IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received October 21, 2019, accepted December 19, 2019, date of publication December 27, 2019, date of current version January 6, 2020.

Digital Object Identifier 10.1109/ACCESS.2019.2962512

Robust Tracking Control of Aerial Robots via a Simple Learning Strategy-Based Feedback Linearization

MOHIT MEHNDIRATTA[1,2], ERKAN KAYACAN[3], MAHMUT REYHANOGLU[4], AND ERDAL KAYACAN[2]

[1]School of Mechanical and Aerospace Engineering, Nanyang Technological University, Singapore 639798
[2]Department of Engineering, Aarhus University, 8200 Aarhus, Denmark
[3]School of Mechanical and Mining Engineering, The University of Queensland, Brisbane, QLD 4072, Australia
[4]Department of Engineering, University of North Carolina at Asheville, Asheville, NC 28804, USA

Corresponding author: Erdal Kayacan (erdal@eng.au.dk)

This work was supported by the Department of Engineering, Aarhus University, under Grant 28173.

ABSTRACT To facilitate accurate tracking in unknown/uncertain environments, this paper proposes a simple learning (SL) strategy for feedback linearization control (FLC) of aerial robots subject to uncertainties. The SL strategy minimizes a cost function defined based on the closed-loop error dynamics of the nominal system via the gradient descent technique to find the adaptation rules for feedback controller gains and disturbance estimate in the feedback control law. In addition to the derivation of the SL adaptation rules, the closed-loop stability for a second-order uncertain nonlinear system is proven in this paper. Moreover, it is shown that the SL strategy can find the global optimum point, while the controller gains and disturbance estimate converge to a finite value which implies a bounded control action in the steady-state. Furthermore, utilizing a simulation study, it is shown that the simple learning-based FLC (SL-FLC) framework can ensure desired closed-loop error dynamics in the presence of disturbances and modeling uncertainties. Finally, to validate the SL-FLC framework in real-time, the trajectory tracking problem of a tilt-rotor tricopter unmanned aerial vehicle under uncertain conditions is studied via three case scenarios, wherein the disturbances in the form of mass variation, ground effect, and wind gust, are induced. The real-time results illustrate that the SL-FLC framework facilitates a better tracking performance than that of the traditional FLC method while maintaining the nominal control performance in the absence of modeling uncertainties and external disturbances, and exhibiting robust control performance in the presence of modeling uncertainties and external disturbances.

INDEX TERMS Feedback linearization control, nonlinear system, uncertain systems, learning control, unmanned aerial vehicle.

I. INTRODUCTION

Owing to the recent advances in automation, computation power, sensors and actuation technology, unmanned aerial vehicles (UAVs) have been explored for a wide variety of applications ranging from search and rescue [1], [2], package delivery [3], [4], traffic monitoring [5], collaboration control [6], and exploration tasks in an unknown environment [7]. Howbeit, for their safe operation in urban environments, a precise path tracking is of extreme importance. Unfortunately, UAVs are strongly coupled, inherently nonlinear systems that are open-loop unstable, which render their control a challenging problem. Besides these difficulties, there are operation and/or environment-specific challenges such as varying operational conditions that induce internal and external uncertainties in the system. All the aforementioned issues imply strong reasons to incorporate either sophisticated model-based controllers or simpler yet learning-based controllers that can learn the system dynamics throughout the operation.

Another issue with the nonlinear systems is the difficulty that lies in their mathematical modeling. Precise modeling requires a number of experimental trials of the system which is arduous as well as time-consuming, and in the end, the

The associate editor coordinating the review of this manuscript and approving it for publication was Yuan Yuan.

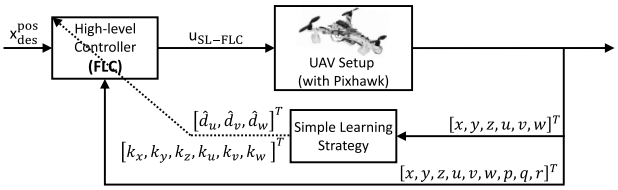


FIGURE 1. The closed-loop control diagram of the SL-FLC framework where the controller gains and disturbance estimate are updated to obtain precise path tracking performance for aerial robots in the presence of uncertainties.

possibility of obtaining the accurate model is limited [8], [9]. Moreover, there are certain applications of UAVs that result in time-varying model parameters, which imply deteriorated performance by model-based controllers. This is mainly due to their dependence on the quality of the available model. Therefore, there is a need for computationally efficient controllers that are accurate, reliable, and adaptive to unpredictable operating conditions.

A. RELATED WORK

The aim of this study is to devise a simple learning (SL) strategy for feedback linearization control (FLC), making the system adaptive to the changing working conditions. In that vein, this section discusses control techniques that have been widely utilized for uncertain nonlinear systems in the literature. In addition, it provides a brief background on the FLC method and learning algorithms that are commonly incorporated within the FLC scheme.

To deal with the challenges of the uncertain nonlinear systems, numerous robust control approaches have been proposed for both unmanned aerial and ground vehicles [10]–[16]. Although the aforementioned methodologies show impressive results in the presence of uncertainties and disturbances, they sacrifice nominal performance, i.e., tracking accuracy, in their absence. Another popular control approach for uncertain systems is the use of inherently robust controllers, e.g., sliding mode controller (SMC) [17]–[20]. According to the SMC theory, selection of high controller gains will result in achieving robust performance against uncertainties. However, this high gain selection might cause undesirable effects, including the common chattering effect, on the system response [21]. The use of learning controllers such as learning-based nonlinear model predictive control, is shown to be yet another alternative approach to deal with uncertainties in the system. Recently, this approach has gained enough popularity and has been demonstrated for the control of aerial robots in [4], [22], [23], a field robot in [24], a tractor-trailer system in [25]. Neural networks (NNs)-based learning is another common learning control approach for which various implementations include NN-based online learning for feedback control in [26], adaptive NN-based stabilization control in [27], and adaptive NN-based backstepping dynamic surface control in [28]. A well-known concern in such applications involving NN-based learning is the difficulty in realizing the analytical proof for the system stability.

To address the stability proof issue, an artifice is to incorporate the feedback linearization technique that transforms the nonlinear system into its linear equivalent; wherein the advanced linear control approaches can be freely utilized. However, the performance of the traditional FLC method is sensitive to uncertainties and disturbances in the system dynamics such that the closed-loop error dynamics of the system cannot converge to zero [29], [30]. For instance, within the trajectory tracking application demonstrated in this study, the induced disturbances vary over time. Hence, the traditional FLC method cannot provide precise path tracking performance. Additionally, FLC with an integral action (FLC-I) has been proposed in the literature to ensure robust control performance. While it can only handle time-invariant uncertainties, it deteriorates nominal control performance in their absence [29]. Unlike FLC and FLC-I methods, the proposed SL strategy does not deteriorate nominal control performance in the absence of disturbances and uncertainties. Moreover, it ensures robust control performance in the presence of time-varying disturbances and uncertainties as distinct from the FLC-I technique.

To overcome the limitations of the traditional FLC, several learning approaches have been utilized within the feedback linearization framework. In [31], artificial NN with a nonlinear autoregressive-moving average model has been developed to learn the feedback linearized inputs for a nonlinear plant. In [32], the Gaussian process method has been used to identify a model for the FLC scheme with limited prior knowledge of the system. Also, a dynamic linearization method has been developed for discrete-time nonlinear systems in the literature [33]–[37]. To summarize, this method builds a linearized model, which is equivalent to the real nonlinear system at every operation point of the closed-loop system dynamics. Since robots usually encounter varying working conditions, the aforementioned techniques may not be good candidates. This is because these techniques utilize the data which is generated in advance, which may no longer fully represent the system during operation. For instance, the total mass of an aerial robot can vary so that the system model changes overtime. Moreover, unlike the aforementioned learning techniques, the proposed simple learning strategy does not require any training data, and it learns the system behavior online.

Another nonlinear control approach using dynamic NN-based input-output feedback linearization has been proposed in [38]. In this approach, the controller generates a control signal that can eliminate the system's nonlinearities utilizing the trained dynamic NN. The controller gains and the weights of the dynamic NN are selected by using particle swarm optimization (PSO) algorithm. Whereas PSO requires powerful computing platforms to obtain the optimal controller gains in real-time operation, the proposed simple learning strategy can obtain the optimal controller gains in less than a millisecond on an inexpensive computing platform. This aspect is crucial for fast robotic applications, wherein there is a tendency to

use cost-effective processors in which computationally efficient algorithms are needed.

B. CONTRIBUTIONS

Motivated by the limitations of the aforementioned methodologies, we develop and implement a simple learning strategy-based feedback linearization control (SL-FLC) algorithm in this study. Within the SL-FLC framework, the controller coefficients and disturbance estimate are updated in the feedback control law by minimizing a cost function which is defined based on the closed-loop error dynamics of the nominal system. Consequently, the SL-FLC framework helps in maintaining the nominal control performance in the absence of uncertainties and disturbances, while exhibits robust control performance in their presence. Additionally, the SL-FLC framework ensures the desired closed-loop error dynamics in the presence of uncertainties and disturbances. The stability of the proposed approach is proven for the considered second-order uncertain nonlinear system. It is also illustrated that the SL strategy can find the global optimum point, and the controller gains and disturbance estimate converge to a finite value. Thus, controller gains and disturbance term will not be modified anymore, which eventually results in a bounded control signal at the steady-state. Along with the theoretical aspects, the proposed algorithm is tested in simulation for the tracking problem of a second-order nonlinear system in the presence of uncertainties and disturbances; wherein the performance of the SL-FLC framework is compared with (nominal) traditional FLC method – without uncertainties and disturbances –, traditional FLC method, and FLC-I method. Furthermore, a real-time performance evaluation of the SL-FLC framework is performed in contrast to the traditional FLC method for the path tracking problem of a tilt-rotor tricopter UAV in the presence of internal and external uncertainties. This paper is a major extension of the authors' previous work in [39]. Significant additions to the previous work include:

1) Global minimum analysis to illustrate the optimality of the obtained gains and disturbance estimate.
2) Simulation study to demonstrate the desired closed-loop error dynamics ensuring capability of the proposed framework in the presence of uncertainties.
3) Extensive experimental evaluation of the proposed framework for three different disturbance scenarios.

C. ORGANIZATION

The organization of this paper is as follows: Section II describes the feedback linearization control problem formulation. Thereafter, the utilized simple learning technique along with the stability proof for a second-order uncertain nonlinear system and the proof of the existence of global minimum are shown in Section III. Then, in Section IV, the proposed SL-FLC framework is validated via a simulation study, followed by an experimental evaluation on a tilt-rotor tricopter UAV for three different case scenarios in Section V. Finally, concluding remarks are given in Section VI.

NOMENCLATURE

$\mathtt{x}$	= state vector
$\mathtt{u}$	= control input
$\Delta(\cdot)$	= modeling uncertainties
$\mathtt{w}$	= external disturbance
$\mathtt{y}$	= system output
$\mathtt{u}_b$	= feedback control action
$\mathtt{u}_f$	= feedforward control action
k	= control gain vector
e	= error vector
r	= reference vector
d	= actual disturbance
$\hat{d}$	= estimated disturbance
$c(\cdot,\cdot)$	= desired closed-loop error dynamics
C	= cost function
α_i	= learning rate for the i^{th} controller gain
$\alpha_{\hat{d}}$	= learning rate for disturbance estimate

II. PROBLEM FORMULATION

Consider a second-order uncertain nonlinear system defined as:

$$\dot{x}_1 = x_2, \tag{1a}$$
$$\dot{x}_2 = f(\mathtt{x},\mathtt{u}) + \Delta(\mathtt{x}) + \mathtt{w}(t), \tag{1b}$$

where $\mathtt{x} = [x_1, x_2]^T \in \mathbb{R}^2$ is the state vector, $\mathtt{u} \in \mathbb{R}$ is the control input, and $\mathtt{w}(t)$ is the time-varying external disturbance. Also, $f(\mathtt{x},\mathtt{u}) \in \mathbb{R}$ and $\Delta(\mathtt{x}) \in \mathbb{R}$ are smooth, continuous differentiable and nonlinear functions, wherein, $\Delta(\mathtt{x}) \in \mathbb{R}$ represents the modeling uncertainties. The objective is to find a control input $\mathtt{u}$ such that the system tracks the reference trajectory $r(t) = [r_1(t), r_2(t)]^T$, where $r_2 = \dot{r}_1$, with an acceptable accuracy while all the states and the control remain bounded.

