S. Meraglia, M. Lovera
Smoother-Based Iterative Learning Control for UAV Trajectory Tracking
IEEE Control Systems Letters, Vol. 6, 2022, p. 1501-1506
doi:10.1109/LCSYS.2021.3116263

Smoother-based Iterative Learning Control for UAV Trajectory Tracking

Salvatore Meraglia[1], Marco Lovera[1]

***Abstract*— This letter presents a data-based control approach to achieve high-performance trajectory tracking with Unmanned Aerial Vehicles (UAVs). We revisit an existing Iterative Learning Control (ILC) algorithm based on the notion that the performance of a system that executes the same task multiple times can be improved by learning from previous executions. While we will specifically refer to multirotor platforms for the experimental validation, the formulation can be applied to any dynamic system (including systems with underlying feedback loops). The novelty of this work is the introduction of a smoother to estimate the repetitive disturbance to improve the learning performance. This estimator must rely on an accurate system model that has been obtained through a black-box identification procedure using the Predictor-Based Subspace Identification (PBSID) algorithm. A Monte Carlo analysis has been carried out with the aim of showing the performance improvements and limitations of the proposed algorithm with respect to existing approaches. Finally, the proposed approach has been validated through experimental activities involving a small quadrotor performing an aggressive manoeuver.**

***Index Terms*— Aerospace, iterative learning control.**

I. Introduction

In recent years, the study of Unmanned Aerial Vehicles (UAVs) has received increasing attention thanks to their wide range of application. Certain problems of practical interest require the generation and repeated execution of a path, where the choice of path and the related tracking accuracy can have a dramatic impact on performance. As an example, UAVs have been widely used for performing a persistent surveillance mission over a very small domain, where they are required to precisely track a desired trajectory in order to perform the task safely and effectively.

Trajectory tracking with UAVs is typically achieved using feedback control approaches [1]. Specifically, linear control techniques are widely used in commercial autopilots, but are not usually able to achieve high performance. On the other hand, non linear methods (*e.g.*, exact input–output feedback linearization, backstepping, *etc.*.) can yield controllers with a significantly better performance, but require a careful *ad-hoc* tuning of the parameters. Moreover, the performance of feedback control approaches is limited by the accuracy of the dynamics model and the causality of the control action that is compensating only for disturbances as they occur. To overcome these limitations, when a system executes the same task multiple times, Iterative Learning Control (ILC) can be used. ILC is based on the notion that the performance of a system that executes the same task multiple times can be improved by learning from previous executions. The goal of ILC is to generate a feedforward control that tracks a specific reference or rejects a repeating disturbance exploiting the past tracking errors. ILC has been successfully applied to many practical industrial systems in manufacturing, robotics, and chemical processing, where mass production on an assembly line entails repetition. For details, the readers are referred to the survey papers [2], [3] and references therein. In recent years ILC algorithms have been also developed and applied to UAVs [4], [5], [6]. In this letter, we revisit the data-based control approach presented in [6]. It modifies the reference before sending it to the UAV, therefore it is an add-on algorithm that fits with any commercial controllers. Moreover it requires only the knowledge of the UAV complementary sensitivity function (the transfer function from the reference to the actual trajectory) without going into details of the open-loop dynamics and the baseline controller. The novelty of this work is the introduction of a smoother to estimate the repetitive disturbance to improve the learning performance and speed up the convergence. Namely, a fixed-interval smoothing algorithm is implemented that uses the entire batch of measurements over a fixed interval to estimate all the states in the interval. This smoother can be derived from a combination of two Kalman filters, one of which works forward over the data and the other of which works backward over the fixed interval [7]. In contrast to previous estimation-based ILC algorithms (see, *e.g.*, [8], [9]), the proposed estimator works in the time-domain and can be extremely helpful when accuracy is an issue. As highlighted in [10], the time-domain estimator in the ILC framework must rely on an accurate system model to not degrade the performance. For the experimental activities involving a small quadrotor, the system model has been obtained through a black-box identification procedure using the Predictor-Based Subspace Identification (PBSID) algorithm [11]. A Monte Carlo analysis has been carried out to show the performance improvements and limitations of the proposed algorithm with respect to existing approaches. In the final part of the letter, experimental results obtained on a quadrotor performing an aggressive manoeuver are reported to show that the proposed approach is capable of remarkably reducing the tracking errors in few iterations.

