Multirotors From Takeoff to Real-Time Full Identification Using the Modified Relay Feedback Test and Deep Neural Networks

Abdulla Ayyad Member, IEEE, Mohamad Chehadeh Member, IEEE, Pedro Silva, Member, IEEE, and Yahya Zweiri Member, IEEE

Abstract—Low cost real-time identification of multirotor unmanned aerial vehicle (UAV) dynamics is an active area of research supported by the surge in demand and emerging application domains. Such real-time identification capabilities shorten development time and cost, making UAVs’ technology more accessible, and enable a wide variety of advanced applications. In this paper, we present a novel comprehensive approach, called DNN-MRFT, for real-time identification and tuning of multirotor UAVs using the Modified Relay Feedback Test (MRFT) and Deep Neural Networks (DNN). The main contribution is the development of a generalized framework for the application of DNN-MRFT to higher-order systems. One of the notable advantages of DNN-MRFT is the exact estimation of identified process gain, which mitigates the inaccuracies introduced due to the use of the describing function method in approximating the response of Lure’s systems. A secondary contribution is a generalized controller based on DNN-MRFT that takes-off a UAV with unknown dynamics and identifies the inner loops dynamics in-flight. Using the developed framework, DNN-MRFT is sequentially applied to the outer translational loops of the UAV utilizing in-flight results obtained for the inner attitude loops. DNN-MRFT takes on average 15 seconds to get the full knowledge of multirotor UAV dynamics and without any further tuning or calibration the UAV would be able to pass through a vertical window, and accurately follow trajectories achieving state-of-the-art performance. Such demonstrated accuracy, speed, and robustness of identification pushes the limits of state-of-the-art in real-time identification of UAVs.

Index Terms—System Identification, Unmanned Aerial Vehicles, Multirotor, Learning Systems, Sliding Mode Control, Process Control.

I. INTRODUCTION

LINE controller parameters’ estimation which accounts for unknown or changing process parameters has always been of interest in the controls community. Maintaining performance and safety are the main challenges that are often tackled in adaptive control research. A unified definition for adaptive control has always been a topic of discussion in the controls community, but we found the one from (1) suitable and covers most relevant research. In (1), an adaptive controller is “a controller with adjustable parameters and a mechanism for adjusting the parameters”. In this sense, adaptive controllers are of many different types and can extend to very complex formulations (2). In this paper we build on a novel technique (3) that uses deep neural network (DNN) and the Modified Relay Feedback Test (MRFT) (4) to identify unknown process parameters. Specifically, in this work we investigate extending the approach suggested in (3) to identify side motion dynamics of a symmetric multirotor vertical take-off and landing (VTOL) unmanned aerial vehicle (UAV) (in this document referred to simply as multirotor UAVs) which is under-actuated and has modeled process dynamics of relative degree five in addition to time delay. We show that our two-stage adaptive scheme can identify process parameters in real-time with high accuracy (first stage) and then suggest optimal controller gains based on the identified system parameters (second stage). We demonstrate that using our approach, a multirotor UAV can take-off without any pre-tuned controller gains and find the optimal controller parameters in-flight. To the best of our knowledge, this is the first adaptive controller that is capable of performing a takeoff and reach optimal controllers without initial stabilizing controller gains.

Such demonstrated capability can be a game changer in the UAV industry as it shortens development time and cost, and expands the accessibility of UAV technology. For example, it benefits both the hobbyists community and enterprises that require custom UAV solutions by enabling safe and high-performance operation of custom built models in the shortest possible time. Additionally, the presented take-off and self-tuning approaches can be used in more advanced applications that requires real-time control gains adaptation while guaranteeing stability limits. A demonstration video of the presented approach applied to multiple multirotor UAVs that shows robustness in identification phase and high performance in the control phase can be found in (5).

A. Relation to Existing Adaptive Control Approaches

Adaptive control approaches are broad in nature and studying the relation of this work to all of them is not feasible.
Many adaptive methods were applied to UAVs in simulation but they might suffer when applied experimentally due to over-simplified models or unrealistic adaptation gains \[6, 7, 8, 9\]. Thus we focus our study on adaptive approaches applied to UAVs experimentally. Also, we chose to extend our literature review to system identification methods that outputs an identified model in a form suitable for controller design as long as this identification was demonstrated experimentally. We chose to limit the scope of the literature review by excluding adaptive methods that deal with very specific cases; e.g. adapting to thrust coefficient change due to ground effect, change of lift force due to a propulsion fault, weight imbalance across a single axis, etc.

One of the earliest approaches which demonstrated great success in UAV tuning is iterative learning control (ILC) \[10\]. ILC tunes a feedforward law that compensates for repeatable model uncertainties. ILC requires a high number of experimental iterations and hence cannot adapt in real-time. Additionally, the feedforward compensation technique might suffer from severely degraded performance under unseen external disturbances or changes of model parameters. The approach presented in \[11\] utilizes Gaussian processes with Bayesian optimization to learn feedback control parameters for the translational control loops. This approach requires a lot of iterations to converge and hence cannot be applied to real-time applications. Deep model-based reinforcement learning (RL) was used to adapt a RL based control policy from experimentation \[12\]. This method requires excessive experimental data, is computationally expensive, and does not provide stability guarantees. Another RL method that relies on training an RL agent by closing the simulation to reality (S2R) gap is presented in \[13\]. In the work of \[14\], the use of domain randomization technique and detailed models helped reducing the S2R gap such that generalized policies that are directly transferable to platforms of different sizes were successfully generated. State space sampling exploration techniques through deep learning \[14\], and apprenticeship learning \[15\] were used to fine-tune and improve the performance of outer loop controllers. These techniques require an abundance of experimental data and offline computation.

Other tuning approaches based on relay methods have been applied in practice. Recent work in \[16\] shows near-optimal attitude loops controller tuning based on MRFT. Though this tuning method can run in real-time, its tuning performance degrades in the presence of biases in the system \[3\]. Another relay based tuning method that uses relay feedback test (RFT) in a cascaded arrangement is presented in \[17\]. The tuning assumes first order plus time delay (FOPTD) model and was only performed on a testbed. Heuristic and model-free approaches were widely investigated in literature \[18, 19, 20, 21\] but their tuning time is generally large (a few minutes at least) and there is no guarantee of optimality of the achieved controller gains. Few other methods of UAV tuning are based on experimental system identification. In \[22\], frequency-domain identification using an adaptive genetic algorithm was performed on an unmanned helicopter. The identification method requires a fair amount of flight data, which in turn requires a pre-processing stage that includes human expertise. Similar drawbacks are present in the approach used by \[23, 24\] where UAV models based on fuzzified eigensystem realization algorithm were identified.

A common limitation of all reviewed approaches is that they require a stabilized system to begin with. This is usually done through an extensive trial and error process or initial rough tuning based on pre-measured physical parameters. This leads to a prolonged development time and increased cost especially for larger UAVs. Also, most of the presented approaches can be exclusively used either to adapt attitude and attitude rate loops gains (inner loops), or outer loops gains. The literature lacks a unified robust approach for tuning of the inner and outer loops of multirotor UAVs. Another limitation specific to data-driven approaches like ILC, state-space sampling approaches, and other identification methods widely adopted in literature \[25, 26, 27\] is that tuning performance is dependent on how data is generated. Data generation for these adaptation techniques has its own complexities and requires an expert human to perform.

Our proposed approach uses MRFT, which can be considered as an extension of the widely used RFT, to excite a certain system response. This system response is fed to a DNN that is able to infer system parameters. Therefore, we refer to the approach presented in this paper by DNN-MRFT. DNN-MRFT is the most appealing approach compared to the other relevant adaptive approaches described in literature due to its stability guarantees, its minimal data requirements, and its computational efficiency which enables it application in real-time. DNN-MRFT provides additional benefit that results in accurate identification of model parameters, permitting the design of controllers other than PID. Thus DNN-MRFT can be also considered as a system identification method. The DNN is only trained on simulation data which greatly simplifies the identification algorithm design process.

B. Contributions

DNN-MRFT provides a unified approach for the identification of a linear system’s parameters. It was first introduced and applied to a second order plus integrator plus time delay (SOIPTD) system depicting multirotor UAV attitude dynamics \[3\]. The main contribution of the present paper is to investigate the viability of using the DNN-MRFT approach to multirotor UAV side motion dynamics. These dynamics can be formulated as a fifth order system with time delay, which requires a different treatment than SOIPTD model presented in our previous work \[3\]. Existing DNN-MRFT formulation is not suited for the identification of such higher order systems as MRFT can only produce periodic oscillations corresponding to the second and third quadrants of the complex plane. Harmonic balance (HB) anticipates oscillations with amplitudes that can reach up to tens of meters when running conventional DNN-MRFT on processes considered in Table \[1\] which clearly indicates impracticality of the conventional formulation. The solution to this challenge is in the extension of DNN-MRFT to higher order dynamics in a hierarchical fashion where identified inner loop dynamics, including inner loop controller, are considered in the identification of outer loop dynamics.
This hierarchical identification approach through DNN-MRFT can be repeated to generalize for any higher order systems. As a result of the DNN-MRFT approach, it is possible to exactly identify open-loop system gain utilizing simulated knowledge of the considered systems. This mitigates amplitude determination inaccuracy due to the Describing Function (DF) method’s low pass filtering assumption. In literature, the Locus of the Perturbed Relay System (LPRS) \[23\] was suggested as an exact description of discontinuous systems. The proposed amplitude scaling technique in this paper is simpler, and can be directly used in controller tuning. A secondary contribution of this paper is the development of an algorithm based on the DNN-MRFT hierarchical identification approach that allows a multirotor UAV to takeoff without any initial controller parameters and perform identification and tuning safely for all control loops. This takeoff algorithm is specific to multirotor UAVs and was tested experimentally on multiple sizes and designs of multirotor UAVs. The overall performance of UAV identification is demonstrated by two experiments. First we achieve trajectory tracking performance on par with state-of-the-art. Second, using the DNN-MRFT identification results, the multirotor UAV can pass through a vertical narrow window without the need of fine-tuning or any other sort of human expert input to the controller structure or gains. To the best of our knowledge, this is the only adaptive approach that allows a multirotor UAV to takeoff without any other feedback controller that can perform such aggressive maneuvers in a completely autonomous manner.

