arXiv:2404.10231v1 [cs.RO] 16 Apr 2024

Improving Disturbance Estimation and Suppression via Learning among Systems with Mismatched Dynamics

Harsh Modi[1], Zhu Chen[2], Xiao Liang[3,*], and Minghui Zheng[1,*]

***Abstract*— Iterative learning control (ILC) is a method for reducing system tracking or estimation errors over multiple iterations by using information from past iterations. The disturbance observer (DOB) is used to estimate and mitigate disturbances within the system, while the system is being affected by them. ILC enhances system performance by introducing a feedforward signal in each iteration. However, its effectiveness may diminish if the conditions change during the iterations. On the other hand, although DOB effectively mitigates the effects of new disturbances, it cannot entirely eliminate them as it operates reactively. Therefore, neither ILC nor DOB alone can ensure sufficient robustness in challenging scenarios. This study focuses on the simultaneous utilization of ILC and DOB to enhance system robustness. The proposed methodology specifically targets dynamically different linearized systems performing repetitive tasks. The systems share similar forms but differ in dynamics (e.g. sizes, masses, and controllers). Consequently, the design of learning filters must account for these differences in dynamics. To validate the approach, the study establishes a theoretical framework for designing learning filters in conjunction with DOB. The validity of the framework is then confirmed through numerical studies and experimental tests conducted on unmanned aerial vehicles (UAVs). Although UAVs are nonlinear systems, the study employs a linearized controller as they operate in proximity to the hover condition. A video introduction of this paper is available via this link.**

I. Introduction

Deploying safety-critical robotic systems such as unmanned aerial vehicles (UAVs) in the vicinity of human presence requires them to possess robustness against external factors, such as wind disturbances. These disturbances can significantly impact the trajectory of the UAVs, posing potential dangers to external subjects [1] [2]. Therefore, it is crucial for these robotic systems to estimate and mitigate the effects of disturbances to ensure the necessary level of safety. Also, various tasks may require the use of dynamically different systems in disturbance-prone environments.

©2024 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

[1]Harsh Modi (harsh.modi@tamu.edu, current Ph.D. student) and Minghui Zheng (mhzheng@tamu.edu, faculty member) are with the Department of Mechanical Engineering, Texas A&M University, TX 77843.

Zhu Chen (zhuchen@buffalo.edu, former Ph.D. student) was with Mechanical and Aerospace Engineering, University at Buffalo, NY 14228

[3]Xiao Liang (xliang@tamu.edu, faculty member) is with the Department of Civil & Environmental Engineering, Texas A&M University, TX 77843.

This work is supported by U.S. National Science Foundation (Grants: No. 2030375/2422579 and No. 2046481/2422698). Correspondence to Minghui Zheng and Xiao Liang.

Iterative learning control (ILC) is effective in reducing system error over multiple iterations in repetitive tasks, thereby enhancing performance in each iteration. It has been successfully applied in various applications, including manipulator-based robotics systems [3]. Recently, ILC has been utilized to improve UAV's trajectory tracking performance [4] using optimization-based filter designs. Researchers in [5] implemented parameter determination-based ILC for robotic manipulators, while [6] combined ILC with sliding mode control to enhance the trajectory tracking for UAVs.

The disturbance observer (DOB) has been widely used to enhance the robustness of the controller against external disturbances. The article [7] provides an overview of advancements in DOB from 1985 to 2020. DOB-based controllers have been employed to compensate for unknown disturbances in small UAV systems [8]. In [9], the authors used DOB with a disturbance rejection signal in the form of acceleration, which is similar to the force exerted on the UAV due to disturbances like wind. [10] used finite-time disturbance observer to mitigate disturbance effects for quadrotor UAVs, and [11] developed a linear dual disturbance observer to improve UAV trajectory tracking. DOB has also been used to enhance the robustness of fixed-wing UAVs [12], and in [13], a disturbance observer was designed for nonlinear and nonautonomous systems.

The objective of this research is to combine the advantages of both the ILC and DOB to improve system robustness. Specifically, this study focuses on increasing the robustness of UAV trajectory tracking against external disturbances, while simultaneously estimating the disturbance present in the environment. When UAVs follow the same trajectories within a relatively short period, it can be assumed that the disturbances will not vary significantly. In such cases, the benefits of ILC can be leveraged to proactively compensate for repetitive errors in trajectory tracking caused by disturbance or controller limitations. However, ILC alone cannot account for changing conditions, which can be addressed by incorporating DOB. By combining ILC and DOB, we can utilize the proactivity of ILC and the ability of DOB to adapt to new disturbances.

Many studies have explored this direction. [14] employed ILC along with DOB to account for non-repetitiveness in the disturbances. In [15], the performance of ILC was enhanced with DOB for wafer scanning systems. [16] utilized the combined ILC and DOB to reject near-repetitive disturbances in excavation operations. [17] combined ILC with DOB to improve the robustness of machine tool feed drives. [18] improved the closed-loop performance using ILC based on

DOB. [19] employed ILC with DOB for rehabilitation. In [20], ILC was combined with disturbance estimation for unmatched model uncertainties and matched disturbances. However, all these studies utilized the same system in each iteration, limiting the robustness to a single system.

Some research has focused on implementing iterative learning in dynamically different systems. [21] extended the capability of ILC in heterogeneous systems with different initial conditions. [22] used a transfer learning approach to transfer input learning to a different system, but it was specifically targeted at improving trajectory tracking and did not include disturbance estimation. This study is an extension of [23], where we combine ILC with DOB for dynamically different systems. In [23], the authors designed learning filters for dynamically different systems, allowing each subsequent system to learn from the errors and learning signal of the previous system. The design was based on guaranteeing that the learning-based trajectory tracking error of each system would be smaller than the system without learning. However, explicit disturbance rejection was not included, which is a focus of the present study, along with the learning.

The main contributions of this study are as follows: we explicitly incorporate DOB as part of the ILC update process and consider differences in system dynamics among different systems to enable learning. To the best of our knowledge, this is the first attempt to implement learning with DOB for systems with mismatched dynamics. Furthermore, the designed methodology has undergone rigorous verification and validation through simulations and experiments. In the current study, the disturbance rejection and learning framework is implemented as follows: (1) All systems operate with an underlying PID baseline controller that remains unmodified in this study. As the systems operate near hover conditions, the linearization approximation holds. (2) All systems utilize a DOB algorithm in conjunction with the PID controller to estimate and reject the disturbances. However, DOB alone cannot fully compensate for disturbance effects. Therefore, information regarding the tracking error is passed to the next system. (3) The next system utilizes this information to generate a learning signal, which aids in improved disturbance estimation and rejection compared to non-learning scenarios.

