Knowledge Transfer between Different UAVs for Trajectory Tracking

Zhu Chen¹, Xiao Liang², and Minghui Zheng¹

Abstract-Robots are usually programmed for particular tasks with a considerable amount of hand-crafted tuning work. Whenever a new robot with different dynamics is presented. the well-designed control algorithms for the robot usually have to be re-tuned to guarantee good performance. It remains challenging to directly program a robot to automatically learn from the experiences gathered by other dynamically different robots. With such a motivation, this paper proposes a learning algorithm that enables a quadrotor unmanned aerial vehicle (UAV) to automatically improve its tracking performance by learning from the tracking errors made by other UAVs with different dynamics. This learning algorithm utilizes the relationship between the dynamics of different UAVs, named the target and training UAVs, respectively. The learning signal is generated by the learning algorithm and then added to the feedforward loop of the target UAV, which does not affect the closedloop stability. The learning convergence can be guaranteed by the design of a learning filter. With the proposed learning algorithm, the target UAV can improve its tracking performance by learning from the training UAV without baseline controller modifications. Both numerical studies and experimental tests are conducted to validate the effectiveness of the proposed learning algorithm.

I. INTRODUCTION

Unmanned aerial vehicles (UAVs) such as quadrotors have been widely used to perform various tasks [1]-[4]. Many of these tasks are conducted in a cluttered environment which requires accurate trajectory tracking. As such, extensive efforts in control area have been made to guarantee UAV's tracking performance. These methods include feedback control (e.g., proportional-integral-derivative control [5], linear quadratic regulator [6], disturbance observer [7], and sliding mode control [8]) and feedforward control such as iterative learning control (ILC) [9], [10]. In practice, the design of these model-based controllers usually encounters a considerable amount of hand-crafted tuning work for almost every distinct UAV. Whenever a new UAV with different dynamics is presented, the well-designed control algorithms usually have to be re-tuned to guarantee satisfactory tracking performance. It remains challenging to enable a UAV to automatically learn from the experiences gathered by another with different dynamics.

This paper presents a new learning algorithm that enables knowledge transfer for trajectory tracking between UAVs with different dynamics, i.e., heterogeneous UAVs. This learning algorithm utilizes the relationship between the dynamics of heterogeneous UAVs, named the target and training UAVs, respectively. The target UAV automatically improves its tracking performance by learning from the tracking errors made by the training UAV. The learning signal is generated by the learning algorithm and is added to the feedforward loop of the target UAV. This learning algorithm is motivated by ILC that allows the UAV to learn from its own historical data in repetitive operations. Compared with standard ILC, the proposed learning algorithm here augments the learning between different UAVs and is not limited to repetitive systems anymore. It is worth mentioning that this learning algorithm is not to replace or modify the existing baseline controller; it is an add-on algorithm to further improve the target UAV's tracking performance.

The contributions of this work are summarized as follows. (1) It develops a systematic model-based learning framework for knowledge transfer in terms of trajectory tracking between UAVs with different dynamics. (2) The learning algorithm removes the requirement of repetitive operations in standard ILC. (3) The learning algorithm is an add-on algorithm which does not affect the stability of the closedloop system. This learning algorithm is particularly useful (1) when the baseline control is not easy or not allowed to modify, (2) prior training for the target UAV is not possible, and (3) experience from another UAV with different dynamics is available. For example, the proposed algorithm can be used when two different UAVs are flying in a leaderfollower mode to execute certain tasks, in which the follower could utilize the flight data of the leader to improve its tracking performance. In summary, this learning algorithm can improve the tracking performance of the target UAV without modifying the baseline control by learning from the experience gained by the training UAV.

II. RELATED WORK

This section summarizes related existing work in learning control. These methods are mainly classified into three categories that leverage ILC, adaptive control, and data-driven methods, respectively.

A. ILC-based methods

ILC is a feedforward control technique that enables a repetitively operated system to learn from its previous iterations [11] and has been developed and applied to UAVs [12], [13] to improve the trajectory tracking performance. Traditional ILC requires that the system operates in a repetitive way, which means the following conditions need to be satisfied: (1) the system has an identical initial state for each iteration [14]; (2) the system has no iteration-varying

¹Zhu Chen and Minghui Zheng are with the Department of Mechanical and Aerospace Engineering, University at Buffalo, Buffalo, NY, 14260 USA zhuchen@buffalo.edu, mhzheng@buffalo.edu.