A. TRADITIONAL FLC

To facilitate the tracking of the given reference trajectory, we define the tracking error vector $e(t) = [e_1(t), e_2(t)]^T$, where $e_1 = r_1 - x_1$ and $e_2 = r_2 - x_2$. Consider the following traditional FLC law:

$$\mathtt{u} = \beta(\mathtt{x}, \mathtt{u}_b^{\text{FLC}}, \mathtt{u}_f), \tag{2}$$

where $\mathtt{u}_b^{\text{FLC}}$ and $\mathtt{u}_f$ are the feedback and feedforward control actions, respectively, which are expressed as follows:

$$\mathtt{u}_b^{\text{FLC}} = ke = k_1e_1 + k_2e_2, \tag{3}$$
$$\mathtt{u}_f = \dot{r}_2, \tag{4}$$

where $k = [k_1, k_2], k_i > 0\ (i = 1, 2)$, is a control gain vector. The feedback function $\beta(\mathtt{x}, \mathtt{u}_b^{\text{FLC}}, \mathtt{u}_f)$ is chosen such that

$$f(\mathtt{x}, \beta(\mathtt{x}, \mathtt{u}_b^{\text{FLC}}, \mathtt{u}_f)) = \mathtt{u}_b^{\text{FLC}} + \mathtt{u}_f, \tag{5}$$

and consequently, the closed-loop error dynamics take the following form:

$$\dot{e}_2 + k_2e_2 + k_1e_1 = -d(t). \tag{6}$$

Here, the term $d(t)$ represents a lumped disturbance parameter which comprises of modeling uncertainties and external disturbance in the closed-loop error dynamics, which is expressed as:

$$d(t) = \Delta(\mathrm{x}) + \mathrm{w}(t). \tag{7}$$

Assumption 1: The lumped disturbance parameter in (6) is bounded such that $d^* = \sup_{t>0}|d(t)|$ exists.

Since $k_i > 0$ $(i = 1, 2)$, the closed-loop error dynamics for $d(t) = 0$ are globally exponentially stable at $e_1 = e_2 = 0$.

Remark 1: It is to be noted that the current paper assumes that the disturbances consist of a constant and vanishing perturbation terms. The cases where the disturbance includes non-vanishing perturbation terms are left as future work.

Remark 2: Equation (6) demonstrates that it is not possible to drive the error dynamics to the desired equilibrium point utilizing the FLC law proposed in (2). The reason being is the nonzero right-hand side in (6) which is due to the presence of modeling uncertainties in the system model and/or external disturbances. This explains why the traditional FLC method is sensitive to disturbances.

B. FLC WITH INTEGRAL ACTION

Now, to make the FLC method robust against modeling uncertainties as well as disturbances, an integral action can be added to the control law of traditional FLC scheme. That is, the control law for FLC-I method is formulated in a similar manner to the traditional FLC method in (2). While the expression for the feedforward control action is the same as specified in (4), the feedback control expression, on the other hand, is modified to include the effect of the integral action:

$$\mathrm{u}_b^{\text{FLC-I}} = ke = k_1e_1 + k_2e_2 + k_{int}\int_0^t e_1 dt, \tag{8}$$

where $k = [k_1, k_2, k_{int}]$ is the updated positive control gain vector and e is the error vector as defined previously. Additionally, it is assumed that all the assumptions utilized in the traditional FLC method's case are valid.

Applying the control law for the FLC-I method to the uncertain nonlinear system in (1), the closed-loop error dynamics can be obtained as follows:

$$\ddot{e}_2 + k_2\dot{e}_2 + k_1\dot{e}_1 + k_{int}e_1 = -\dot{d}(t), \tag{9}$$

where the term $d(t)$ is the same as defined in (7). The above equation implies that if the modeling uncertainties and external disturbances have a steady-state value, i.e., $\dot{d}(t) = 0$, then the state can be driven to the desired equilibrium point.

Remark 3: Equation (9) shows that if the modeling uncertainties and external disturbances have a steady-state value, i.e, they are time-invariant, then the FLC-I method is robust against them and the steady-state error in the system is eliminated. However, it is to be noted that adding an integral action may result in a degraded performance in comparison to the nominal performance of the traditional FLC method in the absence of uncertainties [29].

III. SIMPLE LEARNING STRATEGY

The SL strategy for the feedback linearization method is designed according to the desired closed-loop error dynamics of the nonlinear system. In order to make the FLC method adaptive, the SL strategy updates the controller gains and disturbance term within the FLC formulation by minimizing a cost function which is defined as the square of the closed-loop error dynamics. The new feedback control law is written as follows:

$$\mathrm{u}_b^{\text{SL-FLC}} = ke - \hat{d} = k_1e_1 + k_2e_2 - \hat{d}, \tag{10}$$

where $\hat{d}$ represents the estimated disturbance term. The closed-loop error dynamics can be written as:

$$\dot{e}_1 = e_2, \tag{11a}$$
$$\dot{e}_2 = -k_1e_1 - k_2e_2 - d + \hat{d}. \tag{11b}$$

A. UPDATE RULES

The requirement to realize robust control performance by the FLC method is that the desired closed-loop error dynamics, defined as:

$$c(e, k^{\text{des}}) = \dot{e}_2 + k_2^{\text{des}}e_2 + k_1^{\text{des}}e_1, \tag{12}$$

should converge to zero. Since $\dot{e}_2 = -k_1e_1 - k_2e_2 - d + \hat{d}$, $c(e, k^{\text{des}})$ depends on k_1, k_2, $\hat{d}$, e_1, e_2, and d, i.e., the controller gains, disturbance estimate, error variables and the disturbance. This forms the working principle within the SL-FLC framework i.e., to minimize a cost function defined based on the closed-loop error dynamics such that the system error and disturbance estimation error can instantaneously converge to zero. In that vein, the cost function which is minimized to obtain the robust control performance is considered as the square of the desired closed-loop error dynamics defined in (12). That is, the closed-loop error function (or cost function) is written as follows:

$$C = \tfrac{1}{2}\Big(c(e, k^{\text{des}})\Big)^2. \tag{13}$$

The above equation implies that if the closed-loop error function C converges to zero, then the robust control performance condition, i.e., $c = 0$, is satisfied such that the error will converge to zero.

In this study, a first-order iterative optimization algorithm, i.e., gradient descent is favored to minimize the closed-loop error function C. The reason behind this selection is the requirement of lower computational power to compute the optimal solution. In the gradient descent approach, steps are taken proportional to the negative of the gradient of the closed-loop error function, i.e., ΔC, to find the minimum. Since the expression for the proposed closed-loop error function C comprises of the errors, controller gains, and disturbance estimates, its partial derivative has to be taken to obtain the gradient. Therefore, the following rule is utilized

for updating the controller gains of the feedback linearization controller:

$$\dot{k}_i = -\alpha_i \frac{\partial C}{\partial k_i}, \tag{14}$$

where $\alpha_i > 0$ is the learning rate for the i^{th} controller gain. Next, utilizing the chain rule in (14), the expression can be rewritten as follows:

$$\dot{k}_i = -\alpha_i c(e, k^{\text{des}}) \frac{\partial c(e, k^{\text{des}})}{\partial k_i}. \tag{15}$$

It is implicit from the system definition in (1) that the control input gains appear only with $\dot{e}_2$ within the expression for the desired closed-loop error dynamics $c(e, k^{\text{des}})$. Thus, (15) can be rewritten as:

$$\dot{k}_i = -\alpha_i c(e, k^{\text{des}}) \frac{\partial \dot{e}_2}{\partial k_i}, \tag{16}$$

which eventually reduces to the following form, utilizing (11b):

$$\dot{k}_i = \alpha_i c(e, k^{\text{des}}) e_i. \tag{17}$$

Similarly, the formulation of the update rule for the disturbance estimate is obtained. That is, like (14), the update rule for the disturbance estimate is written incorporating the gradient descent rule as:

$$\dot{\hat{d}} = -\alpha_{\hat{d}} \frac{\partial C}{\partial \hat{d}}, \tag{18}$$

where $\alpha_{\hat{d}} > 0$ is the learning rate for the disturbance estimate $\hat{d}$. Again, after using the chain rule and subsequently performing some algebraic manipulations, we can compute the final expression for the update rule (18) as:

$$\dot{\hat{d}} = -\alpha_{\hat{d}} c(e, k^{\text{des}}). \tag{19}$$

We would like to emphasize that within (17) and (19), the controller gains and disturbance estimate are updated until the condition $c(e, k^{\text{des}}) = 0$ is fulfilled.

Remark 4: It is possible to design a nonlinear disturbance observer to estimate the lumped disturbance $d(t)$ following the methodology in [40]. The disturbance estimator design in this paper is based on a gradient descent rule while, in general, nonlinear disturbance observers are designed based on Lyapunov techniques. In essence, both approaches follow the same control design strategy: (1) design a controller to achieve stability and other performance specifications (e.g., asymptotic tracking) assuming that the disturbance is measurable and available for feedback, (2) design a disturbance estimator, and (3) use the disturbance estimate in place of the disturbance in the control law. Furthermore, unlike any (nonlinear) disturbance observer-based control paradigm, the proposed control framework updates the controller gains of the underlying FLC method which further exhibits robust control performance in the presence of disturbances.

B. STABILITY PROOF

In this study, a Routh-Hurwitz criterion-based stability analysis is utilized to prove the closed-loop system stability with the SL algorithm. In that vein, the closed-loop dynamics can be rewritten utilizing (11a) and (11b) as follows:

$$\ddot{e}_1 + k_2 \dot{e}_1 + k_1 e_1 - \hat{d} + d = 0. \tag{20}$$

Time differentiating the above equation and using the fact that $\dot{d} = 0$[1], we obtain,

$$\dddot{e}_1 + k_2 \ddot{e}_1 + k_1 \dot{e}_1 + \dot{k}_2 \dot{e}_1 + \dot{k}_1 e_1 - \dot{\hat{d}} = 0. \tag{21}$$

utilizing (12), (17), and (19), the expressions for $\dot{k}_i$ and $\dot{\hat{d}}$ can be derived as:

$$\dot{k}_i = \alpha_i (\ddot{e}_1 + k_2^{\text{des}} \dot{e}_1 + k_1^{\text{des}} e_1) e_i, \tag{22}$$

$$\dot{\hat{d}} = -\alpha_{\hat{d}} (\ddot{e}_1 + k_2^{\text{des}} \dot{e}_1 + k_1^{\text{des}} e_1). \tag{23}$$

Let $\eta = [e_1 \quad \dot{e}_1 \quad \ddot{e}_1]^T$ denote the state. Plugging the above expressions into the closed-loop dynamics yields the following:

$$\dddot{e}_1 + b_1(\eta)\ddot{e}_1 + b_2(\eta)\dot{e}_1 + b_3(\eta) e_1 = 0, \tag{24}$$

where,

$$b_1(\eta) = k_2 + \alpha_{\hat{d}} + \beta(\eta), \; b_2(\eta) = k_1 + k_2^{\text{des}}(\alpha_{\hat{d}} + \beta(\eta)), \tag{25}$$

$$b_3(\eta) = k_1^{\text{des}}(\alpha_{\hat{d}} + \beta(\eta)), \; \beta(\eta) = \alpha_1 e_1^2 + \alpha_2 \dot{e}_1^2. \tag{26}$$

The closed-loop dynamics (21) can now be expressed in a pseudo-linear form [41] as:

$$\dot{\eta} = A(\eta)\eta, \tag{27}$$

where,

$$A(\eta) = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -b_3(\eta) & -b_2(\eta) & -b_1(\eta) \end{bmatrix}. \tag{28}$$

Clearly, $b_i(\eta) > 0, \quad i = 1, 2, 3, \quad \forall \eta$. By applying Routh-Hurwitz criterion, it can be shown that the eigenvalues of $A(0)$ are in the open left-half plane (i.e., the closed-loop system is asymptotically stable) if $b_1(0)b_2(0) > b_3(0)$, where,

$$b_1(0) = k_2 + \alpha_{\hat{d}}, \; b_2(0) = k_1 + k_2^{\text{des}} \alpha_{\hat{d}}, \; b_3(0) = k_1^{\text{des}} \alpha_{\hat{d}}, \tag{29}$$

or equivalently if:

$$k_2^{\text{des}} \alpha_{\hat{d}}^2 + (k_1 + k_2 k_2^{\text{des}} - k_1^{\text{des}}) \alpha_{\hat{d}} + k_1 k_2 > 0. \tag{30}$$

In what follows, we choose k_i, k_i^{des}, and $\alpha_{\hat{d}}$ such that this stability condition is satisfied.

[1]In this work, we assume that the average rate of change of the induced disturbances is much lower than the states e_1 and e_2.

