\mathbf{Q}\hat{\mathbf{d}}_{j|j}.$$

Equating to 0 and solving for $\bar{\mathbf{r}}_{2,j+1}$, we get:

$$\bar{\mathbf{r}}_{2,j+1} = -\left(\mathbf{F}_{\text{ILC}}^T\mathbf{Q}\mathbf{F}_{\text{ILC}} + \mathbf{W}\right)^{-1}\left(\mathbf{F}_{\text{ILC}}^T\mathbf{Q}\hat{\mathbf{d}}_{j|j}\right). \tag{C1}$$

By definition, $\mathbf{Q}$ is positive definite and $\mathbf{F_{ILC}}$ is full rank according to Assumption 5; hence, $\mathbf{F}_{\text{ILC}}^T\mathbf{Q}\mathbf{F}_{\text{ILC}}$ is positive definite. Furthermore, the sum of a positive definite and a positive semidefinite matrix is itself positive definite. Since $\mathbf{W}$ is positive semidefinite by definition, $\mathbf{F}_{\text{ILC}}^T\mathbf{Q}\mathbf{F}_{\text{ILC}} + \mathbf{W}$ is positive definite and invertible. The Kalman filter is asymptotically stable; hence, $\hat{\mathbf{d}}_{j|j} \to \mathbf{d}_\infty$ as $j \to \infty$. Therefore, $\bar{\mathbf{r}}_{2,j+1}$ also converges. Substituting (C1) into (24), we obtain:

$$\bar{\mathbf{y}}_{2,j+1} = \mathbf{F}_{\text{ILC}}\left(-\left(\mathbf{F}_{\text{ILC}}^T\mathbf{Q}\mathbf{F}_{\text{ILC}} + \mathbf{W}\right)^{-1}\left(\mathbf{F}_{\text{ILC}}^T\mathbf{Q}\hat{\mathbf{d}}_{j|j}\right)\right) + \mathbf{d}_\infty. \tag{C2}$$

Zeroing of the error is possible for the following choice of weighting matrices: $\mathbf{Q} = q\mathbf{I}$ and $\mathbf{W} = 0$. Substituting in (C2), we obtain:

$$\begin{aligned}\bar{\mathbf{y}}_{2,j+1} &= \mathbf{F}_{\mathrm{ILC}}\left(-\left(\mathbf{F}_{\mathrm{ILC}}^T q\mathbf{I}\mathbf{F}_{\mathrm{ILC}}\right)^{-1}\left(\mathbf{F}_{\mathrm{ILC}}^T q\mathbf{I}\hat{\mathbf{d}}_{j|j}\right)\right) + \mathbf{d}_\infty \\ &= -\mathbf{F}_{\mathrm{ILC}}\mathbf{F}_{\mathrm{ILC}}^{-1}\mathbf{F}_{\mathrm{ILC}}^{-T}\mathbf{F}_{\mathrm{ILC}}^T\hat{\mathbf{d}}_{j|j} + \mathbf{d}_\infty \\ &= \mathbf{d}_\infty - \hat{\mathbf{d}}_{j|j},\end{aligned}$$

where $\hat{\mathbf{d}}_{j|j} \to \mathbf{d}_\infty$ and $\bar{\mathbf{y}}_{2,j+1} \to 0$ as $j \to \infty$.

If the inequality constraints are active, we add Lagrangian multipliers to (28) such that:

