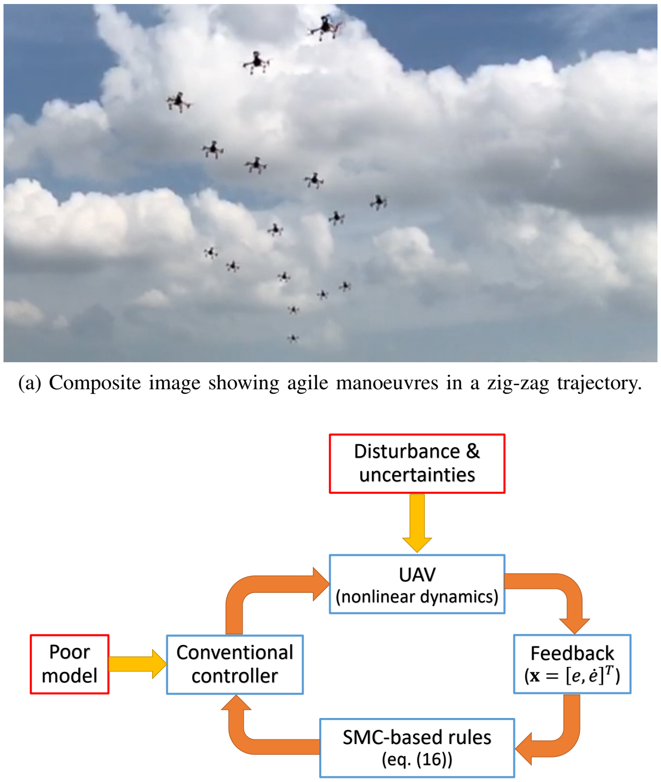
An Artificial Neural Network-Assisted Controller for Fast and Agile UAV Flight: Onboard Implementation and Experimental Results

Siddharth Patel[1], Andriy Sarabakha[2], Dogan Kircali[1], Giuseppe Loianno[3] and Erdal Kayacan[4]

***Abstract*— In this work, we address fast and agile manoeuvre control problem of unmanned aerial vehicles (UAVs) using an artificial neural network (ANN)-assisted conventional controller. Whereas the need for having almost perfect control accuracy for UAVs pushes the operation to boundaries of the performance envelope, safety and reliability concerns enforce researchers to be more conservative in tuning their controllers. As an alternative solution to the aforementioned trade-off, a reliable yet accurate controller is designed for the trajectory tracking of UAVs by learning system dynamics online over the trajectory. What is more, the proposed online learning mechanism helps us to deal with unmodelled dynamics and operational uncertainties. Experimental results validate the proposed approach and show the superiority of our method compared to conventional controller for fast and agile manoeuvres, at speeds as high as 20m/s. An onboard implementation of the sliding mode control theory-based adaptation rules for the training of the proposed ANN is computationally efficient which allows us to learn system dynamics and operational variations instantly using a low-cost and low-power computer.**

I. INTRODUCTION

Unmanned aerial vehicles (UAVs) are starting to play an important role in transportation [1], manipulation [2], and search and rescue [3]. The ability to fly in an agile and aggressive manner is useful in tasks with time constraints, in densely populated and cluttered environments, especially in search and rescue scenarios. Similar to any other robotics application, robustness and performance trade-off exists in UAV applications where researchers mostly prefer safe and robust controller tuning when dealing with more aggressive controllers. In this work, we show that an artificial neural network (ANN)-assisted control method, which enables fast and aggressive manoeuvres, allows us to obtain accurate trajectory tracking results without compromising the robustness and safety in the system. We also demonstrate that although the conventional controller – proportional-derivative (PD) in this study – is not tuned thoroughly, the proposed adaptation algorithm allows to learn online the system dynamics, internal and external uncertainties, and improve the overall trajectory tracking performance throughout the fast and agile manoeuvres.



(a) Composite image showing agile manoeuvres in a zig-zag trajectory.

(b) Control logic of the proposed approach.

Fig. 1. Control architecture for fast and agile UAV flight.

In the literature, UAV operation near the boundaries has always been an alluring research topic, like performing multi-flips using a simple learning strategy and the first-principles model [4]. In [5], a manoeuvre regulation perspective follows a geometric path, as assigned, with a certain velocity. Since the path to be followed is not a time-based reference state, unlike in trajectory tracking, a linear quadratic regulator-based controller ensures the exact path-following to perform such *space-dependent* manoeuvres. As an alternative solution, the aerodynamic effects of blade flapping and thrust variations on a rotor at higher angles of attack are studied and used for developing control techniques for operating at high-speed aggressive manoeuvres [6]. The authors present a novel feedback linearization controller to take into account such aerodynamic disturbances. A hardware solution to the mentioned problem, used in [7], is the implementation of the variable pitch rotors. This method expands the normal rotor