 $^{^2 \}rm Xiao$ Liang is with the Department of Civil, Structural and Environmental Engineering, University at Buffalo, Buffalo, NY, 14260 USA liangx@buffalo.edu.

modeling uncertainties [15]; (3) the system performs each iteration with a fixed trial length [16]; and (4) the reference trajectory for each iteration is identical [17]. Moreover, whenever the reference is changed, the learning process has to start from the first iteration. Many efforts have been made to relax these conditions in order to have broader applications (e.g., [17]–[19]). For example, a linear map method [19] has been designed such that the prior tracking knowledge can be used to reduce the initial tracking error for an unseen trajectory. Similar methods have been proposed in [20], [21] as well. Nevertheless, these ILCs and their variants are limited within the same systems.

B. Adaptive control-based methods

While adaptive control has been widely applied to UAVs to address unknown parameters (e.g., [22]-[24]), here we focus on the ones that leverage adaptive control to enable learning between different systems (e.g., [25], [26]). An L_1 adaptive controller is incorporated into ILC to enable the learning between different UAV systems [25], in which adaptive feedback control is purposely designed to make different systems behave as the same reference model, while the feedforward ILC improves system performance over iterations. Nevertheless, the learning is conducted via the same "reference model" over iterations. Another work that leverages adaptive control to enable learning between different systems is presented in [26], in which the baseline control is adapted to compensate the model difference between the two systems. The above-mentioned methods need baseline feedback control being adapted during the flight to conduct the learning, which may affect the stability of the systems. Alternatively, the proposed learning algorithm in this paper is an add-on algorithm which does not modify baseline feedback control.

C. Data-driven methods

Data-driven methods, which usually do not explicitly rely on the dynamic models, have been utilized to develop control and learning algorithms for UAVs as well. By training and learning processes, UAVs can make high-level decisions (e.g., generating desired paths) and execute low-level tasks (e.g., trajectory tracking). For example, reinforcement learning and deep learning techniques have been proposed for UAV's control [27]–[30]. Reinforcement learning has also been utilized to transfer task information among multiple manipulator systems [31]. While these data-driven methods provide additional flexibility to the control of robotic systems, they are usually highly dependent on the training data and do not explicitly consider UAV's dynamics in the learning algorithm design.

III. LEARNING ALGORITHM

A. Quadrotor dynamics

Quadrotor UAV's dynamics is highly nonlinear and the system's baseline controller design is not trivial. Before a quadrotor is ready to fly, the baseline controllers for both attitude and position loops have to be carefully designed and tuned to guarantee the desired trajectory tracking performance. More detailed introduction of quadrotor dynamics can be referred to [32].

It is worth noting that the proposed learning algorithm does not require baseline control modification, so the learning algorithm is not explicitly dependent on the baseline control. Alternatively, we assume that both UAV systems are already stabilized by their baseline feedback controllers, and treat the plant and the baseline control as a whole closedloop system. Therefore, there is no particular constraint on which type of the baseline control (e.g., cascaded positionattitude control or full-state model predictive control) should be implemented in the actual UAV system. Furthermore, considering that the bandwidth of the attitude control loop is much higher than that of the position loop, the closedloop trajectory tracking dynamics can be approximated as a linear time-invariant (LTI) system [32], [33]. In this study, the proposed learning algorithm is only for the position loop of the UAV while its attitude loop is not explicitly considered.

B. Overview of the proposed learning algorithm



Fig. 1: Overview of the proposed learning algorithm. The trajectory tracking experience of the training UAV is learned by the target UAV. The training UAV and the target UAV are with different dynamics.

The proposed learning algorithm is illustrated in Fig. 1. There are two UAVs involved: one target UAV and one training UAV. The two UAVs are with different physical properties (e.g., mass, size, propellers) such that their dynamic models are different. A desired reference trajectory is provided to both UAVs. The training UAV tracks the reference first; no matter how the tracking performance is, the tracking error is recorded and sent to the learning module (learning filter). The learning module generates the learning signal that will be sent to the target UAV when it tracks the same reference. The goal is to improve the tracking performance of the target UAV by learning from the tracking error made by the training UAV.

C. Learning algorithm: mathematical details

This subsection presents mathematical details of the learning algorithm. Assume both UAV systems are stabilized by their baseline controllers. Denote P_0 (and P) as the positionloop dynamics of the training (and target) UAV with the baseline controller incorporated. As explained in Subsection A, P_0 and P are approximated as LTI systems. The proposed learning algorithm is given in Fig. 2, where L is the to-bedesigned learning filter, r_d is the reference trajectory, v is



Fig. 2: Proposed learning algorithm. The tracking error of the training UAV goes through the learning filter to generate the learning signal, and the learning signal is added to the feedforward loop of the target UAV.