C. GLOBAL MINIMUM

The most important concern in the simple learning strategy is that the closed-loop error dynamics may reach some local minima and stay in that location. Hence, this subsection provides an analytical proof to show that there exists no local minima for the formulation of the simple learning strategy. In particular, it is shown that the second derivatives of the cost function with respect to the variables k_i and $\hat{d}$ have the same sign, thus the cost function does not have a change in the curvature sign through the variables. This implies that the cost function does not have local minima through these variables.

Utilizing the method in the previous subsection, the second derivative of the cost function C with respect to k_i is obtained as follows:

$$\frac{\partial^2 C}{\partial k_i^2} = -e_i \underbrace{\frac{\partial (c(e, k^{\text{des}})}{\partial k_i}}_{-e_i}, \\ = (e_i)^2. \tag{31}$$

Similarly, the second derivative of the cost function with respect to the disturbance estimate $\hat{d}$ is computed as follows:

$$\frac{\partial^2 C}{\partial \hat{d}^2} = \underbrace{\frac{\partial (c(e, k^{\text{des}})}{\partial \hat{d}}}_{1}, \\ = 1. \tag{32}$$

Equations (31) and (32) show that the sign of the curvature of the cost function for the controller gains and the disturbance estimate is always positive; thus, there exist no local minima such that the closed-loop error dynamics reach to the global minimum. After reaching the global minimum, since α_i and α_d are constant and positive, the controller gains update law in (17) and the disturbance estimate update law in (19) converge to a finite value. Furthermore, a finite value for the coefficients in steady-state results in a bounded control action.

IV. SIMULATION STUDY

This section illustrates the implementation of the proposed SL-FLC framework together with traditional FLC and FLC-I methods on a simulation example. The main purpose of this simulation study is to demonstrate that the SL-FLC can ensure the desired closed-loop error dynamics of the system in the presence of model uncertainties and external disturbances.

For the simulation study, a second-order dynamical system is considered which is of the following form:

$$\dot{x}_1 = x_2, \tag{33a}$$
$$\dot{x}_2 = x_2^3 + \mathtt{u} + \Delta(\mathtt{x}) + \mathtt{w}(t), \tag{33b}$$

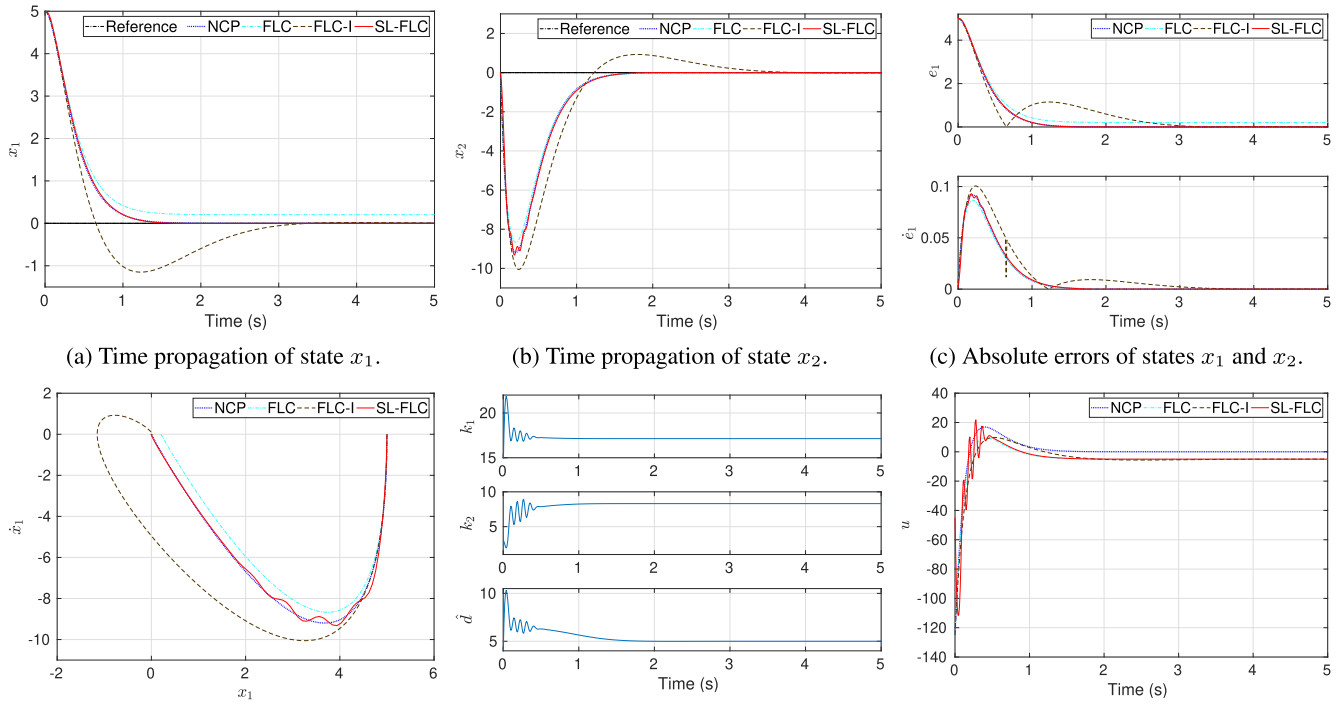
The entire simulation runs for a total of 5s with a sampling time of 0.01s. The initial conditions on the states are $\mathtt{x} = [5, 0]^T$. In addition, the reference signal r is set to zero throughout the simulation time, i.e., $r = [0, 0]^T$. The controller gains for the desired closed-loop error dynamics ($c = \ddot{e}_1 + k_2^{\text{des}}\dot{e}_1 + k_1^{\text{des}}e_1$) are selected as $k^{\text{des}} = [k_1^{\text{des}}, k_2^{\text{des}}]^T = [25, 10]^T$. Moreover, the initial conditions for the controller gains are selected as $k(0) = [9, 3]^T$, while the initial condition for the disturbance estimate is taken as $\hat{d}(0) = 0$. Furthermore, the learning rates α_i and α_d are set to 0.75 and 3, respectively.

To show the desired closed-loop error dynamics ensuring-capability of the SL-FLC framework, i.e., realizing the nominal control performance (NCP) which describes the control performance of the system controlled by the traditional FLC method in the absence of model uncertainties and external disturbance, the SL-FLC framework controls the aforementioned system in the presence of model uncertainties and external disturbance. In addition, its control performance is compared with traditional FLC and FLC-I methodologies. According to (7), the lumped disturbance formulation is equal to the summation of the model uncertainties and external disturbance, i.e., $d(t) = \Delta(\mathtt{x}) + \mathtt{w}(t)$. An external force is imposed on the system, i.e., $\mathtt{w}(t) = 5$, as the external disturbance while there exist the modeling uncertainties $\Delta(\mathtt{x}) = (x_1)^2$ on the system. Thus, the disturbance imposed on the system is equal to 5, i.e., $d(t) = 5$ after the states converge to zero.

The control performances of all the control algorithms are shown in Figs. 2a and 2b. While NCP shows the nominal control performance of the system in the absence of model uncertainties and external disturbance, the other control algorithms reflect their respective performance in the presence of model uncertainties and external disturbance. As observed, the SL-FLC framework ensures the nominal control performance (i.e., the desired closed-loop error dynamics) in the presence of disturbances while the FLC and FLC-I methodologies cannot ensure the nominal control performance. Moreover, the FLC method is not robust to disturbances as stated in Remark 2 while the FLC-I method provides robust control performance against disturbance. However, the FLC-I method causes undesired effects such as overshoots and large settling time as stated in Remark 3.

The absolute values of the errors and error rates are shown in Fig. 23. The SL-FLC framework realizes a similar performance as that of the NCP which validates that it ensures the nominal control performance in the presence of model uncertainties and external disturbance. Moreover, the error of the system controlled by the FLC method cannot converge to zero, whereas the error of the system controlled by the FLC-I method can converge to zero with a large settling time. In addition, the phase portraits for all the controllers are shown in Fig. 2d. It is visualized that the phase portrait for the SL-FLC framework converges to the nominal phase portrait (i.e., NCP) in the presence disturbances. This again implies that the SL-FLC framework is capable of ensuring the desired closed-loop error dynamics of the system in the presence of modeling uncertainties and external disturbance.

Next, the time updates of the controller gains and disturbance estimate for the SL-FLC framework are presented



(a) Time propagation of state x_1.

(b) Time propagation of state x_2.

(c) Absolute errors of states x_1 and x_2.

(d) Phase portraits of states x_1 and x_2.

(e) Adaptations of the controller gains and disturbance estimate.

(f) Control signals from different controllers.

FIGURE 2. Nominal control performance along with the traditional FLC method, FLC-I method, and SL-FLC framework for a second-order dynamical system in the presence of modeling uncertainties and external disturbance. From the results, it is evident that even in the presence of disturbance, the SL-FLC framework can match the nominal control performance.

in Fig. 2e. As observed, the controller gains converge to some values in steady-state response from their selected initial values ($k(0) = [9, 3]^T$). Also, the disturbance estimate becomes equal to the summation of the modeling uncertainties and external disturbance, i.e., $d(t) = \Delta(\mathbf{x}) + \mathrm{w}(t) = 5$. Furthermore, the control inputs for all algorithms are shown in Fig. 2f. Note that, in steady-state response, the control signal for the NCP is equal to zero since there exists no disturbance, whereas for the others the control signals are nonzero due to the external disturbance imposed on the system.

V. EXPERIMENTAL VALIDATION

After the implementation in simulation, next, the experimental validation of the proposed SL-FLC framework on an aerial robot is exhibited. Since its dynamics is non-trivial, first a brief discussion about the considered aerial system along with its mathematical model is provided. Thereafter, the design of the SL-FLC framework as a high-level controller for the aerial robot is illustrated. Finally, the comparison results of the SL-FLC and the traditional FLC method for three disturbance scenarios are presented.

A. AERIAL ROBOT

To test the efficacy of the proposed algorithm, an aerial package delivery problem is considered in this work. The motivation to analyze this problem is the accompanying disturbances that the UAV has to tackle while executing the delivery tasks. That includes mass variation while dropping the packages, ground effect disturbance while descending to precisely place the package on the ground, and wind gust disturbance due to outdoor flights.

The aerial robot incorporated for this study is shown in Fig. 3. It is the same 3D-printed tilt-rotor tricopter that is utilized in our previous work [39]. To summarize, it is a custom-designed experimental platform, wherein all the required electronics are integrated into the frame design to obtain a compact and lightweight system. Also, to mimic the package delivery application, it can sequentially drop weight blocks with the help of a payload dropping mechanism which is mounted at the base of the UAV. In addition, the UAV houses a Pixhawk flight controller for low-level stabilization along with an on-board embedded processor (Raspberry Pi 3) which executes all the control codes as well as controls the servomotor of the dropping mechanism. For more details regarding the UAV setup, one may refer to [39].

1) SYSTEM MODELING

For the system modeling, the aerial robot shown in Fig. 3 is considered to be a rigid-body having two stationary rotors in the front and one non-stationary (or tilting) rotor at the back. Amongst the front rotors, the right rotates clockwise and the left rotor rotates counter-clockwise, whereas the back rotor rotates counter-clockwise.

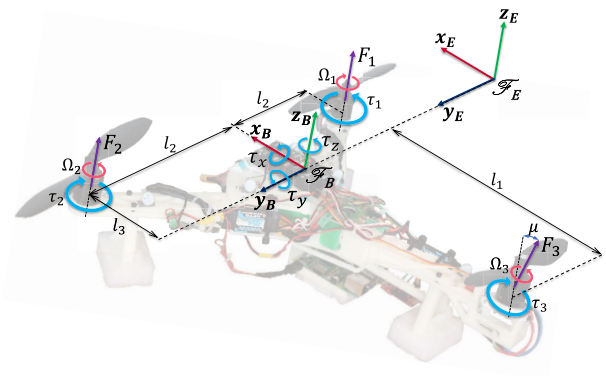


FIGURE 3. The 3D-printed tilt-rotor tricopter UAV, wherein, the body frame is represented by $\mathcal{F}_B$ and the Earth-fixed frame is represented by $\mathcal{F}_E$. Notation: $[\Omega_1, \Omega_2, \Omega_3]$ - rotational RPM of three rotors; μ - tilting angle of the back rotor; $[F_1, F_2, F_3]$ - forces generated by three rotors; $[\tau_1, \tau_2, \tau_3]$ - moments generated by three rotors; $[\tau_x, \tau_y, \tau_z]$ - external moments acting on the tricopter body.

