IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received May 10, 2020, accepted June 25, 2020, date of publication July 1, 2020, date of current version July 16, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3006277

Real-Time System Identification Using Deep Learning for Linear Processes With Application to Unmanned Aerial Vehicles

ABDULLA AYYAD[1], (Member, IEEE), MOHAMAD CHEHADEH[1], (Member, IEEE), MOHAMMAD I. AWAD[1,2], (Member, IEEE), AND YAHYA ZWEIRI[1,3], (Member, IEEE)

[1]Khalifa University Center for Autonomous Robotic Systems, Khalifa University, Abu Dhabi, United Arab Emirates
[2]Healthcare Engineering Innovation Center (HEIC), Department of Biomedical Engineering, Khalifa University, Abu Dhabi, United Arab Emirates
[3]Faculty of Science, Engineering and Computing, Kingston University London, London SW15 3DW, U.K.

Corresponding author: Abdulla Ayyad (abdulla.ayyad@ku.ac.ae)

This work was supported by the Khalifa University of Science and Technology under Awards RC1-2018-KUCARS and CIRA-2020-082.

ABSTRACT System identification is a key discipline within the field of automation that deals with inferring mathematical models of dynamic systems based on input-output measurements. Conventional identification methods require extensive data generation and are thus not suitable for real-time applications. In this paper, a novel real-time approach for the parametric identification of linear systems using Deep Learning (DL) and the Modified Relay Feedback Test (MRFT) is proposed. The proposed approach requires only a single steady-state cycle of MRFT, and guarantees stability and performance in the identification and control phases. The MRFT output is passed to a trained DL model that identifies the underlying process parameters in milliseconds. A novel modification to the Softmax function is derived to better conform the DL model for the process identification task. Quadrotor Unmanned Aerial Vehicle (UAV) attitude and altitude dynamics were used in simulation and experimentation to verify the presented approach. Results show the effectiveness and real-time capabilities of the proposed approach, which outperforms the conventional Prediction Error Method in terms of accuracy, robustness to biases, computational efficiency and data requirements.

INDEX TERMS System identification, unmanned aerial vehicles, learning systems, sliding mode control, process control.

NOMENCLATURE

β	Phase switching parameter for the MRFT controller.
β_d	Phase switching parameter for the MRFT controller corresponding to φ_d.
θ	Spherical coordinates' azimuth angle.
τ	Overall time delay.
ϕ	Spherical coordinates' zenith angle.
ϕ_m	Phase margin of a given process.
φ	Phase angle for the excitation of a given process.
φ_d	Distinguishing phase for a set of dynamic systems.
Ω_o	Frequency of the steady-state MRFT oscillations.
$\aleph$	Measurement noise.
a_0	Amplitude of the steady-state MRFT oscillations.
a_i	Logit for the i'th neuron of the DL model's output layer.
$C(s)$	Feedback controller acting on a dynamic system.
$C^*(s)$	Optimal feedback controller for process G(s).
c_i	Parameter of the homogeneous tuning rules with index i.
D	Subspace of model parameters.
$\bar{D}$	Discretization of subspace D.
e	Error signal.
$G(s)$	Transfer function for a dynamic system.
$\hat{G}(s)$	Estimated transfer function for a dynamic system.
h	Relay amplitude for the MRFT controller.
K_{eq}	Lumped gain of dynamic system.
K_c	PID controller gain.
L	Loss function for training the DL model.

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

N	Number of measurement samples.
N_d	Describing function of a system.
$\vec{p}$	Vector representing model parameters.
pv	Process variable.
p_i	Softmax probability for the i'th neuro of the DL model's output layer.
Q	Number of unknown model parameters.
r	Reference signal.
r_0	Spherical coordinates' radial distance.
T_{body}	Time constant associated with body dynamics.
T_d	PID derivative time.
T_i	PID integral time.
T_{prop}	Time constant associated with propulsion dynamics.
t	Index of time.
u	Controller action.
u_0	Input bias.
W_p	Transfer function for a process under test.
X	A vector combining N uniformly sampled measurements of pv and u.

I. INTRODUCTION

Since the third industrial revolution, system identification has been a key element in the development of autonomous technologies in a wide set of industrial applications. Accurate knowledge of system dynamics enables the design of robust and high-performance systems for prediction, planning and control. Unmanned Aerial Vehicles (UAVs) are an example of an autonomous system that has seen diverse utilization in areas of agriculture, disaster relief, remote sensing, surveillance, etc. [1]–[5]. UAVs are often deployed in uncontrolled environments, and are hence required to adapt to dynamic conditions in real-time with minimal sacrifice to functionality and performance.

To meet the aforementioned requirements of autonomy, extensive research has been carried out to develop effective methods of system identification and adaptation. These methods are generally classified as parametric or non-parametric depending on the control requirements and design constraints. Parametric means are data-driven approaches where model parameters of a Process Under Test (PUT) are identified based on observation data. Such approaches include prediction error methods (PEM) [6]–[8], maximum likelihood (ML) methods [9], [10], least square (LS) methods [11], frequency response identification methods [12]–[14], and neural network-based methods [15]–[17]. Several studies in the literature applied these techniques to UAV operation with accurate identification results [18]–[23]. Nonetheless, these methods require extensive data generation and accurate selection of optimizer initial conditions, which demand human experience and cause susceptibility to data biases and overfitting. Furthermore, most of these methods are computationally expensive and not suitable for real-time applications; they are instead applied offline to process an abundance of previously collected flight or operational data [18]–[22].

On the other hand, non-parametric identification includes methods that rely on the partial knowledge of the PUT to tune a predefined controller structure. Such methods include, for example, the classical Ziegler-Nichols method [24], the Relay Feedback Test (RFT) [25], and the Modified Relay Feedback Test (MRFT) [26]. In all these methods, the knowledge of the PUT response to a single excited frequency is sufficient to tune controller parameters. It was shown in recent work [27] that a near-optimal controller for quadcopter attitude dynamics can be designed using MRFT based tuning rules. However, non-parametric methods in the literature are limited to PID tuning and do not provide full insight into the system dynamics as they cannot be used to obtain model parameters.

Recent advancements in the fields of Iterative Learning Control (ILC), Reinforcement Learning (RL) and Deep Learning (DL), and the growth of computational capabilities have given rise to new approaches of controller design and tuning [28]–[34]. These approaches have introduced advantages in regards to accuracy of models and controllers, adaptation time, and the ability to handle nonlinearities in the PUT; with the limiting requirement of abundant observation data. Similar to non-parametric tuning, these approaches do not generate explicit estimates of model parameters, but rather implicitly consider these parameters in controller design.

This paper bridges the gap between parametric and non-parametric methods and presents a novel methodology to infer accurate estimates of model parameters with the prime motivation of designing high-performance controllers online and in real-time. The novelty of the proposed methodology lies in utilizing self-excited oscillations (i.e. chattering) resulting from a sliding mode controller, which is the MRFT, to reveal distinguishing information about the PUT. The information revealed by MRFT is then fed to a DL classifier that selects model parameters that best represent the PUT. Fig. 1 illustrates the proposed comprehensive system identification approach. We show that this identification method can be performed in real-time such that a UAV adapts to changes to its own physical dynamics during a flight mission. The suggested online identification methodology is safe with guaranteed stability, and results in controllers with assessable levels of robustness and performance. The presented approach is mainly applied to Second-Order with Integrator Plus Time Delay (SOIPTD) linear systems, but is also applicable to other system models with an equal or lower number of model parameters. Due to its real-time capabilities, the proposed technique can handle static nonlinearities by applying the identification process in multiple operation modes (e.g. near ground hovering or drag dynamics caused by large translational speeds); to obtain locally linear descriptions of the system.

There are two inherent features of the proposed approach that highlight its advantages over other existing methods. First, the required amount of data needed for identification is minimal as it consists of a single excited frequency of the system. In contrast, other data-driven classical identification methods used in the literature require extensive data

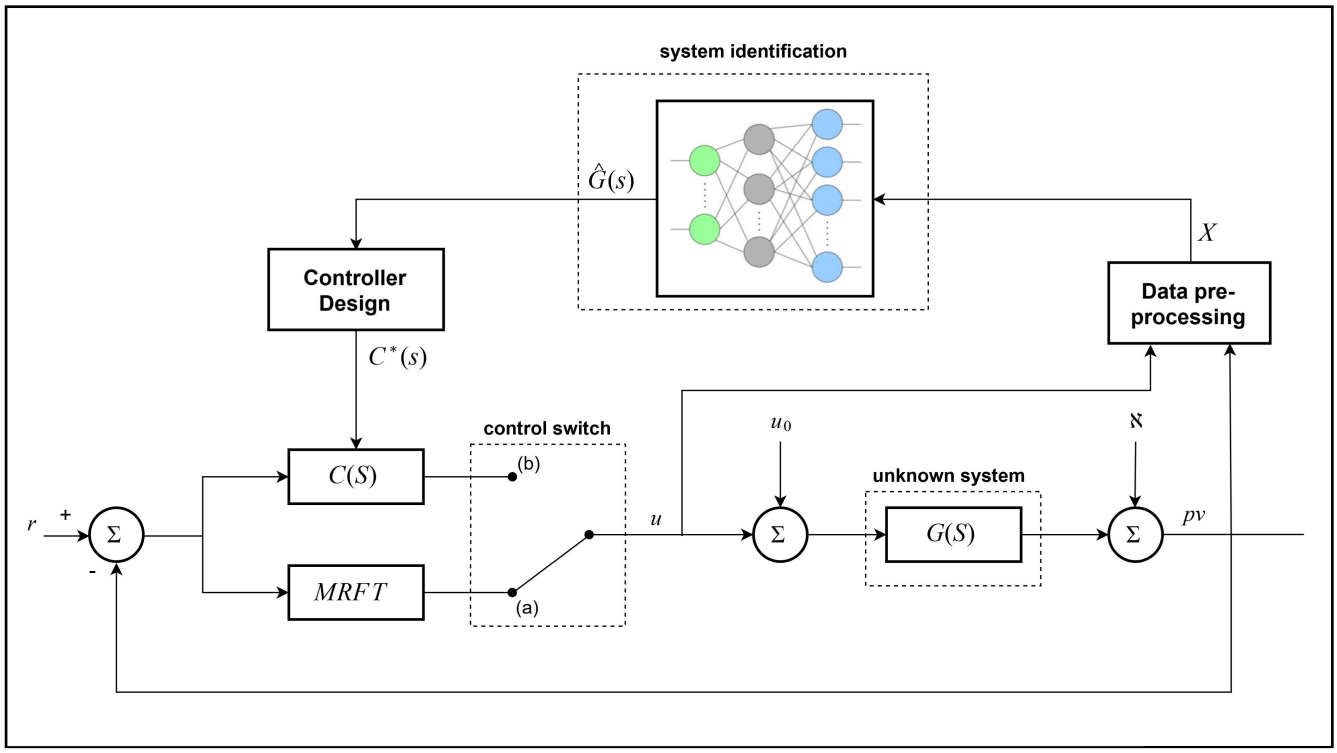

FIGURE 1. The proposed system identification scheme: (1) The control switch starts at position (a) where MRFT is used to excite an unknown plant $G(s)$ with stable oscillations. (2) These oscillations are pre-processed and forwarded to a DL classifier to identify the system parameters as $\hat{G}(s)$. (3) After the PUT is identified, a suitable controller $C^*(s)$ can be designed and applied to the plant by shifting the control switch to position (b).

generation [6]–[10], [13], [14]. Due to the reduction in data requirements, the proposed methodology does not require human experience for data generation, and the model fitting process is not prone to data bias. This makes the proposed method precise and accurate in identifying unknown model parameters. Other advantages of data reduction include minimizing computational requirements and shortening the period of the identification phase. In fact, we have found that the required computational time on modern commercial processors is in the order of milliseconds; and the identification phase lasts for a maximum of a few seconds (this depends on the PUT dynamics, e.g. mass).