[1] Siddharth Patel and Dogan Kircali are with School of Electrical and Electronic Engineering (EEE), Nanyang Technological University (NTU), 50 Nanyang Avenue, Singapore, pate0006@e.ntu.edu.sg, dkircali@ntu.edu.sg

[2] Andriy Sarabakha is with School of Mechanical and Aerospace Engineering (MAE), Nanyang Technological University (NTU), 50 Nanyang Avenue, Singapore, andriy001@e.ntu.edu.sg

[3] Giuseppe Loianno is with the New York University, Tandon School of Engineering, 6 MetroTech Center, 11201 Brooklyn NY, USA, loiannog@nyu.edu

[4] Erdal Kayacan is with Department of Engineering, Aarhus University, Aabogade 34, Aarhus, 8000, Denmark, erdal@eng.au.dk

operation regime by varying the blade pitch which provides better thrust vectoring to achieve agile UAV manoeuvres.

Feasible aggressive trajectories emulating constrained indoor environment are designed in [8], [9]. The algorithm generates trajectories in real-time such that it ensures safe passage through corridors and satisfies constraints on velocities and accelerations. A nonlinear controller based on the changing dynamics of the UAV and errors in the model is developed which ensures the tracking of desired states in the three-dimensional (3D) space. A simple model describing the essential dynamics of the system is used in an iterative learning algorithm to perform an aggressive motion [10]. The knowledge obtained from the successful trajectories is used to reduce the transients when performing similar subsequent manoeuvres.

In lieu of the literature above, other groups analyze the perception and planning aspects to enable agile navigation using onboard sensors such as laser scanner with fixed-wing aircraft [11], stereo cameras [12], a single camera and IMU [13]. However, our work focuses on the control part to cope with uncertainties and disturbances deriving from unmodelled dynamics and external effects. We show that an accurate time-based trajectory tracking for agile flights is achieved in the outdoor environment, as seen from the composite image in Fig. 1a, with steady wind gusts. Moreover, the proposed ANN structure is computationally inexpensive to be implemented on a low-cost onboard computer. The performance of the ANN-assisted controller outmatches the conventional PID controller.

The key aspects of this work can be enlisted as follows:

- To the best of our knowledge, this is the first time an ANN is used for accurate trajectory tracking in high-speed and agile manoeuvres.
- A comparison of the ANN with classical model-based controllers (PID) is carried out showing the superiority of the proposed approach.
- Outdoor real-time high-speed flight tests are performed using the real-time kinematic global positioning system (RTK GPS) for localization.

This work is structured as follows. In Section II, the dynamic model of the coaxial hexacopter is given. Section III gives the description of the control architecture of the proposed controller. The experimental results are shown in Section IV. A brief conclusion of this work is given in Section V.

II. Coaxial Hexacopter Model

In this study, a custom-made coaxial hexacopter with six rotors attached to three arms is used as the experimental platform. The differential thrust produced by each motor is used for the basic control of the hexacopter. Figure 2 shows the considered reference frames and sign conventions for the forces (F_i), torques (τ_i), and rotational speeds (Ω_i) of the rotors, where i is the rotor number. The three rotors on the top rotate clockwise and the bottom three rotate counter-clockwise. A brief description of the dynamics and kinematics of the coaxial hexacopter is presented in [14].

The world fixed inertial frame or Earth frame is $\mathcal{F}_E = \{\vec{\mathbf{x}}_E, \vec{\mathbf{y}}_E, \vec{\mathbf{z}}_E\}$ and the body frame is $\mathcal{F}_B = \{\vec{\mathbf{x}}_B, \vec{\mathbf{y}}_B, \vec{\mathbf{z}}_B\}$. The dynamics of the system are defined by assuming the UAV as a rigid body with origin at the centre of gravity of the UAV. The rolling (τ_p), pitching (τ_q) and yawing (τ_r) moments, as shown in the figure, are maintanied by the rotors. Considering the dynamics, the system is underactuated, since there are four control inputs ($T,\ \tau_p,\ \tau_q,\ \tau_r$) which can be summarized as:

$$T = F_1 + F_2 + F_3 + F_4 + F_5 + F_6, \tag{1}$$

where T is the total thrust produced by the six rotors at any given time. The moments acting on the UAV with moment arms $l_1 = l$, $l_2 = \frac{\sqrt{3}}{2}l$ and $l_3 = \frac{1}{2}l$, are given by:

$$\begin{bmatrix} \tau_p \\ \tau_q \\ \tau_r \end{bmatrix} = \begin{bmatrix} (F_5 + F_6 - F_1 - F_2)\frac{\sqrt{3}}{2}l \\ (F_3 + F_4)l - (F_1 + F_2 + F_5 + F_6)\frac{1}{2}l \\ \tau_1 + \tau_3 + \tau_5 - \tau_2 - \tau_4 - \tau_6 \end{bmatrix} \tag{2}$$