The translational kinematic equations are given as:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} u \\ v \\ w \end{bmatrix}, \tag{34}$$

where x, y, z represent the translational positions in the frame $\mathcal{F}_B$, while u, v, w represent the translational velocities defined in the frame $\mathcal{F}_B$. On the other hand, the rigid-body dynamic equations for a tilt-rotor tricopter UAV in the body coordinate system ($\mathcal{F}_B$) are formulated based on the Newton-Euler methodology [42]. Within these equations, the tricopter is assumed to be a point mass, wherein all the forces act at the CG:

$$\dot{u} = rv - qw + g\sin(\theta) + \frac{1}{m}F_x, \tag{35a}$$

$$\dot{v} = pw - ru - g\sin(\phi)\cos(\theta) + \frac{1}{m}F_y, \tag{35b}$$

$$\dot{w} = qu - pv - g\cos(\phi)\cos(\theta) + \frac{1}{m}F_z, \tag{35c}$$

where ϕ, θ represent the rotational attitude in frame $\mathcal{F}_E$, and p, q, r are the rotational velocities that are defined in frame $\mathcal{F}_B$. In addition, the terms F_x, F_y, F_z represent the total external forces that act on the UAV in frame $\mathcal{F}_B$. Moreover, some constant intrinsic parameters for the utilized custom-designed UAV are listed in Table 1. For more details regarding the system modeling and UAV, one is referred to [39]. It is to be noted that the takeoff mass of the UAV inclusive of all payload is about $m = 1.875$ kg.

Finally, the continuous-time nonlinear model of the considered UAV can be represented as:

$$\dot{\mathrm{x}}(t) = \mathrm{f}\big(\mathrm{x}(t), \mathrm{u}(t)\big), \tag{36}$$

where the state vector $\mathrm{x} \in \mathbb{R}^6$ and control vector $\mathrm{u} \in \mathbb{R}^3$ are:

$$\mathrm{x} = [x, y, z, u, v, w]^T, \tag{37}$$

$$\mathrm{u} = [\phi, \theta, F_z]^T, \tag{38}$$

and $\mathrm{f}(\cdot,\cdot)$:$\mathbb{R}^6 \times \mathbb{R}^3 \longrightarrow \mathbb{R}^6$.

TABLE 1. Intrinsic parameters for the 3D-printed tilt-rotor tricopter.

Par.	Description	Value
m	Mass of tricopter UAV	1.418 kg
l_1	Moment arm	0.284m
l_2	Moment arm	0.212m
l_3	Moment arm	0.092m
K_f	Aerodynamic force coefficient	4.209×10^{-4} N-s^2

2) SL-FLC FOR HIGH-LEVEL CONTROL

Three SL-FLCs frameworks are independently employed to achieve a precise tracking performance by the UAV. In principle, they simultaneously perform position as well as velocity control along the three x-, y-, and z-axes. The overall control framework is illustrated in Fig. 1. Once the position reference ($\mathrm{x}_{\mathrm{des}}^{\mathrm{pos}}$) is given to the controller, the desired velocities ($\mathrm{x}_{\mathrm{des}}^{\mathrm{vel}} = [u_{\mathrm{des}}, v_{\mathrm{des}}, w_{\mathrm{des}}]^T$) are computed utilizing the rate of change of the position setpoints information, i.e., $\mathrm{x}_{\mathrm{des}}^{\mathrm{vel}} = \dot{\mathrm{x}}_{\mathrm{des}}^{\mathrm{pos}}$ [2]. Subsequently, these velocity references ($\mathrm{x}_{\mathrm{des}}^{\mathrm{vel}}$) along with the position reference ($\mathrm{x}_{\mathrm{des}}^{\mathrm{pos}}$) are utilized in computing the final attitude angles and throttle commands ($\mathrm{u}_{\text{SL-FLC}} = [\phi_{\mathrm{des}}, \theta_{\mathrm{des}}, F_{z_{\mathrm{des}}}]^T$) that are given to the low-level controller. Moreover, the following update rules are utilized within the three feedback linearization controllers [39]:

$$\begin{aligned}\theta_{\mathrm{des}} &= sin^{-1}[\tfrac{1}{g}\{qw - rv + \dot{u}_{\mathrm{des}} + k_u(u_{\mathrm{des}} - u) \\ &\quad + k_x(x_{\mathrm{des}} - x) - \hat{d}_u\}],\end{aligned} \tag{39}$$

$$\begin{aligned}\phi_{\mathrm{des}} &= sin^{-1}[\tfrac{1}{gcos(\theta_{\mathrm{des}})}\{pw - ru - \dot{v}_{\mathrm{des}} - k_v(v_{\mathrm{des}} - v) \\ &\quad - k_y(y_{\mathrm{des}} - y) + \hat{d}_v\}],\end{aligned} \tag{40}$$

$$\begin{aligned}F_{z_{\mathrm{des}}} &= m[pv - qu + gcos(\phi_{\mathrm{des}})cos(\theta_{\mathrm{des}}) + \dot{w}_{\mathrm{des}} \\ &\quad + k_w(w_{\mathrm{des}} - w) + k_z(z_{\mathrm{des}} - z) - \hat{d}_w],\end{aligned} \tag{41}$$

where k_x, k_y, k_z and k_u, k_v, k_w are the controller gains for position and velocity, respectively. In addition, the terms $\hat{d}_u$, $\hat{d}_v$, $\hat{d}_w$ are the disturbance estimates which are previously defined in (7).

As mentioned before, the proposed SL strategy makes the FLC adaptive to the changing operational conditions. It updates the controller gains and the disturbance estimates according to the expressions in (17) and (19), respectively, while utilizing the position error, and the velocity errors along with their first derivatives. For our application, the expressions for the desired closed-loop error dynamics along the three axes reduce to:

$$c_x(e, k^{\mathrm{des}}) = \ddot{e}_x + k_u^{\mathrm{des}}\dot{e}_x + k_x^{\mathrm{des}}e_x, \tag{42}$$

$$c_y(e, k^{\mathrm{des}}) = \ddot{e}_y + k_v^{\mathrm{des}}\dot{e}_y + k_y^{\mathrm{des}}e_y, \tag{43}$$

$$c_z(e, k^{\mathrm{des}}) = \ddot{e}_z + k_w^{\mathrm{des}}\dot{e}_z + k_z^{\mathrm{des}}e_z, \tag{44}$$

where e_x, e_y, e_z are the translational position error, $\dot{e}_x$, $\dot{e}_y$, $\dot{e}_z$ are the translational velocity error, and $\ddot{e}_x$, $\ddot{e}_y$, $\ddot{e}_z$ are the translational acceleration error. Also, the terms $k_x^{\mathrm{des}}, k_y^{\mathrm{des}}, k_z^{\mathrm{des}}$

[2]It is to be noted that the position derivatives are filtered to ensure smooth velocity references.

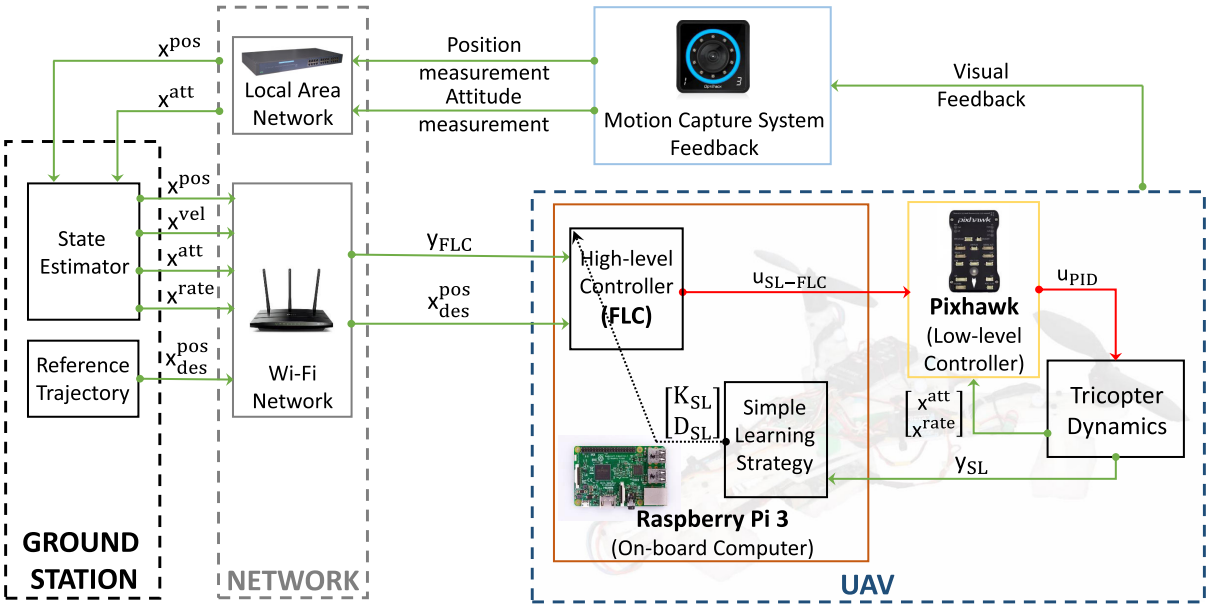


FIGURE 4. Real-time implementation: block diagram of the overall implementation, wherein the block naming 'Raspberry Pi 3' represents the on-board computer which executes the proposed SL-FLC framework. It is to be noted that the low-level attitude controller in Pixhawk comprises of three proportional-integral-derivative (PID) controllers, which are designed separately for the three axes (roll, pitch, yaw). Also, the overall output vector of the low-level controller is: $\mathrm{u_{PID}} = [\Omega_1, \Omega_2, \Omega_3, \mu]^T$. Notation: $\mathbf{x}^{\mathrm{pos}} = [x, y, z]^T$; $\mathbf{x}^{\mathrm{vel}} = [u, v, w]^T$; $\mathbf{x}^{\mathrm{att}} = [\phi, \theta, \psi]^T$; $\mathbf{x}^{\mathrm{rate}} = [p, q, r]^T$; $\mathbf{x}^{\mathrm{pos}}_{\mathrm{des}} = [x_{\mathrm{des}}, y_{\mathrm{des}}, x_{\mathrm{des}}]^T$; $\mathrm{y_{FLC}} = [x, y, z, u, v, w, p, q, r]^T$; $\mathrm{y_{SL}} = [x, y, z, u, v, w]^T$; $\mathrm{K_{SL}} = [k_x, k_y, k_z, k_u, k_v, k_w]^T$; $\mathrm{D_{SL}} = [\hat{d}_u, \hat{d}_v, \hat{d}_w]^T$.

and k_u^{des}, k_v^{des}, k_w^{des} are the desired FLC gains for position and velocity, respectively.

Remark 5: To obtain the above update rules (39)-(41), the external forces F_x and F_y are taken to be zero within the system model (35).

Remark 6: Due to a time-varying trajectory, a nonzero tracking error always occurs. Therefore, to avoid the continuous update (or rise) of the controller gain values for insignificant errors, a dead-zone (or saturation zone) is implemented. That is, below the threshold for tracking error (and its derivatives), they are taken to be zero within the cost function, and hence, the controller gains do not get updated. In practice, the threshold must be selected to be less than the desired accuracy but more than the noise on the measurements.

B. EXPERIMENTAL RESULTS

This section illustrates the real-time experimental results of the proposed SL-FLC framework which is utilized for the high-level position tracking control of the considered aerial robot. In addition to the tracking performance, its robustness to varying UAV dynamics is also analyzed, wherein, the uncertainties are explicitly induced in the form of: (i) mass variation, (ii) ground effect, and (iii) wind gust. In order to appreciate the importance of learning, all the experiments are performed with two types of controllers namely, traditional FLC (without learning) and FLC incorporating the SL algorithm (i.e. SL-FLC framework).

In terms of the real-time implementation, all the control codes are written in C++ for their efficient execution on-board the Raspberry Pi 3 flight computer in a ROS (robot-operating-system) environment. The average sampling frequency achieved for the controllers throughout each experiment is about 100-Hz. Additionally, an indoor motion capture system from OptiTrack, consisting of eight 240 FPS (frames per second) cameras, is utilized for localization during the experiments, which is depicted in the indoor experimental setup shown in Fig. 5. The motion capture system provides the position and attitude feedback of the UAV over a local network. Utilizing this information, the high-level FLC algorithm computes the desired attitude angles and throttle setpoints for the low-level controller which are sent via serial communication. Finally, the actuator commands computed by the low-level controller are given to the rotors and tilting servo of the UAV. Moreover, the overall real-time implementation can be visualized from the block diagram in Fig. 4.