C. Paper Outline

This paper is organized as follows: aspects related to dynamics modeling and relevant assumptions are discussed in Section \[II\] Overall theoretical framework of DNN-MRFT identification including the design of MRFT parameters through finding the distinguishing phase and relevant stability considerations are discussed in Section \[III\] The process of discretizing the model parameter space into a finite set of representative processes is described in Section \[IV\] The DNN model development and the generation of training data through simulation is discussed in Section \[V\] A modified method for finding exact system gain that mitigates the DF approximation is presented in Section \[VI\] The design of an algorithm that can perform safe take-off, identification, and tuning of optimal controllers for UAVs is shown in Section \[VII\] Finally, extensive simulation and experimental results which demonstrate state-of-the-art performance and adaptation robustness are presented in Section \[VIII-A2\]

II. MODELLING OF DYNAMICS

In this work, we define the inertial frame \(\mathcal{F}_I\) to be earth-fixed right-handed reference frame with \(z_I(+)\) pointing upwards. The right-handed body reference frame \(\mathcal{F}_B\) is attached to the multirotor UAV center of mass, with \(z_B(+)\) perpendicular to the body upper surface, and is always aligned with its attitude and heading angles. A rotation matrix used to transform between reference frames is denoted by \(\frac{T_s}{s}R\), where \(T\) is the target reference frame and \(S\) is the source one.
external disturbances such as gravity, sensor bias, unmatched propulsion thrust, or model asymmetry.

The coupled dynamics of rotational motion due to the gyroscopic effect is assumed to be negligible. This is because in the operational limits we are interested in, the torques generated due to gyroscopic effects are considerably smaller than the torques contributed by other dynamics of the system [16]. Thus the assumption of single input single output (SISO) system dynamics for every rotational control loop is valid. Coupling of rotational dynamics can also occur due to other reasons like sensor misplacement, asymmetric center of mass, etc. Care was taken to minimize such effects when preparing the experimental setup.

B. Modeling of Side Motion Dynamics

Multirotor UAVs are underactuated due to the fact that movements in the $x_B(\cdot)$ and $y_B(\cdot)$ due to actuator action are not possible. But side movements in the inertial frame $F_I$ are possible and can be approximated to be linear for small attitude angles. The linearized side motion dynamics are given by:

$$G_{outer}(s) = \frac{K_{eq,att}K_{eq,side}e^{-(\tau_{rot} + \tau_{side})s}}{s^2(T_{prop}s + 1)(T_1s + 1)(T_2s + 1)}$$

Here we assume a linear drag term $T_2$ that describes air resistance acting on the body of the multirotor UAV frame due to translational motion. The assumption of linearity works well in practice for small angles due to the fact that translational drag terms remain similar, and altitude loss due to thrust vector change is negligible (i.e. assume a nominal value of thrust provided by the motors). Overall system dynamics with feedback control design for small angles are shown in Fig. 2.

Note that the attitude (inner loop) model parameters directly affect the performance of the outer loop dynamics.

In cases where aggressive maneuvers require large rotation angles, we present a different treatment for the generation of reference attitude and thrust. This is achieved by utilizing the controller structure presented in [32], where the differential flatness properties of multirotor UAVs are exploited such that the outer loop controller outputs a desired inertial force vector $\vec{F}_{des}$ instead of directly producing reference rotation angles and thrust. The feedback control structure of [32] neglects actuator and inner-loop dynamics, nonetheless, the identification and tuning approach presented in this paper takes these dynamics into account. The inertial force vector is related to the UAV’s attitude and thrust by the following equation:

$$\vec{F} = \begin{pmatrix} F_x \\ F_y \\ F_z \end{pmatrix} = I_B R \begin{pmatrix} 0 \\ 0 \\ U_z \end{pmatrix} - m \begin{pmatrix} 0 \\ 0 \\ g \end{pmatrix} \quad (3)$$

Following [3], we use $\vec{F}_{des}$ to solve for the desired UAV attitude and thrust. We define a reference frame $\tilde{F}_{B*}$ that represents the desired UAV attitude as follows:

$$z_{B*}(\cdot) = \frac{\vec{F}_{des}}{||\vec{F}_{des}||} \quad (4)$$

$$y_{B*}(\cdot) = \frac{\vec{F}_{des} \times \begin{pmatrix} \cos r_{\psi} \\ \sin r_{\psi} \end{pmatrix}}{||\vec{F}_{des} \times \begin{pmatrix} \cos r_{\psi} \\ \sin r_{\psi} \end{pmatrix}||} \quad (5)$$

$$x_{B*}(\cdot) = y_{B*}(\cdot) \times z_{B*}(\cdot) \quad (6)$$

where $r_{\psi}$ is the reference yaw angle.

The roll, pitch, and yaw components of the orientation error successively denoted by $e_{\phi}$, $e_{\theta}$, and $e_{\psi}$ can then be computed as:

$$\begin{pmatrix} e_{\phi} \\ e_{\theta} \\ e_{\psi} \end{pmatrix} = E(I_{ij}^* R_{ij} R) \quad (7)$$

where $E$ is a function that converts a rotation matrix in $SO(3)$ to the corresponding rotation vector in $R^3$.

Finally, we solve for $U_z$ as follows:

$$U_z = \vec{F}_{des} \cdot z_{B*}(\cdot) \quad (8)$$

C. Yaw Dynamics

Rotation around $z_B$ axis results in change of the yaw angle. Yaw has second order dynamics and is given by:

$$G_{yaw}(s) = \frac{K_{eq}e^{-\tau_{att}s}}{s(T_{prop}s + 1)} \quad (9)$$

Because yaw controller is easy to tune due to the small delay value and the presence of full state measurements, we assume that a controller with satisfactory performance exists prior to the flight.

D. Bounds of Considered Model Parameters

The identification method presented in this paper requires the considered model parameters to be bounded. This would limit the amount of data and labels to be handled by the DNN classifier presented later. In this work, we consider commonly used multirotor UAV designs ranging from small racing quadrotors to larger multirotors with take-off weight of up to approximately 50Kgs. The selection of the parameters domain was based both on experimental findings of previous work in the literature [33], [16], [30], [31], in addition to modeling equations like those discussed in [16], [30], [29].
It is worth noting that the identification performance is not sensitive to the selection of the parameters’ range, rather, the selection of parameter bounds can be safely expanded to include UAV designs beyond the specified ranges. Such expansion would be at the cost of increased simulation and DNN training times. The selected bounds of model parameters for the considered control loops can be found in Table I.

### TABLE I: The model parameters’ ranges for all the feedback loops considered in this paper

| Parameters | Domain |
|------------|--------|
| Attitude   | \(D_{\text{att}} := \{(T_{\text{prop}}, T_1, \tau_{\text{att}}) : 0.015 \leq T_{\text{prop}} \leq 0.3, \ 0.2 \leq T_1 \leq 2, \ 0.0005 \leq \tau_{\text{att}} \leq 0.03\}\) |
| Altitude   | \(D_{\text{alt}} := \{(T_{\text{prop}}, T_1, \tau_{\text{alt}}) : 0.015 \leq T_{\text{prop}} \leq 0.3, \ 0.2 \leq T_1 \leq 2, \ 0.0005 \leq \tau_{\text{alt}} \leq 0.15\}\) |
| Side       | \(D_{\text{side}} := \{(T_2, \tau_{\text{side}}) : 0.2 \leq T_2 \leq 6, \ 0.0005 \leq \tau_{\text{side}} \leq 0.15\}\) |
| Peak. Thrust-to-Weight Ratio | \(C_{\text{TW}} \in [1.5, 5]\) |
| Motor-to-motor distance (m) | \(l_{\text{mm}} \in [0.15, 1.5]\) |
| Moment of Inertia (kg \(\cdot\) m\(^2\)) | \(\{I_x, I_y\} \in [1.6 \times 10^{-3}, 2.25]\) |

### III. MRFT and Identification Approach

#### A. The Modified Relay Feedback Test

DNN-MRFT relies on exciting certain system response using MRFT as a controller. MRFT is an algorithm that can excite self-sustained oscillations at a specific phase \(\phi\), and is realized by the following equation [33]:

\[
\begin{align*}
u_M(t) = & \begin{cases} 
  h & : e(t) \geq b_1 \lor (e(t) > -b_2 \land u_M(t-) = h) \\
  -h & : e(t) \leq -b_2 \lor (e(t) < b_1 \land u_M(t-) = -h)
\end{cases}
\end{align*}
\]

(10)

where \(b_1 = -\beta e_{\text{min}}\) and \(b_2 = \beta e_{\text{max}}\), \(e_{\text{max}} > 0\) and \(e_{\text{min}} < 0\) 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: \(e_{\text{max}} = e_{\text{min}} = 0\), \(\beta\) is a constant parameter that dictates the phase of the excited oscillations as:

\[
\varphi = \arcsin(\beta)
\]

(11)

Using the DF method, it could be shown that the MRFT achieves oscillations at a specified phase angle by satisfying the HB equation [33]:

\[
N_d(a_0)G(j\Omega_0) = -1
\]

(12)

The DF of MRFT is presented in [4] as:

\[
N_d(a_0) = \frac{A_h}{\pi a_0} (\sqrt{1 - \beta^2} - j\beta)
\]

(13)

The DF method provides an approximate solution that is valid only if \(G(s)\) 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) [36] could be thought of as a special case of the MRFT algorithm where \(\beta = 0\).