The position and orientation of the UAV are defined by vectors, $\begin{bmatrix} x & y & z \end{bmatrix}^T \in \mathbb{R}^3$ and $\begin{bmatrix} \phi & \theta & \psi \end{bmatrix}^T \in \mathbb{R}^3$ in $\mathcal{F}_E$, respectively. The time derivatives of these state vectors gives the translational and rotational kinematic equations, which are described as:

$$\begin{cases} \mathbf{v} & = \mathbf{R}\mathbf{v}_B \\ \omega & = \mathbf{T}\omega_B \end{cases}, \tag{3}$$

where $\mathbf{v} = \begin{bmatrix} \dot{x} & \dot{y} & \dot{z} \end{bmatrix}^T$, $\omega = \begin{bmatrix} \dot{\phi} & \dot{\theta} & \dot{\psi} \end{bmatrix}^T$, $\mathbf{v}_B$ and ω_B are linear and angular velocities given by vectors $\begin{bmatrix} u & v & w \end{bmatrix}^T \in \mathbb{R}^3$ and $\begin{bmatrix} p & q & r \end{bmatrix}^T \in \mathbb{R}^3$ in $\mathcal{F}_B$, respectively. $\mathbf{R}$ and $\mathbf{T}$ are the rotation and transformation matrices, respectively.

The rigid body dynamic equations are derived using the Newton-Euler formulation in the body frame and translate to the final form as:

$$\begin{cases} m\dot{\mathbf{v}}_B & = F - (\omega_B \times m\mathbf{v}_B) \\ \mathbf{I}\dot{\omega}_B & = \tau - (\omega_B \times \mathbf{I}\ \omega_B) \end{cases}, \tag{4}$$

where m is mass and $\mathbf{I}$ is the inertia matrix and is generalized by $\mathbf{I} = diag\,(I_x,\ I_y,\ I_z)$.

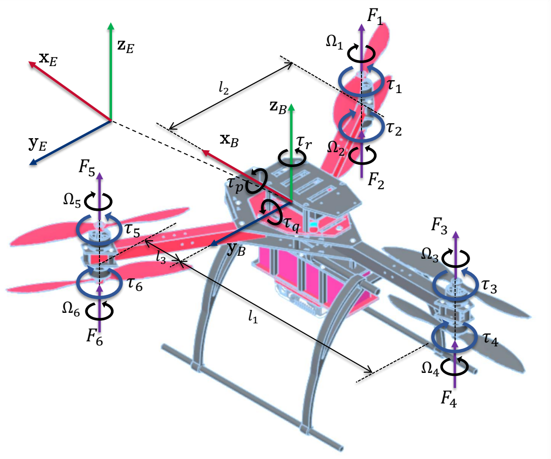


Fig. 2. Coordinate reference frame and sign conventions.

External force F and torque τ are expressed as:

$$F = \begin{bmatrix} 0 \\ 0 \\ T \end{bmatrix} - \begin{bmatrix} -mg\sin\theta \\ mg\cos\theta\sin\phi \\ mg\cos\theta\cos\phi \end{bmatrix}, \quad \tau = \begin{bmatrix} \tau_x \\ \tau_y \\ \tau_z \end{bmatrix}, \tag{5}$$

Thus, the kinematic and dynamic differential equations in (3) and (4) describe the general mathematical model of the hexacopter.

The forces and reaction torques exerted by the rotors, rotating at Ω_i angular velocities, can be formulated as:

$$\begin{cases} F_i &= K_F\Omega_i^2 \\ \tau_i &= K_\tau\Omega_i^2 \end{cases}, \tag{6}$$

where K_F and K_τ are termed as force and torque coefficients and are modeled based on the motor-propeller combination.

III. CONTROL SCHEME

A. Artificial Neural Networks

A basic neural network imitates the working principle of a human brain. ANNs are highly regarded for their learning ability from input-output data. An inter-connected structure of neurons receives an input, processes it and generates an output depending on the input and internal state. In a general ANN structure, the neurons are linked together, as shown in Fig. 3. The neural network is organized into the input, hidden and output layers. Between the output of each neuron and the input to the next neuron weights (v_i or w_i) are applied, which is updated in the learning process by a set of rules. The evaluation of the distance from the sliding-mode manifold determines these weights.

B. Learning-Based Controller

In the proposed control scheme, the ANN works in parallel with a conventional PD controller which is used as a feedback controller to provide sufficient time for the initialization of the learning process of ANN [16]. Moreover, ANN will learn the UAV dynamics and take over the control responsibility of the system. With its adaptive learning rates, ANN is very fast to learn and can instantaneously contribute to a better performance, i.e., an accurate trajectory tracking.