In terms of the controller tuning, the following gains for the desired closed-loop error dynamics are obtained via the trial-and-error procedure and are directly used within the traditional FLC method:

$$k_x^{\mathrm{des}} = 4.5, \quad k_y^{\mathrm{des}} = 4.7, \quad k_z^{\mathrm{des}} = 8.0, \tag{45}$$

$$k_u^{\mathrm{des}} = 4.0, \quad k_v^{\mathrm{des}} = 4.1, \quad k_w^{\mathrm{des}} = 3.8. \tag{46}$$

On the other hand, the controller gains and learning rates utilized for initializing the SL-FLC framework are as follows:

$$k_x(0) = 4.3, \quad k_y(0) = 4.5, \quad k_z(0) = 7.7, \tag{47}$$

$$k_u(0) = 3.8, \quad k_v(0) = 3.9, \quad k_w(0) = 3.6, \tag{48}$$

$$\alpha_x = 0.02, \quad \alpha_y = 0.02, \quad \alpha_z = 0.03, \tag{49}$$

$$\alpha_u = 0.02, \quad \alpha_v = 0.02, \quad \alpha_w = 0.03. \tag{50}$$

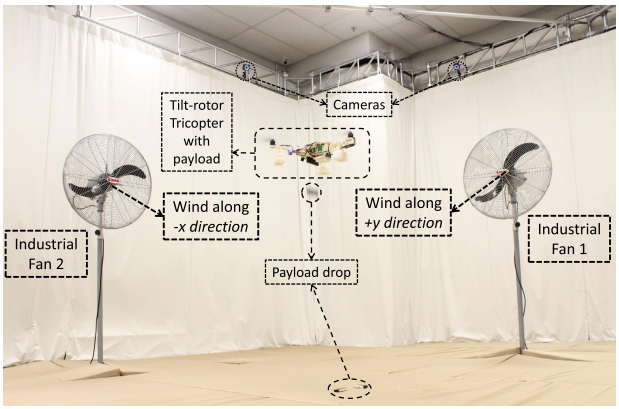


FIGURE 5. Experimental setup: 3D-printed tilt-rotor tricopter UAV in the motion capture system lab along with two industrial fans.

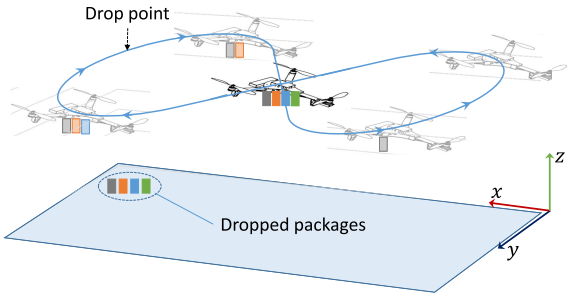


FIGURE 6. Package delivery problem in case scenario I.

The above control gains and learning rates are also obtained via the trial-and-error procedure such that a stable UAV response is obtained.

In the subsequent part, each of the three experiments is discussed in detail. It is emphasized that different trajectories are utilized for all experiments in order to exhibit the robustness of the control framework towards various trajectories.

1) CASE SCENARIO I - MASS VARIATION

In the first experiment, a package delivery problem is analyzed where a UAV carrying payload tracks a reference trajectory in 3D while dropping each package to its time-based designated location, as depicted in Fig. 6. The empty weight of the considered tilt-rotor tricopter UAV is 1.418 kg while the total weight after the addition of full payload is 1.875 kg. Therefore, the total payload drop of 457 g, which is more than 32% of the empty mass of the UAV, results in a significant variation in the UAV dynamics that favors the use of learning for this application.

For this payload dropping experiment, a time-based 8-shaped trajectory of length 2.6m along x-direction and 1.3m along y-direction is incorporated, while the initial state of the UAV is $\mathrm{x}(0) = [0, 0, 1.5, 0, 0, 0]^T$. For both traditional FLC method and SL-FLC framework, first a complete 8-shaped trajectory is tracked by the UAV with a full payload, followed by the dropping of each payload block once per lap at fixed time intervals in the increasing order of their mass: $86\ g \rightarrow 114\ g \rightarrow 114\ g \rightarrow 143\ g$. Besides the controller tuning, which is done incorporating the control gains and learning rates specified in (45) - (50), the disturbance vector is initialized as:

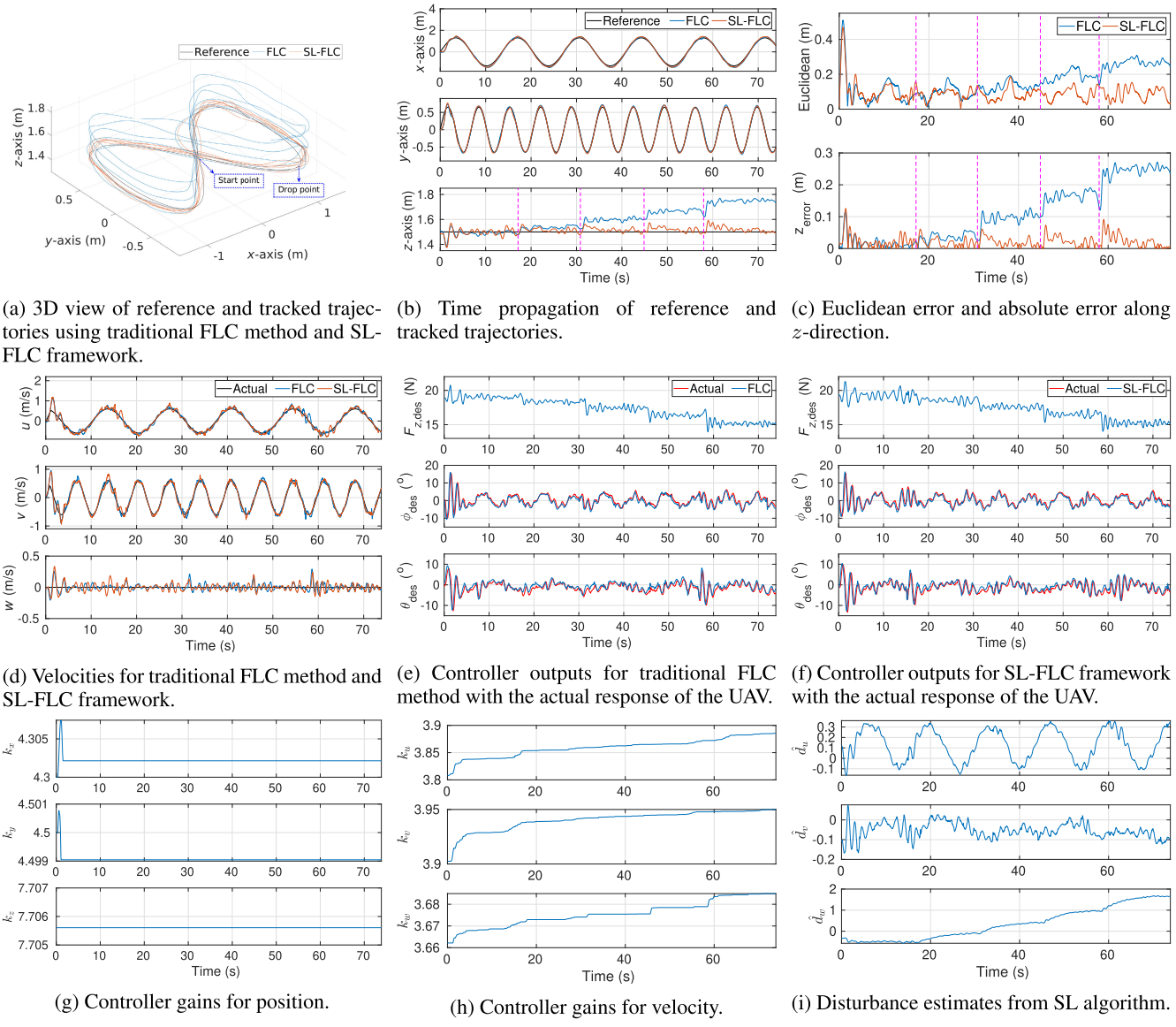
$$\hat{d}_u(0) = 0.15, \quad \hat{d}_v(0) = -0.02, \quad \hat{d}_u(0) = -0.3, \quad (51)$$

wherein, the above disturbance values are ones that resulted in an offset-free hover tracking of the UAV by the traditional FLC method. Additionally, the disturbance estimation learning rates for SL-FLC framework are chosen to be:

$$\alpha_{\hat{d}_u} = 0.2, \quad \alpha_{\hat{d}_v} = 0.2, \quad \alpha_{\hat{d}_w} = 0.25. \quad (52)$$

Position tracking performance for the traditional FLC method and the SL-FLC framework is shown together in Figs. 7a and 7b. As visualized, the tracking performance along the z-direction for the SL-FLC framework is comparatively much precise as compared to the traditional FLC method. This effect is in accordance with our anticipation as the SL algorithm appropriately modifies the control and disturbance parameters within the feedback linearization method to help compensate for the disturbance which is induced due to the sequential payload drops. On account of this modification, the plant-model uncertainties that arise at the dropping instant diminish with time which eventually results in a precise trajectory tracking by the SL-FLC framework. Apart from the qualitative comparison, the Euclidean error and the absolute error along z-direction (z_{error}) are selected as the quantitative comparison measures, which are depicted in Fig. 7c for both the controllers. It is evident from the figure that the z_{error} (also Euclidean error) accumulates with each payload drop for the FLC method without learning, whereas for the SL-FLC framework although the error rises at the dropping instant, the learning brings it back to its initial value. Furthermore, the traditional FLC method resulted in the mean values for Euclidean error and z_{error} of 0.1573m and 0.1103m, respectively, and the mean Euclidean error and z_{error} for the SL-FLC framework are 0.0923m and 0.0220m, respectively. This implies that the SL algorithm helped FLC scheme to reduce the Euclidean error by 6.5 cm and z_{error} by 8.83 cm, over the traditional FLC method.

The velocity tracking performance for both are presented in Fig. 7d. From the figure, it is seen that both the controllers resulted in a similar performance. Mostly, the responses are overlapping but a slight more oscillatory behavior is obtained with the SL-FLC framework. This comparatively more oscillatory response for the SL-FLC framework can be explained due to the transition (or leaning) phase. In addition, the control outputs for both the controllers are presented in Figs. 7e and 7f. These figures also illustrate a similar performance as achieved by both the controllers. Another point to take note in Figs. 7e and 7f is that the low-level controller of Pixhawk is well-tuned such that it is able to accurately follow the references from high-level FLC. Finally, the variation of the control gains for position and velocity within the SL-FLC framework are plotted in Figs. 7g and 7h, respectively.



(a) 3D view of reference and tracked trajectories using traditional FLC method and SL-FLC framework.

(b) Time propagation of reference and tracked trajectories.

(c) Euclidean error and absolute error along z-direction.

(d) Velocities for traditional FLC method and SL-FLC framework.

(e) Controller outputs for traditional FLC method with the actual response of the UAV.

(f) Controller outputs for SL-FLC framework with the actual response of the UAV.

(g) Controller gains for position.

(h) Controller gains for velocity.

(i) Disturbance estimates from SL algorithm.

FIGURE 7. Case Scenario I: trajectory tracking control performance of the tilt-rotor tricopter UAV with varying mass for the 8-shaped reference, where the vertical magenta lines represent the instants of payload drop.

While there is no substantial change for position gains (except an abrupt change in the beginning), there is a significant increase in the velocity gains, especially k_w, with each payload drop. Similarly, the disturbance estimates presented in Fig. 7i, get accurately estimated throughout the experiment such that the SL-FLC framework results in a precise trajectory tracking of the UAV.

Remark 7: It is to be noted that the effect of mass disturbance is not visible in controller gains for the position. The reason for this behavior is that the induced force disturbance is only acting along the dynamic force equations (given in (35)), which govern the velocity response of the system.

In order to illustrate the viability of the proposed SL-FLC framework is not only limited to small weight variations, additional statistical experiments are performed, wherein four different levels of weight drop – with each drop larger than the previous drop – are executed while the UAV is hovering at a stationary point ($\mathrm{x}_{\mathrm{des}}^{\mathrm{pos}} = [0, 0, 1.5]^T$). For each level of weight drop, the experiments are repeated ten times while the mean Euclidean error values for both the controllers are recorded. Thereafter, a box plot is prepared which is presented in Fig. 8. It is evident that the mean Euclidean error for the traditional FLC method increases with the rising weight drop level, whereas the mean Euclidean error rise for the SL-FLC remains comparably negligible. This again is in accordance with our expectation as the learning helps FLC scheme to adapt according to the changing environment even when the magnitude of the change is abruptly varying.