The rest of the paper is organized as follows: Section II establishes the theoretical framework and describes the design of the learning filters. Section III presents the simulation and the experimental results. Section IV concludes the article. Please note that throughout the paper, the term "UAV" indicates the particular hardware used in the experiment while the term "system" is used to describe the general order of the learning iterations.

II. Learning Framework

A. Variable Definition and Standard DOB Basics

We first introduce notations that will be used in our framework. We denote the signals as follows: $r(k)$ as the reference input, $y(k)$ as the output, $e(k)$ as the tracking error,

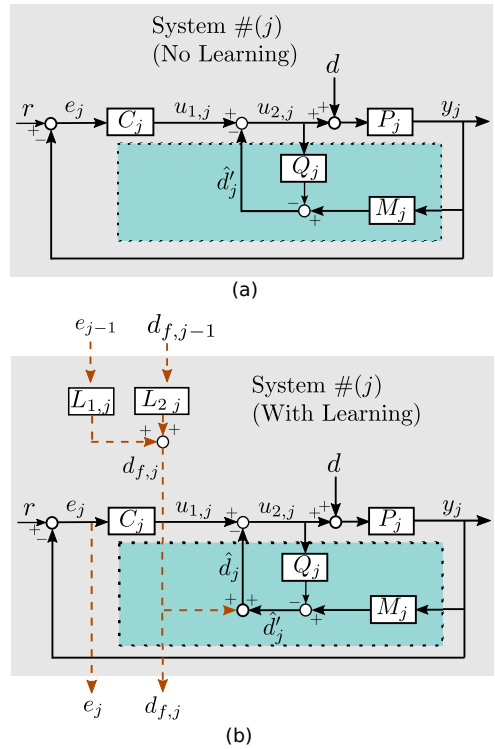

Fig. 1. System block diagram with (a) a basic DOB framework (b) an iterative learning framework along with DOB

$d(k)$ and $\hat{d}(k)$ as the disturbance and its estimate, $u_1(k)$ as the control signal directly generated by the baseline feedback control, $u_2(k)$ as the modified control signal that is sent to the plant. All these signals are time series. We also introduce the following notations for different subsystems: $P(z)$ as the plant, $C(z)$ as the baseline controller, $Q(z)$ as a low-pass filter, $M(z)$ as the plant inverse, $L_1(z)$ and $L_2(z)$ as learning filters; all of these are transfer functions in discrete time. In addition, we use j to index different systems; and use the prime symbol ($'$) to distinguish signals in systems without learning. For ease of reading, we will omit k in signals and z in transfer functions.

We now introduce the standard DOB and explain how it works. As shown in Fig. 1(a), the DOB is added to System j. It consists of a plant inverse M and a low-pass filter Q, as highlighted by the dotted box. When a disturbance d is present, the DOB can provide a disturbance estimate $\hat{d}$ which will be subtracted from u_1 to cancel. Ideally, if a plant inverse can be accurately obtained, and the intrinsic delay in P is small, $\hat{d}$ would be close to d so that the disturbance can be suppressed. However, it is difficult to accurately estimate the plant inverse and the delays exist in dynamic systems. These limitations of the DOB can be addressed by using a learning approach across multiple iterations, as proposed in this study. The subsequent section elaborates on the development of this learning framework over DOB for dynamically different systems.

B. Iterative Learning with DOB Framework

In this subsection, we will introduce the iterative learning framework with DOB. Fig. 1(b) shows the detailed system block diagram with the learning framework introduced,

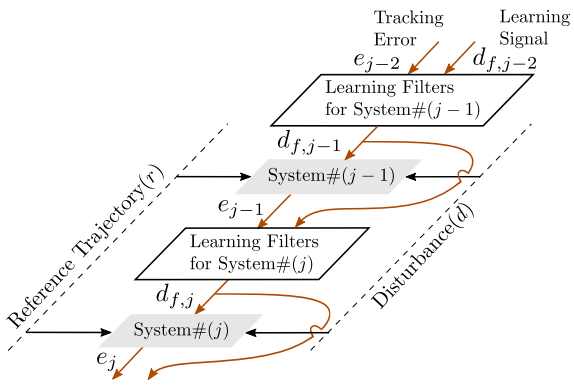

Fig. 2. Learning relationship among systems

whereas Fig. 2 shows the overall learning relationship between two systems having different dynamics, i.e. System #$(j-1)$ and System #(j). The System #$(j-1)$'s learning signal ($d_{f,j-1}$) and its trajectory tracking error (e_{j-1}) can qualify the accuracy of its disturbance estimation. Hence, $d_{f,j}$ is generated using e_{j-1} and $d_{f,j-1}$ and to-be-deisgned learning filters for System #(j). We propose two different learning filters $L_{1,j}$ and $L_{2,j}$ respectively for e_{j-1} and $d_{f,j-1}$ as illustrated in Fig. 1(b). As the framework aims to improve the system's disturbance estimation and suppression capability as well as its trajectory-tracking performance, the learning signal is added to the disturbance estimate ($\hat{d}'_j$) from the DOB. It is important to note that learning aims to improve the performance of the system with learning compared to its performance without learning. We do not compare the performance of the system #(j) with system #$(j-1)$ as each system's dynamics can affect individual performance.

The derivation is based on the following assumptions: 1. We consider that all the systems are linear time-invariant (LTI) as a large number of non-linear time-invariant systems can be linearized near the equilibrium point. 2. As the research is aimed at reducing the effects of disturbances on repetitive tasks (such as industrial assembly or delivery robots), we consider they follow the same trajectories and are subject to similar disturbances 3. The current research aims at slowly time-varying or stationary reference signals.

In the following paragraphs, we will establish the relationship between the tracking error with learning (e_j) and the tracking error without learning (e'_j) after stating some system parameters' definitions.