#### B. The Distinguishing Phase

The idea of distinguishing phase is based on the supposition that the optimal phase angle at which the test oscillations are generated and which is obtained through the design of optimal tuning rules [37], [4], [16] would reveal the most distinguishing characteristics of the considered processes domain. In a previous work [3], we showed that for an LTI system \(G(s)\) with known model structure and unknown set of bounded model parameters \(D\), there exists a distinguishing phase \(\varphi_d\) at which the characteristics of the self-excited oscillations induced by the MRFT can be used to identify the corresponding processes in \(D\). The distinguishing phase \(\varphi_d\) can be determined by the process of designing optimal non-parametric tuning rules as outlined in [3], [37]. Note that MRFT parameter \(\beta\) is related to the distinguishing phase by Eq. (11). Algorithm 1 summarizes the steps taken to find the value of \(\varphi_d\).

**Algorithm 1: Finding distinguishing phase through optimal non-parametric tuning rules design**

**INPUT:** \((G(s), D)\) - Model Structure, Parameters Domain

**OUTPUT:** \(\varphi_d\) - Distinguishing Phase

1: Discretize the desired parameters subspace \(D\) to obtain \(\hat{D}\);
2: Select phase margin or gain margin tuning specifications;
3: Find the set of locally optimal tuning rules \(\Delta\) for every process in \(\hat{D}\);
4: Apply every optimal tuning rule in \(\Delta\) to all other processes in \(D\) and get the set \(\Sigma\) corresponding to the value of the worst performance deterioration of every process in \(\hat{D}\) due to the application of the non-optimal tuning rule ;
5: Select the tuning rule from \(\Delta\) that corresponds to the least worst deterioration value from \(\Sigma\) as the globally optimum tuning rule \(\Delta^*\);
6: Compute \(\varphi_d\) from \(\beta \in \Delta^*\);

#### C. Identification Approach

The MRFT can excite stable periodic oscillations only in the second and third quadrants of the complex plane. As a result, for high relative degree systems, the generated test oscillations have values of \(a_0\) and \(\Omega_0\) from Eq. (12) that may not be practically useful. This is a reflection of the
importance of using a cascade controller arrangement, which would in turn require to organize the MRFT tests in each loop separately to tune each controller. It would also eliminate the indicated problem of the test oscillations possibly being of low frequency and high amplitude. For this reason, the considered high order LTI system must be split into a composition set \( G_{HO} := \{G_1, G_2, \ldots, G_M\} \) where \( G_1 \) represents the sub-system with smallest relative degree with respect to the control command. The iterative design method required to generate the feedback structure and hence required set of distinguishing phases for higher order LTI processes can be found in Algorithm 2. Note that Algorithm 2 assumes that inner feedback loops are observable and controllable.

**Algorithm 2:** Generating cascaded feedback structure

**INPUT:** \( (G_{HO}(s), D) \) - Model Structure, Parameters Domain

**OUTPUT:** \( G_{FB} \) - Resulting Feedback Structure

1. Split \( G_{HO} \) to \( \{G_1, G_2, \ldots, G_M\} \) based on M observable outputs with unity open loop gain
2. \( G_{res} \leftarrow 1 \)
3. for \( i=1, \ldots, M-1 \) do
4. \( G_{res} \leftarrow G_{res} G_i \)
5. \( V_i \leftarrow \) All processes at the vertices of \( D \) parameters in \( G_{res} \)
6. \( (a_0, \Omega_0) \leftarrow MRF(T_{i+1} V_i, \varphi_d) \) from Eq. (10)
7. if \( Re\{N_d(a_0) G_{i+1}(j\Omega_0) V_i(j\Omega_0)\} \geq 0 \) or Impractical \( a_0 \) or \( \Omega_0 \) values then
8. \( C^* \leftarrow \) Find optimal controller of \( G_{res} \)
9. \( G_{res} \leftarrow \) Feedback\( (C^*, G_{res}) \)
10. end if
11. end for
12. \( G_{res} \leftarrow G_{res} G_M \)
13. \( C^* \leftarrow \) Find optimal controller of \( G_{res} \)
14. \( G_{FB} \leftarrow \) Feedback\( (C^*, G_{res}) \)

For the particular case presented in this paper, we obtain two cascaded feedback loops by applying Algorithm 2 to the side motion model in Eq. (2). From Algorithm 2 line 1 we get:

\[
G_{HO}(s) \to \{G_1, G_2, G_3, G_4\} = \left\{ \frac{e^{-\tau_{tot} s}}{(T_{body} s + 1) (T_{prop} s + 1)^2} \right\}
\]

Where \( G_1 \) represents attitude rate dynamics, \( G_1 G_2 \) represents attitude dynamics given in Eq. (1), \( G_1 G_2 G_3 \) represents side motion velocity dynamics, and \( G_{tot} = G_1 G_2 G_3 G_4 \) represents side motion dynamics given in Eq. (2). The condition in Algorithm 2 at line 7 is met only when \( i = 2 \) for multicopter UAV side motion case which will result in two cascaded feedback loops shown in Fig. 2. Also, this means that we will end up with a set of distinguishing phases; one distinguishing phase to reveal the inner loop attitude dynamics and a distinguishing phase for every process in \( D_{att} \) to reveal outer loop position dynamics. Note that the distinguishing phase of the particular outer loop system depends on the inner closed-loop dynamics. Thus prior to outer loop identification, the parameters of the inner loop dynamics, and the optimal controller for the inner loop have to be identified first. The overall identification scheme used in this paper is shown in Fig. [3]. It is important to note that this approach is generic and can be applied to higher order LTI systems as long as the distinguishing phase corresponds to second or third quadrants in the complex plane.

**D. Stability Considerations**

Stability aspects of the DNN-MRFT approach can be divided into two categories. First category includes stability of the periodic oscillations generated by MRFT when the system is in the identification phase. Existence of stable periodic solutions will be proved in this section using the DF method and Loeb’s criterion [38]. Other aspects related to stability in the sense specific to UAVs like boundedness of bias and oscillation’s amplitude are discussed in Section [VII]. The second aspect of stability is the guarantee of system stability upon the controller parameters selection by the DNN classifier. Similar to the original formulation of the MRFT and coordinated test and tuning [4], in which the stability is guaranteed with a specified gain or phase margins; the current approach guarantees stability for the class of dynamic systems being considered. Monte-Carlo sampling is used to demonstrate this aspect of stability as discussed in Section VIII-A1.

Loeb's criterion provides necessary condition for asymptotic orbital stability [38]. We follow Loeb criterion formulation similar to the one used in [35], [39]. Let a perturbation in amplitude denoted by \( \Delta a \) cause a change in the Laplace variable \( \Delta s = \Delta \sigma + j \Delta \Omega \) such that HB still holds:

\[
N(a_0 + \Delta a)G(j\Omega_0 + \Delta \sigma + j \Delta \Omega) = -1
\]

Using fundamental theorem of calculus and differentiation chain rule we rearrange to obtain:

\[
\frac{\partial N}{\partial a} \Delta a + \frac{\partial G}{\partial \sigma} \Delta \sigma + \frac{\partial G}{\partial \Omega} \Delta \Omega = -1
\]

We cancel resultant \( N(a_0)G(j\Omega_0) \) and \( (\Delta a)^2 \) terms and then divide all terms by \( N(a_0)G(j\Omega_0) \). Using the property \( \frac{\partial G}{\partial \Omega} = -j \frac{\partial G}{\partial \Omega} \) we obtain:

\[
\frac{\partial \ln N}{\partial a} - j \frac{\partial \ln G}{\partial \sigma} \frac{\partial \sigma}{\partial a} + \frac{\partial \ln G}{\partial \Omega} \frac{\partial \Omega}{\partial a} = 0
\]

By using the property \( \ln z = \ln|z| + j \arg z \) we obtain:

\[
\left( \frac{\partial \ln |G|}{\partial \Omega} + j \frac{\partial \arg G}{\partial \Omega} \right) \left( \frac{\partial \Omega}{\partial a} - j \frac{\partial \sigma}{\partial a} \right) = - \frac{\partial \ln |N|}{\partial a} - j \frac{\partial \arg N}{\partial a}
\]

By multiplying the real part of the above equation with \( \frac{\partial \arg G}{\partial \Omega} \) and the imaginary part with \( \frac{\partial \ln |G|}{\partial \Omega} \) and rearranging, we get the following equation:

\[
\frac{d\sigma}{da} = - \frac{\partial \ln |N| \partial \arg G}{\partial \Omega} \frac{\partial \Omega}{\partial a} + \frac{\partial \arg N \partial \ln |G|}{\partial \Omega} \frac{\partial \Omega}{\partial a}
\]