The second inherent feature of the presented identification method is the guarantee of stability during the identification phase. This relieves the need for hand-tuned initial stabilizing controllers as opposed to classical identification methods, and makes the presented method less prone to estimation biases and non-optimality caused by the selection of initial control parameters (i.e. initial parameters for the optimizer decision variables). As stability is guaranteed, the PUT can be directly started in the identification phase as demonstrated in the results section; thus further minimizing the required operation time to estimate model parameters. Table 1 provides a qualitative comparison between the method suggested in this paper and other related work in literature. Additionally, the Results section provides a quantitative comparison between this paper's method and two other identification methods: PEM and non-parametric tuning based on MRFT.

The main contributions of this paper can be summarized as:

- We introduce a novel approach of parametric system identification with guaranteed stability, real-time capabilities, and minimal requirements of observation data.
- We optimize the identification phase to reveal distinctive information about the PUT by means of finding the distinguishing phase for a set dynamic systems.
- We present a discretization technique to address system identification as a classification problem by utilizing the concept of controller performance deterioration.
- We devise a modified formulation of the Softmax activation function that adds a meaningful discrepancy to the cost of misclassification, leading to faster and more accurate training of the DL model.

The remainder of this paper is organized as follows. The system identification problem is first formulated in Section II. Section III describes the comprehensive identification approach proposed in this paper. In section IV, simulation and experimental results for the suggested approach are presented, discussed and compared against PEM based system identification and the non-parametric tuning rules of [27]. Finally, Section V summarizes the findings of this paper and provides concluding remarks.

TABLE 1. Qualitative comparison with selected recent methods from literature addressing the problem of automatic controller tuning and adaptation for UAVs.

Method	Experimental data generation	Computational Resources	Stability	Comments
This work	Single steady-state oscillation at a specified phase	A few milliseconds with modern on-board processors	Guaranteed in data generation phase by Loeb criterion [35], [26]	Not investigated for lateral UAV motion yet. Provides PUT model parameters
Sim-to-(Multi)-Real using proximal policy optimization (PPO) RL [32]	Not required	Inference model is running in real-time	No stability guarantees	This work's problem statement is closest to the one presented in this paper, but with no guarantee of stability or explicit inference of model parameters
ILC [28]	Requires a lot of iterations	Computationally expensive	Stability guaranteed within ILC iterations	Feedforward compensation terms
Deep Model-Based Reinforcement Learning [29]	Requires a lot of experimental data	Computationally expensive	Not guaranteed	Early adaptation of model-based RL
Learning through Gaussian processes with Bayesian optimization [30]	Requires a lot of experimental data	Computationally expensive	Guaranteed stability during learning within model uncertainty margins	
Non-parametric tuning of UAV inner loops [27], [36]	Single steady-state oscillation at a specified phase	Negligible	Guaranteed stability during identification	Does not provide model parameters
Heuristics based tuning [37]	Requires a lot of experimentation data	Low computational resources	Not guaranteed and subject to selected optimization parameters constraints	

II. PROBLEM STATEMENT

Considering an LTI system $G(s)$ with known model structure and unknown set of bounded model parameters $\vec{p} \in (D \subset \mathbb{R}^Q)$. Let us also assume a feedback controller $C(s)$ that acts on $G(s)$. Given vector $X \in \mathbb{R}^{2N}$ that contains N uniformly sampled measurements of both the process variable pv and controller output u signals. Inner system states are considered to be unobservable. We wish to find the mapping $\Gamma : X \rightarrow \bar{D}$ where $\bar{D}$ is a discretization of the subspace D.
In this work, we limit the order of the LTI system to SOIPTD, which corresponds to multirotor attitude and altitude dynamics as presented in [27]:

$$G(s) = \frac{K_{eq} e^{-\tau s}}{s(T_{prop}s + 1)(T_{body}s + 1)} \tag{1}$$

The nonlinearity of the system is mainly exhibited by the change in the value of the parameter T_{body} as a function of velocity representing drag dynamics. The assumption that such drag coefficient can be considered constant works well in practice and was deliberately analyzed in [38], [39] where drag sources caused by air inflow, blade flapping, and body drag were investigated. Propulsion systems, consisting of electronic speed controllers ESCs and motors, are assumed to provide linear steady-state thrust response; and hence, K_{eq} can be considered constant. From bench propulsion system tests similar to the ones performed in [40], it can be concluded that T_{prop} is constant across the whole operating range. Additionally, network communication and processing delays are almost constant, permitting us to consider the time delay τ as a constant. The dynamics in (1) relate motor commands sent by the flight controller to the observed roll, pitch, or altitude. The considered attitude and altitude dynamics are subject to measurement noise $\aleph$ and forced bias u_0 caused by external disturbances (e.g. gravity) or sensor bias.

III. METHODOLOGY

A. FINDING THE DISTINGUISHING PHASE

The identification method presented in this paper builds on two propositions.

Proposition 1: There is a distinguishing phase φ_d at which the sustained self-excited oscillation characteristics can be used to identify the corresponding processes in $\bar{D}$.

Proposition 2: The distinguishing phase φ_d corresponds to the optimal tuning rules of [26]. As such, φ_d can be determined by the process of designing optimal non-parametric tuning rules as outlined in [26], [41].

For completeness, we summarize the steps to obtain optimal non-parametric tuning rules as follows [26], [27], [41]:

1) Select process model and the range of the normalized model parameters.
2) Discretize the selected range of model parameters to a finite set of dynamics processes.

3) Select tuning rule specifications based on gain margin or phase margin requirements.
4) Generate a locally optimal tuning rule for every process in the range.
5) Apply every locally optimal tuning rule to all other processes in the range. Note the performance deterioration for every process due to the application of the non-optimal tuning.
6) Select the tuning rule with the least deterioration in performance as the global optimum.

In our case, the model structure is SOIPTD; therefore, the vector $\vec{p}$ that contains the model parameters is defined as:

$$\vec{p} = \begin{bmatrix} T_{prop} & T_{body} & \tau \end{bmatrix}^T, \quad \vec{p} \in R^3 \tag{2}$$

and the model parameters range is selected to be

$$\begin{bmatrix} 0.015, & 0.2, & 0.0005 \end{bmatrix}^T \leq \vec{p} \leq \begin{bmatrix} 0.3, & 2, & 0.1 \end{bmatrix}^T \tag{3}$$

which includes a wide variety of multirotor UAV designs and sizes from racing quadrotors to larger multirotor UAVs having a takeoff weight of up to 50Kg. The selection of these parameter ranges was based both on experimental results and parameters from literature like the ones provided in [27], [39], [40], [42], in addition to modeling equations like those discussed in [27], [38], [39], [43]. The selection of parameter bounds can be safely expanded to include UAV designs beyond the specified ranges as they can be handled by the approach presented in this section.

Using the aforementioned method to generate optimal non-parametric tuning rules, we found φ_d to be $-46.89°$. In practice, a self-sustained oscillation with a specific phase can be excited using MRFT. MRFT is an algorithm that can be realized with the following equation [26]:

$$u_M(t) = \begin{cases} h & \text{if } e(t) \geq b_1 \text{ or } (e(t) > -b_2 \text{ and } u_M(t-) = h) \\ -h & \text{if } e(t) \leq -b_2 \text{ or } (e(t) < b_1 \text{ and } u_M(t-) = -h) \end{cases} \tag{4}$$

where $b_1 = -\beta e_{min}$ and $b_2 = \beta e_{max}$. $e_{max} > 0$ and $e_{min} < 0$ are respectively the last maximum and minimum values of the error signal after crossing the zero level; and $u_M(t-) = lim_{\epsilon \to 0^+} u_M(t - \epsilon)$ is the previous control signal. Prior to the start of MRFT the maximum and minimum error values are set as: $e_{max} = e_{min} = 0$. β is a constant parameter that dictates the phase of the excited oscillations as:

$$\varphi = \arcsin(\beta) \tag{5}$$

Using the Describing Function (DF) method, it could be shown that the MRFT achieves oscillations at a specified phase angle by satisfying the Harmonic Balance (HB) equation [44]:

$$N_d(a_0) W_p(j\Omega_0) = -1 \tag{6}$$

The DF of MRFT is presented in [26] as:

$$N_d(a_0) = \frac{4h}{\pi a_0}(\sqrt{1-\beta^2} - j\beta) \tag{7}$$

The DF method provides an approximate solution that is valid only if $W_p(s)$ has sufficient low pass filtering properties. It is worth mentioning that the MRFT control signal $u_M(t)$ has a phase lead relative to the error signal $e(t)$ in the case of $\beta < 0$, and lags in the case of $\beta > 0$. The MRFT DF intersects the Nyquist plot in the second quadrant for $\beta < 0$; while this intersection occurs in the third quadrant when $\beta > 0$. The Relay Feedback Test (RFT) [25] could be thought of as a special case of the MRFT algorithm where $\beta = 0$. For our case, the value of the MRFT parameter β that corresponds to $\varphi_d = -46.89°$ is $\beta_d = sin(\varphi_d) = -0.73$.