System Parameters: Based on the system block diagram in Fig. 1(b), $G_{r,j}$ (dynamics from reference signal r to output y_j), $G_{d,j}$ (dynamics from disturbance d to output y_j), and $G_{f,j}$ (dynamics from learning signal $d_{f,j}$ to output y_j) can be described by:

$$G_{r,j} = [1 - Q_j + P_j(M_j + C_j)]^{-1} P_j C_j \tag{1}$$

$$G_{d,j} = [1 - Q_j + P_j(M_j + C_j)]^{-1} P_j (1 - Q_j) \tag{2}$$

$$G_{f,j} = [1 - Q_j + P_j(M_j + C_j)]^{-1} (-P_j) \tag{3}$$

Establishing Relationship between e_j and e'_j: Using the system parameters defined above, the output of system #(j) with learning is given by:

$$y_j = G_{r,j}\{r\} + G_{d,j}\{d\} + G_{f,j}\{d_{f,j}\} \tag{4}$$

and the output of the system #(j) without learning is given by:

$$y'_j = G_{r,j}\{r\} + G_{d,j}\{d\} \tag{5}$$

where the notation $\{\}$ indicates that the signal inside is sent to a system which can be represented by the outside transfer function.

In order to establish the relationship between e_j and e'_j, let us expand $e_j - e'_j$:

$$e_j - e'_j = (r - y_j) - (r - y'_j) = y'_j - y_j \tag{6}$$

Now, using Eq. (4) and Eq. (5):

$$e_j = e'_j + y'_j - y_j = e'_j - G_{f,j}\{d_{f,j}\} \tag{7}$$

As discussed earlier, the learning signal of the current system ($d_{f,j}$) is based on e_{j-1}, and $d_{f,j-1}$ and the respective to-be-designed learning filters $L_{1,j}$, and $L_{2,j}$. Hence:

$$d_{f,j} = L_{1,j}\{e_{j-1}\} + L_{2,j}\{d_{f,j-1}\} \tag{8}$$

Using this, we expand Eq. (7) further as:

$$e_j = e'_j - G_{f,j}(L_{1,j}\{e_{j-1}\} + L_{2,j}\{d_{f,j-1}\}) \tag{9}$$

Now, e_{j-1} can be expressed using Eq. (4) as:

$$\begin{aligned} e_{j-1} &= r - y_{j-1} \\ &= (1 - G_{r,j-1})\{r\} - G_{d,j-1}\{d\} - G_{f,j-1}\{d_{f,j-1}\} \end{aligned} \tag{10}$$

Using this in Eq. (9) and after some simplification:

$$\begin{aligned} e_j =& e'_j - G_{f,j}L_{1,j}(1 - G_{r,j-1})\{r\} + G_{f,j}L_{1,j}G_{d,j-1}\{d\} \\ &+ G_{f.j}(L_{1,j}G_{f,j-1} - L_{2,j})\{d_{f,j-1}\} \end{aligned} \tag{11}$$

Eq. (11) contains variables r, d, and $d_{f,j-1}$ apart from e'_j. In order to effectively establish a relationship between e_j and e'_j, let us try to reduce the number of variables in the equation. Using Eq. (5), we can express e'_j as:

$$e'_j = r - y'_j = (1 - G_{r,j})\{r\} - G_{d,j}\{d\} \tag{12}$$

Hence, d can be expressed in terms of r and e'_j as:

$$d = G_{d,j}^{-1}(r - G_{r,j}\{r\} - e'_j) \tag{13}$$

substituting this in Eq. (11) and with some re-arrangements, we get

$$\begin{aligned} e_j =& (1 - G_{f,j}L_{1,j}G_{d,j-1}G_{d,j}^{-1})\{e'_j\} \\ &- G_{f,j}L_{1,j}((1 - G_{r,j-1}) - G_{d,j-1}G_{d,j}^{-1}(1 - G_{r,j}))\{r\} \\ &+ G_{f,j}(L_{1,j}G_{f,j-1} - L_{2,j})\{d_{f,j-1}\} \end{aligned} \tag{14}$$

Considering that the trajectory tracking controller C is well designed such that

$$G_r(j\omega) = \frac{C(j\omega)P(j\omega)}{1 + C(j\omega)P(j\omega)} \approx 1 \tag{15}$$

for $\omega < \omega_0$, where ω_0 is the desired unity-gain bandwidth of the closed-loop system G_r.

As seen in a later section, for the UAVs used in this study, the gain of the transfer function G_r is close to 0 dB and the phase is close to 0° for $\omega < 1\ rad/s$. Considering Eq. (15) and with the presence of $(1-G_{r,j-1})$ and $(1-G_{r,j})$, the following term

$$-G_{f,j}L_{1,j}((1-G_{r,j-1})-G_{d,j-1}G_{d,j}^{-1}(1-G_{r,j}))\{r\} \approx 0 \quad (16)$$

for stationary or slowly time-varying trajectories.

With Eq. (16), we can re-write Eq. (14) as:

$$\begin{aligned} e_j =& (1-G_{f,j}L_{1,j}G_{d,j-1}G_{d,j}^{-1})\{e'_j\}+ \\ & G_{f,j}(L_{1,j}G_{f,j-1}-L_{2,j})\{d_{f,j-1}\} \end{aligned} \quad (17)$$

This relationship between e_j and e'_j will be used in designing the learning filters.

C. Learning Filter Design

In this subsection, we will introduce the design of the learning filters, i.e., $L_{1,j}$ and $L_{2,j}$, such that e_j can be reduced compared to e'_j. For simplification, we introduce the following notations:

$$\begin{aligned} T_{e_{1,j}} =& (1-G_{f,j}L_{1,j}G_{d,j-1}G_{d,j}^{-1}) \\ T_{e_{2,j}} =& G_{f,j}(L_{1,j}G_{f,j-1}-L_{2,j}) \end{aligned} \quad (18)$$

Using this, Eq. (17) can be re-written as:

$$e_j = T_{e_{1,j}}\{e'_j\} + T_{e_{2,j}}\{d_{f,j-1}\} \quad (19)$$

To explicitly analyze robustness, we denote the modeling uncertainty as Δ_j and the relationship between the actual plant (P_j) and the identified plant model ($\hat{P}_j$) as:

$$P_j = (1+\Delta_j)\hat{P}_j \quad (20)$$

In the following paragraphs, we will introduce the learning filter design in a theorem-proof format. We also split our proof into two cases: without and with modeling uncertainties. We use $\|\cdot\|$ to denote the 2-norm of a signal and the H-infinity norm of a system. That is,

$$\|s\| = (\sum_0^\infty [s(k)]^2)^{1/2} \text{ and } \|G\| = \max_\omega |G(j\omega)| \quad (21)$$

where s is a discrete-time signal and G is a LTI transfer function.