For asymptotic orbital stability the condition \( \frac{d\sigma}{da} < 0 \) is necessary. Given the MRFT describing function presented in Eq. (13), the second term in the numerator of the above
equation is zero. Also, considering the positive denominator in the above equation we can state the conditions of orbital stability of MRFT oscillations as:

\[- d\ln |N| \frac{\partial \text{Arg } G}{\partial \Omega} < 0 \]  \tag{20}

Considering the DF of the MRFT we have:

\[- d\ln |N| = \frac{1}{|N|} d|N| = \frac{1}{a} \]  \tag{21}

which is always positive. Then the term \( \frac{\partial \text{Arg } G}{\partial \Omega} \) needs to be strictly negative for at least a single solution of the HB equation for the MRFT to generate a stable periodic motion. For the system model presented in Eq. (1) we have:

\[ \frac{\partial \text{Arg } G_{\text{inner}}(j\Omega)}{\partial \Omega} = -\tau - \frac{T_{\text{prop}}}{1 + (T_{\text{prop}}\Omega)^2} - \frac{T_I}{1 + (T_I\Omega)^2} \]  \tag{22}

which is always negative, and hence satisfy the stability condition in Eq. (20). For the outer loop case, we need to analyze the closed loop dynamics given by:

\[ G_{cl} = \frac{C_{\text{att}}G_{\text{att}}}{1 + C_{\text{att}}G_{\text{att}}} \]  \tag{23}

where \( C_{\text{att}} \) is a PD controller. Define \( G_{ol} = C_{\text{att}}G_{\text{att}} \). This closed-loop structure is detailed in Fig. 1 and shown in the context of the outer-loop in Fig. 2. Given the properties of logarithmic derivatives of complex numbers, we have:

\[ \frac{1}{G} \frac{dG}{d\Omega} = \frac{d\ln G}{d\Omega} = d\ln |G| + j\frac{d\text{Arg } G}{d\Omega} \]  \tag{24}

which leads to:

\[ \frac{d\text{Arg } G_{cl}}{d\Omega} = \text{Im}\left\{ \frac{1}{G_{cl}} \frac{dG_{cl}}{d\Omega} \right\} = \text{Im}\left\{ \frac{1 + G_{ol}}{G_{ol}} \frac{dG_{ol}}{d\Omega} \right\} = \text{Re}\left\{ \frac{d\ln G_{ol}}{d\Omega} \right\} \frac{1}{1 + G_{ol}} + \text{Re}\left\{ \frac{d\ln G_{ol}}{d\Omega} \right\} \text{Im}\left\{ \frac{1}{1 + G_{ol}} \right\} \]  \tag{25}

This expression needs to be negative in order for the stability condition in Eq. (20) to be satisfied. The terms \( \frac{d\text{Arg } G_{ol}}{d\Omega} \) and \( \frac{d\ln |G_{ol}|}{d\Omega} \) in Eq. (25) are always negative as shown earlier. Because \( G_{cl} \) is always tuned for stability the gain margin would be greater than one. As a result the term \( \text{Re}\left\{ \frac{1}{1 + G_{cl}} \right\} \geq 1 \) always positive for test parameter \( \beta \) close to zero. The term \( \text{Im}\left\{ \frac{1}{1 + G_{cl}} \right\} \) is close to zero for the considered values of \( \beta \) which makes the first term in Eq. (25) dominant. As a result the expression in Eq. (25) is always negative which leads to the satisfaction of the asymptotic orbital stability condition given in Eq. (20). This finding was confirmed numerically for the range of parameters given in Table I.

**IV. GENERATING REPRESENTATIVE PROCESSES**

This section describes the steps undertaken to discretize the model parameter subspace \( D \) into a discretized set of representative processes \( \tilde{D} := \{ G_1, G_2, ..., G_N \} \) that capture the main dynamics of the full range of parameters shown in Table I. The discretization of \( D \) enables tackling parametric identification as a classification problem, where a classifier maps a process under test to the most appropriate process in \( \tilde{D} \). The recognition of representative processes serves several objectives in the DNN-MRFT approach. First, it alleviates the need of an online real-time controller optimizer as optimal controllers are designed offline for all processes in \( \tilde{D} \). Second, knowledge of the dynamics of \( \tilde{D} \) is exploited to identify exact process gains as later explained in section VII. The third objective of discretizing \( D \) is providing a measure of discrepancy between all representative processes that correspond to the controller auto-tuning objective. This measure of discrepancy is utilized for the training of DNN-classifiers as discussed in detail in section VII. These advantages outweigh the marginal loss of accuracy resulting from the discretization process, which is shown to be negligible by the results obtained in Table VII.

A proper criteria must be defined for the discretization of \( D \) that establishes sufficient guarantees on the performance of system identification and controller auto-tuning without sacrificing the distinguishability of the discretized processes. For
instance, an equispaced discretization with a small partitioning distance would generate an overly-discretized \( D \) with an imbalanced representation of the frequency response characteristics of \( D \); this in turn introduces undesirable biases in training the DNN classifiers. Alternatively, a very large partitioning distance generates substantial discretization errors and does not guarantee proper performance margins for the optimal controllers designed offline for \( D \). As our objective is auto-tuning controller parameters, for the criterion of discretization, we adopt the concept of controller performance deterioration used for the system identification approach presented in [3].

Given a performance index \( Q \) that quantifies errors resulting from a closed loop application of controller \( C \) to a process \( G \), the controller performance deterioration \( J_{ij} \) between two dynamic processes \( \{G_i, G_j\} \) is defined as:

\[
J_{ij} = \frac{Q(C_i, G_j) - Q(C_j, G_j)}{Q(C_i, G_j)} \times 100\%
\]

where \( J_{ij} \) represents the relative degradation in performance in terms of \( Q_{ISE}(C_i, G_i) \) when the optimal controller of \( G_i \) is replaced by \( C_j \), which is optimal for another process \( G_j \). Eq. (26) is the relative sensitivity function introduced in [40].

It must be noted that the above formulation of the controller deterioration is non-commutative, that is \( J_{ij} \neq J_{ji} \). Therefore, the joint cost function \( J_{ij}^{max} = \max\{J_{ij}, J_{ji}\} \) is used as the discretization criteria in the remainder of this paper. Additionally, the design of optimal controllers is limited to a PD structure with a minimum phase margin constraint imposed to the controller optimization problem. The performance index \( Q \) used for controller synthesis is the conventional ISE criterion applied to a unit step response, and is given by:

\[
Q_{ISE}(C, G) = \frac{1}{T_s} \int_0^{T_s} e(t)^2 dt
\]

Following the criterion in Eq. (26), discretization is performed such that adjacent processes in \( D \) achieve a target joint cost \( J^* \) within an admissible tolerance value. We first discretize the three-dimensional parameters space of altitude and attitude models. For computational efficiency we have utilized the linear time scale property of cost function \( Q(T, s), G(T, s) = \alpha_c Q(C(s), G(s)) \), where \( \alpha_c \) is an unknown dimensionless constant, to perform sampling in a two-dimensional hemispherical hyper-surface \( S \) as shown in part (a) of Fig. 4. These discretized processes in \( S \) get scaled in time to populate \( D \) as illustrated in Fig. 4 and detailed in our previous work [3].

Based on the model parameter bounds given in Table II and the discretization accuracy specifications given in Table II, discretization of \( D_{att} \) and \( D_{att} \) yield a total number of \( N_{att} = 208 \) and \( N_{att} = 48 \) representative processes respectively.

The discretization of \( D_{side} \) depends on the identified inner loop process and its corresponding optimal controller. Hence \( D_{side,i} \) is unique for every process \( G_{att,i} \in D_{att} \) where \( i = 1, ..., N_{att} \). The discretization of a given \( D_{side} \) domain is performed based on the maximum sensitivity values obtained from \( \frac{\partial J_{att}}{\partial C_{att,i}}, \frac{\partial J_{att}}{\partial C_{att,i}} \). Spacing of the discretization was based on the most sensitive \( J_{ij} \) to changes in the parameters space which resulted in a slightly over-discretized \( D_{side} \) but guarantees that the cost between adjacent processes does not exceed \( J^* \).

To summarize, discretization steps can be stated as follows:

1. Identify domain of discretization \( D_{alt}, D_{att}, \) and \( D_{side} \).
2. Set the three relative sensitivity parameters: \( J^* \), admissible tolerance, and minimum phase margin \( \phi_m \) as shown in Table II.
3. Discretize \( D_{att} \) and \( D_{alt} \) as illustrated in Fig. 4 and outlined in [3].
4. Run Algorithm 3 on \( D_{alt} \) to get \( D_{side} \) and \( C_{side}^* \).