B. DISCRETIZATION OF SYSTEM PARAMETERS' SUBSPACE

System identification has been generally considered in the literature as a regression problem [45]–[47]. However, training a deep learning regression model can raise several instability and complexity concerns as suggested in [48], where a DL network was used for age prediction. To avoid these shortcomings with DL regression, the system identification conundrum in this study is formulated as a classification problem by discretizing the parameter space D into N unique sets of system parameters $\bar{D} = \{G_1, G_2, \ldots, G_N\}$. System identification hence becomes the problem of selecting a candidate set of process parameters G_i within $\bar{D}$ that best resembles the dynamics of the ground truth dynamic system G_{act}.

A trivial approach to obtain $\bar{D}$ would be discretizing D based on an equispaced distance of the model parameters $\vec{p}$. Assuming that the equi-space distance was small enough to represent all the pivotal processes, $\bar{D}$ would end up being an over-discretized representation of D. Adjacent processes of a given subspace of $\bar{D}$ would have similar frequency response characteristics while adjacent processes in another subspace of $\bar{D}$ would have vastly different frequency response characteristics. Thus, a trained classifier would be biased towards the subspace where adjacent processes have similar frequency response characteristics. Therefore, a meaningful criterion for discretization must be developed to ensure a balance between the distinguishability of these processes (i.e. in terms of their frequency response characteristics) and their accuracy in representing D. For this purpose, a joint cost function is introduced based on the concept of controller performance deterioration. To illustrate such joint cost function, let us consider $\{G_i(s), G_j(s)\} \in \bar{D}$; the joint cost associated with applying $C_i(s)$, which is the optimal controller of process $G_i(s)$, to the process $G_j(s)$ would be given by:

$$J_{ij} = \frac{J(C_i(s), G_j(s)) - J(C_j(s), G_j(s))}{J(C_j(s), G_j(s))} \times 100\% \tag{8}$$

where J is a cost function relating a controller $C(s)$ to a process $G(s)$ (i.e. IAE, ISE, etc.). The self-joint cost is defined by the case of $i = j$, where $J_{ij} = 0$. For the case where $i \neq j$, $J_{ij} > 0$ by definition. It must be noted that the joint cost

function is non-commutative, i.e. $J_{ij} \neq J_{ji}$. Therefore, $J_{(ij)} = max\{J_{ij}, J_{ji}\}$ is used as the discretization criteria to provide a performance guarantee among adjacent members of $\bar{D}$. In this paper, we have chosen ISE as a system performance index. ISE cost function is given by:

$$J_{ISE}(C, G) = \frac{1}{T_s} \int_0^{T_s} e(C, G)^2 dt \tag{9}$$

where in this paper we use $J := J_{ISE}$ for convenience.

For discretizing D, we choose a desired value of the joint cost between adjacent processes J^*. In this paper, we use $J^* = 10\%$ as it provides sufficient accuracy without requiring excessive simulation time (simulation time have a cubical relationship with the reciprocal of J^*). To find the process adjacent to a known one, G_i, we use an optimizer (Nelder-Mead simplex algorithm realized by "fminsearch" function in MATLAB® have been used) that takes a vector of model parameters $\vec{p}_j$ as the set of decision variables and uses $E = (J^* - J_{(ji)})^2$ as a cost function. We have found that discretizing D requires excessive simulation time (might take days to several weeks depending on the selected parameters range of D). For that we propose reducing the dimensions of the parameters space by transforming the describing subspace D from rectangular to spherical coordinates. The transformation is given by:

$$\begin{aligned} r_0 &= \sqrt{T_{prop}^2 + T_{body}^2 + \tau^2} \\ \theta &= \arctan\left(\frac{T_{body}}{T_{prop}}\right) \\ \phi &= \arccos\left(\frac{\tau}{r_0}\right) \end{aligned} \tag{10}$$

It is worth noting that the parameter r in (10) represents time scaling of process parameters $\vec{p}$ as in $s' = rs$. This allows us to introduce two properties of the spherical representation that will make the discretization process more efficient.

Property 1: For subsequent time scaling of a system $G_i(s)$ along the radial direction $G_j(\alpha s)$, $G_k(\alpha^2 s)$, the joint cost between successive scaled systems remain constant as in: $J_{ij} = J_{jk}$, $J_{ji} = J_{kj}$ for $\alpha \in \mathbb{R}_{>0}$.

Property 2: Considering two radially scaled systems: $G_i(s)$ and $G_j(\alpha_1\ s)$ with a joint cost J_{ij}, and another pair of radially scaled systems $G_k(s)$ and $G_l(\alpha_2\ s)$ with the same joint cost $J_{kl} = J_{ij}$; then $J^* = J_{(ik)} = J_{(jl)}$ for $\alpha_1, \alpha_2 \in \mathbb{R}_{>0}$.

Property 1 allows us to discretize a subsurface of a sphere S that we choose its radius to satisfy $r_0 = ||\vec{p}_{min}||$, where $\vec{p}_{min}$ corresponds to the minimum model parameters set in D. This effectively reduces the discretization problem by one dimension. Fig. 2 provides a three-dimensional illustration of D and S. To discretize S we set $J^* = 10\%$ and we proceed with the discretization by varying the values of ϕ and θ. To prevent excessive discretization and to increase robustness of controllers against varying model parameters and linearization assumptions, we impose phase margin constraints on controllers used to find the joint cost function $J_{(ij)}$. The phase margin constraint can be imposed using a set of

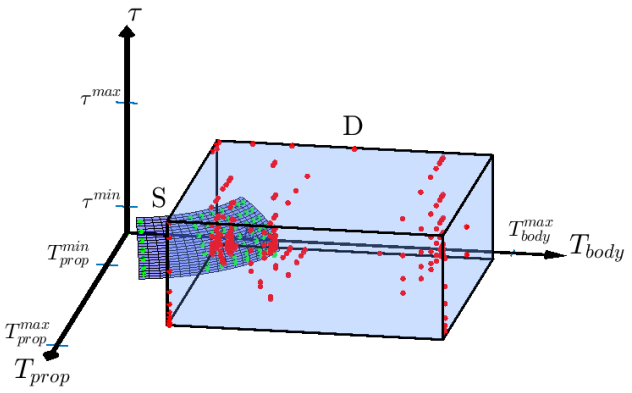

FIGURE 2. Showing a representation of D and S with discretized processes in $\bar{S}$ shown in green and discretized processes in $\bar{D}$ shown in red. The discretized processes in $\bar{D}$ are denser at parts of D where the ratio of $\frac{\tau}{T_{prop}}$ and $\frac{\tau}{T_{body}}$ are highest.

three equations. The first equation relates PID parameters with the PUT amplitude and frequency responses when a steady-state oscillation is excited at a certain phase [41]:

$$K_c = c_1 \frac{4h}{\pi a_0}, \quad T_i = c2 \frac{2\pi}{\Omega_0}, \quad T_d = c_3 \frac{2\pi}{\Omega_0} \tag{11}$$

c_1, c_2 and c_3 are related to the excitation phase φ characterized by the MRFT parameter β through the two following equations [41]:

$$\beta = sin(\phi_m + arctan(\frac{1}{2\pi c_2} - 2\pi c_3)) \tag{12}$$

and:

$$c_1 \sqrt{1 + (2\pi c_3 - \frac{1}{2\pi c_2})^2} = 1 \tag{13}$$

In this work, we choose $\phi_m = 20°$. A higher imposed value of the phase margin constraint would result in a lower number of discretized processes. A modified version of Nelder-Mead simplex algorithm that accepts constraints on optimization decision variables is used to realize (12) and (13). Then we proceed by finding $\bar{S}$ which is the set of the discretized processes in S. Once we have $\bar{S}$, we find the scaling parameter α for every process in $\bar{S}$ as proposed in property 1. Fig 3 illustrates these steps with the properties 1 and 2. Property 2 guarantees that all adjacent systems have a joint cost within J^*. Fig 2 shows the set of discretized processes.

For the parameters' range presented in (3), we found the size of $\bar{D}$ to be 208 processes. The discretized processes are denser at the parts of D where the ratio between the time delay τ and the other process time constants is the highest. Therefore, the identification and control of small UAVs with sensors and actuators that have high delays is found to be more challenging.

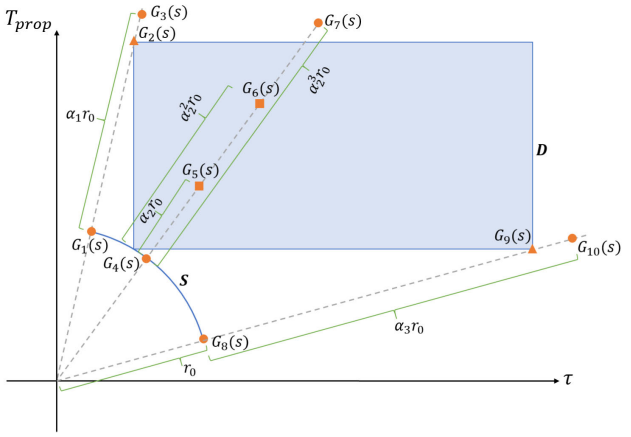

FIGURE 3. A projected side view showing parameters space D and the surface S with radius r_0. J^* is achieved by using the time scales $\alpha_1, \alpha_2, \alpha_3$. Note that $\{G_1(s), G_4(s), G_8(s)\} \in \bar{S}$ and $\{G_2(s), G_5(s), G_6(s), G_9(s)\} \in \bar{D}$. $J^* = J_{45} = J_{56}$ illustrates property 1. Property 2 is illustrated by $J^* = J_{14} = J_{35}$. $G_2(s)$ and $G_9(s)$ represent a scaled version of processes $G_1(s)$ and $G_8(s)$ respectively in D.

C. DEEP LEARNING MODEL DEVELOPMENT AND TRAINING

In this section, the process of developing and training a deep learning model for system parameter identification is discussed. The objective of the DL model in this study is to find the mapping $\Gamma : X \rightarrow \bar{D}$ as illustrated in Section II. The input X to the DL classifier is a uniformly time sampled vector concatenating the controller and plant response, while the output of the DL network is one of N sets of process parameters in $\bar{D}$.