Theorem: The following learning filters

$$L_{1,j} = (\hat{G}_{f,j}\hat{G}_{d,j-1})^{-1}\hat{G}_{d,j} \quad (22)$$

and

$$L_{2,j} = (\hat{G}_{f,j}\hat{G}_{d,j-1})^{-1}\hat{G}_{d,j}\hat{G}_{f,j-1} \quad (23)$$

can ensure $||e_j|| < ||e'_j||$ if

$$||\Delta_{j-1}|| < ||(1+P_{j-1}C_{j-1})/2|| \quad (24)$$

Proof: With the learning filters in Eq. (22) and Eq. (23), Eq. (18) can be expanded as:

$$\begin{aligned} T_{e_{1,j}} =& (1-G_{f,j}[\hat{G}_{f,j}\hat{G}_{d,j-1})^{-1}\hat{G}_{d,j}]G_{d,j-1}G_{d,j}^{-1}) \\ T_{e_{2,j}} =& G_{f,j}([\hat{G}_{f,j}\hat{G}_{d,j-1})^{-1}\hat{G}_{d,j}]G_{f,j-1} \\ & -[\hat{G}_{f,j}\hat{G}_{d,j-1})^{-1}\hat{G}_{d,j}\hat{G}_{f,j-1}]) \end{aligned} \quad (25)$$

Now, let us analyze Eq. (25) in 2 different cases.

Case I: No modeling uncertainty

When there is no modeling uncertainty, i.e., $\Delta_j = 0$ and $\Delta_{j-1} = 0$, we have $G_{f,j} = \hat{G}_{f,j}$, $G_{d,j} = \hat{G}_{d,j}$, $G_{f,j-1} = \hat{G}_{f,j-1}$, and $G_{d,j-1} = \hat{G}_{d,j-1}$. Hence, Eq. (25) reduces to

$$\begin{aligned} T_{e_{1,j}} &= 0 \\ T_{e_{2,j}} &= 0 \end{aligned} \quad (26)$$

Therefore, with the derived learning filters, e_j will converge to zero as per Eq. (19) when there is no modeling uncertainty.

Case II: With some modeling uncertainty:

When there is some modelling uncertainty present, i.e. $||\Delta_j|| \neq 0$ or $||\Delta_{j-1}|| \neq 0$, we can simplify Eq. (25) using Eq. (2), Eq. (3), and Eq. (20) as:

$$T_{e_{1,j}} = -\frac{\Delta_{j-1}}{1+P_{j-1}C_{j-1}} \quad (27)$$

$$T_{e_{2,j}} = -\frac{\Delta_{j-1}}{1+P_{j-1}C_{j-1}}G_{d,j-1}^{-1}G_{d,j}G_{f,j-1} \quad (28)$$

Using Eq. (12), Eq. (19), Eq. (27), Eq. (28), and with the approximation $||\Delta_j\Delta_{j-1}|| \approx 0$ for Δ_j and Δ_{j-1} of small gains, the error "with learning" for System #(j) can be approximated as:

$$e_j \approx \frac{2\cdot\Delta_{j-1}Gd_j}{1+C_{j-1}P_{j-1}}\{d\} \approx \frac{-2\cdot\Delta_{j-1}}{1+C_{j-1}P_{j-1}}\{e'_j\} \text{ (for } j \geq 2) \quad (29)$$

As per Eq. (29), to achieve $||e_j|| < ||e'_j||$, we need

$$\left\| \frac{-2\cdot\Delta_{j-1}}{1+C_{j-1}P_{j-1}} \right\| < 1 \quad (30)$$

As $1/(1+C_{j-1}P_{j-1})$ is the transfer function from r to e_{j-1}, it will have a bounded gain for a stable closed-loop system. Hence, as per Eq. (30), for $||\Delta_{j-1}|| < ||(1+P_{j-1}C_{j-1})/2||$, the $||e_j||$ is guaranteed to be less than $||e'_j||$. This result also confirms that if the system parameters are accurate (i.e. $\Delta_j = 0$ for $\forall j$), then the error with learning will be 0 as proven earlier in case I. This is the end of the theorem proof.

It is noted that the learning filters are explicitly dependent on model inverse; when the systems are not minimum-phase, we need to find their stable inverse approximation in the learning filter design [24].

III. Simulation and Experimental Validation

In order to assess the effectiveness of the learning framework, the simulations and experiments have been performed on three dynamically different quadrotor UAVs as described in Table I. The quadrotor UAVs are inherently non-linear

TABLE I

Specifications of the systems used

	UAV#(1)	**UAV#(2)**	**UAV#(3)**
Frame Brand	F450	S500	Tarot 650
Mass	0.921 kg	1.001 kg	1.234 kg
Motor Distance from Center of Mass	228 mm	240 mm	318 mm
Proportional Gain of PID controller	3.0	1.2	1.9

systems but are operating near their hover conditions using the linearized controller and hence we can assume that the linear system approximation holds here. It is to be noted that the proportional gains of the outer-loop PID controllers are kept different for each UAV to further vary their dynamics from each other.

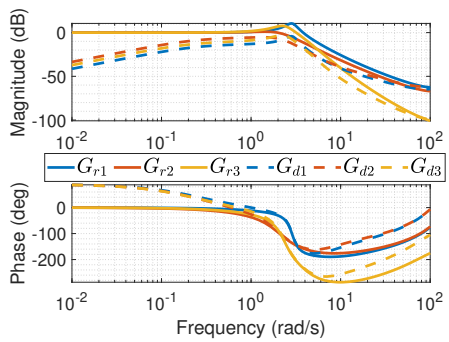

Fig. 3. Bode plots for $G_{r,j}$ and $G_{d,j}$

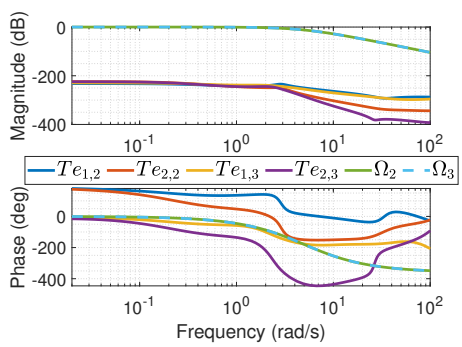

Fig. 4. Bode plots for $Te_{1,j}$, $Te_{2,j}$, and Ω_j

Using the identified transfer functions, we plot the Bode plots for the response of the systems given the reference ($G_{r,j}$) and the response of the systems given the disturbance ($G_{d,j}$) as shown in Fig. 3. The $G_{d,j}$ for lower frequencies and very high frequencies have very low gain, indicating UAVs' inherent disturbance rejection capabilities in those frequencies. The middle frequencies (0.1 rad/s to 5 rad/s) are where the disturbances can have the maximum effects and hence we analyze the performance of the learning framework for the disturbances within this range.