[1]Dipartimento di Scienze e Tecnologie Aerospaziali, Politecnico di Milano, Via La Masa 34, 20156 Milano, Italy {salvatore.meraglia, marco.lovera}@polimi.it

II. Problem Statement: ILC-enhanced tracking

In this section we introduce and formalize the problem of trajectory tracking in the ILC framework. While we will specifically refer to multirotor platforms for the experimental validation, the formulation is general and can be applied to any dynamic system (including systems with underlying feedback loops). Firstly, we briefly describe the system to be controlled following the approach in [3] focusing on Linear Time-Invariant Single Input Single Output (LTI SISO) systems for the sake of simplicity. Then we present the goal and the structure of the proposed learning algorithm.

A. System description

Consider the discrete-time LTI SISO 2-dimensional system

$$y_j(k) = F(\mathrm{q})u_j(k) + d(k) \quad \text{with} \quad y_j(0) = y_0 \tag{1}$$

where q is the forward time-shift operator $\mathrm{q}x(k) \equiv x(k+1)$, y_j is the output, u_j is the control input, and d is an exogenous signal that repeats at each iteration; the plant $F(\mathrm{q})$ is a proper rational function of q with a delay, or equivalently relative degree, of m; k represents discrete-time points along the time axis and the subscript j represents the iteration trial number along the iteration axis. Notice that $y_j(0) = y_0$ for all j. This is a key assumption in the ILC process and is called the initial reset condition [2]. We assume that $F(\mathrm{q})$ is asymptotically stable[1] and the plant delay[2] $m = 1$. In the following we consider the N-sample sequence of inputs and outputs ($u_j(k),\ k \in \{0,1,\dots,N-1\}$ and $y_j(k),\ k \in \{1,2,\dots,N\}$) and the desired system output ($y_d(k),\ k \in \{1,2,\dots,N\}$). The performance or error signal is defined by $e_j(k) = y_d(k) - y_j(k)$. When the system is described in the *time*-domain, the so called *lifted*-representation is preferred in describing the input/output relation and the ILC update algorithm [3]. The lifted form, in fact, allows to write the SISO time and iteration-domain dynamic system (1) as a Multiple Input Multiple Output (MIMO) iteration-domain dynamic system. To obtain the lifted form, the rational LTI plant (1) is first expanded as an infinite power series by dividing its denominator into its numerator:

$$F(\mathrm{q}) = f_1\mathrm{q}^{-1} + f_2\mathrm{q}^{-2} + f_3\mathrm{q}^{-3} + \cdots \tag{2}$$

where the coefficients f_k are the Markov parameters. Note that $f_1 \neq 0$ since $m = 1$ is assumed. Considering the state space description

$$\begin{aligned} x_j(k+1) &= Ax_j(k) + Bu_j(k) \\ y_j(k) &= Cx_j(k) \end{aligned} \tag{3}$$

we have that $f_k = CA^{k-1}B$. Stacking the signals in vectors, the system dynamics in (1) can be written equivalently as the lifted system:

$$\underbrace{\begin{bmatrix} y_j(1) \\ y_j(2) \\ \vdots \\ y_j(N) \end{bmatrix}}_{Y_j} = \mathbf{F} \underbrace{\begin{bmatrix} u_j(0) \\ u_j(1) \\ \vdots \\ u_j(N-1) \end{bmatrix}}_{U_j} + \underbrace{\begin{bmatrix} d(1) \\ d(2) \\ \vdots \\ d(N) \end{bmatrix}}_{D} \tag{4}$$

$$\mathbf{F} := \begin{bmatrix} f_1 & 0 & \cdots & 0 \\ f_2 & f_1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ f_N & f_{N-1} & \cdots & f_1 \end{bmatrix} \in \mathbb{R}^{N\times N} \quad . \tag{5}$$

The proposed learning algorithm fundamentally relies on the static linear mapping (4) and is explained in the following.

B. Learning algorithm

In most physical implementations, a well-designed feedback controller must be used in combination with ILC [2]. In many cases, a feedback controller already exists on the system, and ILC can be implemented without modifying the feedback controller. Specifically, we combine ILC with the feedback loop in a *serial* arrangement, where the ILC control input is applied to the reference before the feedback loop [12]. This concept is useful when applying ILC to a pre-existing system that uses a commercial controller that does not allow to modify the control signal to the plant.