Remark 8: In Fig. 8, one may expect to see a constant mean Euclidean error with SL-FLC framework for all levels

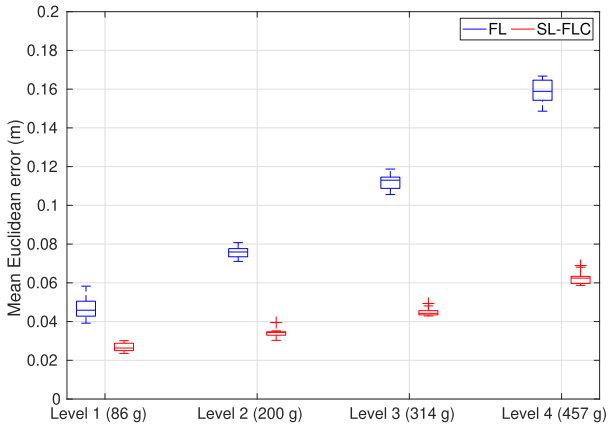


FIGURE 8. Mean Euclidean error for four levels of abrupt weight drops, where ten tests for each weight level are performed.

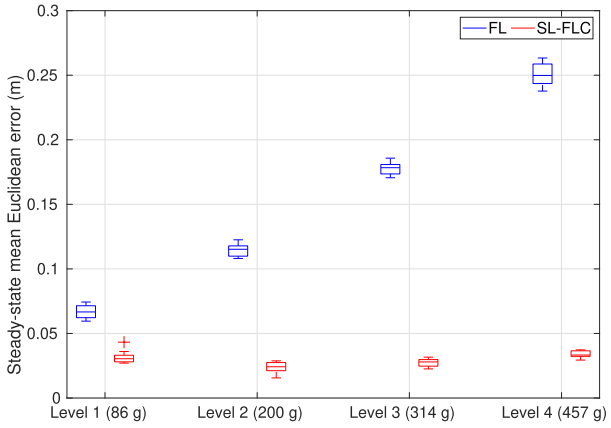


FIGURE 9. Steady-state mean Euclidean error for four levels of abrupt weight drops, where ten tests for each weight level are performed.

of weight drop. However, it is to be noted that rising mean Euclidean error for the SL-FLC framework is mainly due to the error that arises at the dropping instant. That is, for heavier weight drops, the instantaneous error at the dropping instant gets higher which subsequently diminishes over time. Therefore, the overall mean Euclidean error increases by a small amount with the increasing weight drop level. This argument is further validated with the help of the box plot for steady-state mean Euclidean error values presented in Fig. 9.

2) CASE SCENARIO II - GROUND EFFECT

In the second experiment, another disturbance type, the ground effect, is analyzed that is commonly encountered during takeoff, landing and ground proximity flying. Within this case scenario, the UAV (without any payload) tracks a circular trajectory of 1m radius along x- and y-directions that incorporates two different height (along z-direction) levels namely, 0.8m and 0.125m. The overall reference trajectory can be visualized from Fig. 10a. In the beginning, UAV starts to follow the circular trajectory at $z = 0.8$m (state vector $\mathrm{x}(0) = [0, 0, 0.8, 0, 0, 0]^T$) with a negligible influence of the ground effect. Subsequently, while following the reference, it gradually descends to $z = 0.125$m where the ground effect gets dominating. After completing about half-a-circle, it climbs back to $z = 0.8$m and maintains the height until the end of the circular path, and thereafter, repeats the process for another time.

In this scenario, both the controllers are tuned utilizing the same control gains as specified in (45) - (48), and the SL-FLC framework is designed with the controller learning rates that are given in (49) - (50). Additionally, for the SL-FLC framework, the initial value for disturbance vector is taken to be:

$$\hat{d}_u(0) = 0.15, \quad \hat{d}_v(0) = -0.02, \quad \hat{d}_u(0) = -0.3, \quad (53)$$

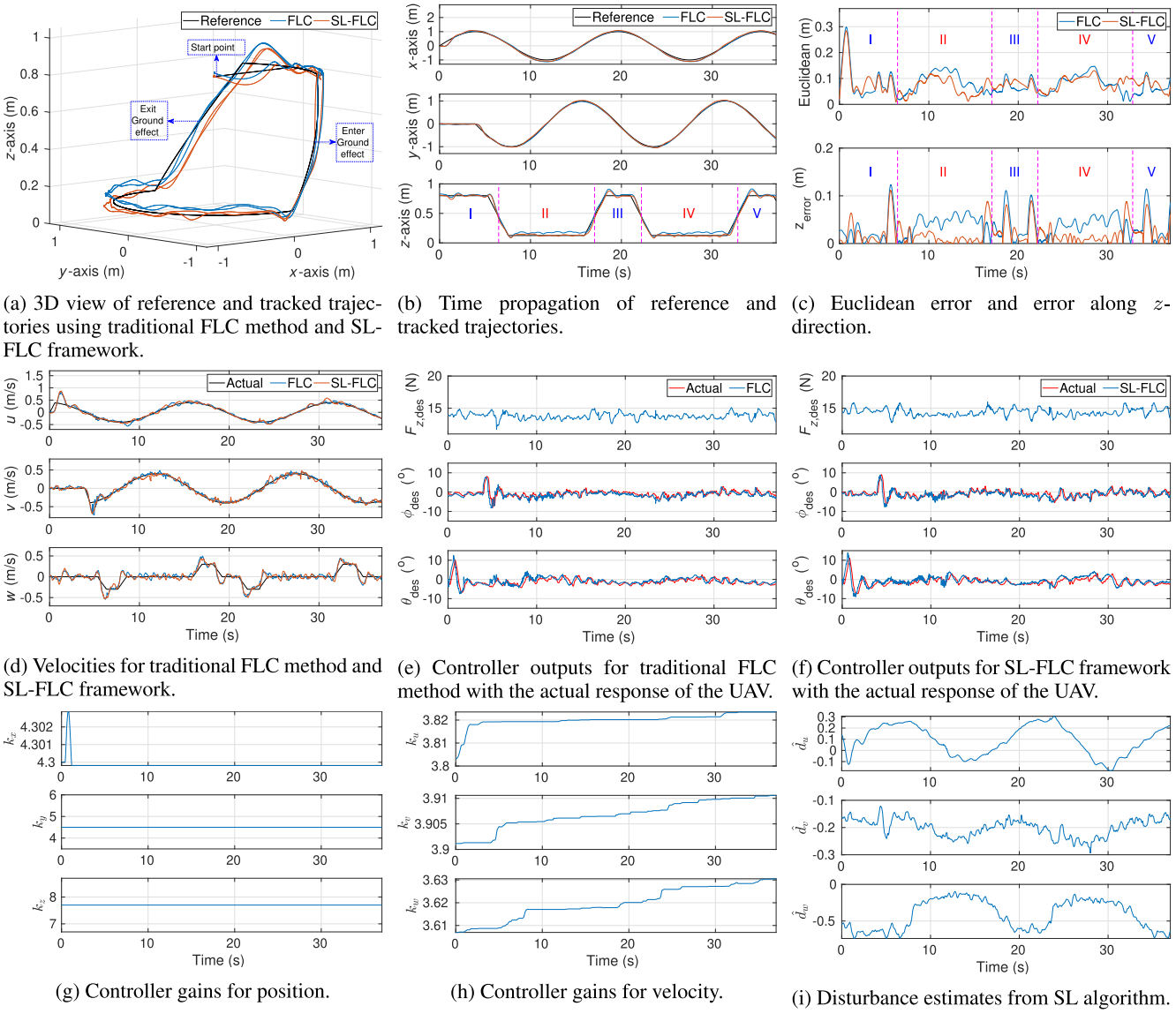
while the following learning rates are selected for the disturbance estimation:

$$\alpha_{\hat{d}_u} = 0.2, \quad \alpha_{\hat{d}_v} = 0.2, \quad \alpha_{\hat{d}_w} = 0.5. \quad (54)$$

Remark 9: In order to increase the ground clearance for the UAV, the foam attached to its landing skid is removed which further reduced its empty weight to $m = 1.410$ kg. This is the reason why a new disturbance vector is obtained above.

Remark 10: In terms of the disturbance learning rates, a higher value for $\alpha_{\hat{d}_w}$ is utilized as the disturbance due to ground effect is expected to vary at a higher rate in comparison to the previous case scenario.

The position tracking performance for both, traditional FLC method and SL-FLC framework, are presented in Figs. 10a and 10b. As anticipated, when the UAV is not exposed to ground effect both the controllers result in similar tracking performance, whereas the SL-FLC framework dominates the traditional FLC method under the ground effect influence, primarily in terms of z-directional tracking. The superiority of the former is best visualized from the Euclidean error and z_{error} plots presented in Fig. 10c. Particularly in the z_{error} plot, it is seen that the error for traditional FLC method is always more in comparison to the SL-FLC framework. Although there are instances (mainly during descending and climbing) when the errors for both the controllers get almost the same, these instances are mostly associated with the transition response of the SL-FLC framework. That is, during that phase, the learning is in progress for the SL-FLC framework due to which the tracking performance degrades a bit. Moreover, in the experiments with ground effect, the mean values for Euclidean error and z_{error} in traditional FLC method are 0.0803m and 0.0369m, respectively, while for SL-FLC framework the mean error values are 0.0783m and 0.0213m, respectively. Hence, the SL algorithm resulted in an improvement of 0.2 cm in terms of Euclidean error and 1.56 cm in terms of z_{error}. At the first look, these numbers may appear less. However, if one may look closely again at Fig. 10c, it is evident that within the ground effect zone (represented by areas II and IV), the maximum z_{error} for traditional FLC method reaches until 7 cm, whereas for the SL-FLC framework the z_{error} always stays below 2.5 cm. This difference could be very crucial in certain situations, for instance, search



(a) 3D view of reference and tracked trajectories using traditional FLC method and SL-FLC framework.

(b) Time propagation of reference and tracked trajectories.

(c) Euclidean error and error along z-direction.

(d) Velocities for traditional FLC method and SL-FLC framework.

(e) Controller outputs for traditional FLC method with the actual response of the UAV.

(f) Controller outputs for SL-FLC framework with the actual response of the UAV.

(g) Controller gains for position.

(h) Controller gains for velocity.

(i) Disturbance estimates from SL algorithm.

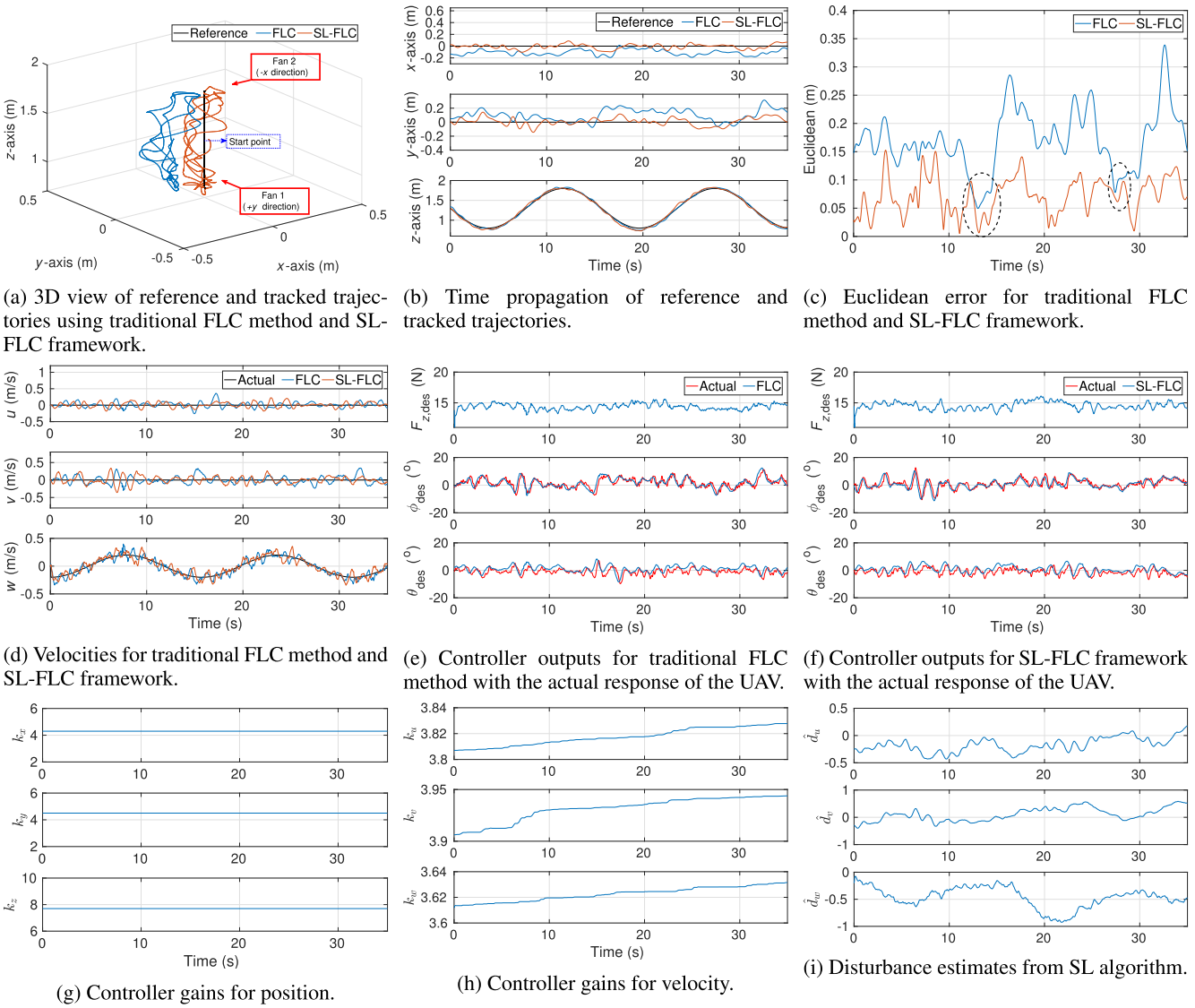
FIGURE 10. **Case Scenario II: trajectory tracking control performance of the tilt-rotor tricopter UAV in presence of ground effect for a circular reference. It is to be noted that in Figs. (b) and (c) the vertical magenta lines represent the transition points, wherein, the areas I, III, and V are the regions without ground effect influence while areas II and IV are with the influence of ground effect.**

and rescue, where a precise path tracking is required from the UAV while staying very close to the ground.