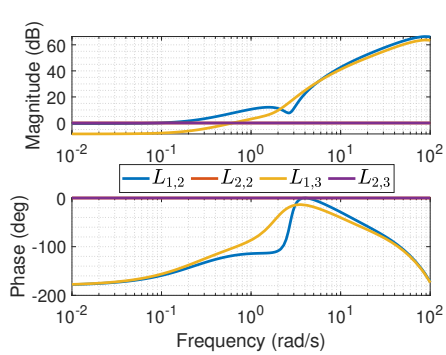

Fig. 5. Bode plots for $L_{1,j}$ and $L_{2,j}$

Fig. 5 shows the bode plots of designed learning filters for UAV#(2) learning from UAV#(1) and UAV#(3) learning from UAV#(2). Fig. 4 shows the bode plots for $Te_{1,j}$, and $Te_{2,j}$ (Eq. (19)). The bode plots of Ω_j are also shown in the same figure, where

$$\hat{d}'_j = \Omega_j\{d\} \tag{31}$$

and Ω_j can be given by

$$\Omega_j = [1 - Q_j + P_j(M_j + C_j)]^{-1}(M_j + Q_jC_j)P_j \tag{32}$$

A. Simulation and Experimental Procedure

The simulation for all the validations is done in MATLAB using the transfer functions derived from system identification experiments. The Eq. (4) and Eq. (5) are used to simulate the trajectories followed by the UAVs for "without learning" and "with learning" cases respectively. Also, as the system parameters for the simulations are accurate, they reflect the ideal case presented in the proof, while the experiments reflect the performance of the learning framework with some deviation from the ideal scenario. For experiments, the UAVs are equipped with a Pixhawk flight controller responsible for controlling the UAV's attitude. Additionally, a Raspberry Pi serves as a companion computer to the Pixhawk. The position of the UAVs is tracked using Vicon motion capture cameras, and the position data is transmitted to the Raspberry Pi. The Raspberry Pi runs a trajectory (outerloop) controller that generates desired attitude and thrust inputs for the Pixhawk's attitude controller. Both the trajectory controller and the attitude controller utilize PID control algorithms. The gains of the PID controllers are not a concern for this study, as long as the system remains stable. The basic PID controller for trajectory control is based on the work presented in [25]. Furthermore, all systems employ the DOB algorithm as part of the trajectory controller. To keep the validation and description neat, the disturbance is added to the x-direction only and thus the learning is just for the x-direction. To introduce the disturbance into the system, the corresponding acceleration is virtually added to the control signal just before it reaches the plant. Importantly, the disturbance remains unknown to the controller.

For learning on hardware, the error data and the learning signal from a previous UAV are passed through their

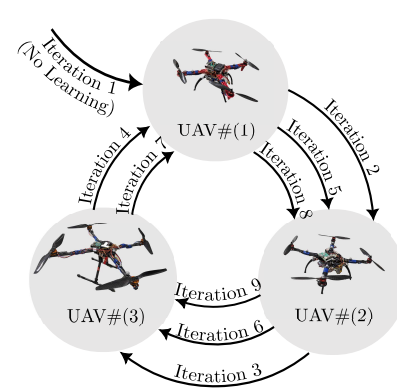

Fig. 6. Learning flow for simulations and experiments

respective learning filters and a low-pass filter. This process generates the learning signal for the next UAV. The resulting learning signal is then stored in the next UAV and used by the trajectory controller. Alternatively, the learning signal can also be generated onboard in the next UAV by performing a simple time series conversion. Also, for both the simulations and experiments, the learning is done in a cyclic way (i.e. after system 3, system 1 is tried again with the learning data from system 3) until a convergence is achieved as shown in Fig. 6. We also perform 2 additional experiments for each UAV for data comparison purposes: (1) without learning and without DOB and (2) without learning but with DOB.

B. Validation Scenarios

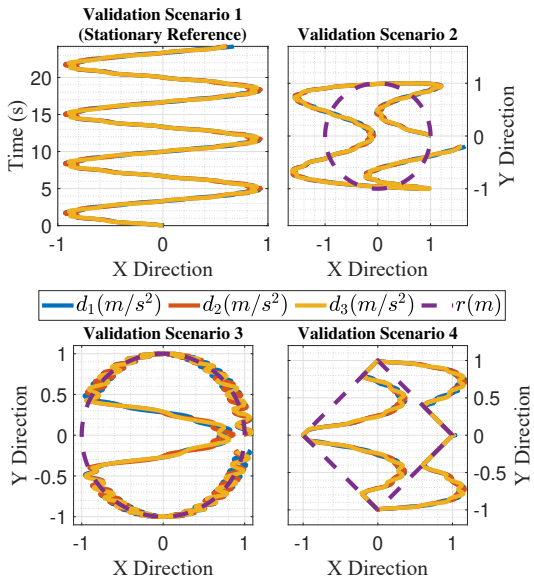

Fig. 7. Reference trajectories (dashed lines) and disturbance profiles (solid lines)

With the simulation and experimental procedure described in Section III-A, we assess the effectiveness of the proposed learning methodology with various reference trajectories and disturbance scenarios. We consider 4 distinct reference trajectory-disturbance combinations with increasing complexities as shown in Fig. 7. The reference trajectory is a stationary point (x: 0 m, y: 0 m, z: 2.5 m) in scenario 1, while it is a circle of radius 1 m with a constant altitude of 2.5 m in scenarios 2 and 3, and a diamond shape of 2 m diagonal length in scenario 4.