We revisit the data-based control approach proposed in [6], whose goal is to achieve high-performance tracking of a dynamic system executing multiple times the same task. The related algorithm is an optimization-based (also called *norm-optimal*) ILC [13]. As such, it uses information of the input and the error at the current trial to design the next trial input that minimizes the tracking error, *i.e.*, the discrepancy between the actual and the desired output at the upcoming iteration. Namely, in [6] Schoellig *et al.* introduce a Kalman filter that estimates the repetitive disturbance based on the current input and error measurement. This estimate is then used to update the next trial input that is generated by solving an optimization problem (with possible constraints in the input).

The novelty of this work is the introduction of a smoother to estimate the repetitive disturbance to improve the learning performance and speed up the convergence. In contrast to previous estimation-based ILC algorithms, the proposed estimator works in the time-domain and can be extremely helpful when accuracy is an issue, exploiting the potentiality of the batch state estimation [14].

III. Smoother-based ILC

In this section the two steps of the proposed learning algorithm are described highlighting the novelties with the existing approaches. The overall approach is schematized in Figure 1.

[1] When $F(\mathrm{q})$ is not asymptotically stable, it can be stabilized with a feedback controller, and the ILC can be applied to the closed-loop system. This agrees with the focus of the ILC algorithm that is to improve the performance of the system [3].

[2] This assumption is made only in this section to simplify the system description without loss of generality. For the analysis with a generic plant delay the interested reader is referred to [3].

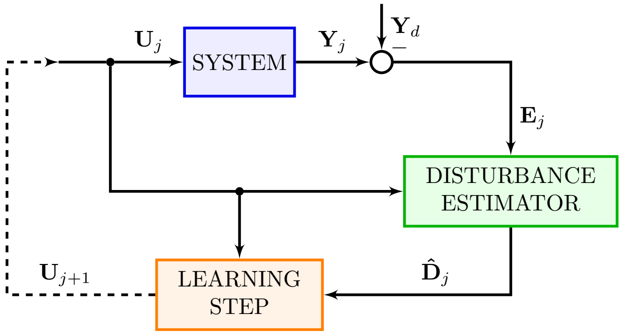

Fig. 1. Estimator-based ILC block diagram in the iteration domain.

A. Disturbance estimation: Smoother

The iteration-to-iteration learning of ILC provides opportunities for advanced filtering and signal processing [3]. For instance, zero-phase filtering [15], which is non-causal, allows for high-frequency attenuation without introducing lag. Previous works estimate the repetitive disturbance using sequential measurements in the iteration domain [8], [9].

In this work the disturbance has been estimated in the time domain: since the estimation is carried out offline, a batch estimation method is used to achieve high accuracy. In fact, batch state estimation methods (also known as smoothers, since they are typically used to *smooth* out the effects of measurement noise) have the advantage of providing state estimates with a smaller error covariance than sequential ones. Basically, smoothers are used to estimate the state at time t, using measurements obtained both before and after t. To accomplish this task, two filters are usually used: a forward-time filter and a backward-time filter. The first practical smoothing algorithms are attributed to Bryson and Frazier [16], as well as Rauch, Tung, and Striebel (RTS) [17]. In particular, the RTS smoothing algorithm has maintained its popularity since the initial paper, and is likely the most widely used algorithm for smoothing to date. In fact, it is one of the most convenient and efficient forms of the fixed-interval smoother, because it combines the backward filter and smoother into one single backward recursion [14]. Using this algorithm, the disturbance estimate $\hat{d}$ is obtained at each iteration j by applying the RTS smoother to the discrete-time stochastic model[3]

$$d(k+1) = d(k) + \omega(k) \tag{6}$$

with the measurement model

$$y(k) = F(\mathrm{q})u_j(k) + d(k) + \mu(k) \quad , \tag{7}$$

where ω and μ are two random vectors described as Gaussian noises with zero mean values and given variance matrices: $\omega(k) \sim \mathcal{N}(0,Q)$, $\mu(k) \sim \mathcal{N}(0,R)$. In the proposed model, the disturbance d represents the state of the dynamical system, while the error e defines the output measurement. The implementation of the RTS smoother specialized for this problem is shown in Algorithm 1.

[3]If a more accurate model of the disturbance is available, it can be incorporated into this algorithm.

Assuming the disturbance remains constant among the different trials, the next iteration predicted error can be written as a function of the disturbance estimate and the next input. Namely, the lifted form is $\hat{E}_{j+1}(U_{j+1}) = \mathbf{F}\, U_{j+1} - Y_d + \hat{D}_j$.