Algorithm 3: Side motion parameters domain discretization

**INPUT:** \((D_{att}, D_{side}, J^*)\) - Attitude loop set of discretized parameters, parameters domain of outer loop dynamics, target joint cost

**OUTPUT:** \((D_{side}, C_{side}^*)\) - Set of outer loop discretized processes, Lookup table of outer loop optimal controller parameters

1. for all \( G_{att,i} \in D_{att} \) do
2. Identify inner loop optimal controller \( C_{att,i}^* \);
3. Utilizing \( C_{att,i}^* \), \( G_{att,i} \), discretize \( D_{side} \) into \( D_{side,i} \) based on \( J^* \);
4. \( \varphi_{d,i} \leftarrow \) GetDistinguishingPhase\((C_{att,i}^*, G_{att,i}, D_{side})\) from Algorithm 1.
5. for all \( G_{side,i} \in D_{side,i} \) do
6. Identify outer loop optimal controller \( C_{side,i}^* \);
7. \( C_{side,i}^* \leftarrow C_{side,i}^* \cup C_{side,i}^* \);
8. end for
9. \( D_{side} \leftarrow D_{side} \cup D_{side,i} \);
10. \( C_{side}^* \leftarrow C_{side}^* \cup C_{side,i}^* \);
11. end for

**TABLE II:** Specifications for the process of discretizing the parameter space \( D \)

| Specification              | Value |
|----------------------------|-------|
| Target joint cost \( J^* \) | 10%   |
| Admissible tolerance       | 3%    |
| Minimum PM constraint      | 20    |
| Optimization algorithm     | Nelder-Mead simplex algorithm |

V. DATA GENERATION AND DEEP NEURAL NETWORK MODEL TRAINING

The deep neural network component of the DNN-MRFT approach provides a mapping from the MRFT response of the unknown process to the best representative process in \( D \). This mapping is denoted by \( \Gamma : X \rightarrow D \); where \( X \in \mathbb{R}^{2\times n_s} \) is a vector concatenating \( n_s \) samples of the controller output and process variable of the MRFT response. We have previously demonstrated the appropriateness of DNN for the system identification task in [3], where a single network was utilized for the identification of attitude and altitude model parameters.
processes in the parameter space. (a) Inner loop dynamics

![Image](49x604 to 300x736)

Fig. 4: The overall discretization scheme for identifying key

In this section, we build upon our previous results and present

overall pipeline of UAV system identification and controller tuning using deep neural network.

All the developed DNN models follow the same architecture shown in Fig. 5, where sequences of fully-connected layers and activation functions are concatenated. Dropout and batch normalization are applied to the outputs of each fully-connected layer to avoid over-fitting and accelerate the training process [42], [43]. After the final fully-connected layer, we utilize the cost-augmented soft-max formulation introduced in [3], which exhibited performance improvements over the conventional soft-max formulation for system identification tasks due to introducing meaningful discrepancies to the cost of miss-classification. The augmented formulation is given by:

\[
p_i = \frac{e^{(1+J_{ct})a_i}}{\sum_{j=1}^{N} e^{(1+J_{ct})a_j}}
\]

where the controller deterioration joint cost \( J_{ct} \) is utilized as the measure of discrepancy between the DNN prediction and the ground truth model parameters \( G_T \). Cross-entropy is then utilized as the loss function for training the DNN models.

We utilized the ADAM optimization algorithm for training as it is a well-established algorithm with proven advantages in terms of convergence speeds and robustness to noisy gradients [44], [45]. We implemented an automated search approach to determine the best network size and set of hyper-parameters for each of the developed 50 DNN models. The variables included in the search process along with their corresponding search space are shown in Table III. For each classification
task, the network that performed best on the validation set was selected as the preeminent DNN model.

### TABLE II: Search space of DNN structure and hyper-parameter optimization process

| Parameter               | Search Space                        |
|-------------------------|-------------------------------------|
| Number of layers        | {1, 2, 3}                           |
| Neurons per layer       | {50, 100, 1000, 3000}               |
| Activation function     | ReLU, tanh                          |
| Base learning rate      | 0.005                               |
| Gradient decay factor   | 0.9                                 |
| Gradient decay factor   | 0.999                               |

**VI. IDENTIFICATION OF EXACT PROCESS GAIN**

The information contained in the MRFT response of the system are embedded within the generated oscillations in three forms: frequency, amplitude, and shape. The DNN classifier only utilizes the frequency and shape of the oscillation to identify model parameters as the amplitude is normalized to one during pre-processing. The importance of the shape of the oscillation emphasizes the fact that the MRFT excites multiple frequencies of the linear system. The periodic components of the self-excited oscillation can be given by the Fourier series [55]:

$$y(f) = \sum_{n=0}^{\infty} a_n \cos(nf) + b_n \sin(nf)$$  

where $a_n$ and $b_n$ are Fourier series coefficients. For an odd symmetric nonlinearity (note that MRFT switching at steady state resembles a hysteresis relay), coefficients for even values of $n$ and other odd harmonics exist. The DF solution, presented in Eq. [13] for the MRFT, accounts for the first order harmonic only. The amplitude of the harmonics depend on the low pass filtering properties for every process in $\bar{D}$. Therefore, if we have identified process parameters experimentally, we can use such knowledge to predict exact system amplitude response. Exact analytical solution of Lure systems can be provided by the LPRS method [28, 46], or Tsypkin’s method [47]. To achieve exactness and real-time capability, we simply introduce a scaling coefficient $\zeta$ that provides exact system gain for every system in $\bar{D}$ and make these values available in a look-up table. The values of $\zeta$ are found by simulating MRFT control with each system in $\bar{D}$. In simulation, we use the same MRFT implementation used experimentally and measure the system steady-state response amplitude to find $\zeta$. During the DNN-MRFT identification phase, the proper $\zeta$ value is selected from the look-up table and is used to scale the identified controller parameters as shown in Fig. 5. The results in Table VI show the improvement in controller performance on a simulated test set due to the use of identified exact gain scale compared to amplitude reported by the DF method. Note that the DF method uses $\zeta = \frac{2}{\pi}$ for all processes.

**VII. TAKE-OFF CONTROLLER DESIGN**

Using the trained DNN and MRFT, the UAV can takeoff and immediately identify UAV dynamic parameters. MRFT parameter $\beta$ corresponds to the distinguishing phase and was found in Section III-B. MRFT parameter $h$ needs to be designed such that it provides adequate amplitude response and robustness against sensor noise and model biases. When MRFT is running, we can consider the following simplified system structure:

$$\dot{x} = Ax(t) + B(u + f_d)$$  

where $|f_d| \leq L > 0$ is unknown bounded constant bias inherent in the system. Periodic oscillations would be achieved whenever $u_0 - h \leq f_d \leq u_0 + h$ where biased MRFT output is given by $u_M = u_0 \pm h$. In this section we suggest algorithms to ensure: (1) condition $u_0 - h \leq f_d \leq u_0 + h$ is met while (2) avoiding excessive $h$ values that would result in impractical amplitudes of the oscillations like roll angles beyond $\pi$. The procedure for the design of these algorithms can cover any size and configuration of multicopter UAVs. Bias caused by the gravity makes identification of altitude dynamics particularly challenging. The elimination of the gravity bias without prior knowledge of a UAV’s total generated thrust and mass requires an algorithm that can handle the take-off state. For that we use a cascaded switched PID controller as shown in Fig. 5 by adding switching position $1^*$ to altitude. In the first stage ($S$ is at $1^*$), a PI controller is used for take-off:

$$u_z(t) = \begin{cases} 
K_p e_z(t) + \int_0^T K_i e_z(t) dt : \dot{z}_t \leq g + \delta \\
U_M(e_z(t), h_{alt}) + u_{z0} : C
\end{cases}$$  

where $\delta > 0$ is a bias factor to compensate for increased efficiency in take-off due to ground effect, $u_{z0}$ is the last output of the PI controller, and $C$ is a condition that is set permanently to true once the condition in the first line is violated. The first line of Eq. (31) corresponds to $1^*$ position of $S$ in Fig. 5 while the second line corresponds to switch position 1. The switching condition aims at minimizing the value of the bias present in the MRFT switching, which is perfectly achieved when $u_{z0}$ produces a thrust that causes the UAV to hover. It is not always possible to have a clean measurement of acceleration which was the case in our experimental setup, and therefore we were using position measurement. Though position measurement is lagged by a phase of $\pi$, take-off can be slowed down and a value of $u_{z0}$ close to hover thrust can still be achieved. For position measurement case, the condition in the first line of Eq. (31) would be $z_t \leq \delta_p$ instead of $\dot{z}_t \leq g + \delta$.

Because the presented controllers in Eq. (31) will be applied for all multicopter UAVs with the full model parameter’s range in Table I, suitable values of the take-off controller parameters need to be designed. The optimization decision variables are the parameters $K_p$, $K_i$, $\delta_p$, and $h$ presented in Eq. (31). A cost function has been designed to address the optimization of
where $J_{bias}$, $J_{time}$, $J_{amp}$ are the costs associated with bias in relay, system rise-time, and excited process amplitude respectively. $t_h$ is the duration MRFT switches high, $t_l$ is the duration MRFT switches low, and $t_{b0} = 0.5$ is a constant that corresponds to the case when MRFT switching is symmetric, i.e. $t_h = t_l = t_{b0} = t_h + t_l$. The value of $t_r$ is the time it takes to reach 90% of desired altitude from take-off (take-off is defined as passing $2cm$ altitude), and $t_{r0}$ corresponds to the desired maximum rise time which we chose to be $5s$. The value of $a_r$ corresponds to the steady-state amplitude of the self-excited oscillation due to MRFT. The value $a_{r0}$ corresponds to the desired maximum MRFT amplitude response and was chosen to be $0.3m$. The selection of cost function input parameters reflects essential practical requirements of the auto-tuner. We found that a severely biased relay might force motors to function near their operational extremes. A long take-off time is not desired and can be dangerous due to the fact that at take-off, MRFT is also running on roll and pitch where rotor tips might hit the floor. The last risk accounted for is associated with excessively large amplitudes of the response, which might lead to crashes or undesired aggressiveness. The collective responses of systems at the vertices of $D$ (actually $D$ resembles a cuboid in the system parameters space) was used to find $J_{opt}$. We found that this optimization problem is non-convex so that multiple initial points were tested. Nelder-Mead simplex algorithm realized by “fminsearch” function in MATLAB® has been used. The resulted optimal decision variables are:

$$h = 0.10746, \quad K_{eff}^f = 9.4969 \times 10^{-2},$$

$$K_{eff}^f = 9.8754 \times 10^{-3}, \quad \delta_p = 0.11984 \quad (33)$$

The responses of systems at the vertices of $D$ to the take-off algorithm with optimal take-off parameters can be seen in Fig. 10 in the appendix, where it can be clearly seen that all UAV variants are stable and operating within the physical limits. Note that some systems take very long to start taking-off compared to $t_{r0}$. This is due to the use of the position measurement for $\delta_p$ instead of acceleration measurement. Though the values presented in Eq. (33) guarantee stability, tuning from take-off can be made faster and smoother with smaller amplitude of the excited oscillations and faster take-off time. We find this possible with prior knowledge of the peak-thrust to weight ratio $C_{TW}$ of the UAV (the considered range of $C_{TW}$ based on Table II is 1.5 to 5). The value of $C_{TW}$ is easy to find (i.e. motor datasheet and a weighing scale) which makes the suggested auto-tuner still suitable for non-experts. The optimization of the take-off parameters can be run again while using systems in $\bar{D}$ which satisfies selected $C_{TW}$ value. Other control loops have smaller bias values and MRFT can be started immediately. To design suitable $h$ value for roll and pitch control loops we consider the parameters ranges in Table III and a limit of 30° on the maximum expected oscillation’s amplitude. A suitable selected $h$ value for roll and pitch is 5%.