In scenarios 1 and 2, the sinusoidal disturbance of 0.9425 rad/s and amplitude 1 m/s^2 is introduced in the x direction. In scenario 3, a half sinusoidal impulse disturbance of the same frequency, but with a larger magnitude of 2 m/s^2 is introduced. In scenario 4, a sinusoidal disturbance of 1.4138 rad/s frequency and 1 m/s^2 amplitude, but rectified only above 0 is introduced. For each scenario, a sinusoidal noise of larger frequency but smaller amplitude is added to each system differently. This replicates the real-life conditions, in which the disturbance is not exactly the same among the systems.

Consider the frequencies used for the validations with the bode gain of $G_{d,j}$ in Fig. 3. For all 3 UAVs, the frequency of 0.9425 rad/s used in scenarios 1 through 3 has a high $G_{d,j}$ gain. This gain is almost maximum at 1.4138 rad/s, a disturbance frequency used in scenario 4. These disturbance frequencies for validations are chosen considering this gain as the UAVs' trajectories are most affected by the disturbances at these frequencies, which the learning framework is designed to mitigate. The large magnitude impulse disturbance introduced for scenario 3 replicates a sudden gust of the wind on the UAVs. In scenario 4, the sinusoidal disturbance clipped only in a positive direction replicates the intermittent wind between the buildings. Also, scenario 4 is coupled with sharp corners reference trajectory, which is difficult to track for dynamic systems like UAVs. The UAVs are flown in a different order in scenario 4 (UAV#(1) $\implies$ UAV#(3) $\implies$ UAV#(2)) to test the adaptability of the learning framework. In summary, these scenarios are designed to validate the proposed learning framework rigorously, proving the robustness of the same.

C. Results

In this section, we present detailed results from the validation simulations and experiments. The video from the experiments is available at Link. Fig. 8 shows the Root Mean Square Error (RMSE) of the trajectory tracking for each scenario in both simulations and experiments. The RMSE of the trajectory tracking reflects the summary of each experiment so that we can quantitatively compare them. For each scenario and each system, we compare the errors in "No DOB" cases (blue bars), "Only DOB" cases without learning (orange bars), and the errors in 1st, 2nd, and 3rd learning iterations of each system. Please note that we are comparing the results between various experiments within the systems and not with other systems as differences in system dynamics affect the performance. For all the validation scenarios, the trajectory tracking error without learning is maximum, which reduces marginally with DOB. However, the errors with learning reduce significantly for each system. For simulations, the errors with learning converge in the first learning iteration itself as the system models are accurate. For experiments, we can observe that in the 2nd and the 3rd learning iterations, the errors are very close to 0 for each system.

Expanding further into the summary of Fig. 8, Fig. 9 shows the trajectory tracking errors in the experiments for all the validation scenarios against time. Each row of plots refers to a particular system whereas each column of plots refers to a validation scenario. Here also, it is evident that with DOB but without learning, the errors are marginally reduced compared to "without DOB" cases. With learning, the errors reduce significantly compared to the "only DOB" cases, which become almost negligible in the 2nd and the 3rd iterations.

The learning framework also improves the disturbance estimates as shown in Fig. 10. The shaded shapes on each plot reflect the actual disturbances while the lines show the

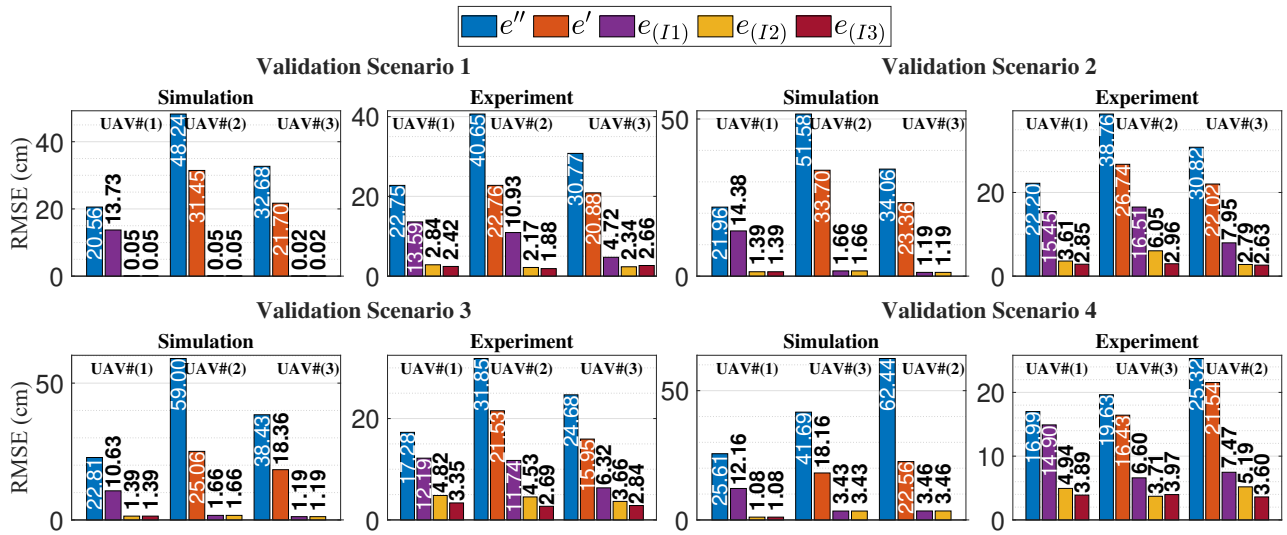

Fig. 8. RMSE of the trajectory tracking for all the scenarios in both the simulations and experiments. e'' indicates the "no DOB" case, e' indicates the "only DOB" case and $e_{(Ik)}$ indicates the kth iteration for each system.