Remark 1: The use of a time-domain filter in ILC can be effective at reducing the variance of the output error resulting from random process and measurement noise. However, an important issue is the influence of having an imperfect model used to design the filter [10]. This error could produce deterministic non-zero steady state errors; however if the model is accurate, then the filter is optimal and will outperform a simpler model-free method. As a consequence, an accurate model is required to exploit this design. ⌟

Algorithm 1 Disturbance estimator: RTS Smoother

Forward Filter Initialization:
  Disturbance: $\hat{d}_f(0) = \hat{d}_{f0}$
  Covariance: $P_f(0) = E\left\{\hat{d}_f(0)\hat{d}_f^T(0)\right\}$
for each $k \in 1,2,\ldots,N$ **do**
  Kalman Gain Computation:
  $K(k) = P_f^-(k)\left[P_f^-(k) + R\right]^{-1}$
  Forward State Propagation:
  $\hat{d}_f(k+1) = \hat{d}_f(k) + K(k)\left[y(k) - \hat{d}_f(k) - F(\mathrm{q})u_j(k)\right]$
  Forward Covariance Update:
  $P_f^+(k) = [I - K(k)]\,P_f^-(k)$
  Forward Covariance Propagation:
  $P_f^-(k+1) = P_f^+(k) + Q$
end for

Smoother Initialization:
  Disturbance: $\hat{d}(N) = \hat{d}_f(N)$
for each $k \in N-1, N-2, \ldots, 1$ **do**
  Smoother Gain:
  $\mathscr{K}(k) = P_f^+(k)\left(P_f^-(k+1)\right)^{-1}$
  Smoother Estimate:
  $\hat{d}(k) = \hat{d}_f(k) + \mathscr{K}(k)\left[\hat{d}(k+1) - \hat{d}_f(k+1)\right]$
end for

B. Input Update: Quadratically-Optimal Design

The learning algorithm is completed by the input update step using a quadratically optimal design. A huge body of literature is available on this design (see, *e.g.*, [18] in which the robustness of this algorithm is analysed). Specifically, the next input U_{j+1} is obtained in the lifted system minimizing the quadratic cost criterion:

$$J = \hat{E}_{j+1}^T W_e \hat{E}_{j+1} + \Delta U_{j+1}^T W_{\Delta u} \Delta U_{j+1} + U_{j+1}^T W_u U_{j+1} \tag{8}$$

where $\Delta U_{j+1} = U_{j+1} - U_j$, W_e is a positive-definite matrix, and $W_{\Delta u}$, W_u are positive-semidefinite matrices. The optimization formulation enables the integration of input and output constraints as shown in [19] and the resulting minimization is a convex optimization problem which can be solved very efficiently by existing software tools.

Remark 2: As noted in [3], the weight on the change in control $W_{\Delta u}$ has no effect on the asymptotic error, but affects how quickly the ILC converges. Instead, the weight on the control W_u degrades the asymptotic performance, but may be useful for limiting the control action to prevent actuator saturation, particularly for non-minimum phase systems. ⌟

IV. Simulation results

In this section we compare in simulation the proposed approach with the Kalman-filter-based ILC (K-ILC) presented in [8]. We apply these algorithms on a 1D mass-spring-damper system using the nominal model:

$$G(s) = \frac{\omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2} \tag{9}$$

with natural frequency $\omega_n = 1.8\ rad/s$ and damping ratio $\xi = 0.8$. The algorithms performance is evaluated in a realistic scenario in which the system is affected by parametric uncertainties. Further, as in [8], we add a disturbance $d(t) = 0.5\sin^2(t)$ and a Gaussian noise $\mathcal{N} \sim (0, 0.01)$ to the output signal $y(t)$. The learning parameters chosen are listed in Table I and the desired trajectory is $y_d(t) = \sin(0.5\pi t)$.

TABLE I

Parameters used in the simulations.