The UAV is prone to drift side ways as the outer-loop control is not active during roll and pitch identification. This is not an issue when plenty of space is available, e.g. when the identification is performed in an open space. But when identification needs to be performed in a constrained space, a temporary outer loop controller is needed to bound the UAV to the vicinity of the take-off location until roll and pitch identification is completed. We suggest the following bounding controller composed of a hysteresis relay cascaded with a relay with dead-band:

$$u_{bb}(t) = \begin{cases} 0 & \text{if } |e(t)| \leq \epsilon_1 \land u_{bb}(t-)=0 \\ h_{o1}sgn(e(t)) & \text{if } |e(t)| > \epsilon_1 \land u_{bb}(t-) = 0 \\ h_{o2}sgn(e(t)) & \text{if } |e(t)| \leq \epsilon_2 \land u_{bb}(t-) \neq 0 \\ (h_{o1} + h_{o2})sgn(e(t)) & \text{if } |e(t)| > \epsilon_2 \\ \end{cases} \quad (34)$$

where $\{\epsilon_1, \epsilon_2\}$ are predefined position thresholds, and $h_{o1}$ and $h_{o2}$ are the amplitudes of the hysteresis relay and the relay with dead-band respectively. To guarantee that inner-loop MRFT oscillations reach steady-state, the threshold on position must be large enough such that the switching frequency of the bounding controller is considerably lower than that of the inner-loop oscillations. Table IV lists all parameters specific to the take-off stage that can be used across any multicopter UAV design and size within the ranges considered in Table II.

| Parameter | Value (Unit) | Comment |
|-----------|-------------|---------|
| $h_{alt}$ | 10.746 (% ESC) | See Eq. (33) |
| $K_{eff}^f$ | 9.4969 × 10^{-2} | See Eq. (33) |
| $K_{eff}^f$ | 9.8754 × 10^{-3} | See Eq. (33) |
| $\delta_p$ | 0.11984 | See Eq. (33) |
| $h_{alt}$ | 5.0 (% ESC) | Same for roll and pitch |
| $h_{o1}$, $h_{o2}$ | 0.05 (rad) | Same for roll and pitch |
| $\epsilon_1, \epsilon_2$ | 1.2 (m) | Selection depends on space constraints |

**VIII. SIMULATION AND EXPERIMENTAL RESULTS**

This section presents the simulation and experimental evaluation of the DNN-MRFT approach. Both evaluation methods follow the protocol demonstrated in Fig. 5. DNN-MRFT is first used to identify altitude and attitude model parameters $\{\hat{C}_{alt}, \hat{G}_{alt}\}$ and their corresponding optimal controllers $\{C_{alt}, \hat{G}_{alt}\}$. Then depending on the estimated inner-loop
dynamics, outer-loop distinguishing phase and DNN classifiers are selected and applied immediately to side-motion dynamics to identify $G_{\text{alt}}$ parameters and optimal controller $C^*_{\text{side}}$ gains. The analysis presented in this section assesses the DNN-MRFT approach for: accuracy and persistence of the parametric identification, the capability of the take-off controller to successfully lift and stabilize a UAV with no prior knowledge of system dynamics, and the performance of the auto-tuned controllers in aggressive trajectory following maneuvers.

Experimental tests were conducted on a variety of UAV designs as shown in Table V. For clarity, we only present results obtained on the QDrone in this section while the remaining experimental results can be found in Fig. 11 in the appendix and are also presented in the companion video. Optitrack’s motion capture system was used for UAVs localization.

TABLE V: Specifications of the UAV designs used in experimental analysis

| Processor       | QDrone [19] | DJI F550   | DJI F550 with extended arms |
|-----------------|-------------|------------|-----------------------------|
| Dimensions (cm) | 40 x 40 x 15 | 79 x 72 x 27 | 111 x 100 x 27 |
| Mass (kg)       | 1.5         | 2.09       | 3.38                        |
| $\{I_x, I_y, I_z\}$ | {0.010, 0.008, 0.015} | {0.031, 0.030, 0.052} | {0.005, 0.089, 0.156} |
| Number of propellers | 4           | 6          | 6                           |

A. Persistence and Accuracy of Identification

1) Simulation Results: The objective of simulation analysis is to evaluate the optimality of the DNN-MRFT auto-tuned controllers for the full parameters’ range presented in Table I with exact knowledge of the ground truth model. Five hundred different model parameter sets were randomly sampled from $D$ to form a testing set $D_{\text{test}}$. For each $G_T \in D_{\text{test}}$, the DNN-MRFT approach predicts a process $G_p$ and a controller $C^*_p$ for both the inner and outer control loops under randomly varied conditions of noise $n$ and bias $u_0$. We utilize the controller deterioration criterion $J_pT$ from Eq. (26) to quantify the accuracy of $G_p$ estimation. Additionally, the phase margin $PM$ of $C^*_p(s)G_T(s)$ is presented to assess the robustness of the synthesized controller. Average and worst-case results on the entire testing set are reported in Table VI. The worst-case phase margin of the side-motion control loop is reported as the average of the worst-case prediction of each of the 48 outer-loop DNNs. Results are reported with two different gain scaling methods: the first method uses the DF approximation with $\zeta = \frac{1}{2}$ to approximate the gain of the unknown system, and the second one uses the exact scaling method described in Section VI. In Table VI, the case when the gain is normalized (does not include errors introduced by gain scaling) is also presented for comparison.

The results in Table VI demonstrate two features of DNN-MRFT. First it demonstrates stability by the guarantee of a minimum phase margin value, which agrees with the original formulation of the MRFT and coordinated test and tuning [4]. This is clearly indicated by the minimum phase margin values that are always positive. Note that the average $J_{\text{att}}$ and $J_{\text{side}}$ values when exact gain scaling is used are negative. Negative joint cost values occur whenever tuning resulted in a phase margin that is lower than the minimum phase margin constraint used in parameter space discretization process and shown in Table I. The second feature of DNN-MRFT is that it results in near-optimal performance for the full range of $D$. Average controller deterioration cost for both inner-loop and outer-loop dynamics are near zero, with the worst-case deterioration being 15.68% for the side-motion auto-tuning case. Furthermore, our proposed DF gain scaling approach results in a generally lower deterioration than the DF approximation, especially when considering worst-case results. These results show an inherent advantage of using DNN-MRFT: it employs nonlinear methods (i.e. MRFT and DNN) to obtain a linear description of the underlying dynamics, which preserves the useful properties of linear systems such as the measurable robustness and performance margins.

2) Experimental Results: This section assess the DNN-MRFT’s experimental performance and persistence for synthesizing controller parameters for attitude and side-motion dynamics. Starting from a hovering state, Fig. 6 shows the two stages of the DNN-MRFT auto-tuning procedure. The identified model parameters were $G_{\text{att}} = \{K_{\text{eq,att}} = 1.72, T_{\text{prop}} = 0.0150, T_1 = 0.2005, \tau_{\text{att}} = 0.0250\}$ and $G_{\text{side}} = \{K_{\text{eq,side}} = 4.34, T_2 = 0.3812, \tau_{\text{side}} = 0.1\}$ with the corresponding ISE optimal PD controllers $C^*_p = \{K_c = 1.72, K_d = 0.15\}$ and $C^*_p = \{K_c = 1.89, K_d = 0.56\}$. The online synthesized controllers stabilize the UAV and smoothly drive the UAV to origin point.

Fig. 6: DNN-MRFT auto-tuning experiment for the inner and outer control loops. (a) MRFT is performed on the inner-loop until steady-state oscillations are acquired. The last cycle is passed to a DNN that predicts model and controller parameters. (b) DNN-MRFT is repeated for the identification and tuning of side motion dynamics. (c) The UAV is driven back to origin by the auto-tuned controllers.