the output of P_0 ; e_0 and h are the input and output of L, respectively; define y_1 (and y_2) as the output, and e_1 (and e_2) as the tracking error of the target UAV when it tracks r_d without (and with) the learning signal. Denote k as the discrete-time index. Assume P_0 has the following state-space realization

$$x_{0}(k+1) = A_{w}x_{0}(k) + B_{w}r_{d}(k)$$

$$v(k) = C_{w}x_{0}(k)$$

$$e_{0}(k) = r_{d}(k) - v(k)$$
(1)

where (A_w, B_w, C_w) are the system matrix, input matrix and output matrix of P_0 , respectively; x_0 is the state variable of P_0 . Assume the learning filter *L* has the following state-space realization

$$\begin{aligned}
\kappa_l(k+1) &= A_l x_l(k) + B_l e_0(k) \\
h(k) &= C_l x_l(k) + D_l e_0(k)
\end{aligned}$$
(2)

where x_l denotes the state variable of L, h denotes the output of the learning filer, and (A_l, B_l, C_l, D_l) are the system matrix, input matrix, output matrix, and feedforward matrix of L, respectively. Assume P has the following state-space realization

$$P \sim \left[\begin{array}{c|c} A & B \\ \hline C & 0 \end{array} \right] \tag{3}$$

where (A, B, C) are the system matrix, input matrix, and output matrix of P, respectively. When there is no learning signal h injected to P, the system can be represented as

$$x_{1}(k+1) = Ax_{1}(k) + Br_{d}(k)$$

$$y_{1}(k) = Cx_{1}(k)$$

$$e_{1}(k) = r_{d}(k) - y_{1}(k)$$
(4)

where x_1 is the state variable of *P* without *h* injected, i.e., without learning. When the learning signal *h* is added to the feedforward loop of *P*, the system can be represented as

$$x_{2}(k+1) = Ax_{2}(k) + B(r_{d}(k) + h(k))$$

$$y_{2}(k) = Cx_{2}(k)$$

$$e_{2}(k) = r_{d}(k) - y_{2}(k)$$
(5)

where x_2 is the state variable of *P* with *h* injected.

As mentioned above, the goal of this learning algorithm is to design a learning filter L which can improve the tracking performance of the target UAV by learning from the experience (e.g., the tracking error) of the training UAV. To relate the tracking errors e_1 with e_2 which are tracking errors made by the target UAV without and with learning, we define the system T_e such that

$$e_2 = T_e\{e_1\}\tag{6}$$

where the operator $\{\cdot\}$ describes the system response, that is, with an input e_1 to the system T_e , the output is e_2 . The system T_e can be represented as

$$T_e \sim \left[\begin{array}{c|c} A_e & B_e \\ \hline C_e & D_e \end{array} \right] \tag{7}$$

where (A_e, B_e, C_e, D_e) are the system matrix, input matrix, output matrix, and feedforward matrix of T_e , respectively.

System T_e lumps the system dynamics of P_0 , P and L. To obtain T_e explicitly for further system analysis, we introduce the following variable

$$\tilde{x}(k) = x_2(k) - x_1(k)$$
 (8)

By selecting x_1 , x_0 , x_l , \tilde{x} as the state variables, we derive the state-space realization of T_e as follows.

From (4) we have

$$r_d(k) = e_1(k) + y_1(k) = e_1(k) + Cx_1(k)$$
(9)

and

$$x_1(k+1) = Ax_1(k) + B(e_1(k) + Cx_1(k))$$
(10)

Plug (9) into (1) to have

$$x_0(k+1) = A_w x_0(k) + B_w r_d(k)$$

= $A_w x_0(k) + B_w (e_1(k) + C x_1(k))$ (11)

Plug (1) into (2) to cancel the $e_0(k)$ term and re-organize (2) as

$$x_{l}(k+1) = A_{l}x_{l}(k) + B_{l}(r_{d}(k) - v(k))$$

= $A_{l}x_{l}(k) + B_{l}(e_{1}(k) + Cx_{1}(k) - C_{w}x_{0}(k))$ (12)

From (8), we have

$$\tilde{x}(k+1) = x_2(k+1) - x_1(k+1)$$

= $[Ax_2(k) + B(r_d(k) + h(k))] - [(Ax_1(k) + Br_d(k))]$
= $A(x_2(k) - x_1(k)) + Bh(k)$ (13)

Plug (2) into (13) to cancel the h(k) term, and re-organize (13) as

$$\begin{aligned} \tilde{x}(k+1) &= A\tilde{x}(k) + B(C_l x_l(k) + D_l e_0(k)) \\ &= A\tilde{x}(k) + B(C_l x_l(k) + D_l(r_d(k) - v(k))) \\ &= A\tilde{x}(k) + B(C_l x_l(k) + D_l(e_1(k) + Cx_1(k) - C_w x_0(k))) \end{aligned}$$
(14)