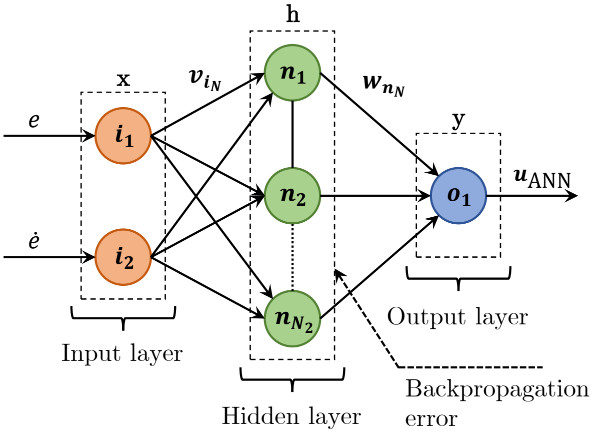


Fig. 3. Structure of the ANN.

The overall control input u to the system to be controlled is defined by:

$$u = u_{\text{PD}} - u_{\text{ANN}}, \tag{7}$$

where u_{PD} and u_{ANN} are the control signals produced by PD controller and ANN controller, respectively. The PD control law is written in the following form:

$$u_{\text{PD}} = k_p e + k_d \dot{e}, \tag{8}$$

where e and $\dot{e}$ are the feedback error and its time derivative, respectively, while k_p and k_d are some positive constants corresponding to proportional and derivative gains, respectively.

The position error e and its time derivative $\dot{e}$ are the two inputs provided to the proposed ANN controller, i.e., $\mathbf{x} = \begin{bmatrix} e & \dot{e} \end{bmatrix}^T$, which are the same inputs as to PD. One output generates the control signal u_{ANN}, i.e., $y = u_{\text{ANN}}$. The ANN architecture used for the design of the controller in this work has: two input neurons ($N_1 = 2$), one output neuron ($N_3 = 1$) and three neurons ($N_2 = 3$) in the hidden layer. The hidden layer defines the learning capabilities and the complexity of the ANN. The choice of number of neurons in the hidden layer is clarified from the simulation studies in Section IV. One may note from Fig. 4 that the control output u seems one-dimensional which can be misleading. The same control structure is used to generate all the four control signals described in (4), but only one is shown here for the sake of simplicity and to avoid repetition.

The output control signal from ANN can be computed as the weighted sum of each input:

$$u_{\text{ANN}} = \frac{\sum_{i=1}^{3} h_i w_i}{\sum_{i=1}^{3} h_i} = \sum_{i=1}^{3} \bar{h}_i w_i, \tag{9}$$

where $\bar{h}_i$ is the normalized value of the output from the i^{th} neuron in the hidden layer:

$$\bar{h}_i = \frac{h_i}{\sum_{i=1}^{3} h_i}. \tag{10}$$

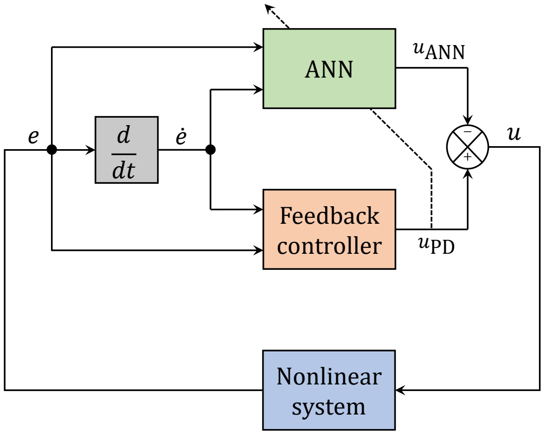


Fig. 4. Schematics of ANN-based controller.

The activation functions in (9) are linear. Based on the control scheme in Fig. 4, it is assumed that the incoming signals, $e(t)$ and $\dot{e}(t)$, and their time derivatives, $\dot{e}(t)$ and $\ddot{e}(t)$, cannot have infinite values [17]. Thus, they can be considered bounded, i.e.:

$$\begin{cases} |e(t)| \leq B_e \\ |\dot{e}(t)| \leq B_{\dot{e}} \qquad \forall t, \\ |\ddot{e}(t)| \leq B_{\ddot{e}} \end{cases} \tag{11}$$

where the positive constants $B_e > 0$, $B_{\dot{e}} > 0$ and $B_{\ddot{e}} > 0$ are assumed to be some known values. Similarly, the weight coefficient can be considered also bounded, i.e.:

$$\begin{cases} |w_i(t)| & \leq B_w \quad \forall t \\ |v_i(t)| & \leq B_v \quad \forall t \end{cases}, \tag{12}$$

where $B_w > 0$ and $B_v > 0$ are some positive constant. From (11) and (12), it is evident that $u_{\text{ANN}}(t)$ and $\dot{u}_{\text{ANN}}(t)$ are also bounded signals:

$$\begin{cases} |u_{\text{ANN}}(t)| \leq B_u \\ |\dot{u}_{\text{ANN}}(t)| \leq B_{\dot{u}} \end{cases} \quad \forall t, \tag{13}$$

where $B_u > 0$ and $B_{\dot{u}} > 0$ are some known positive constants.