To evaluate the persistence of system identification, the same experiment in Fig. 6 was repeated five times. In all experiments, the DNN-MRFT approach identifies identical model parameters, which transcribes into a $J_{\text{cross}} = 0\%$ controller deterioration joint cost across the identified model parameters from all experiments. The auto-tuning experiment was also conducted with an artificial delay of 0.025 seconds added to the side-motion control loop; and the identified
TABLE VI: Simulation results of the DNN-MRFT approach on 500 randomly selected processes in D

| Gain scaling          | Average $J_{att}$ | Maximum $J_{att}$ | Average $PM_{att}$ | Minimum $PM_{att}$ | Average $J_{side}$ | Maximum $J_{side}$ | Average $PM_{side}$ | Average Minimum $PM_{side}$ |
|-----------------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|
| Normalized gain       | 0.45%             | 5.10%             | 19.65             | 14.21             | -0.19%            | 4.91%             | 19.70             | 17.18             |
| DF gain approximation | 1.45%             | 9.68%             | 20.04             | 14.20             | 2.04%             | 57.44%            | 19.46             | 16.11             |
| Exact gain scaling    | -1.77%            | 7.14%             | 18.41             | 11.24             | -1.73%            | 15.68%            | 18.41             | 14.23             |

TABLE VII: Full system identification and tuning results on the QDrone.

| Control Loop | Full System Identification | Single Loop Identification |
|--------------|----------------------------|---------------------------|
|              | $K_c$ | $K_d$ | $K_c$ | $K_d$ |
| Roll         | 1.63  | 0.14  | 1.72  | 0.15  |
| Pitch        | 1.16  | 0.13  | 1.36  | 0.13  |
| Altitude     | 62.92 | 9.63  | 65.05 | 10.49 |
| x            | 1.69  | 0.50  | 1.85  | 0.55  |
| y            | 1.79  | 0.53  | 1.89  | 0.56  |

to the origin point and hold it at hover. The resulting controller parameters for the full system auto-tuning experiment are presented in Table VII along with those obtained from single loop auto-tuning experiments. The similarity between the two sets of controller parameters further indicates the persistence of DNN-MRFT and validates its capability for full-system auto-tuning without prior knowledge of system dynamics.

C. Trajectory Tracking Performance

optimal controller parameters being $C_{side-delay}^* = \{K_c = 1.58, K_d = 0.56\}$. The identification results across five repetitions of the experiment with added calibrated delay were persistent with $J_{cross} = 0\%$. The controller performance deterioration cost across the two sets of experiments (with and without the added delay) was also persistent at $J_{cross} = 3\%$.

B. Takeoff and Full System Identification

Following the formulation of the take-off controllers presented in Section VII, we evaluate the capability of DNN-MRFT for full system identification and auto-tuning of UAV’s starting from a landing state with no prior knowledge of system dynamics. The take-off and auto-tuning procedure is carried out in two subsequent stages: inner-loops and outer-loops identification stages. In the first stage, DNN-MRFT with the take-off controller designed in Section VII is applied to the altitude loop, corresponds to $S$ at position 1 in Fig. 3, while placing $S$ at position 1 for all other loops. Once the condition $C$ in Eq. (31) is met switch $S$ switches to position 1 for altitude case as well. Once steady-state oscillations are detected on each inner-loop, a DNN identifies optimal controller parameters that are instantaneously applied to the corresponding control loop. Once inner-loop identification is complete, the switch $S$ switches to position 2 such that DNN-MRFT is carried out in the same manner for the identification of side-motion loops. Similarly, once steady-state oscillations are detected on outer-loops, the corresponding DNN network identifies the dynamics and applies optimal controller by switching $S$ to position 3.

Fig. 9 (a, b) shows the profile of all control loops during the full-system auto-tuning experiment. The take-off controller successfully lifts the UAV while the MRFT controller stabilizes all control-loops. Once the identification phase is complete, the synthesized controllers smoothly drive the UAV back

Fig. 7: Simulation and Experimental results of the DNN-MRFT auto-tuned controller on a figure-eight trajectory.

1) Figure Eight Trajectory: We assess the performance of the auto-tuned controller parameters shown in Table VII on a figure-eight trajectory as a widely used benchmark in literature. For this purpose, we use the concept of minimum-snap trajectory optimization [32], [50] to design a figure-eight trajectory with a period of 5.5 seconds, and we provide position, velocity, and angle references for tracking. The full simulation and experimental profile of the trajectory following maneuver can be observed in Fig. 7 and a quantification of the resultant errors is provided in Table VIII. The similarity between the simulation and experimental profile indicates accuracy in terms of identified model parameters, which in turn implies the optimality of the auto-tuned controllers.

We benchmark DNN-MRFT’s performance on figure-eight trajectory tracking against two recent approaches: the S2R approach based on RL [13], and differential flatness for accurate trajectory tracking approach [51]. We find the problem of S2R is closest to ours in literature, yet the nature of adaptation is quite different. We did not find other adaptive approaches applied to trajectory tracking of multirotor UAVs to compare with. We evaluate DNN-MRFT on two variants

1As reported in [13]

2Approximated from a figure-eight trajectory with a similar maximum speed of around 3.2m/s
TABLE VIII: Experimental evaluation of tracking errors $\|e(t)\|$ in meters for two specifications of a figure-eight trajectory.

| Approach          | Trajectory specs              | Size:          | Period:   |
|-------------------|------------------------------|----------------|-----------|
|                   |                              | $2 \times 1m$ | 5.5s      |
| Ours (simulation) | $l_{\text{mm}} = 0.28m$      | $0.039$        |           |
| S2R (simulation)  | $l_{\text{mm}} = 0.13m$     | $0.023$        |           |
| Faessler et al.   | $l_{\text{mm}} = 0.13m$     | $0.04$         |           |

of the figure-eight trajectory. The first figure-eight trajectory we chose to perform is identical to the one reported in S2R results [13], while the second trajectory is similar in size and maximum speed to the one reported in [51]. The comparison does not take into account differences in experimental setup and physical capabilities of each platform. We could not achieve speeds higher than 3.2m/s due to platform mechanical limits. Table VIII shows comparative results for the average euclidean position error $\|e(t)\|$ during trajectory following. The DNN-MRFT approach achieves tracking performance that is an order of magnitude better than the S2R approach [13]. We have also found that the tracking performance is better, or at least on par with manually tuned trajectory tracking controllers. The performance achieved by [51] relied on manual tuning and used a tailored controller to the specific trajectory being followed, which clearly requires a lot of time resources for compensation and optimization. This further confirms our conclusion that results achieved by DNN-MRFT are the new state of the art for real-time adaptive methods in trajectory tracking.

![Fig. 8: The vertical window passage maneuver: the UAV passes through the window with a near 90° roll angle.](image)

2) **Vertical Window Passage:** The performance of the DNN-MRFT synthesized controllers have been evaluated for an aggressive vertical-window passage maneuver as the one shown in Fig. 8. We designed a minimum snap trajectory to pass through a vertically aligned window with a roll angle of 90°. The attitude constraint was implemented by enforcing a specific relative value between the $x$, $y$, and $z$ acceleration components of the designed trajectory. The target window is of size $0.3 \times 0.6m$, leaving only 0.1 m of vertical clearance and 0.075m of horizontal clearance for the QDrone. Given this clearance and the geometry of QDrone, the minimum possible window passage velocity is 1.45 m/s. This constraint arises from the under-actuated nature of multi-rotor UAVs; which limits the attainable vertical thrust to near-zero at high pitching or rolling angles leaving the UAV at a state of free-falling. We designed two different trajectories to pass through the window at different speeds of 2.75 m/s and 1.75 m/s, with the second nearing the geometrical limit of feasible trajectories.

Fig. 9 shows the full state profile during both vertical window passage trajectories proceeded by the DNN-MRFT takeoff and auto-tuning phase. The auto-tuned controllers successfully maneuver the UAV and achieve an almost 90° rolling angle at the window location. The capability of the auto-tuned controllers to execute this maneuver despite the rigid clearance constraints and the low speeds indicates that these controllers are indeed near-optimal, and constitute state-of-the-art in terms of real-time multirotor UAV auto-tuning.

![Fig. 9: Full system DNN-MRFT auto-tuning experiment starting from a landing state without prior knowledge of system dynamics followed by two aggressive maneuvers of vertical window passage at different speeds. (a) The UAV takes off and performs MRFT on altitude and attitude control loops. Once steady-state behaviour is observed for each control-loop, DNN-MRFT identifies the optimal controller parameters which are directly applied to control the plant. (b) DNN-MRFT is performed on both side motion control loops and tunes controllers accordingly. (c) The online tuned controllers smoothly drive the UAV back to origin and hold it at hover. (d) Vertical Window maneuver at 2.75 m/s. (e) Vertical Window maneuver at 1.75 m/s. A video of the full experiments can be seen in [5].](image)
the best of our knowledge, DNN-MRFT is the first UAV full auto-tuning approach to successfully perform such aggressive maneuvers without any prior knowledge of system dynamics. A video demonstration of the DNN-MRFT auto-tuning capability for the three different UAV designs listed in Table 6_3 and vertical narrow window passage can be found in [5].

IX. CONCLUSION

This paper extended the capabilities of a novel real-time system identification approach presented in [3] referred to as DNN-MRFT. DNN-MRFT uses oscillations excited by MRFT at a specific phase, called the distinguishing phase, to identify dynamical model parameters using a DNN classifier. DNN-MRFT was extended for higher order dynamical systems in a cascaded manner. Such extension was used to identify linearized system dynamics of side translational movement having a relative degree of five with a time delay in real-time. Accuracy of system gain identification was improved by using exact solutions of Lure’s systems from simulation data. DNN-MRFT precision and accuracy were validated both in simulation and experimentation. A take-off controller suitable for a wide range of UAV designs demonstrated experimental real-time identification capability. The generalized take-off controller with DNN-MRFT was verified on three different UAV designs. As a result of identification, aggressive maneuvers were possible without any form of hand tuning or biasing. The results were benchmarked against state-of-the-art and showed outstanding performance as a result of using DNN-MRFT.