$$\begin{aligned}\mathcal{L}(\bar{\mathbf{r}}_{2,j}, \lambda_1, \lambda_2) = \frac{1}{2}&\left\{\left(\mathbf{F}_{\mathrm{ILC}}\bar{\mathbf{r}}_{2,j+1} + \hat{\mathbf{d}}_{j|j}\right)^T\mathbf{Q}\left(\mathbf{F}_{\mathrm{ILC}}\bar{\mathbf{r}}_{2,j+1} + \hat{\mathbf{d}}_{j|j}\right) + \bar{\mathbf{r}}_{2,j+1}^T\mathbf{W}\bar{\mathbf{r}}_{2,j+1}\right\} \\ &- \sum\nolimits_{l\in V_{\mathrm{act}}}\lambda_{1,l}\left(\mathbf{V}_{c,l}\mathbf{F}_{\mathrm{ILC}}\bar{\mathbf{r}}_{2,j+1} + \mathbf{V}_{c,l}\hat{\mathbf{d}}_{j|j} - \hat{\mathbf{y}}_{\max}\right) \\ &- \sum\nolimits_{l\in Z_{\mathrm{act}}}\lambda_{2,l}\left(\mathbf{Z}_{c,l}\bar{\mathbf{r}}_{2,j+1} - \bar{\mathbf{r}}_{2,\max}\right),\end{aligned}$$

where $\mathbf{V}_{c,l}$ is the lth row of $\mathbf{V}_c$, $\mathbf{Z}_{c,l}$ is the lth row of $\mathbf{Z}_c$, and $\lambda_{1,l}$ and $\lambda_{2,l}$ are Lagrange multipliers for the set V_{act} of estimated output $\hat{\mathbf{y}}_{j+1|j}$ active constraints and the set Z_{act} of input $\bar{\mathbf{r}}_{2,j+1}$ active constraints. The first-order necessary conditions[28] for $\bar{\mathbf{r}}_{2,j+1}$ to be a solution of (28) subject to (29) state that there are vectors λ_1^* and λ_2^* such that the following system of equations is satisfied:

$$\begin{bmatrix}\mathbf{F}_{\mathrm{ILC}}^T\mathbf{Q}\mathbf{F}_{\mathrm{ILC}} + \mathbf{W} & -(\mathbf{V}_{c,\mathrm{act}}\mathbf{F}_{\mathrm{ILC}})^T & -\mathbf{Z}_{c,\mathrm{act}}{}^T \\ \mathbf{V}_{c,\mathrm{act}}\mathbf{F}_{\mathrm{ILC}} & 0 & 0 \\ \mathbf{Z}_{c,\mathrm{act}} & 0 & 0\end{bmatrix}\begin{bmatrix}\bar{\mathbf{r}}_{2,j+1} \\ \lambda_1^* \\ \lambda_2^*\end{bmatrix} = \begin{bmatrix}-\mathbf{F}_{\mathrm{ILC}}^T\mathbf{Q}\hat{\mathbf{d}}_{j|j} \\ \hat{\mathbf{y}}_{\max} - \mathbf{V}_{c,\mathrm{act}}\hat{\mathbf{d}}_{j|j} \\ \mathbf{r}_{2,\max}\end{bmatrix}, \tag{C3}$$

where $\mathbf{V}_{c,\mathrm{act}}$ is the matrix whose rows are $\mathbf{V}_{c,l}, \forall l \in V_{\mathrm{act}}$ and $\mathbf{Z}_{c,\mathrm{act}}$ is the matrix whose rows are $\mathbf{Z}_{c,l}, \forall l \in Z_{\mathrm{act}}$. These conditions are a consequence of the first-order optimality conditions described in theorem 12.2 in the work of Wright and Nocedal.[28] We denote $L_{V,Z} \leq N$ as the number of elements in $\mathbf{V}_{c,\mathrm{act}} \cup \mathbf{Z}_{c,\mathrm{act}}$. We use Z to denote the $N \times (N - L_{V,Z})$ matrix whose columns are a basis for the null space of $[\mathbf{V}_{c,\mathrm{act}}\mathbf{F}_{\mathrm{ILC}}\,\mathbf{Z}_{c,\mathrm{act}}]^T$, ie, Z has full rank and satisfies $[\mathbf{V}_{c,\mathrm{act}}\mathbf{F}_{\mathrm{ILC}}\,\mathbf{Z}_{c,\mathrm{act}}]^T Z = 0$.