C. Sliding Mode Control Theory-Based Learning Algorithm

A sliding mode control (SMC)-based parameter adaptation scheme is used for the learning process of ANN. The SMC framework as designed by first selecting a suitable sliding manifold that will ensure desired system dynamics. Moreover, it is desired to design a dynamical feedback adaptation mechanism or an online learning algorithm for ANN parameters such that the sliding mode constraints/conditions are fulfilled. SMC provides robustness to parameter uncertainties and external disturbances – thus, is a widely used control method for nonlinear systems applications. The zero dynamics of the learning error coordinate $u_{\text{PD}}(t)$ can be described as a time-varying sliding surface S_{PD} by utilizing the principles of the SMC theory [18]:

$$S_{\text{PD}}(u_{\text{ANN}}, u) = u_{\text{PD}}(t) = u_{\text{ANN}}(t) + u(t) = 0. \tag{14}$$

By using (14), ANN is trained to become a nonlinear regulator which assists the conventional PD controller in parallel so that desired response can be obtained. Hence, the sliding surface for the nonlinear system under control is:

$$S(e, \dot{e}) = \dot{e} + \lambda e, \tag{15}$$

where $\lambda > 0$ is a parameter determining the reference trajectory of the error signal. A sliding motion will appear on the sliding manifold $S_{\text{PD}}(u_{\text{ANN}}, u) = u_{\text{PD}}(t) = 0$ after a finite time t_h, if the condition $S_{\text{PD}}(t)\dot{S}_{\text{PD}}(t) = u_{\text{PD}}(t)\dot{u}_{\text{PD}}(t) < 0$ is satisfied for all t in some nontrivial semi-open subinterval of time of the form $[t, t_h) \subset (-\infty, t_h)$.

The adaptation laws for the parameters of the considered ANN are given as follows:

$$\begin{cases} \dot{w}_i & = -\alpha \frac{\sum_{i=1}^{N_1} x_i}{N_2|\mathbf{x}|} \text{sign}(u_{\text{PD}}) \\ \dot{\alpha} & = \gamma |u_{\text{PD}}| \end{cases}, \tag{16}$$

where $\alpha > 0$ is the adaptive learning rate. The learning error $\tau(t)$ will converge to a small neighbourhood of zero during a finite time t_h for any arbitrary initial condition $\tau(0)$. The reader can refer to [19] for the stability of the proposed learning algorithm.

IV. RESULTS AND DISCUSSION

A. Simulation Studies

At first, the controller is tested in simulation to determine the ANN's meta-parameters like learning rates, number of neurons in the hidden layer, and test different trajectories. Gazebo simulator is used to simulate and model the UAV because of its powerful physics engine. A thorough analysis of the different number of neurons in the hidden layer is done to observe its effects on computation times and learning performance. The computation time is calculated for the different number of neurons, in a simple circular trajectory of 5m radius at desired speed of 1m/s, as shown in Table I. Note that the time given in the table is the average computation time (in ms) taken to run one loop of the ANN controller. It is evident from the study that no significant change is observed in terms of tracking improvement with the increase in the number of neurons, albeit the computation time multiplies.

B. Real-Time Experiments

The real-time tests are conducted on the coaxial hexacopter to validate the performance of the proposed controller. A trajectory with two segments – zig-zag and straight line – is chosen to make the UAV experience both agile and fast manoeuvres at high speeds. In the first segment of the trajectory, the UAV follows a zig-zag path for 55m along x-axis and a periodic change of $\pm 5\text{m}$ along y-axis at a target speed of 5m/s. Then, in the second segment, the UAV follows a straight line path at the target speed of 15m/s for another 70m. Extensive experimentations with various learning rates of ANN were carried out. The results of the ANN controller are then compared with two other controllers. One is the widely known position controller of the autopilot stack – Pixhawk – (referred as PID_{FCU}), while the other a conventional PID position controller (referred as PID_{pos}) sending attitude-setpoints – roll, pitch, yaw angles – and thrust commands. The same experimental scenario is repeated with a different controller each time. Notably, the gains used for the PID controller are tuned; although not for a specific test scenario. The tests are performed in an outdoor environment with the use of RTK GPS, which provides the position information with an accuracy of approximately

TABLE I
COMPARISON OF COMPUTATION TIMES AND EUCLIDEAN ERROR FOR DIFFERENT NUMBER OF NEURONS IN HIDDEN LAYER.

Neurons	3	9	20	50	100	500
Time (ms)	0.092	0.127	0.143	0.245	0.427	2.57
Error (m)	1.55	1.52	1.56	1.55	1.58	1.69

$5-20$ cm. It is to be noted that the experiments were conducted with average wind gusts of 5m/s.