Training data was generated based on members of $\bar{D}$. The MRFT response of each system within $\bar{D}$ was simulated multiple times according to the process diagram shown in Fig. 1. Measurement noise power $\aleph$ was randomly varied between different simulations to add a regularization effect and prevent over-fitting [49]. To further prompt robustness and generalization against varied testing conditions, simulations were carried out with varied values of forced input bias u_0, which introduces asymmetry to the MRFT controller output. The values of u_0 were limited to half the relay amplitude h of the MRFT controller as a reasonable bias magnitude in practical settings. Thirty simulations were performed for each candidate system in $\bar{D}$ to produce a training set of size 6240. Five additional simulations per system were carried out to generate 1040 samples to be used as a verification set. Both the training and the verification datasets incorporate the full range of process parameters in (3) and take into consideration practical conditions of bias and noise. As such, these datasets accurately represent the behavior of a wide variety of UAV systems in different experimental configurations.

The pre-processing steps undertaken to prepare the DL input data can be summarized as: sampling adjustment, cropping, zero-padding, amplitude normalization, and concatenation. Fig. 4 illustrates these pre-processing steps. The sampling period was fixed to be 1ms. The size of the input vector was set to be $X \in \mathbb{R}^{2\times 2260}$ to accommodate the response of the slowest system in $\bar{D}$ (i.e. a period of 2.26s).

Fig. 5 shows the structure of the developed deep learning model. The DL network consists of two hidden layers of size 3000 and 1000 respectively. This structure was chosen upon testing with several DL models of different depth and width, up to four layers and 10000 neurons, as it showed the best results with suitable computational performance on a single-core processor. Convolutional Neural Networks were also tested with no noticeable performance improvements. Rectified Linear Units (ReLU) were utilized as the activation function for both hidden layers due to its simplicity, reliability, and to avoid gradient vanishing [50]. Dropout is used after each layer for its regularization effect to avoid overfitting and prompt a noise rejection behavior [51]. Batch normalization is applied to the outputs of the hidden layers to accelerate training and to add a slight regularization effect [52]. The output layer consists of 208 neurons, one for each system in $\bar{D}$.

The Softmax activation function and the Cross-entropy loss function are among the most utilized combinations for training deep learning networks. For the case of system identification, a conventional application of this combination lacks in the sense that the cost of incorrect classifications is identical regardless of the corresponding error in parameter space. To undermine this shortcoming, we introduce a modified formulation of the Softmax function. The modified formulation utilizes the joint cost function presented in (8) to add a meaningful discrepancy to the cost of misclassification.

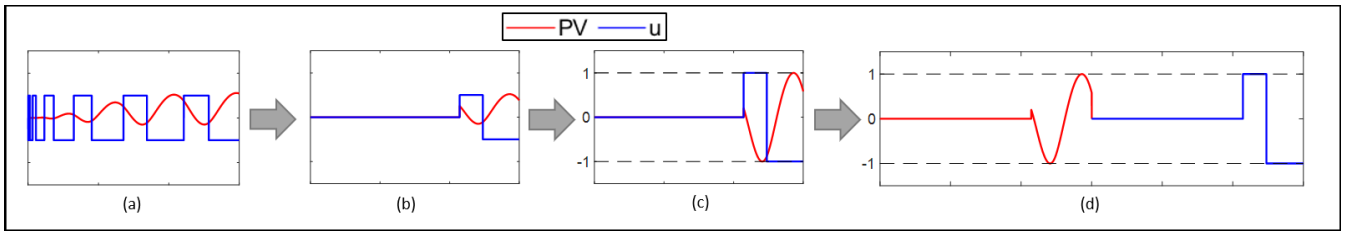

FIGURE 4. Pre-processing the DL classifier input vector: (a) The system's MRFT response is obtained and the sampling time is adjusted to be 1ms. (b) A single cycle of the steady-state oscillation is selected, zero-padding is applied elsewhere. (c) The response is zero-center and scaled to an amplitude of 1. (d) *PV* and *u* are concatenated to form a 1D vector.

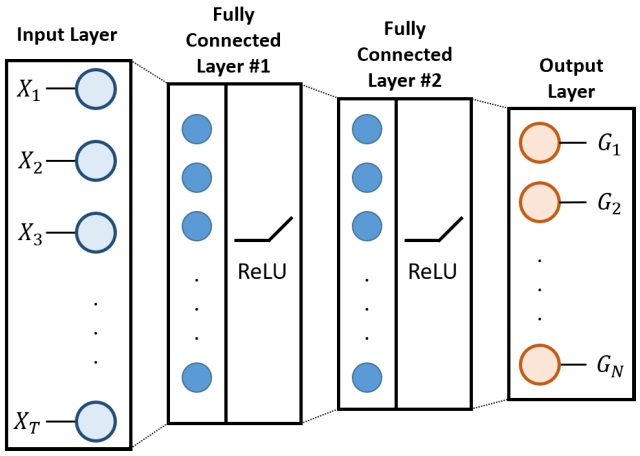

FIGURE 5. DL model architecture. The input vector length is 4520. Layer 1 contains 3000 neurons and Layer 2 contains 1000 neurons.

For the i^{th} logit a_i in the output layer corresponding to a process G_i, the modified Softmax probability p_i is introduced as:

$$p_i = \frac{e^{\gamma_{iT} \cdot a_i}}{\sum_{j=1}^{N} e^{\gamma_{jT} \cdot a_j}} \tag{14}$$

where T is the class corresponding to the ground truth system G_T, and $\gamma_{iT} = 1 + J_{iT}$ is a scaling factor.

As a loss function for training the DL network, the modified formulation in (14) is utilized alongside the cross-entropy function $L = -\sum_{i=1}^{N} y_i \log(p_i)$, where y is a one-hot encoded vector that indicates the ground truth class T. The partial derivative of L with respect to a_i is computed as:

$$\frac{\partial L}{\partial a_i} = \gamma_{iT} \times (p_i - y_i) \tag{15}$$

For an exact derivation of (15), readers can refer to Appendix A.

The DL network was trained using a Stochastic Gradient Descent approach with data shuffling for faster and more accurate convergence [53]. The ADAM optimization algorithm was utilized to further accelerate training and add robustness against noisy gradients as it was shown to outperform other adaptive learning rate techniques in [53], [54]. An automated search was conducted to find the most appropriate set of learning hyper-parameters that performed best on a held-out validation set. The final selection of hyper-parameters is shown in Table 2.

TABLE 2. Hyper-parameters for training the DL neural network.

Number of epochs	60
Batch size	1000
Base learning rate	0.005
Gradient decay factor	0.9
Squared gradient decay factor	0.999

TABLE 3. DL verification set results on simulation data.

	Modified Softmax	Conventional Softmax
Classification Accuracy	38.46%	30.38%
Average J_{pT}	0.30%	0.41%
Maximum J_{pT}	5.03%	13.29%
Minimum ϕ_m	13.73°	4.89°

Table 3 demonstrates the classifier's performance on the verification set utilizing both the modified and conventional Softmax formulations. Although the DL network was used as a classifier, classification accuracy is not a suitable measure of the performance as it does not reflect the error between the predicted system G_p and G_T. As such, controller performance deterioration J_{pT} is considered as a better and impartial evaluation criterion, especially as it was the basis for discretizing the system identification problem. The phase margin ϕ_m resulting from applying an optimal PD controller designed for G_p on the ground truth system G_T is also provided to evaluate the robustness of the proposed identification methodology for controller design purposes.

Results show that although classification accuracy is relatively low, the average joint cost J_{pT} is well below $J^* = 10\%$. The modified Softmax formulation results in a generally lower J_{pT} than the conventional Softmax function, particularly when comparing the maximum cost of misclassification. The modified formulation also provides a significantly larger margin for ϕ_m. These results highlight a promising performance of the developed DL scheme in system identification applications under varied measurement noise $\aleph$ and input bias u_0. A single inference run of the DL model on a single core of an i5-6300U processor requires 5 ms, which reflects the suitability of the developed deep learning framework for real-time identification applications.

D. IDENTIFICATION PROCEDURE

Once the distinguishing phase has been identified and the DL neural network has been trained, the real-time identification procedure shown in Fig. 1 can be applied to determine the model parameters of any system in D. First, MRFT is performed with β_d obtained from section III-A to excite the system with stable oscillations. System stability during the MRFT phase is guaranteed by the Loeb criterion [26]. Upon reaching steady-state, a single cycle of the system response and controller action is pre-processed as illustrated in Fig. 4. The trained DL classifier maps the pre-processed response to a set of model parameters $\vec{p}$ that better describes the PUT.

The identified process parameters $\vec{p}$ can subsequently be utilized to infer optimal controller parameters. To ensure that a controller is obtained in real-time, PD controllers are initially optimized offline for each system in $\bar{D}$ to form a lookup table of optimal controller parameters. During operation, the process parameters are identified in real-time and an optimal controller is selected from the pre-designed lookup-table. It is important to note that the proposed methodology

is not limited to PD controllers, but they are rather used as an example. Any other suitable controller design or structure can be implemented and optimized based on the identified model parameters.

IV. RESULTS

This section evaluates the effectiveness of the comprehensive system identification approach proposed in this paper. The testing approach follows the procedure explained in section III-D. Performance evaluation was performed using simulation data in addition to experimental tests on a UAV. Simulation results are compared against PEM as a well-established system identification method and the non-parametric tuning rules as an optimal controller design criteria.

A. SIMULATION RESULTS

Fifty different system parameter combinations were randomly sampled from the parameter space D to form a testing set $\bar{D}_{test}$. Unlike the DL verification set in III-C, these systems are not members of the discretized parameter space $\bar{D}$ and are thus better suited to evaluate the generalization performance of the inclusive system identification solution. Testing data is generated by simulating the MRFT response for each system in $\bar{D}_{test}$ under varied u_0 and $\aleph$. Data is then pre-processed and passed to the DL classifier to predict the parameters of a system G_p.

The controller performance deterioration J_{pT} is utilized to evaluate the accuracy of identification. The mean and maximum deterioration values for the testing set are shown in Table 4 using both the conventional Softmax function and the modified formulation in (14). Both the average and maximum J_{pT} are within $J^* = 10\%$, which validates the proposed means of system identification. The modified Softmax formulation outperforms the conventional one as it results in lower controller performance deterioration and provides a larger margin for ϕ_m. These results demonstrate the generalization capabilities of the system identification DL framework to accommodate the full parameter space D under different conditions of system bias and measurement noise.

TABLE 4. DL testing set D_{test} results on simulation data.