	W_u	$W_{\Delta u}$	W_e	$P(0)$	R	Q
Proposed	0_N	0.01 I_N	I_N	0.1	0.01	0.1
K-ILC	0_N	0.01 I_N	I_N	0.1 I_N	0.01 I_N	0.1 I_N

Firstly, a Monte Carlo study has been carried out with respect to uncertainty (assuming standard deviation equal to 5% of the nominal values) on the natural frequency and on the damping ratio. The learning process (30 iterations of ILC algorithm with a sampling time $t_{ILC} = 0.01\ s$) has been performed simulating different plants (100 dynamic parameters samples), maintaining the same model in the learning algorithm. For each plant the results have been averaged repeating the entire process 10 times to reduce the noise influence. The most relevant statistics (mean and standard deviation) of the error 2-norm ($\|E_j\|_2$) at the first iteration and at steady-state[4] are collected in Table II.

TABLE II

Error 2-norm with 5% level of uncertainty.

Algorithm	First Iteration	Steady-State
Proposed	0.84 ± 0.10	0.23 ± 0.01
K-ILC	1.46 ± 0.21	0.23 ± 0.01

To visualize the algorithms rate of convergence, the error evolution in the iteration domain of a learning cycle[5] is shown in Figure 2. We can see that the proposed approach achieves a faster convergence with respect to K-ILC maintaining similar steady-state performance.

[4]The steady-state error norm was approximated by the average error norms of iterations 20 to 30.

[5]The error evolution for most of the learning cycles has a behaviour similar to the one in Figure 2.

After that, 3 different Monte Carlo studies have been carried out with respect to larger uncertainties (standard deviation equal respectively to 10%, 20%, 40% of the nominal values) on the dynamic parameters to assess the robustness of the proposed approach. The results are summarized in Table III. We can notice that the proposed approach is capable to improve the performance also with a considerable error in the parameters (20% standard deviation).

However, as expected from *Remark 1*, when the model is very inaccurate (40% standard deviation), K-ILC outperforms our approach at steady-state due to the deterministic error caused by an imperfect knowledge of the model.

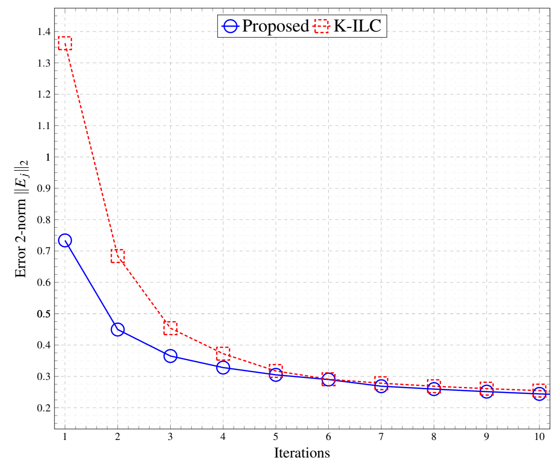

Fig. 2. Error 2-norm evolution in the iteration domain (first 10 iterations).

TABLE III

Error 2-norm with different levels of uncertainty.

Algorithm	Uncertainty level	First Iteration	Steady-State
Proposed	±10%	1.30 ± 0.32	0.23 ± 0.01
	±20%	2.39 ± 0.65	0.23 ± 0.01
	±40%	4.33 ± 2.53	0.38 ± 0.25
K-ILC	±10%	1.99 ± 0.42	0.23 ± 0.01
	±20%	2.74 ± 0.66	0.23 ± 0.01
	±40%	4.10 ± 2.16	0.28 ± 0.18

V. Experimental results

In this section we apply the proposed ILC algorithm and the K-ILC to a multirotor UAV to achieve high performance tracking. Firstly we specialized the algorithm for this task and then we present the results obtained in the experiments.

A. Applying ILC to UAV trajectory tracking

The ILC scheme of Section III can be applied to any dynamic systems with underlying feedback loops. Specifically, it is applied to improve the tracking accuracy in executing a complex manoeuver with an UAV guided by a commercial controller.

Remark 3: As pointed out in Section II-A, the ILC scheme requires the disturbance to be iteration-invariant (or at least slowly iteration-varying). In this application, this can be reasonably assumed true, because the ILC objective is to achieve a high level of performance eliminating the unmodelled (possibly nonlinear) dynamics (*e.g.,* aerodynamic effects). Other iteration-varying disturbances (*e.g.,* wind) have to be counteracted by the feedback controller. Furthermore, the feedback loop allows us to use a linear model as a good approximation of the UAV closed-loop dynamics. ⌟

In this work we consider without loss of generality the case of in-plane trajectory ($y_d = [x_d,\ y_d]^T$). This is done by acting on the set-points ($u = [x_{sp},\ y_{sp}]^T$) commanded to the stock controller exploiting measurements of the position ($y = [x_m,\ y_m]^T$) to estimate the disturbance.