Even in this case scenario, the velocity tracking results for both the controllers, presented in Fig. 10d, are similar. Their plots are mostly overlapping but with a bit more oscillatory response from the SL-FLC framework. In terms of the control outputs, again a similar performance is observed by both, as also visualized from Figs. 10e and 10f. Furthermore, the variation of the position and velocity control gains are presented in Figs. 10g and 10h. Although the position control gains do not change much during the trajectory (same reason as Remark 7), the velocity control gains get updated whenever a disturbance is induced. It is to be noted that even though there is no explicit induction of the disturbance along x- and y-directions, the velocity control gains are also updated to achieve a precise tracking performance. Finally, the disturbance estimates from the SL algorithm are presented in Fig. 10i. They vary with the induced disturbance to result in an offset-free tracking. Moreover, from the plot of $\hat{d}_w$, it is deduced that whenever the UAV enters the ground effect zone, the magnitude of $\hat{d}_w$ increases and decreases for vice-versa. This is also per our intuition as the UAV is expected to experience a positive force along z-direction under the ground effect influence.

3) CASE SCENARIO III - WIND GUST

In the third experiment, another commonly experienced disturbance while flying outdoors, i.e, the wind gust, is examined. To mimic the outdoor wind in the motion capture laboratory, the wind gust disturbance is artificially induced with



(a) 3D view of reference and tracked trajectories using traditional FLC method and SL-FLC framework.

(b) Time propagation of reference and tracked trajectories.

(c) Euclidean error for traditional FLC method and SL-FLC framework.

(d) Velocities for traditional FLC method and SL-FLC framework.

(e) Controller outputs for traditional FLC method with the actual response of the UAV.

(f) Controller outputs for SL-FLC framework with the actual response of the UAV.

(g) Controller gains for position.

(h) Controller gains for velocity.

(i) Disturbance estimates from SL algorithm.

FIGURE 11. Case Scenario III: trajectory tracking control performance of the tilt-rotor tricopter UAV in presence of wind gust disturbance for a hover with varying z-position reference.

the help of two industrial fans that are placed perpendicular to each other at a distance of 2.2 m from the origin, as can be seen in Fig. 5. For this experiment, they produce wind with speeds in the range of 4.2-4.8 m/s (each fan) along $-x$- and $+y$-directions. To test the efficacy of the SL-FLC framework in compensating the wind gust disturbance, trajectory tracking performance of the UAV is analyzed for a hover with varying height (changing z-position) reference. At the start, the state of the UAV is $\mathrm{x}(0) = [0, 0, 1.3, 0, 0, 0]^T$, and then, the z-position reference varies sinusoidally with magnitude ± 0.5m. It is to be noted that in this experiment, wind gust disturbance is induced from the very beginning for both the controllers.

Remark 11: It is worth to mention that due to a changing z-position reference, the overall magnitude of the wind disturbance acting on the UAV varies with time. However, we would like to emphasize that the average change in the wind disturbance is much slower than the states e_1 and e_2. Consequently, our assumption $\dot{d} = 0$ is still valid.

For this case scenario also, both the controllers are tuned with the gains specified in (45) - (48), and the SL-FLC framework is designed with the controller learning rates that are given in (49) - (50). In addition, for the SL-FLC framework, the initial value for the disturbance vector is taken to be:

$$\hat{d}_u(0) = 0.15, \quad \hat{d}_v(0) = -0.02, \quad \hat{d}_u(0) = -0.3, \quad (55)$$

while the following learning rates are selected for the disturbance estimation:

$$\alpha_{\hat{d}_u} = 0.4, \quad \alpha_{\hat{d}_v} = 0.4, \quad \alpha_{\hat{d}_w} = 0.4, \quad (56)$$

which are obtained by utilizing the trial-and-error procedure.

Remark 12: During the tuning phase for SL-FLC framework, it is observed that for low values (less than 0.2) of disturbance learning rates, the disturbance estimation is unable to follow the variation within the induced wind gust disturbance due to which the overall control performance is poor. On the other hand, for the too high values (more than 0.5) of the disturbance learning rates, the disturbance estimation follows the trend of the variation of wind gust disturbance but with high-frequency oscillations. Hence, the values in between these extreme limits are utilized in the experiment.

The position tracking results of a hover with changing z-position reference for both, traditional FLC method and SL-FLC framework, with wind gust disturbance are presented in Figs. 11a and 11b. As anticipated, the SL algorithm helps FLC scheme (within SL-FLC framework) to predict and compensate for the acting wind gust disturbance. As a result, the UAV can closely follow the commanded trajectory. On the other hand, there lies a constant offset (along $-x$- and $+y$-directions) in tracking performance by the traditional FLC method, which is mainly because of the induced plant-model uncertainties. Additionally, superior tracking by the SL-FLC framework is even more explicit in Fig. 11c, where the Euclidean errors for both the controllers are plotted. As can be seen from the figure, the Euclidean error by SL-FLC framework stays substantially below the Euclidean error by traditional FLC method throughout the experiment. Moreover, the mean Euclidean error values for traditional FLC method and SL-FLC framework are 0.1713m and 0.0725m, respectively, which eventually illustrates an improvement of 9.88 cm due to the SL algorithm for this case scenario.

Remark 13: In Fig. 11c, one may notice two instances (marked by two black ellipses) when the error for both the controllers become almost the same. It is emphasized here that at these instants the UAV is at the maximum height, which is outside the fans' influence, as can also be seen in Fig. 11a.

The velocity tracking performance for both the controllers is presented in Fig. 11d. Yet again, nothing much is deducted from the figure as both the plots are almost overlapping throughout the trajectory. Next, the control outputs for traditional FLC method and SL-FLC framework are presented in Figs. 11e and 11f, respectively. In terms of controller gains for the SL-FLC framework, the position control gains are presented in Fig. 11g while the velocity control gains are shown in Fig. 11h. Although there is no change in the position control gains (same reason as Remark 7), the velocity control gains get updated throughout to realize an accurate tracking performance. In addition, the disturbance estimation performance by the SL algorithm, which is depicted in Fig. 11i, further facilitates a minimum-error trajectory tracking by the SL-FLC framework.

Remark 14: Although the wind disturbance is introduced along $-x$- and $+y$-directions, some of its influence is also experienced along z-direction (due to a non-stationary trajectory). As a result, a non-zero disturbance value is estimated along z-direction in Fig. 11i.

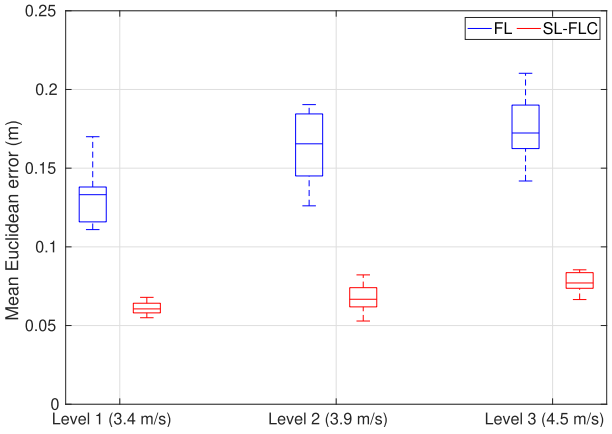


FIGURE 12. Mean Euclidean error for three wind speed levels, wherein ten tests for each speed level are performed.

Finally, in order to validate the claim that the SL algorithm helps to improve tracking performance in the presence of wind gust disturbance which is primarily stochastic, some statistical results are also obtained. That is, the tracking performance of each controller is tested for three different levels of fan speeds, whereby, each experiment is repeated ten times. From every single test, the mean Euclidean error values for both the controllers are recorded and later accumulated to obtain a box plot which is shown in Fig. 12. As visualized from the figure, the mean Euclidean error values for both the controllers increase with the rising fan speed level. However, the error rise for the SL-FLC framework is negligibly small in comparison to the traditional FLC method. This behavior is in accordance with our expectation, as the SL algorithm within the SL-FLC framework helps to compensate for the induced disturbance, irrespective of its magnitude.

Remark 15: One may note that the control gains in Fig. 11h get updated continuously even without much change in the disturbance value. The reason for this continuous update is a low-value selection for the dead-zone threshold (Remark 6), which is kept the same for all the experiments. Nevertheless, while the increase is less in comparison to the overall magnitude of the control gains, this continuous update can itself be avoided by selecting a higher value for the dead-zone threshold.

VI. CONCLUSION

In this paper, a simple learning strategy for uncertain nonlinear systems has been proposed. The efficacy of the proposed SL-FLC framework is first validated via a simulation study where the control problem of a second-order dynamical system is considered. It is shown that the SL-FLC framework can ensure the desired closed-loop error dynamics in the presence of modeling uncertainties and external disturbance. In addition to simulation, the performance of the SL-FLC framework is also experimentally validated for the position tracking of a

3D-printed tilt-rotor tricopter UAV. The tracking performance of both the controllers, traditional FLC method and SL-FLC framework, are analyzed for three different case scenarios, wherein the disturbances in the form of mass variation, ground effect, and wind gust are explicitly induced. Thanks to the SL algorithm, SL-FLC framework resulted in z-direction tracking improvements of 80.05% and 42.27% for the case scenarios I and II, respectively. In addition, on account of learning, the tracking improvement of 57.68% in terms of the Euclidean error is achieved for the case scenario III. Moreover, the additional statistical results presented for the case scenarios I and III illustrated consistent tracking improvements even with the increasing magnitude of disturbances. Overall, the presented simulation and experimental results demonstrated a substantially superior tracking performance by the SL-FLC framework over the traditional FLC method.

ACKNOWLEDGMENT

The authors would like to thank Mr. K. Singh for his support during the experiments and Prof. T. I. Fossen for some technical discussions about the proposed algorithm.

REFERENCES

[1] E. T. Alotaibi, S. S. Alqefari, and A. Koubaa, "LSAR: Multi–UAV collaboration for search and rescue missions," *IEEE Access*, vol. 7, pp. 55817–55832, 2019.

[2] J. Qi, D. Song, H. Shang, N. Wang, C. Hua, C. Wu, X. Qi, and J. Han, "Search and rescue rotary–wing UAV and its application to the lushan Ms 7.0 Earthquake," *J. Field Robot.*, vol. 33, no. 3, pp. 290–321, May 2016.

[3] Amazon. (2017). *Amazon Prime Air*. Accessed: Sep. 4, 2017. [Online]. Available: https://www.amazon.com/Amazon-Prime-Air/b?node=8037720011

[4] M. Mehndiratta, E. Kayacan, S. Patel, E. Kayacan, and G. Chowdhary, "Learning-based fast nonlinear model predictive control for custom-made 3D printed ground and aerial robots," in *Handbook of Model Predictive Control* (Control Engineering). Cham, Switzerland: Birkhäuse, 2019.