	Modified Softmax	Conventional Softmax
Average J_{pT}	0.53%	0.75%
Maximum J_{pT}	3.51%	6.38%
Minimum ϕ_m	15.53°	12.15°

B. COMPARISON WITH THE PREDICTION ERROR METHOD

The system identification performance of the proposed method was benchmarked against PEM using the same testing set $\bar{D}_{test}$. PEM was implemented using Matlab's System Identification Toolbox [55] with Sequential Quadratic Programming as the search algorithm. To generate input/output estimation data for PEM, the closed-loop response of each system in $\bar{D}_{test}$ was simulated under varied u_0 and $\aleph$. MRFT was chosen as the closed-loop controller is due to its guarantee of stability for all systems in D. Accordingly, two different sets of estimation data were simulated for each PUT in order to assess PEM's performance against the amount of observation data. The first set (Estimation Set-I) consists of a single cycle of the MRFT response with the distinguishing phase $\beta_d = -0.73$ of section III-A. Fig. 6-a shows a sample PEM estimation data of the first set, which is on par with the requirements of the DL approach proposed in this paper. The second set (Estimation Set-II) consists of 20 seconds of the simulated MRFT response with the β parameter continuously swept from $\beta_{max} = -0.1$ to $\beta_{min} = -0.9$; resulting in oscillations of different magnitude and frequency as shown in Fig. 6-b. Finally, the most sensitive system in D for parameters variation which is $G_{initial} =: \{T_{prop} = 0.015, T_{body} = 0.2, \tau = 0.1\}$ was selected as the initial guess for PEM predictions unless explicitly stated otherwise.

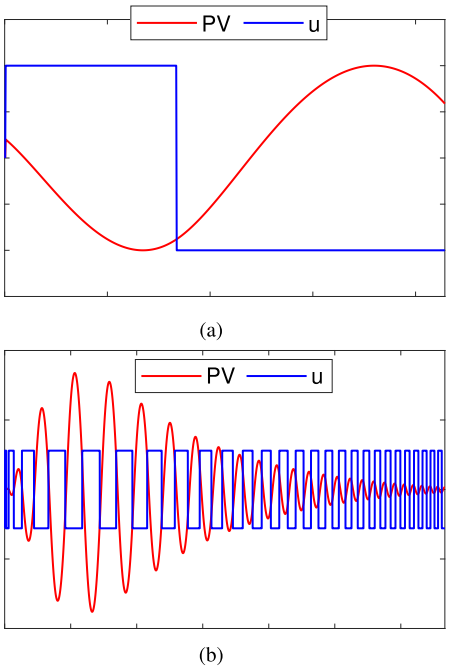

FIGURE 6. Sample of PEM input/output estimation data (a) Estimation Set-I with a single cycle of MRFT response (b) Estimation Set-II with multiple cycles of MRFT response with varied β.

Table 5 compares PEM identification results against our DL approach in terms of controller performance deterioration J_{pT} and computation time per inference. Unstable predictions are defined as those that result in J_{pT} which grows to infinity with time (note that a condition to handle steady-state errors is used). PEM results on Estimation Set-I show that unlike the deep learning approach, PEM fails to generate reliable predictions of system parameters from just a single cycle of the MRFT response. Increasing the amount of observation data enhances the prediction accuracy of PEM as indicated

TABLE 5. Comparison of system identification performance on simulation data for testing set $\bar{D}_{test}$.

	PEM		DL
	Estimation Set-I	Estimation Set-II	
Average J_{pT}	10.82%*	4.41%*	0.52%
Maximum J_{pT}	unstable	unstable	3.51%
Number of unstable predictions	5	1	0
Computation time per inference (s)	1.8936	4.8740	0.005

*: Excluding unstable predictions.

by results on Estimation Set-II, but it requires significantly larger processing time and still does not satisfy the maximum deterioration target of $J^* = 10\%$.

By examining the cases where PEM fails to produce accurate estimations, two main factors affecting PEM performance were identified. The first of which is PEM's requirement of a good initial guess. Table 6 shows PEM predictions for a system $G_T =: \{T_{prop} = 0.02, T_{body} = 0.3, \tau = 0.001\}$ with different initial guesses. PEM predictions differ significantly with the initial guess and do not consistently converge to a suitable solution. By contrast, our proposed approach does not require an initial guess or prior knowledge of the PUT to generate appropriate system identification results.

TABLE 6. PEM system identification results with different initial guesses for a simulated process $G_T =: \{T_{prop} = 0.02, T_{body} = 0.3, \tau = 0.001\}$.

Initial Guess	Controller Performance Deterioration J_{pT}	
$G_p =: \{T_{prop}, T_{body}, \tau\}$	Estimation Set-I	Estimation Set-II
$\{0.015, 0.2, 5 \times 10^{-4}\}$	0.0%	6.7%
$\{0.015, 0.2, 0.1\}$	unstable	21.2%
$\{0.3, 2, 0.0005\}$	unstable	unstable
$\{0.3, 2, 0.1\}$	unstable	26.0%
$\{0.15, 1, 0.05\}$	unstable	23.9%

The second factor disturbing PEM predictions is the input bias u_0. Table 7 compares identification results for a system $G_T =: \{T_{prop} = 0.02, T_{body} = 0.3, \tau = 0.001\}$ under different values of input bias using PEM and the suggested deep learning method. Referring to the results in Table 7, PEM predictions worsen as input bias get larger; by contrast, no significant differences are observed using our deep learning system identification method.

C. COMPARISON WITH NON-PARAMETRIC TUNING RULES

In order to assess the significance of the deep learning system identification solution for controller synthesis, it was benchmarked against the non-parametric tuning rules of [27] as it has comparable data and computational time requirements. These tuning rules were used to infer optimal PD controllers for each system in $\bar{D}_{test}$. Using these rules, an average performance deterioration of 1.67% and a maximum of 13.29% were observed on the testing set. By comparison, the approach presented in this paper reduces these deterioration values to a third as indicated in Table 4. These performance improvements would be difficult to observe in practice as both approaches result in a low performance deterioration value.

The main advantage the work presented in this paper holds over the non-parametric tuning rules lies in its scalability to various controller structures and system models. The proposed approach identifies the dynamic model parameters, which enables the design of a wide set of controllers fitting to specific practical and performance requirements; as opposed to the PID structure limitation of the tuning rules. Accurate knowledge of the model parameters can further be utilized in designing other sub-systems such as: trajectory generation, state estimation, or multi-loop cascaded controllers. The presented solution can be extended to any parametric system identification or controller tuning problem with minimal modifications to the approach given the constraint on the number of unknown parameters. In contrast, adapting the tuning rules for different model structures would require extensive theoretical adjustments and analysis.

TABLE 7. Comparison of system identification performance on simulation data for process $G_T =: \{T_{prop} = 0.02, T_{body} = 0.3, \tau = 0.001\}$ with varied input bias.

Input Bias u_0	Controller Performance Deterioration J_{pT}		
	PEM		DL
	Estimation Set-I	Estimation Set-II	Identification
0.0	0.0%	0.0%	0.0%
$-0.1 \times h$	0.0%	0.0%	0.0%
$-0.2 \times h$	0.0%	11.13%	0.0%
$-0.3 \times h$	0.0%	15.95%	0.0%
$-0.4 \times h$	7.49%	16.09%	0.0%

D. EXPERIMENTAL RESULTS

The approach presented in this paper was implemented to independently identify the altitude and attitude dynamics of a UAV. We utilized the Quanser QDrone [56] as the testing platform for our experiments. The physical parameters and specifications of the Qdrone are provided in Table 8. The onboard IMU data was fused with Optitrack's Prime 17W motion capture system [57] to estimate the pose of the drone. The procedure summarized in section III-D is applied to a single control loop of the QDrone to identify its underlying process parameters and optimal PD controller.

TABLE 8. QDrone specifications.

Dimensions (cm)	$40 \times 40 \times 15$
Mass (kg)	1.21
Moments of Inertia (kgm^2)	$\{J_{xx}, J_{yy}, J_{zz}\} = \{0.010, 0.0082, 0.015\}$
Processor	Intel Aero Compute Board – Intel Atom x7-Z8750

One advantage of our approach is the guarantee of stability during the identification phase by the MRFT controller [26]. Therefore, the identification procedure can be safely carried out without the need for prior knowledge of the underlying system dynamics. To illustrate this feature, identification of the altitude dynamics was done with the UAV starting from the ground with no PD controller. A generalized controller consisting of the summation of an MRFT controller and an integrator was used to elevate the UAV and excite stable oscillations around a predefined set-point. The objective of the integrator action is to counter the gravitational force. The following equation illustrates the quadrotor takeoff controller:

$$u_i(t) = \begin{cases} k_i \int (z_{ref} - z)dt & \text{if } z < z_{ref} \text{ or } \dot{z} < \dot{z}_{max} \\ u_i(t-) & \text{otherwise} \end{cases} \quad (16)$$

where k_i is a constant gain, z and $\dot{z}$ are the altitude and altitude change rate respectively, z_{ref} is the set point for altitude, $\dot{z}_{max}$ is the maximum allowed altitude rate, and $u_i(t-)$ is the previous controller output.

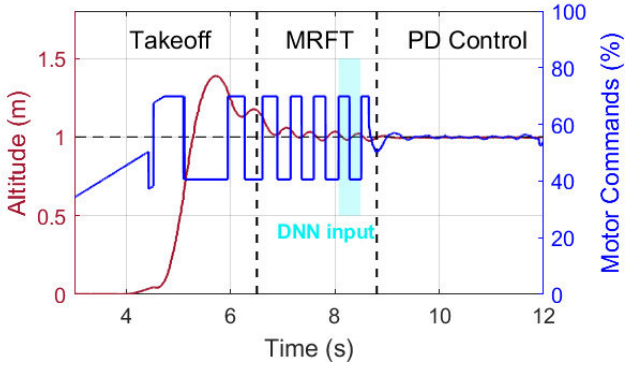

FIGURE 7. UAV experimental results for the altitude control loop. In the takeoff phase the algorithm presented in (16) was used. Note that MRFT takes a few oscillations to reach steady-state. A single oscillation at steady-state is selected as the input to the trained DL classifier.

Once a steady-state oscillation is acquired, the identification scheme takes place using the pre-trained DL classifier. Based on the identified process parameters, an optimal PD controller is inferred and applied to the plant. Fig. 7 shows the altitude and the controller action during the end-to-end identification and control process. The proposed take-off method was capable of stably lifting the UAV while simultaneously exciting oscillations. The DL network then identified the process parameters as $G_h =: \{T_{prop} = 0.0321, T_{body} = 1.6886, \tau = 0.0237\}$. Accordingly, the ISE optimal controller parameters were selected as $C_h^* =: \{K_p = 59.0220, K_d = 9.0356\}$. The controlled system response demonstrates a stable and smooth performance. In the absence of a ground truth system, this favorable controller performance indicates the effectiveness of the presented identification technique in targeting realistic control applications.