To analyse the learning performance, the following metrics are defined. A synthetic indicator at a specific iteration j is the *average position error* along the trajectory:

$$e_{pos,j} = \frac{1}{N}\sum_{k=1}^{N}\sqrt{(x_{m,j}(k)-x_d(k))^2+(y_{m,j}(k)-y_d(k))^2}\ . \tag{10}$$

This indicator can be adimentionalized with respect the average position error at the initial iteration $\frac{e_{pos,j}}{e_{pos,0}}$. Additionally the performance indicator $\frac{e_{pos,j}-e_{pos,j-1}}{e_{pos,j}}$ has been used to highlight the difference between two consecutive iterations.

B. Experimental setup

Flight tests are carried out inside the Flying Arena for Rotorcraft Technologies (FlyART) of Politecnico di Milano which is an indoor facility equipped with a Motion Capture system (Mo-Cap). The drone is a fixed-pitch quadrotor designed by ANT-X [20], with a maximum take-off weight below 300g. Data is collected through the Mo-Cap system composed by 12 cameras which detect markers mounted on the drones. A ground control station receives measurements from the Mo-Cap system, reconstructs the state of the drone and, then, computes the next iteration set-points according to the proposed approach. The overall strategy has been integrated in the PX4 autopilot [21] using the ANT-X rapid prototyping system for multirotor control.

C. Model Identification

The system dynamics (*i.e.*, the complementary sensitivity of the UAV) was identified as a black-box model, by applying the Predictor-Based Subspace Identification algorithm (PBSID, see also [22], [23], [24] for applications of PBSID to rotorcraft dynamics) to input-output data gathered in dedicated identification experiments. The model for the transfer function from x_{sp} to x_m is[6]

$$G_x(s) = e^{-0.25s}\cdot\frac{0.08s+1.94}{s^2+1.752s+2.01}. \tag{11}$$

The model was validated against flight data collected in another experiment. Figure 3 shows the measured response to the reference signal against the simulated response obtained with the identified model, showing a close match to the measured data.

[6]The PBSID algorithm returns a discrete time state space model, but the continuous time transfer function obtained with the Tustin approximation has been reported to give more physical insight. In fact, in this formulation we can clearly notice that the system is characterized by a natural frequency $\omega_n \approx 1.4\ rad/s$ and damping ratio $\xi \approx 0.6$.

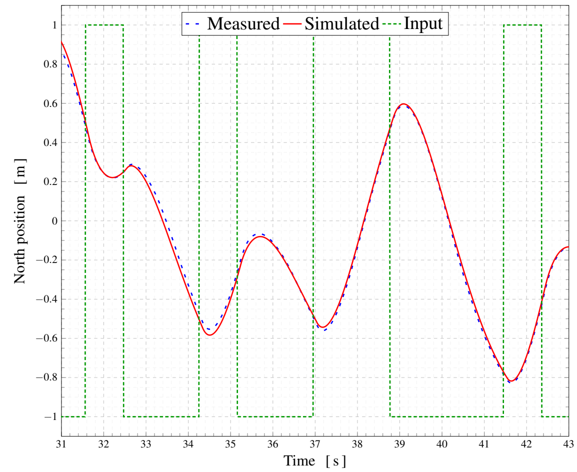

Fig. 3. Validation of the identified model.

Note that due to symmetry, and based on previous experience, we can use the model identified for longitudinal dynamics (x-direction) also for the lateral dynamics (y-direction), *i.e.*, assume that ($G_x(s) = G_y(s)$).

These models are used in the estimation step (time-domain) and in the input update step (iteration-domain) constructing **F** following the approach in [12].

D. Results

In this section, we present the results obtained by applying the proposed algorithm and the K-ILC on the ANT-X UAV with the same learning parameters used in Section IV and with a sampling time $t_{ILC} = 0.05\ s$. In the experiment the manoeuvre to be learned is an aggressive eight-shape trajectory flown in the horizontal plane characterized by a maximum velocity $v_{max} = 2.1\,\mathrm{m/s}$ and maximum acceleration $a_{max} = 4\,\mathrm{m/s^2}$. The quadrotor is required to hover at the beginning of the eight-shape trajectory: the learning motion starts and ends in the same hovering point. The acceleration and deceleration phases at the beginning and the end of the eight-figure must also be learned precisely. The data comparison between the two methods is reported in Table IV. We can notice that, as in Section IV, the proposed approach achieves faster convergence with respect to K-ILC.