With (4), (5) and (8), we have

$$e_{2}(k) - e_{1}(k) = (r_{d}(k) - y_{2}(k)) - (r_{d}(k) - y_{1}(k))$$

= $-(y_{2}(k) - y_{1}(k))$
= $-C(x_{2}(k) - x_{1}(k))$
= $-C\tilde{x}(k)$ (15)

With (10), (11), (12), (14) and (15), the state-space realization of T_e can be presented as

$$\begin{bmatrix} x_{1}(k+1) \\ x_{0}(k+1) \\ x_{l}(k+1) \\ \tilde{x}(k+1) \end{bmatrix} = \begin{bmatrix} A+BC & 0 & 0 & 0 \\ B_{w}C & A_{w} & 0 & 0 \\ B_{l}C & -B_{l}C_{w} & A_{l} & 0 \\ BD_{l}C & -BD_{l}C_{w} & BC_{l} & A \end{bmatrix} \begin{bmatrix} x_{1}(k) \\ x_{l}(k) \\ \tilde{x}(k) \end{bmatrix}$$
$$+ \begin{bmatrix} B \\ B_{w} \\ B_{l}C \\ BD_{l} \end{bmatrix} e_{1}(k)$$
$$e_{2}(k) = \begin{bmatrix} 0 & 0 & 0 & -C \end{bmatrix} \begin{bmatrix} x_{1}(k) \\ x_{0}(k) \\ x_{l}(k) \\ \tilde{x}(k) \end{bmatrix} + e_{1}(k)$$
$$(16)$$

Then (7) can be presented as

$$T_{e} \sim \begin{bmatrix} A + BC & 0 & 0 & 0 & B \\ B_{w}C & A_{w} & 0 & 0 & B_{w} \\ B_{l}C & -B_{l}C_{w} & A_{l} & 0 & B_{l}C \\ BD_{l}C & -BD_{l}C_{w} & BC_{l} & A & BD_{l} \\ \hline 0 & 0 & 0 & -C & 1 \end{bmatrix}$$
(17)

and the system T_e from e_1 to e_2 is obtained. To improve the target UAV's tracking performance, that is, to make $||e_2|| < ||e_1||$, the following conditions should be satisfied

• (a) all the eigenvalues of the system matrix of T_e are within the unit circle, i.e.,

$$|\lambda_i(A_e)| < 1, \ \forall \ i \tag{18}$$

where $\lambda_i(A_e)$ is the *i*th eigenvalue of A_e .

• (b) The minimum γ is less than 1 where γ satisfies

$$\bar{\sigma}\{D_e + C_e(\eta I - A_e)^{-1}B_e\} < \gamma \quad \forall |\eta| > 1$$
(19)

and $\bar{\sigma}\{\cdot\}$ denotes the maximum singular value of a matrix.

D. Learning algorithm: design guidelines and analysis

It is worth noting that it is very challenging to directly design (A_l, B_l, C_l, D_l) such that the above conditions in (18) and (19) are satisfied. To simplify this design procedure, we compactly use the $\{\cdot\}$ operator, and re-derive the dynamic system T_e and represent it in terms of dynamic systems (P_0, P, L) explicitly as follows.

As illustrated in Fig. 2, we have

$$e_0 = r_d - P_0\{r_d\}$$
(20)

$$e_1 = r_d - P\{r_d\} \tag{21}$$

$$e_2 = r_d - P\{r_d + h\}$$
(22)

in which all the signals (e_0, e_1, e_2, r_d, h) are time series. The reference r_d can be represented as

$$r_d = (1 - P_0)^{-1} \{ e_o \} = (1 - P)^{-1} \{ e_1 \}$$
(23)

and we have

$$e_{2} = r_{d} - P\{r_{d}\} - PL\{e_{0}\}$$

= $r_{d} - P\{r_{d}\} - PL\{r_{d} - P_{0}\{r_{d}\}\}$
= $e_{1} - PL(1 - P_{0})\{r_{d}\}$ (24)
= $e_{1} - PL(1 - P_{0})(1 - P)^{-1}\{e_{1}\}$
= $[(1 - PL(1 - P_{0})(1 - P)^{-1})]\{e_{1}\}$

Considering (6), it is worth noting that the dynamic system in (24) is equivalent to the one in (16). In the following, we utilize the representation of (24) to design the learning filter L and check whether the conditions of (18) and (19) are satisfied.