The identification experiment for the altitude dynamics was repeated with a payload of 400g attached to the drone; which corresponds to a 30% increase in the drone's mass. The optimal PD controller parameters identified under the increase in mass were $\{K_p = 69.9732, K_d = 11.0002\}$, which shows reasonable inflation over the nominal parameters in C_h^*.

To assess the precision of the proposed identification scheme, the DL framework was tested with multiple cycles of the MRFT response as indicated in Fig. 8. Each cycle serves as an independent input to the DL model, which predicts system parameters accordingly. Table 9 shows the cross-performance deterioration matrix among the multiple identified systems from multiple steady-state oscillations corresponding to a single MRFT run. The maximum joint cost observed was 1.97% despite noticeable noise and variations among subsequent cycles of the response, which illustrates the precision and noise rejection capability of the trained DL classifier.

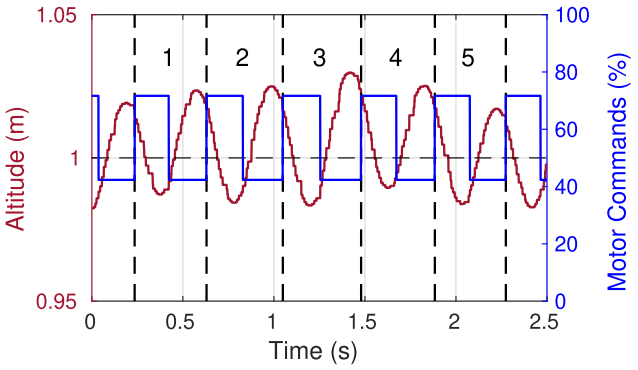

FIGURE 8. Multiple cycles of the experimental MRFT response for the UAV height control loop.

TABLE 9. Cross performance deterioration matrix showing controller performance deterioration J_{ij} for DL system identification of different cycles of the experimental MRFT response in Fig. 8.

cycle i \ cycle j	1	2	3	4	5
1	-	0.19%	0.0%	0.9%	0.0%
2	0.77%	-	0.0%	1.58%	0.01%
3	1.25%	0.27%	-	**1.97%**	0.35%
4	0.0%	1.03%	0.45%	-	0.0%
5	0.14%	0.64%	0.23%	0.95%	-

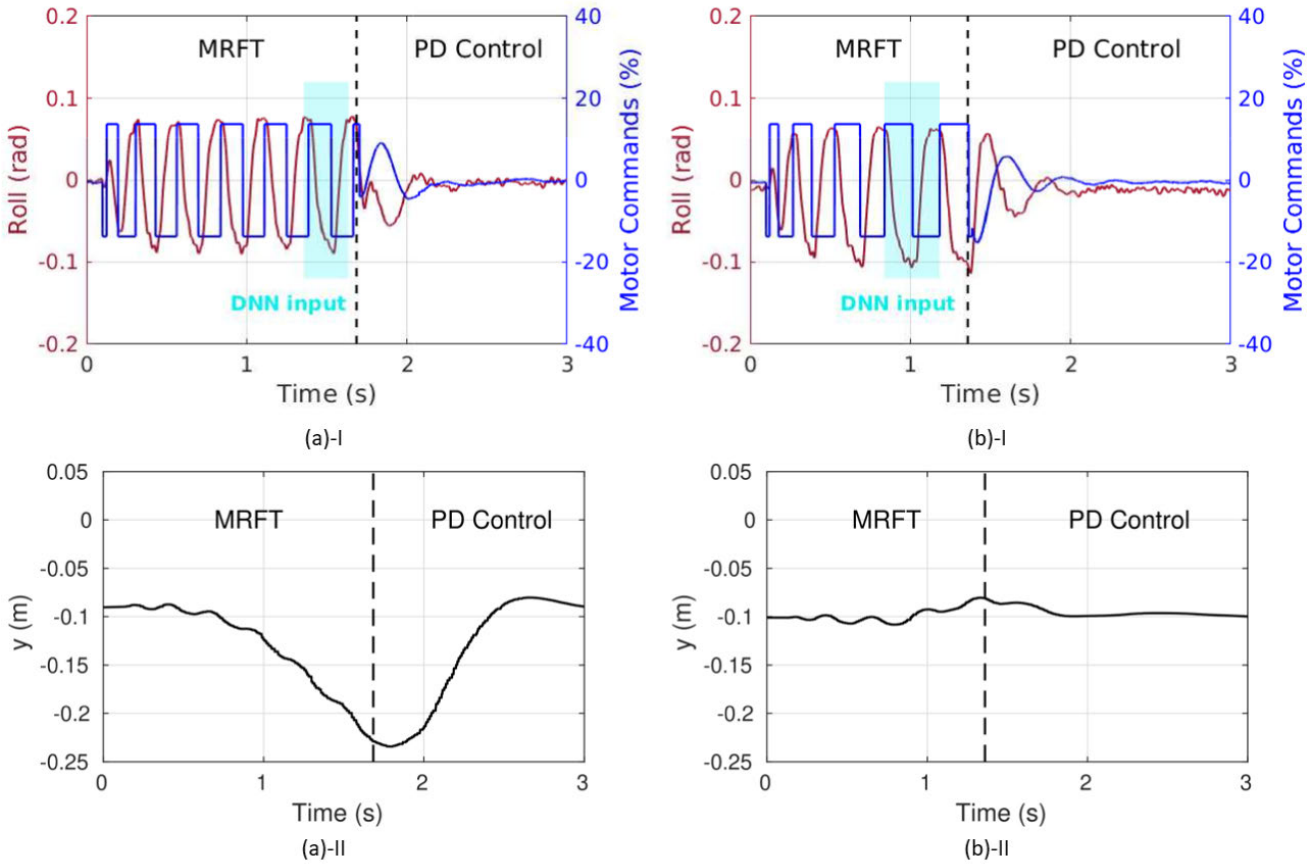

FIGURE 9. UAV experimental results for the roll control loop: (a) Without bias compensation for the outer-loop, inherent system biases cause a drift in the lateral position of the drone. (b) With bias compensation, lateral drift is minimized. Proper system and controller parameters for the roll dynamics were identified in both scenarios.

The identification process was further applied to the roll dynamics of the QDrone. The closed-loop system starts with a sub-optimal stabilizing controller $C(s)$ that enables that UAV to safely take-off; the system identification procedure is then applied and an optimal controller $C^*(s)$ is inferred accordingly. It must be noted that the outer-loop position controller is disabled during the identification phase to maintain a constant reference value for the MRFT controller. As a result, inherent system biases can cause a lateral drift in the drone's position. To undermine the translational drift, a bias compensation technique was implemented where the output of the roll channel's sub-optimal controller $C(s)$ is filtered and stored prior to initiating the identification procedure. The filtered output is then used to offset the MRFT controller during the system identification process.

Fig. 9 shows the results of the roll channel identification experiments during the subsequent identification and control phases with and without compensating for biases. When applying the bias compensation technique, the DL classifier identified the process parameters as $G_r =: \{T_{prop} = 0.02, T_{body} = 1.6889, \tau = 0.0121\}$, and the corresponding optimal PD controller parameters were $C_r^* =: \{K_p = 1.1000, K_d = 0.0985\}$. The PD control stage shows a positive response of the attitude dynamics to the designed controller and no substantial drift in the drone's position was observed. For the experiment where bias is not compensated for, the identified optimal controller parameters were $C_{r^-}^* =: \{K_p = 1.1082, K_d = 0.1013\}$. $C_{r^-}^*$ is almost identical to C_r^*, and would cause a practically negligible performance deterioration of less than 0.2% when applied to G_r. This demonstrates the robustness of the developed identification framework under different testing conditions, which can widen its scope of application to a broad range of practical control problems.

E. SUPPLEMENTARY MATERIAL

For better visualization of the experimental results, readers are encouraged to refer to the supplemental video in [58], which better highlights the performance of the proposed identification and control framework when applied to altitude and attitude loops of a quadrotor UAV. In addition to showing the capability of devising high-performance controllers in real-time, the video demonstrates the robustness of these controllers to several artificial disturbances applied during operation time. These disturbances mimic practical conditions a UAV might encounter during a mission flight;

and include weight changes, external nudges, and induced wind speeds up to 5 m/s. The UAV sustains stability and performance despite the extreme conditions; which highlights the robust capabilities of the presented methodology and its applicability to practical identification and control problems.

V. CONCLUSION

This paper introduced a novel approach for linear systems identification of a degree up to SOIPTD. The proposed method combines MRFT and DL to obtain a distinctive frequency response from an unknown plant, and map this response to a set of process parameters. The design of the identification procedure is presented as finding the distinguishing phase for a family of processes with the same model structure. System identification is then approached as a classification problem by using the principle of controller performance deterioration. Subsequently, a DL classifier is constructed and trained on noisy simulated MRFT responses. The end-to-end identification process takes place online requiring few seconds of observation data and microsecond level inference.

The suggested approach was verified through simulation and experimentation. Experiments were carried out to identify the altitude and attitude dynamics of a UAV. Results show the effectiveness of the presented techniques by demonstrating stability in the adaptation phase, accuracy of identification, and real-time computation capabilities. The proposed method was bench-marked against PEM and the optimal tuning rules; and exhibited advantages in accuracy, robustness to input biases, less observation data requirements, and faster inference. All of which makes the presented approach applicable to a wide set of practical control problems.

For future work, we aim to evaluate the proposed identification scheme for higher-order systems by extending the techniques of finding the distinguishing phase and parameter space discretization to higher dimensions.