Odroid XU4 is used as the low-cost and low-power onboard computer which runs all the codes in C++ on robot operating system (ROS) – thus, making the system autonomous. The main constraint being the computation power available on the onboard computer, desirable 3 sets of neurons are selected from hardware-in-the-loop simulation to ensure that sensible computation is utilised in real-time. All the sensory feedbacks and controller outputs are fed to the local position estimator of the Pixhawk, which estimates the pose of UAV at 30Hz. The communication with the UAV is achieved with ROS over a 5GHz wireless network.

Remark 1: As aforementioned, the ANN begins to learn online from a pre-set learning rate, each time it is initialized and applies a correction to the model-based techniques. This allows us to keep the original benefits of the control, including stability properties, while the proposed algorithm adds effort to improve performance metrics. Thus, any particular data set for a scenario is not fed to the controller to learn any specific trajectory, rather the controller is designed to perform better in any arbitrary condition. The main goal for the ANN-assisted controller is to learn in a very short time and perform better than the commonly used conventional controllers – thus, a comparison with just an ANN control is not suitable for this application. Moreover, implementing a simple neural network (as used in this case) with just a single hidden layer with very few neurons is not sufficient to learn the complex system dynamics of the UAV. A more complex and interlaced network of hidden layers might be an approach, albeit it is not the scope of this work.

Prior to discussing the final results, a statistical comparison of the controllers is worth mentioning, which also justifies the above-mentioned *Remark 1*. Table II gives a brief overview of the range of experiments carried out for the different controllers numerous times on various trajectories. In particular, a zig-zag path at high speed and a simple circular trajectory at nominal speed are traced. The zig-zag path is a pattern stretching 30m along x-axis and ± 5m along y-axis, while the latter is a circle of 2m radius circling three times. The experiments are performed for each of the three controllers in discussion and are repeated twice for sake of repeatability. The average of each metric is calculated and tabulated in Table II. This shows an overall improvement of the proposed ANN-based controller and that the performance is independent of trajectory chosen. The results from the key experiment devised in this work are discussed next.

TABLE II
STATISTICAL COMPARISON OF EUCLIDEAN ERROR, MEAN ABSOLUTE ERROR (MAE), AND STANDARD DEVIATION (σ).

Trajectory	Controller	Euc. error (m)	MAE (m)	σ
Zig-zag	PID_FCU	1.331	5.378	2.178
	PID_pos	1.022	4.942	1.834
	ANN-PD	0.861	4.550	1.721
Circle	PID_FCU	1.147	1.776	0.396
	PID_pos	1.042	1.617	0.542
	ANN-PD	0.511	0.757	0.299

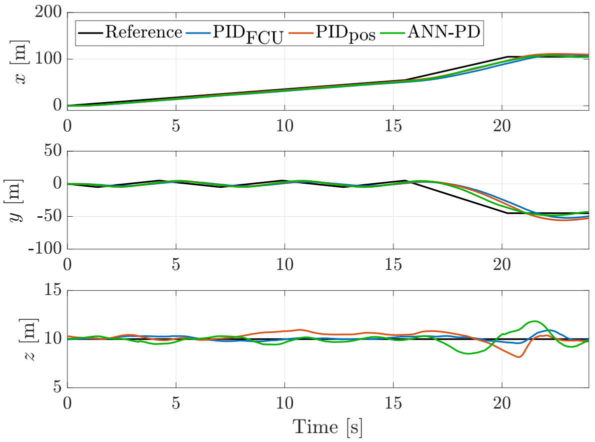


Fig. 5. Real-time trajectory tracking of the UAV.

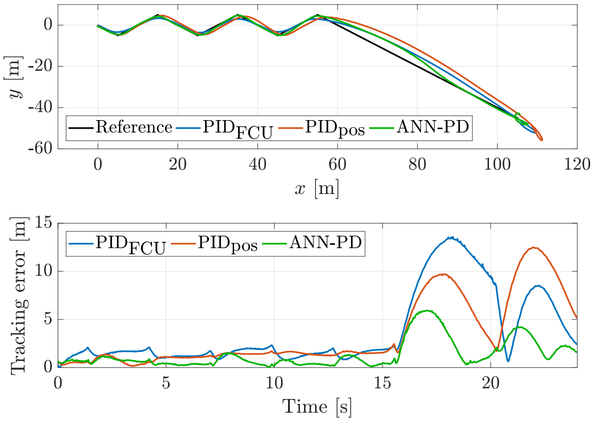


Fig. 6. Top view and tracking error of the considered controllers.