Ideally, if *L* can make the system of $PL(1-P_0)(1-P)^{-1}$ close to 1, then $||e_2||$ would be close to zero. Therefore, we target to design a *L* such that

$$PL(1-P_0)(1-P)^{-1} \approx 1$$
 (25)

which implies

$$L \approx P^{-1}(1-P)(1-P_0)^{-1}$$
(26)

With (3), the state-space realization of 1-P is

$$1 - P \sim \left[\begin{array}{c|c} A & B \\ \hline -C & 1 \end{array} \right] \tag{27}$$

Similarly, with (1) we have

$$1 - P_0 \sim \begin{bmatrix} A_w & B_w \\ -C_w & 1 \end{bmatrix}$$
(28)

and the system $(1-P_0)^{-1}$ can be represented as

$$(1-P_0)^{-1} \sim \left[\begin{array}{c|c} A_w + B_w C_w & B_w \\ \hline C_w & 1 \end{array} \right]$$
 (29)

While it is easier to directly obtain the state space realization for $(1-P)(1-P_0)^{-1}$, the remaining design part P^{-1} will be manually tuned such that (A_l, B_l, C_l, D_l) satisfy the two conditions in (18) and (19).

Stability: The learning signal is generated by the learning algorithm and then added to the feedforward loop of the target UAV. Therefore, the learning signal only modifies the reference and does not affect the closed-loop stability of the target UAV. The target UAV system remains stable when the learning algorithm is added.

Learning convergence: The learning convergence can be guaranteed by designing the learning filter such that conditions (18) and (19) are satisfied.

IV. VALIDATION

This section presents both numerical studies and experimental tests to validate the proposed learning algorithm. We (1) conduct the system identification for the target and training UAVs, (2) simulate the UAVs' tracking performance based on the identified models, and (3) conduct the experimental tests to validate the proposed learning algorithm.

A. Test platform

Two quadrotors are customized and assembled for experimental tests: one as the target UAV and the other as the training UAV. The two UAVs are different in terms of body dimension, motor to motor distance, net mass, frame structures, etc., as shown in Table I. These physical specifications will result in different dynamics of UAVs. The experimental platform, which consists of two UAVs and a Vicon motion capture system used to capture the pose of the UAVs, is shown in Fig. 3.

	Target UAV	Training UAV
Frame Brand	F450	S500
Body Dimension	363×363 ×120 mm	382×382 ×82 mm
Motor to Motor Distance (diagonal)	455 mm	480 mm
Net mass	1.2 kg	0.9 kg

TABLE I: Basic specifications of the target and training UAVs



Fig. 3: Experimental platform: Vicon motion capture system, Target and Training UAVs. Both UAVs fly with a thin and light power cable attached to them. The cable is wrapped with a light protective film. The influence of this cable to the system dynamics is negligible.

B. System identification

We first conduct the system identification for both UAVs. Since the proposed learning algorithm does not require the modification of the baseline controllers, the system identification has been conducted to estimate the closed-loop dynamics of both UAVs with their baseline controllers incorporated. Therefore, the dynamic models from the reference trajectory to the actual one are identified. Here we use the recursive least square based parameter adaptation algorithm [34]–[36] for system identification. In particular, an LTI model structure with unknown parameters is selected, and the actual input and output data of each UAV are collected. The objective function is chosen as the norm of the model prediction error, which is minimized via recursive least square method to obtain the optimal parameters [35]. In this study, since we test the learning algorithm for trajectory tracking in the vertical direction (z-direction) and the heading direction (x-direction), the UAVs' dynamic models in both directions are identified. The system identification for the training and target UAVs in the vertical direction is provided

in Figs. 4 and 5 respectively. Both figures show that the output of the identified model fits the output measurement of the actual UAV, which indicates that the identified model properly represents the UAV's dynamic model. It is noted that since system identification is an approximation method, small estimate errors are acceptable. The same system identification procedure is applied to estimate the UAV's dynamic models in the heading direction.



Fig. 4: System identification of the training UAV in the vertical direction. The top shows the input (given trajectory) and output (actual trajectory) measurements of the training UAV when it tracks the given trajectory in the vertical direction. The bottom shows the actual trajectory and the trajectory generated from the identified model when it is given the same reference trajectory.



Fig. 5: System identification of the target UAV in the vertical direction. The top shows the input (given trajectory) and output (actual trajectory) measurements of the target UAV when it tracks the given trajectory in the vertical direction. The bottom shows the actual trajectory and the trajectory generated from the identified model when it is given the same reference trajectory.

The bode plots of the identified dynamic models for both UAVs in the vertical direction are given in Fig. 6. Though the two UAVs share the same flying mechanism, it shows that their dynamic models in the vertical direction are quite different. Similar differences between the identified dynamic models of the two UAVs in the heading direction are observed, and the corresponding bode plots are not shown here to avoid redundancy.