APPENDIX A
DERIVATIVE OF THE MODIFIED SOFTMAX FUNCTION WITH THE CROSS-ENTROPY LOSS FUNCTION

The derivative of the Softmax probability p_i from (14) with respect to the k^{th} logit can be obtained using the quotient rule:

$$\begin{aligned}\frac{\partial p_i}{\partial a_k} &= \frac{\frac{\partial}{\partial a_k}(e^{\gamma_{iT}\cdot a_i})\cdot\sum_{j=1}^{N} e^{\gamma_{jT}\cdot a_j} - \frac{\partial}{\partial a_k}(\sum_{j=1}^{N} e^{\gamma_{jT}\cdot a_j})\cdot e^{\gamma_{iT}\cdot a_i}}{(\sum_{j=1}^{N} e^{\gamma_{jT}\cdot a_j})^2} \\ &= \frac{\frac{\partial}{\partial a_k}(e^{\gamma_{iT}\cdot a_i})\cdot\sum_{j=1}^{N} e^{\gamma_{jT}\cdot a_j} - \gamma_{kT}e^{\gamma_{kT}\cdot a_k}\cdot e^{\gamma_{iT}\cdot a_i}}{(\sum_{j=1}^{N} e^{\gamma_{jT}\cdot a_j})^2}\end{aligned} \tag{17}$$

In the case $i = k$:

$$\begin{aligned}\frac{\partial p_i}{\partial a_k} &= \frac{\partial p_k}{\partial a_k} \\ &= \frac{\gamma_{kT}e^{\gamma_{kT}\cdot a_k}\cdot\sum_{j=1}^{N} e^{\gamma_{jT}\cdot a_j} - \gamma_{kT}e^{\gamma_{kT}\cdot a_k}\cdot e^{\gamma_{kT}\cdot a_k}}{(\sum_{j=1}^{N} e^{\gamma_{jT}\cdot a_j})^2} \\ &= \gamma_{kT}\frac{e^{\gamma_{kT}\cdot a_k}}{\sum_{j=1}^{N} e^{\gamma_{jT}\cdot a_j}}\cdot\frac{\sum_{j=1}^{N} e^{\gamma_{jT}\cdot a_j} - e^{\gamma_{kT}\cdot a_k}}{\sum_{j=1}^{N} e^{\gamma_{jT}\cdot a_j}} \\ &= \gamma_{kT}p_k(1-p_k)\end{aligned} \tag{18}$$

Alternatively, for $i \neq k$, $\frac{\partial}{\partial a_k}(e^{\gamma_{iT}\cdot a_i}) = 0$. (17) hence simplifies as:

$$\begin{aligned}\frac{\partial p_i}{\partial a_k} &= \frac{0 - \gamma_{kT}e^{\gamma_{kT}\cdot a_k}\cdot e^{\gamma_{iT}\cdot a_i}}{(\sum_{j=1}^{N} e^{\gamma_{jT}\cdot a_j})^2} \\ &= \gamma_{kT}\frac{-e^{\gamma_{kT}\cdot a_k}}{\sum_{j=1}^{N} e^{\gamma_{jT}\cdot a_j}}\cdot\frac{e^{\gamma_{iT}\cdot a_i}}{\sum_{j=1}^{N} e^{\gamma_{jT}\cdot a_j}} = -\gamma_{kT}\cdot p_k\cdot p_i\end{aligned} \tag{19}$$

Therefore, the derivative of the modified Softmax formulation is:

$$\frac{\partial p_i}{\partial a_k} = \begin{cases} -\gamma_{kT}\cdot p_k(1-p_k) & \text{if } i = k \\ -\gamma_{kT}p_k p_i & \text{if } i \neq k \end{cases} \tag{20}$$

The cross-entropy function $L = -\sum_{i=1}^{N} y_i \log(p_i)$ has the following derivative with respect to the k^{th} logit:

$$\frac{\partial L}{\partial a_k} = -\sum_{i=1}^{N} y_i \frac{1}{p_i}\frac{\partial p_i}{\partial a_k} \tag{21}$$

where y is a One-Hot encoded vector that points out the ground truth class T.

By plugging $\frac{\partial p_i}{\partial a_k}$ from (20) to (21), the backpropagation term $\frac{\partial L}{\partial a_k}$ becomes:

$$\begin{aligned}\frac{\partial L}{\partial a_k} &= -y_k\gamma_{kT}(1-p_k) + \sum_{i\neq k} y_i(\gamma_{kT}\cdot p_k) \\ &= \gamma_{kT}[-y_k + y_k p_k + \sum_{i\neq k} y_i p_k] \\ &= \gamma_{kT}[-y_k + p_k(y_k + \sum_{i\neq k} y_i)]\end{aligned} \tag{22}$$

As y is a one-hot encoded vector, $y_k + \sum_{i\neq k} y_i = 1$. The backpropogation term hence becomes:

$$\frac{\partial L}{\partial a_k} = \gamma_{kT}(y_k - p_k) \tag{23}$$

ACKNOWLEDGMENT

The authors would like to thank Quanser for the generous support and technical assistance. They also thank Mohamad Wahbah and AbdulRahaman Al-Marzooqi for their technical assistance with the experiments. *(Abdulla Ayyad and Mohamad Chehadeh contributed equally to this work.)*

REFERENCES

[1] C. Wu, B. Ju, Y. Wu, X. Lin, N. Xiong, G. Xu, H. Li, and X. Liang, "UAV autonomous target search based on deep reinforcement learning in complex disaster scene," *IEEE Access*, vol. 7, pp. 117227–117245, 2019.

[2] J. A. Shaffer, E. Carrillo, and H. Xu, "Hierarchal application of receding horizon synthesis and dynamic allocation for UAVs fighting fires," *IEEE Access*, vol. 6, pp. 78868–78880, 2018.

[3] H. Shakhatreh, A. H. Sawalmeh, A. Al-Fuqaha, Z. Dou, E. Almaita, I. Khalil, N. S. Othman, A. Khreishah, and M. Guizani, "Unmanned aerial vehicles (UAVs): A survey on civil applications and key research challenges," *IEEE Access*, vol. 7, pp. 48572–48634, 2019.