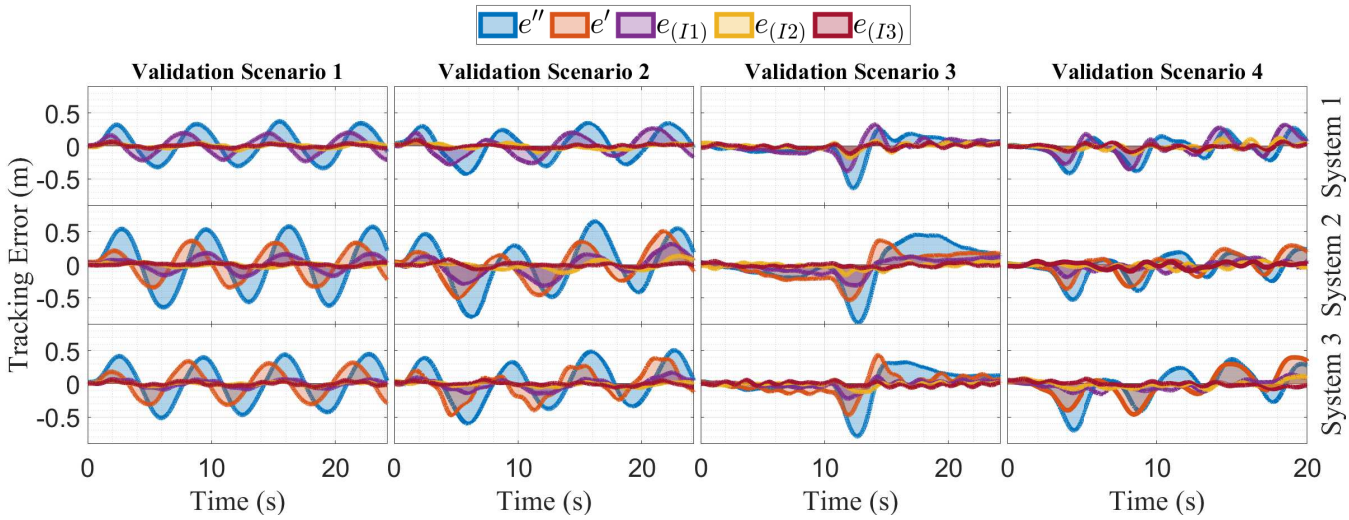

Fig. 9. Trajectory tracking errors in experiments for all 4 Scenarios. e'' indicates the "no DOB" case, e' indicates the "only DOB" case and $e_{(Ik)}$ indicates the kth learning iteration for each system.

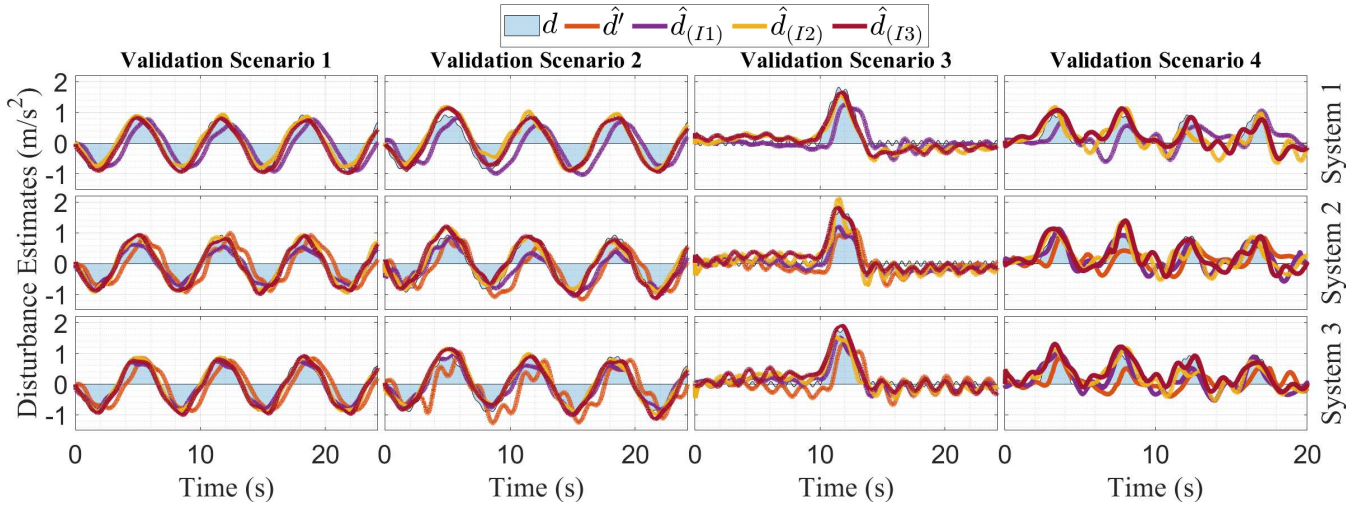

Fig. 10. Disturbance estimates for experiments in all the scenarios. d' indicates the disturbance estimates in the "only DOB" case and $d_{(Ik)}$ indicates the disturbance estimates with learning in k^{th} iteration for each system.

estimated disturbances. We can observe that in the "only DOB" cases, the disturbance estimates in scenarios 1 and 2 are almost perfect, but are delayed. This is expected as the DOB estimates the performance using the output of the system, which is delayed compared to the disturbances. In scenario 3, the estimates are delayed as well as scaled down in the "only DOB" cases. In scenario 4, the estimates are highly inaccurate in the "only DOB" cases, possibly because the disturbance was clipped in a positive direction. However, with the first iteration of learning, the estimates are improved significantly, which explains why the trajectory tracking performance is better with learning. In 2nd and 3rd learning iterations of each system, the disturbances are nearly fully recovered, which validates the effectiveness of the learning framework.

IV. Conclusions

This study presents a framework for designing ILC with DOB in systems with mismatched dynamics. Both numerical and experimental validations demonstrate the effectiveness of the framework and the learning filters in reducing the tracking error and improving the disturbance estimate under various disturbance scenarios, enhancing the system's robustness to external disturbances. Future research directions include exploring learning with highly aggressive reference trajectories. Additionally, the current approach utilizes errors and learning signals from only the previous system for learning. To further improve the learning framework, future work can explicitly incorporate data from all previous systems into the learning process, rather than relying solely on the implicit information contained in the learning signal of the previous system.

References

[1] Wang, Bo Hang, Dao Bo Wang, Zain Anwar Ali, Bai Ting Ting, and Hao Wang. "An overview of various kinds of wind effects on unmanned aerial vehicle." Measurement and Control 52, no. 7-8 (2019): 731-739.

[2] Howard, John, Vladimir Murashov, and Christine M. Branche. "Unmanned aerial vehicles in construction and worker safety." American journal of industrial medicine 61, no. 1 (2018): 3-10.