[5] M. Li, L. Zhen, S. Wang, W. Lv, and X. Qu, "Unmanned aerial vehicle scheduling problem for traffic monitoring," *Comput. Ind. Eng.*, vol. 122, pp. 15–23, Aug. 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0360835218302481

[6] L. Huang, H. Qu, and L. Zuo, "Multi–type UAVs cooperative task allocation under resource constraints," *IEEE Access*, vol. 6, pp. 17841–17850, 2018.

[7] A. Bircher, M. Kamel, K. Alexis, H. Oleynikova, and R. Siegwart, "Receding horizon 'next-best-view' planner for 3D exploration," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2016, pp. 1462–1468.

[8] J. A. Meda, "Estimation of complex systems with parametric uncertainties using a JSSF heuristically adjusted.," *IEEE Latin Amer. Trans.*, vol. 16, no. 2, pp. 350–357, Feb. 2018.

[9] J. A. Meda-Campana, "On the estimation and control of nonlinear systems with parametric uncertainties and noisy outputs," *IEEE Access*, vol. 6, pp. 31968–31973, 2018.

[10] C. Izaguirre-Espinosa, A.-J. Muñoz-Vázquez, A. Sanchez-Orta, V. Parra-Vega, and P. Castillo, "Contact force tracking of quadrotors based on robust attitude control," *Control Eng. Pract.*, vol. 78, pp. 89–96, Sep. 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0967066118301898

[11] M. Mammarella, E. Capello, H. Park, G. Guglieri, and M. Romano, "Tube-based robust model predictive control for spacecraft proximity operations in the presence of persistent disturbance," *Aerosp. Sci. Technol.*, vol. 77, pp. 585–594, Jun. 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1270963817321223

[12] P. R. Ambati and R. Padhi, "Robust auto-landing of fixed-wing UAVs using neuro-adaptive design," *Control Eng. Pract.*, vol. 60, pp. 218–232, Mar. 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0967066116300582

[13] D. Muniraj, M. C. Palframan, K. T. Guthrie, and M. Farhood, "Path-following control of small fixed-wing unmanned aircraft systems with H_∞ type performance," *Control Eng. Pract.*, vol. 67, pp. 76–91, Oct. 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0967066117301570

[14] B. Zhao, B. Xian, Y. Zhang, and X. Zhang, "Nonlinear robust adaptive tracking control of a quadrotor UAV via immersion and invariance methodology," *IEEE Trans. Ind. Electron.*, vol. 62, no. 5, pp. 2891–2902, May 2015.

[15] A. Mystkowski, "Implementation and investigation of a robust control algorithm for an unmanned micro-aerial vehicle," *Robot. Auto. Syst.*, vol. 62, no. 8, pp. 1187–1196, Aug. 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0921889014000682

[16] H. Liu, Y. Bai, G. Lu, and Y. Zhong, "Robust attitude control of uncertain quadrotors," *IET Control Theory Appl.*, vol. 7, no. 11, pp. 1583–1589, Jul. 2013.

[17] Z. Liu, X. Liu, J. Chen, and C. Fang, "Altitude control for variable load quadrotor via learning rate based robust sliding mode controller," *IEEE Access*, vol. 7, pp. 9736–9744, 2019.

[18] A.-R. Babaei, M. Malekzadeh, and D. Madhkhan, "Adaptive super-twisting sliding mode control of 6-DOF nonlinear and uncertain air vehicle," *Aerosp. Sci. Technol.*, vol. 84, pp. 361–374, Jan. 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1270963818312446

[19] L. Liu, J. Zhu, G. Tang, and W. Bao, "Diving guidance via feedback linearization and sliding mode control," *Aerosp. Sci. Technol.*, vol. 41, pp. 16–23, Feb. 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1270963814002594

[20] H. Ramirez-Rodriguez, V. Parra-Vega, A. Sanchez-Orta, and O. Garcia-Salazar, "Robust backstepping control based on integral sliding modes for tracking of quadrotors," *J. Intell. Robot. Syst.*, vol. 73, nos. 1–4, pp. 51–66, Jan. 2014.

[21] E. Kayacan, "Sliding mode control for systems with mismatched time-varying uncertainties via a self-learning disturbance observer," *Trans. Inst. Meas. Control*, vol. 41, no. 7, pp. 2039–2052, Apr. 2019.

[22] B. B. Kocer, T. Tjahjowidodo, and G. G. L. Seet, "Centralized predictive ceiling interaction control of quadrotor VTOL UAV," *Aerosp. Sci. Technol.*, vol. 76, pp. 455–465, May 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1270963817315572

[23] M. Mehndiratta and E. Kayacan, "Online learning-based receding horizon control of tilt-rotor tricopter: A cascade implementation," in *Proc. Annu. Amer. Control Conf. (ACC)*, Jun. 2018, pp. 1–6.

[24] E. Kayacan, Z. Zhang, and G. Chowdhary, "Embedded high precision control and corn stand counting algorithms for an ultra-compact 3D printed field robot," in *Proc. Robot., Sci. Syst.*, Pittsburgh, PA, USA, Jun. 2018.

[25] E. Kayacan, E. Kayacan, I.-M. Chen, H. Ramon, and W. Saeys, *On the Comparison of Model-Based and Model-Free Controllers in Guidance, Navigation and Control of Agricultural Vehicles*. Cham, Switzerland: Springer, 2018, pp. 49–73.

[26] T. Dierks and S. Jagannathan, "Output feedback control of a quadrotor UAV using neural networks," *IEEE Trans. Neural Netw.*, vol. 21, no. 1, pp. 50–66, Jan. 2010.

[27] C. Nicol, C. J. B. Macnab, and A. Ramirez-Serrano, "Robust neural network control of a quadrotor helicopter," in *Proc. Can. Conf. Elect. Comput. Eng.*, May 2008, pp. 1233–1238.

[28] C. Fu, W. Hong, H. Lu, L. Zhang, X. Guo, and Y. Tian, "Adaptive robust backstepping attitude control for a multi-rotor unmanned aerial vehicle with time-varying output constraints," *Aerosp. Sci. Technol.*, vol. 78, pp. 593–603, Jul. 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1270963818304292

[29] E. Kayacan and T. I. Fossen, "Feedback linearization control for systems with mismatched uncertainties via disturbance observers," *Asian J. Control*, vol. 21, no. 3, pp. 1064–1076, May 2019.

[30] J. D. J. Rubio, "Robust feedback linearization for nonlinear processes control," *ISA Trans.*, vol. 74, pp. 155–164, Mar. 2018.[Online]. Available: http://www.sciencedirect.com/science/article/pii/S001905781830017X

[31] S. Åžahin, "Learning feedback linearization using artificial neural networks," *Neural Process Lett*, vol. 44, no. 3, pp. 625–637, Dec. 2016, doi: 10.1007/s11063-015-9484-8.

[32] J. Umlauft, T. Beckers, M. Kimmel, and S. Hirche, "Feedback linearization using Gaussian processes," in *Proc. 2017 IEEE 56th Annu. Conf. Decis. Control (CDC)*, Dec. 2017, pp. 5249–5255.

[33] Z. Hou and S. Jin, *Model Free Adaptive Control: Theory and Applications*. Boca Raton, FL, USA: CRC Press, 2013.

[34] R. Chi, Z. Hou, B. Huang, and S. Jin, "A unified data-driven design framework of optimality-based generalized iterative learning control," *Comput. Chem. Eng.*, vol. 77, pp. 10–23, Jun. 2015.
[35] Y. Zhu and Z. Hou, "Data–driven MFAC for a class of discrete–time nonlinear systems with RBFNN," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 5, pp. 1013–1020, May 2014.
[36] Z. Hou and Y. Zhu, "Controller-dynamic-linearization-based model free adaptive control for discrete–time nonlinear systems," *IEEE Trans. Ind. Inf.*, vol. 9, no. 4, pp. 2301–2309, Nov. 2013.
[37] Z. Hou, R. Chi, and H. Gao, "An overview of dynamic-linearization-based data-driven control and applications," *IEEE Trans. Ind. Electron.*, vol. 64, no. 5, pp. 4076–4090, May 2017.
[38] J. O. Pedro, M. Dangor, O. A. Dahunsi, and M. M. Ali, "Intelligent feedback linearization control of nonlinear electrohydraulic suspension systems using particle swarm optimization," *Appl. Soft Comput.*, vol. 24, pp. 50–62, Nov. 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1568494614002403
[39] M. Mehndiratta, E. Kayacan, and E. Kayacan, "A simple learning strategy for feedback linearization control of aerial package delivery robot," in *Proc. IEEE Conf. Control Technol. Appl. (CCTA)*, Aug. 2018, pp. 361–367.
[40] W.-H. Chen, "Disturbance observer based control for nonlinear systems," *IEEE/ASME Trans. Mechatronics*, vol. 9, no. 4, pp. 706–710, Dec. 2004.
[41] W. Langson and A. Alleyne, "A stability result with application to nonlinear regulation: Theory and experiments," in *Proc. Amer. Control Conf.*, vol. 5, Jun. 1999, pp. 3051–3056.
[42] S. Bouabdallah, "Design and control of quadrotors with application to autonomous flying," Ph.D. dissertation, Dept. Sect. De Microtechnique, EPFL, Lausanne, Switzerland, 2007.

MOHIT MEHNDIRATTA received the B.Tech. degree in aerospace engineering from Amity University, India, in 2012, and the joint M.Sc. degree in aerospace engineering from the Technische Universität München (TUM), Germany, and from Nanyang Technological University (NTU), Singapore, in August 2015. He is currently pursuing the Ph.D. degree, by research, with the School of Mechanical and Aerospace Engineering, NTU. His research primarily focuses on optimization-based learning control of unmanned aerial vehicles incorporating some machine learning techniques, including Gaussian process regression and reinforcement learning.

ERKAN KAYACAN received the B.Sc. degree in mechanical engineering and the M.Sc. degree in system dynamics and control from Istanbul Technical University, Istanbul, Turkey, in 2008 and 2010, respectively, and the Ph.D. degree in mechatronics, biostatistics, and sensors from the University of Leuven (KU Leuven), Leuven, Belgium, in 2014. He is currently a Lecturer with the School of Mechanical and Mining Engineering, The University of Queensland (UQ), Australia. Prior to UQ, he was a Postdoctoral Researcher with the Delft Center for Systems and Control (DCSC), Delft University of Technology, The Netherlands, the Distributed Autonomous Systems Laboratory, University of Illinois at Urbana-Champaign, IL, USA, and the Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, MA, USA. His research interests include real-time optimization-based control and estimation methods, nonlinear control theory, learning algorithms, and machine learning with a heavy emphasis on applications to autonomous systems and field robotics. He was a recipient of the Best Systems Paper Award from Robotics: Science and Systems (RSS), in 2018.

MAHMUT REYHANOGLU received the Ph.D. degree in aerospace engineering from the University of Michigan, Ann Arbor, MI, USA, in 1992. He is currently the Glaxo Wellcome Distinguished Professor and Chair of Engineering and the Director of the UNC Asheville-NC State Joint Engineering Programs. He has published three edited books, has authored or coauthored six book chapters, and over 130 peer-reviewed journal/proceedings articles, which as of August 2018; he have resulted in around 6000 citations and an H-index of 28 according to Google Scholar. He served on the IEEE TRANSACTIONS ON AUTOMATIC CONTROL Editorial Board, as an Associate Editor, from 2001 to 2007, on the IEEE Control Systems Society CEB, as an Associate Editor, from 1996 to 2001, and on AIAA Guidance, Navigation, and Control Technical Committee, as a member, from 1999 to 2002. He also served as an international program committee member for several conferences. He is currently an Editor of *International Journal of Aerospace Engineering* and a Book Series Editor on *Nonlinear Systems*.

ERDAL KAYACAN received the B.Sc. degree in electrical engineering from Istanbul Technical University, Istanbul, Turkey, in 2003, the M.Sc. degree in systems and control engineering from Bogazici University, Istanbul, in 2006, and the Ph.D. degree in electrical and electronic engineering from Bogazici University, Istanbul. After finishing his Postdoctoral Research with KU Leuven at the Division of Mechatronics, Biostatistics and Sensors (MeBioS), in 2014, he worked at the School of Mechanical and Aerospace Engineering, Nanyang Technological University, as an Assistant Professor, for four years. He is currently doing his research at the Department of Engineering, Aarhus University, as an Associate Professor, and also the Director of the Artificial Intelligence in Robotics Laboratory (Air Laboratory). His research areas are computational intelligence methods, sliding mode control, model predictive control, mechatronics, and unmanned aerial vehicles.