TABLE IV
LEARNING PERFORMANCE RESULTS FOR 8-SHAPE TRAJECTORY.

Iteration	e_{pos_j} [m]	$\frac{e_{pos_j}}{e_{pos_0}}\cdot 100$ [%]	$\frac{\lvert e_{pos_j}-e_{pos_{j-1}}\rvert}{\lvert e_{pos_{j-1}}\rvert}\cdot 100$ [%]
$j=0$	0.422	100.0	-
Proposed approach			
$j=1$	0.040	9.59	90.41
$j=2$	0.018	4.29	55.33
K-ILC			
$j=1$	0.072	17.2	82.84
$j=2$	0.050	11.9	30.52
$j=3$	0.044	10.5	12.13

Due to space limitations only the trajectories obtained by to the proposed approach are plotted in the following figures.

Specifically, in Figure 4 the evolution in xy-plane is depicted without the acceleration and deceleration phases, while in Figure 5 the time evolution of the North position is plotted (the East direction shares similar behaviour).

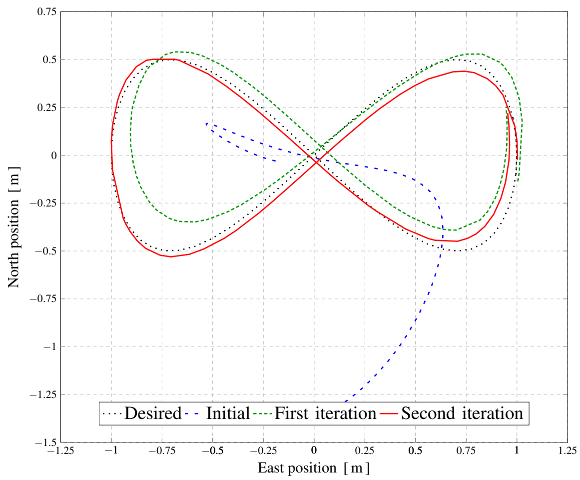

Fig. 4. 8-shape trajectories for the iterations 0, 1 and 2.

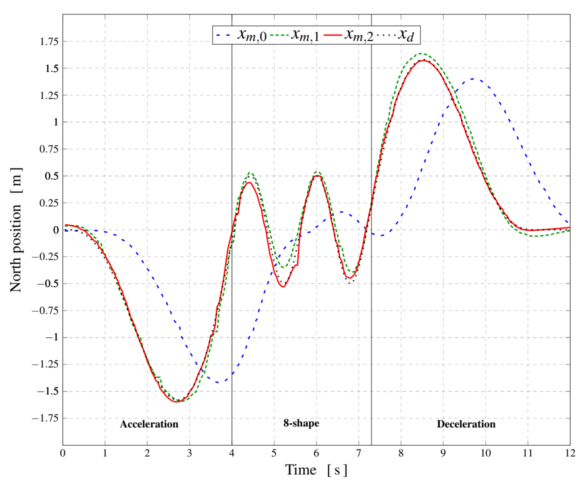

Fig. 5. UAV North-position for iterations 0, 1 and 2.

We can state that the proposed approach is very effective in improving the tracking performance of the UAV, while being robust to small initial errors in the positioning, unmodelled system dynamics, process and measurement noises.

VI. Conclusions

In this letter, we tackled the problem of high-performance tracking of UAVs for the solution of which an ILC-based control approach has been developed. The novelty of this work is the introduction of a smoother to estimate the repetitive disturbance and the usage of a black-box identification procedure to obtain an accurate system model. A Monte Carlo analysis has been carried out to show the performance improvement and limits of the proposed algorithm with respect to existing approaches. Finally, an experimental campaign involving a small quadrotor has shown the effectiveness of the proposed strategy.

References

[1] M.-H. Hua, T. Hamel, P. Morin, and C. Samson, "Introduction to feedback control of underactuated VTOL vehicles: A review of basic control design ideas and principles," *IEEE Control Systems Magazine*, vol. 33, no. 1, pp. 61–75, 2013.

[2] H. Ahn, Y. Chen, and K. Moore, "Iterative Learning Control: Brief Survey and Categorization," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 37, no. 6, pp. 1099–1121, Nov 2007.