[4] J. Kim, S. Kim, C. Ju, and H. I. Son, "Unmanned aerial vehicles in agriculture: A review of perspective of platform, control, and applications," *IEEE Access*, vol. 7, pp. 105100–105115, 2019.
[5] I. Kalyaev, S. Kapustyan, D. Ivanov, I. Korovin, L. Usachev, and G. Schaefer, "A novel method for distribution of goals among UAVs for oil field monitoring," in *Proc. 6th Int. Conf. Inform., Electron. Vis. 7th Int. Symp. Comput. Med. Health Technol. (ICIEV-ISCMHT)*, Sep. 2017, pp. 1–4.
[6] L. Ljung, *System Identification*. Boston, MA, USA: Birkhäuser, 1998, pp. 163–173, doi: 10.1007/978-1-4612-1768-8_1.
[7] L. Ljung, "Prediction error estimation methods," *Circuits, Syst., Signal Process.*, vol. 21, no. 1, pp. 11–21, Jan. 2002.
[8] P. M. Van den Hof, A. Dankers, P. S. Heuberger, and X. Bombois, "Identification of dynamic models in complex networks with prediction error methods—Basic methods for consistent module estimates," *Automatica*, vol. 49, no. 10, pp. 2994–3006, 2013.
[9] A. Hagenblad, L. Ljung, and A. Wills, "Maximum likelihood identification of Wiener models," *Automatica*, vol. 44, no. 11, pp. 2697–2705, Nov. 2008.
[10] L. Tong and S. Perreau, "Multichannel blind identification: From subspace to maximum likelihood methods," *Proc. IEEE*, vol. 86, no. 10, pp. 1951–1968, Oct. 1998.
[11] Y. Hu, J. Wu, and C. Zeng, "Robust adaptive identification of linear time-varying systems under relaxed excitation conditions," *IEEE Access*, vol. 8, pp. 8268–8274, 2020.
[12] R. Pintelon, P. Guillaume, Y. Rolain, J. Schoukens, and H. Van Hamme, "Parametric identification of transfer functions in the frequency domain–A survey," *IEEE Trans. Autom. Control*, vol. 39, no. 11, pp. 2245–2260, Nov. 1994.
[13] N. Nevaranta, S. Derammelaere, J. Parkkinen, B. Vervisch, T. Lindh, K. Stockman, M. Niemela, O. Pyrhonen, and J. Pyrhonen, "Online identification of a mechanical system in frequency domain using sliding DFT," *IEEE Trans. Ind. Electron.*, vol. 63, no. 9, pp. 5712–5723, Sep. 2016.
[14] S. Villwock and M. Pacas, "Application of the welch-method for the identification of two- and three-mass-systems," *IEEE Trans. Ind. Electron.*, vol. 55, no. 1, pp. 457–466, Jan. 2008.
[15] S. Chen, S. A. Billings, and P. M. Grant, "Non-linear system identification using neural networks," *Int. J. Control*, vol. 51, no. 6, pp. 1191–1214, Jan. 1990.
[16] S. Lu and T. Basar, "Robust nonlinear system identification using neural-network models," *IEEE Trans. Neural Netw.*, vol. 9, no. 3, pp. 407–429, May 1998.
[17] A. Delgado, C. Kambhampati, and K. Warwick, "Dynamic recurrent neural network for system identification and control," *IEE Proc.-Control Theory Appl.*, vol. 142, no. 4, pp. 307–314, Jul. 1995.
[18] D. J. Grymin and M. Farhood, "Two-step system identification and trajectory tracking control of a small fixed-wing UAV," *J. Intell. Robotic Syst.*, vol. 83, no. 1, pp. 105–131, Nov. 2015.
[19] M. Ahsan and M. Choudhry, "System identification of an airship using trust region reflective least squares algorithm," *Int. J. Control, Autom. Syst.*, vol. 15, pp. 1–10, May 2017.
[20] W. Wei, M. B. Tischler, and K. Cohen, "System identification and controller optimization of a quadrotor unmanned aerial vehicle in hover," *J. Amer. Helicopter Soc.*, vol. 62, no. 4, pp. 1–9, Oct. 2017.
[21] S. Bhandari, P. Navarro, and A. Ruiz, "Flight testing, data collection, and system identification of a multicopter UAV," in *Proc. AIAA Model. Simulation Technol. Conf.*, Jan. 2017. [Online]. Available: https://arc.aiaa.org/doi/abs/10.2514/6.2017-1558
[22] L. E. Hale, M. Patil, and C. J. Roy, "Aerodynamic parameter identification and uncertainty quantification for small unmanned aircraft," in *Proc. AIAA Guid., Navigat., Control Conf.*, Jan. 2015, pp. 1–12. [Online]. Available: https://arc.aiaa.org/doi/abs/10.2514/6.2015-1538
[23] J. Guo, B. Mu, L. Y. Wang, G. Yin, and L. Xu, "Decision-based system identification and adaptive resource allocation," *IEEE Trans. Autom. Control*, vol. 62, no. 5, pp. 2166–2179, May 2017.
[24] J. G. Ziegler and N. B. Nichols, "Optimum settings for automatic controllers," *J. Dyn. Syst., Meas., Control*, vol. 115, no. 2B, pp. 220–222, Jun. 1993.
[25] K. J. Åström and T. Hägglund, "Automatic tuning of simple regulators with specifications on phase and amplitude margins," *Automatica*, vol. 20, no. 5, pp. 645–651, Sep. 1984. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0005109884900141
[26] I. Boiko, "Modified relay feedback test (MRFT) and tuning of PID controllers," in *Non-parametric Tuning of PID Controllers*. London, U.K.: Springer, 2013, pp. 25–79, doi: 10.1007/978-1-4471-4465-6_3.
[27] M. S. Chehadeh and I. Boiko, "Design of rules for in-flight non-parametric tuning of PID controllers for unmanned aerial vehicles," *J. Franklin Inst.*, vol. 356, no. 1, pp. 474–491, Jan. 2019. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0016003218306604
[28] A. P. Schoellig, F. L. Mueller, and R. D'Andrea, "Optimization-based iterative learning for precise quadrocopter trajectory tracking," *Auto. Robots*, vol. 33, nos. 1–2, pp. 103–127, Aug. 2012. [Online]. Available: https://link.springer.com/article/10.1007/s10514-012-9283-2
[29] N. O. Lambert, D. S. Drew, J. Yaconelli, S. Levine, R. Calandra, and K. S. J. Pister, "Low-level control of a quadrotor with deep model-based reinforcement learning," *IEEE Robot. Autom. Lett.*, vol. 4, no. 4, pp. 4224–4230, Oct. 2019.
[30] F. Berkenkamp, A. P. Schoellig, and A. Krause, "Safe controller optimization for quadrotors with Gaussian processes," 2015, *arXiv:1509.01066*. [Online]. Available: https://arxiv.org/abs/1509.01066
[31] W. Koch, R. Mancuso, R. West, and A. Bestavros, "Reinforcement learning for UAV attitude control," *ACM Trans. Cyber-Phys. Syst.*, vol. 3, no. 2, p. 22, 2019.
[32] A. Molchanov, T. Chen, W. Hönig, J. A. Preiss, N. Ayanian, and G. S. Sukhatme, "Sim-to-(multi)-real: Transfer of low-level robust control policies to multiple quadrotors," 2019, *arXiv:1903.04628*. [Online]. Available: http://arxiv.org/abs/1903.04628
[33] J. Hwangbo, I. Sa, R. Siegwart, and M. Hutter, "Control of a quadrotor with reinforcement learning," *IEEE Robot. Autom. Lett.*, vol. 2, no. 4, pp. 2096–2103, Oct. 2017.
[34] S. Bansal, A. K. Akametalu, F. J. Jiang, F. Laine, and C. J. Tomlin, "Learning quadrotor dynamics using neural network for flight control," in *Proc. IEEE 55th Conf. Decis. Control (CDC)*, Dec. 2016, pp. 4653–4660.
[35] J. Loeb, "Recent advances in nonlinear servo theory," *Trans. ASME*, vol. 76, pp. 1281–1290, Nov. 1954.
[36] P. Poksawat, L. Wang, and A. Mohamed, "Automatic tuning of attitude control system for fixed-wing unmanned aerial vehicles," *IET Control Theory Appl.*, vol. 10, no. 17, pp. 2233–2242, Nov. 2016.
[37] W. Giernacki, D. Horla, T. Báča, and M. Saska, "Real-time model-free minimum-seeking autotuning method for unmanned aerial vehicle controllers based on fibonacci-search algorithm," *Sensors*, vol. 19, no. 2, p. 312, Jan. 2019.
[38] G. Hoffmann, H. Huang, S. Waslander, and C. Tomlin, "Quadrotor helicopter flight dynamics and control: Theory and experiment," in *Proc. AIAA Guid., Navigat. Control Conf. Exhibit. Hilton Head*. Reston, VA, USA: American Institute of Aeronautics and Astronautics, Aug. 2007. [Online]. Available: http://arc.aiaa.org/doi/10.2514/6.2007-6461
[39] P. Pounds, R. Mahony, and P. Corke, "Modelling and control of a large quadrotor robot," *Control Eng. Pract.*, vol. 18, no. 7, pp. 691–699, Jul. 2010.
[40] C. Cheron, A. Dennis, V. Semerjyan, and Y. Chen, "A multifunctional HIL testbed for multirotor VTOL UAV actuator," in *Proc. IEEE/ASME Int. Conf. Mech. Embedded Syst. Appl.*, Jul. 2010, pp. 44–48. [Online]. Available: http://ieeexplore.ieee.org/document/5552032/
[41] I. Boiko, "Design of non-parametric process-specific optimal tuning rules for PID control of flow loops," *J. Franklin Inst.*, vol. 351, no. 2, pp. 964–985, Feb. 2014. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0016003213003487
[42] S. Bouabdallah, A. Noth, and R. Siegwart, "PID vs LQ control techniques applied to an indoor micro quadrotor," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, vol. 3, Sep./Oct. 2004, pp. 2451–2456.
[43] R. W. Prouty, *Helicopter Performance, Stability, and Control*. Melbourne, FL, USA: Krieger, 2002.
[44] D. P. Atherton and G. M. Siouris, "Nonlinear control engineering," *IEEE Trans. Syst., Man, Cybern.*, vol. TSMC-7, no. 7, pp. 567–568, Jul. 1977.
[45] R. E. Kopp and R. J. Orford, "Linear regression applied to system identification for adaptive control systems," *AIAA J.*, vol. 1, no. 10, pp. 2300–2306, Oct. 1963.
[46] M. Jansson and B. Wahlberg, "A linear regression approach to state-space subspace system identification," *Signal Process.*, vol. 52, no. 2, pp. 103–129, Jul. 1996.
[47] C.-F. Juang and C.-D. Hsieh, "A locally recurrent fuzzy neural network with support vector regression for dynamic-system modeling," *IEEE Trans. Fuzzy Syst.*, vol. 18, no. 2, pp. 261–273, Apr. 2010.

[48] R. Rothe, R. Timofte, and L. Van Gool, "Deep expectation of real and apparent age from a single image without facial landmarks," *Int. J. Comput. Vis.*, vol. 126, nos. 2–4, pp. 144–157, Apr. 2018.
[49] H. Noh, T. You, J. Mun, and B. Han, "Regularizing deep neural networks by noise: Its interpretation and optimization," in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Red Hook, NY, USA: Curran Associates, 2017, pp. 5109–5118. [Online]. Available: http://papers.nips.cc/paper/7096-regularizing-deep-neural-networks-by-noise-its-interpretation-and-optimization.pdf
[50] Y. Li and Y. Yuan, "Convergence analysis of two-layer neural networks with ReLU activation," in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Red Hook, NY, USA: Curran Associates, 2017, pp. 597–607. [Online]. Available: http://papers.nips.cc/paper/6662-convergence-analysis-of-two-layer-neural-networks-with-relu-activation.pdf
[51] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014. [Online]. Available: http://jmlr.org/papers/v15/srivastava14a.html
[52] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. 32nd Int. Conf. Mach. Learn.*, 2015, pp. 448–456. [Online]. Available: http://jmlr.org/proceedings/papers/v37/ioffe15.pdf
[53] A. Shrestha and A. Mahmood, "Review of deep learning algorithms and architectures," *IEEE Access*, vol. 7, pp. 53040–53065, 2019.
[54] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: http://arxiv.org/abs/1412.6980
[55] T MathWorks. (2019). *System Identification Toolbox*. Accessed: Dec. 15, 2019. [Online]. Available: https://www.mathworks.com/products/sysid.html
[56] Quanser. (2020). *Qdrone—Quanser*. Accessed: May 6, 2020. [Online]. Available: https://www.quanser.com/products/qdrone/
[57] Optitrack. (2020). *Optitrack—Prime 17W*. Accessed: May 10, 2020. [Online]. Available: https://optitrack.com/products/prime-17w/
[58] A. Ayyad. (2020). *Real-Time System Identification Using Deep Learning for Linear Processes*. [Online]. Available: https://youtu.be/dz3WTFU7W7c

ABDULLA AYYAD (Member, IEEE) received the M.Sc. degree in electrical engineering from The University of Tokyo, in 2019. He conducted research at the Spacecraft Control and Robotics Laboratory, The University of Tokyo. He is currently a Research Associate with the Khalifa University Center for Autonomous Robotic Systems (KUCARS) working on several robot autonomy projects. His current research interest includes the application of AI in the fields of perception, navigation, and control.

MOHAMAD CHEHADEH (Member, IEEE) received the M.Sc. degree in electrical engineering from Khalifa University, Abu Dhabi, United Arab Emirates, in 2017. He is currently with the Khalifa University Center for Autonomous Robotic Systems (KUCARS). His research interests include the identification, perception, and control of complex dynamical systems utilizing the recent advancements in the field of AI.

MOHAMMAD I. AWAD (Member, IEEE) received the Ph.D. degree in robotics from the Khalifa University of Science and Technology, in 2018, and the master's degree in mechanical engineering from the Jordan University of Science and Technology, in 2011. He is currently a Postdoctoral Research Fellow with the Khalifa University Robotics Institute, Khalifa University of Science and Technology. He had established his career starting in 2007 as an automation engineer in a private company in Jordan. During that time, he was a Research and Teaching Assistant with the Jordan University of Science and Technology, Jordan, where he was exposed to work partially in projects related to the industrial consultancy. In 2010, he had joined the King Abdullah II Design and Development Bureau (KADDB) and served as a Product Development Engineer. He was involved in developing several military and civilian service vehicles. In 2013, he joined the Masdar Institute for Science and Technology as a Research Associate, where he is conducting research on concentrated solar power control systems and water distillation. Since 2014, he has been a Teaching and Research Associate with the Khalifa University of Science and Technology. He is a member of ASME.

YAHYA ZWEIRI (Member, IEEE) received the Ph.D. degree from King's College London, in 2003. He is currently the School Director of the Research and Enterprise, Kingston University London, U.K. He is also an Associate Professor with the Department of Aerospace, Khalifa University, United Arab Emirates. He was involved in defense and security research projects in the last 20 years at the Defence Science and Technology Laboratory, King's College London, and the King Abdullah II Design and Development Bureau, Jordan. He has published over 100 refereed journal and conference papers and filed six patents in USA and U.K. in the unmanned systems field. His central research interests include interaction dynamics between unmanned systems and unknown environments by means of deep learning, machine intelligence, constrained optimization, and advanced control.