[3] H. -S. Ahn, Y. Chen and K. L. Moore, "Iterative Learning Control: Brief Survey and Categorization," in IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), vol. 37, no. 6, pp. 1099-1121, Nov. 2007, doi: 10.1109/TSMCC.2007.905759.

[4] R. Adlakha and M. Zheng, "An Optimization-Based Iterative Learning Control Design Method for UAV's Trajectory Tracking," 2020 American Control Conference (ACC), Denver, CO, USA, 2020, pp. 1353-1359, doi: 10.23919/ACC45564.2020.9147752.

[5] Nguyen, Phuoc Doan, and Nam Hoai Nguyen. "An intelligent parameter determination approach in iterative learning control." European Journal of Control 61 (2021): 91-100.

[6] Nguyen, Lanh Van, Manh Duong Phung, and Quang Phuc Ha. "Iterative Learning Sliding Mode Control for UAV Trajectory Tracking." Electronics 10, no. 20 (2021): 2474.

[7] E. Sariyildiz, R. Oboe and K. Ohnishi, "Disturbance Observer-Based Robust Control and Its Applications: 35th Anniversary Overview," in IEEE Transactions on Industrial Electronics, vol. 67, no. 3, pp. 2042-2053, March 2020, doi: 10.1109/TIE.2019.2903752.

[8] Chen, Jieqing, Ruisheng Sun, and Bin Zhu. "Disturbance observer-based control for small nonlinear UAV systems with transient performance constraint." Aerospace Science and Technology 105 (2020): 106028.

[9] S. J. Lee, S. Kim, K. H. Johansson and H. J. Kim, "Robust acceleration control of a hexarotor UAV with a disturbance observer," 2016 IEEE 55th Conference on Decision and Control (CDC), Las Vegas, NV, USA, 2016, pp. 4166-4171, doi: 10.1109/CDC.2016.7798901.

[10] F. Wang, H. Gao, K. Wang, C. Zhou, Q. Zong and C. Hua, "Disturbance Observer-Based Finite-Time Control Design for a Quadrotor UAV With External Disturbance," in IEEE Transactions on Aerospace and Electronic Systems, vol. 57, no. 2, pp. 834-847, April 2021, doi: 10.1109/TAES.2020.3046087.

[11] Wang H, Chen M. Trajectory tracking control for an indoor quadrotor UAV based on the disturbance observer. Transactions of the Institute of Measurement and Control. 2016;38(6):675-692. doi:10.1177/0142331215597057

[12] Smith, Jean, Jinya Su, Cunjia Liu, and Wen-Hua Chen. "Disturbance observer based control with anti-windup applied to a small fixed wing UAV for disturbance rejection." Journal of Intelligent & Robotic Systems 88 (2017): 329-346.

[13] Nguyen, Phuoc Doan, and Nam Hoai Nguyen. "A simple approach to estimate unmatched disturbances for nonlinear nonautonomous systems." International Journal of Robust and Nonlinear Control 32, no. 17 (2022): 9160-9173.

[14] Yang Quan Chen and K. L. Moore, "Harnessing the nonrepetitiveness in iterative learning control," Proceedings of the 41st IEEE Conference on Decision and Control, 2002., Las Vegas, NV, USA, 2002, pp. 3350-3355 vol.3, doi: 10.1109/CDC.2002.1184392.

[15] S. Yu and M. Tomizuka, "Performance enhancement of iterative learning control system using disturbance observer," 2009 IEEE/ASME International Conference on Advanced Intelligent Mechatronics, Singapore, 2009, pp. 987-992, doi: 10.1109/AIM.2009.5229715.

[16] G. J. Maeda, I. R. Manchester and D. C. Rye, "Combined ILC and Disturbance Observer for the Rejection of Near-Repetitive Disturbances, With Application to Excavation," in IEEE Transactions on Control Systems Technology, vol. 23, no. 5, pp. 1754-1769, Sept. 2015, doi: 10.1109/TCST.2014.2382579.

[17] Simba, Kenneth Renny, Ba Dinh Bui, Mathew Renny Msukwa, and Naoki Uchiyama. "Robust iterative learning contouring controller with disturbance observer for machine tool feed drives." ISA transactions 75 (2018): 207-215.

[18] Sun, Jiankun, and Shihua Li. "Disturbance observer based iterative learning control method for a class of systems subject to mismatched disturbances." Transactions of the Institute of Measurement and Control 39, no. 11 (2017): 1749-1760.

[19] B. Huo, Y. Liu, Y. Qin, B. Chu and C. T. Freeman, "Disturbance Observer Based Iterative Learning Control for Upper Limb Rehabilitation," IECON 2020 The 46th Annual Conference of the IEEE Industrial Electronics Society, Singapore, 2020, pp. 2774-2779, doi: 10.1109/IECON43393.2020.9254696.

[20] Thanh Cao, Trung, Phuoc Doan Nguyen, Nam Hoai Nguyen, and Ha Thu Nguyen. "An indirect iterative learning controller for nonlinear systems with mismatched uncertainties and matched disturbances." International Journal of Systems Science 53, no. 16 (2022): 3375-3389.

[21] Li, Jinsha, and Junmin Li. "Iterative learning control approach for a kind of heterogeneous multi-agent systems with distributed initial state learning." Applied Mathematics and Computation 265 (2015): 1044-1057.

[22] V. L. Donatone, S. Meraglia and M. Lovera, "H∞-based Transfer Learning for UAV Trajectory Tracking," 2022 International Conference on Unmanned Aircraft Systems (ICUAS), Dubrovnik, Croatia, 2022, pp. 354-360, doi: 10.1109/ICUAS54217.2022.9836187.

[23] Z. Chen, X. Liang and M. Zheng, "Iterative Learning for Heterogeneous Systems," in IEEE/ASME Transactions on Mechatronics, vol. 27, no. 3, pp. 1510-1521, June 2022, doi: 10.1109/TMECH.2021.3085211.

[24] Lyu, Ximin, Minghui Zheng, and Fu Zhang. "H_∞ Based Disturbance Observer Design for Non-minimum Phase Systems with Application to UAV Attitude Control." In 2018 Annual American Control Conference (ACC), pp. 3683-3689. IEEE, 2018.

[25] Mellinger D, Michael N, Kumar V. Trajectory generation and control for precise aggressive maneuvers with quadrotors. The International Journal of Robotics Research. 2012;31(5):664-674. doi:10.1177/0278364911434236