For future work, we aim to evaluate DNN-MRFT for systems with other dynamical properties. We see that DNN-MRFT can be extended to MIMO systems with strong cross-loop couplings. Also, DNN-MRFT might be successful when applied to dynamical systems with large time delays. We are also aiming to handle systems with varying dynamics, where DNN-MRFT can be re-run to adapt to the change in dynamics. In such case, the identification test can be re-run to either a single or multiple control loops at which the change in dynamics occur. The time such adaptation would consume is dependent on the number of control loops to re-tune and the overall dynamics of the UAV, but should not exceed the time consumed by the full take-off and identification tests, which was limited to 15 seconds in our experimental setup. Such extensions require better understanding of the theoretical basis of information embedded on dynamical systems.

ACKNOWLEDGMENT

We would like to thank Quanser team for their generous and timely support. We also thank Eng. Yehya Farhoud for his help in preparing the experimental setup.

REFERENCES

[1] K. J. Astrom and B. Wittenmark, Adaptive Control, 2nd ed. Addison Wesley, 1995.
[2] K. Guo, Y. Pan, and H. Yu, “Composite learning robot control with friction compensation: A neural network-based approach,” IEEE Transactions on Industrial Electronics, vol. 66, no. 10, pp. 7841–7851, 2019.
[3] A. Ayyad, M. Chehadeh, M. I. Awad, and Y. Zweiri, “Real-time system identification using deep learning for linear processes with application to unmanned aerial vehicles,” IEEE Access, vol. 8, pp. 122539–122553, 2020.
[4] I. Boiko, “Modified relay feedback test (mrf) and tuning of pid controllers,” in Non-parametric Tuning of PID Controllers: A Modified Relay-Feedback-Test Approach. London: Springer London, 2013, pp. 25–79. [Online]. Available: https://doi.org/10.1007/978-1-4471-4465-6_3
[5] A. Ayyad, M. Chehadeh, P. Silva, M. Wahbah, O. AbdulHay, I. Boiko, and Y. Zweiri, Multirotors From Takeoff to Real-Time Full Identification Using MRFT and DNNs, 2021. [Online]. Available: https://youtu.be/07RtnjTRM
[6] B. Xu, “Composite learning finite-time control with application to quadrotors,” IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 48, no. 10, pp. 1806–1815, 2018.
[7] O. Mofid and S. Mobayen, “Adaptive sliding mode control for finite-time stability of quad-rotor uavs with parametric uncertainties,” ISA Transactions, vol. 72, pp. 1–14, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0019057817306171
[8] T.-T. Tran, S. S. Ge, and W. He, “Adaptive control of a quadrotor aerial vehicle with input constraints and uncertain parameters,” International Journal of Control, vol. 91, no. 5, pp. 1140–1160, 2018. [Online]. Available: https://doi.org/10.1080/00207179.2017.1390957
[9] S. Nadda and A. Swarup, “On adaptive sliding mode control for improved quadrotor tracking,” Journal of Vibration and Control, vol. 24, no. 14, pp. 3219–3230, 2018. [Online]. Available: https://doi.org/10.1177/1077546317703541
[10] A. P. Schoellig, F. L. Mueller, R. Raffaele D’Andrea, A. P. Schoellig, F. L. Mueller, R. D’Andrea, and R. D’Andrea, “Optimization-based iterative learning for precise quadrocopter trajectory tracking,” vol. 33, pp. 103–127, 2012. [Online]. Available: link.springer.com/article/10.1007/s10514-012-9283-2
[11] F. Berkenkamp, A. P. Schoellig, and A. Krause, “Safe controller optimization for quadrotors with gaussian processes,” CoRR, vol. abs/1509.01066, 2015. [Online]. Available: http://arxiv.org/abs/1509.01066
[12] N. O. Lambert, D. S. Drew, J. Yaconelli, S. Levine, R. Calandra, and K. S. Pister, “Low-level control of a quadrotor with deep model-based reinforcement learning,” IEEE Robotics and Automation Letters, vol. 4, no. 4, pp. 4224–4230, 2019.
[13] A. Molchanov, T. Chen, W. Höng, J. A. Preiss, N. Ayaniyan, and G. S. Sukhatme, “Sim-to-(multi-)real: Transfer of low-level robust control policies to multiple quadrotors,” in 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2019, pp. 59–66.
[14] Q. Li, J. Qian, Z. Zhu, X. Bao, M. K. Helwa, and A. P. Schoellig, “Deep neural networks for improved, impromptu trajectory tracking of quadrotors,” in 2017 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2017, pp. 5183–5189.
[15] P. Abbeel, A. Coates, and A. Y. Ng, “Autonomous helicopter aerobatics through apprenticeship learning,” The International Journal of Robotics Research, vol. 29, no. 13, pp. 1608–1639, 2010.
[16] M. S. Chehadeh and I. Boiko, “Design of rules for in-flight non-parametric tuning of PID controllers for unmanned aerial vehicles,” Journal of the Franklin Institute, vol. 356, no. 1, pp. 474–491, jan 2019. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0020085418306604
[17] P. Poksawat, L. Wang, and A. Mohamed, “Automatic tuning of attitude control system for fixed-wing unmanned aerial vehicles,” IET Control Theory & Applications, vol. 10, no. 17, pp. 2233–2242, 2016.
[18] W. Giernacki, “Iterative learning method for in-flight auto-tuning of uv controllers based on basic sensory information,” Applied Sciences, vol. 9, no. 4, p. 648, 2019.
[19] W. Giernacki, D. Horla, T. Băca, 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, 2019.
[20] D. A. Tesch, D. Eckhard, and W. C. Guarienti, “Pitch and roll control of a quadcopter using cascade iterative feedback tuning,” IFAC-PapersOnLine, vol. 49, no. 30, pp. 30–35, 2016.
[21] D. Howard, “A platform that directly evolves multicopter controllers,” IEEE Transactions on Evolutionary Computation, vol. 21, no. 6, pp. 943–955, 2017.
[22] W. Du, J. Fang, and C. Miao, “Frequency-domain system identification of an unmanned helicopter based on an adaptive genetic algorithm,” IEEE Transactions on Industrial Electronics, vol. 61, no. 2, pp. 870–881, 2014.
Fig. 10: Response of processes at the vertices of $D_{alt}$ for the takeoff controller with parameters in Eq. (33). The parameters at the top of each graph represents the vector $[\tau, T_{\text{prop}}, T_1, C_{TW}, J_{\text{tot}}]$. Vertical dashed line shows $t_{r0}$ and the horizontal ones show $a_{r0}$. 
Fig. 11: Full system DNN-MRFT auto-tuning experiment starting from a landing state without prior knowledge of system dynamics applied to DJI F550 custom hexarotor UAV (left column) and DJI F550 custom hexarotor UAV with extended arms (right column). In period (a) identification of inner loops parameters was performed and in period (b) identification was performed on outer loop parameters. After auto-tuning, the multirotor UAVs followed a trajectory resembling a square. Both auto-tuning experiments are shown in the video in [5].
Abdulla Ayyad received his MSc in Electrical Engineering from The University of Tokyo in 2019 where he conducted research in the Spacecraft Control and Robotics laboratory. He is currently a Research Associate in Khalifa University Center for Autonomous Robotic Systems (KUCARS) and the Aerospace Research and Innovation Center (ARIC) working on several robot autonomy projects. His current research targets the application of AI in the fields of perception, navigation, and control.

Mohamad Chehadeh received his MSc in Electrical Engineering from Khalifa University, Abu Dhabi, UAE, in 2017. He is currently with Khalifa University Center for Autonomous Robotic Systems (KUCARS). His research interest is mainly focused on identification, perception, and control of complex dynamical systems utilizing the recent advancements in the field of AI.

Pedro Henrique Silva received the B.Sc. degree in control engineering from the Federal University of Itajubá - Brazil, in 2017, and is currently pursuing his M.Sc. in AI from the University of Bath. He started in the field of UAVs first as a participant on international competitions and later professionally in the agriculture industry. He is currently a Research Associate with the Khalifa University Center for Autonomous Robotic Systems (KUCARS). His research interests include perception, control and AI applied to autonomous vehicles.

Mohamad Wahbah received his BSc and MSc in Electrical Engineering from Khalifa University, Abu Dhabi, UAE. He is currently working as a research associate at Khalifa University Center for Autonomous Robotic Systems (KUCARS). Currently he is involved in perception and estimation solutions for autonomous robots.

Oussama Abdul Hay received his MSc in Thermal Power and Fluids Engineering from The University of Manchester, UK. He is currently working as a research associate at Khalifa University Center for Autonomous Robotic Systems (KUCARS). His research interests are in the fields of perception and visual servoing for autonomous robots.

Igor Boiko received his MSc, PhD and DSc degrees from Tula State University and Higher Attestation Commission, Russia. His research interests include frequency-domain methods of analysis and design of nonlinear systems, discontinuous and sliding mode control systems, PID control, process control theory and applications. Currently he is a Professor with Khalifa University, Abu Dhabi, UAE.

Yahya Zweiri received the Ph.D. degree from the 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. His central research focus is interaction dynamics between unmanned systems and unknown environments by means of deep learning, machine intelligence, constrained optimization, and advanced control. He has published over 100 refereed journal and conference papers and filed six patents in USA and U.K. in unmanned systems field.