[3] D. Bristow, M. Tharayil, and A. Alleyne, "A survey of Iterative Learning Control," *IEEE Control Systems Magazine*, vol. 26, no. 3, pp. 96–114, June 2006.

[4] R. Adlakha and M. Zheng, "Two-Step Optimization-Based Iterative Learning Control for Quadrotor UAVs," *Journal of Dynamic Systems, Measurement, and Control*, 2021.

[5] M. Cobb, K. Barton, H. Fathy, and C. Vermillion, "An Iterative Learning Approach for Online Flight Path Optimization for Tethered Energy Systems Undergoing Cyclic Spooling Motion," in *2019 American Control Conference (ACC)*, 2019, pp. 2164–2170.

[6] A. Schoellig and R. D'Andrea, "Optimization-based Iterative Learning Control for trajectory tracking," in *2009 European Control Conference (ECC)*, 2009, pp. 1505–1510.

[7] D. Fraser and J. Potter, "The optimum linear smoother as a combination of two optimum linear filters," *IEEE Transactions on Automatic Control*, vol. 14, no. 4, pp. 387–390, 1969.

[8] N. Degen and A. Schoellig, "Design of norm-optimal Iterative Learning controllers: The effect of an iteration-domain Kalman filter for disturbance estimation," in *53rd IEEE Conference on Decision and Control*, 2014, pp. 3590–3596.

[9] M. Norrlof, "An adaptive Iterative Learning Control algorithm with experiments on an industrial robot," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 2, pp. 245–251, 2002.

[10] B. Panomruttanarug and R. Longman, "Using Kalman filter to attenuate noise in learning and repetitive control can easily degrade performance," in *2008 SICE Annual Conference*, 2008, pp. 3453–3458.

[11] G. V. der Veen, J.-W. van Wingerden, M. Bergamasco, M. Lovera, and M. Verhaegen, "Closed-loop subspace identification methods: an overview," *IET Control Theory and Applications*, vol. 7, no. 10, pp. 1339–1358, 2013.

[12] R. Longman, "Iterative Learning Control and Repetitive Control for engineering practice," *International Journal of Control*, vol. 73, no. 10, pp. 930–954, 2000.

[13] S. Gunnarsson and M. Norrlof, "On the design of ILC algorithms using optimization," *Automatica*, vol. 37, no. 12, pp. 2011–2016, 2001.

[14] J. Crassidis and J. Junkins, *Optimal Estimation of Dynamic Systems*. Chapman and Hall/CRC, 2011.

[15] H. Elci, R. Longman, M. Phan, J.-N. Juang, and R. Ugoletti, "Simple learning control made practical by zero-phase filtering: applications to robotics," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 49, no. 6, pp. 753–767, 2002.

[16] A. Bryson and M. Frazier, "Smoothing for linear and nonlinear dynamic systems," in *Proceedings of the optimum system synthesis conference*, 1963, pp. 353–364.

[17] H. E. Rauch, F. Tung, and C. Striebel, "Maximum likelihood estimates of linear dynamic systems," *AIAA Journal*, vol. 3, no. 8, pp. 1445–1450, 1965.

[18] D. Gorinevsky, "Loop shaping for Iterative Control of batch processes," *IEEE Control Systems Magazine*, vol. 22, no. 6, pp. 55–65, 2002.

[19] D. Owens and J. Hätönen, "Iterative Learning Control — An optimization paradigm," *Annual Reviews in Control*, vol. 29, no. 1, pp. 57–70, 2005.

[20] *ANT-X Website*. [Online]. Available: https://antx.it/.

[21] *PX4 community*, Dronecode Project, Inc., San Francisco, CA, USA. [Online]. Available: https://docs.px4.io/en/.

[22] M. Bergamasco and M. Lovera, "Continuous-time predictor-based subspace identification for helicopter dynamics," in 37th European Rotorcraft Forum, Gallarate, Italy, 2011.

[23] J. Wartmann and S. Seher-Weiss, "Application of the Predictor-Based Subspace Identification Method to Rotorcraft System Identification," in 39th European Rotorcraft Forum, Moscow, Russia, 2013.

[24] M. Bergamasco and M. Lovera, "Identification of Linear Models for the Dynamics of a Hovering Quadrotor," *IEEE Transactions on Control Systems Technology*, vol. 22, no. 5, pp. 1696–1707, 2014.