The results plotted in Fig. 5 show the trajectory tracking of the UAV in 3D space over time. The wind gusts and the high speeds of the UAV exert huge stresses on the rotors, thus slight deviations from the trajectory are inevitable. Note that all the iterations with different controllers were carried out in similar outdoor conditions. The slight deviation in z-axis towards the end of the trajectoryis because of the tilting thrust vector of the UAV, reducing the vertical component of thrust compared to the weight of the UAV. As seen from the top view of the trajectory in Fig. 6, that the maximum deviation from the trajectory in case of ANN is about 1m and 5m in any direction for the zig-zag and straight-line parts, respectively. However, for PID_FCU and PID_pos it is about 2.5m and 2m for zig-zag and 13m and 9m for straight-line parts, respectively. Moreover, it is seen that even at high speeds and very sharp turns, ANN is tracking the trajectory to the closest point on the bends.

In the trajectory tracking problem, the Euclidean error is usually calculated to determine the controller's performance, but it may penalize the algorithm as it takes into account the time delay in following the trajectory and not how accurately it is following [20]. Thus, the overall tracking error in x, y, and z axes, plotted in Fig. 6, shows how closely or accurately the actual path is followed despite such tight constraints. The improvement achieved by ANN in terms of tracking error accuracy is 63% and 60% compared to PID_{FCU} and PID_{pos}, respectively. Even on the straight-line part of the trajectory, it converges to the actual trajectory despite the initial deviation. Considering the high speeds and attitude angles attained during the entire 155m long trajectory, the error for the ANN is significantly smaller. Two crests are seen on the plot of tracking error towards the end; first is the straight-line segment of the trajectory, while the second crest is caused due to the trajectory pattern. The UAV is to come to a halt at the end of the straight path when it is traveling close to 18m/s, and physically it is not possible for the UAV to stop in an instant – thus, the overshoot at the end and then the UAV converges to hover states. The ground speed achieved for the different controllers is compared in Fig. 7. The ANN is the fastest to accelerate and complete the trajectory, as shown in the acceleration plot in Fig 7. The ANN-assisted controller is able to maintain stable flight while reaching peak velocities of 18m/s (and above) and attitude angles of $45°$, during the trajectory. Highest average speeds are observed for the ANN during the zig-zag path as ANN follows the trajectory to minimize the tracking error.

Overall, it is shown that ANN accelerates faster to follow the desired trajectory and results in the best trajectory tracking among the three controllers. ANN's learning capability is able to minimize the tracking error over time and provide superior performance. The mean absolute tracking error for each of the three controllers along with the maximum speed and acceleration attained are given in Table III. The experimental video is available at `https://www.dropbox.com/s/sm5ggrt7jg3dk67/video.mp4`.

V. CONCLUSIONS

In this work, an ANN-based controller is proposed to perform three-dimensional aggressive manoeuvres of UAVs. The real-time tests were carried out in the outdoor environment using the RTK GPS for localization. We show that the controller is robust and can be implemented onboard UAV with limited computational resources. Two scenarios are tested in the experiments: agile zig-zag manoeuvres and straight-line fast flight. The ANN controller's trajectory tracking performance is outstanding as compared to conventional PID controllers with about 60% of improvement. Highest acceleration is observed in case of ANN showing that it is the fastest to reach high-speed and complete the trajectory. The experiments were repeated several times and, on each occasion, ANN outperformed the PID controllers, which shows the repeatability of the approach. The coaxial hexacopter used is tested to its physical limits while reaching velocities as high as 20m/s.

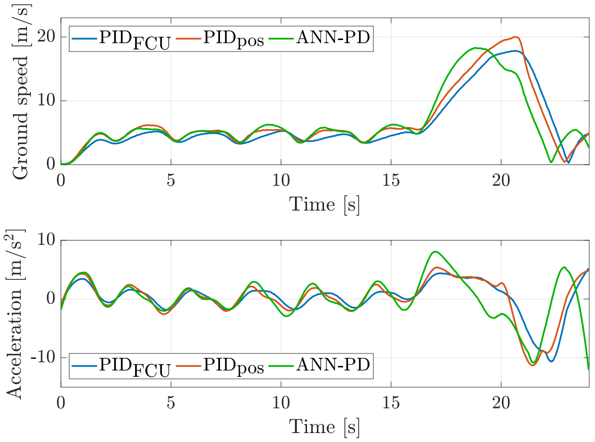


Fig. 7. Ground speed and acceleration of different controllers.

ACKNOWLEDGMENT

This work was partially financially supported by the Singapore Ministry of Education (RG185/17). In addition, this research was also partially supported by the ST Engineering - NTU Corporate Lab through the NRF corporate lab@university scheme.

REFERENCES

[1] G. Loianno, V. Spurny, T. Baca, J. Thomas, D. Thakur, D. Hert, R. Penicka, T. Krajnik, A. Zhou, A. Cho, M. Saska, and V. Kumar, "Localization, grasping, and transportation of magnetic objects by a team of mavs in challenging desert like environments," *IEEE Robotics and Automation Letters*, vol. PP, no. 99, pp. 1–1, 2018.

[2] S. Kim, S. Choi, and H. J. Kim, "Aerial manipulation using a quadrotor with a two dof robotic arm," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Nov 2013, pp. 4990–4995.