According to theorem 16.2 in the work of Wright and Nocedal,[28] if $[\mathbf{V}_{c,\mathrm{act}}\mathbf{F}_{\mathrm{ILC}}\,\mathbf{Z}_{c,\mathrm{act}}]^T$ has full rank and the reduced Hessian matrix $Z^T(\mathbf{F}_{\mathrm{ILC}}^T\mathbf{Q}\mathbf{F}_{\mathrm{ILC}} + \mathbf{W})Z$ is positive definite, then $\bar{\mathbf{r}}_{2,j+1}$ satisfying (C3) is the unique global solution of (28) under (29). We first note that, according to Assumption 6, $[\mathbf{V}_{c,\mathrm{act}}\mathbf{F}_{\mathrm{ILC}}\mathbf{Z}_{c,\mathrm{act}}]^T$ has full rank. Since Z has full rank and $\mathbf{F}_{\mathrm{ILC}}^T\mathbf{Q}\mathbf{F}_{\mathrm{ILC}} + \mathbf{W}$ is positive definite as described, then $Z^T(\mathbf{F}_{\mathrm{ILC}}^T\mathbf{Q}\mathbf{F}_{\mathrm{ILC}} + \mathbf{W})Z$ is positive definite and $\bar{\mathbf{r}}_{2,j+1}$ is the unique global solution to the minimization problem.

The unique global solution to the minimization problem with active constraints is:

$$\bar{\mathbf{r}}_{j+1} = \left(\mathbf{F}_{\mathrm{ILC}}^T\mathbf{Q}\mathbf{F}_{\mathrm{ILC}} + \mathbf{W}\right)^{-1}\left(-\mathbf{F}_{\mathrm{ILC}}^T\mathbf{Q}\hat{\mathbf{d}}_{j|j} + (\mathbf{V}_{c,\mathrm{act}}\mathbf{F}_{\mathrm{ILC}})^T\lambda_1^* + \mathbf{Z}_{c,\mathrm{act}}^T\lambda_2^*\right). \tag{C4}$$

As $j \to \infty$, $\hat{\mathbf{d}}_{j|j} \to \mathbf{d}_\infty$, and $\bar{\mathbf{r}}_{j+1}$ converges. Substituting (C4) into (25), we obtain:

$$\bar{\mathbf{y}}_{2,j+1} = \mathbf{F}_{\mathrm{ILC}}\left(\mathbf{F}_{\mathrm{ILC}}^T\mathbf{Q}\mathbf{F}_{\mathrm{ILC}} + \mathbf{W}\right)^{-1}\left(-\mathbf{F}_{\mathrm{ILC}}^T\mathbf{Q}\hat{\mathbf{d}}_{j|j} + (\mathbf{V}_{c,\mathrm{act}}\mathbf{F}_{\mathrm{ILC}})^T\lambda_1^* + \mathbf{Z}_{c,\mathrm{act}}^T\lambda_2^*\right) + \mathbf{d}_\infty. \tag{C5}$$

Zeroing of the error is not possible even with the previous choice of weighting matrices $\mathbf{Q} = q\mathbf{I}$ and $\mathbf{W} = 0$:

$$\begin{aligned}\bar{\mathbf{y}}_{2,j+1} &= \mathbf{F}_{\mathrm{ILC}}\left(\mathbf{F}_{\mathrm{ILC}}^T q\mathbf{I}\mathbf{F}_{\mathrm{ILC}}\right)^{-1}\left(-\mathbf{F}_{\mathrm{ILC}}^T q\mathbf{I}\hat{\mathbf{d}}_{j|j} + (\mathbf{V}_{c,\mathrm{act}}\mathbf{F}_{\mathrm{ILC}})^T\lambda_1^* + \mathbf{Z}_{c,\mathrm{act}}^T\lambda_2^*\right) + \mathbf{d}_\infty \\ &= q^{-1}\mathbf{F}_{\mathrm{ILC}}^{-T}\left((\mathbf{V}_{c,\mathrm{act}}\mathbf{F}_{\mathrm{ILC}})^T\lambda_1^* + \mathbf{Z}_c^T\lambda_2^*\right) + \mathbf{d}_\infty - \hat{\mathbf{d}}_{j|j}.\end{aligned}$$