Fig. 6: Bode plots of UAV's trajectory tracking dynamics in the vertical direction

C. Numerical verification

In this study, we consider two scenarios: (1) the two UAVs track an aggressive step input in the vertical direction, and (2) the two UAVs track a square reference in the 2D space (x-z plane). Numerical studies are performed based on the identified models for both training and target UAVs. In each scenario, the training UAV tracks the reference trajectory first, and its tracking error is recorded and sent to the learning filter to generate the learning signal. The learning signal is then added to the feedforward loop of the target UAV when the target UAV tracks the same reference trajectory. Both UAVs track the same reference trajectory only once and there is no repetitive training for each UAV. It is worth noting that: (1) no matter whether the tracking performance of the training UAV is better or worse than that of the target one, its tracking error will be sent to the proposed learning algorithm to improve the target UAV's tracking performance; (2) the two UAVs are not necessary to fly simultaneously or consecutively. As long as the flight data of the training UAV is recorded and stored, it can be utilized by the target one for performance enhancement.

Scenario 1: *Target UAV tracks a step reference in the vertical direction.* Fig. 7 provides the reference trajectory (i.e., the target reference), the tracking performance of the training UAV, the learning signal generated by the proposed learning algorithm, and the tracking performance of the target UAV with and without learning. It shows that the target UAV's trajectory tracking performance has been improved with the proposed learning algorithm.

Scenario 2: Target UAV tracks a square reference in the 2D space. The tracking performance is given in Fig. 8. It shows that the tracking performance of the target UAV has been improved in both directions with the learning algorithm. The improvement in the z-direction is more obvious than the improvement in the x-direction, which is because the x-direction baseline control of the target UAV is better than the z-direction baseline control.

D. Experimental validation

We also conduct experimental tests. In order to be consistent with the numerical studies, the same two scenarios



Fig. 7: Numerical study: trajectory tracking in the vertical direction



Fig. 8: Numerical study: trajectory tracking in 2D space

are performed in the experimental tests. In each scenario, both training and target UAVs hover at an altitude of 0.5 meters as the initial condition and then track the given references. While the training UAV is tracking the reference, the motion capture system captures the pose of the training UAV. The training UAV's tracking error is calculated and used to generate the learning signal which is then added to the feedforward loop of the target UAV. The detailed stepby-step implementation of the proposed learning algorithm is described in Algorithm 1, where the weighting factor w_l is a scalar to scale down the learning signal before being injected to the target UAV. Such a factor is to make the learning process robust and less aggressive in consideration of modeling uncertainties in actual experimental tests. In each scenario, the trajectory tracking of the target UAV without learning is also conducted for performance comparison purpose.

Scenario 1: Target UAV tracks a step reference in the vertical direction. Fig. 9 shows the tracking performance of the target UAV with and without learning, as well as the learning signal generated by the proposed learning algorithm. It shows that the transient tracking performance $(0 \sim 4s)$ of the target UAV is greatly enhanced by the learning algorithm,

Algorithm 1 Learning algorithm

Inputs:

- **1**. Dynamic models P, P_0
- **2**. Weighting filter w_l
- 3. Learning filter L

Initialization: Set trajectory r_d for training and target UAVs

if $t < T_s$ then

| Training UAV is flying while Target UAV is waiting. else

- while Training UAV is flying do
 - 1. Calculate Training UAV's tracking error $e_0(t)$
 - 2. Training UAV sends $e_0(t)$ to Target UAV
 - 3. The learning algorithm makes $e_0(t)$ go through L
 - and generates the learning signal *h*: $h = L\{e_0(t)\}$
 - 4. Target UAV weights the learning signal by w_l : $h = w_l \{h\}$

5. The learning signal h is added to the feedforward loop of the target UAV

end

Target UAV learns from Training UAV's experience and tracks the same reference r_d .

end

which validates the effectiveness of the proposed learning method. It is noted that the target UAV reaches the reference signal around 4s, and stays around with small variations in an acceptable range.



Fig. 9: Experimental test: trajectory tracking in the vertical direction

Scenario 2: Target UAV tracks a square reference in the 2D space. The tracking performance is given in Figs. 10-12. Fig. 10 shows that the z-direction baseline controller is not good, and the tracking error has been significantly reduced by using the proposed learning algorithm. Fig. 11 shows that the x-direction baseline controller is good, and still, the tracking performance in the x-direction can be further enhanced by using the learning algorithm. Fig. 12 compares the 2D trajectory tracking without and with learning. The experimental video is available via the following link: http://zh.eng.buffalo.edu/

PaperDemo/KnowledgeTransfer.mp4.