[3] T. Tomic, K. Schmid, P. Lutz, A. Domel, M. Kassecker, E. Mair, I. L. Grixa, F. Ruess, M. Suppa, and D. Burschka, "Toward a fully autonomous uav: Research platform for indoor and outdoor urban search and rescue," *IEEE Robotics Automation Magazine*, vol. 19, no. 3, pp. 46–56, Sept 2012.

[4] S. Lupashin, A. Schöllig, M. Sherback, and R. D'Andrea, "A simple learning strategy for high-speed quadrocopter multi-flips," in *2010 IEEE International Conference on Robotics and Automation*, May 2010, pp. 1642–1648.

[5] S. Spedicato, G. Notarstefano, H. H. Bülthoff, and A. Franchi, *Aggressive Maneuver Regulation of a Quadrotor UAV*. Cham: Springer International Publishing, 2016, pp. 95–112.

[6] H. Huang, G. M. Hoffmann, S. L. Waslander, and C. J. Tomlin, "Aerodynamics and control of autonomous quadrotor helicopters in aggressive maneuvering," in *2009 IEEE International Conference on Robotics and Automation*, May 2009, pp. 3277–3282.

[7] A. Pretorius and E. Boje, "Design and modelling of a quadrotor helicopter with variable pitch rotors for aggressive manoeuvres," *IFAC Proceedings Volumes*, vol. 47, no. 3, pp. 12 208 – 12 213, 2014, 19th IFAC World Congress.

[8] D. Mellinger, N. Michael, and V. Kumar, "Trajectory generation and control for precise aggressive maneuvers with quadrotors," *The International Journal of Robotics Research*, vol. 31, no. 5, pp. 664–674, 2012.

TABLE III

MEAN ABSOLUTE ERROR (MAE), MAXIMUM SPEED, AND ACCELERATION ACHIEVED FOR THE CONSIDERED CONTROLLERS.

Controller	PID_{FCU}	PID_{pos}	ANN-PD
MAE (m)	3.749	3.429	1.356
Max. speed (m/s)	17.8	20.0	18.3
Max. acceleration (m/s^2)	5.222	5.373	8.065

[9] D. Mellinger and V. Kumar, “Minimum snap trajectory generation and control for quadrotors,” in *2011 IEEE International Conference on Robotics and Automation*, May 2011, pp. 2520–2525.

[10] O. Purwin and R. D’Andrea, “Performing aggressive maneuvers using iterative learning control,” in *2009 IEEE International Conference on Robotics and Automation*, May 2009, pp. 1731–1736.

[11] A. Bry, C. Richter, A. Bachrach, and N. Roy, “Aggressive flight of fixed-wing and quadrotor aircraft in dense indoor environments,” *The International Journal of Robotics Research*, vol. 34, no. 7, pp. 969–1002, 2015.

[12] S. Shen, Y. Mulgaonkar, N. Michael, and V. Kumar, “Vision-based state estimation and trajectory control towards high-speed flight with a quadrotor.” in *Robotics: Science and Systems*, vol. 1. Citeseer, 2013.

[13] G. Loianno, C. Brunner, G. McGrath, and V. Kumar, “Estimation, control, and planning for aggressive flight with a small quadrotor with a single camera and imu,” *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 404–411, April 2017.

[14] A. Sarabakha and E. Kayacan, “Y6 Tricopter Autonomous Evacuation in an Indoor Environment Using Q-Learning Algorithm,” in *2016 IEEE 55th Conference on Decision and Control (CDC)*, Dec 2016, pp. 5992–5997.

[15] A. Sarabakha, N. Imanberdiyev, E. Kayacan, M. A. Khanesar, and H. Hagras, “Novel levenberg–marquardt based learning algorithm for unmanned aerial vehicles,” *Information Sciences*, vol. 417, pp. 361 – 380, 2017.

[16] Y. Yildiz, A. Sabanovic, and K. Abidi, “Sliding-mode neuro-controller for uncertain systems,” *IEEE Transactions on Industrial Electronics*, vol. 54, no. 3, pp. 1676–1685, June 2007.

[17] M. Ö. Efe, *Sliding Mode Control for Unmanned Aerial Vehicles Research*. Cham: Springer International Publishing, 2015, pp. 239–255.

[18] E. Kayacan, E. Kayacan, and M. A. Khanesar, “Identification of nonlinear dynamic systems using type-2 fuzzy neural networks - a novel learning algorithm and a comparative study,” *IEEE Transactions on Industrial Electronics*, vol. 62, no. 3, pp. 1716–1724, March 2015.

[19] B. K. Wilburn, M. G. Perhinschi, H. Moncayo, O. Karas, and J. N. Wilburn, “Unmanned aerial vehicle trajectory tracking algorithm comparison,” *International Journal of Intelligent Unmanned Systems*, vol. 1, no. 3, pp. 276–302, 2013.