Fig. 10: Experimental test: trajectory tracking in the vertical direction with a time-varying reference signal



Fig. 11: Experimental test: trajectory tracking in the heading direction with a time-varying reference signal



Fig. 12: Experimental test: trajectory tracking in 2D space.

V. CONCLUSIONS

This paper presents a new learning algorithm that can transfer the trajectory tracking experience from one UAV

to another with different dynamics. It aims to automatically improve the target UAV's trajectory tracking performance by learning from another UAV without modifying the target UAV's baseline controller. This learning algorithm generates a learning signal which can be directly added to the feedforward loop of the target UAV and enhances its tracking performance. Essentially, this learning algorithm is an augmented ILC: it extends the property of "learning from its own experiences" in standard ILC to the property of "learning from the experience of others even with different dynamics". It establishes a learning mechanism that can be conducted among heterogeneous systems, i.e., systems with different dynamics. This paper provides detailed formulation, derivation, and analysis for the proposed learning algorithm and validates its effectiveness via both numerical studies and experimental tests.

REFERENCES

- J. Y. Chen, "UAV-guided navigation for ground robot tele-operation in a military reconnaissance environment," *Ergonomics*, vol. 53, no. 8, pp. 940–950, 2010.
- [2] X. Liang, "Image-based post-disaster inspection of reinforced concrete bridge systems using deep learning with Bayesian optimization," *Computer-Aided Civil and Infrastructure Engineering*, vol. 34, no. 5, pp. 415–430, 2019.
- [3] E. T. Bokeno, T. M. Bort, S. S. Burns, M. Rucidlo, W. Wei, D. L. Wires *et al.*, "Package delivery by means of an automated multi-copter UAS/UAV dispatched from a conventional delivery vehicle," Mar. 13 2018, uS Patent 9,915,956.
- [4] S. O. Sajedi and X. Liang, "A Convolutional Cost-Sensitive Crack Localization Algorithm for Automated and Reliable RC Bridge Inspection," arXiv preprint arXiv:1905.09716, 2019.
- [5] A. L. Salih, M. Moghavveni, H. A. Mohamed, and K. S. Gaeid, "Modelling and PID controller design for a quadrotor unmanned air vehicle," in 2010 IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR), vol. 1. IEEE, 2010, pp. 1–5.
- [6] L. M. Argentim, W. C. Rezende, P. E. Santos, and R. A. Aguiar, "PID, LQR and LQR-PID on a quadcopter platform," in 2013 International Conference on Informatics, Electronics and Vision (ICIEV). IEEE, 2013, pp. 1–6.
- [7] M. Zheng, X. Lyu, X. Liang, and F. Zhang, "A generalized design method for learning-based disturbance observer," *IEEE/ASME Transactions on Mechatronics*, 2020. Doi: 10.1109/TMECH.2020.2999340.
- [8] S. Bouabdallah and R. Siegwart, "Backstepping and sliding-mode techniques applied to an indoor micro quadrotor," in *Proceedings of the 2005 IEEE international conference on robotics and automation*. IEEE, 2005, pp. 2247–2252.
- [9] Pipatpaibul, Pong-in and Ouyang, PR, "Application of online iterative learning tracking control for quadrotor UAVs," *ISRN robotics*, vol. 2013, 2013.
- [10] M. Zhaowei, H. Tianjiang, S. Lincheng, K. Weiwei, Z. Boxin, and Y. Kaidi, "An iterative learning controller for quadrotor UAV path following at a constant altitude," in 2015 34th Chinese Control Conference (CCC). IEEE, 2015, pp. 4406–4411.
- [11] D. A. Bristow, M. Tharayil, and A. G. Alleyne, "A survey of iterative learning control," *IEEE control systems magazine*, vol. 26, no. 3, pp. 96–114, 2006.
- [12] O. Purwin and R. D'Andrea, "Performing aggressive maneuvers using iterative learning control," in 2009 IEEE international conference on robotics and automation. IEEE, 2009, pp. 1731–1736.
- [13] K. Barton and D. Kingston, "Systematic surveillance for UAVs: A feedforward iterative learning control approach," in 2013 American Control Conference. IEEE, 2013, pp. 5917–5922.
- [14] X.-D. Li, T.-F. Xiao, and H.-X. Zheng, "Adaptive discrete-time iterative learning control for non-linear multiple input multiple output systems with iteration-varying initial error and reference trajectory," *IET control theory & applications*, vol. 5, no. 9, pp. 1131–1139, 2011.
- [15] D. Meng and K. L. Moore, "Robust iterative learning control for nonrepetitive uncertain systems," *IEEE Transactions on Automatic Control*, vol. 62, no. 2, pp. 907–913, 2016.

- [16] D. Shen and X. Li, Iterative Learning Control for Systems with Iteration-varying Trial Lengths: Synthesis and Analysis. Springer, 2019.
- [17] M. Zheng, Z. Chen, and X. Liang, "A preliminary study on a physical model oriented learning algorithm with application to uavs," in *Dynamic Systems and Control Conference*. ASME, 2019, p. V003T17A009.
- [18] D. J. Hoelzle, A. G. Alleyne, and A. J. W. Johnson, "Basis task approach to iterative learning control with applications to microrobotic deposition," *IEEE Transactions on Control Systems Technolo*gy, vol. 19, no. 5, pp. 1138–1148, 2010.
- [19] M. Hamer, M. Waibel, and R. D'Andrea, "Knowledge transfer for high-performance quadrocopter maneuvers," in 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, 2013, pp. 1714–1719.
- [20] M. Arif, T. Ishihara, and H. Inooka, "Incorporation of experience in iterative learning controllers using locally weighted learning," *Automatica*, vol. 37, no. 6, pp. 881–888, 2001.
- [21] P. Janssens, G. Pipeleers, and J. Swevers, "Initialization of ILC based on a previously learned trajectory," in 2012 American Control Conference (ACC). IEEE, 2012, pp. 610–614.
- [22] Z. T. Dydek, A. M. Annaswamy, and E. Lavretsky, "Adaptive control of quadrotor UAVs: A design trade study with flight evaluations," *IEEE Transactions on control systems technology*, vol. 21, no. 4, pp. 1400– 1406, 2012.
- [23] B. Whitehead and S. Bieniawski, "Model reference adaptive control of a quadrotor uav," in AIAA Guidance, Navigation, and Control Conference, 2010, p. 8148.
- [24] Y. Jung, S. Cho, and D. H. Shim, "A trajectory-tracking controller design using 1 1 adaptive control for multi-rotor uavs," in 2015 International Conference on Unmanned Aircraft Systems (ICUAS). IEEE, 2015, pp. 132–138.
- [25] K. Pereida, D. Kooijman, R. R. Duivenvoorden, and A. P. Schoellig, "Transfer learning for high-precision trajectory tracking through adaptive feedback and iterative learning," *International Journal of Adaptive Control and Signal Processing*, vol. 33, no. 2, pp. 388–409, 2019.
- [26] G. Chowdhary, T. Wu, M. Cutler, and J. P. How, "Rapid transfer of controllers between UAVs using learning-based adaptive control," in 2013 IEEE International Conference on Robotics and Automation. IEEE, 2013, pp. 5409–5416.
- [27] H. Bou-Ammar, H. Voos, and W. Ertel, "Controller design for quadrotor UAVs using reinforcement learning," in 2010 IEEE International Conference on Control Applications. IEEE, 2010, pp. 2130–2135.
- [28] J. Hwangbo, I. Sa, R. Siegwart, and M. Hutter, "Control of a quadrotor with reinforcement learning," *IEEE Robotics and Automation Letters*, vol. 2, no. 4, pp. 2096–2103, 2017.
- [29] S. Bansal, A. K. Akametalu, F. J. Jiang, F. Laine, and C. J. Tomlin, "Learning quadrotor dynamics using neural network for flight control," in 2016 IEEE 55th Conference on Decision and Control (CDC). IEEE, 2016, pp. 4653–4660.
- [30] Q. Li, J. Qian, Z. Zhu, X. Bao, M. K. Helwa, and A. P. Schoellig, "Deep neural networks for improved, impromptu trajectory tracking of quadrotors," in 2017 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2017, pp. 5183–5189.
- [31] C. Devin, A. Gupta, T. Darrell, P. Abbeel, and S. Levine, "Learning modular neural network policies for multi-task and multi-robot transfer," in 2017 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2017, pp. 2169–2176.
- [32] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on.* IEEE, 2011, pp. 2520–2525.
 [33] X. Liang, M. Zheng, and F. Zhang, "A Scalable Model-Based Learning
- [33] X. Liang, M. Zheng, and F. Zhang, "A Scalable Model-Based Learning Algorithm with Application to UAVs," *IEEE Control Systems Letters*, 2018.
- [34] F. Ding, X. Wang, Q. Chen, and Y. Xiao, "Recursive least squares parameter estimation for a class of output nonlinear systems based on the model decomposition," *Circuits, Systems, and Signal Processing*, vol. 35, no. 9, pp. 3323–3338, 2016.
- [35] M. Verhaegen and V. Verdult, Filtering and system identification: a least squares approach. Cambridge university press, 2007.
- [36] R. Adlakha and M. Zheng, "An optimization-based iterative learning control design method for UAVs trajectory tracking," in 2020 American Control Conference (ACC), 2020.